American Option Price Approximation for Real-Time Clearing

ANDREAS BLANCK

MASTER'S THESIS IN ENGINEERING PHYSICS

UMEÅ UNIVERSITY
JANUARY 25, 2018

American Option Price Approximation for Real-Time Clearing

January 2018

Supervisors
Jonas Nylén
Cinnober Financial Technology AB
Anna Filipsson
Department of Physics

Examiner
Markus Ådahl
Department of Mathematics and Mathematical Statistics

Andreas Blanck
anbl0063@student.umu.se
Master's Thesis in Engineering Physics
Umeå University
901 87 Umeå, Sweden

Abstract

American-style options are contracts traded on financial markets. These are derivatives of some underlying security or securities that in contrast to European-style options allow their holders to exercise at any point before the contracts expire. However, this advantage aggravates the mathematical formulation of an option's value considerably, explaining why essentially no exact closed-formed pricing formulas exist. Numerous price approximation methods are although available, but their possible areas of application as well as performance, measured by speed and accuracy, differ. A clearing house offering real-time solutions are especially dependent on fast pricing methods to calculate portfolio risk, where accuracy is assumed to be an important factor to guarantee low-discrepancy estimations. Conversely, overly biased risk estimates may worsen a clearing house's ability to manage great losses, endangering the stability of a financial market it operates.

The purpose of this project was to find methods with optimal performance and to investigate if price approximation errors induce biases in option portfolios' risk estimates. Regarding performance, a Quasi-Monte Carlo least squares method was found suitable for at least one type of exotic option. Yet none of the analyzed closed-form approximation methods could be assessed as optimal because of their varying strengths, where although the Binomial Tree model performed most consistently. Moreover, the answer to which method entails the best risk estimates remains inconclusive since only one set of parameters was used due to heavy calculations. A larger study involving a broader range of parameter values must therefore be performed in order to answer this reliably. However, it was revealed that large errors in risk estimates are avoided only if American standard options are priced with any of the analyzed methods and not when a faster European formula is employed. Furthermore, those that were analyzed can yield rather different risk estimates, implying that relatively large errors may arise if an inadequate method is applied.

Acknowledgements

I wish to offer many thanks to Cinnober Financial Technology AB for the opportunity of being assigned the project that subsequently paved the way for this master's thesis. It has been an exciting time providing enriching experiences which will be to my benefit in the future.

Contents

List of Figures

List of Tables

List of Symbols

Φ	Payoff function
σ	Stock volatility
K	Strike price
r	Risk-free interest rate
S	Stock price
T	Time to maturity
t	Time
V	Option value

Chapter 1

Introduction

1.1 American Options and Risk

Financial markets consist of various traded instruments such as options,[1] stocks and futures to name a few. In this thesis, our focus will be directed towards American-style options, where price estimations become cumbersome since their holders have the right to exercise any time prior to these financial contracts' expirations. As a consequence of this, mathematical formulations of their values involve optimal stopping times, making them more complex than the European-style counterparts where exercise is only possible at the time of expiration [1]. It is therefore hard, if not impossible [2] to derive exact closed-formed pricing formulas, why simulations and analytical approximations need to be applied.[2] Yet this increases the probability of incorrect pricing, something that may affect an option portfolio's estimated risk and hence also capital earmarked to act as a cushion for potential losses incurred from it.

An example of importance to this thesis that may be affected by this problem is a clearing house, which will also be the topic of the following section. It acts as a market's hub by streamlining trades as well as taking on risks inherited from portfolios of great values belonging to large financial institutions. To manage this however, the clearing house acquires collaterals from the institutions that are based on risk estimations. How sensitive estimated portfolio

[1] See Chapter 2 for a formal definition of options.

[2] The American nondividend-payment vanilla call and perpetual options are however special cases with known closed-form pricing formulas [2].

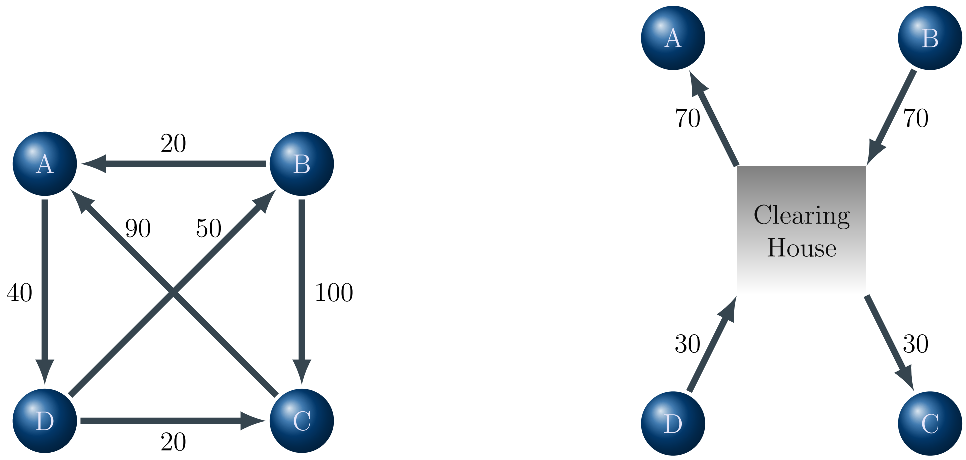
risk is because of incorrect pricing has not been widely investigated since there is a lack of papers available concerning this subject. However, there are reasons to believe it depends on the pricing method chosen, portfolio positions and proportions of American-style options. Should it be sensitive, a clearing house faces the risk of acquiring too little collateral in order to manage a large loss, or conversely, too much which could as well be dangerous since a financial institution may need the faulty excess in its operation. It is then important to find and apply robust as well as reliable methods that can estimate prices with high precision in order to guarantee sound risk estimates.

1.2 Clearing House

A clearing house (CH) is as mentioned a financial infrastructure that plays a crucial role when it comes to risk estimations as well as handling traded instruments like American-style options, and serves to mitigate the risk they induce [3]. Without clearing houses, trades would become much more complicated where two counterparties with matching bids would have to find each other on their own on the market. They would also have to rely on each other that obligations written in the traded contract will be fulfilled [3] and keep track of all their outstanding positions. A simplified example of a market like this is depicted in Figure 1, panel (a), where four market members (A, B, C and D) have various positions to one another. The arrows here show their mutual positions of arbitrary values in each trade. In the event of a defaulting member, an exacerbating effect may arise since it is very likely that it has also performed trades with other members. The result of this is that even though only one member had financial difficulties, it may also cause others to default due to failed incomes needed to cover their own outstanding positions [4]. With the presence of a CH, the issues stated above are resolved since the complex web of positions disappears. In place of one member has to trade directly with another member, it only needs to interact with the CH since it takes the role as the counterparty to all members [3] which we see in Figure 1, panel (b).[3]

Another big benefit with a CH, among others, is that it ensures that obligations will be met and thereby removes the risk that a member makes a loss when its de facto counterpart defaults [3]. However, this implies that the

[3]Accessing membership of the CH is however restricted to financial institutions with great resources and with good creditworthiness in order to reduce the risk of a member defaulting [3].



(a) *Trading without clearing. Each market member has several positions and counterparties.*

(b) *The clearing house acts as a counterparty to all market members and nets a clearing member's outstanding positions.*

Figure 1.1: *Panel (a) shows a market with trades of values marked by arbitrary values, wherein every market member has several counterparties. By introducing a clearing house, the number of counterparties for each member reduces to one as shown in panel (b).*

CH requires a large amount of capital in order to enable trade settlements to counterparties a defaulting member indirectly was responsible for. So to manage potential great losses, a CH has a hierarchical chain of strategies for exploiting its resources in such situations, designated as the default waterfall [3]. It consists of several steps in which capital is utilized, but where the first is the only one associated with this project. This is known as the initial margin, a certain level of collateral members involved in a trade need to transfer to the CH and orientate to, such that potential losses most likely can be covered. It is estimated by considering the worst case scenarios of the trade's involved instruments' values [3, 5]. So, collateral is transferred to the CH in conjunction with a trade, but the risk level of a member's portfolio may alter as its constituting instruments' values change. In reflection of this, additional collateral may hence have to be collected in order to align with the initial margin [5]. Calculations are therefore frequently performed to estimate the risk in a member's portfolio, possibly constituting a broad range of financial instruments, ensuring that a correct amount of collateral is always acquired to prevent great losses [3].

Since a CH plays a central and important role for the stability of the financial markets it operates, it is important that the initial margins and risk calculations are accurate, especially if the managed portfolios are of enormous values. Would these be too low and a member would default, it may force the CH to use capital further down the steps of the default waterfall,[4] or even worse, being unable to settle trades since all possible capital has been drained and in effect default itself. The latter would of course be a significant crisis for the entire market and luckily this has not occured often [3]. The fact that a CH works as a backbone for and facilitates financial markets highlights the importance of reliable risk calculations to improve its ability to cover losses if a member would default.

1.3 Project Purpose

Cinnober Financial Technology AB develops software solutions to clearing houses around the world and therefore also deals with risk calculations. With their real-time clearing solutions, subjects such as initial margins are updated more often than was previously done before. This enables a CH to frequently alter the initial margin to match the level of risk a portfolio of instruments carries, which may vary due to market conditions. The CH can hence reduce the risk of severe losses it may incur since it can acquire more collateral on short notice if necessary, but also allow its members to decrease their collateral, should the level of risk be reduced. This ensures that an adequate amount of collateral is continuously collected and also that each member can use its capital more effectively. To enable this, one must apply accurate pricing methods that are also fast when implemented in a suitable software.

This project aims to help Cinnober investigate and answer what methods are best suited to price American-style options accurately when computation times are limited. Further, they are also interested in how incorrectly priced American-style options affect a portfolio's corresponding risk estimate that is of great importance when determining initial margins. This raises the questions whether erroneously priced options imply significant errors in risk estimates and also if these risk estimates deviates too much from the correct values when a certain percentage of European-style options also comprises a portfolio. This could be of interest since if the measured risk

[4] Something that may affect other clearing members financially even though the purpose of a CH is to prevent this as long as it is possible.

is acceptable given a certain (presumably high) percentage, or even if it is independent of such a level, an incorrect way of estimating prices may enable fast calculations.[5] If risks would be unacceptable no matter what, it is important to adopt a method that ensures the correct amount of risk collateral is collected in light of market conditions. This project's purpose is therefore to explore these questions in order to understand what methods are most appropriate for real-time clearing and how risk is affected by incorrectly priced American-style options.

To answer these questions properly, the outline of the thesis will be as follows; we begin by elaborating the theory regarding both European- and American-style options and why the latter are problematic in order to get a basic overview of the subject we will deal with. Subsequently, we carry on by looking at pricing methods that apply the theory dealt in the foregoing chapter, specifically, we begin with closed-form approximations formulas devoted exclusively to standard/vanilla call and put options. Two numerical methods will then be presented that are applicable to a broad range of American-style options, but which are better suited for the ones with more than one underlying security due to their inability to produce compelling results within a reasonable amount of time. This is then followed by how the pricing methods were implemented and applied for portfolio risk estimations. As a resolution, an analysis of the obtained results is performed, where also the most important findings are stated.

[5]To be clear, a portfolio consisting of European-style options exclusively is assumed to have no or negligible risk estimation error, especially when we only consider vanillas that can be priced exactly with the Black-Scholes formula. So, if a high percentage of the included options in a portfolio is of European-style, the induced error in risk implied by deliberately pricing the remaining American-style the same (although erroneously) and fast way *may* be negligible.

Chapter 2

Option Theory

Preface

As was mentioned at the end of the previous chapter, we will now explore the theory of options in order to shed light on the subject we are about to deal with. Firstly, we will look at the very basic definition and lastly have a brief overview of the mathematical definitions surrounding option pricing that is useful to have seen when we later will look at different pricing methods.[1]

We will start by looking at the theory of European-style options and later that of American-style since they are closely related. In this way we will better understand their differences and why the field of the latter is so problematic.

2.1 Definition

An option is a derivative of an underlying security or combination of securities, e.g. stocks, currencies, interest rates. The *holder* of an option is the part who buys this financial contract from the issuing counterpart denoted as the *writer*. The price paid is determined by the contract's underlying security/securities among other explanatory variables. In return

[1]Remark: Throughout this thesis we will encounter the terms price and value. To be clear, they mean the same thing. Here though, we are only interested in options' values/prices for the purpose of risk calculations *solely*, not selling. The term price may incorrectly suggest the latter and it is therefore good to clarify the project's purpose in detail.

for paying, the holder has the option to exercise (buy or sell the underlying security/securities at a price agreed upon in advance) anytime before or only at the time of maturity, i.e. the contract's expiration, depending on if it is of American- or European-style respectively. The writer is however obligated to fulfill its part written in the contract in compensation for the received payment [4].

As mentioned above, the holder can exercise the option by buying or selling the underlying security or securities at a predetermined price known as the *strike price*. There are two types of options that distinguish this exercise decision:

Call The holder has the right but not the obligation to *buy* the underlying security/securities at the strike price.

Put The holder has the right but not the obligation to *sell* the underlying security/securities at the strike price.

Options may be structured in various styles and with different combinations as was mentioned earlier. The simplest examples are those that only have a single underlying security, why they are sometimes denoted as vanilla options [4]. Exotic options are those that in some ways depend on several underlying securities or for instance, the historic movements of a single security within their lifetimes.[2] These can also be regarded as calls or puts in their nature, where an example of this can be a basket put option which gives the holder the choice of selling the underlying securities constituting the basket to a determined strike price similarly to a vanilla put [6].

The example described above is one of many styles other than plain vanilla financial contracts that are traded and the majority of all options are in fact of American-style [4]. Since it is possible to exercise any time within the lifetime of these options, there is no simple way to price them in contrast to European-styled.

2.2 Pricing Options

2.2.1 European Options

European-style options do not suffer from an optimal exercise decision and hence not a vast pricing complexity since there is only one point in time

[2]A good example is an Asian option whose value depends on the underlying security's average value during the option's lifetime.

of uncertainty concerning the underlying securities, namely the time of maturity. Also, there exists a single known value that governs whether exercise should occur or not, the strike price. These characteristics entail that pricing an European-style option can be done by closed-formed formulas or relatively easy (depending on how accurate you want your solution to be) numerical pricing algorithms.

Perhaps the most applied and known example of this is the Black-Scholes formula for pricing European vanilla call options from 1973 by Fischer Black and Myron Scholes [7]. Following the outlines given in [1], they used a set of preliminary assumptions, omitted here though, and that only two assets primarily constitute the market, as defined in Definition 2.1 known as the Black-Scholes model.

Definition 2.1. Black-Scholes Model *This model is represented by the two processes*

$$dB_t = rB_t dt, \tag{2.1}$$

$$dS_t = \mu S_t dt + \sigma S_t d\bar{W}_t. \tag{2.2}$$

Here B_t is a riskless asset, r is the risk-free interest rate and S is the underlying stock price movement following a geometric brownian motion. The constants μ and σ are the stock's drift rate as well as volatility respectively. Time is denoted as t and $\bar{W}_t$ is a Wiener process, also known as a Brownian motion in physics contexts [4].

Since the Wiener process is crucial for our understanding of the assumed behavior of a stock movement, we also look at its definition as presented by Björk [1], that is

Definition 2.2. Wiener Process *The Wiener process W is defined as*

i. $\bar{W}_0 = 0$

ii. *The differences $\bar{W}_u - \bar{W}_t$ and $\bar{W}_s - \bar{W}_r$ are independent, where $r < s \leq t < u$.*

iii. $\bar{W}_t - \bar{W}_s \sim \mathcal{N}[0, t-s]$ *given $s < t$.*

iv. *The trajectories of $\bar{W}$ are continuous.*

Here $\mathcal{N}[0, t-s]$ is the normal distribution with mean 0 and variance $t-s$.

Based on this, the properties of the Wiener process and that the stock follows a geometric brownian motion (GBM), we have that the stock price is log-normally distributed [4] or formally $\ln(\frac{S_t}{S_s}) \sim \mathcal{N}[(\mu - \frac{\sigma^2}{2})(t-s), \sigma^2(t-s)]$, as well as we can write the stock price movement at a point in time as

$$S_t = S_s e^{(\mu - \frac{\sigma^2}{2})(t-s) + \sigma(\bar{W}_t - \bar{W}_s)}.$$

The equation above is a standard way of modeling stock prices in Monte Carlo simulations and we will in Chapter 4 see a generalization that is used in two of the numerical methods to be presented.

Returning to the derivation, they then introduce an option value function $V_t(S_t)$ contingent on a single underlying stock on this market. By applying Itô's lemma[3] on the stochastic price process of $V_t(S_t)$, they ended up with Black-Scholes equation defined as

Theorem 2.1. Black-Scholes Equation *An option value function $V_t(S_t)$ that satisfies this equation is free from arbitrage possibilities.*

$$\frac{\partial V}{\partial t} + rS\frac{\partial V}{\partial S} + \frac{1}{2}S^2\sigma^2\frac{\partial V^2}{\partial S^2} - rV = 0, \tag{2.3}$$

$$V_T(S_T) = \Phi[S_T], \tag{2.4}$$

where $\Phi[S_T]$ is the contract's payoff function at the time of maturity $t = T$ and serves as the boundary value.

where V's and S's dependence of t have been have suppressed to ease notations [1]. From this partial differential equation (PDE), V can be shown to be

Theorem 2.2. Risk Neutral Valuation *The value of the financial contract under risk neutral condition.*

$$V_t(S_t) = e^{-r(T-t)}\,\mathbf{E}_{t,S}\big[\Phi[S_T]\big] \tag{2.5}$$

Here S is driven by a risk neutral process.

by using the Feyman-Kač proposition [1].[4] Important to note is that the expection value is taken with t and S_t fixed, as to explain the subscripts, and

[3] Itô's lemma describes the stochastic differential of a process that is contingent on another process with a particular stochastic differential [1].

[4] For a thorough discussion surrounding the Feyman-Kač proposition and the general derivation, see [1].

under risk neutral conditions in order to avoid arbitrage[5]. As a consequence of this, the S-process leading to the solution in Equation (2.5) is defined as

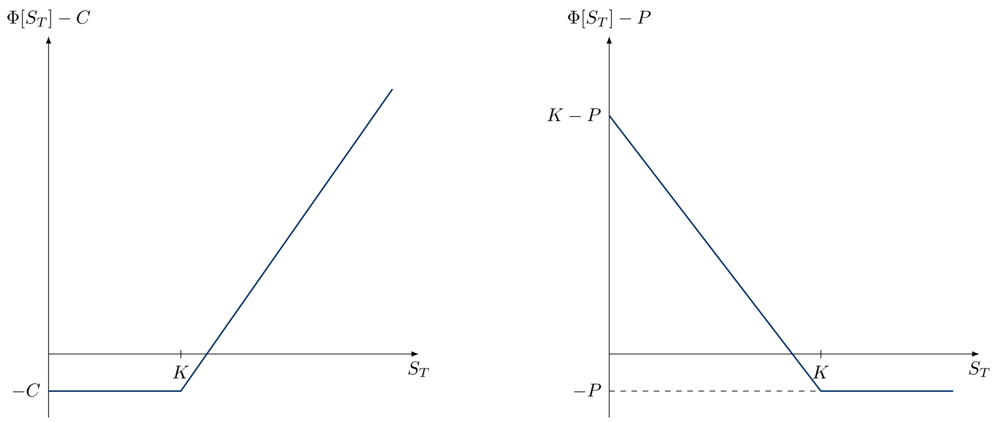
$$dS_t = rS_t dt + \sigma S_t dW_t \tag{2.6}$$

and not as in Equation (2.2) even though this was used to derive the final solution. In Equation (2.6) we also see that there is a new Wiener process, W, under risk-neutral valuation in place of the former $\bar{W}_t$ to notify this fact.

Using Equation (2.5), different types of explicit value functions can be derived, although limited by how complex $\Phi[S_T]$ is. One should remember that this derivation and the PDE was solved for a contract contingent on a single underlying stock and that a d-dimensional problem would require a differential equation that is likely too complex to be solved analytically. Yet, the common thing with one-dimensional problems is that the general solution to every PDE looks the same and as in Equation (2.5), only that the payoff function is now dependent on more than one stock, that is $\Phi[S_T^1, S_T^2, ..., S_T^d]$ [1].

If we return to the simpler vanilla call with strike price K, the payoff function would be defined as $\Phi[S_T] = \max(S_T - K, 0)$ and generate a payoff diagram in resemblance to Figure 2.1, panel (a). In panel (b), we can also see the payoff diagram of a put option with $\Phi[S_T] = \max(K - S_T, 0)$ just to get a visual feeling of how the two option types differ. In panel (a), we see the holder's payoff at the option's time of maturity and also the price paid to the writer when he or she issued it, this clarifies the fact that if the payoff function is zero, there is still an incurred loss of C paid for the contract. The writer's payoff function is intuitively the opposite and would want the stock price to be lower than the strike such that he or she avoids the obligation to sell the underlying stock at a price lower than that can be done on the market. The same arguments also apply to the put option where its price is set to P.

[5] The possibility to make a profit by trading securities that are mispriced [4].



(a) *European vanilla call with strike price K and price C.*

(b) *European vanilla put with strike price K and price P.*

Figure 2.1: *Vanilla European options with their respective payoff functions $\Phi(S_T)$ translated downward by the prices paid to hold the contracts.*

Using the call's payoff function just discussed, Black and Scholes derived in their paper [7] the arbitrage free price of a call, C.

Proposition 2.3. Black-Scholes Formula *The arbitrage free value of a call option $C_t(S_t)$ at time t is*

$$C_t(S_t) = S_t \mathcal{N}\big[d_1\big] - e^{-r(T-t)} K \mathcal{N}\big[d_2\big], \tag{2.7}$$

where $\mathcal{N}[\cdot]$ denotes the standard normal cumulative distribution function. The arguments d_1 and d_2 are defined by

$$d_1 = \frac{1}{\sigma\sqrt{T-t}}\left\{\ln\left(\frac{S_t}{K}\right) + \left(r + \frac{1}{2}\sigma^2\right)(T-t)\right\}, \tag{2.8}$$

$$d_2 = d_1 - \sigma\sqrt{T-t}. \tag{2.9}$$

The Black-Scholes formula is one example of an exact formula for an option and is also used to price put options, with a payoff function as panel (b) in Figure 2.1, by using the put-call parity.[6] When it comes to pricing options on dividend-paying stocks though, the formula in Equation (2.7) is slightly changed, as is the approach to derive it and this was first performed by Merton [8] the same year, 1973. The only difference we have to bother about however, is that we interchange S_t in Equation (2.7) and (2.8) with

[6] An expression that couples the option prices of a put and a call with the same underlying security, defined as $C_t(S_t) - P_t(S_t) = S_t - Ke^{-r(T-t)}$.

$S_t e^{-\delta(T-t)}$ where δ is the continuously compounded dividend yield [4].

In contrast, exotic options may lack closed form solutions as was previously discussed, but since there is only one exercise opportunity and the fact that the risk-neutral valuation formula looks the same, methods such as Monte Carlo simulation can address this problem without that much of a hassle.

2.2.2 American Options

If the complexity of pricing options increases for exotic European options, it is already remarkably high for the vanilla American calls and puts. So why is that? Since exercise is possible any time prior to the time of maturity, there is an imposed constraint that the option value must be in excess of or equal to the value of its payoff function throughout its entire lifetime in order to avoid arbitrage. That is, would the American-style option be erroneously valued and thus not fulfill this, a profit could be made by buying and shortly thereafter exercise it. Just before though, one must net the position by either short sell or buy the underlying security at the market price depending on if it is a call or a put and thus, all these operations then result in an arbitrage payoff [9]. This important constraint must hence be embodied in the problem formulation, aggravating it considerably.

For the one dimensional problem, this constraint alters the Black-Scholes Equation in Equation (2.3) to an inequality since the former boundary condition is supplemented with two more. Because we will be working with dividends henceforward, we jump directly to the modified PDE that includes a constant continuous dividend[7] factor δ seen in the following boundary value problem [1, 9].

Proposition 2.4. *The American one dimensional problem with payoff function* $\Phi[S_t]$ *satisfies*

$$\frac{\partial V}{\partial t} + (r-\delta)S\frac{\partial V}{\partial S} + \frac{1}{2}S^2\sigma^2\frac{\partial V^2}{\partial S^2} - rV \leq 0, \tag{2.10}$$

$$V_T(S) = \Phi[S_T], \tag{2.11}$$

$$V_t(S_f(t)) = \Phi[S_f(t)], \tag{2.12}$$

$$\frac{\partial V(S_f(t))}{\partial S_f(t)} = \frac{\partial \Phi[S_f(t)]}{\partial S_f(t)}, \tag{2.13}$$

where $S \in [0, S_f]$ *for dividend-paying calls and* $S \in [S_f, \infty)$ *for puts and* $S_f(t)$ *is continuously differentiable.*

[7]To see derivation of the Black-Scholes Equation with dividends, see [1].

As we can see, the boundary value problem is dependent on an unknown free boundary, S_f, whose characteristics depends on option-style. This boundary defines a region in which the option is optimally exercised, but since it is unknown, it is practically impossible to find a specific closed-form valuation formula based on Proposition 2.4. Despite this limitation, a general solution exists and whose formulation is in close resemblance to the one for European options seen in Equation (2.5). This solution is seen in the following proposition and is deduced from optimal stopping theory, which however is a subject beyond the scope of this project.

Proposition 2.5. *Based on optimal stopping theory, we have the American option value $V_t(S_t)$ defined as*

$$V_t(S_t) = \sup_{t \leq \tau \leq T} e^{-r(\tau - t)} \, \mathbf{E}_{t,S_\tau} \big[\Phi[S_\tau] \big], \tag{2.14}$$

where the expected value is taken in with respect to risk-neutral valuation and τ is the optimal stopping time defined by

$$\tau = \inf\{t \geq 0 : S_t = S_f(t)\}.$$

Specifically, τ is the first time the underlying stock hits the free boundary [1, 6].

To visualize this, consider an example with the American vanilla call option with dividend payments (since the nondividend-paying call collapses to its European counterpart) and its exercise boundary as in Figure 2.2.

In Figure 2.2, we see a scenario in which the underlying stock price S moves until it hits the free exercise boundary S_f at the optimal stopping time τ, where it is therefore exercised to yield an immediate payoff of $S_\tau - K$. So why is it a good idea to exercise early? Perhaps the most evident example is to consider a put option when its underlying stock price is close to zero. Then it can be favorable to exercise because the maximum possible profit is at hand and waiting any longer increases the risk that the stock price will rise, hence reducing the possible gain [4]. When it comes to call options the discussion becomes less evident and it is only when the underlying stock pays dividends this becomes optimal as previously noted.

Since it can be optimal to exercise early, this extra benefit must be taken into account when valuing the American-style options in order to avoid arbitrage opportunities. The main problem arising as a consequence of this is the unknown free boundary and that the Black-Scholes equation

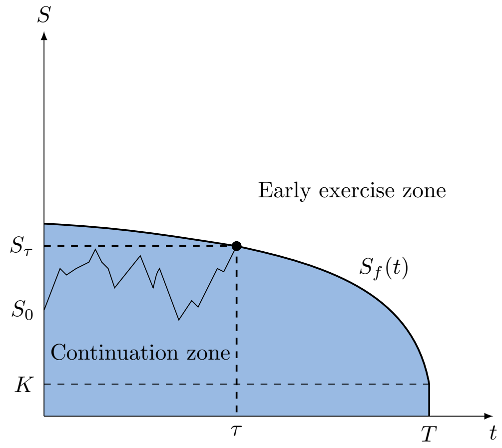


Figure 2.2: *American vanilla call option (on a dividend-paying stock) with strike price K and its optimal exercise boundary $S_f(t)$. We also see its underlying stock's movement and the optimal time of exercise at $t = \tau$. The entire blue shaded area marks the continuation zone where the option is not exercised optimally.*

becomes an inequality. Even more problematic is the case when we look at d-dimensional American option problems and their free boundaries, but as in the European case, the expression in Equation (2.14) that solves the PDE can be applied to these as well. It is only a question of how the payoff function is defined, why $\Phi[S_\tau]$ can be replaced by $\Phi[S_\tau^1, S_\tau^2, ..., S_\tau^d]$ [6] hence giving us a general guideline to find the values of American-style options, something that will be elaborated in the upcoming chapters.

In these chapters we will explore the theory and procedures of specifically chosen closed-form approximation formulas and also three numerical approaches. Two of these numerical methods also extend to the even trickier exotic options to which no known approximation formulas exist, thus making them more flexible. There are however a lot of available models to tackle the pricing problem and a delimitation has been made by considering possible speed, complexity, precision and generality. The candidates that have been found most applicable and fitted these descriptions will now be presented in the subsequent chapters, beginning with closed-form approximations formulas devoted exclusively to options on a single underlying security.

Chapter 3

Closed-Form Approximation Methods

Preface

Here we will look at the best closed-form approximation formulas available today that are limited to American vanilla options. The fact that these pricing methods are analytical makes them attractive from a computational speed point of view compared to numerical methods and after all, we seek methods where the combination of this and precision is optimized.

The methods' full underlying theories are more or less abstract, tedious and are for obvious reasons better to explore in the original papers, why much of it will be omitted. However, basic assumptions in order to understand their approaches are presented and most importantly, their rather awful expressions.

3.1 Bjerksund and Stensland

We begin by looking at an approximation method by Bjerksund and Stensland [10] originally developed in 1993 but later enhanced 2002, why we will only look at the resulting formulas from the latest version in a moment. First however, it is useful to note some basic facts about the two models and their main difference. So, in the two models, they use the same assumptions as in the Black-Scholes methodology but with the supplement of a continuously compounded dividend yield δ as in Merton [8] and they also assume that the option price is generally defined by Equation (2.14). Furthermore, in their first model from 1993 they approximate the true (but still unknown) exercise boundary, S_f, for a call option discussed in the previous chapter with a simple horizontal line. With a simple exercise boundary of this sort, they derived a formula that resembles a European up-and-out call [10]. One might argue that a horizontal line is an exceptionally bad approximation if we compare it to Figure 2.2 in Chapter 2, however, Glasserman [6] actually discusses that the imposed exercise boundary's location and form compared to the true one is not too crucial for a model's precision. Yet, this does not mean that a boundary that has got the shape of the true boundary does not outperform the simpler one.[1]

Based on this fact, the major improvement by Bjerksund and Stensland in their 2002 version was to include another line and separate the two into the time periods I and II at different vertical displacements,[2] together better mimicking the decreasing function we saw in Figure 2.2. Hence we have that the exercise boundary $S_f(t)$ will be defined as

$$S_f(t) = \begin{cases} X, & t \leq t' \\ x, & t' < t \leq T. \end{cases}$$

An example similar to the one given by Bjerksund and Stensland [10] depicting how S_f's two lines are placed is seen in Figure 3.1.

As was mentioned previously, we omit the rather complex theory of this model and therefore jump abruptly to the long expressions resulting from the derivation made by Bjerksund and Stensland as well as stating almost exactly what is given in [10], although with notations matching this thesis. So, to price a vanilla call option with price c we have the main formula

$$\begin{aligned} c = \alpha(X)S_0^{\beta} - \alpha(X)\varphi(S_0, t' \,|\, \beta, X, X) + \varphi(S_0, t' \,|\, 1, X, X) \\ -\varphi(S_{t_0}, t \,|\, 1, x, X) - K\varphi(S_0, t' \,|\, 0, X, X) + K\varphi(S_0, t' \,|\, 0, x, X) + \\ \alpha(x)\varphi(S_0, t \,|\, \beta, x, X) - \alpha(x)\Psi(S_0, T \,|\, \beta, x, X, x, t') + \\ \Psi(S_0, T \,|\, 1, x, X, x, t') - \Psi(S_0, T \,|\, 1, K, X, x, t') \\ -K\Psi(S_0, T \,|\, 0, x, X, x, t') + K\Psi(S_0, T \,|\, \beta, 0, K, X, t'). \end{aligned} \tag{3.1}$$

[1] It is reasonable to believe that a function better resembling the true optimal exercise boundary than a simple horizontal line, or the like, entails a too cumbersome problem to be solved analytically which is a plausible explanation since such models seem to be non-existent.

[2] In all pricing methods presented in the thesis, time is specified in years.

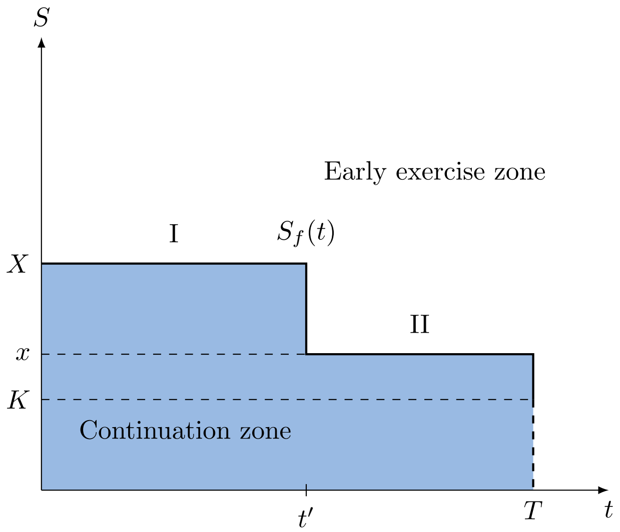


Figure 3.1: *Representation of the approximate exercise region composed of two horizontal lines. The entire shaded area in blue denotes the continuation region.*

Here α and β are defined as

$$\alpha(\tilde{x}) = (\tilde{x} - K)\,\tilde{x}^{-\beta},$$
$$\beta = \left(\frac{1}{2} - \frac{\xi}{\sigma^2}\right) + \sqrt{\left(\frac{\xi}{\sigma^2} - \frac{1}{2}\right)^2 + 2\frac{r}{\sigma^2}},$$

where $\xi = r - \delta$. Further, we have two values defining the exercise boundary lines seen in Figure 3.1;

$$X = B_0 + (B_\infty - B_0)(1 - e^{h(T)}), \tag{3.2}$$
$$x = B_0 + (B_\infty - B_0)(1 - e^{h(T-t')}) \tag{3.3}$$

and Bjerksund and Stensland [10] sets t' to

$$t' = \frac{1}{2}(\sqrt{5} - 1)T.$$

Appearing in the expressions of Equations (3.2) and (3.3) we have

$$B_\infty = \frac{\beta}{\beta - 1}K,$$
$$B_0 = \max\left[K,\ \left(\frac{r}{r - \xi}\right)K\right],$$
$$h(\tilde{t}) = -\left(\xi\tilde{t} + 2\sigma\sqrt{\tilde{t}}\right)\left(\frac{K^2}{(B_\infty - B_0)B_0}\right).$$

We now continue by looking at the two functions φ and Ψ as well as their auxiliary functions, beginning with φ defined as

$$\begin{aligned}\varphi(S_0, T \mid \gamma, H, X) = e^{\lambda T} S_0^{\gamma} \Bigg(\mathcal{N}\Bigg[-\frac{\ln(S_0/H) + (\xi + (\gamma - \frac{1}{2})\sigma^2)T}{\sigma\sqrt{T}} \Bigg] \\ - \Big(\frac{X}{S_0}\Big)^{\kappa} \mathcal{N}\Bigg[-\frac{\ln(X^2/(S_0 H)) + (\xi + (\gamma - \frac{1}{2})\sigma^2)T}{\sigma\sqrt{T}} \Bigg] \Bigg).\end{aligned} \tag{3.4}$$

In Equation (3.4), λ and κ are defined by

$$\begin{aligned}\lambda &= -r + \gamma\xi + \frac{1}{2}\gamma(\gamma - 1)\sigma^2, \\ \kappa &= \frac{2\xi}{\sigma^2} + (2\gamma - 1),\end{aligned}$$

respectively. With this in mind, we are now ready to look at the final function Ψ defined as

$$\begin{aligned}\Psi(S_0, T \mid \gamma, H, X, x, t') = e^{\lambda T} S_0^{\lambda} \Bigg(\mathcal{N}_1\Big[[d_1, D_1]\Big] - \Big(\frac{X}{S_0}\Big)^{\kappa} \mathcal{N}_1\Big[[d_2, D_2]\Big] \\ - \Big(\frac{x}{S_0}\Big)^{\kappa} \mathcal{N}_2\Big[[d_3, D_3]\Big] + \Big(\frac{x}{X}\Big)^{\kappa} \mathcal{N}_2\Big[[d_4, D_4]\Big] \Bigg),\end{aligned}$$

where we have the two 2-D multivariate normal cumulative distribution functions $\mathcal{N}_1[\cdot]$ and $\mathcal{N}_2[\cdot]$. These are distributed as $\mathcal{N}_1[\vec{0}, \Sigma_1]$ respectively $\mathcal{N}_2[\vec{0}, \Sigma_2]$ with covariances defined as

$$\Sigma_1 = \begin{pmatrix} 1 & \sqrt{\frac{t'}{T}} \\ \sqrt{\frac{t'}{T}} & 1 \end{pmatrix} \quad \text{and} \quad \Sigma_2 = \begin{pmatrix} 1 & -\sqrt{\frac{t'}{T}} \\ -\sqrt{\frac{t'}{T}} & 1 \end{pmatrix}.$$

After all these tedious expressions, we conclude by looking at the final ones appearing as arguments of the 2-tuple vectors evaluated in the two multivariate normal cumulative distributions, that is

$$
\begin{aligned}
d_1 &= -\frac{\ln(S_0/x) + (\xi + (\gamma - \frac{1}{2})\sigma^2)t'}{\sigma\sqrt{t'}}, \\
d_2 &= -\frac{\ln(X^2/(S_0 x)) + (\xi + (\gamma - \frac{1}{2})\sigma^2)t'}{\sigma\sqrt{t'}}, \\
d_3 &= -\frac{\ln(S_0/x) - (\xi + (\gamma - \frac{1}{2})\sigma^2)t'}{\sigma\sqrt{t'}}, \\
d_4 &= -\frac{\ln(X^2/(S_0 x)) - (\xi + (\gamma - \frac{1}{2})\sigma^2)t'}{\sigma\sqrt{t'}}, \\
D_1 &= -\frac{\ln(S_0/(H)) + (\xi + (\gamma - \frac{1}{2})\sigma^2)T}{\sigma\sqrt{T}}, \\
D_2 &= -\frac{\ln(X^2/(S_0 H)) + (\xi + (\gamma - \frac{1}{2})\sigma^2)T}{\sigma\sqrt{T}}, \\
D_3 &= -\frac{\ln(x^2/(S_0 H)) + (\xi + (\gamma - \frac{1}{2})\sigma^2)T}{\sigma\sqrt{T}}, \\
D_4 &= -\frac{\ln(S_0 x^2/(H X^2)) + (\xi + (\gamma - \frac{1}{2})\sigma^2)T}{\sigma\sqrt{T}}.
\end{aligned}
$$

With all expressions given above, we are now able to price American vanilla call options, but in order to price corresponding put options, Bjerklund and Stensland [10] introduce a transformation defined as

$$
p(S_0, K, T, r, \xi, \sigma) = c(K, S_0, T, r - \xi, -\xi, \sigma),
$$

where p is the put price. The input parameters seen above implicitly alter the exercise boundary to be an increasing function and of course also the payoff function in contrast to what characterizes a call option.

We thus have explicit expressions enabling us to price both call and put options without the need of simulation, which is a good thing when many types of vanilla options have to be priced in a short amount of time as is the case of portfolio risk estimation in real-time clearing. Since time and precision are desirable, we will look at another two methods for comparison purposes and where we begin with the one developed by Barone-Adesi and Whaley.

3.2 Barone-Adesi and Whaley

In this model from 1987 by Barone-Adesi and Whaley [11] lies Black and Scholes' model assumptions once again as a foundation for the derivation. But the distinction here compared to the Bjerksund-Stensland model is that Barone-Adesi and Whaley do not impose an exercise boundary in advance, rather, they start by assuming that both the European and American vanilla options follow the same partial differential equation.

To be more concrete, remember the PDE we saw in the previous chapter for American options; the equality was to capture the behavior of the European case, whereas the inequality handles the American characteristics of the option. The latter is because there might be a point in time where early exercise is possible, hence invoking the potential premium value it implies. Barone-Adesi and Whaley, however, work only with the equality and discusses that

$$\epsilon(S, t') = V_{t'}^{A}(S) - V_{t'}^{E}(S) \tag{3.5}$$

also follows this PDE and where now $t' = T - t$, $V_{t'}^{A}(S)$ is the American option price and $V_{t'}^{E}(S)$ its European counterpart. Suppressing $\epsilon(S, t')$'s dependence of t' and S, we have

$$\frac{\partial \epsilon}{\partial t'} + \xi S \frac{\partial \epsilon}{\partial S} + \frac{1}{2} S^2 \sigma^2 \frac{\partial \epsilon^2}{\partial S^2} - r\epsilon = 0.$$

They then perform a few transformations of this PDE and state that the solution can be written in the form of

$$\epsilon(S_{t'}, T) = h(t') f(S, t') \tag{3.6}$$

as well as setting $h(t') = 1 - e^{-rt'}$. Since this is the most crucial part in their approach, the derivations that follow are omitted to save space. Nevertheless, they end up with a PDE where they are able to neglect a term that, based on the properties of Equation (3.6), approaches zero in the limit where the option either have got a very short or very long time to maturity.

Because this term vanishes, what is left is a second-order ordinary differential equation (ODE) which, regarding the complexity of American options, is relatively easy to solve. The solution to that ODE is then substituted in Equation (3.5), solved for $V_{t'}^{A}(S)$ and subject to the boundary conditions we saw in the previous chapter. Nevertheless, using these boundary conditions the problem of an unknown boundary (stock price) defined as S_f still remains.[3]

One might think that we are back to where we started, but using the boundary conditions and the fact that the option is exercised exactly when the stock price hits S_f, they end up with the nonlinear equation

$$\phi(S_f - K) = V^E_{t'}(S_f) + \phi\big[1 - e^{-\delta t'}\mathcal{N}[\phi d(S_f)]\big]\,\frac{S_f}{\lambda}, \tag{3.7}$$

where

$$\begin{aligned} d(S^f) &= \frac{\ln(S_f/K) + (\xi - \frac{\sigma^2}{2})t'}{\sigma\sqrt{t'}} \\ \lambda &= \big[-(\beta-1) + \phi\sqrt{(\beta-1)^2 + 4\frac{\alpha}{h}})\big]/2. \end{aligned}$$

Here $\alpha = \frac{2r}{\sigma^2}$, $\beta = \frac{2\xi}{\sigma^2}$, $\phi = 1$ for calls and $\phi = -1$ for puts. So with Equation (3.7) we are not lost, on the contrary, we have an equation that can be solved implicitly by using some relatively easy numerical method. Important to note here is that the exercise boundary S_f is not the same for the put and call as previously mentioned, this should be apparent by the different expressions (based on ϕ) from which they are solved. However, the fact the same notation is used for the two may raise confusion and is therefore good to bear in mind.

At this point, it is worthwhile to highlight the distinction between the two until now presented models where Bjerksund and Stensland from the very start impose a specific form of the exercise boundary. In contrast, Barone-Adesi and Whaley instead obtain a boundary value S_f implicitly from Equation (3.7) as a consequence of their different approach. So, once this boundary value has been obtained numerically, we obtain the American option price using the solution for $V^A_{t'}(S)$ defined as

$$V^A_{t'}(S) = \begin{cases} V^E_{t'}(S) + A(S/S_f)^\lambda, & \phi(S_f - S) > 0 \\ \phi(S - K), & \phi(S_f - S) \leq 0 \end{cases} \tag{3.8}$$

and where $A = \phi\big(\frac{S_f}{\lambda}\big)\Big[1 - e^{\delta t'}\mathcal{N}[\phi d(S_f)]\Big]$.

We now have expressions with which we can calculate American option prices that although require us to calculate S_f by Equation (3.7) in advance. To

[3] Previously S_f denoted a decreasing (increasing) function of time for the American call (put). In this model it is just a single stock price and not a function, but since it has got the same meaning, we stick to the same notation.

do that, we also need to make an initial guess and Barone-Adesi and Whaley use the following for a call option

$$S_f = K + [S_f(\infty) - K][1 - e^{h_\infty}]$$

where

$$\begin{aligned} S_f(\infty) &= \frac{K}{1 - 1/\lambda_\infty} \\ \lambda_\infty &= \big[-(\beta - 1) + \phi\sqrt{(\beta-1)^2 + 4\alpha}\big]/2. \\ h_\infty &= -\phi(\xi t' + 2\sigma\sqrt{t'})\Bigg[K/[\phi(S_f(\infty) - K)]\Bigg]. \end{aligned}$$

For the put option, we instead use the similar expression

$$S_f = S_f(\infty) + [K - S_f(\infty)][1 - e^{h_\infty}].$$

Thus, we can with the formulas above price American vanilla options relatively easy and fast even though the method requires us to find S_f numerically, which by all means are no heavy calculations. In addition, the method is based on the fact that a term is omitted as a consequence of being close to zero when using either a very short or long time to maturity, why it should provide good estimations in those specific cases. An estimated price when using a time maturity that is in between those limits might not be as accurate and we will therefore move on to the closely related method by Ju and Zhong formed to address this issue.

3.3 Ju and Zhong

Here we have a pricing method that enhances the model by Barone-Adesi and Whaley and which was presented by Ju and Zhong [12] in 1999. Specifically, they begin by assuming that the difference in Equation (3.5) can be described by the PDE from the previous chapter using only an equality, that is, in accordance with the Barone-Adesi and Whaley model. They then assume that the solution can be written in resemblance to Equation (3.6), but it is also here the two methods differ.

Ju and Zhong however, set that $f(S, t') = f_1 + f_2$ where the case of $f_2 = 0$ entails the same solution as in the previous model. Keeping this supplemental function though, it will serve to correct the solution implied by $f_1 = A(S/S_f)^\lambda$ mainly in between the limits where the discussed term in the PDE does not vanish. They then set $f_2 = \varepsilon f_1$ where ε is a correction parameter and assumed to be a small number. Using this, they end up with a new PDE where they neglect a derivative term involving ε as well as making the simplification that $1+\varepsilon$ is constant. By using these approximations, they obtain an ODE slightly different to the one of the Barone-Adesi and Whaley model as a consequence of including f_2. Using the solution for this ODE, plugging it into Equation (3.5) and solving for $V_{t'}^A(S)$ as before we get

$$V_{t'}^A = \begin{cases} V_{t'}^E(S) + \frac{hA(S/S_f)^\lambda}{1-b(\ln(S/S_f))^2 - c\ln(S/S_f)}, & \phi(S_f - S) > 0 \\ \phi(S-K), & \phi(S_f - S) \leq 0, \end{cases} \tag{3.9}$$

where $hA = \phi(S_f - K) - V_{t'}^E(S_f)$. Further we have

$$\begin{aligned} b &= \frac{(1-h)\alpha\lambda'}{2(2\lambda+\beta-1)}, \\ c &= -\frac{(1-h)\alpha}{2\lambda+\beta-1}\Big[\frac{1}{hA}\frac{\partial V^E(S_f)}{\partial h} + \frac{1}{h} + \frac{\lambda'}{2\lambda+\beta-1}\Big], \\ \lambda' &= -\frac{\phi\alpha}{h^2\sqrt{(\beta-1)^2 + 4\frac{\alpha}{h}}}, \end{aligned}$$

$$\frac{\partial V^E(S_f)}{\partial h} = \frac{S_f n[d(S_f)]\sigma e^{\xi t'}}{2r\sqrt{t'}} - \phi\delta S_f \mathcal{N}[\phi d(S_f)]e^{\xi t'}/r + \phi K \mathcal{N}\big[\phi(d(S_f) - \sigma\sqrt{t'})\big]$$

and $n[\cdot]$ is the normal probability density function. Finally, the last unknown here is the exercise boundary S_f and it is once again found by solving Equation (3.7) numerically due to simplifications made by Ju and Zhong [12].[4]

However, these expressions do not apply when $r = 0$, why we have the following formulas for a call in that scenario;

$$\begin{aligned} \lambda &= \Big[-(\beta-1) + \phi\sqrt{(\beta-1)^2 + \frac{8}{\sigma^2 t'}}\Big]/2, \\ b &= -\frac{2}{\sigma^4 t'^2\big[(\beta-1)^2 + \frac{8}{\sigma^2 t'}\big]}, \end{aligned}$$

[4]Conveniently, this also means that we use the same initial guesses as in the Barone-Adesi and Whaley model.

$$c = -\frac{\phi}{\sqrt{(\beta-1)^2+\frac{8}{\sigma^2 t'}}}\left(\frac{S_f n[d(S_f)]e^{-\delta t'}}{hA\sigma\sqrt{t'}} - \frac{\phi 2\delta S_f \mathcal{N}[\phi d(S_f)]e^{-\delta t'}}{hA\sigma^2}\frac{2}{\sigma^2 t'} - \frac{4}{\sigma^4 t'^2\big((\beta-1)^2+\frac{8}{\sigma^2 t'}\big)}\right).$$

Despite this special case for a call option, Ju and Zhong [12] discuss that it should be sufficient to use the original expressions with an interest rate close to zero, i.e. the limit value.

To summarize, by including the correction parameter f_2 instead of using merely f_1 corresponding to the function in Equation (3.6), this method better approximates prices in regions where the assumptions made by Barone-Adesi and Whaley are weak, that is in situations where times to maturities are not very long or short. Further, since this model is based on Barone-Adesi Whaley, it should perform about equally well regarding speed and perhaps even better when it comes to reliability in precision.

Common for all methods presented in this chapter is the lack of ability to price high-dimensional and exotic American options, a subject that aggravates the complexity even more as we already know. We will therefore move on and look at a pair of methods that can handle this, but which use numerical schemes instead of closed-form analytical expressions.

Chapter 4

Numerical Methods

Preface

Throughout this chapter we will look at several numerical methods to which we can approximate Bermudan options. These are characterized by a limited amount of exercise opportunities compared to the infinitively many of American options within their respective lifetimes. However, using numerical methods to price the latter require us to discretize time and consequently also the spacing between exercise opportunities. In addition, increasing the amount of exercise opportunities (i.e. using smaller time steps to mimic continuous time) will logically improve the following pricing methods' estimates and thus approach the true values, but unfortunately to the price of heavier computations.

Two of methods presented below are possible to apply for both exotic and vanilla options, but mostly for the former due to their inability to perform well in precision and speed simultaneously.[1] Before we explore these however, we will take a look at the Binomial Tree method limited to vanilla options.

[1]Since we have already looked at closed-form approximation formulas that intuitively are fast, we consider the numerical methods exclusively suited for exotic options where approximation formulas are non-existent.

4.1 Binomial Tree

Even though the primary focus of the selected numerical methods is to tackle the more complex exotic options, we will here begin with a method limited to vanilla options that is also relatively simple, namely the Binomial Tree by Cox, Ross and Rubinstein [13] from 1979. This method is commonly used as a benchmark in papers presenting new option pricing methods since its estimates converge to the true values when using infinitesimal small time steps [4], that is, when the number of exercise opportunities approaches infinity as a genuine American-style option. However, even if the Binomial Tree estimates are often regarded as true values when the time step is sufficiently small ($\sim \frac{1}{10\,000}$), they come at a high price; long computational times. This is a consequence of the trees' exponential growth, something that is not appropriate when speed is an important factor. We will briefly look at this model as presented by Hull [4] since it will also here work as a benchmark to vanilla calls and puts in the analyses to come.

The simple idea of the method is that the initial stock price S_{t_0} either increases or decreases with growth factors u and d respectively ($u > d$), where $d = \frac{1}{u}$. This results in that the tree recombines meaning that, given the initial price, an upward movement followed by a downward equals a downward movement followed by a upward. There are also assigned probabilities to the events occurring, where

$$\begin{cases} u, & \text{with probability} \quad p \\ d, & \text{with probability} \quad 1-p. \end{cases}$$

Similar to the Black-Scholes model we must use risk-neutral valuation and extend this to the idea that the stock price evolves with a drift rate equal to the risk-free interest rate r minus any possible dividend yield δ in order to avoid arbitrage possibilities [4]. We therefore impose the restriction that the discrete expectation value of the growth rate described by the up and down factors matches the continuously compounded arbitrage-free rate in the following way

$$pu + (1-p)d = e^{(r-\delta)\Delta t}, \tag{4.1}$$

where $\Delta t = t_{i+1} - t_i$ is the discretized time step. Conversely, would we have an up factor return of $u < e^{(r-\delta)\Delta t}$, we could short sell the stock and invest it at the risk-free interest rate r, or contrary, if $d > e^{(r-\delta)\Delta t}$ we could borrow money and invest it in the stock. That is, both cases yield unwanted arbitrage payoffs.

Hull [4] uses Equation (4.1), reshuffles it and finds that new expression's variance is

$$pu^2 + (1-p)d^2 - e^{2(r-\delta)\Delta t}.$$

Assuming also that the stock price is log-normally distributed as discussed in Chapter 2, we match the variance defined there with the new expression's equivalent in the following way

$$pu^2 + (1-p)d^2 - e^{2(r-\delta)\Delta t} = \sigma^2 \Delta t,$$

where σ is the volatility of the underlying stock. Given these statements, it is possible to derive the important parameters needed to construct the tree which is defined in [4] accordingly

$$p = \frac{a-d}{u-d} \tag{4.2}$$

$$u = e^{\sigma\sqrt{\Delta t}} \tag{4.3}$$

$$d = e^{-\sigma\sqrt{\Delta t}} \tag{4.4}$$

$$a = e^{(r-\delta)\Delta t}, \tag{4.5}$$

from which we can construct the tree by

$$S_{t_i}^{j,i-j} = S_{t_0} u^j d^{i-j}, \quad j = 0, 1, ..., i, \tag{4.6}$$

where i is the time step index. Using Equation (4.6) with the auxiliary expressions shown in Equations (4.2-4.5), we are able to construct a tree similar to the one seen in Figure 4.1.

Once we have constructed the tree of stock prices, we consider the option values belonging to the nodes at the time of maturity, t_m, as known and equal to the payoff function $\Phi[S_{t_m}^{j,i-j}]$. In the example shown in Figure 4.1, we therefore set the node values to $\Phi[S_{t_3}^{j,i-j}]$. At times $t < t_m$, we proceed using dynamic programming by working through the tree from the end with the known values toward the root node and sought option price at t_0 by using

$$V_{t_i}^{j,i-j} = \max\Big[\Phi\big[S_{t_i}^{j,i-j}\big], e^{-r\Delta t}\,\mathbf{E}\big[V_{t_{i+1}} |\, S_{t_i}^{j,i-j}\big]\Big]. \tag{4.7}$$

In this equation, it is worthwhile to point out the expression $e^{-r\Delta t}\,\mathbf{E}\big[V_{t_{i+1}} |\, S_{t_i}^{j,i-j}\big]$, which is known as the continuation value and is defined as the discounted expected value of the two option values emanating from node $\{j, i-j\}$ given $S_{t_i}^{j,i-j}$. That is, it is the option's present value as a consequence of refraining immediate exercise and thereby holding it until

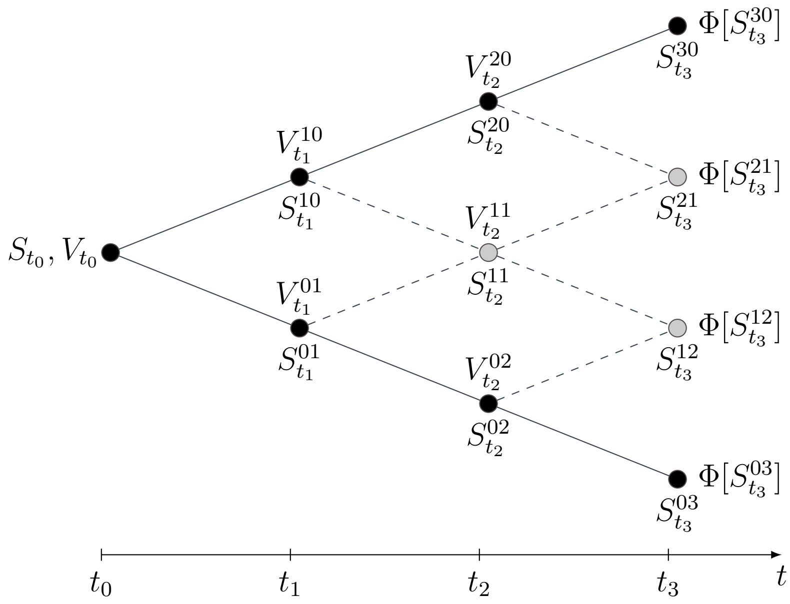


Figure 4.1: *The figure depicts a binomial tree with three time steps. The interior nodes which are highlighted with grayish color is to emphasize the fact that the stock values at these nodes are already calculated since the tree recombines. Remark: The superscripts denote number of up and down steps.*

the following time step.

After having seen the example above it is worth to remark a key feature of the specific dynamic programming employed in this and the models to come, namely that we do not have to impose an unknown and presumingly complicated exercise boundary, nor do we need to find one implicitly. It is simply not necessary when we model the underlying stock(s) and this points out a big difference compared to the analytical approximation methods elaborated in the previous chapter. Returning to the continuation value, we will in the two upcoming numerical models see that this concept is also crucial, but where the ways we evaluate it differ. In this model though, we utilize the up and down probabilities shown above and assign those to the two option values at time t_{i+1} emanating from the node in question at t_i. In Example 4.1.1, we see how we apply these probabilities to estimate the expectation values when we as here work with discrete events.

Example 4.1.1. *Continuation value.* Consider Figure 4.1 and specifically node 20. Assume that we know the probability of an up move p and we want to use Equation 4.7 to find the node value $V_{t_2}^{20}$. Since the terminal nodes 30 and 21 are known ($V_{t_3}^{30} = \Phi[S_{t_3}^{30}]$ and $V_{t_3}^{21} = \Phi[S_{t_3}^{21}]$) we estimate $V_{t_2}^{20} = \max\left[\Phi\left[S_{t_2}^{2,0}\right], e^{-r\Delta t}\left[pV_{t_3}^{30} + (1-p)V_{t_3}^{21}\right]\right]$, revealing how we can calculate a continuation value. □

If the option would be an European vanilla call or put, the continuation statement would be the only thing we would have to bother about when we recursively work toward the root node since immediate exercise is not allowed. Yet, since we are analyzing American options, there might be nodes where early exercise is favorable and we must therefore keep track of the larger of this premium value and the discussed continuation value, explaining the necessity of a maximum function in Equation (4.7).

4.2 Random Tree

This Monte Carlo method by Broadie and Glasserman [14] from 1997 creates a non-recombining tree since the tree nodes represented by the underlying stock(s) price movements are randomly generated by using a geometric brownian motion (GBM) formula in contrast to the deterministic up and down probabilities we saw earlier in the Binomial Tree. Consequently, a large number of trees and thereby tree estimates must be replicated in order to approximate the true value by the law of large numbers. Another difference is that this method allows an arbitrary number of branches, b, emanating from every node, making the tree potentially extremely large.

To calculate the value of the option, we begin at the time of maturity, t_m and work backward in time in a similar fashion as in the Binomial Tree method. Here though, we use a generalized version of the dynamic programming expression we saw earlier, defined as

$$V_t(\mathbb{S}_t) = \max\left[\Phi\left[\mathbb{S}_t\right], e^{-r\Delta t}\,\mathbf{E}\left[V_{t+1}(\mathbb{S}_{t+1})|\,\mathbb{S}_t\right]\right], \tag{4.8}$$

to account for options contingent on multiple underlying stocks. Hence we use the notation $\mathbb{S}_t = \{S_t^1, S_t^2, ..., S_t^d\}$ where d is the number of underlying stocks.

4.2.1 Generating Stock Prices

To create stock price paths for every underlying stock in every replication, we will apply a generalized version of the GBM we saw in Chapter 2 (that also invokes dividend yields) and under risk-neutral valuation.[2] This means that each stock will have drift rate of $r - \delta^{\beta}$, where $\beta = 1, 2, ..., d$ denotes which stock. With this in mind, the generalized GBM takes the form of

$$S_{t_i}^{\beta} = S_{t_{i-1}}^{\beta} e^{(r-\delta^{\beta}-\frac{1}{2}\sigma_{\beta}^2)(t_i-t_{i-1})+\sqrt{t_i-t_{i-1}}\sum_{k=1}^{d} A_{\beta k}\,\mathbb{Z}_{k,i}}, \tag{4.9}$$

where $i = 1, 2, ..., m$. Further on, we will assume that time is discretized uniformly by replacing $t_i - t_{i-1}$ with Δt. The vector $\mathbb{Z}_i$ contains independent and identically distributed (*i.i.d.*) random variables $\mathbb{Z}_i = [Z_{1,i}, Z_{2,i}, ..., Z_{d,i}]^T \sim \mathcal{N}[\vec{0}, \mathrm{I}]$, where $\vec{0}$ is the zero vector and I is the identity matrix. Moreover, A is the Cholesky matrix capturing the stocks' mutual correlations, ρ, and is derived from the relation

$$AA^T = \Sigma = \begin{pmatrix} \sigma_1^2 & \dots & \rho_{1d}\sigma_1\sigma_d \\ \vdots & \ddots & \vdots \\ \rho_{d1}\sigma_1\sigma_d & \dots & \sigma_d^2 \end{pmatrix} \tag{4.10}$$

by using the positive definite covariance matrix Σ of the stocks' returns [6]. The lower triangular Cholesky matrix A is thus the multivariate equivalence of volatility we have to apply when we generate paths for several, possibly correlated, stocks simultaneously.

We will soon see how we can construct a random tree of stock prices generated by the GBM in Equation (4.9) and of course also its corresponding option *values* at time t_0, which in fact are two for every replication. For starters, we must therefore understand how these two values are obtained and why they are needed.

4.2.2 Low and High Estimator

It may sound strange that we will need to calculate two prices for every replication in order to estimate the true value V of the option, but valuing the option as we did in the Binomial Tree will induce a high bias $\hat{V}$, which

[2] Since the general solution to the PDE describing an American-style option's value we saw in Chapter 2 lies as a foundation for the numerical methods in this chapter and the fact that it is under risk-neutral valuation, the underlying stocks must have drifts equal to the risk-free interest rate, subtracted by possible stock-specific dividend yields, when we generate their respective paths.

is why we also need a low bias estimator $\hat{v}$ to create an unbiased confidence interval [14]. The expression in Equation (4.8) is although valid for both the high and low estimator, but their strategies as to estimate the continuation value in it differ which we will soon see. Remark: since we will work recursively with estimated values, hats has been introduced to distinguish them from the true values.

Before jumping directly to these strategies, we introduce similar notation as Glasserman [6], where $j_1 j_2 ... j_m$ and $j_i \in \{1, 2, ..., b\}$, in order to trace every node from the overall parent node at t_0. Since we will work backward in time, our "initial values" will be the payoff function values at the time of maturity, t_m, precisely as in the Binomial Tree and these are the same for both the high and low estimator, i.e.

$$\hat{V}_{t_m}^{j_1 j_2 \cdots j_m} = \hat{v}_{t_m}^{j_1 j_2 \cdots j_m} = \Phi\left[\mathbb{S}_{t_m}^{j_1 j_2 \cdots j_m}\right]. \tag{4.11}$$

For $t < t_m$ however, the high estimator is similar to what we saw in the Binomial framework only that we do not work with probabilities anymore, that is

$$\hat{V}_{t_i}^{j_1 j_2 \cdots j_i} = \max\left[\Phi\left[\mathbb{S}_{t_i}^{j_1 j_2 \cdots j_i}\right], \frac{1}{b} e^{-r\Delta t} \sum_{j=1}^{b} \hat{V}_{t_{i+1}}^{j_1 j_2 \cdots j_i j}\right] \tag{4.12}$$

and it is possible to show that this expression is biased high by using Jensen's inequality,[3] but the proof is formally given in [6].

To adjust for the inconvenience of a biased high-estimator, the low estimator for $t < t_m$ is set to

$$\hat{v}_{t_i,k}^{j_1 j_2 \cdots j_i} = \begin{cases} \Phi\left[\mathbb{S}_{t_i}^{j_1 j_2 \cdots j_i}\right] & \text{if } \frac{1}{b-1} e^{-r\Delta t} \sum_{j=1,\, j\neq k}^{b} \hat{v}_{t_{i+1}}^{j_1 j_2 \cdots j_i j} \leq \Phi\left[\mathbb{S}_{t_i}^{j_1 j_2 \cdots j_i}\right], \\ e^{-r\Delta t}\, \hat{v}_{t_{i+1}}^{j_1 j_2 \cdots j_i k} & \text{else,} \end{cases} \tag{4.13}$$

where $k = \{1, 2, ..., b\}$. This is this done b times where the continuation value is either the omitted discounted value one time step ahead or the immediate exercise value at the current time step depending on the condition stated in Equation (4.13). These b values are then averaged simply as

$$\hat{v}_{t_i}^{j_1 j_2 \cdots j_i} = \frac{1}{b} \sum_{k=1}^{b} \hat{v}_{t_i,k}^{j_1 j_2 \cdots j_i} \tag{4.14}$$

[3] Jensen's inequality shows the connection between a convex function, f, evaluated at the expectation of a stochastic variable, X, versus taking the expectation value of the same function evaluated at the same stochastic variable , i.e. $f\left[\mathbf{E}[X]\right] \leq \mathbf{E}\left[f[X]\right]$.

to obtain the biased low estimator. The strategy to show that this way of estimating the expectation value really is biased low is nearly the same as for the high estimator and the proof can be found in [14]. An example of a random tree with $b = 3$ and three possible exercise opportunities, $\{t_0, t_1, t_2 = t_m\}$, is depicted in Figure 4.2 to visualize its construction and how we keep track of every node.

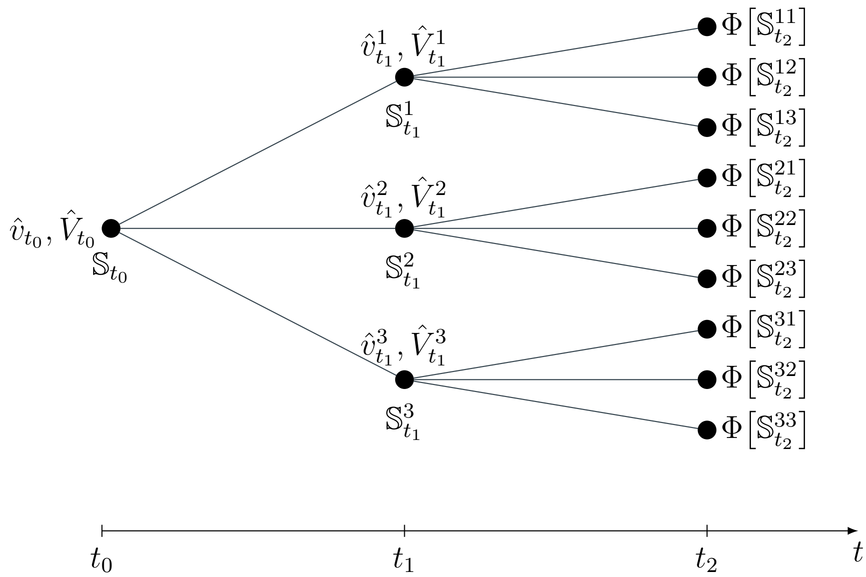


Figure 4.2: *A random tree, with branch parameter $b = 3$ and three exercise opportunities, showing the low and high estimators together with the generated underlying stocks. At maturity, $t = t_m$, both $\hat{V}^{j_1 j_2}_{t_2}$ and $\hat{v}^{j_1 j_2}_{t_2}$ are equal and considered as known values, why only the payoff function given different stock scenarios is shown (see Equation (4.11)).*

Once we have worked through the entire tree and obtained both $\hat{v}_{t_0}$ as well as $\hat{V}_{t_0}$, we have our first "samples" and we have to replicate the tree and this procedure n times in order to create $1 - \alpha$ confidence intervals set as

$$\text{CI} = \left[\bar{v}_{t_0}(n, b) - z_{1-\frac{\alpha}{2}} \frac{s_v(n, b)}{\sqrt{n}},\ \bar{V}_{t_0}(n, b) + z_{1-\frac{\alpha}{2}} \frac{s_V(n, b)}{\sqrt{n}}\right]. \tag{4.15}$$

Here $\bar{v}_{t_0}(n, b)$ and $\bar{V}_{t_0}(n, b)$ are the sample means, $s(n, b)$ is the standard deviation (with the subscript indicating which estimator) after n tree replications and $z_{1-\frac{\alpha}{2}}$ is the standard normal quantile at significance level α. As we can see in Equation (4.15), the high estimator's upper confidence bound and the low estimator's lower bound are merged, together forming the method's confidence interval. In this way we can with probability $1 - \alpha$ ensure that the true value V_0 will be enclosed in this confidence interval compared to the confidence intervals of either the low or the high estimator solely. Using either one of those intervals instead, we would face the risk that the true value is located outside their bounds with greater probability since they are skewed upwards and downwards respectively.

However, in [14] it is shown that both the low and high estimator converge to V_0 as $b \to \infty$ and/or $n \to \infty$, meaning that also CI $\to V_0$ [6], tempting one to assign large values to these parameters. Yet, the tree will grow exponentially and so will also the computational time, considerably limiting the types of options the method can handle conveniently. The method is therefore restricted to plain Bermudan options and more specifically, Glasserman [6] argues that five exercise opportunities is the critical limit before computational time gets unmanageable. Still, the method is attractive because it is relatively easy to price high-dimensional options and Broadie et al. [15] also suggest that Richardson extrapolation may be used to approximate values of American options by blending Bermudan options with different number of exercise opportunities. In the same article, they also propose a variance reduction technique closely related to antithetic variates which also happens to reduce the need of data storage and this will be the theme of the next section.

4.2.3 Antithetic Branching

This version of creating a random tree demands that the branching parameter b is an even number since for a node containing the stock prices $\mathbb{S}_{t_i}^{j_1 j_2 \dots j_i}$ generated by the GBM in Equation (4.9), we also create an antithetic mate. That is, a node with stock prices generated by applying the same vector of *i.i.d.* random variables as was used to obtain $\mathbb{S}_{t_i}^{j_1 j_2 \dots j_i}$, but with opposite signs. In Figure 4.3 we see how we use the *i.i.d.* random vectors to generate stock paths.

With this type of strategy to construct a random tree, we can reduce the variations in the estimates, but also the amount of random variables and hence storage space needed. Concerning the variations, these can be reduced due to that if a random variable is very large, it will give rise to a high stock price that may be far beyond the log-normally distributed stock's mean. The inclusion of the same random variable with an opposite sign will then yield a very low stock price located far away in the other direction, such that these extreme values will more or less offset each other [15]. This offsetting

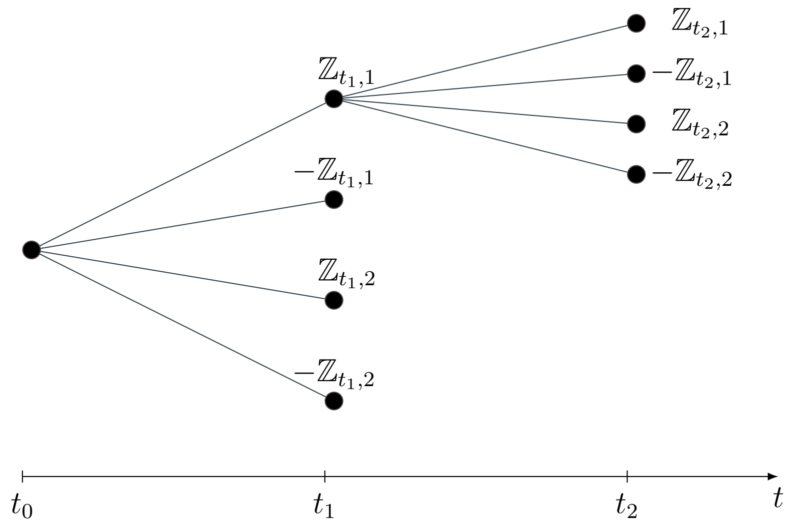


Figure 4.3: *Illustrative figure of how the i.i.d. random vectors are used at every node and where the second subscript indicates which vector. Remark: Branches from the remaining three nodes at $t = t_1$ are omitted to save space, but the principle is the same.*

would seldom occur when we as in the original version generate new random numbers for every node, implying a larger bias in its estimates.

Broadie et al. [15] present an algorithm of how we can work through the tree in a similar fashion as was done originally. To calculate the high estimator $\hat{V}$ we proceed as was done before using Equation (4.12), but the strategy to evaluate the low estimator, $\hat{v}$, is slightly changed. It is explained in the following algorithm where we assume every antithetic mates are placed next to each other as in Figure 4.3.

1. Separate two antithetic mates, k and $k+1$ from the $\frac{b}{2}$ pairs emanating from a parent node to calculate the node value as

$$\hat{v}_{t_i,l}^{j_1 j_2 \dots j_i} = \begin{cases} \Phi\left[\mathbb{S}_{t_i}^{j_1 j_2 \dots j_i}\right] \text{ if } \frac{1}{b-2} e^{-r\Delta t} \sum_{j=1,\, j\neq k,k+1}^{b} \hat{v}_{t_{i+1}}^{j_1 j_2 \dots j_i j} \leq \Phi\left[\mathbb{S}_{t_i}^{j_1 j_2 \dots j_i}\right], \\ \frac{1}{2} e^{-r\Delta t}\left[\hat{v}_{t_{i+1}}^{j_1 j_2 \dots j_i k} + \hat{v}_{t_{i+1}}^{j_1 j_2 \dots j_i k+1}\right] \quad \text{else.} \end{cases} \tag{4.16}$$

   This strategy is conducted for all mates why we will have $l = 1, 2, ..., \frac{b}{2}$.

2. Once we have $\frac{b}{2}$ low estimates for the node in question, we average

these to get the low estimator, i.e.

$$\hat{v}_{t_i}^{j_1 j_2 \dots j_i} = \frac{2}{b} \sum_{l=1}^{\frac{b}{2}} \hat{v}_{t_i,l}^{j_1 j_2 \dots j_i}. \tag{4.17}$$

The remaining procedure is then the same as in the original Random Tree method where we finally construct the confidence intervals using Equation (4.15).

4.3 Least Squares Monte Carlo

We will now consider a somewhat different method, compared to the previous ones in this chapter, that uses least squares regression to solve the complex pricing problem. Here we will employ a methodology similar to Longstaff and Schwartz [16], why we henceforth will denote this model as LSM (Longstaff-Schwartz Method). The theory presented will although be in style of Glasserman [6] where a more explicit description of the algorithm is elaborated and that also differs slighty from the original method, which is worth to be noted. As in the other two numerical methods to price options, we will also here use dynamic programming but slightly different to Equation (4.8) as we will soon see.

In the method, we simply assume that the continuation values at each time step, $C_{t_i}(\mathbb{S}_{t_i})$, can be written as a linear combination of some set of basis functions $\psi_j(\mathbb{S}_{t_i})$, with $j = 1, 2, ..., M$. More specifically, we have

$$C_{t_i}(\mathbb{S}_{t_i}) = \beta_{t_i}^T \psi(\mathbb{S}_{t_i}), \tag{4.18}$$

where $\beta_{t_i}^T = [\beta_{t_i,1}, \beta_{t_i,2}, ..., \beta_{t_i,M}]$ and $\psi(\mathbb{S}_{t_i}) = [\psi_1(\mathbb{S}_{t_i}), \psi_2(\mathbb{S}_{t_i}), ..., \psi_M(\mathbb{S}_{t_i})]^T$. These basis functions are set freely, but preferably with respect to the financial derivative in question to better resemble the true value.

The continuation values, as have been described previously, are the discounted expected value seen in the right expression of Equation (4.8). So, at each time step, it is the maximum of this and the immediate exercise premium we must evaluate in order to estimate the true value. Before we can do that however, we must use Equation (4.18) with an unknown coefficient vector $\beta_{t_i}^T$. To find its constituting elements for a time step t_i, we will now make use of least square regression.

To find the coefficients, we will first need to construct a path, $t_0, t_1, ..., t_m$, of the underlying stock vector $\mathbb{S}_{t_i}$ using the GBM in Equation (4.9) n times, such that we can set up a system of equations in similarity to Equation (4.18), that is

$$\Psi[\mathbb{S}_{t_i}]\,\beta_{t_i} = \Pi_{t_i,V}, \tag{4.19}$$

where we have

$$\Psi[\mathbb{S}_{t_i}] = \begin{bmatrix} \psi^T(\mathbb{S}_{t_i,1}) \\ \psi^T(\mathbb{S}_{t_i,2}) \\ \vdots \\ \psi^T(\mathbb{S}_{t_i,n}) \end{bmatrix} \quad \text{and} \quad \Pi_{t_i,V} = \begin{bmatrix} e^{-r\Delta t}\, V_{t_{i+1,1}} \\ e^{-r\Delta t}\, V_{t_{i+1,2}} \\ \vdots \\ e^{-r\Delta t}\, V_{t_{i+1,n}} \end{bmatrix}.$$

Thus we have that $\Psi[\mathbb{S}_{t_i}]$ is a $n \times M$ sized matrix containing n replications of the M basis functions that may or may not be functions of $\mathbb{S}_{t_i}$. Also, a second subscript is introduced here to indicate replication. We then use least square regression to solve for $\hat{\beta}_{t_i}$ in Equation (4.19), that is

$$\hat{\beta}_{t_i} = \left(\Psi^T[\mathbb{S}_{t_i}]\,\Psi[\mathbb{S}_{t_i}]\right)^{-1} \cdot \Psi^T[\mathbb{S}_{t_i}] \cdot \Pi_{t_{i+1},\hat{V}}. \tag{4.20}$$

From the resulting $\hat{\beta}_{t_i}$, we can now determine the estimated continuation value for a single replication l at time step t_i as

$$\hat{C}_{t_i}(\mathbb{S}_{t_i,l}) = \hat{\beta}_{t_i}^T\,\psi(\mathbb{S}_{t_i,l}) \tag{4.21}$$

and $l = 1, 2, ..., n$. In Figure 4.4 we see an illustration of how the method works at the penultimate nodes when using a vanilla put.[4] Looking at this figure, it should become clear that the choice of basis functions is important in order to get a good fit when we perform the least square regression, it is after all with these we approximate the continuation values.

Following Glasserman [6], we will now look at how we can utilize the presented theory to estimate an option's price using the following algorithm:

1. Use the GBM in Equation (4.9) to simulate the d underlying stocks' prices at the specified exercise opportunities $t_1, t_2, ..., t_m$ n times independently of each other.

2. We need initial values and as usual we consider the option values at t_m as known. Thus we have $\hat{V}_{t_m,l} = \Phi\left[\mathbb{S}_{t_m,l}\right]$.

[4] Illustrating an exotic option would be much more difficult, why we restrict ourselves to a vanilla option to get the basic idea of the method.

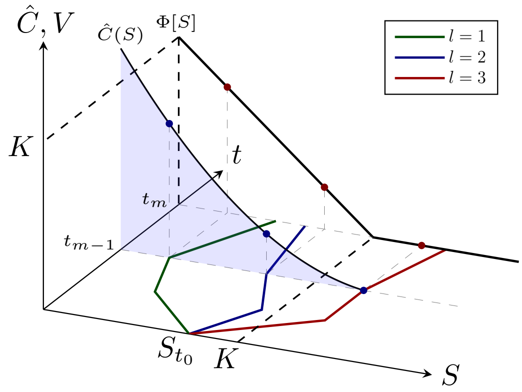


Figure 4.4: *In the figure, we see how the continuation values at t_{m-1} for an American put is obtained when using three replications and a hypothetical choice of basis functions. Red dots represent the regressands at the terminal nodes whereas blue dots are the values obtained by the expression in Equation (4.21) for every replication.*

3. For every time step $t_{m-1}, t_{m-2}, ..., t_1$, we estimate $\hat{\beta}_{t_i}$ and $\hat{C}_{t_i}(\mathbb{S}_{t_i,l})$ with Equations (4.20) and (4.21) respectively.

   3.1. Set the option value estimate at time t_i and replication l to

$$\hat{V}_{t_i,l} = \begin{cases} \Phi\left[\mathbb{S}_{t_i,l}\right] \text{ if } \Phi\left[\mathbb{S}_{t_i,l}\right] \geq \hat{C}_{t_i}(\mathbb{S}_{t_i,l}), \\ e^{-r\Delta t}\, \hat{V}_{t_{i+1},l} \quad \text{else.} \end{cases}$$

4. Finally, set the discounted mean value of all replications at $t = t_1$ as our estimation of the option value, namely

$$\hat{V}_{t_0} = e^{-r\Delta t}\, \frac{1}{n} \sum_{l=1}^{n} \hat{V}_{t_1,l}.$$

So, with this method we can like in the Random Tree method estimate prices of a broad range of American-style options since it allows us to handle d-dimensional problems. Another thing to note with the LSM method is the way we estimate the option values for every time step and replication which, as previously mentioned, differs somewhat from Equation (4.7). Instead of taking the maximum of the immediate exercise $\Phi\left[\mathbb{S}_{t_i,l}\right]$ and the continuation value $\hat{C}_{t_i}(\mathbb{S}_{t_i,l})$, we just use the continuation value as benchmark to determine whether we set the value of $\hat{V}_{t_i,l}$ to the discounted value of $\hat{V}_{t_{i+1},l}$ or the immediate exercise.

In conclusion, we remember that this is a Monte Carlo method and hence we expect the estimated values to fluctuate every time we apply it, but less dramatically when number of replications is large as a consequence of the law of large numbers. A large number of replications can although be time consuming and we will therefore look at two methods and their very basic ideas to overcome this problem.

4.3.1 Antithetic Variates

To reduce the variance in the estimates without using a larger number of replications, we can exploit antithetic variates in similarity to the antithetic branching procedure used in the Random Tree method, only that what we will see here is the original theoretical approach that is readily applicable to other methods. Both however, are based on the fact that we can "recycle" *i.i.d.* normal random variables by just using opposite signs. Another difference here is that we will generate another set of option values based on the random vectors with opposite signs, yielding a total of $2n$ option values.

The principle is thus that we generate two option values for each replication and the fact that the following two sets of *i.i.d.* normal random variables got the same distribution,

$$\begin{cases} \mathbb{Z} & = \{Z_1, Z_{2,}, ..., Z_d\} \sim \mathcal{N}[\vec{0}, \mathrm{I}] \\ -\mathbb{Z} & = \{-Z_1, -Z_{2,}, ..., -Z_d\} \sim \mathcal{N}[\vec{0}, \mathrm{I}], \end{cases}$$

is of importance since this will entail that probabilistically unlikely prices generated by GBM in Equation (4.9) and its corresponding option values are offset in the same manner as was discussed in the section about antithetic branching [6].

We will therefore denote $\tilde{V}_{t_0,l}$ as the sample using the same *i.i.d.* random variables as $\hat{V}_{t_0,l}$ in the original LSM method, but with opposite signs. This implies that these must be dependent and identically distributed, a fact we will soon exploit when need their respective variances. Before doing that, we define our antithetic variate $\hat{V}_{AV,l}$ for a single replication l to be the mean of the two related samples, i.e.

$$\hat{V}_{AV,l} = \frac{\hat{V}_{t_0,l} + \tilde{V}_{t_0,l}}{2} \tag{4.22}$$

and furthermore collect n of these new random variables defined in Equation (4.22). These will thus form an $i.i.d$ set, which is an important property in order to rely on the law of large numbers such that $\hat{V}_{AV} \to V_{t_0}$ for a large number n [17]. Hence, the antithetic estimator is simply the mean of all n samples;

$$\hat{V}_{AV} = \frac{1}{2n} \sum_{l=1}^{n} \hat{V}_{t_0,l} + \tilde{V}_{t_0,l}. \tag{4.23}$$

To see why fluctuations will be reduced by using Equation (4.23), we evaluate its variance;

$$\begin{aligned} \text{Var}[\hat{V}_{AV}] = \text{Var}\left[\frac{1}{2n} \sum_{l=1}^{n} \hat{V}_{t_0,l} + \tilde{V}_{t_0,l}\right] = \\ \frac{1}{4n^2} \sum_{l=1}^{n} \text{Var}[\hat{V}_{t_0,l}] + \text{Var}[\tilde{V}_{t_0,l}] + 2\text{Cov}[\hat{V}_{t_0,l}, \tilde{V}_{t_0,l}] = \left[\text{Var}[\tilde{V}_{t_0,l}] = \text{Var}[\hat{V}_{t_0,l}]\right] = \\ \frac{1}{4n^2} \sum_{l=1}^{n} 2\text{Var}[\hat{V}_{t_0,l}] + 2\text{Cov}[\hat{V}_{t_0,l}, \tilde{V}_{t_0,l}] = \frac{1}{2n}\text{Var}[\hat{V}_{t_0,l}] + \frac{1}{2n}\text{Cov}[\hat{V}_{t_0,l}, \tilde{V}_{t_0,l}]. \end{aligned} \tag{4.24}$$

and if we would have generated $2n$ replication without antithetic variates we would have

$$\text{Var}[\hat{V}_{t_0}] = \text{Var}\left[\frac{1}{2n} \sum_{l=1}^{2n} \hat{V}_{t_0,l}\right] = \frac{1}{4n^2} \sum_{l=1}^{2n} \text{Var}[\hat{V}_{t_0,l}] = \frac{1}{2n}\text{Var}[\hat{V}_{t_0,l}]. \tag{4.25}$$

It is the observation of the covariance term in Equation (4.24) that is the decisive evidence of variance reduction compared to the variance in Equation (4.25). This is due to that we earlier assumed that $\hat{V}_{t_0,l}$ and $\tilde{V}_{t_0,l}$ are identically distributed, but since they depend on the same $i.i.d.$ random variables but with different signs, we have the condition that

$$\text{Cov}[\hat{V}_{t_0,l}, \tilde{V}_{t_0,l}] < 0,$$

finally leading us to the conclusion

$$\text{Var}[\hat{V}_{AV}] < \text{Var}[\hat{V}_{t_0}].$$

Hence we see that we do not have to generate $2n$ option value samples based on entirely new random variables (something that may imply a much longer computational time) to get better precision in a Monte Carlo estimate, it requires only a total of n replications. So, we can with this relatively easy variance reduction method reduce the likelihood of unlikely option estimates, presumably to a less amount of time than what would have been the case otherwise.

4.3.2 Quasi-Monte Carlo

Instead of sampling two option values for every replication as in the antithetic variate method, we will here look at how we sample the *random variables*. Previously it has been assumed that the underlying stocks are modeled by Equation (4.9) which in its stated form depends on *i.i.d.* normal random variables. These can also be seen as

$$\mathbb{Z} = \{\mathcal{N}^{-1}[U^1], \mathcal{N}^{-1}[U^2], ..., \mathcal{N}^{-1}[U^d]\}, \tag{4.26}$$

where $U^\beta \sim U[\vec{0}, \mathrm{I}]$ are *i.i.d.* uniform random variables for each modeled stock β. These uniformly distributed random variables can in fact be replaced with something called a low-discrepancy sequence of deterministic uniform numbers. Using these, Glasserman [6] argues that the rate of convergence can be streamlined to the order of approximately $\mathcal{O}(\frac{1}{n})$ compared to $\mathcal{O}(\frac{1}{\sqrt{n}})$ of a standard Monte Carlo method.

To give an intuition of how these deterministic uniform numbers can be created, we will look at the Van der Corput sequence limited to problems in one dimension. Despite this sequence's lack of versatility, the foundation of it imbues those that can tackle higher-dimensional problems and is therefore a suitable, yet simple example to understand the basic idea of generating a low-discrepancy sequence. So to create this specific sequence, we follow Glasserman [6], where a positive integer k can be written as

$$k = \sum_{j=0}^{\infty} a_j(k) b^j,$$

where b is an integer base $b \geq 2$, $a_j(k)$ is a coefficient vector whose elements take the values 1 or 0 (mostly zeros) depending on the mirrored binary representation of k around the decimal point. This description might be vague, but we will look at an example shortly that hopefully straighten out the explanation.

Before this example though, we introduce the radical inverse function that is based on the expression above and defined as

$$\alpha_b(k) = \sum_{j=0}^{\infty} \frac{a_j(k)}{b^{j+1}},$$

designed to generate a low-discrepancy sequence of uniform numbers. To see how this radical inverse function works, we look at the same example that is given in Glasserman [6], but with an extension to better understand the composition of $a(k)$ as seen in Table 4.1. Here $b = 2$ and we note that the points of increasing k are placed uniformly alongside that of $k = 1$ on the line $[0, 1)$.

Table 4.1: Van der Corput sequence with base $b = 2$.

k	**Binary**	**Mirrored Binary**	$a(k)$	$\alpha_2(k)$
0	0	0	$0, 0, 0, 0, \ldots, 0$	0
1	1	0.1	$1, 0, 0, 0, \ldots, 0$	1/2
2	10	0.01	$0, 1, 0, 0, \ldots, 0$	1/4
3	11	0.11	$1, 1, 0, 0, \ldots, 0$	3/4
4	100	0.001	$0, 0, 1, 0, \ldots, 0$	1/8
5	101	0.101	$1, 0, 1, 0, \ldots, 0$	5/8
6	110	0.011	$0, 1, 1, 0, \ldots, 0$	3/8
7	111	0.111	$1, 1, 1, 0, \ldots, 0$	7/8

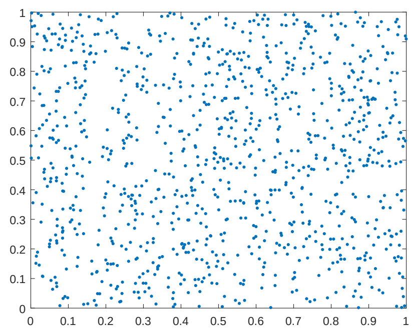
Since the primary goal of the numerical methods seen in this chapter is to solve exotic options that often depend on several stocks, it is of course insufficient to use a one-dimensional sequence. There are several rather complicated ways to generate higher-dimensional sequences, why we omit that theory since it is discussed in detail in Glasserman [6]. Still, it is worth mentioning one and for the context of financial problems, the Sobol'-sequence has been proven favorable. It uses the base $b = 2$ and is constructed from the Van der Corput sequence by specific permutation matrices [6].

With a generated high-dimensional low-discrepancy sequence like Sobol' at our disposal, we simply replace the *i.i.d.* uniform random variables in Equation (4.26) with the quasi random numbers

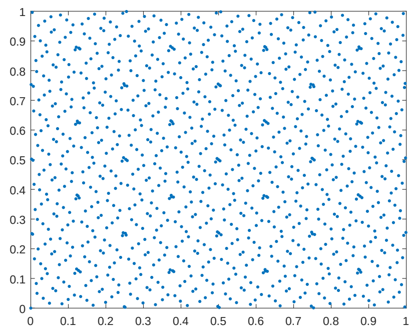
$$\mathbb{Z}_k^Q = \{\mathcal{N}^{-1}[\alpha_2^1(k)], \mathcal{N}^{-1}[\alpha_2^2(k)], ..., \mathcal{N}^{-1}[\alpha_2^d(k)]\}$$

for a given k when we generate the underlying stocks by Equation (4.9) in what we will now call Quasi-Monte Carlo computations.

We conclude this chapter with two panels depicting uniformly distributed (pseudo) random numbers and a two-dimensional Sobol' sequence to visualize their differences in the region $[0, 1)^2$ as seen in Figure 4.5. Here we clearly see the deterministic pattern of the generated Sobol' points compared to the the uniform random numbers.



(a) *Uniformly distributed random numbers.*



(b) *Quasi-random numbers using a Sobol' sequence.*

Figure 4.5: *A representation of the difference between deterministic quasi-random numbers and uniformly random numbers in 2-D, each with* 1024 *points.*

Chapter 5

Method

Preface

We have now seen four pricing methods that are applicable exclusively to plain vanilla American options and also a couple of numerical versions that have the ability to price a broad range of American-style options. Nevertheless, the latter suffer from the burden of time-consuming calculations, making them less desirable for the context of vanilla options.

As a consequence of this, we will only devote our attention to the closed-form approximation formulas when we evaluate how different pricing methods affect portfolio risk, the main objective of this thesis. However, we will still analyze how close the numerical methods' resulting prices are to retrieved benchmark prices to at least see how they perform and to answer which of them is best to apply for a specific exotic option. In this chapter, a short description will be given in order to explain how the different methods were implemented and in what ways the final results were acquired.

5.1 Implementation

In this section a brief description will be presented of how every pricing method was implemented and other relevant information regarding this. Another thing to point out is that the entire project was carried out in MATLAB® R2017a and calculations have hence been done by using matrix operations wherever it was possible in order to speed up runtime.

Another fact that relates to speed in calculations is the computer that was at disposal. It was equipped with an Intel® Xeon® E5-2630 2.60 GHZ processor, 6 cores and an installed physical memory (RAM) of 32 GB.

5.1.1 Closed-Form Approximations

There is actually not much to say about how the long closed-formed approximation formulas were implemented for obvious reasons, especially the Bjerksund-Stensland model. This is also more or less the case for the two other, Barone-Adesi and Whaley and Ju and Zhong models. Although, the latter models require that we implicitly find the exercise boundary using Equation (3.8) by applying a numerical method and here we will use Newton-Raphson. To do that we simply subtract both sides with the left-hand side such that we get an equation equal to zero, i.e.

$$0 = \underbrace{V^E(S_f) + \phi\big[1 - e^{-\delta t'}\mathcal{N}[\phi d(S_f)]\big]\frac{S_f}{\lambda} - \phi(S_f - K)}_{f(S_f)}, \tag{5.1}$$

where we set f to be the right-hand side and a function of S_f. Straightforward calculations and some rearrangements then reveal that the derivative $\frac{df}{dS_f} = f'(S_f)$ is

$$f'(S_f) = \phi\left[e^{-\delta t'}\mathcal{N}[\phi d(S_f)] - 1\right]\left(1 - \frac{1}{\lambda}\right) - \frac{e^{-\delta t'}n[\phi d(S_f)]}{\lambda\sigma\sqrt{t'}},$$

where $\phi^2 = 1$ have been used. We then use both f and f' to find the root S_f of Equation (5.1) using the Newton-Raphson algorithm

Algorithm 1 Newton-Raphson (NR)

1: **procedure** NR($S_0, K, r, t, T, \sigma, \delta$)
2: $\quad S_f^{Old} \leftarrow S_f^{Seed}$
3: $\quad$ **while** $|S_f^{New} - S_f^{Old}| > \varepsilon$ **do**
4: $\quad\quad S_f^{New} \leftarrow S_f^{Old} - \frac{f(S_f^{Old})}{f'(S_f^{Old})}$
5: $\quad\quad S_f^{Old} \leftarrow S_f^{New}$
6: $\quad$ **return** S_{Old}^f

where S_f^{Seed} is the initial guess value suggested by Barone-Adesi and Whaley [11] given in Chapter 3. The tolerance parameter ε is set freely but will here be $\varepsilon = 10^{-8}$, deemed sufficiently small.

5.1.2 Numerical Methods

5.1.2.1 Binomial Tree

The intuition behind the Binomial Tree method is not too complex, but to write a computer-efficient code for a framework entailing exponentially increasing storage might not be as obvious. Using the fact that the tree is recombining, we do not need to compute the nodes "within" the tree, that is, the ones that were highlighted with grayish color in the example in Chapter 4.

Broadie and Detemple [18] provide a clever pseudo-code that circumvent the problem of a large storage need and are able to reduce the complexity from $\mathcal{O}(m^2)$ to $\mathcal{O}(m)$ and as a reminder, m is the the number of time steps. Thus, we can calculate large trees at reasonable amount of times, such as sizes of $m = 10\,000$ to be used here to estimate true option values. Their pseudo-code was hence implemented and presented below in Algorithm 2 is a modified version (to match notation here) of the one given in [18].[1]

Algorithm 2 Binomial Method by Broadie and Detemple

procedure BINOMIAL($S_0, K, r, t, T, m, \sigma, \delta$)
    for $i = -m$ to m by 1 **do** ▷ Preallocate vectors
        $V[i], S[i]$
    $\Delta t \leftarrow \frac{T-t}{m} \quad a \leftarrow e^{(r-\delta)\Delta t} \quad b \leftarrow a\sqrt{e^{\sigma^2 \Delta t - 1}}$ ▷ Initial values
    $\nu \leftarrow a^2 + b^2 + 1 \quad u \leftarrow (\nu + \sqrt{\nu^2 - 4a^2})/(2a) \quad d \leftarrow 1/u$
    $p \leftarrow (a-d)/(u-d) \quad q \leftarrow 1 - p \quad D \leftarrow e^{-r\Delta t}$
    $p' \leftarrow D \cdot p \quad q' \leftarrow D \cdot q \quad S[0] \leftarrow S_0$
    for $i = 1$ to m by 1 **do** ▷ Build tree
        $S[i] = S[i-1] \cdot u$
        $S[-i] = S[-i+1] \cdot d$
    for $i = -m$ to m by 2 **do** ▷ Values at terminal nodes $t_m = T$
        $V[i] = \max(\phi(S(i) - K), 0)$
    for $j = m-1$ to 0 by -1 **do** ▷ Recursive option calculation
        for $i = -j$ to j by 2 **do**
            $V[i] = \max(p' \cdot V[i+1] + q' \cdot V[i-1], \phi(S[i] - K))$
    return $V[0]$

[1] In [18] an important caveat is given regarding dividend yields. There the authors state that abnormally large yields are not compatible with the pseudo-code given in Algorithm 2.

5.1.2.2 Random Tree

In the Binomial Tree we had two nodes emanating from every parent node, but with the Random Tree the number is set to b where potentially $b \gg 2$, severely increasing the storage need and computational time. Moreover, if that problem was not enough, we must also replicate the tree n times to create a confidence interval and a point estimator that we will use as the option price.

To solve this issue, Broadie and Glasserman [14] also present a pseudo-code to decrease storage and increase speed. Contrary to the Binomial Tree where the entire tree is implicitly built[2], they build the tree and calculate the option values part by part using depth-first programming which is visualized and explained in the simplified example in Figure 5.1. This example is similar to what is given in [6] but based on the tree in Chapter 4. Here the white nodes imply that memory has been freed since we do not need those values anymore, meaning that we can reduce storage space.

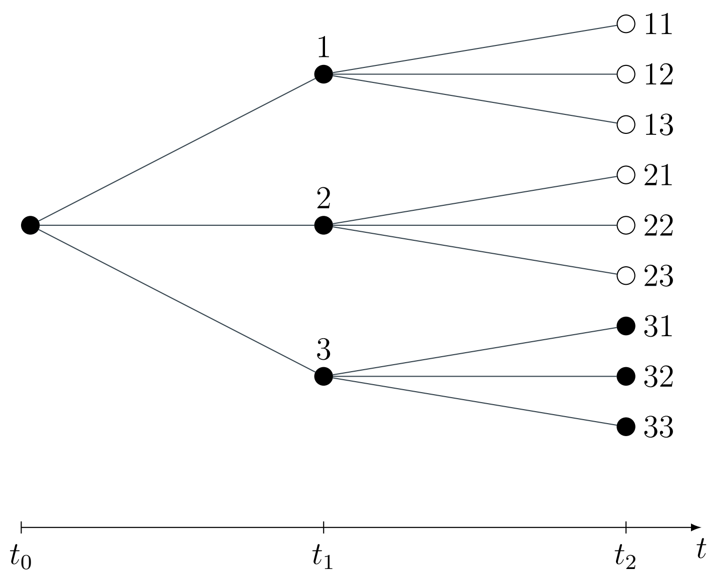


Figure 5.1: *Example of depth-first programming illustrating how we can free memory that is not needed anymore. White nodes here represent freed memory and numbers denote branch paths.*

To estimate the option values $\hat{v}^1_{t_1}$ and $\hat{V}^1_{t_1}$ of node 1 at t_1 we only generate

[2] As we know, we do not create the entire tree, but all nodes are accounted for since the interior ones are copies already calculated, which was meant by the line ”implicitly built”.

stock prices at the nodes that are necessary, namely 1, 11, 12, and 13. Once these are known, we free the memory of 11, 12, and 13 and continue with the goal of estimating $\hat{v}^2_{t_1}$ and $\hat{V}^2_{t_1}$. Now we only need stock prices at 21, 22 and 23 and after having generated as well as used them, we free memory again. Figure 5.1 depicts the case where we have done the above steps and are about to price $\hat{v}^3_{t_1}$ and $\hat{V}^3_{t_1}$ using option and stock values at 31, 32 and 33. Remark: it should be pointed out that when the values $\hat{v}^1_{t_1}$ and $\hat{V}^1_{t_1}$ are calculated, the stock prices at nodes 2 or 3 (as well as their emanating nodes) have not been generated yet, they are simply not necessary. Likewise, we do not need node 3 when we evaluate 2 for the same reason. When all option values at $t = t_1$ are known, we can then obtain the option prices $\hat{v}_{t_0}$ and $\hat{V}_{t_0}$.

This strategy is of course applicable to even larger trees with more than two time steps and three branches. In those cases, we also begin working with the uppermost arc of nodes and then continue in the same manner we saw in the example above, which should point out the principle of the depth-first programming used here. Figure 5.1 also reveals that we at most need $mb+1$ nodes in the memory, something that is discussed in a general manner by Glasserman [6]. Hence, we can with depth-first programming dramatically reduce the complexity of the method, from the horrifying $\mathcal{O}((db)^m)$ to the more acceptable $\mathcal{O}(bdm)$, where we remember that d is the number of underlying stocks. In Algorithm 3 we see a slightly modified version of the pseudo-code (although limited to a single underlying stock) given in [14] to perform depth-first programming that also inspired the implementation here.

5.1.2.3 Least Squares Monte Carlo

Finally, we have the least squares Monte Carlo method by Longstaff and Schwartz and as was the case for the closed-formed approximation formulas, there is actually not much to say about its implemenation. The only thing worth noting though regards the Quasi-Monte Carlo method, where a 2-D Sobol' sequence was formed using MATLAB®'s built-in function `sobolset`. In essence, it suffices to follow the theory presented in Chapter 4 to implement this methods properly. This is because we do not suffer from exponentially increasing tree that needs a special and clever implementation in order to avoid an extreme computational time. For a single replication, we "only" have to generate $dm + 1$ stock prices implying a complexity of $\mathcal{O}(dm)$, obviously much better than the Random Tree method.

Algorithm 3 Random Tree by Broadie and Glasserman

```
procedure Random Tree(S_0, K, r, t, T, m, b, σ, δ)
    for i = 1 to m by 1 do                                  ▷ Preallocate vector
        row[i]
    for j = 1 to b by 1 do                                  ▷ Preallocate matrix
        for i = 1 to m by 1 do
            V[j, i]
    V[1, 1] ← Stock Price    row(1) = 1
    for i = 2 to m by 1 do                                  ▷ Uppermost arc of stock prices
        V[1, i] = Stock Price    row(i) = 1
    i = m
    while i > 0 do                                          ▷ Dynamic programming
        if i = m and row[i] < b then
            V[row[i], i] = Φ[Stock Price]
            V[row[i] + 1, i] = Stock Price
            row[i] = row[i] + 1
        if i = m and row[i] = b then
            V[row[i], i] = Φ[Stock Price]
            row[i] = 0    i = i - 1
        if i < m and row[i] < b then
            V[row[i] + 1, i] = Stock Price    row[i] = row[i] + 1
            if i > 1 then
                V[row[i], i] = Option Value
                row[i] = 0    i = i - 1
                for j = i + 1 to m by 1 do
                    V[1, j] = Stock Price    row(j) = 1
                i = m
            else
                i = 0
        if i < m and row[i] = b then
            V[row[i], i] = Option Value
            row(j) = 1    i = i − 1
    return V[1, 1]
```

5.2 Price Approximation

5.2.1 Vanilla Options

To evaluate model performances for vanilla options with the payoff function

$$\Phi[S] = \max\big(\phi(S-K), 0\big),$$

where $\phi = 1$ for calls and $\phi = -1$ for puts, the Binomial Tree method was used as a benchmark using $m = 10\,000$ time steps. This method's prices based on this number of time steps are as mentioned commonly considered as true values in papers presenting new pricing models in order to evaluate their precision. Thus this method was deemed appropriate to serve as a benchmark in this project as well.

Since there are quite many input parameters for the models, we will restrict our analysis to the inputs $S_0 = \{90, 100, 110\}$, $\sigma = \{0.2, 0.4\}$, $K = 100$, $r = 0.04$, $\delta = 0.08$ and focus on $T = 1$ (year), even though results will be presented also for $T = 3$ (years) in Appendix A. Surface plots will also be shown depicting how the relative error $\text{RE} = \frac{|V-\hat{V}|}{V}$, where V is the true value and $\hat{V}$ is the approximation, varies in regions where time to maturity as well as the stock price range respectively as $T = \{0.5, 0.525, 0.55, ..., 3\}$ (years) and $S_0 = \{90, 90.35, 90.7, ..., 110\}$. Volatilities used for the plots were $\sigma = \{0.2, 0.4\}$, interest rates were $r = \{0.03, 0.1\}$ and dividend yield was set to $\delta = 0.08$.

5.2.2 Exotic Options

For the exotic option case, we focus only on the American call max-option with two underlying stocks and the payoff function

$$\Phi[\mathbb{S}] = \max\big(\max(\mathbb{S}) - K\big).$$

Here $\mathbb{S} = \{S^1, S^2\}$ and the true values for a specific set of input parameters were retrieved from a paper by Broadie and Glasserman [14]. True values for other exotic options, or the one in consideration here with more than two stocks, are rare and cannot be evaluated with great certainty why they are better omitted.

The parameters available in [14] were however $S_0 = \{80, 90, 100, 110, 120\}$, $\delta = 0.1$ and $\sigma = 0.2$ for both stocks. Remaining parameters were $T = 1$

(year), $\rho = 0.3$, $K = 100$ and $r = 0.05$. For the Random Tree structure, we use $b = 50$ and $n = 100$ replications, whereas $n = 100\,000$ in the LSM.

In the Random Tree method, we have only discussed confidence intervals until now, but it is more interesting to see a price approximation that would have been used in a real-life situation, why we use the point estimator introduced in [14] simply defined as

$$\text{Price} = 0.5\max(\Phi[\mathbb{S}], \hat{v}) + 0.5\hat{V}.$$

In the LSM method, prices are directly acquired without the need of a point estimator. On the other hand, we need a good choice of basis functions to properly fit the regressand values $\Pi_{t_i,V}$ during the regressions. One example is the set

$$\psi(\mathbb{S}_{t_i}) = \left[1, S^1_{t_i}, S^2_{t_i}, \left(S^1_{t_i}\right)^2, \left(S^2_{t_i}\right)^2, S^1_{t_i} \cdot S^2_{t_i}, \left(S^1_{t_i}\right)^3, \left(S^2_{t_i}\right)^3, \left(S^1_{t_i}\right)^2 \cdot S^2_{t_i}, S^1_{t_i} \cdot \left(S^2_{t_i}\right)^2, \Phi[\mathbb{S}_{t_i}]\right]^T$$

used in [6] for the same exotic option and this will also be used here. Note that brackets are introduced to distinguish superscripts indicating underlying stocks from exponents. We could of course have evaluated other basis functions also, but since these ones have been shown to work sufficiently well, we stick to them. After all, the point is to show how this method can perform given a good choice of basis functions. However, an important remark is that this do by no means indicate it was the best choice.

5.3 Portfolio Risk

5.3.1 Loss Distribution and Risk Measures

Now when we know how the different pricing methods were implemented and tested, we move on to the perhaps most important part of the thesis, namely how portfolio risk is affected by approximation errors in option prices. This has been done in two ways, firstly by using a standard Monte Carlo procedure and lastly by using historical simulation (HS). Using these methods, we can create scenario losses L_k and subsequently a Loss distribution, f_L, from which we can extract a risk measure like Value at Risk (VaR), henceforth considered to be a portfolio's *quantified risk estimate.*

Before we analyze these two methods, it is good to see how we can create a loss distribution out of an arbitrary option portfolio Λ. We begin with simulating losses L_k n times,[3] where $k = 1, 2, ..., n$, using

$$L_k = -\Delta\Lambda_k = -\sum_{j=1}^{N} \eta_j \cdot \left[V^j_{t+\Delta t,k} - V^j_{t,k} \right]. \tag{5.2}$$

Here η is the integer position on a particular option which may be positive or negative (i.e. long or short) and $j = 1, 2, ..., N$ specifies the unique options in the portfolio.[4] Remark: since we use $-\Delta\Lambda$, losses are treated as positive and not negative.

The distribution f_L is then obtained by sorting the set $L = \{L_1, L_2, ..., L_n\}$ of scenario losses in an increasing order and from which we can estimate the risk by e.g. VaR, explained in the following definition [19].

Definition 5.1. Value at Risk *Value at Risk is defined from*

$$\Pr(L \geq \mathrm{VaR}(L)_{1-\alpha}) = \alpha,$$

where a loss L will be greater than VaR with probability α.

This means that we find $\mathrm{VaR}_{1-\alpha}(L)$ as the loss distribution f_L's $(1-\alpha)$%-quantile [6], or more concrete, by extracting the element $f_L(\lceil(1-\alpha)n\rceil)$ once we have estimated the distribution. Another risk measure, although not of primary concern here, is Expected Shortfall (ES) where we instead consider the expectational loss beyond $\mathrm{VaR}(L)_{1-\alpha}$ and it is defined in similarity to [19].

Definition 5.2. Expected Shortfall *Expected Shortfall is defined as*

$$\mathrm{ES}_{1-\alpha} := \mathbf{E}[L | L \geq \mathrm{VaR}_{1-\alpha}(L)],$$

that is, the expectation value of the loss distribution's tail beyond $\mathrm{VaR}_{1-\alpha}(L)$.

Thus, once f_L is known, we simply take the mean value of all losses beyond index $\lceil(1-\alpha)n\rceil$ of f_L to estimate $\mathrm{ES}_{1-\alpha}$. In Figure 5.2 below, we see a hypothetical loss distribution where $\mathrm{VaR}_{1-\alpha}(L)$ is the quantile situated at the color transition between red and green. The area marked with red color is to highlight the losses beyond $\mathrm{VaR}_{1-\alpha}(L)$ necessary in determining $\mathrm{ES}_{1-\alpha}$.

[3] A "loss" L_k for a scenario k may actually be a profit, but the purpose is to create a loss distribution which is why we denote both profits and losses as merely losses.

[4] This means that we have N different options in the portfolio that may differ with respect to underlying stock, time to maturity and strike price.

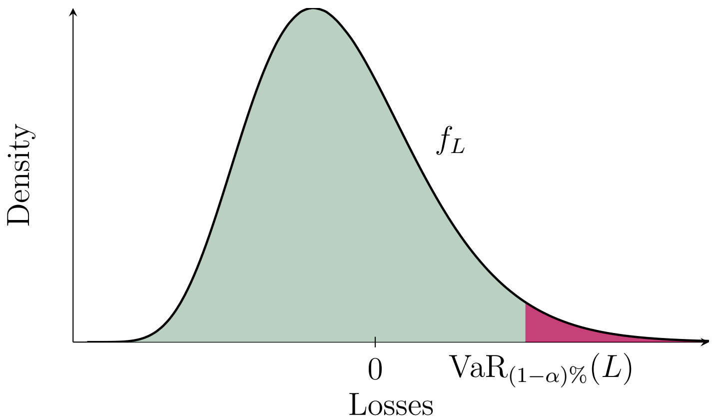


Figure 5.2: *Returns' loss distributions can be skewed and possess fat tails, based on stylized facts, thus assigning greater probability to extreme outcomes contrary to a normal distribution* [19]. *The exact shape of an option portfolio's loss distribution, however, depends on its specific option positions. Nevertheless, the figure should at least provide the basic intuition.*

We now know how a loss distribution can be created given a portfolio Λ but not how we obtain the unknown future values $V^j_{t+\Delta t,k}$. It is here we now need the Monte Carlo or the HS method to generate new stock prices at a time $t + \Delta t$ into the future.[5]

5.3.2 Monte Carlo Method

Here, new stock prices $\mathbb{S}_{t+\Delta t}$ given $\mathbb{S}_t$ are generated using the the price change $\Delta\mathbb{S} = (\Delta S_1, \Delta S_2, ..., \Delta S_d)$ in line with the notation made by Glasserman [6] instead of directly acquiring new prices through a GBM as previously. Further, to simulate these price changes, we set the distribution of $\Delta\mathbb{S}$ as

$$\Delta\mathbb{S} \sim \mathcal{N}[\vec{0}, \Sigma_S \cdot \Delta t], \tag{5.3}$$

meaning that the price change is normally distributed. This assumption is specifically used in [6] for the Delta and Delta-Gamma Approximation methods to create loss distributions, but is also adopted here for simplicity. As is mentioned in [6], the change in stock prices might anyway be vanishingly small for small time steps Δt, hence motivating the choice of the distribution seen in Equation (5.3). At the same time, we also avoid formal risk-neutral prices, usually prohibited by risk regulations, we would obtain

[5] Stocks that are underlying securities to the portfolio's constituting options. Yet, important to remember is that we only consider vanilla options in this thesis.

by the GBM stated in Chapter 4.

Furthermore, Glasserman [6] discusses that this also means that we can assume normally distributed returns R defined as

$$R = \left(\frac{\Delta S_1}{S_1}, \frac{\Delta S_2}{S_2}, ..., \frac{\Delta S_d}{S_d} \right). \tag{5.4}$$

At first glance, all this might be ambiguous but using a sufficiently small time step, this definition of returns are comparable to those of a GBM [6]. So what do we need this for? The answer is that it is a simple approach to how we can find the unknown matrix Σ_S. By rewriting Equation (5.4) as $\text{diag}(\mathbb{S}^{-1})\,\Delta\mathbb{S}^T$, where $\mathbb{S}^{-1}$ is to be understood as $\mathbb{S}^{-1} = (\frac{1}{S_1}, \frac{1}{S_2}, ..., \frac{1}{S_d})$ and using the linear transformation property defined in [6] as

Definition 5.3. Linear Transformation Property *If a normal variable $\mathbb{X}$, with mean vector $\mu_{\mathbb{X}}$ and covariance matrix $\Sigma_{\mathbb{X}}$, is linearly transformed by a matrix M we have*

$$\mathbb{X} \sim \mathcal{N}[\,\mu_{\mathbb{X}}, \Sigma_{\mathbb{X}}\,] \implies M\mathbb{X} \sim \mathcal{N}[\,M\mu_{\mathbb{X}}, M\Sigma_{\mathbb{X}}M^T\,],$$

implying that the operation $M\mathbb{X}$ is also normally distributed.

we have that

$$R \sim \mathcal{N}[\,\vec{0}, \text{diag}(\mathbb{S}^{-1}) \cdot \Sigma_S \cdot \text{diag}(\mathbb{S}^{-1}) \cdot \Delta t].$$

We can now see that the ordinary covariance matrix Σ introduced in Chapter 4 must be $\Sigma = \text{diag}(\mathbb{S}^{-1}) \cdot \Sigma_S \cdot \text{diag}(\mathbb{S}^{-1})$ because we work with stocks' returns. The covariance Σ is relatively easy to construct if we know the stocks' variances and mutual correlations, why Σ_S can be determined with little effort.

Based on the result above, we draw independent price changes based on Equation (5.3) by

$$\Delta S_k = \sqrt{\Delta t} \cdot C\,\mathbb{Z},$$

where as usual $\mathbb{Z} = \{Z_1, Z_{2,}, ..., Z_d\}^T \sim \mathcal{N}[\,\vec{0}, \text{I}\,]$ and C is the lower triangular Cholesky factor of Σ_S defined as $CC^T = \Sigma_S$. Using this, we find new stock prices for a scenario k by $\mathbb{S}_{t+\Delta t,k} = \mathbb{S}_t + \Delta\mathbb{S}_k$, whose constituting stock prices are used to revalue the options at $t + \Delta t$ combined with the new and reduced time to maturity $T - t - \Delta t$.

5.3.2.1 Setup

To see how portfolio risk using Value at Risk (results for Expected Shortfall is provided in Appendix A) is affected by pricing errors in the Monte Carlo method, an arbitrary set of 60 fictitious options (vanilla calls and puts) was defined and from which different portfolios could be composed. Half of these options were American whereas the other half were their European replicas. These options had either one or three years to maturity, different strike prices and three underlying stocks in total. A detailed description of the set is seen in Appendix B.

Further, for all stocks we have $S_0 = 100$, $\sigma = 0.2$ and the mutual correlations were set to $\rho = 0.3$ whilst the interest rate was $r = 0.04$. Using these inputs to calculate option values and setting the number of replications to $n = 5000$, a unit loss distribution matrix Γ_L could be constructed for every pricing method. This matrix facilitates the analysis of portfolio risk since when it is obtained, which with $n = 5000$ might take long time, it requires only that we multiply it with any portfolio position vector w (i.e. $\Delta\Lambda = w\,\Gamma_L$) to get its corresponding vector $\Delta\Lambda$ and subsequently, loss distribution f_L. An example of its composition using 5000 scenarios and 60 options is seen in Example 5.3.1.

Example 5.3.1. *Unit loss distribution matrix.* This matrix collects losses from all unique options in a specified set based on a unit long position on each and every option. It can be set as

$$\Gamma_L = \begin{pmatrix} V^1_{t+\Delta t,1} - V^1_{t,1} & V^1_{t+\Delta t,2} - V^1_{t,2} & \cdots & V^1_{t+\Delta t,5000} - V^1_{t,5000} \\ V^2_{t+\Delta t,1} - V^2_{t,1} & V^2_{t+\Delta t,2} - V^2_{t,2} & \cdots & V^2_{t+\Delta t,5000} - V^2_{t,5000} \\ \vdots & \vdots & \ddots & \vdots \\ V^{60}_{t+\Delta t,1} - V^{60}_{t,1} & V^{60}_{t+\Delta t,2} - V^{60}_{t,2} & \cdots & V^{60}_{t+\Delta t,5000} - V^{60}_{t,5000} \end{pmatrix}$$

and any portfolio specific loss distribution can be obtained from it. □

Using the loss distribution matrices, VaR estimates for every method could then be calculated where a time step of one week was used in this analysis, that is $\Delta t = 1/52$ (year).[6] As was the case with comparing different pricing methods, the Binomial Tree method with 10 000 time steps is considered as the method implying the "true" portfolio VaR. The portfolios evaluated will contain a certain percentage of American and European options where we will encounter notation such as $14\%E$ (with

[6]About a week was chosen since it is approximately the duration of an option position "disengagement" [6].

rounded percentages) meaning that the portfolio consists of five European options. In this case, we use a subset of all 60 options including (all) 30 American and the five first European options that follow in the list in Appendix B, remaining ones are omitted. If we instead have $33\%E$, there are 15 European options and 30 American options, where we use the first 15 European that follow (omitting the rest), thus the pattern should be clear.

In the analysis, we will begin with no European options and 30 American ($0\%E$), then increase the the number of European in steps of five until we use all 60 options in the set ($50\%E$). Contrary, above 50% European options (where we always use all 30 of them) we instead (somewhat confusingly) exclude American options from the top of the list. E.g. if $67\%E$, we instead omit the first 15 American options. Important to note is that the American options were included similarly to the 30 European as earlier, i.e. in steps of five. Here though, five American options were included from the start since the pure European portfolio is already calculated by using Black-Scholes-Merton formula for $0\%E$. To all this, we use a position vector w (also called weights) that defines a unique portfolio.

The elements of the position vector $w = [\eta_1, \eta_2, ..., \eta_{60}]$ were set to random integer numbers between -10 and 10, or in advanced imposed as zero if the option corresponding to this element was not included in the portfolio. For example, consider the portfolio $w^{14\%E}$, there we have a vector of 35 potentially nonzero elements (since we randomize), rest are forced to zero. In contrast, for $w^{75\%E}$ we have a vector beginning with 20 elements of zero, then followed by 40 potentially nonzero elements. Since we are working with random portfolio positions that might imply different VaR values (VaR is portfolio-specific), many combinations of positions had to be examined. Hence, for a given percentage of European options,[7] 10 000 randomized position vectors were generated and means of all VaR and VaR relative error values were taken. Moreover, a standard deviation, which is particularly interesting, was estimated from the resulting relative errors in VaR. The standard deviation can thus tell us in what range the relative error in VaR corresponding to an arbitrary vector of positions might entail. This approach might be loose, but was considered to be the best way of analyzing VaR when extremely many combinations of positions on a list of 60 options are possible. Again, VaR is portfolio-specific and a single VaR estimate would be insufficient to draw any inferences from.

[7] That is, using a particular subset of the full set of options in Appendix B.

This strategy of creating loss distributions is quite straightforward, however, when we draw price changes through a normal distribution we produce scenarios that are not objective, which are more or less demanded in risk regulations today. To at least see such an example, we will now look at the strategy of HS where we in contrast obtain objective results that are market implied.

5.3.3 Historical Simulation

Here we have a method that is less abstract than the one above since it requires only historic stock prices of the underlying securities. What we do is that we collect prices of the underlying stocks within a given window of W days which needs to be sufficiently large such that many possible market moments are captured.

We then transform the historic prices in the window for a given stock to logarithmic returns, Y_t, using the formula $Y_t = \ln(\frac{S_t}{S_{t-1}})$. Later, we fix time t and consider this to be "today" where after we randomly draw returns from the window for every day ahead until we reach the time $t + \Delta T$. This chain of returns is then transformed to obtain a stock price at $t + \Delta T$, i.e. $S_{t+\Delta t,k} = S_t e^{Y_{D_1}+Y_{D_2}+\ldots+Y_{D_h}}$ where D_1 denotes the first draw, D_2 the second and so on up till h draws.[8] This is then repeated n times in order to create enough scenarios as in the Monte Carlo simulation. To get a feeling for this method, a simple schematic example of how we draw three draws from a window W is shown in Figure 5.3.

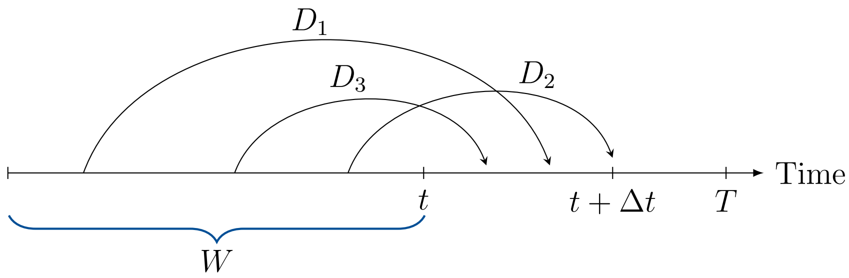


Figure 5.3: *A schematic example how we draw logarithmic returns from the window W up to the point in consideration, i.e. $t + \Delta t$.*

We can also supplement this procedure by using Exponentially Weighted Moving Average (EWMA) probabilities to increase the likelihood of drawing

[8] Important to note is that a specific draw can be drawn several time since we sample with replacement.

returns that better reflect the recent market conditions. These probabilities are defined by

$$Pr_{t_i}(Y_{t_i}) = \frac{1-\lambda}{\lambda(1-\lambda^W)} \cdot \lambda^{(W+1)-i},$$

with the natural constraint $\sum_{i=1}^{W} Pr_{t_i} = 1$. For daily returns as we consider here, $\lambda = 0.94$ is an adequate choice [19]. With HS in combination with EWMA probabilities, we now also have a method that produces objective prices that we can apply when we revalue the options at $t + \Delta t$ in the same manner as in the Monte Carlo method.

5.3.3.1 Setup

The setup here is similar to the one used in the Monte Carlo method, where a set of 60 fictitious options are used and half of these are American respectively European with different strikes and times to maturity.[9] A detailed description is again shown in Appendix B. Contrary to the Monte Carlo setup, the underlying stocks were now increased to five, where these were chosen to be the following (along with their respective tickers in brackets) listed on The New York Stock Exchange:

- The Boeing Company (BA)
- General Motors Company (GM)
- Wells Fargo & Company (WFC)
- Johnson & Johnson (JNJ)
- IBM Corporation (IBM)

Stock prices were collected using a window of 1000 trading days ($W = 1000$) between 2013-12-10 and 2017-11-29 such that enough market movements would be captured. Further, the vector $\mathbb{S}_t$ composed of "known" stock prices was assigned with the latest prices as of 2017-11-29, that is, what we consider is "today" in the simulation.

Even though the HS method is nonparametric the option revaluation step, where we use the closed-form approximation formulas and the Binomial Tree method, requires parameters. Beginning with the volatilities, these were

[9] Even though we use real market data, a restriction to only consider fictitious options was made. The most important thing is that we collect the underlying stocks' historic market movements which the options are contingent on anyway.

simply set as the standard deviation of every stock's logarithmic returns in the window whereas dividend yields were retrieved from Nasdaq, all these are specified in the next chapter. The risk-free interest rate was set to $r = 0.021$, matching the US 5-Year Treasury Yield as of 2017-11-30. In conclusion, seven draws ($h = 7$) implying a time step of $\Delta t = 1/52$ (same as in the Monte Carlo method) were made for every 2000 scenarios ($n = 2000$) to construct unit loss distribution matrices. Moreover, the way portfolios were created was identical to that of the Monte Carlo method, in fact, the same 10 000 position vectors for every subset mixture of American and European options were used.[10]

We have now gone through the underlying theory of options necessary to understand the subject and art of pricing American vanilla options using different approximation methods. All this have then been boiled down to how we can apply it for portfolio risk evaluation as we have seen in this chapter. With this knowledge, we are now ready to jump to the main results of this project, where we will begin with analyzing the methods' performances.

[10] That is, the only difference here is that there are 60 new options from which we construct new portfolios by using the same position vectors.

Chapter 6

Results

6.1 Method Performance

We begin to look at all pricing methods' performances by evaluating their relative error (RE) using the Binomial Tree with $m = 10\,000$ as a benchmark and "true value" for vanilla options, whereas we use retrieved prices from [14] when we consider the exotic option.

6.1.1 Vanilla Options

In the following, we introduce the abbreviations BS02 for the 2002 Bjerklund Stensland, JZ for Ju and Zhong, BAW for Barone-Adesi and Whaley and BSM for the Black-Scholes-Merton model to ease notation.[1] BSM prices for European options are mainly included in the analysis to see how they differ from their American counterparts. With this in mind, we now start by looking at tabulated prices for a call as well as a put and then analyze how the models perform in different parameter settings by evaluating relative error surface plots. In these plots, we will encounter the term moneyness which is simply defined as $S_0 - K$ for calls and $K - S_0$ for puts.

[1] For the Black-Scholes-Merton model, we use the Black-Scholes formula in Chapter 2 with the referred substitution that allow us to price European options on dividend-paying stocks.

Table 6.1: Performances of different approximation methods and the Binomial model with m time steps for the American *call* option. Time to maturity $T = 1$ (year).

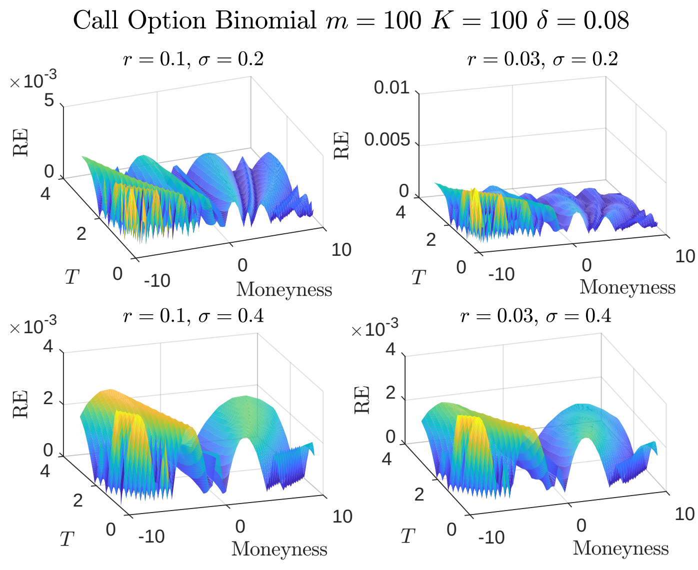
		Binomial $m = 10\,000$	**Binomial** $m = 100$	**JZ**	**BS02**	**BAW**	**BSM**
$S_0 = 90$, $\sigma = 0.2$	Price	2.5861	2.5974	2.5846	2.5511	2.6362	2.4322
	RE	-	0.0044	0.0006	0.0135	0.0194	0.0595
	Time (s)	0.8708	0.0004	0.0046	0.0067	0.0040	0.0017
$S_0 = 100$, $\sigma = 0.2$	Price	6.2590	6.2511	6.2391	6.2000	6.2841	5.7686
	RE	-	0.0013	0.0032	0.0094	0.0040	0.0784
	Time (s)	0.8213	0.0002	0.0039	0.0065	0.0038	0.0017
$S_0 = 110$, $\sigma = 0.2$	Price	12.1367	12.1468	12.1077	12.0769	12.0930	10.9008
	RE	-	0.0008	0.0024	0.0049	0.0036	0.1018
	Time (s)	0.8282	0.0002	0.0039	0.0066	0.0039	0.0018
$S_0 = 90$, $\sigma = 0.4$	Price	8.9400	8.9635	8.9194	8.8937	9.0180	8.6140
	RE	-	0.0026	0.0023	0.0052	0.0087	0.0365
	Time (s)	0.8742	0.0004	0.0060	0.0069	0.0059	0.0018
$S_0 = 100$, $\sigma = 0.4$	Price	13.7023	13.6839	13.6618	13.6448	13.7639	13.1217
	RE	-	0.0013	0.0030	0.0042	0.0045	0.0424
	Time (s)	0.8348	0.0002	0.0057	0.0066	0.0057	0.0019
$S_0 = 110$, $\sigma = 0.4$	Price	19.4931	19.5103	19.4338	19.4262	19.5194	18.5428
	RE	-	0.0009	0.0030	0.0034	0.0014	0.0487
	Time (s)	0.8260	0.0002	0.0057	0.0066	0.0057	0.0017

The remaining parameters were set to $K = 100$, $r = 0.04$ and $\delta = 0.08$.
Here RE stands for the Relative Error and is calculated with the Binomial method ($m = 10\,000$) and time is the wall-clock time. Except for Binomial with $m = 10\,000$, these are mean values from 10 runs to obtain reliable time estimates.

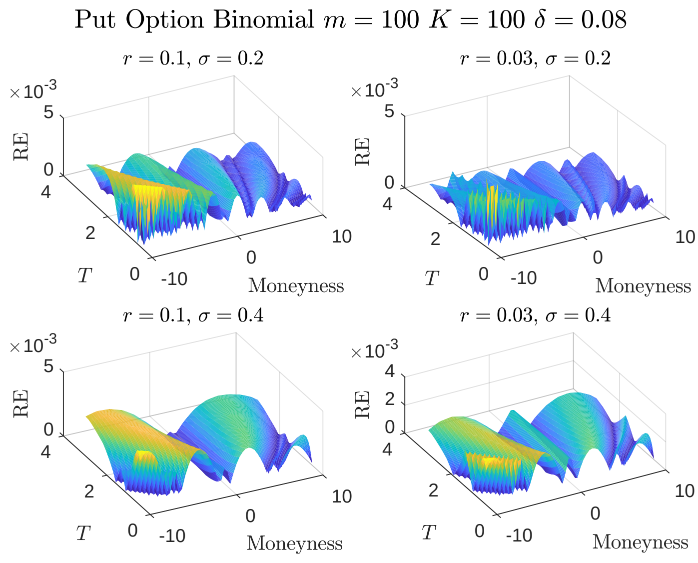
Table 6.2: Performances of different approximation methods and the Binomial model with m time steps for the American *put* option. Time to maturity $T = 1$ (year).

		Binomial $m = 10\,000$	**Binomial** $m = 100$	**JZ**	**BS02**	**BAW**	**BSM**
$S_0 = 90$, $\sigma = 0.2$	Price	15.4309	15.4424	15.4324	15.4308	15.4392	15.4307
	RE	-	0.0007	0.0001	0.0000	0.0005	0.0000
	Time (s)	1.2152	0.0004	0.0062	0.0073	0.0059	0.0018
$S_0 = 100$, $\sigma = 0.2$	Price	9.5357	9.5174	9.5366	9.5359	9.5405	9.5359
	RE	-	0.0019	0.0001	0.0000	0.0005	0.0000
	Time (s)	1.0382	0.0002	0.0056	0.0068	0.0057	0.0016
$S_0 = 110$, $\sigma = 0.2$	Price	5.4369	5.4508	5.4373	5.4369	5.4396	5.4369
	RE	-	0.0025	0.0001	0.0000	0.0005	0.0000
	Time (s)	1.1084	0.0002	0.0057	0.0067	0.0057	0.0016
$S_0 = 90$, $\sigma = 0.4$	Price	21.6304	21.6548	21.6446	21.6298	21.6917	21.6125
	RE	-	0.0011	0.0007	0.0000	0.0028	0.0008
	Time (s)	1.3291	0.0004	0.0077	0.0068	0.0077	0.0017
$S_0 = 100$, $\sigma = 0.4$	Price	16.8977	16.8629	16.9099	16.8977	16.9474	16.8890
	RE	-	0.0021	0.0007	0.0000	0.0029	0.0005
	Time (s)	1.0288	0.0002	0.0075	0.0065	0.0075	0.0016
$S_0 = 110$, $\sigma = 0.4$	Price	13.0835	13.0928	13.0932	13.0834	13.1233	13.0790
	RE	-	0.0007	0.0007	0.0000	0.0030	0.0003
	Time (s)	1.0400	0.0002	0.0075	0.0065	0.0075	0.0016

The remaining parameters were set to $K = 100$, $r = 0.04$ and $\delta = 0.08$.
Here RE stands for the Relative Error and is calculated with the Binomial method ($m = 10\,000$) and time is the wall-clock time. Except for Binomial with $m = 10\,000$, these are mean values from 10 runs to obtain reliable time estimates.

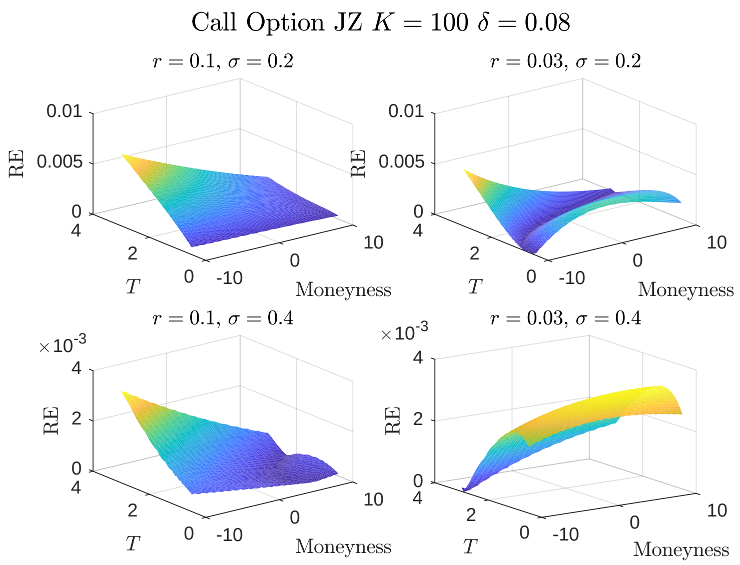


(a) *Call relative error surface of Binomial Tree with $m = 100$.*

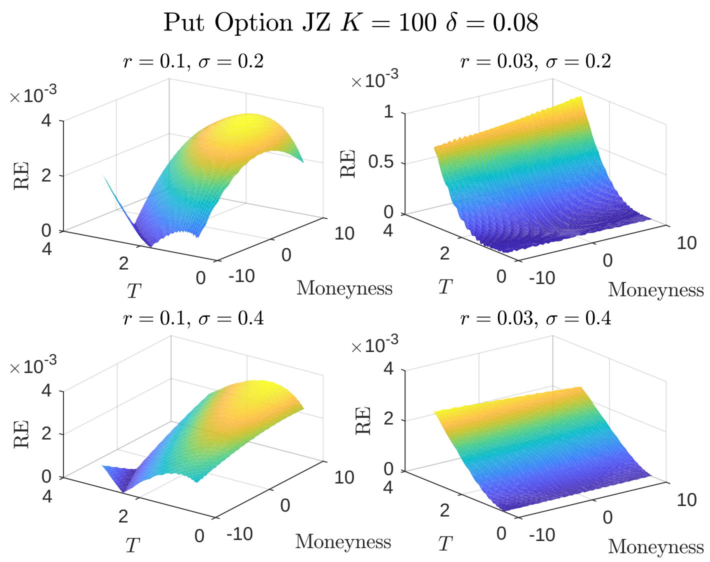


(b) *Put relative error surface of Binomial Tree with $m = 100$.*

Figure 6.1: *Relative error of prices from Binomial Tree ($m = 100$) with the same method using $m = 10\,000$ as a benchmark.*

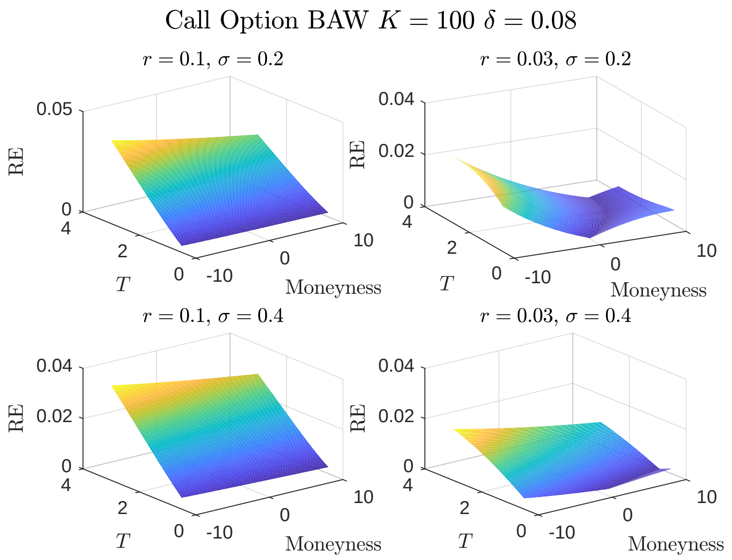


(a) *Call relative error surface of Ju Zhong.*

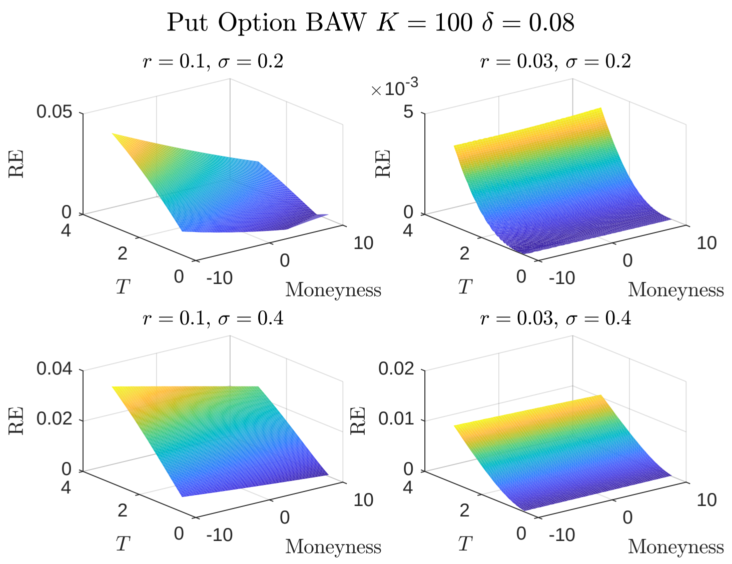


(b) *Put relative error surface Ju and Zhong.*

Figure 6.2: *Relative error of prices from Ju and Zhongs' method using $m = 10\,000$ as a benchmark.*

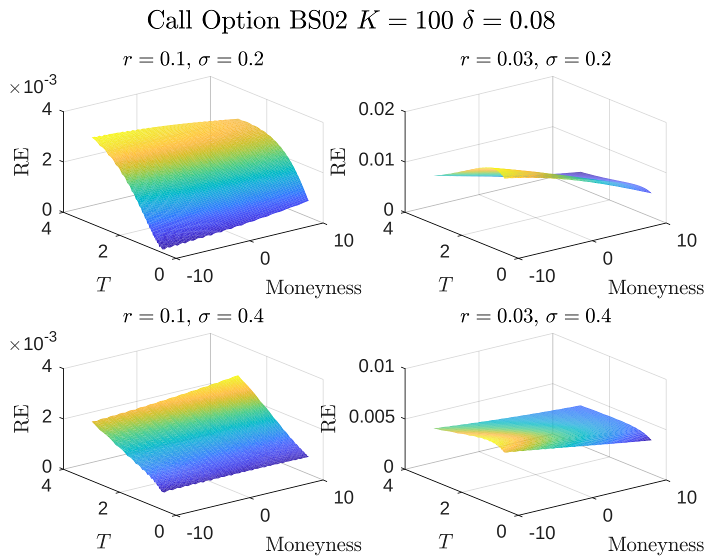


(a) *Call relative error surface Barone-Adesi and Whaley.*

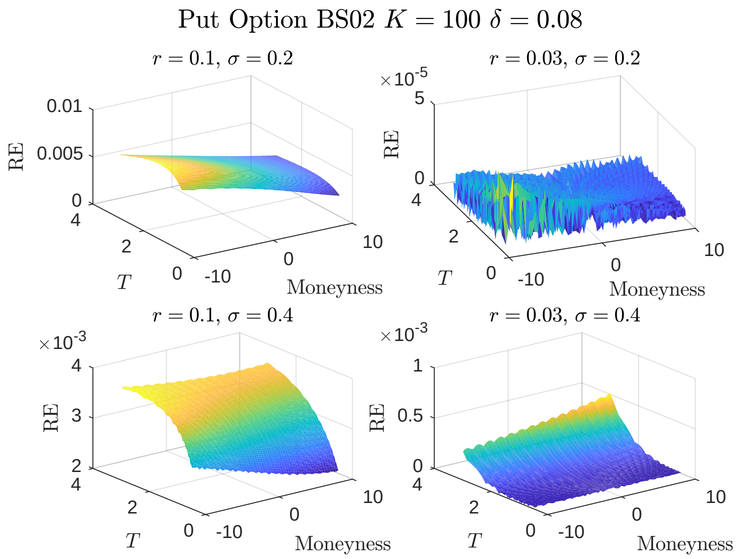


(b) *Put relative error surface Barone-Adesi and Whaley.*

Figure 6.3: *Relative error of prices from Barone-Adesi and Whaleys' method using* $m = 10\,000$ *as a benchmark.*

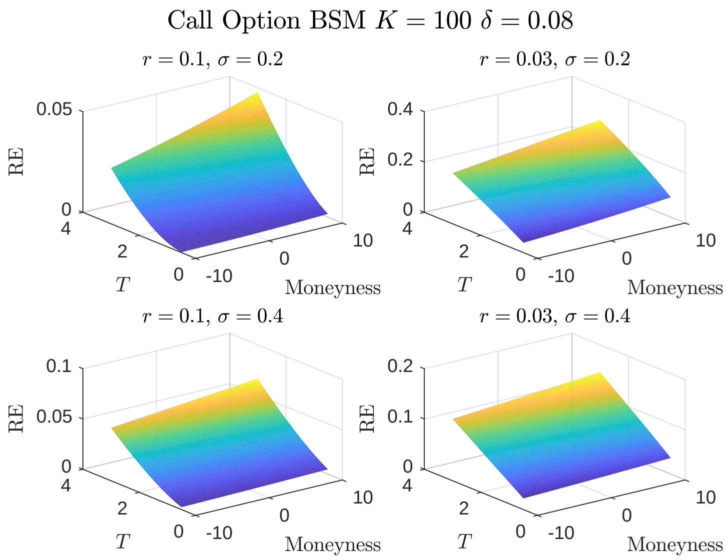


(a) *Call relative error surface Bjerksund and Stensland (2002).*

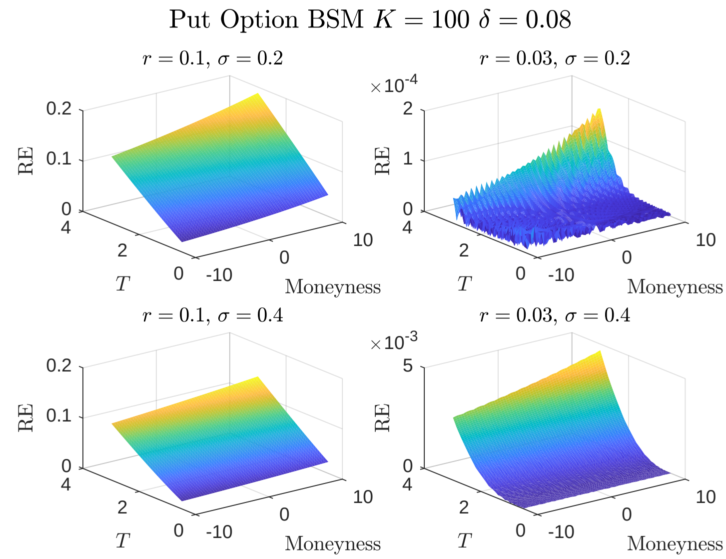


(b) *Put relative error surface Bjerksund and Stensland (2002).*

Figure 6.4: *Relative error of prices from Bjerksund and Stenslands' method using $m = 10\,000$ as a benchmark.*



(a) *Call relative error surface Black-Scholes-Merton.*



(b) *Put relative error surface Black-Scholes-Merton.*

Figure 6.5: *Relative error of prices from Black-Scholes-Merton's formula using $m = 10\,000$ as a benchmark.*

Comments on Tables

For the call option we see that all approximation methods perform about equally well with precision mostly at the third decimal, but where Ju and Zhongs' model is the best overall and the Black-Scholes-Merton model performs worst as expected. Suprisingly, we have that the Binomial Tree with $m = 100$ performs best and is almost ten times faster than all closed-form approximations.

The put option reveals that the Bjerksund and Stensland performs exceptionally well with precision at the fifth decimal, why we only see zeros. The two related methods by Ju and Zhong and BAW also perform well, where the former is slightly better, making its enhancement of BAW evident. What is more interesting is that the Black-Scholes-Merton model results in almost the true value for this set of input parameters and is beaten only by Bjerksund Stensland model. Compared to the call option case, we now see that Binomial Tree with $m = 100$ is less accurate for this set of parameters and option-type, however, still producing prices that do not deviate much.

Regarding speed, the Random Tree method with $m = 100$ is always the fastest as well as the two related models BAW and JZ are about equally fast. BS02 is somewhat slower than these two methods even though it does not need a numerical aid, but on the other hand it uses longer expressions and a multivariate normal distribution two times per computation.

Comments on Figures

The summary and tables above is not enough to draw any final conclusions about their true performances, a legitimate analysis requires that we stress test the models using a broader range of input parameters and in the surface plots we also vary the interest rate.

One thing that is interesting to see is the enhancement of Ju and Zhong compared to BAW at the case of intermediate times to maturity. Here BAW almost exclusively gets worse, but performs much better at short maturities where its first approximations is valid (the second regards very long times maturity, but we only look four years ahead). Another finding is that when we look at all plots, increasing the volatility seems for most cases increase error slightly. Further, increasing the interest rate changes the shape of all surfaces and almost every time worsening the results. The model that seems to perform consistently despite different inputs is in fact the Binomial Tree with $m = 100$ whereas the exceptional results for Bjerksund and Stensland in the tables appear to be just a fluke as a consequence of the specific inputs used (especially when using low risk-free interest rates).

6.1.2 Exotic Options

We now move on to the case of exotic options which unfortunately is a little bit out of focus due to their complexity and our inability to easily acquire other benchmark prices than those used here. With the single exotic option considered, the idea is at least that we should be able to see some useful results of how effective the models are when applying their respective variance reductions methods. Whether or not one can extrapolate the found facts about their performances to other option styles and inputs is although better left unsaid. The results are now following in Table 6.3.

Table 6.3: American max-option with time to maturity $T = 1$ (year) and two underlying stocks. Also including 95% confidence intervals (CI).

		Random Tree	**Random Tree (AB)**	**LSM**	**LSM (AV)**	**LSM Quasi**	**True**
$S_0 = 80$	Price	1.2185	1.2650	1.2574	1.2486	1.2523	1.259
	CI	[1.1608,1.2773]	[1.1928,1.3388]	[1.2311,1,2837]	[1.2307,1.2665]	-	
	RE	0.0322	0.0048	0.0013	0.0083	0.0053	
	Time (s)	191.9036	193.0081	3.4410	6.6126	3.6049	
$S_0 = 90$	Price	4.0276	4.0299	4.0449	4.0660	4.0664	4.077
	CI	[3.8595,4.1986]	[3.8714,4.1899]	[3.9972,4.0925]	[4.0352,4.0967]	-	
	RE	0.0121	0.0116	0.0079	0.0027	0.0026	
	Time (s)	190.8559	193.2802	4.1445	6,6519	3.4290	
$S_0 = 100$	Price	9.1074	9.0939	9.3723	9.3188	9.3338	9.361
	CI	[8.7896,9.4287]	[8.7982,9.3903]	[9.3034,9.4421]	[9.2798,9.3578]	-	
	RE	0.0271	0.0285	0.0012	0.0045	0.0025	
	Time (s)	191.6637	194.2802	4.10378	6.7364	3.4829	
$S_0 = 110$	Price	16.6423	16.7855	16.8880	16.9463	16.9153	16.924
	CI	[16.1922,17.0888]	[16.3445,17.2233]	[16.8005,16,9756]	[16.9032,16.9893]	-	
	RE	0.0166	0.0082	0.0021	0.0013	0.0005	
	Time (s)	191.4960	195.1984	4.24357	6.8800	3.5394	
$S_0 = 120$	Price	26.2261	25.6599	25.8922	25.9621	25.9685	25.980
	CI	[25.5744,26.8805]	[25.0484,26.2658]	[25.7923,25.9922]	[25.9176,26.0067]	-	
	RE	0.0095	0.0123	0.0034	0.0007	0.0004	
	Time (s)	190.0072	195.6748	4.346746	7.7617	3.6082	

Remaining parameters were set to $K = 100$, $r = 0.05$, $\rho = 0.3$ and $\delta = 0.1$ $\sigma = 0.2$ for both stocks. The number of exercise opportunities was four; $\{0, \frac{T}{3}, \frac{2T}{3}, T\}$, all in accordance to the numerical evaluation by Broadie and Glasserman [14]. Following their set-up, the number of replications for the Random Tree methods was $n = 100$, whereas the branching parameter was $b = 50$. For the Least Squares versions, the number of replications was $n = 100\,000$. Here RE stands for the Relative Error and time is the wall-clock time.

Comments on Table

Compared to the vanilla option pricing methods, the Random Tree method fails to produce the same accuracy and having relative errors mostly at the second decimal place. Nor did the antithetic branching improve the prices substantially and the confidence intervals are not much narrower than the standard model. Another thing we note is that the variance reduction method is somewhat slower even though it requires less space.

For the LSM model, we can however see slightly narrowed confidence intervals when we use antithetic variates and the method also produces results that have less relative errors for all but two of the cases. This shows that the variance reduction had an effect, although at the cost of longer computational times. Most interesting results here is however the LSM using Quasi-Monte Carlo. Here the resulting prices are basically second to none, yielding accurate results even at the fourth decimal place in two cases. Not only is it accurate, it is also the fastest method of them all. This in contrast to the Random Tree methods that are, despite depth-first programming and far less replications than in LSM, almost about 50 times slower.

6.2 Portfolio Risk

After having analyzed the different methods' performances, we now continue with the portfolio risk part using VaR at a confidence level $\alpha = 0.01$. Shown below in Table 6.4 and 6.5 are mean values for the relative errors and also the important standard deviations in order to capture the span of VaR values that an arbitrary vector with positions (that defines a portfolio and VaR is portfolio-specific) on a given subset of options may yield. This is shown for all rows except for the one corresponding to $w_u^{0\%E}$, simply because this is a single portfolio using only a unit long position on all 30 American options, which is highlighted by the subscript u.

Table 6.4: Monte Carlo portfolio VaR for $\Delta t = 1/52$ (year) using different pricing methods and different positions. All values are mean values of 10 000 randomized integer portfolio positions on the 60 options between -10 and 10. These options are listed in Appendix B.

		Binomial $N = 10\,000$	**Binomial** $N = 100$	**JZ**	**BAW**	**BS02**	**BSM**
$w_u^{0\%E}$	$\text{VaR}_{99\%}$	14.287956	13.985472	14.400976	14.806463	14.366683	16.875264
	RE	-	0.021628	0.007848	0.035019	0.005480	0.153320
	s_{RE}	-	-	-	-	-	-
$w^{0\%E}$	$\text{VaR}_{99\%}$	153.707073	153.673432	153.654880	153.567863	153.591219	150.430670
	$\bar{RE}$	-	0.006172	0.001202	0.003616	0.001053	0.029794
	s_{RE}	-	0.006859	0.001264	0.003931	0.000977	0.027899
$w^{14\%E}$	$\text{VaR}_{99\%}$	169.658666	169.617405	169.609804	169.530742	169.551793	166.622474
	$\bar{RE}$	-	0.005617	0.001072	0.003218	0.000942	0.026131
	s_{RE}	-	0.006086	0.001137	0.003545	0.000907	0.024920
$w^{25\%E}$	$\text{VaR}_{99\%}$	175.713284	175.658952	175.664747	175.592745	175.609956	172.823675
	$\bar{RE}$	-	0.005580	0.001051	0.003072	0.000916	0.024817
	s_{RE}	-	0.006676	0.001150	0.003463	0.000877	0.024148
$w^{33\%E}$	$\text{VaR}_{99\%}$	191.241256	191.232772	191.195998	191.115285	191.141578	188.480756
	$\bar{RE}$	-	0.005068	0.000942	0.002819	0.000846	0.022321
	s_{RE}	-	0.005672	0.001023	0.003051	0.000872	0.021925
$w^{40\%E}$	$\text{VaR}_{99\%}$	197.053870	197.031540	197.013544	196.953341	196.963323	194.439141
	$\bar{RE}$	-	0.004976	0.000922	0.002754	0.000814	0.021868
	s_{RE}	-	0.006012	0.001022	0.003198	0.000816	0.023354
$w^{45\%E}$	$\text{VaR}_{99\%}$	210.211891	210.186128	210.171452	210.105659	210.126140	207.776049
	$\bar{RE}$	-	0.004532	0.000878	0.002614	0.000750	0.020217
	s_{RE}	-	0.004896	0.000969	0.002893	0.000772	0.020073
$w^{50\%E}$	$\text{VaR}_{99\%}$	215.314671	215.306624	215.275881	215.213101	215.230639	212.978534
	$\bar{RE}$	-	0.004453	0.000853	0.002575	0.000715	0.019549
	s_{RE}	-	0.005329	0.000917	0.002871	0.000714	0.019298
$w^{86\%E}$	$\text{VaR}_{99\%}$	202.297186	202.269201	202.251215	202.164887	202.208969	199.860018
	$\bar{RE}$	-	0.004288	0.000932	0.002731	0.000772	0.021087
	s_{RE}	-	0.004606	0.001010	0.002888	0.000757	0.021498
$w^{75\%E}$	$\text{VaR}_{99\%}$	196.392835	196.376047	196.358672	196.340549	196.331627	194.506889
	$\bar{RE}$	-	0.004331	0.000881	0.002631	0.000636	0.019964
	s_{RE}	-	0.005221	0.001017	0.003135	0.000696	0.022018
$w^{67\%E}$	$\text{VaR}_{99\%}$	183.676768	183.646480	183.636115	183.598264	183.611104	181.653702
	$\bar{RE}$	-	0.004080	0.000923	0.002737	0.000675	0.021229
	s_{RE}	-	0.004877	0.001052	0.003075	0.000737	0.022346
$w^{60\%E}$	$\text{VaR}_{99\%}$	172.261958	172.241763	172.230990	172.268156	172.225222	170.885501
	$\bar{RE}$	-	0.003993	0.000833	0.002470	0.000455	0.018839
	s_{RE}	-	0.004798	0.001035	0.003158	0.000583	0.025643
$w^{55\%E}$	$\text{VaR}_{99\%}$	157.674479	157.670492	157.628234	157.529329	157.644719	156.293001
	$\bar{RE}$	-	0.002610	0.000824	0.002110	0.000447	0.019385
	s_{RE}	-	0.003126	0.001041	0.002512	0.000567	0.022748

The number of replications was $n = 5000$ and correlations between the three stocks were equal to $\rho = 0.3$. Remaining parameters for all stocks was $S_0 = 100$, $\delta = 0.02$ and $\sigma = 0.2$, whereas the risk-free interest rate was $r = 0.04$. Here w denotes the portfolio position vector with fraction indicating number of European options and u stands for the unit portfolio (unit long positions). Except for the unit portfolio, the positions range randomly between -10 and 10. $\bar{RE}$ is the relative error mean and s_{RE} is the standard deviation for the 10 000 weights.

Table 6.5: HS Portfolio VaR for $\Delta t = 1/52$ (year) using different pricing methods and different positions. All values are mean values of 10 000 randomized integer portfolio positions on the 60 options between -10 and 10. Window size was 1000 days ranging between 2013-12-10 to 2017-11-29. Options are listed in Appendix B.

		Binomial $N = 10\,000$	**Binomial** $N = 100$	**JZ**	**BAW**	**BS02**	**BSM**
$w_u^{0\%E}$	$\text{VaR}_{99\%}$	6.333053	6.372454	6.301798	6.303850	6.295001	6.105141
	RE	-	0.006183	0.004960	0.004633	0.006045	0.037331
	s_{RE}	-	-	-	-	-	-
$w^{0\%E}$	$\text{VaR}_{99\%}$	143.507758	143.508882	143.380749	143.282936	143.299492	141.128324
	$\bar{\text{RE}}$	-	0.005024	0.001036	0.002038	0.001774	0.018844
	s_{RE}	-	0.005285	0.000689	0.001781	0.001892	0.013852
$w^{14\%E}$	$\text{VaR}_{99\%}$	164.220229	164.230923	164.109577	164.020095	164.042717	162.096406
	$\bar{\text{RE}}$	-	0.004493	0.000893	0.001792	0.001532	0.015997
	s_{RE}	-	0.004794	0.000696	0.001691	0.001686	0.013316
$w^{25\%E}$	$\text{VaR}_{99\%}$	170.104760	170.103446	169.998920	169.914318	169.933368	168.091900
	$\bar{\text{RE}}$	-	0.004278	0.000844	0.001703	0.001435	0.014864
	s_{RE}	-	0.004577	0.000710	0.001639	0.001560	0.012717
$w^{33\%E}$	$\text{VaR}_{99\%}$	173.226389	173.219217	173.121456	173.036160	173.052855	171.223994
	$\bar{\text{RE}}$	-	0.004171	0.000817	0.001658	0.001450	0.014453
	s_{RE}	-	0.004357	0.000655	0.001578	0.001665	0.012165
$w^{40\%E}$	$\text{VaR}_{99\%}$	179.221330	179.220533	179.118222	179.034891	179.053139	177.262842
	$\bar{\text{RE}}$	-	0.004013	0.000785	0.001599	0.001379	0.013736
	s_{RE}	-	0.004157	0.000622	0.001509	0.001508	0.011639
$w^{45\%E}$	$\text{VaR}_{99\%}$	186.501286	186.502453	186.404422	186.326980	186.341634	184.686668
	$\bar{\text{RE}}$	-	0.003909	0.000743	0.001551	0.001301	0.012813
	s_{RE}	-	0.004132	0.000609	0.001456	0.001465	0.011000
$w^{50\%E}$	$\text{VaR}_{99\%}$	199.079006	199.089719	198.987622	198.916571	198.927821	197.375860
	$\bar{\text{RE}}$	-	0.003679	0.000706	0.001439	0.001187	0.011959
	s_{RE}	-	0.003869	0.000606	0.001391	0.001370	0.011040
$w^{86\%E}$	$\text{VaR}_{99\%}$	181.334818	181.338463	181.277441	181.209194	181.224179	179.853959
	$\bar{\text{RE}}$	-	0.003277	0.000585	0.001408	0.001124	0.011900
	s_{RE}	-	0.003482	0.000554	0.001416	0.001426	0.011128
$w^{75\%E}$	$\text{VaR}_{99\%}$	175.541397	175.539077	175.488843	175.413656	175.434179	174.076512
	$\bar{\text{RE}}$	-	0.003091	0.000594	0.001478	0.001167	0.012425
	s_{RE}	-	0.003277	0.000581	0.001517	0.001539	0.011706
$w^{67\%E}$	$\text{VaR}_{99\%}$	172.450650	172.451728	172.399602	172.337470	172.386665	171.217024
	$\bar{\text{RE}}$	-	0.003091	0.000578	0.001332	0.000514	0.010840
	s_{RE}	-	0.003331	0.000575	0.001503	0.000572	0.011400
$w^{60\%E}$	$\text{VaR}_{99\%}$	166.925762	166.920452	166.888568	166.860842	166.867057	166.197550
	$\bar{\text{RE}}$	-	0.002983	0.000518	0.001025	0.000513	0.006810
	s_{RE}	-	0.003256	0.000522	0.001194	0.000568	0.006886
$w^{55\%E}$	$\text{VaR}_{99\%}$	156.205976	156.209139	156.181563	156.176756	156.180476	155.919275
	$\bar{\text{RE}}$	-	0.001543	0.000373	0.000590	0.000297	0.003361
	s_{RE}	-	0.001765	0.000442	0.000895	0.000344	0.004292

The number of replications was $n = 2000$, volatilities $\sigma = [0.2089, 0.2394, 0.1914, 0.1422, 0.1881]$ and dividend yields $\delta = [0.0231, 0, 0.0267, 0.0278, 0.0274]$ respectively corresponding to stocks BA, GM, WFC, JNJ and IBM. Risk-free interest rate was $r = 0.021$ (US 5 Year Treasury 2017-11-30). Here w denotes the portfolio position vector with fraction indicating number of European options and u stands for the unit portfolio (unit long positions). Except for the unit portfolio, the positions range randomly between -10 and 10. $\bar{RE}$ is the relative error mean and s_{RE} is the standard deviation for the 10 000 weights.

Comments on Tables

In Table 6.4 and the first row we have the *portfolio* with unit positions on all American options solely to see how price errors affect risk under no influence of possible error offsetting due to particular portfolio positions. Then, for every row (each with a specified percentage of European and American options from which 10 000 random portfolios are constructed from), relative error mean values are shown to indicate what an arbitrary portfolio VaR might be. Evidently, the BS02 performs best overall, even in the unit case, with errors mostly at the fourth decimal place, closely followed by the model of Ju and Zhong.

Apparently, all models seem to produce VaR estimates roughly on the same order of magnitude even though the number of included European options increases (which can be priced by the exact Black-Scholes formula). What we also see is that their respective standard deviations are always of the same order as their corresponding mean relative error, meaning that a VaR outcome of a single portfolio might be ”far” (given the decimal place we are talking about) from the mean. Hence, we virtually see no sign of that a higher percentage of European options in a portfolio, based on mean relative errors, would reduce relative error in risk in any way. Another thing we note is that the different portfolio positions appear to more or less offset pricing errors for all models, most evident when we look at the unit portfolio using Black-Scholes-Merton. Here the relative error is about 15% and then drops dramatically to a mean around 2% for all other rows when using random positions. Even though this sudden drop, this model is still being outperformed by the others which are tailored to price American options.

Almost the same results are then visible in Table 6.5 when we look at the risk estimates based on historic data. The fact of using this instead of non-objective data should although not matter that much for the context of analyzing error in risk estimates. Something that differs to Table 6.4 however, is that JZ method now performs best overall.

Chapter 7

Discussion

7.1 Pricing Methods

We now carry on with analyzing the obtained results and we begin with the different methods to price options. Regarding the methods limited to vanilla calls and puts, it is hard to directly determine which is better or worse since all seem to perform good in different input parameter settings as we saw in the previous chapter.[1] Here however, a remark is that a broader range of input parameters would be needed to draw compelling conclusions on performances. Examples of this are especially longer times to maturity and other dividend yields. Yet surprisingly, the Binomial model with $m = 100$ time steps produced the most consistent results whatever the input parameters were in this project. Besides, it was also remarkably fast, meaning that the accuracy might be increased even more if we allow longer computational times matching those of the other methods.

Comparing speeds using MATLAB® is however not perfectly reliable and the fairest condition to judge the models' performances. Although MATLAB® is nice to work in, it is a high-level language that consumes considerable amount time especially in loops, why faster low-level ones are needed to truly unravel the methods full potential. Another thing to note is that all methods except the Binomial Tree use the normal cumulative and probability density functions, where calls to these demand quite amount of computational time

[1] One can make use of this fact by considering an implementation that selects the optimal method in response to the current input parameter setting, thus always producing the most accurate prices.

in MATLAB®. E.g. we see that even the Black-Scholes-Merton formula that requires few lines of code and operations is rather slow compared to the numerical Binomial Tree. The point is, if we would have used some other language, function calls to the normal distribution functions would perhaps require less effort, implying that the elapsed times would not be on the scales we see here. However, we can see some indications of their speeds *relative to each other* and important to remember is that the way the implementations were done may have influenced the obtained results negatively. That is, these could perhaps have been streamlined to yield better wall-clock times. Further, it is always important to open up for the possibility that mistakes in some of the implementations have been done, potentially inducing flawed results.

Now considering the relative error plots. Here the methods mostly appear to have smooth surfaces, but sometimes distinguishing themselves with steep curvatures. This is not easy to explain since the reason for this may lay within the complex mathematical framework of the different methods. However, what probably is the reason for peculiar look of the Binomial Tree ($m = 100$) surfaces is the fact that this model converges in an oscillatory manner, which is shown graphically by Broadie and Detemple [18]. Since this was the pricing method deemed to obtain most consistent results in considerably short amounts of time, it is likely that if we increase the number of time steps m, we would obtain surfaces with less oscillatory shapes and most importantly, lower errors at wall-clock times still better than the others.

A method that outclasses all other implementations concerning the exotic option is the Quasi-Monte Carlo version of the least squares model. We saw that this method simply got the best of both worlds, speed and accuracy. Whether or not it performs well for other exotic options as well is as previously discussed left unsaid. The LSM along with the antithetic variate version are however also performing well, but the two cannot produce a compelling combination of precision and speed, this is specially clear when the option is in the money, i.e. when the payoff function is non-zero (with respect to the initial stock prices). Another thing that is important to note is that we have to decide a good set of basis functions in order to obtain good regression fits and in this thesis we only used one set, which may not be the best choice. This method hence requires that we in advance know what is the optimal set, this is of course not perfect but once we know it, it should be readily applicable.

Regarding the Random Tree method, we can basically rule it out as an adequate model for real-time clearing based on the the observed computational times. Also, the depth-first programming and the method itself is heavily reliant on loops, why other programming languages might handle these problems better, but since the computational times are already so slow, it is hard to believe that the method can be fast enough for application regardless of language. Moreover, it is not a wild guess that any other exotic option would require the same amount of time when using the same parameter settings. After all, we would build the tree of stock prices and calculate the estimators almost in an exact same way. Further, its dependence of a point estimator using the mean value of the low and high estimator is not convincing. Both these should however converge to the true option price according to the law of large numbers, but as we saw, the model is exceptionally slow even for a number of replications that is not near the one used in the least squares versions.

To summarize, none of the methods were even close to the speeds of the ones tailored to vanilla options, but on the other hand, the option evaluated was by far more complex. Unfortunately, it is hard to find benchmark prices for exotic options, why this analysis was limited. It is therefore hard to say if other parameter settings, e.g. one with an increased number of exercise opportunities, would imply different results. Further, it is once again important to note that the implementations may not have been perfect, which could have impacted both speed and precision. Despite this, the results in this project at least give us some useful information about their performances and applicability.

7.2 Error in Risk Estimates

When the approximation methods' performances were analyzed based on the tables, we saw that BS02 resulted in the best results for the put option in one specific parameter setting. This is also what we see in the Monte Carlo VaR table where essentially the same parameter setting was used. As we know, it did not perform astonishingly well for the call options however and one might therefore think that errors linked to these would have offsetted the more accurate put prices, resulting in worse VaR values than the Binomial Tree ($m = 100$) that performs most consistently. This is not what we see though, in fact, it performs worst of all methods designed for American vanilla options. It is not easy to explain this behavior but it might be that, for the specific input parameters, the BS02's very good precision for put options prices outweighs the worse, although somewhat good, call prices such that good VaR estimates are attained compared to the Binomial Tree ($m = 100$) with "acceptable" prices for both calls and puts in all parameter settings. Although, these "acceptable" prices seem to have been insufficiently precise in order to yield better VaR estimates than all other methods in this particular parameter setting. An example that supports this hypothesis is that for the BSM model the (understandably) bad precision for the call options evidently offsets the very good precision for puts (since we use about the same parameter setting here as when the BSM provided surprisingly good put prices). Hence, the other methods' call prices were probably enough precise such that this offsetting on relative errors would not be in excess of those produced by the Binomial Tree.

Yet it is difficult to say which model is better or worse since only one setting of input parameters was studied in the project, mainly due to heavy calculations.[2] Other input parameters might reveal different results why a broader range of these must be used and analyzed. Using only one setting, as here and in the tabulated prices, might delude ourselves to believe a model like BS02 is the optimal choice for risk calculations. But varying the parameters like was done in the relative error plots revealed otherwise, hence clarifying the importance and need of a broader study with a wider range of parameters to reliably answer what method is optimal. Moreover, a wider range of portfolio positions would also be necessary to analyze as well as to include exotic options (perhaps even other financial instruments) to better mimic real portfolios. Other parameter settings might therefore result in that the Binomial Tree with $m = 100$ time steps is in fact the optimal choice because of its consistent price approximations. Similar discussion can also be done for the historical simulation results where almost the same input parameters were used. Here however, relative errors of BS02 and JZ where about equal, supporting the discussion about that it is hard to say which method is better or worse.

To conclude, the way the risk analysis was conducted by varying portfolio positions and evaluating error mean values might not have been a flawless approach, but as discussed, deemed necessary. Evaluating only a few portfolios and their relative errors would not reveal an overall result since different portfolio positions (which as we know may be extremely many for a given set of options) might offset pricing errors, that in turn affect portfolio risk estimates.

[2]Creating benchmark VaR estimates for Table 6.4 using $n = 5000$ replications with the Binomial Tree method took nearly two days of computational time.

Chapter 8

Conclusion

This project has uncovered useful results about performances of methods limited to vanilla options, broadening the understanding of how they work in different parameter settings. However, the analysis of exotic options was unfortunately limited, why more styles need to be considered in order to better evaluate and stress test the numerical methods. If more time is devoted to this, it might be possible to find benchmark prices for other option styles than the single one considered in this thesis, thus opening up for a deeper and more legitimate analysis of model performances. Despite this, the two methods used here gave us insight into their applicability since after all, they should in principle work the same way regardless of option style. Furthermore, the closed-form approximation methods and the Binomial Tree provided valuable data of how they perform compared to each other. Yet the answer to which is better or worse is unfortunately inconclusive since all methods perform good in different settings. This is of course inconvenient, but as mentioned, it is possible to benefit from this fact by simply make an implementation that chooses the optimal method in response to the current parameter setting, thus always producing the best possible approximations.

Such an implementation might then entail very small deviations from the true risk estimates, why an analysis of this sort might be a further development. Regarding this topic, only one setting of input parameters was used for each (that were similar) of the two methods to generate loss distributions due to heavy calculations. This resulted in that it was difficult to determine the different methods' general ability to estimate portfolio risk and draw parallels to their precision in price approximation. A broader range of parameter values and a larger set of unique options than the 60 used here (per loss distribution-generating method) would therefore be required in order to carefully answer these questions.

Besides this, useful results of the risk analysis have still been obtained. We have for example seen that the percentage of European options does not seem to reduce the relative errors in portfolio risk such that we can price American options erroneously deliberately in order to increase speed in calculations. Even though this percentage was high, only minor reductions in errors were observed. Thus it still requires us to price American options with any of the methods developed for this purpose despite they are few compared to the number of European in a mixed portfolio. This is as we have seen crucial in order to reduce the relative error significantly. Also, the approximation method with which we price options obviously does matter since risk estimates can be at scales of thousandth or ten thousandth depending on choice. This hence highlights the importance of accurately pricing American options to ensure that risk estimates truly reflect corresponding market conditions, especially in clearing houses where portfolios of potentially great values are handled.

References

[1] T. Björk, *Arbitrage Theory in Continuous Time.* Oxford Unversity Press, 3rd ed., 2009. p. 51, 98, 102, 111, pp. 186-187, 346-347.

[2] E. G. Haug, *The Complete Guide to Option Pricing Formulas.* McGraw-Hill Professional, 2nd ed., 2006. p. 109, 422.

[3] A. Rehlon and D. Nixon, "Central counterparties: What are They, Why do They Matter and How does the Bank Supervise Them?," *Q2 Quarterly Bulletin, Bank of England*, 2013.

[4] J. C. Hull, *Options, Futures and other Derivatives.* Prentice Hall Pearson, 8th ed., 2009. pp. 32-33, p. 196, 230, 282, 428, 433, 574, 791.

[5] D. D. Evanoff, D. Russo, and R. S. Steigerwald, "Policymakers, Researchers, and Practitioners Discuss the Role of Central Counterparties," in *The Role of Central Counterparties*, European Central Bank, 2007.

[6] P. Glasserman, *Monte Carlo Methods in Financial Engineering.* Springer, 2003. p. 65, 303, 335 pp. 104-105, 205-208, 281-287 422-479, 482-490.

[7] F. Black and M. Scholes, "The Pricing of Options and Corporate Liabilities," *The Journal of Political Economy*, 1973.

[8] R. C. Merton, "Theory of Rational Option Pricing," *The Bell Journal of Economics and management Science*, 1973.

[9] P. Wilmott, S. Howison, and J. Dewynne, *The Mathematics of Financial Derivatives: A Student Introduction.* Cambridge University Press, 1995. p. 106, 109.

[10] P. Bjerksund and G. Stensland, "Closed Form Valuation of American Options," 2002.

[11] G. Barone-Adesi and R. E. Whaley, "Efficient Analytic Approximation of American Options Values," *The Journal of Finance*, 1987.

[12] N. Ju and R. Zhong, "An Approximate Formula for Pricing American Options," *The Journal of Derivatives*, 1999.

[13] J. Cox, S. A. Ross, and M. Rubinstein, "Option Pricing: A Simplified Approach," *The Journal of Financial Economics*, 1979.

[14] M. Broadie and P. Glasserman, "Pricing American-Style Securities Using Simulation," *The Journal of Economic Dynamics and Control*, 1997.

[15] M. Broadie, P. Glasserman, and G. Jain, "Enhanced Monte Carlo Estimates for American Options Prices," *The Journal of Derivatives*, 2000.

[16] F. Longstaff and E. S. Schwartz, "Valuing American Options by Simulation: A Simple Least-Squares Approach," *The Review of Financial Studies*, 2001.

[17] R. Johnson and D. Wichern, *Applied Multivariate Statistical Analysis.* Pearson Education Limited, 6th ed., 2013. p. 175.

[18] M. Broadie and J. Detemple, "American Option Valuation: New Bounds, Approximations, and a Comparison of Existing Methods," *Review of Financial Studies*, 1996.

[19] J. Daníelsson, *Financial Risk Forecasting.* Wiley Finance, 2011. p. 9, 34, 76.

Appendix A

Supplementary Results

Table A.1: Performances of different approximation formulas and the Binomial model with m time steps for the American *call* option. Time to maturity $T = 3$ (years).

		Binomial $m = 10\,000$	**Binomial** $m = 100$	**JZ**	**BS02**	**BAW**	**BSM**
$S_0 = 90$, $\sigma = 0.2$	Price	5.0806	5.0937	5.0989	5.0323	5.2313	4.2259
	RE	-	0.0026	0.0036	0.0095	0.0296	0.1682
	Time (s)	0.8964	0.0004	0.0051	0.0067	0.0049	0.0018
$S_0 = 100$, $\sigma = 0.2$	Price	8.9439	8.9316	8.9465	8.8885	9.0524	7.1676
	RE	-	0.0014	0.0003	0.0062	0.0121	0.1986
	Time (s)	0.8221	0.0002	0.0048	0.0066	0.0049	0.0017
$S_0 = 110$, $\sigma = 0.2$	Price	14.3025	14.3127	14.2923	14.2507	14.3242	10.9961
	RE	-	0.0007	0.0007	0.0036	0.0015	0.2312
	Time (s)	0.8330	0.0002	0.0047	0.0069	0.0048	0.0019
$S_0 = 90$, $\sigma = 0.4$	Price	15.4548	15.4847	15.4538	15.3810	15.8127	13.7430
	RE	-	0.0019	0.0001	0.0048	0.0232	0.1108
	Time (s)	0.8701	0.0004	0.0050	0.0067	0.0051	0.0018
$S_0 = 100$, $\sigma = 0.4$	Price	20.4706	20.4517	20.4474	20.3847	20.8183	17.9845
	RE	-	0.0009	0.0011	0.0042	0.0170	0.1214
	Time (s)	0.8238	0.0002	0.0048	0.0066	0.0047	0.0017
$S_0 = 110$, $\sigma = 0.4$	Price	26.1242	26.1540	26.0789	26.0229	26.4380	22.6724
	RE	-	0.0011	0.0017	0.0039	0.0120	0.1321
	Time (s)	0.8177	0.0002	0.0048	0.0066	0.0047	0.0017

The remaining parameters were set to $K = 100$, $r = 0.04$ and $\delta = 0.08$.
Here RE stands for the Relative Error and is calculated with the Binomial method ($m = 10\,000$) and time is the wall-clock time. Except for Binomial with $m = 10\,000$, these are mean values from 10 runs to obtain reliable time estimates.

Table A.2: Performances of different approximation formulas and the Binomial model with m time steps for the American *put* option. Time to maturity $T = 3$ (years).

		Binomial $m = 10\,000$	**Binomial** $m = 100$	**JZ**	**BS02**	**BAW**	**BSM**
$S_0 = 90$, $\sigma = 0.2$	Price	22.1615	22.1820	22.2126	22.1602	22.3714	22.1214
	RE	-	0.0009	0.0023	0.0001	0.0095	0.0018
	Time (s)	1.0916	0.0004	0.0068	0.0068	0.0067	0.0016
$S_0 = 100$, $\sigma = 0.2$	Price	17.2146	17.1863	17.2552	17.2141	17.3798	17.1968
	RE	-	0.0016	0.0024	0.0000	0.0096	0.0010
	Time (s)	1.0284	0.0002	0.0067	0.0067	0.0066	0.0016
$S_0 = 110$, $\sigma = 0.2$	Price	13.1674	13.1894	13.1982	13.1669	13.2970	13.1591
	RE	-	0.0017	0.0023	0.0000	0.0098	0.0006
	Time (s)	1.0847	0.0002	0.0066	0.0065	0.0065	0.0016
$S_0 = 90$, $\sigma = 0.4$	Price	32.0665	32.0931	32.1464	32.0326	32.5416	31.6386
	RE	-	0.0008	0.0025	0.0011	0.0148	0.0133
	Time (s)	1.1459	0.0004	0.0069	0.0067	0.0067	0.0016
$S_0 = 100$, $\sigma = 0.4$	Price	28.3305	28.2828	28.4187	28.3089	28.7862	28.0137
	RE	-	0.0017	0.0031	0.0008	0.0161	0.0112
	Time (s)	1.0352	0.0001	0.0065	0.0065	0.0065	0.0016
$S_0 = 110$, $\sigma = 0.4$	Price	25.0746	25.1194	25.1654	25.0595	25.5061	24.8354
	RE	-	0.0018	0.0036	0.0006	0.0172	0.0095
	Time (s)	1.0318	0.0002	0.0065	0.0065	0.0065	0.0016

The remaining parameters were set to $K = 100$, $r = 0.04$ and $\delta = 0.08$.
Here RE stands for the Relative Error and is calculated with the Binomial method ($m = 10\,000$) and time is the wall-clock time. Except for Binomial with $m = 10\,000$, these are mean values from 10 runs to obtain reliable time estimates.

Table A.3: Monte Carlo portfolio ES for $\Delta t = 1/52$ (year) using different pricing methods and different positions. All values are mean values of 10 000 randomized integer portfolio positions on the 60 options between -10 and 10. Options are listed in Appendix B.

		Binomial $N = 10\,000$	**Binomial** $N = 100$	**JZ**	**BAW**	**BS02**	**BSM**
$w_u^{0\%E}$	$ES_{99\%}$	14.953113	14.698011	15.072011	15.519217	15.042327	18.029416
	RE	-	0.017356	0.007889	0.036478	0.005931	0.170627
	s_{RE}	-	-	-	-	-	-
$w^{0\%E}$	$ES_{99\%}$	176.912874	176.876259	176.855105	176.753328	176.771542	173.110003
	$\bar{RE}$	-	0.004914	0.001075	0.003317	0.001037	0.029637
	s_{RE}	-	0.005635	0.001127	0.003649	0.000913	0.027825
$w^{14\%E}$	$ES_{99\%}$	195.166490	195.120122	195.113242	195.025047	195.036740	191.644051
	$\bar{RE}$	-	0.004410	0.000952	0.002953	0.000923	0.026038
	s_{RE}	-	0.004933	0.001014	0.003348	0.000834	0.025124
$w^{25\%E}$	$ES_{99\%}$	202.088180	202.030861	202.033872	201.950646	201.963917	198.728137
	$\bar{RE}$	-	0.004403	0.000930	0.002803	0.000890	0.024610
	s_{RE}	-	0.005728	0.000997	0.003274	0.000821	0.024221
$w^{33\%E}$	$ES_{99\%}$	220.113016	220.095645	220.062726	219.973545	219.993382	216.910328
	$\bar{RE}$	-	0.003978	0.000829	0.002546	0.000817	0.022042
	s_{RE}	-	0.004544	0.000893	0.002838	0.000813	0.021986
$w^{40\%E}$	$ES_{99\%}$	226.752882	226.718699	226.707553	226.635642	226.643318	223.717768
	$\bar{RE}$	-	0.003945	0.000825	0.002515	0.000792	0.021644
	s_{RE}	-	0.004857	0.000936	0.003032	0.000774	0.023397
$w^{45\%E}$	$ES_{99\%}$	241.838915	241.814989	241.794179	241.719734	241.735361	239.027722
	$\bar{RE}$	-	0.003536	0.000776	0.002375	0.000718	0.019910
	s_{RE}	-	0.004018	0.000850	0.002675	0.000708	0.020026
$w^{50\%E}$	$ES_{99\%}$	247.819956	247.808523	247.776588	247.704627	247.718621	245.109292
	$\bar{RE}$	-	0.003489	0.000761	0.002355	0.000681	0.019276
	s_{RE}	-	0.004353	0.000818	0.002681	0.000659	0.019056
$w^{86\%E}$	$ES_{99\%}$	232.800530	232.768198	232.750416	232.650612	232.694812	229.979054
	$\bar{RE}$	-	0.003363	0.000827	0.002492	0.000734	0.020832
	s_{RE}	-	0.003695	0.000877	0.002668	0.000696	0.021506
$w^{75\%E}$	$ES_{99\%}$	225.991709	225.973924	225.954541	225.931290	225.917403	223.797858
	$\bar{RE}$	-	0.003382	0.000793	0.002426	0.000615	0.019742
	s_{RE}	-	0.004248	0.000916	0.002975	0.000655	0.022223
$w^{67\%E}$	$ES_{99\%}$	211.155375	211.119253	211.109521	211.062273	211.076355	208.805983
	$\bar{RE}$	-	0.003188	0.000831	0.002543	0.000647	0.020989
	s_{RE}	-	0.003846	0.000933	0.002893	0.000672	0.022395
$w^{60\%E}$	$ES_{99\%}$	198.115825	198.095605	198.082593	198.120963	198.073254	196.511596
	$\bar{RE}$	-	0.003143	0.000757	0.002298	0.000423	0.018650
	s_{RE}	-	0.003676	0.000914	0.002977	0.000533	0.026540
$w^{55\%E}$	$ES_{99\%}$	181.461538	181.455937	181.408563	181.292351	181.425212	179.844672
	$\bar{RE}$	-	0.002075	0.000747	0.001967	0.000419	0.019098
	s_{RE}	-	0.002392	0.000910	0.002303	0.000505	0.022785

The number of replications was $n = 5000$ and correlations between the three stocks were equal to $\rho = 0.3$. Remaining parameters for all stocks was $S_0 = 100$, $\delta = 0.02$ and $\sigma = 0.2$, whereas the risk-free interest rate was $r = 0.04$. Here w denotes the portfolio position vector with fraction indicating number of European options and u stands for the unit portfolio (unit long positions). Except for the unit portfolio, the positions range randomly between -10 and 10. $\bar{RE}$ is the relative error mean and s_{RE} is the standard deviation for the 10 000 weights.

Table A.4: HS Portfolio ES for $\Delta t = 1/52$ (year) using different pricing methods and different positions. All values are mean values of 10 000 randomized integer portfolio positions on the 60 options between -10 and 10. Window size was 1000 days ranging between 2013-12-10 to 2017-11-29. Options are listed in Appendix B.

		Binomial $N = 10\,000$	**Binomial** $N = 100$	**JZ**	**BAW**	**BS02**	**BSM**
$w_u^{0\%E}$	$\text{ES}_{99\%}$	6.747965	6.821561	6.720805	6.731374	6.716925	6.584441
	RE	-	0.010789	0.004041	0.002465	0.004621	0.024835
	s_{RE}	-	-	-	-	-	-
$w^{0\%E}$	$\text{ES}_{99\%}$	167.010567	167.003688	166.867949	166.745729	166.751097	164.176128
	$\bar{\text{RE}}$	-	0.003686	0.000951	0.001868	0.001685	0.018833
	s_{RE}	-	0.003754	0.000551	0.001500	0.001541	0.013343
$w^{14\%E}$	$\text{ES}_{99\%}$	191.369869	191.377181	191.246581	191.133647	191.148550	188.836650
	$\bar{\text{RE}}$	-	0.003292	0.000789	0.001611	0.001404	0.015848
	s_{RE}	-	0.003534	0.000573	0.001433	0.001319	0.013042
$w^{25\%E}$	$\text{ES}_{99\%}$	198.078555	198.065764	197.960118	197.854479	197.862582	195.675678
	$\bar{\text{RE}}$	-	0.003169	0.000752	0.001515	0.001345	0.014641
	s_{RE}	-	0.003346	0.000558	0.001361	0.001288	0.012330
$w^{33\%E}$	$\text{ES}_{99\%}$	201.549162	201.533571	201.431830	201.329055	201.334014	199.184973
	$\bar{\text{RE}}$	-	0.003067	0.000720	0.001444	0.001339	0.014033
	s_{RE}	-	0.003147	0.000523	0.001286	0.001337	0.011683
$w^{40\%E}$	$\text{ES}_{99\%}$	208.368875	208.362436	208.253219	208.149069	208.154617	206.028500
	$\bar{\text{RE}}$	-	0.002933	0.000688	0.001420	0.001287	0.013464
	s_{RE}	-	0.003004	0.000495	0.001231	0.001250	0.011070
$w^{45\%E}$	$\text{ES}_{99\%}$	217.071286	217.068272	216.963099	216.866032	216.874929	214.901136
	$\bar{\text{RE}}$	-	0.002832	0.000656	0.001356	0.001168	0.012400
	s_{RE}	-	0.002948	0.000493	0.001205	0.001162	0.010696
$w^{50\%E}$	$\text{ES}_{99\%}$	231.709124	231.708197	231.607276	231.517853	231.522833	229.674371
	$\bar{\text{RE}}$	-	0.002652	0.000611	0.001261	0.001078	0.011414
	s_{RE}	-	0.002762	0.000478	0.001171	0.001082	0.010429
$w^{86\%E}$	$\text{ES}_{99\%}$	211.135291	211.135899	211.071694	210.985940	210.995913	209.338743
	$\bar{\text{RE}}$	-	0.002424	0.000494	0.001203	0.000998	0.011417
	s_{RE}	-	0.002494	0.000441	0.001168	0.001143	0.010800
$w^{75\%E}$	$\text{ES}_{99\%}$	204.418348	204.411606	204.360150	204.266092	204.281075	202.628518
	$\bar{\text{RE}}$	-	0.002309	0.000502	0.001271	0.001044	0.011912
	s_{RE}	-	0.002358	0.000461	0.001264	0.001237	0.011458
$w^{67\%E}$	$\text{ES}_{99\%}$	200.835630	200.838468	200.779757	200.699543	200.749941	199.315895
	$\bar{\text{RE}}$	-	0.002364	0.000488	0.001181	0.000534	0.010544
	s_{RE}	-	0.002468	0.000453	0.001253	0.000533	0.011139
$w^{60\%E}$	$\text{ES}_{99\%}$	194.588034	194.584564	194.547069	194.508818	194.508912	193.681208
	$\bar{\text{RE}}$	-	0.002266	0.000430	0.000915	0.000535	0.006614
	s_{RE}	-	0.002352	0.000412	0.001039	0.000546	0.006817
$w^{55\%E}$	$\text{ES}_{99\%}$	181.678640	181.679958	181.651777	181.639625	181.647039	181.317472
	$\bar{\text{RE}}$	-	0.000882	0.000321	0.000559	0.000277	0.003340
	s_{RE}	-	0.000950	0.000346	0.000765	0.000293	0.004153

The number of replications was $n = 2000$, volatilities $\sigma = [0.2089, 0.2394, 0.1914, 0.1422, 0.1881]$ and dividend yields $\delta = [0.0231, 0, 0.0267, 0.0278, 0.0274]$ respectively corresponding to stocks BA, GM, WFC, JNJ and IBM. Risk-free interest rate was $r = 0.021$ (US 5 Year Treasury 2017-11-30). Here w denotes the portfolio position vector with fraction indicating number of European options and u stands for the unit portfolio (unit long positions). Except for the unit portfolio, the positions range randomly between -10 and 10. $\bar{RE}$ is the relative error mean and s_{RE} is the standard deviation for the 10 000 weights.

Appendix B

Portfolio Constituents

On the following two pages are the two sets (Tables B.1 and B.2) of 60 options which were used for the portfolio risk analysis. In these tables, the first half corresponds to American options while the remainder are their exact European counterparts. These sets were used to create unit loss distribution matrices for every pricing method and from which many different portfolios could be formed.

Table B.1: 60 fictitious options that are used to construct different portfolios in the Monte Carlo method. The first 30 options are American whereas the following 30 are their exact European counterparts.

Option Type A = American E = European	C = Call P = Put	Underlying Security	Strike Price	Time to Maturity
A	C	1	80	1
A	C	1	90	1
A	C	1	100	1
A	C	1	110	1
A	C	1	120	1
A	P	1	80	1
A	P	1	90	1
A	P	1	100	1
A	P	1	110	1
A	P	1	120	1
A	C	2	80	1
A	C	2	90	1
A	C	2	100	1
A	C	2	110	1
A	C	2	120	1
A	P	2	80	1
A	P	2	90	1
A	P	2	100	1
A	P	2	110	1
A	P	2	120	1
A	C	3	80	3
A	C	3	90	3
A	C	3	100	3
A	C	3	110	3
A	C	3	120	3
A	P	3	80	3
A	P	3	90	3
A	P	3	100	3
A	P	3	110	3
A	P	3	120	3
E	C	1	80	1
E	C	1	90	1
E	C	1	100	1
E	C	1	110	1
E	C	1	120	1
E	P	1	80	1
E	P	1	90	1
E	P	1	100	1
E	P	1	110	1
E	P	1	120	1
E	C	2	80	1
E	C	2	90	1
E	C	2	100	1
E	C	2	110	1
E	C	2	120	1
E	P	2	80	1
E	P	2	90	1
E	P	2	100	1
E	P	2	110	1
E	P	2	120	1
E	C	3	80	3
E	C	3	90	3
E	C	3	100	3
E	C	3	110	3
E	C	3	120	3
E	P	3	80	3
E	P	3	90	3
E	P	3	100	3
E	P	3	110	3
E	P	3	120	3

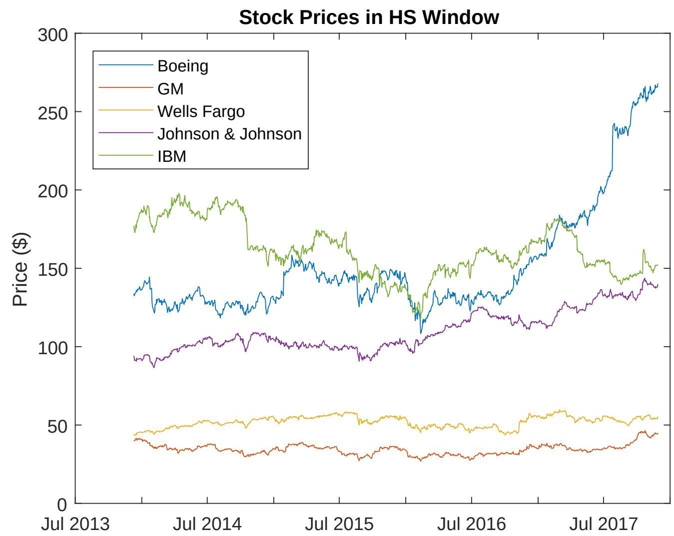
Table B.2: 60 fictitious options that are used to construct different portfolios in the historical simulation method. The first 30 options are American whereas the following 30 are their exact European counterparts.

Option Type A = American E = European	C = Call P = Put	Underlying Security	Strike Price	Time to Maturity
A	C	1	250	1
A	C	1	265	1
A	C	1	275	1
A	P	1	250	1
A	P	1	265	1
A	P	1	275	1
A	C	2	34	3
A	C	2	44	3
A	C	2	54	3
A	P	2	34	3
A	P	2	44	3
A	P	2	54	3
A	C	3	44	1
A	C	3	54	1
A	C	3	64	1
A	P	3	44	1
A	P	3	54	1
A	P	3	64	1
A	C	4	127	3
A	C	4	137	3
A	C	4	147	3
A	P	4	127	3
A	P	4	137	3
A	P	4	147	3
A	C	5	143	1
A	C	5	153	1
A	C	5	163	1
A	P	5	143	1
A	P	5	153	1
A	P	5	163	1
E	C	1	250	1
E	C	1	265	1
E	C	1	275	1
E	P	1	250	1
E	P	1	265	1
E	P	1	275	1
E	C	2	34	3
E	C	2	44	3
E	C	2	54	3
E	P	2	34	3
E	P	2	44	3
E	P	2	54	3
E	C	3	44	1
E	C	3	54	1
E	C	3	64	1
E	P	3	44	1
E	P	3	54	1
E	P	3	64	1
E	C	4	127	3
E	C	4	137	3
E	C	4	147	3
E	P	4	127	3
E	P	4	137	3
E	P	4	147	3
E	C	5	143	1
E	C	5	153	1
E	C	5	163	1
E	P	5	143	1
E	P	5	153	1
E	P	5	163	1

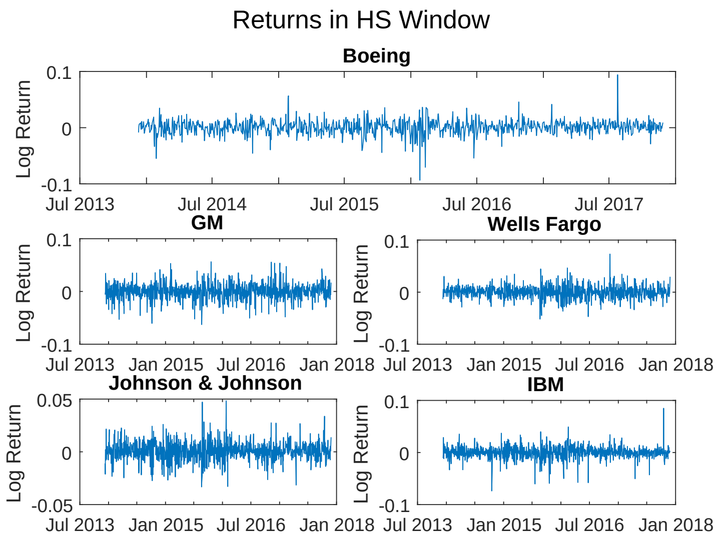
Appendix C

Portfolio Data

In Figure C.1 and panel (a) on the following page, we see the underlying stocks' movements in the 1000 day window employed in the historical simulation method. The stock prices seen in this panel were then transformed to the logarithmic returns we see in panel (b) intended for the probabilistically weighted random draws.



(a) *Stock prices that were transformed to logarithmic returns.*



(b) *Logarithmic returns for HS draws.*

Figure C.1: *Market data for the HS window of 1000 trading days between 2013-12-10 and 2017-11-29.*