TECHNISCHE UNIVERSITÄT KAISERSLAUTERN
FACHBEREICH MATHEMATIK

American-style Option Pricing and Improvement of Regression-based Monte Carlo Methods by Machine Learning Techniques

Tang, Songyin

1.Gutachter: Prof. Dr. Ralf Korn
2.Gutachter: Prof. Dr. Ralf Werner

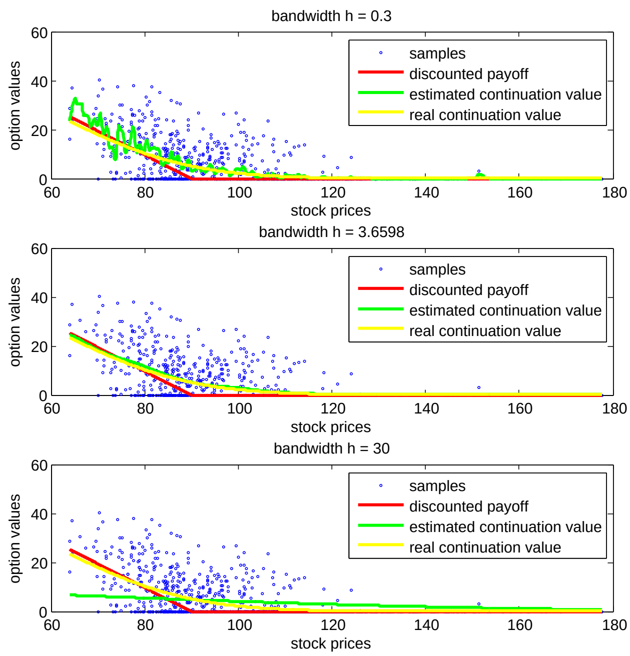
Datum der Disputation: 31. August 2015

Vom Fachbereich Mathematik der Technischen Universität Kaiserslautern
zur Verleihung des akademischen Grades
Doktor der Naturwissenschaften (Doctor rerum naturalium, Dr. rer. nat.)
genehmigte Dissertation

D 386

American-style Option Pricing and Improvement of Regression-based Monte Carlo Methods by Machine Learning Techniques

Tang, Songyin TU Kaiserslautern



1. Gutachter: **Prof. Dr. Ralf Korn**
2. Gutachter: **Prof. Dr. Ralf Werner**

Für Meine Familie

Tong, Xing

Du bist mein, ich bin dein.
Dessen sollst du gewiss sein.
Du bist verschlossen in meinem Herzen.
Verloren ist das Schlüsselein.
Du musst für immer drinnen sein.

Tang, Zhixing

Du bist das größte und beste Geschenk,
das ich von Gott Jehova bekommen habe.
Ich werde dich immer lieben und beschützen.

Acknowledgements

First and foremost, I would like to express my deepest respect and appreciation to my supervisor Prof. Dr. Ralf Korn, who had an open ear for my theoretical problems and always gave me useful advice and valuable recommendations throughout my PhD study. Without his help and supervision, my dissertation would not be possible. I am also very grateful to Prof. Dr. Ralf Werner for accepting to act as a referee for my dissertation.

Second, I sincerely thank the Financial Mathematics Group at the University of Kaiserslautern for giving me the opportunity to do PhD research. Many thanks go to my colleagues at the department for their help and advice: Prof. Dr. Jörn Sass, Prof. Dr. Frank Seifried, Cornelia Türk, Dr. Christoph Belak, Dr. Alona Futorna, Sebastian Geissel, Elisabeth Leoff, Lihua Chen, Dr. Yaroslav Melnyk, The Anh Nguyen, Dr. Ishak Norizarina, William Ntambara, Dr. Giles-Arnaud Nzouankeu Nana, Dr. Thu Tran and so on. Special big thanks go to Dr. Qian Liang for encouraging and helping me.

Third, I cordially acknowledge the Center for Mathematical and Computational Modelling $(CM)^2$ for their financial support. I work with the Electrical Engineering Group in the joint project "HOPP" (Hardware assisted Acceleration for Monte Carlo Simulations in Financial Mathematics with a particular Emphasis on Option Pricing). Big thanks go to the members of the project: Prof. Dr. Norbert Wehn, Javier Alejandro Varela, Dr. Christian de Schryver, Christian Brugger, Dr. Anton Kostiuk, Steffen Omland, Dr. Henning Marxen and so on.

Finally, I am very grateful to my parents Jian Tang and Guiping Cui, to my dear wife Xing Tong and my lovely daughter Zhixing Tang, and also to my bible teachers Jochen & Heike Sternike, for their continuous patience, love and encouragement.

Abstract

In this dissertation, we discuss how to price American-style options. Our aim is to study and improve the regression-based Monte Carlo methods. In order to have good benchmarks to compare with them, we also study the tree methods.

In the second chapter, we investigate the tree methods specifically. We do research firstly within the Black-Scholes model and then within the Heston model. In the Black-Scholes model, based on Müller's work [36], we illustrate how to price one dimensional and multidimensional American options, American Asian options, American lookback options, American barrier options and so on. In the Heston model, based on Sayer's research [39], we implement his algorithm to price one dimensional American options. In this way, we have good benchmarks of various American-style options and put them all in the appendix.

In the third chapter, we focus on the regression-based Monte Carlo methods theoretically and numerically. Firstly, we introduce two variations, the so called "Tsitsiklis-Roy method" and the "Longstaff-Schwartz method". Secondly, we illustrate the approximation of American option by its Bermudan counterpart. Thirdly we explain the source of low bias and high bias. Fourthly we compare these two methods using in-the-money paths and all paths. Fifthly, we examine the effect using different number and form of basis functions. Finally, we study the Andersen-Broadie method and present the lower and upper bounds.

In the fourth chapter, we study two machine learning techniques to improve the regression part of the Monte Carlo methods: Gaussian kernel method and kernel-based support vector machine. In order to choose a proper smooth parameter, we compare fixed bandwidth, global optimum and suboptimum from a finite set. We also point out that scaling the training data to $[0, 1]$ can avoid numerical difficulty. When out-of-sample paths of stocks are simulated, the kernel method is robust and even performs better in several cases than the Tsitsiklis-Roy method and the Longstaff-Schwartz method. The support vector machine can keep on improving the kernel method and needs less representations of old stock prices during prediction of option continuation value for a new stock price.

In the fifth chapter, we switch to the hardware (FGPA) implementation of the Longstaff-Schwartz method and propose novel reversion formulas for the stock price and volatility within the Black-Scholes and Heston models. The test for this formula within the Black-Scholes model shows that the storage of data is reduced and also the corresponding energy consumption.

Contents

List of Figures

List of Tables

List of Algorithms

1 Foundations

In this chapter, we deliver the basic mathematical and financial concepts, definitions and notations, which build the foundation of this thesis. The research focuses on the stability and improvement of regression-based Monte Carlo methods for pricing American-style options. Hence we have five questions to answer:

1. What kind of American-style options do we discuss?
2. What kind of financial models do we use to describe the movement of the stock prices?
3. What kind of other numerical methods should we apply in order to have benchmarks to compare with the results by Monte Carlo methods?
4. What kind of techniques can we make use of, so that we can improve the regression part of the Monte Carlo methods?
5. Do we need some mathematical changes when we design and implement the Monte Carlo methods on hardware instead of software?

These questions are replied in the following sections. This chapter is mainly based on Bishop [6], Glasserman [15], Hull [20], Korn [25] [26] and London[32].

1.1 Option Types

A derivative is defined as a financial instrument for which the value depends on its underlying asset. The derivatives market in the world are divided into five major classes, see Hull [20]. They are interest rate derivatives, equity derivatives, foreign exchange derivatives, credit derivatives and commodity derivatives. In the last 30 years derivatives have become more and more important and frequently traded. The main reason is that they attract many different types of traders, such as hedgers, speculators and arbitrageurs. A simple financial derivative is called the option.

Definition 1.1 (European Option / American Option / Bermudan Option). *European call / put options* give the holder the right to buy / sell the underlying asset at a certain date in the future for a certain price. *American options* can be exercised at any time before the maturity. *Bermduan options* are options whose holder can choose to exercise on a specified finite set of dates before the maturity.

Remark 1.2. The price in the contract is known as the *strike price* K; the date in the contract is known as the *maturity* T. It should be emphasized that an option gives the holder the right to do something. The holder does not have to exercise this right. We are usually interested in the *payoff function* of options:

- The discounted payoff functions of European call / put options are:

$$V_{\text{European}}^{\text{Call}}(0) = e^{-rT}(S(T) - K)^{+} \quad (1.1)$$
$$V_{\text{European}}^{\text{Put}}(0) = e^{-rT}(K - S(T))^{+} \quad (1.2)$$

  where $(S(T) - K)^{+} \doteq \max(S(T) - K, 0)$ and $(K - S(T))^{+} \doteq \max(K - S(T), 0)$, $S(T)$ is the underlying asset price, e.g stock price $S(t)$, at the maturity T, r is the risk-free interest rate, e^{-rT} is the discounting factor at T.

- The discounted payoff functions of American call / put options at the optimal exercise time t^{*}_{Am} are:

$$V_{\text{American}}^{\text{Call}}(0) = e^{-rt^{*}_{\text{Am}}}(S(t^{*}_{\text{Am}}) - K)^{+} \quad (1.3)$$
$$V_{\text{Ameircan}}^{\text{Put}}(0) = e^{-rt^{*}_{\text{Am}}}(K - S(t^{*}_{\text{Am}}))^{+} \quad (1.4)$$

  where $t^{*}_{\text{Am}} \in [0, T]$, $e^{-rt^{*}_{\text{Am}}}$ is the discounting factor at t^{*}_{Am}.

- The discounted payoff functions of Bermudan call / put options at the optimal exercise time t^{*}_{Be} are:

$$V_{\text{Bermudan}}^{\text{Call}}(0) = e^{-rt^{*}_{\text{Be}}}(S(t^{*}_{\text{Be}}) - K)^{+} \quad (1.5)$$
$$V_{\text{Bermudan}}^{\text{Put}}(0) = e^{-rt^{*}_{\text{Be}}}(K - S(t^{*}_{\text{Be}}))^{+} \quad (1.6)$$

  where $t^{*}_{\text{Be}} \in \{t_1, t_2, \ldots, t_m\}$ with $0 \leq t_1 \leq t_2 \ldots \leq t_m \leq T$, $e^{-rt^{*}_{\text{Be}}}$ is the discounting factor at t^{*}_{Be}.

Remark 1.3. Most equity options are American-style options, whereas most index options are European-style. Options traded on future exchanges are mainly American-style, while those traded over-the-counter are mainly European-style. Commodity options can be either style.

In the following, we introduce three payoff path-dependent options. Their payoffs all have three forms: European, Bermudan and American-styles. We will discuss how to price their Bermudan and American forms in the following chapters. Here we only define their European forms for simplicity.

Definition 1.4 (Asian Option). A (discrete) *Asian option* is an option on a time average of the underlying asset price. Asian calls and puts have payoffs $(\bar{S} - K)^{+}$ and $(K - \bar{S})^{+}$, where $\bar{S}$ is the average price of the stock prices over the discrete set of monitoring dates $t_1, \ldots, t_n$. $\bar{S}$ has two forms: arithmetic average and geometric average.

$$\bar{S} = \frac{1}{n}\sum_{i=1}^{n} S(t_i) \qquad \text{arithmetic average}$$
$$\bar{S} = (\prod_{i=1}^{n} S(t_i))^{\frac{1}{n}} \qquad \text{geometric average}$$

Definition 1.5 (Barrier Option). A *barrier option* is an option whose payoff depends on whether the path of the underlying asset has reached a barrier B, which is a certain predetermined level.
Knock-out barrier option is extinguished if the stock price crosses the barrier with the payments:

$$(S(T)-K)^+ \cdot \mathbb{1}\left\{\max_{t\in[0,T]} S(t) < B\right\} \qquad \text{up-and-out barrier call}$$
$$(K-S(T))^+ \cdot \mathbb{1}\left\{\max_{t\in[0,T]} S(t) < B\right\} \qquad \text{up-and-out barrier put}$$
$$(S(T)-K)^+ \cdot \mathbb{1}\left\{\min_{t\in[0,T]} S(t) > B\right\} \qquad \text{down-and-out barrier call}$$
$$(K-S(T))^+ \cdot \mathbb{1}\left\{\min_{t\in[0,T]} S(t) > B\right\} \qquad \text{down-and-out barrier put}$$

Knock-in barrier option springs into existence if the stock price crosses the barrier with the payments:

$$(S(T)-K)^+ \cdot \mathbb{1}\left\{\max_{t\in[0,T]} S(t) \geq B\right\} \qquad \text{up-and-in barrier call}$$
$$(K-S(T))^+ \cdot \mathbb{1}\left\{\max_{t\in[0,T]} S(t) \geq B\right\} \qquad \text{up-and-in barrier put}$$
$$(S(T)-K)^+ \cdot \mathbb{1}\left\{\min_{t\in[0,T]} S(t) \leq B\right\} \qquad \text{down-and-in barrier call}$$
$$(K-S(T))^+ \cdot \mathbb{1}\left\{\min_{t\in[0,T]} S(t) \leq B\right\} \qquad \text{down-and-in barrier put}$$

Definition 1.6 (Lookback Option). A *lookback option* is an option whose payoff depends on the maximum or minimum of the stock price achieved during a certain period with the payments:

$$\left(\max_{t\in[0,T]} S(t) - K\right)^+ \qquad \text{maximum lookback call}$$
$$\left(K - \max_{t\in[0,T]} S(t)\right)^+ \qquad \text{maximum lookback put}$$
$$\left(\min_{t\in[0,T]} S(t) - K\right)^+ \qquad \text{minimum lookback call}$$
$$\left(K - \min_{t\in[0,T]} S(t)\right)^+ \qquad \text{minimum lookback put}$$

Besides one dimensional American-style options, we also test regression-based Monte Carlo methods for pricing multidimensional American-style options, which are introduced as follows.

Definition 1.7 (Basket Option). A *basket option* is an option on a portfolio of underlying assets prices $\{S_1, \cdots, S_d\}$ and has a payoff of, e.g for equal weight:

$$\left(\frac{1}{d}\sum_{i=1}^{d} S_i(T) - K\right)^+ \qquad \text{arthimetic average basket call}$$

$$\left(K - \frac{1}{d}\sum_{i=1}^{d} S_i(T)\right)^+ \qquad \text{arthimetic average basket put}$$

$$\left((\prod_{i=1}^{d} S_i(T))^{\frac{1}{d}} - K\right)^+ \qquad \text{geometric average basket call}$$

$$\left(K - (\prod_{i=1}^{d} S_i(T))^{\frac{1}{d}}\right)^+ \qquad \text{geometric average basket put}$$

Definition 1.8 (Outperformance Option). *Outperformance options* are options on the maximum or minimum of multiple assets with the payments:

$$(\max\{S_1(T), S_2(T), \cdots, S_d(T)\} - K)^+ \qquad \text{maximum outperformance call}$$

$$(K - \max\{S_1(T), S_2(T), \cdots, S_d(T)\})^+ \qquad \text{maximum outperformance put}$$

$$(\min\{S_1(T), S_2(T), \cdots, S_d(T)\} - K)^+ \qquad \text{minimum outperformance call}$$

$$(K - \min\{S_1(T), S_2(T), \cdots, S_d(T)\})^+ \qquad \text{minimum outperformance put}$$

Remark 1.9. We notice that although one dimensional Asian geometric average options and multidimensional basket geometric average options are seldom traded in practice, they are regarded as useful tools to test cases for computational efficiency of different numerical methods, if we discuss within a Black-Scholes model.

1.2 Financial Models

In this dissertation, we study regression-based Monte Carlo methods for pricing American-style options in two different frameworks: *Black-Scholes model* and *Heston model*. Black-Scholes model is a constant volatility model, while Heston model is a type of stochastic volatility model. First we study the form of Black-Scholes model and the Black-Scholes formula.

Black-Scholes Model

We consider a complete probability space $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\in[0,T]}, \mathbb{Q})$ with $\mathbb{Q}$ being the risk-neutral measure, $(\mathcal{F}_t)_{t\in[0,T]}$ denoting the Brownian filtration and $W = (W(t))_{t\in[0,T]} = (W_1(t), \cdots, W_n(t))_{t\in[0,T]}$ as a correlated n-dimensional Brownian motion. The risk-neutral dynamics of the stock prices in the Black-Scholes model are given by:

$$\frac{\mathrm{d}S_i(t)}{S_i(t)} = (r-\delta)\mathrm{d}t + \sigma_i \mathrm{d}W_i(t), \quad i = 1, \ldots, n \tag{1.7}$$

where r is the risk-free interest rate, δ is the continuous dividend, σ_i is the constant volatility of stock $S_i(t)$, $S_i(0)$ is the initial stock price, each W_i is a standard one-dimensional Brownian motion, the instantaneous correlation of W_i and W_j is denoted by ρ_{ij}:

$$\mathbb{C}orr\left[\frac{\mathrm{d}S_i(t)}{S_i(t)}, \frac{\mathrm{d}S_j(t)}{S_j(t)}\right] = \rho_{ij}\mathrm{d}t$$

We denote $\Sigma = (\sigma_{ij})_{i,j=1,\ldots,d} = \rho_{ij}\sigma_i\sigma_j$ as the covariance matrix. We assume the covariance to be symmetric and positive-definite. Then we can apply Cholesky decomposition to Σ and derive a lower triangular matrix $L = (l_{ij})$ with $\Sigma = LL^\top$. Thus we can rewrite the Black-Scholes model as:

$$\frac{\mathrm{d}S_i(t)}{S_i(t)} = (r-\delta)\mathrm{d}t + \sum_{j=1}^{n} l_{ij}\mathrm{d}\tilde{W}_j(t), \quad i = 1, \ldots, n \tag{1.8}$$

with $\tilde{W}_j(t)$ being uncorrelated Brownian motions.

Theorem 1.10 (Black-Scholes Formula). Consider the Black-Scholes market model with dimension $n = 1$. Then the price $V^{Call}(0) = \mathbb{E}^{\mathbb{Q}}(e^{-rT}(S(T) - K)^+)$ of a European call option and the price $V^{Put}(0) = \mathbb{E}^{\mathbb{Q}}(e^{-rT}(K - S(T))^+)$ of a European put option with strike $K > 0$ and maturity T are given by:

$$V^{Call}(0) = e^{-\delta T}S(0)\Phi(d_1) - Ke^{-rT}\Phi(d_2) \tag{1.9}$$

$$V^{Put}(0) = Ke^{-rT}\Phi(-d_2) - e^{-\delta T}S(0)\Phi(-d_1) \tag{1.10}$$

$$d_1 = \frac{\ln\left(\frac{S(0)}{K}\right) + \left((r-\delta) + \frac{1}{2}\sigma^2\right)T}{\sigma\sqrt{T}}$$

$$d_2 = d_1 - \sigma\sqrt{T}$$

Remark 1.11 (Weakness of the Black-Scholes Model). In the Black-Scholes model, the log-returns of stocks are assumed to be normal distributed with constant volatility. Although the Black-Scholes model is still a benchmark in the financial industry, its description of the movements of stocks and options is said to be too simple by researchers and practitioners. In contrast, the volatility of stocks and options in the framework of stochastic volatility model is assumed to follow a separate stochastic process , which is closer to reality and can be explained by the fact that the volume of trading or the demand for the stock can leads to the movement of volatility.

Heston Model

There are different types of stochastic volatility models. Among them, Heston model is the most popular one. To ensure that the volatility keeps nonnegative, Heston used a square-root process for the volatility. The risk-neutral dynamics of the stock price process $S(t)$ and the variance process $V(t)$ are given by:

$$\frac{\mathrm{d}S(t)}{S(t)} = (r-\delta)\mathrm{d}t + \sqrt{V(t)}\mathrm{d}W_1(t) \tag{1.11}$$

$$\mathrm{d}V(t) = \kappa(\theta - V(t))\mathrm{d}t + \sigma\sqrt{V(t)}\mathrm{d}W_2(t) \tag{1.12}$$

with two Brownian motions having a correlation of:

$$\mathbb{C}orr(W_1(t), W_2(t)) = \rho$$

where $S(0)$ and $V(0)$ are the initial values and r and δ are risk-free interest rate and continuous dividend yield. θ is the *long-term level of the variance*, κ denotes the the *speed of mean reversion* to long-term value, σ is the *volatility of the variance*. Typically ρ is negative, which is referred to as a *leverage effect*.

There exists a (semi-) explicit pricing formula for European options in the Heston model, which is the main reason for the success of Heston model with practitioners.

Theorem 1.12 (Heston Formula). The price $V^{Call}(0)$ of a European call in the Heston model is given as:

$$V^{Call}(0) = \frac{1}{2}(S(0)e^{-qT} - Ke^{-rT}) + \frac{1}{\pi}\int_0^\infty \left(f_1(u) - Ke^{-rT}f_2(u)\right)\mathrm{d}u \tag{1.13}$$

The values $f_1(u)$ and $f_2(u)$ are given by:

$$f_1(u) = \Re\left(\frac{e^{-iu\ln(K)}\varphi(u-i)}{iue^{rT}}\right) \qquad f_2(u) = \Re\left(\frac{e^{-iu\ln(K)}\varphi(u)}{iu}\right)$$

where $\Re(\cdot)$ denotes the real part of a complex number. The Heston characteristic function $\varphi(\cdot)$ is given by:

$$\varphi(u) = e^{A_1(u)+A_2(u)+A_3(u)}$$

with

$$\begin{aligned} A_1(u) &= iu[\ln(S(0)) + (r-q)T] \\ A_2(u) &= \frac{\theta\kappa}{\sigma^2}\left((\kappa - \rho\sigma iu - h(u))T - 2\ln\left[\frac{1-g(u)e^{-h(u)T}}{1-g(u)}\right]\right) \\ A_3(u) &= \frac{V(0)(\kappa - \rho\sigma iu - h(u))(1-e^{-h(u)T})}{\sigma^2(1-g(u)e^{-h(u)T})} \end{aligned}$$

and

$$\begin{aligned} g(u) &= \frac{\kappa - \rho\sigma iu - h(u)}{\kappa - \rho\sigma iu + h(u)} \\ h(u) &= \sqrt{(\rho\sigma iu - \kappa)^2 + \sigma^2(iu + u^2)} \end{aligned}$$

with i as the imaginary unit.

1.3 Numerical Methods

Monte Carlo Method

Theorem 1.13 (Strong Law of Large Numbers). Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space and $(X_n)_{n\in\mathbb{N}}$ be a sequence of integrable, real-valued random variables that are independent, identically distributed (i.i.d.) on the space. We define $\mu = \mathbb{E}(X)$, then we have for $\mathbb{P}$-almost all $\omega \in \Omega$:

$$\frac{1}{N}\sum_{i=1}^{N} X_i(w) \quad \overset{N\to\infty}{\longrightarrow} \quad \mu \tag{1.14}$$

i.e the arithmetic mean of the realizations of X_i tends to the theoretical mean of every X_i, its expectation μ.

The basic idea of Monte-Carlo method is the strong law of large numbers. Computing an option price is computing the discounted expectation (with respect to the equivalent martingale measure) of the payoff B, thus we have the Algorithm 1.1 below:

The Monte Carlo estimator is an unbiased estimator. We use the standard deviation of the error for the Monte-Carlo estimator as a measure for the accuracy of the Monte Carlo estimator by the central limit theorem:

Theorem 1.14 (Central Limit Theorem). Let $(X_n)_{n\in\mathbb{N}}$ be a sequence of independent real-valued random variables identically distributed on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$. Assume also that they all have a finite variance $\sigma^2 = \mathbb{V}ar(X)$.

Algorithm 1.1 Monte Carlo method to price European option

Input: final payoff B

Output: option price P_N

1: Simulate N independent realizations B_i of the final payoff B.

2: Choose $P_N = \left(\frac{1}{N} \sum_{i=1}^{N} B_i \right) \cdot e^{-rT}$ as an approximation for the option price $\mathbb{E}^{\mathbb{Q}} \left(e^{-rT} B \right)$.

Then the normalized and centralized sum of these random variables converges in distribution towards the standard normal distribution:

$$\frac{\frac{1}{N} \sum_{i=1}^{N} X_i - \mu}{\frac{\sigma}{\sqrt{N}}} \xrightarrow{D} \mathcal{N}(0,1) \quad N \to \infty \tag{1.15}$$

Remark 1.15 (Confidence Interval). As we know that the asymptotic distribution of the Monte Carlo estimator is approximately normal, we obatin an approximate $(1-\alpha)$-*confidence interval* for the expectation μ:

$$\left[\frac{1}{N} \sum_{i=1}^{N} X_i - z_{1-\frac{\alpha}{2}} \frac{\sigma}{\sqrt{N}}, \quad \frac{1}{N} \sum_{i=1}^{N} X_i + z_{1-\frac{\alpha}{2}} \frac{\sigma}{\sqrt{N}} \right] \tag{1.16}$$

where $z_{1-\frac{\alpha}{2}}$ is the $1-\frac{\alpha}{2}$ -quantile of the standard normal distribution $\mathcal{N}(0,1)$. If we choose $\alpha = 5\%$, we get $1 - \frac{\alpha}{2} = 97.5\%$. As the $97.5\%-$ quantile of $\mathcal{N}(0,1)$ is about 1.96, we obtain an approximate $95\%-$ confidence interval for μ:

$$\left[\frac{1}{N} \sum_{i=1}^{N} X_i - 1.96 \frac{\sigma}{\sqrt{N}}, \quad \frac{1}{N} \sum_{i=1}^{N} X_i + 1.96 \frac{\sigma}{\sqrt{N}} \right] \tag{1.17}$$

Normally, the variance σ^2 is unknown and is estimated by its empirical counterpart sample variance $\hat{\sigma}_N^2$:

$$\hat{\sigma}_N^2 = \frac{1}{N-1} \sum_{i=1}^{N} (X_i - \frac{1}{N} \sum_{i=1}^{N} X_i)^2 \tag{1.18}$$

Binomial Trees

Binomial trees are useful for pricing a lot of European-style and American-style options. In this section, we introduce the general tree framework. In the next chapter, we study different types of trees.

Suppose the stock price $S(t)$ at time $t \in [0,T]$ has the price S_t. Assume the stock price can move up with probability p and move down with probability $q = 1-p$. After one time period Δt, if the stock price moves up, we have the value

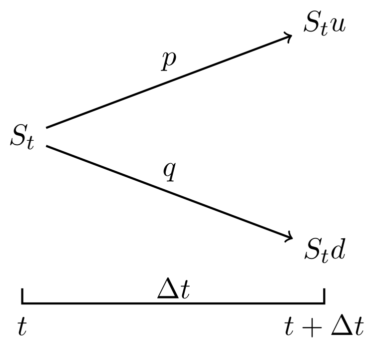


Figure 1.1: One-period movement of the stock price in a binomial tree

S_tu; and if the stock price moves down, the value is S_td. For no-arbitrage reason, we require u and d to satisfy:

$$d \;<\; e^{(r-\delta)\Delta t} \;<\; u \tag{1.19}$$

The mean and the variance of the stock price at the end of the period Δt is:

$$\mathbb{E}^{binomial}[S_{t+\Delta t}] \;=\; p(S_tu)+q(S_td) \tag{1.20}$$
$$\mathbb{V}ar^{binomial}[S_{t+\Delta t}] \;=\; p(S_tu)^2+q(S_td)^2-(p(S_tu)+q(S_td))^2 \tag{1.21}$$

Consider the Black-Scholes model, the price at the end of period Δt is a lognormal random variable (in the risk-neutral world):

$$S_{t+\Delta t} \;=\; S_te^{((r-\delta)-\frac{\sigma^2}{2})\Delta t+\sigma\sqrt{\Delta t}} \tag{1.22}$$

with the mean and the variance:

$$\mathbb{E}^{\mathbb{Q}}[S_{t+\Delta t}] \;=\; S_te^{(r-\delta)\Delta t} \tag{1.23}$$
$$\mathbb{V}ar^{\mathbb{Q}}[S_{t+\Delta t}] \;=\; S_t^2e^{(2(r-\delta)+\sigma^2)\Delta t}-(S_te^{(r-\delta)\Delta t})^2 \tag{1.24}$$

To ensure weak convergence of the tree model to the Black-Scholes model (see Korn [25]), the mean and the variance should be matched in the binomial tree and the Black-Scholes model, thus we have:

$$\mathbb{E}^{binomial}[S_{t+\Delta t}] \;=\; \mathbb{E}^{\mathbb{Q}}[S_{t+\Delta t}] \tag{1.25}$$
$$\mathbb{V}ar^{binomial}[S_{t+\Delta t}] \;=\; \mathbb{V}ar^{\mathbb{Q}}[S_{t+\Delta t}] \tag{1.26}$$

Thus, we have:

$$pu+(1-p)d \;=\; e^{(r-\delta)\Delta t} \tag{1.27}$$
$$pu^2+(1-p)d^2 \;=\; e^{(2(r-\delta)+\sigma^2\Delta t)} \tag{1.28}$$

Algorithm 1.2 Binomial tree method to price general options

1: For $N \gg 1$ set up a suitable binomial tree for the price process $S(t_i)$ in discrete time.
2: Compute the discounted expected payoff $\mathbb{E}^{(N)}(e^{-r\Delta t}B_N)$ in the discrete-time model as approximation for $\mathbb{E}^{\mathbb{Q}}(e^{-r\Delta t}B)$.

We can solve these equations and obtain the moment-matching equations:

$$p = \frac{e^{(r-\delta)\Delta t} - d}{u - d} \tag{1.29}$$

$$q = \frac{u - e^{(r-\delta)\Delta t}}{u - d} \tag{1.30}$$

$$e^{(2(r-\delta)+\sigma^2)\Delta t} = e^{(r-\delta)\Delta t}(u + d) - ud \tag{1.31}$$

The approach is illustrated in Figure 1.1:

The analysis in a one-period binomial tree can be extended to a multi-period binomial tree. To solve equations (1.29), (1.30) and (1.31) for u, d, p and q in terms of r, δ and σ, we need an additional equation since we have 3 equations with 4 unknowns. There are several choices for a second equation, for example: Cox-Ross-Rubinstein (CRR) approach [11] and Jarrow-Rudd (JR) approach [21], which will be studied in the next chapter. We present the general frame work of binomial tree method in Algorithm 1.2.

1.4 Machine Learning Techniques

The core part of the regression-based Monte Carlo methods to price American-style options is the least-squares linear regression. In order to enhance them, we need to improve the regression part of these methods. There are some notable machine learning methods which can do regression better. In this section, we introduce the basic concepts of them, see Bishop [6]. Their powerful performance will be studied in the section 4.

Machine learning theory tries to answer the questions like "Can machines do what human can do?". Assume that some training data are generated from a probability distribution, which is unknown to a machine learning system. After using some machine learning methods, the system can build a model to estimate the unknown distribution. Based on this model, the machine learner can predict accurately for new data generated from the same distribution.

Linear Regression and Bayesian Treatment

The linear regression model is a very popular model in the machine learning theory. It consists of a linear combination of some basis functions, which are nonlinear and fixed. Due to its linearity of basis functions, the linear regression has some good properties, for example, the exact solution of the least-squares problem. However the number and the form of basis functions has to be defined previously or according to the training data [1]. If it is determined by the training data by maximizing the likelihood function, the over-fitting problem is likely to occur and the model can be very complex.

Bayesian linear regression use the Bayesian theorem and assume firstly a prior probability distribution for the model parameters, and then determine a posterior distribution for the parameters by observing the training data. It gives not a point estimate for the model parameters, but a distribution. In this way, the model complexity can be determined automatically.

Neural Network

Linear regression models have obvious shortcomings. If the number of input variables is assumed to be D and the polynomial order of the regression curve function is assumed to be n, we will have D^n number of coefficients, which should be calculated by the training data. The curse of dimensionality limits its practical usefulness. One way to modify this is to use a neural network, which is a nonlinear parametric model. The term "neural network" has the original meaning from biological science, which tries to find out how a system of neurons processes information.

Kernel Method

For linear regression models and nonlinear neural networks, the training data are used to determine the model parameters and then omitted. When we make predictions for new inputs, we only use the learned parameters. Another alternative approach is the kernel methods, in which all of or some of the training data are kept and will also be used to predict. Kernel methods belong to non-parametric models, while linear regression models and neutral networks are parametric approaches. Parametric models have shortcomings that they might estimate the distribution poorly, which leads to a bad performance for predicting. For example, if the training data are generated in a multimodal way, their distribution can never be captured by a unimodal model. In kernel methods, we need a symmetric metric to measure the similarity of any two training data. This metric is called kernel function. Kernel methods make predictions based on linear combinations of kernel functions evaluated at the training data.

[1] Here, in the Monte-Carlo settings, the size of the training data means the number of simulated stock prices.

Support Vector Machine

In kernel methods, we must evaluate the kernel function for all possible pairs of training data. If the number of training data is large, making prediction would be very slow. Support vector machine (*SVM*) is a kind of sparse kernel machine. It firstly defines basis functions centred on all the training data and then select a subset of them for predicting. In the training process, it deals with a nonlinear optimization. Because its objective function is convex, any local optimum is also a global optimum. Since it has sparse solutions, it requires only a subset (not the whole set) of the training data to make predictions for new inputs, which could be much faster than the kernel method.

2 Tree Methods for Pricing American-style Options

2.1 Black-Scholes Model

In the section 1.3 we have introduced the one-period binomial model. In this chapter we will extend the analysis to multi-period case. We remember we need to define the up-movement factor u and down-movement factor d. And this leads to different cases of trees. First we study their form in one-dimension. And then multidimensional case will also be investigated. After binomial tree is constructed, we will focus on pricing different American-style options introduced in the section 1.1 and present their numerical result as benchmarks for testing various regression-based Monte Carlo methods in the chapter 4. This chapter is mainly based on Hoek [19], Liang [31], Müller [36] and Sayer[39].

2.1.1 Jarrow-Rudd Tree for One-Dimension

In the Jarrow-Rudd tree (*JR tree*), Jarrow and Rudd [21] choose equal probabilities for up- and down- movement of stocks:

$$p = q = \frac{1}{2} \tag{2.1}$$

Put this equality into moment-matching equations (1.29), (1.30) and (1.31), we obtain:

$$u + d = 2e^{(r-\delta)\Delta t} \tag{2.2}$$

$$u^2 + d^2 = 2e^{(2(r-\delta)+\sigma^2)\Delta t} \tag{2.3}$$

The exact solution is:

$$u = e^{(r-\delta)\Delta t}(1 + \sqrt{e^{\sigma^2 \Delta t} - 1}) \tag{2.4}$$

$$d = e^{(r-\delta)\Delta t}(1 - \sqrt{e^{\sigma^2 \Delta t} - 1}) \tag{2.5}$$

A most popular choice as an approximate solution is:

$$u = e^{((r-\delta)-\frac{1}{2}\sigma^2)\Delta t+\sigma\sqrt{\Delta t}} \tag{2.6}$$

$$u = e^{((r-\delta)-\frac{1}{2}\sigma^2)\Delta t-\sigma\sqrt{\Delta t}} \tag{2.7}$$

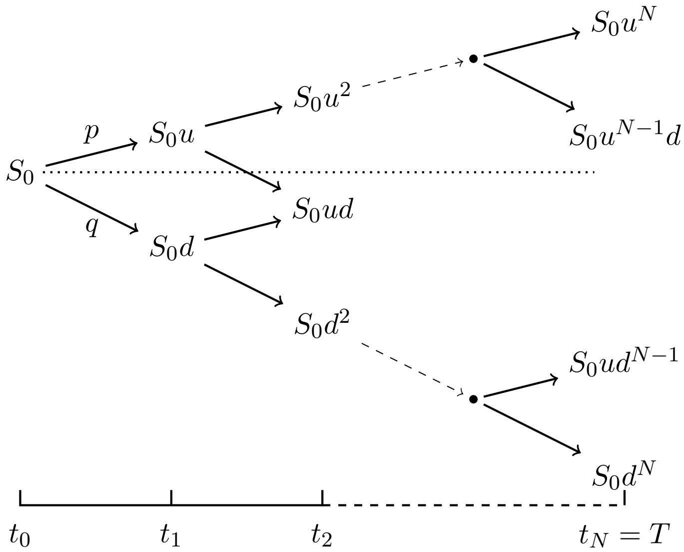


Figure 2.1: Multi-period movement of the stock price in the JR tree

Although the probabilities are equal in the JR tree, the tree is skewed since:

$$ud = e^{2((r-\delta)-\frac{1}{2}\sigma^2)\Delta t} \tag{2.8}$$
$$\neq 1 \tag{2.9}$$

The JR-tree is illustrated in Figure 2.1:

2.1.2 Cox-Ross-Rubinstein Tree for One-Dimension

Cox, Ross and Rubinstein [11] use another equation to construct a symmetric tree (*CR tree*):

$$u \cdot d = 1 \tag{2.10}$$

While the JR-tree is skewed, the CRR-tree is symmetric, since if the stock price S first goes up to Su and then goes down to Sud, it actually returns to the same price as before $Sud = S$. Wenn terms in Δt^2 and higher powers of Δt are ignored, we have an approximate solution:

$$u = e^{\sigma\sqrt{\Delta t}} \tag{2.11}$$
$$d = e^{-\sigma\sqrt{\Delta t}} \tag{2.12}$$
$$p = \frac{e^{(r-\delta)\Delta t} - d}{u - d} \tag{2.13}$$
$$q = \frac{u - e^{(r-\delta)\Delta t}}{u - d} \tag{2.14}$$

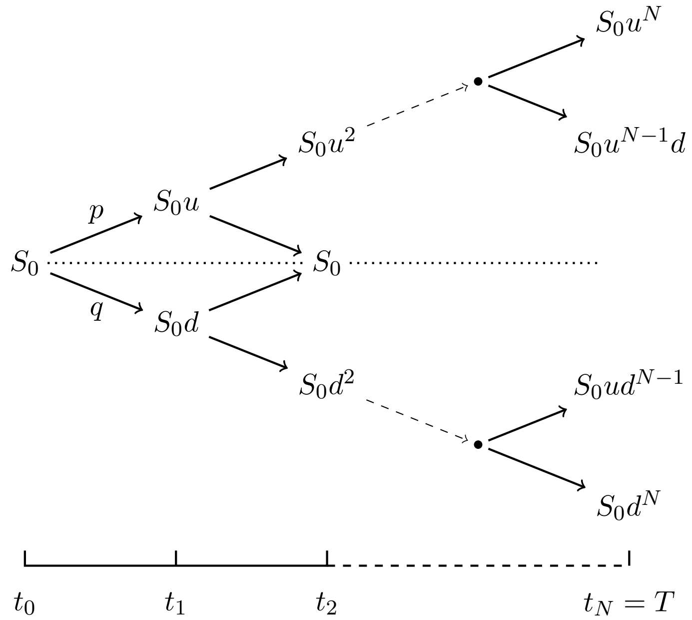


Figure 2.2: Multi-period movement of the stock price in the CRR tree

The CRR tree is illustrated in Figure 2.2:

After the CRR tree for the stock price is constructed, we can now price European, Bermudan and American options. Denote T_{ex} as the set of potential exercise dates. For European options, $T_{ex} = T = \{t_N\}$; for Bermudan options, $T_{ex} \subsetneqq \{t_1, \ldots, t_N\}$; for American options, $T_{ex} = \{t_1, \ldots, t_N\}$. We can combine algorithms for pricing these three options in one algorithm, see Algorithm 2.1.

Algorithm 2.1 CRR tree to price European / Bermduan / American options

1: Forward Step:
Denote node (i,j) as j-th node at the i-th time step, where $i = 0, 1, \ldots, N$ and $j = 0, 1, \ldots, i$. For $i = 0, 1, \ldots, N$, the stock price $S_{i,j}$ at the node (i,j) is:

$$S_{i,j} = S_0 u^j d^{i-j} \tag{2.15}$$

For potential exercise dates $t_i \in T_{ex}$, the payoff of the option is denoted as $V^{\mathrm{E}}_{i,j}$, computed by:

$$\begin{aligned} V^{\mathrm{E}}_{i,j} &= \max((S_{i,j} - K), 0) \qquad \text{for} \quad \text{Call} \\ V^{\mathrm{E}}_{i,j} &= \max((K - S_{i,j}), 0) \qquad \text{for} \quad \text{Put} \end{aligned}$$

The option price at each node is defined as $V_{i,j}$, at the end of periods, we have:

$$V_{N,j} = V^{\mathrm{E}}_{N,j} \tag{2.16}$$

2: Backward Step:
For $i = N-1, N-2, \ldots, 0$, we compute the option price at each node (i,j) over one period Δt backward:

$$V_{i,j} = e^{-(r-\delta)\Delta t}(pV_{i+1,j+1} + qV_{i+1,j}) \tag{2.17}$$

For potential exercise dates $t_i \in T_{ex}$, we should decide whether to exercise the option immediately or to hold it:

$$V_{i,j} = \max(V_{i,j}, V^{\mathrm{E}}_{i,j}) \tag{2.18}$$

3: Output: $V_{0,0}$.

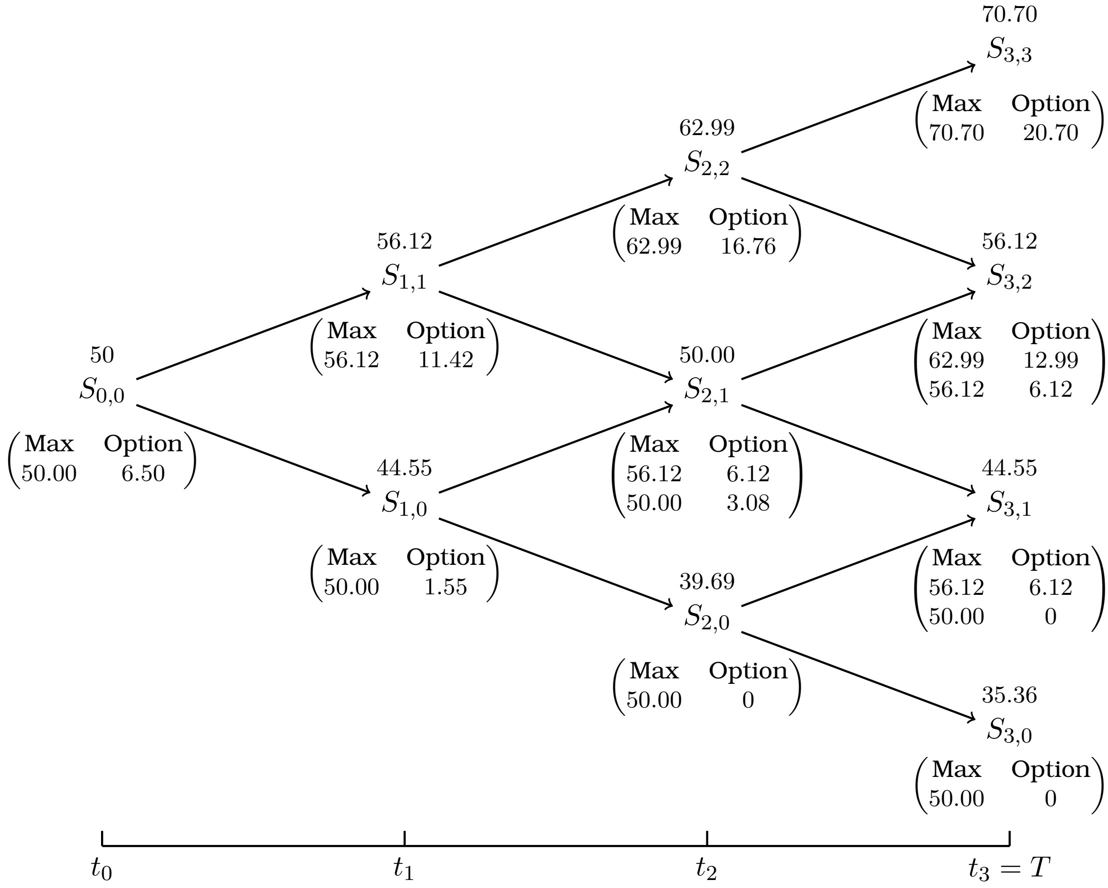


Figure 2.3: Illustration for pricing American-style (maximum) lookback call option using CRR tree

2.1.3 CRR Tree for American-style Path-Dependent Options

A path-dependent option is an option, whose payoff depends on the whole path of stock $S(t)$, not only the final value $S(t_{ex})$ at some exercise date $t_{ex} \in T_{ex}$. As presented in the section 1.1, lookback options, Asian options and barrier options are all path-dependent options. Binomial tree can be extended to price them and is computationally more efficient than Monte Carlo methods, especially when pricing their American forms.

American-style Lookback Option

We give a concrete example to explain how to price American-style lookback option using CRR tree, see Figure 2.3.

Consider a stock $S(t)$ with initial price $S(0) = S_0 = 50$, risk-free interest rate $r = 0.10$, dividend $\delta = 0$, volatility $\sigma = 0.4$, maturity of the option $T = 0.25$, fixed strike price $K = 50$. The option can be exercised at any time before maturity, say $0 < t_{ex} \leq T$. If exercised at t_{ex}, the payoff is the amount by which the maximum stock price between 0 and t_{ex} exceeds the strike K. We set up a three-step CRR tree, i.e $N = 3$. The time step length $\Delta t = T/N = 0.0833$. Up-movement factor u, down-movement factor d, up-movement probability p and down-movement probability q can be computed as: $u = 1.1224$, $d = 0.8909$, $p = 0.5073$, $q = 0.4927$.

In Figure 2.3, the top number above each node is the stock price, the left number below each node is the possible maximum stock prices for all paths to this node, the right number below each node is the option value at this node corresponding to each maximum stock price, assuming that the option is not exercised before this node.

To illustrate the backwards procedure, we consider a specific node $(2, 1)$. The stock price at this node is $S_{2,1} = 50$. The maximum stock price so far is 56.12 or 50.00.

Consider the first case, where the maximum is 56.12. If the stock price goes up, it reaches the node $(3, 2)$, the maximum value is still 56.12 and the option value is $56.12 - 50 = 6.12$. If it goes down, it reaches the node $(3, 1)$, the maximum value is still 56.12 and the option value is $56.12 - 50.00 = 6.12$. Then the value of holding the option is:

$$V_{2,1} = (0.5073 \times 6.12 + 0.4927 \times 6.12) \times e^{-0.1 \times 0.0833} = 6.07$$

The value of exercising the option immediately is:

$$V^{\mathrm{E}}_{2,1} = 56.12 - 50.00 = 6.12$$

Thus the option value at node $(2, 1)$ is:

$$V_{2,1} = \max(6.07, 6.12) = 6.12$$

The optimal strategy in this case is to exercise the option rather than hold it.

Consider the second case, where the maximum is 50.00. If the stock price goes up, it reaches the node $(3, 2)$, the maximum value becomes 56.12 and the option value is $56.12 - 50 = 6.12$. If it goes down, it reaches the node $(3, 1)$, the maximum value is still 50 and the option value is $50.00 - 50.00 = 0$. Then the value of holding the option is:

$$V_{3,1} = (0.5073 \times 6.12 + 0.4927 \times 0) \times e^{-0.1 \times 0.0833} = 3.08$$

But the value of exercising the option immediately is:

$$V^{\mathrm{E}}_{3,1} = 50.00 - 50.00 = 0$$

Thus the option value at the node $(3, 1)$ is:

$$V_{3,1} = \max(3.08, 0) = 3.08$$

The optimal strategy in this case is to hold the option rather than exercise it.

Rolling back in the same way gives the option value at the first node $(0, 0)$: 6.50. This method is computationally applicable because the number of different values of maximum stock price at each node with N time steps is no more than N.

American-style Asian Option

This procedure can also be extended to price Asian options with slight modifications. At each node, there are a lot of different values of arithmetic/geometric average stock price, thus it is often computationally expensive. However we can choose a small number of representative values, for example, the minimum of average, the maximum of average and values equally spaced between the minimum and the maximum. Then we can calculate the option value for these representatives using interpolation from known values. We also illustrate with an example, see Figure 2.4.

Consider a stock price $S(t)$ with initial price $S(0) = S_0 = 50$, risk-free interest rate $r = 0.10$, dividend $\delta = 0$, volatility $\sigma = 0.4$, maturity of the option $T = 1$, fixed strike price $K = 50$. The option can be exercised at any time before maturity, say $0 < t_{ex} \leq T$. If exercised at t_{ex}, the payoff is the amount by which the average of stock prices between 0 and t_{ex} exceeds the strike K. We set up a twenty-step CRR tree, i.e $N = 20$. The time step length $\Delta t = T/N = 0.05$. Up-movement factor u, down-movement factor d, up-movement probability p and down-movement probability q can be computed as: $u = 1.0936$, $d = 0.9144$, $p = 0.5056$, $q = 0.4944$.

To illustrate the backwards procedure, we consider a specific node $(4, 2)$, which is the central node at time 0.2 year. The stock price at this node is $S_{4,2} = 50.00$. Forward procedure shows that the maximum of averaged stock price so far is 53.83 and the minimum of averaged stock price so far is 46.65. If the stock price goes up, it reaches the node $(5, 3)$ with $S_{5,3} = 54.68$; if the stock price goes down, it reaches the node $(5, 2)$ with $S_{5,2} = 45.72$. At the node $(5, 3)$, the average price is between 47.99 and 57.39. At the node $(5, 3)$, the average price is between 43.88 and 52.48.

Suppose we also choose another two representative average price equally spaced between the minimum and the maximum at each node, which means, at the node $(4, 2)$, the representatives are: 46.65, 49.04, 51.44 and 53.83; at the node $(5, 3)$, the representatives are: 47.99, 51.12, 54.26 and 57.39; at the node $(5, 2)$, the representatives are: 43.88, 46.75, 49.61 and 52.48. Assume the option price for each representative average price at the nodes $(5, 3)$ and $(5, 2)$ are already calculated by backwards procedure, for example, at the node $(5, 3)$, if the average is 54.26, the option price is computed as 9.524.

Now we compute the option price for each representative averaged stock price at the node $(4, 2)$. For instance, we consider the averaged stock price as 51.44 at the node $(4, 2)$. If the stock price goes up and reaches the node $(5, 3)$, the new

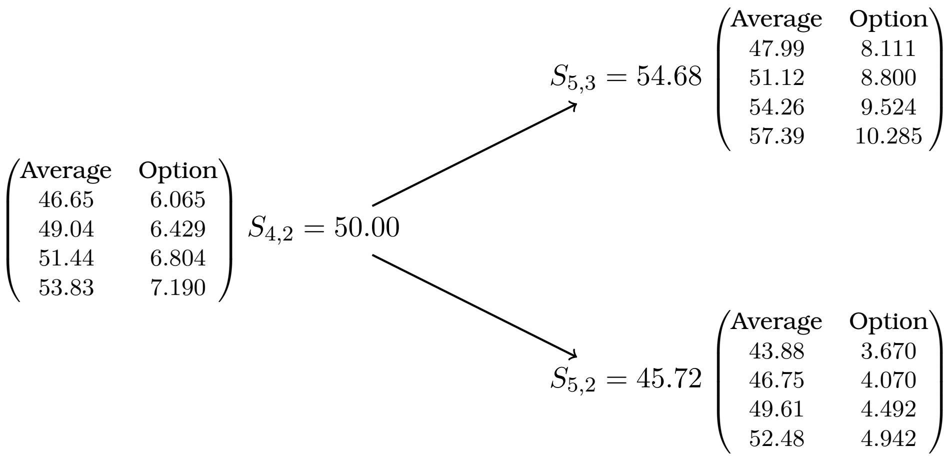


Figure 2.4: Illustration for pricing American-style (arithmetic-average) Asian call option using CRR tree

average is:

$$\frac{5 \times 51.44 + 54.68}{6} = 51.98$$

We notice that 51.98 lies between 51.12 and 54.26 at the node $(5, 3)$, thus we can compute the corresponding option value for 51.98 by interpolating:

$$\frac{(51.98 - 51.12) \times 9.524 + (54.26 - 51.98) \times 8.800}{54.26 - 51.12} = 8.998$$

If the stock price goes down and reaches the node $(5, 2)$, the new average is:

$$\frac{5 \times 51.44 + 45.72}{6} = 50.49$$

The corresponding option value for 50.49 by interpolating is:

$$\frac{(50.49 - 49.61) \times 4.942 + (52.48 - 50.49) \times 4.492}{52.48 - 49.61} = 4.630$$

Thus the value of holding the option for averaged price 51.44 at the node $(4, 2)$ is:

$$(0.5056 \times 8.998 + 0.4944 \times 4.630) \times e^{-0.1 \times 0.05} = 6.804$$

This is American-style, thus we also need to compute the value of exercising the option immediately is:

$$51.44 - 50 = 1.44$$

As $1.44 < 6.804$, the optimal strategy is to hold the option rather than exercise it and the corresponding option value is 6.804.

Rolling back in the same way gives the option value at the first node $(0, 0)$: 7.17. If the number of time steps and the number of representatives for the averages increase, the option value converges to the right result.

The same procedure can also be used to price (geometric-average) Asian options. As the geometric average of log-normally distributed random variables is again log-normally distributed, there exits another simpler way to price them. Although (geometric-average) Asian options are seldom traded in practice, they are mathematically useful to test efficiency for different numerical methods.

From the exact solution of the stock price in the Black-Scholes model:

$$S(t) = S(0)\exp\left([(r-\delta)-\frac{1}{2}\sigma^2]t+\sigma W(t)\right) \tag{2.19}$$

we obtain:

$$\left(\prod_{i=1}^{n} S(t_i)\right)^{\frac{1}{n}} = S(0)\exp\left([(r-\delta)-\frac{1}{2}\sigma^2]\frac{1}{n}\sum_{i=1}^{n} t_i + \frac{\sigma}{n}\sum_{i=1}^{n} W(t_i)\right) \tag{2.20}$$

According to Hull [20], the probability distribution of the geometric average of stock price is the same as that of the stock price with new interest rate r_{geo}, new dividend δ_{geo} and new volatility σ_{geo}. To price the (geometric-average) Asian option is to price a regular option with r_{geo}, δ_{geo} and σ_{geo}:

$$r_{geo} = r \tag{2.21}$$

$$\delta_{geo} = \frac{1}{2}\left(r+\delta+\frac{\sigma^2}{6}\right) \tag{2.22}$$

$$\sigma_{geo} = \frac{\sigma}{\sqrt{3}} \tag{2.23}$$

Typically, for European-style (geometric-average) Asian option, there exits analytic formulas by Theorem 1.10 with r_{geo}, δ_{geo} and σ_{geo} as inputs. For American-style (geometric-average) Asian option, we can set up a one-dimensional binomial tree to price, which is much faster than the previous procedure.

This dimension-reduction procedure can also be used to high dimensional case, i.e geometric average basket options, e.g with the payoff:

$$\left((\prod_{i=1}^{n} S_n(T))^{\frac{1}{n}} - K\right)^+$$

According to the equations 1.7 and 1.8, we have:

$$S_i(t) = S_i(0)\exp\left([(r-\delta)-\frac{1}{2}\sigma_i^2]t+\sum_{j=1}^{n} l_{ij}\tilde{W}_j(t)\right), \quad i=1,2,\ldots,n \tag{2.24}$$

where $L = (l_{ij})_{i,j=1,\ldots,n}$ is obtained by using the Cholesky decomposition to the covaricance matrix $\Sigma = (\sigma_{ij})_{i,j=1,\ldots,j}$ with $\Sigma = LL^\top$. Thus the geometric average on those stocks are:

$$\begin{aligned}\left(\prod_{i=1}^{n} S_n(T)\right)^{\frac{1}{n}} &= \left(\prod_{i=1}^{n} S_i(0)\right)^{\frac{1}{n}} \exp\left([(r-\delta) - \frac{1}{2n}\sum_{i=1}^{n}\sigma_i^2]T + \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n} l_{ij}\tilde{W}_j(T)\right) \\ &= \left(\prod_{i=1}^{n} S_i(0)\right)^{\frac{1}{n}} \exp\left([(r-\delta) - \frac{1}{2n}\sum_{i=1}^{n}\sigma_i^2]T + \frac{1}{n}\sum_{i=1}^{n} l_{\cdot j}\tilde{W}_j(T)\right) \\ &= \left(\prod_{i=1}^{n} S_i(0)\right)^{\frac{1}{n}} \exp\left([(r-\delta) - \frac{1}{2n}\sum_{i=1}^{n}\sigma_i^2]T + \frac{1}{n}\sqrt{\sum_{i=1}^{n} l_{\cdot j}^2}\tilde{W}(T)\right)\end{aligned} \tag{2.25}$$

where $l_{\cdot j} = \sum_{i=1}^{n} l_{ij}$ and $l_{\cdot j}^2 = (\sum_{i=1}^{n} l_{ij})^2$.

According to Glasserman [15], in the Black-Scholes model, the probability distribution of the geometric average price of several correlated/uncorrelated stocks is the same as that of a single stock price with new interest rate r_{geo}, new dividend δ_{geo} and new volatility σ_{geo}, which is again easy to be implemented in the CRR tree.

$$r_{geo} = r \tag{2.26}$$

$$\delta_{geo} = \delta + \frac{1}{2n}\sum_{i=1}^{n}\sigma_i^2 - \frac{1}{2n^2}\sum_{j=1}^{n} l_{\cdot j}^2 \tag{2.27}$$

$$\sigma_{geo} = \frac{1}{n}\sqrt{\sum_{j=1}^{n} l_{\cdot j}^2} \tag{2.28}$$

American-style Barrier Option

Barrier options are often traded in the *OTC* (over the counter) market rather than on exchanges. Normally the plain vanilla options are too expensive and do not satisfy client requirements, thus barrier options are introduced. Like in the previous sections, we also illustrate the pricing for barrier options with an example. First we discuss the knock-out type, see Figure 2.5

Consider a stock $S(t)$ with initial price $S(0) = S_0 = 100$, risk-free interest rate $r = 0.05$, dividend $\delta = 0$, volatility $\sigma = 0.20$. An American-style up-and-out barrier call option can be exercised at any time before maturity $T = 1$, say $0 < t_{ex} \leq T$. If exercised at t_{ex}, the payoff is the amount by which the stock price at t_{ex} exceeds the strike $K = 80$, given the stock price between 0 and t_{ex} does not exceed the barrier $B = 120$. We set up a three-step CRR tree, i.e $N = 3$. The time step length $\Delta t = T/N = 0.3333$. Up-movement factor u, down-movement factor d,

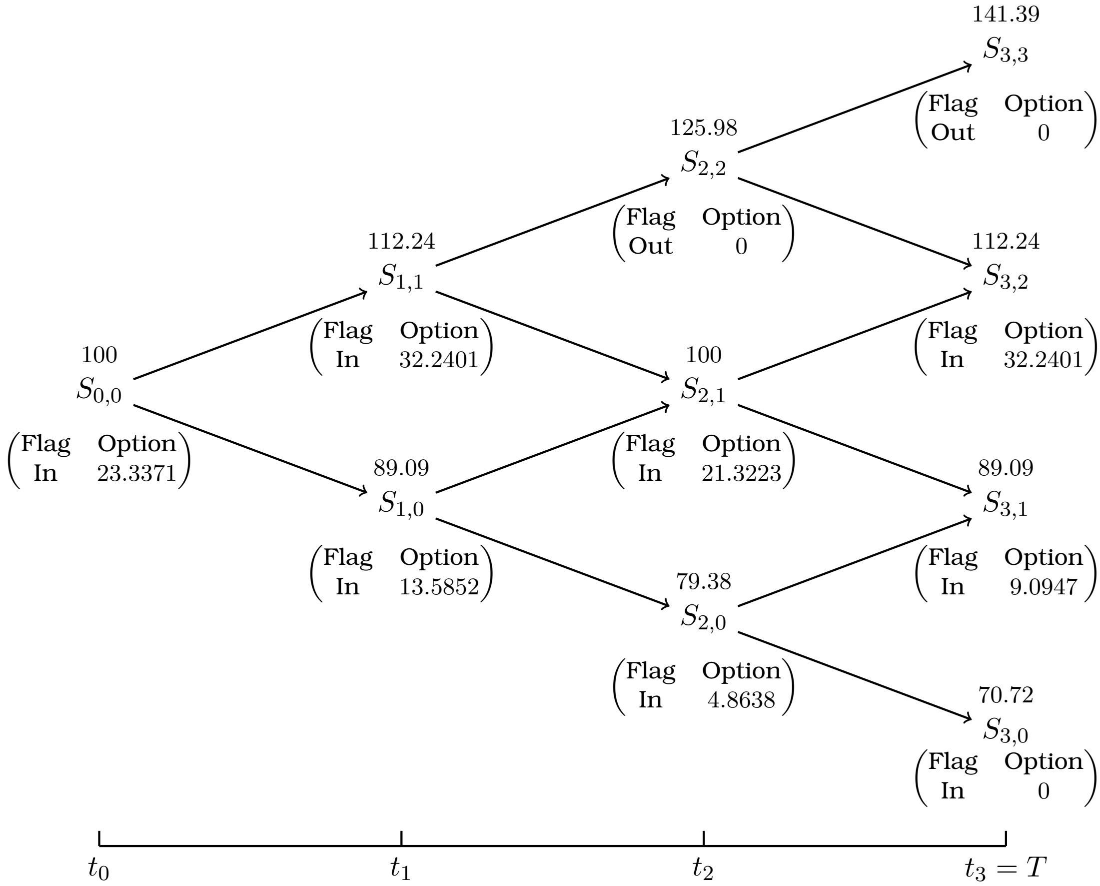


Figure 2.5: Illustration for pricing American-style (up-and-out) barrier call option using CRR tree

up-movement probability p and down-movement probability q are: $u = 1.1224$, $d = 0.8909$, $p = 0.5438$, $q = 0.4562$.

The valuation of the barrier option is the same as for the plain vanilla option except for some adjustment. In Figure 2.5, the top number above each node is the stock price, the left number below each node denotes whether the option is "up-and-out" or still "in", the right number below each node indicates the option price at this node.

First we note that in the last step, $S_{3,3} = 141.39 > 120$, thus we label "Out" under node $(3, 3)$; $S_{3,2}, S_{3,1}, S_{3,0} \leq 120$, thus we label "In" under nodes $(3, 2)$, $(3, 1)$ and $(3, 0)$. Since the option at the node $(3, 3)$ comes to existence, the corresponding option value is 0, rather than $(S_{3,3} - K)^+ = (141.39 - 80)^+ = 61.39$. The option values at other nodes in the last step can be computed the same as for the plain vanilla option.

Now we compute backwards and consider the node $(2, 1)$ for example, the value of holding the option at this node is:

$$V_{2,1} = (0.5438 \times 32.2401 + 0.4562 \times 9.0947) \times e^{-0.05\times 0.3333} = 21.3223$$

The value of exercising the option immediately at this node is:

$$V_{2,1}^{\text{E}} = (100 - 80)^{+} = 20$$

Thus the option value at this node is:

$$V_{2,1} = \max(21.3223, 20) = 21.3223$$

Consider another node $(2, 2)$ in the same step. This node is labeled with "Out", as $S_{2,2} > 120$, the value of the option at this node is then equal to 0. By rolling back we can reach the first node with option value $V_{0,0} = 23.3371$.

Second we study the type of knock-in barrier option, for which the tree is slightly different from the tree of knock-out. We take an American-style up-and-in barrier call option as an example, see Figure 2.6. The input parameters are: $S(0) = S_0 = 100$, $r = 0.05$, $\delta = 0$, $\sigma = 0.20$, $T = 1$, $K = 95$. This up-and-in option doesn't come into existence until the stock price rises to the barrier $B = 120$. In order to price such kind of option, we also need to set up a CRR tree for plain vanilla call option with same combinations of inputs. We still set up a three-step CRR tree, i.e $N = 3$. The time step length $\Delta t = T/N = 0.3333$. Then we can obtain: $u = 1.1224$, $d = 0.8909$, $p = 0.5438$, $q = 0.4562$.

In Figure 2.6, the number above each node is the stock price. The left one of two numbers under each node is the option price for a corresponding plain vanilla American call option without barrier with same input parameters, which is easier to obtain using the normal CRR tree (see Algorithm 2.1), while the right one of two numbers under each node is our barrier option price, which will be shown how to be computed. At the last step of the tree, if the stock price is above the barrier, the option price is equal to the corresponding price of the plain vanilla option, otherwise, it is set to be zero. Thus, $S_{3,3} = 141.39 > 120$ leads to the option price $V_{3,3}$ equal to the vanilla price $V_{3,3}^{\text{Vanilla}}$:

$$V_{3,3} = V_{3,3}^{\text{Vanilla}} = (141.39 - 95)^{+} = 46.40$$

Since $S_{3,2}, S_{3,1}, S_{3,0}$ are also below the barrier, the option price at those nodes are all equal to 0:

$$V_{3,2} = V_{3,1} = V_{3,0} = 0$$

Then going backwards, we consider the stock price at each node also in two cases. First case, the stock price is above the barrier, for instance, $S_{2,2} = 125.98 > 120$, then the option price is set to be the corresponding plain vanilla option at this node:

$$V_{2,2}^{\text{Vanilla}} = (0.5438 \times 46.40 + 0.4562 \times 17.24) \times e^{-0.05\times 0.3333} = 32.55$$

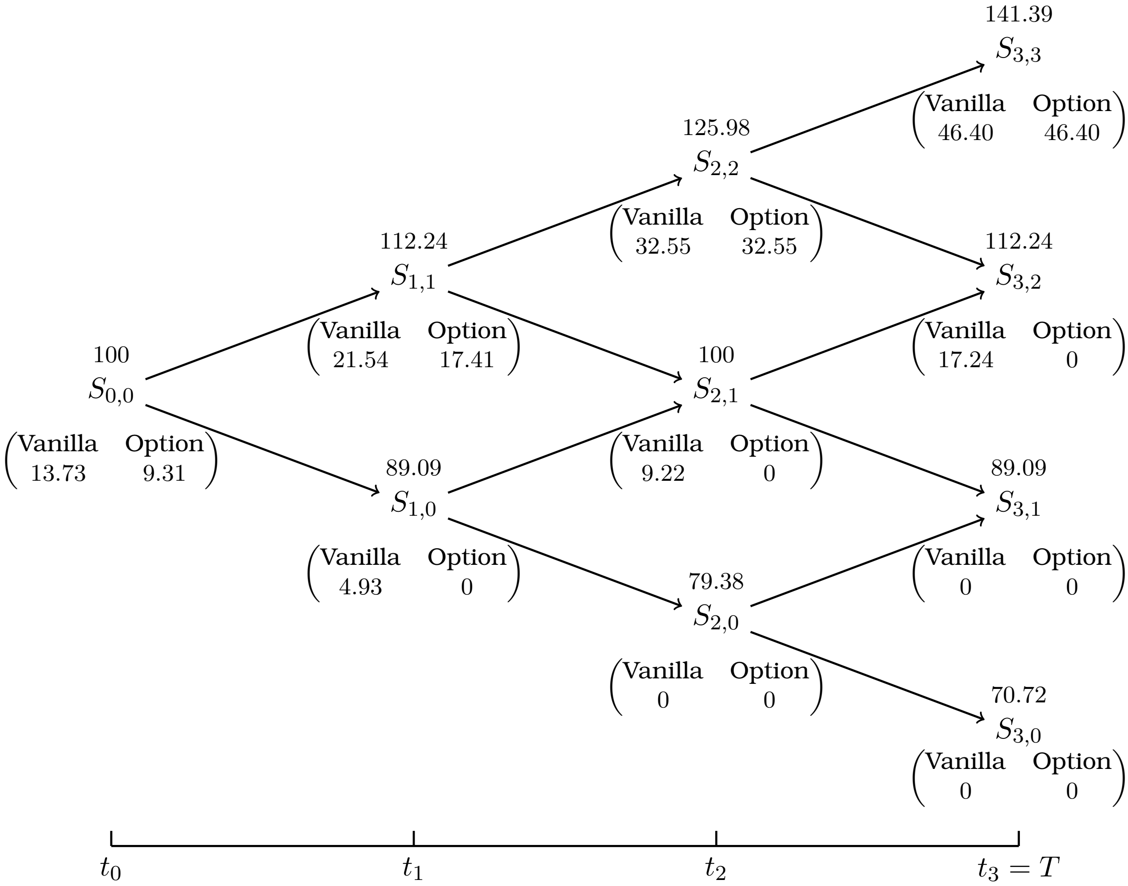


Figure 2.6: Illustration for pricing American-style (up-and-in) barrier call option using CRR tree

$$V_{2,2}^{\text{Vanilla,E}} = (125.98 - 95)^+ = 30.98$$

$$V_{2,2}^{\text{Vanilla}} = \max(32.55, 30.98) = 32.55$$

$$V_{2,2} = V_{2,2}^{\text{Vanilla}} = 32.55$$

Second case, the stock price is below the barrier, for instance, $S_{2,1} = 100 > 120$, then the option price $V_{2,1}$ should be computed by taking the up-movement and down-movement into account:

$$V_{2,1} = (0.5438 \times 0 + 0.4652 \times 0) \times e^{-0.05 \times 0.3333} = 0$$

Finally, the option value at the first node is $V_{0,0} = 9.31$, while the corresponding plain vanilla option price is $V_{0,0}^{\text{Vanilla}} = 13.73$. We can see clearly that the barrier option is cheaper than the corresponding plain vanilla option.

2.1.4 Boyle-Evnine-Gibbs Tree for High-Dimension

JR tree and CRR tree has proved to be useful in the one-dimensional case. There might be slight advantages of one tree over the other, but the practical differences are small. Both of them can be extended to high-dimensional case. The key idea is still to define up- and down- movement factors and probabilities in a proper way, such that the characteristic function of the discrete distribution in the binomial tree converges to the continuous distribution in the Black-Scholes model when the length of each time step Δt tends to zero. Boyle, Evnine and Gibbs [5] study the extension of the CRR tree by first defining the up-/down-movement factors as in the CRR tree and then choosing suitable up-/down-movement probabilities to match the expectation and variance and obtain the *BEG tree*.

First we study the case of two dimensions and then generalize to high dimensions. Consider a pair of two stocks with stock prices $\left(S_t^1, S_t^2\right)$ at time t, with volatilities σ_1, σ_2 and correlation coefficient ρ between these stocks. After a small time interval Δt, we assume that each stock can move up and move down, thus there are $2^2 = 4$ states for the pair $\left(S_{t+\Delta t}^1, S_{t+\Delta t}^2\right)$ at time $t+\Delta t$. Further we assume that the up-movement factor u and down-movement factor d satisfy:

$$
\begin{aligned}
u_1 \cdot d_1 &= 1 \\
u_2 \cdot d_2 &= 1
\end{aligned}
$$

As in the CRR tree, we have:

$$
\begin{aligned}
u_i &= e^{\sigma_i \sqrt{\Delta t}} \\
d_i &= e^{-\sigma_i \sqrt{\Delta t}}
\end{aligned}
$$

where $i = 1, 2$.

If both stock prices move up with probability p_{uu}, the pair at time $t + \Delta t$ is: $\left(S_t^1 u_1, S_t^2 u_2\right)$; if both stock prices move down with probability p_{dd}, the pair at time $t + \Delta t$ is: $\left(S_t^1 d_1, S_t^2 d_2\right)$; if first stock price moves up and second stock price moves down with probability p_{ud}, the pair at time $t + \Delta t$ is: $\left(S_t^1 u_1, S_t^2 d_2\right)$; if first stock price moves down and second stock price moves up with probability p_{du}, the pair at time $t + \Delta t$ is: $\left(S_t^1 d_1, S_t^2 u_2\right)$. The approach is illustrated in Figure 2.7.

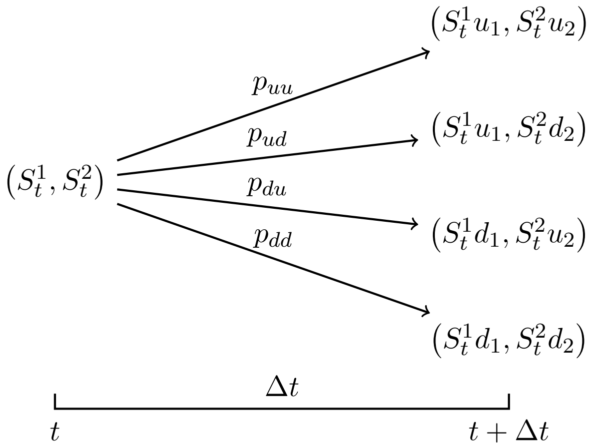


Figure 2.7: One-period movement of two stock prices in a BEG tree

The four probabilities can be solved by matching the characteristic functions:

$$p_{uu} = \frac{1}{4}\left(1+\rho+\sqrt{\Delta t}\left(\frac{r-\frac{\sigma_1^2}{2}}{\sigma_1}+\frac{r-\frac{\sigma_2^2}{2}}{\sigma_2}\right)\right) \tag{2.29}$$

$$p_{ud} = \frac{1}{4}\left(1-\rho+\sqrt{\Delta t}\left(\frac{r-\frac{\sigma_1^2}{2}}{\sigma_1}-\frac{r-\frac{\sigma_2^2}{2}}{\sigma_2}\right)\right) \tag{2.30}$$

$$p_{du} = \frac{1}{4}\left(1-\rho+\sqrt{\Delta t}\left(-\frac{r-\frac{\sigma_1^2}{2}}{\sigma_1}+\frac{r-\frac{\sigma_2^2}{2}}{\sigma_2}\right)\right) \tag{2.31}$$

$$p_{dd} = \frac{1}{4}\left(1+\rho+\sqrt{\Delta t}\left(-\frac{r-\frac{\sigma_1^2}{2}}{\sigma_1}-\frac{r-\frac{\sigma_2^2}{2}}{\sigma_2}\right)\right) \tag{2.32}$$

We note that all of these four probabilities will be positive if the length of time step Δt is small enough.

Now we discuss the high-dimensional case. Consider a group of n stocks with stock prices $(S_t^1, S_t^2, \ldots, S_t^n)$ at time t, with volatilities $\sigma_1, \sigma_2, \ldots, \sigma_n$ and correlation coefficient ρ_{ij} for stock i and stock j. After a small time interval Δt, we assume that each stock can move up and move down, thus there are 2^n states for the group $\left(S_{t+\Delta t}^1, S_{t+\Delta t}^2, \ldots, S_{t+\Delta t}^n\right)$ at time $t+\Delta t$. The up-movement factors u_i and down-movement factors d_i with $i = 1, 2, \ldots, n$ for each stock satisfy:

$$u_i \cdot d_i = 1 \tag{2.33}$$

$$u_i = e^{\sigma_i \sqrt{\Delta t}} \tag{2.34}$$

$$d_i = e^{-\sigma_i \sqrt{\Delta t}} \tag{2.35}$$

Since each node at time t has 2^n successor nodes at time $t+\Delta t$, we must choose suitable probability p_i for each state i with $i = 1, 2, \ldots, 2^n$, such that the first and second moments of the characteristic function of the log-normal distribution in the Black-Scholes model can be matched.

First, the sum of all probabilities should be equal to 1:

$$\sum_{i=1}^{2^n} p_i \quad = \quad 1 \tag{2.36}$$

Second, for each correlation coefficient ρ_{ij}, there are $((n^2 - n)/2)$ equations to be satisfied:

$$\sum_{i=1}^{2^n} \zeta_{km}(i) p_i \quad = \quad \rho_{km} \tag{2.37}$$

where $1 \leq k < m \leq n$, $\zeta_{km}(i) = 1$ if both stock i and stock j move in the same direction in the state i and $\zeta_{km}(i) = -1$ if both stock i and stock j move in the opposite directions.

Third, for each sigma σ_k, there are n equations to be satisfied:

$$\sum_{i=1}^{2^n} \eta_k(i) p_i \quad = \quad \sqrt{\Delta t} \left(\frac{r - \frac{\sigma_k^2}{2}}{\sigma_k} \right) \tag{2.38}$$

where $1 \leq k \leq n$, $\eta_k(i) = 1$ if stock k moves up in the state i and $\eta_k(i) = -1$ if stock k moves down.

For the equations (2.36), (2.37) and (2.38), there are only in total $(n^2 + n + 2)/2$ equations, while there are 2^n unknown probabilities, the number of which is more than the number of equations when $n \geq 3$. Thus there are infinite number of solutions. The solution proposed by Boyle, Evnine and Gibbs [5] is:

$$p_i \quad = \quad \frac{1}{2^n} \left(\sum_{\substack{k,m=1 \\ k<m}}^{n} \zeta_{km}(i) \rho_{km} + \sqrt{\Delta t} \sum_{k=1}^{n} \eta_k(i) \frac{r - \frac{\sigma_k^2}{2}}{\sigma_k} \right) \tag{2.39}$$

with $i = 1, 2, \ldots, 2^n$. However these probabilities are not well-defined and can still be negative even the number of time steps N increases to infinite and the length of time step Δt tends to 0.

2.1.5 Korn-Müller Tree for High-Dimension

Korn and Müller [27], [28], [36] propose a decoupling approach to the JR tree in the high-dimensional case by using Cholesky decomposition to transform correlated geometric Brownian motions to uncorrelated Brownian motions. Compared with BEG tree in the previous section, *KM tree* guarantees non-negative up- and down- movement probabilities and also better convergence performance for discontinue payoffs, such as barrier options. In this section we study KM tree by a concrete example in three dimension.

Consider three stocks $S(t) = (S_1(t), S_2(t), S_3(t))$ within the Black-Scholes model:

$$\begin{aligned} \mathrm{d}S_1(t) &= (r - \delta_1)S_1(t)\mathrm{d}t + \sigma_1 S_1(t)\mathrm{d}W_1(t) \\ \mathrm{d}S_2(t) &= (r - \delta_2)S_2(t)\mathrm{d}t + \sigma_2 S_2(t)\mathrm{d}W_2(t) \\ \mathrm{d}S_3(t) &= (r - \delta_3)S_3(t)\mathrm{d}t + \sigma_3 S_3(t)\mathrm{d}W_3(t) \end{aligned}$$

with volatilities $\sigma_1 = \sigma_2 = \sigma_3 = 0.2$, dividends $\delta_1 = \delta_2 = \delta_3 = 0.1$, interest rate $r = 0.05$, initial stock prices $S_1(0) = S_2(0) = S_3(0) = 100$. Assume the correlation coefficient of the Brownian motions ρ is:

$$\begin{aligned} \rho &= \begin{pmatrix} \rho_{11} & \rho_{12} & \rho_{13} \\ \rho_{21} & \rho_{22} & \rho_{23} \\ \rho_{31} & \rho_{32} & \rho_{33} \end{pmatrix} \\ &= \begin{pmatrix} 1 & -0.25 & 0.25 \\ -0.25 & 1 & 0.3 \\ 0.25 & 0.3 & 1 \end{pmatrix} \end{aligned}$$

Then the corresponding variance-covariance matrix Σ can be computed as:

$$\begin{aligned} \Sigma &= \begin{pmatrix} \sigma_1^2 & \rho_{12}\sigma_1\sigma_2 & \rho_{13}\sigma_1\sigma_3 \\ \rho_{21}\sigma_2\sigma_1 & \sigma_2^2 & \rho_{23}\sigma_2\sigma_3 \\ \rho_{31}\sigma_3\sigma_1 & \rho_{32}\sigma_3\sigma_2 & \sigma_3^2 \end{pmatrix} \\ &= \begin{pmatrix} 0.04 & -0.01 & 0.01 \\ -0.01 & 0.04 & 0.012 \\ 0.01 & 0.012 & 0.04 \end{pmatrix} \end{aligned}$$

Consider an American-style geometric-average basket call option of these three stocks with the payoff at any potential exercise date T_{ex} before or at maturity $T = 1$ year and strike $K = 100$:

$$g(S(T_{ex})) = \left(\left(\prod_{i=1}^{3} S_i(T_{ex}) \right)^{\frac{1}{3}} - K \right)^{+}$$

Korn and Müller define a new process X by $X_t = (\ln(S_1(t)), \ln(S_2(t)), \ln(S_3(t)))$ via log-transformation:

$$\begin{aligned} \mathrm{d}X_1(t) &= (r - \delta_1 - \tfrac{1}{2}\sigma_1^2)\mathrm{d}t + \sigma_1 \mathrm{d}W_1(t) \\ \mathrm{d}X_2(t) &= (r - \delta_2 - \tfrac{1}{2}\sigma_2^2)\mathrm{d}t + \sigma_2 \mathrm{d}W_2(t) \\ \mathrm{d}X_3(t) &= (r - \delta_3 - \tfrac{1}{2}\sigma_3^2)\mathrm{d}t + \sigma_3 \mathrm{d}W_3(t) \end{aligned}$$

with $X_1(0) = \ln(S_1(0)) = 4.6052, X_2(0) = \ln(S_2(0)) = 4.6052$ and $X_3(0) = \ln(S_3(0)) = 4.6052$.

The first step of the KM tree is to decompose the variance-covariance matrix by Cholesky decomposition $\Sigma = LL^\intercal$, thus we have L as follows:

$$L = \begin{pmatrix} 0.2000 & 0 & 0 \\ -0.0500 & 0.1936 & 0 \\ 0.0500 & 0.0749 & 0.1786 \end{pmatrix}$$

The inverse of L is computed as:

$$L^{-1} = \begin{pmatrix} 5.0000 & 0 & 0 \\ 1.2910 & 5.1640 & 0 \\ -1.9412 & -2.1651 & 5.5995 \end{pmatrix}$$

The second step of the KM tree is to transform the stock price process S into a new process Y, for which the Brownian motions are independent. We define:

$$S(t) \quad \mapsto \quad Y(t) = L^{-1}\ln(S(t)) \tag{2.40}$$

Consequently we can compute $Y(0)$ as follows:

$$\begin{aligned}
Y_1(0) &= \sum_{j=1}^{1} l_{1j}^{(-1)} \ln(S_1(0)) \\
&= l_{11}^{-1} \ln(S_1(0)) \\
&= 5.0000 \times 4.6052 \\
&= 23.0259 \\
Y_2(0) &= \sum_{j=1}^{2} l_{2j}^{(-1)} \ln(S_2(0)) \\
&= (l_{21}^{-1} + l_{22}^{-1}) \ln(S_1(0)) \\
&= (1.2910 + 5.1640) \times 4.6052 \\
&= 29.7262 \\
Y_3(0) &= \sum_{j=1}^{3} l_{1j}^{(-1)} \ln(S_3(0)) \\
&= (l_{31}^{-1} + l_{32}^{-1} + l_{33}^{-1}) \ln(S_3(0)) \\
&= (-1.9412 - 2.1651 + 5.5995) \times 4.6052 \\
&= 6.8765
\end{aligned}$$

The dynamics of Y are given by:

$$\begin{aligned}
\mathrm{d}Y_1(t) &= \alpha_1 \mathrm{d}t + \mathrm{d}\tilde{W}_1(t) \\
\mathrm{d}Y_2(t) &= \alpha_2 \mathrm{d}t + \mathrm{d}\tilde{W}_2(t) \\
\mathrm{d}Y_3(t) &= \alpha_3 \mathrm{d}t + \mathrm{d}\tilde{W}_3(t)
\end{aligned}$$

where the vector $\alpha = (\alpha_1, \alpha_2, \alpha_3)^\top$ is calculated as:

$$\begin{aligned}
\alpha &= L^{-1}\begin{pmatrix} r-\delta_1-\frac{1}{2}\sigma_1^2 \\ r-\delta_2-\frac{1}{2}\sigma_2^2 \\ r-\delta_3-\frac{1}{2}\sigma_3^2 \end{pmatrix} \\
&= \begin{pmatrix} 5.0000 & 0 & 0 \\ 1.2910 & 5.1640 & 0 \\ -1.9412 & -2.1651 & 5.5995 \end{pmatrix}\begin{pmatrix} -0.0700 \\ -0.0700 \\ -0.0700 \end{pmatrix} \\
&= \begin{pmatrix} -0.3500 \\ -0.4518 \\ -0.1045 \end{pmatrix}
\end{aligned}$$

KM tree is an extension of JR tree, thus for each node in the Y tree, the up- and down- movement probability are equal:

$$p_i = q_i = \frac{1}{2} \qquad i = 1, 2, 3$$

The up-movement factors u_i and down-movement factors d_i with $i = 1, 2, 3$ satisfy:

$$\begin{aligned}
u_i &= \alpha_i \Delta t + \sqrt{\Delta t} \\
d_i &= \alpha_i \Delta t - \sqrt{\Delta t}
\end{aligned}$$

Thus, for each Y^i tree:

$$\begin{cases}
Y^1 : & \begin{cases} u_1 = -0.3500\Delta t + \sqrt{\Delta t} \\ d_1 = -0.3500\Delta t - \sqrt{\Delta t} \end{cases} \\
Y^2 : & \begin{cases} u_2 = -0.4518\Delta t + \sqrt{\Delta t} \\ d_3 = -0.4518\Delta t - \sqrt{\Delta t} \end{cases} \\
Y^3 : & \begin{cases} u_3 = -0.1045\Delta t + \sqrt{\Delta t} \\ d_3 = -0.1045\Delta t - \sqrt{\Delta t} \end{cases}
\end{cases}$$

We illustrate the Y^i tree by 3 time steps, i.e $N = 3$, with Figure 2.8, where the number above each node is the price of Y^i at this node. $\Delta t = T/N = 0.3333$ leads directly to the results $u = (u_1, u_2, u_3)^\intercal = (0.4607, 0.4267, 0.5425)^\intercal$ and $d = (d_1, d_2, d_3)^\intercal = (-0.6940, -0.7280, -0.6122)^\intercal$. We can compute all the prices of each Y^i tree as follows:

$$Y^1 : \quad \begin{pmatrix} 23.0259 & 23.4865 & 23.9472 & 24.4079 \\ 0 & 22.3318 & 22.7925 & 23.2532 \\ 0 & 0 & 21.6378 & 22.0985 \\ 0 & 0 & 0 & 20.9438 \end{pmatrix}$$

$$Y^2 : \quad \begin{pmatrix} 29.7262 & 30.1530 & 30.5797 & 31.0064 \\ 0 & 28.9983 & 29.4250 & 29.8517 \\ 0 & 0 & 28.2703 & 28.6970 \\ 0 & 0 & 0 & 27.5423 \end{pmatrix}$$

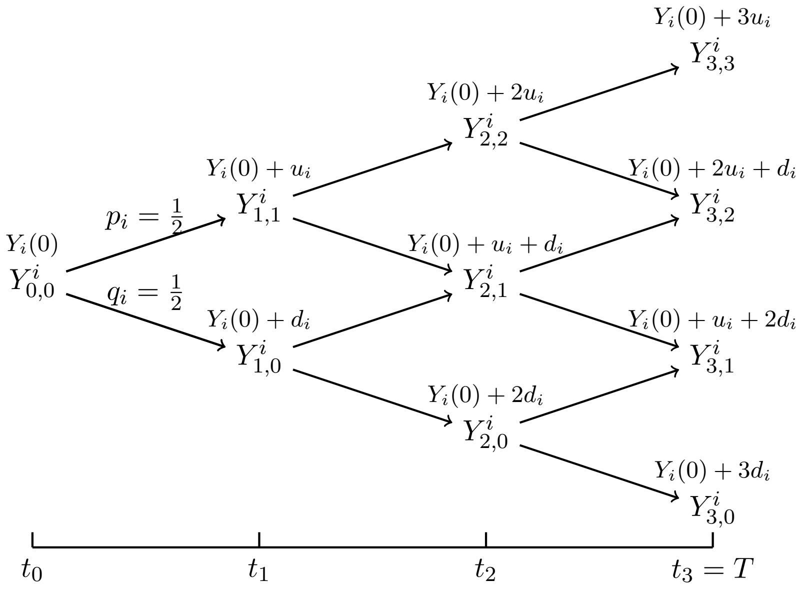


Figure 2.8: Y^i tree in the KM tree, $i = 1, 2, 3$

$$Y^3 : \quad \begin{pmatrix} 6.8765 & 7.4190 & 7.9615 & 8.5040 \\ 0 & 6.2643 & 6.8068 & 7.3493 \\ 0 & 0 & 5.6521 & 6.1946 \\ 0 & 0 & 0 & 5.0399 \end{pmatrix}$$

After each Y^i tree is constructed, we build the whole Y tree by:

$$\begin{aligned} Y &= Y^1 \otimes Y^2 \otimes Y^3 \\ &\in 1^3 \times 2^3 \times 3^3 \times 4^3 \end{aligned}$$

Thus at the beginning t_0, there is only one node in the Y tree, that is:

$$Y_{0,0} = (Y^1_{0,0}, Y^2_{0,0}, Y^3_{0,0})^\intercal = (23.0259, 29.7262, 6.8765)^\intercal$$

At time t_1, there are two nodes in each Y^i tree, thus there are $2 \times 2 \times 2 = 8$ nodes $(Y_{1,0}, Y_{1,1}, \ldots, Y_{1,7})$ in the Y tree:

$$Y_{1,0} = (Y^1_{1,0}, Y^2_{1,0}, Y^3_{1,0})^\intercal = (22.3318, 28.9983, 6.2643)^\intercal$$

$$Y_{1,1} = (Y^1_{1,0}, Y^2_{1,0}, Y^3_{1,1})^\intercal = (22.3318, 28.9983, 7.4190)^\intercal$$

$$Y_{1,2} = (Y^1_{1,0}, Y^2_{1,1}, Y^3_{1,0})^\intercal = (22.3318, 30.1530, 6.2643)^\intercal$$

$$Y_{1,3} = (Y^1_{1,0}, Y^2_{1,1}, Y^3_{1,1})^\intercal = (22.3318, 30.1530, 7.4190)^\intercal$$

$$Y_{1,4} = (Y^1_{1,1}, Y^2_{1,0}, Y^3_{1,0})^\intercal = (23.4865, 28.9983, 6.2643)^\intercal$$

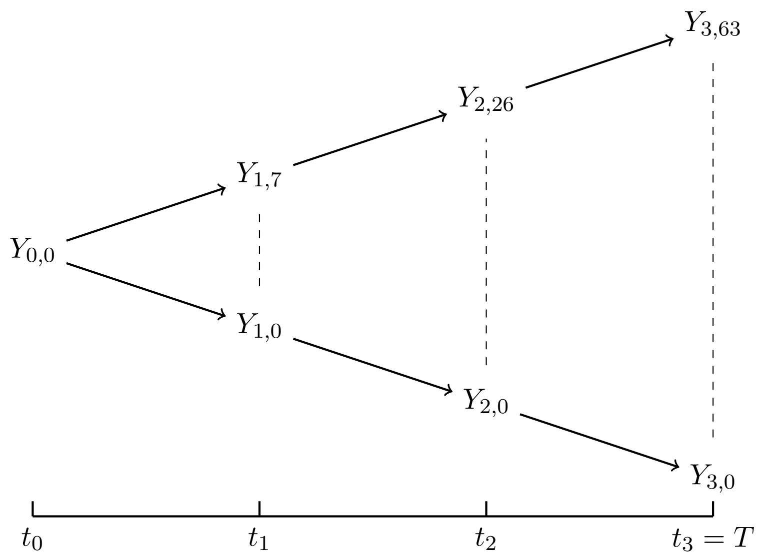


Figure 2.9: Y tree in the KM tree

$$Y_{1,5} = (Y^1_{1,1}, Y^2_{1,0}, Y^3_{1,1})^\intercal = (23.4865, 28.9983, 7.4190)^\intercal$$
$$Y_{1,6} = (Y^1_{1,1}, Y^2_{1,1}, Y^3_{1,0})^\intercal = (23.4865, 30.1530, 6.2643)^\intercal$$
$$Y_{1,7} = (Y^1_{1,1}, Y^2_{1,1}, Y^3_{1,1})^\intercal = (23.4865, 30.1530, 7.4190)^\intercal$$

Similarly at time t_2, there are $3^3 = 27$ nodes $(Y_{2,0}, Y_{2,1}, \ldots, Y_{2,26})$ and at time $t_3 = T$, there are $4^3 = 64$ nodes $(Y_{3,0}, Y_{3,1}, \ldots, Y_{3,63})$ in the Y tree. The Y tree is shown in Figure 2.9.

As there are two successive nodes after each node with transition probability $p = \frac{1}{2}$ in each Y^i tree, there are $2^3 = 8$ successive nodes after each node in the Y tree, and for each successive node, the transition probability is $p = (\frac{1}{2})^3 = \frac{1}{8}$.

The third step of the KM tree is to construct the tree for the original stock price S by applying the inverse of transformation (2.40):

$$Y(t) \quad \mapsto \quad S(t) = \exp(LY(t)) \tag{2.41}$$

As the equation (2.41) said that $Y \in 1^3 \times 2^3 \times 3^3 \times 4^3$, we also have $S \in 1^3 \times 2^3 \times 3^3 \times 4^3$. At time t_0, $S_{0,0}$ can be computed by $Y_{0,0}$ as:

$$\begin{aligned} S_{0,0} &= \exp(LY_{0,0}) \\ &= \exp\left(\begin{pmatrix} 0.2000 & 0 & 0 \\ -0.0500 & 0.1936 & 0 \\ 0.0500 & 0.0749 & 0.1786 \end{pmatrix} \cdot \begin{pmatrix} 23.0259 \\ 29.7262 \\ 6.8765 \end{pmatrix}\right) \\ &= \begin{pmatrix} 100 \\ 100 \\ 100 \end{pmatrix} \end{aligned}$$

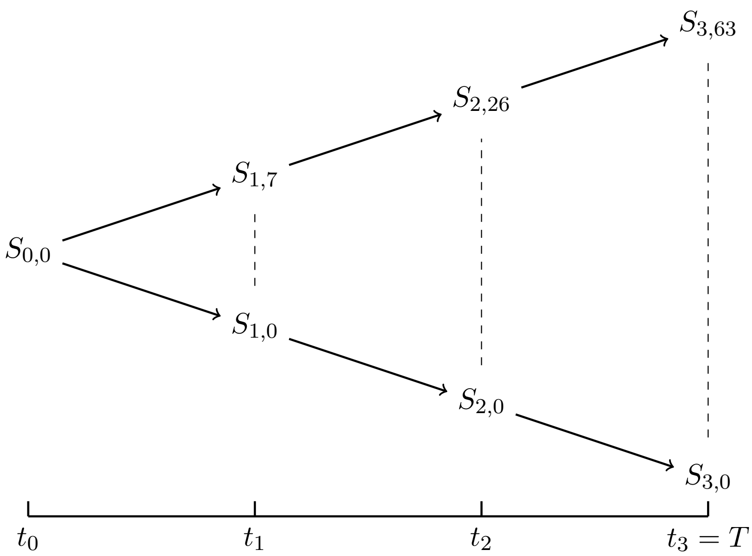


Figure 2.10: S tree in the KM tree

At time t_1, $S_{1,0}$ can be computed by $Y_{1,0}$ as:

$$\begin{aligned} S_{1,0} &= \exp(LY_{1,0}) \\ &= \exp\left(\begin{pmatrix} 0.2000 & 0 & 0 \\ -0.0500 & 0.1936 & 0 \\ 0.0500 & 0.0749 & 0.1786 \end{pmatrix} \cdot \begin{pmatrix} 22.3318 \\ 28.9983 \\ 6.2643 \end{pmatrix}\right) \\ &= \begin{pmatrix} 87.0399 \\ 89.9183 \\ 81.9928 \end{pmatrix} \end{aligned}$$

Other nodes $(S_{1,1}, S_{1,2}, \ldots, S_{1,7})$ at time t_1 can be computed similarly and other nodes at time t_2 and t_3 can also be calculated in a similar way. Thus S tree can be constructed as in Figure 2.10.

The fourth step of the KM tree is to backwards evaluate the option price at each node of the S tree. Like in the Y tree, each node in the S tree also has $2^3 = 8$ nodes afterwards, for each the transition probability is $(\frac{1}{2})^3 = \frac{1}{8}$. We illustrate this procedure by discussing a specific node $(1,7)$ with $S_{1,7} = (109.6515, 106.1415, 116.4011)^\top$. The value of exercising the option immediately is:

$$\begin{aligned} V_{1,7}^{\mathrm{E}} &= ((109.6515 \times 106.1415 \times 116.4011)^{\frac{1}{3}} - 100)^+ \\ &= 10.6504 \end{aligned}$$

There are 8 successive nodes after node $(1,7)$, see Figure 2.11. The stock prices and option values at those nodes are assumed to be already obtained by

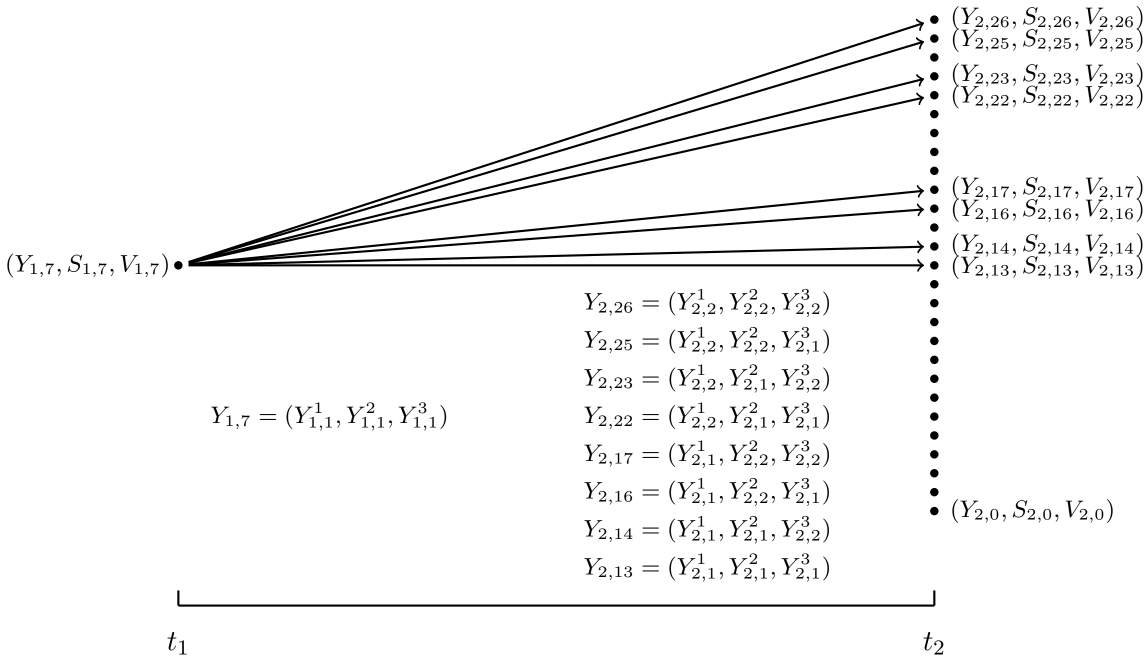


Figure 2.11: Option evaluation in the KM tree

the backwards computation:

$$
\begin{aligned}
S_{2,26} &= (120.2345, 112.6601, 135.4922)^\top, & V_{2,26} &= 22.4347\\
S_{2,25} &= (120.2345, 112.6601, 110.2444)^\top, & V_{2,25} &= 14.3015\\
S_{2,23} &= (120.2345, 90.0863, 124.2696)^\top, & V_{2,23} &= 10.4123\\
S_{2,22} &= (120.2345, 90.0863, 101.1130)^\top, & V_{2,22} &= 3.5646\\
S_{2,17} &= (95.4405, 119.3560, 127.8911)^\top, & V_{2,17} &= 13.3633\\
S_{2,16} &= (95.4405, 119.3560, 104.0597)^\top, & V_{2,16} &= 5.8327\\
S_{2,14} &= (95.4405, 95.4405, 117.2980)^\top, & V_{2,14} &= 3.1317\\
S_{2,13} &= (95.4405, 95.4405, 95.4405)^\top, & V_{2,13} &= 0.6891
\end{aligned}
$$

The value of holding the option at the node $(1, 7)$ is:

$$
\begin{aligned}
V_{1,7} &= (22.4347 + 14.3015 + 10.4123 + 3.5646 + 13.3633\\
&\quad +5.8327 + 3.1317 + 0.6891) \times \frac{1}{8} \times e^{-0.05 \times 0.3333}\\
&= 9.0639
\end{aligned}
$$

Thus the option value at this node $(1, 7)$ is:

$$
\begin{aligned}
V_{1,7} &= \max(10.6504, 9.0639)\\
&= 10.6504
\end{aligned}
$$

The optimal strategy at this node is hence to exercise the option immediately rather than hold it. By rolling back in the same way, we have the option value at the beginning $V_{0,0} = 2.8763$.

2.2 Heston Model

In the previous section, we have studied valuation of various American-style options within the Black-Scholes model. In this section, we perform research within the Heston model, which is one of the most popular stochastic volatility models. This section is mainly based on Sayer [39] and [40]. The main idea of Ruckdeschel-Sayer-Szimayer approach (*RSS tree*) is to firstly set up a binomial tree for the variance process and a trinomial tree for the stock process by matching the first and second moments in the Heston model and then to adjust the transition probabilities in each node in order to match the correlation parameter in the Heston model.

2.2.1 Ruckdeschel-Sayer-Szimayer Binomial Tree for Variance

Recall the stochastic differential equations in the section 1.2:

$$\frac{\mathrm{d}S(t)}{S(t)} = (r-\delta)\mathrm{d}t + \sqrt{V(t)}\mathrm{d}W_1(t) \tag{2.42}$$

$$\mathrm{d}V(t) = \kappa(\theta - V(t))\mathrm{d}t + \sigma\sqrt{V(t)}\mathrm{d}W_2(t) \tag{2.43}$$

$$\mathbb{C}orr(W_1(t), W_2(t)) = \rho \tag{2.44}$$

with $S(0) = S_0$ and $V(0) = V_0$. First, we consider the logarithmic transformation of the equation (2.42) by defining:

$$\begin{aligned} \mathrm{d}X(t) &= \mathrm{d}\ln\left(S(t)e^{-(r-\delta)t}\right) \\ &= \frac{1}{2}V(t)\mathrm{d}t + \sqrt{V(t)}\mathrm{d}W_1(t) \end{aligned} \tag{2.45}$$

with $X(0) = X_0 = \ln(S_0)$. Thus,we have:

$$S(t) = \left(e^{(r-\delta)t} \cdot e^{X(t)}\right)$$

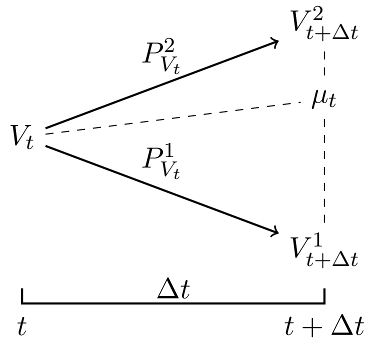


Figure 2.12: One-period movement of variance in the RSS tree

The approximation of the moments $\mathbb{E}(X(t))$, $\mathbb{E}(V(t))$, $\mathbb{V}ar(X(t))$, $\mathbb{V}ar(V(t))$ $\mathbb{C}ov(X(t),V(t))$ and $\mathbb{E}(X(t)^2V(t))$ can be computed as:

$$\mathbb{E}(X(t)) = X(t) - \frac{1}{2}V(t)\Delta t \tag{2.46}$$

$$\mathbb{E}(V(t)) = V(t) + \kappa(\theta - V(t))\Delta t \tag{2.47}$$

$$\mathbb{V}ar(X(t)) = V(t)\Delta t \tag{2.48}$$

$$\mathbb{V}ar(V(t)) = \sigma^2 V(t)\Delta t \tag{2.49}$$

$$\mathbb{C}ov(X(t),V(t)) = \sigma\rho V(t)\Delta t \tag{2.50}$$

$$\begin{aligned}\mathbb{E}(X(t)^2V(t)) &= V(t)^2\Delta t + X(t)^2(V(t) + \kappa\theta\Delta t - \kappa V(t)\Delta t) \\ &\quad -V(t)X(t)(V(t) - 2\sigma\rho)\Delta t\end{aligned} \tag{2.51}$$

At time t, the variance is assumed to be V_t. After small time interval Δt, assume that there are be two nodes at time $t+\Delta t$: $V^2_{t+\Delta t}$ and $V^1_{t+\Delta t}$, with transition probability $P^2_{V_t}$ and $P^1_{V_t}$ for each. The approach is illustrated in Figure 2.12.
Let μ_t be the drift at the node V_t:

$$\mu_t = V_t + \kappa(\theta - V_t)\Delta t$$

$V^2_{t+\Delta t}$ and $V^1_{t+\Delta t}$ must enclose μ_t, thus they can be defined as:

$$V^2_{t+\Delta t} = \frac{\sigma^2}{4}(z_t + j^2_t\sqrt{\Delta t})^2 \tag{2.52}$$

$$V^1_{t+\Delta t} = \frac{\sigma^2}{4}(z_t + j^1_t\sqrt{\Delta t})^2 \tag{2.53}$$

where z_t, j^1_t and j^2_t are defined as follows to ensure that the drift μ always lies between $V^2_{t+\Delta t}$ and $V^1_{t+\Delta t}$ and the transition probabilities $P^2_{V_t}$ and $P^1_{V_t}$ can be well defined:

$$z_t = \frac{2}{\sigma}\sqrt{V_t}$$

$$j_t^2 = \begin{cases} \lceil \frac{\frac{2}{\sigma}\sqrt{\mu_t}-z_t}{\sqrt{\Delta t}} \rceil & \text{if } \lceil \frac{\frac{2}{\sigma}\sqrt{\mu_t}-z_t}{\sqrt{\Delta t}} \rceil \text{ is odd} \\ \lceil \frac{\frac{2}{\sigma}\sqrt{\mu_t}-z_t}{\sqrt{\Delta t}} \rceil + 1 & \text{if } \lceil \frac{\frac{2}{\sigma}\sqrt{\mu_t}-z_t}{\sqrt{\Delta t}} \rceil \text{ is even} \end{cases}$$

$$j_t^1 = \begin{cases} \lfloor \frac{\frac{2}{\sigma}\sqrt{\mu_t}-z_t}{\sqrt{\Delta t}} \rfloor & \text{if } \lfloor \frac{\frac{2}{\sigma}\sqrt{\mu_t}-z_t}{\sqrt{\Delta t}} \rfloor \text{ is odd} \\ \lfloor \frac{\frac{2}{\sigma}\sqrt{\mu_t}-z_t}{\sqrt{\Delta t}} \rfloor - 1 & \text{if } \lfloor \frac{\frac{2}{\sigma}\sqrt{\mu_t}-z_t}{\sqrt{\Delta t}} \rfloor \text{ is even} \end{cases}$$

where $\lceil\cdot\rceil$ denotes the ceiling function and $\lfloor\cdot\rfloor$ denotes the floor function. The reason that j_t^2 and j_t^1 are allowed only to be odd is that we want to restrict each node to jump only to valid nodes.
The transition probabilities $P_{V_t}^2$ and $P_{V_t}^1$ can be determined by matching the first moment of the variance process in the Heston model:

$$\begin{aligned} V_{t+\Delta t}^2 \cdot P_{V_t}^2 + V_{t+\Delta t}^1 \cdot P_{V_t}^1 &= \mu_t \\ P_{V_t}^2 + P_{V_t}^1 &= 1 \end{aligned}$$

Thus we have:

$$P_{V_t}^2 = \frac{(V_t + \kappa(\theta - V_t)\Delta t) - V_{t+\Delta t}^1}{V_{t+\Delta t}^2 - V_{t+\Delta t}^1} \tag{2.54}$$

$$P_{V_t}^1 = \frac{V_{t+\Delta t}^2 - (V_t + \kappa(\theta - V_t)\Delta t)}{V_{t+\Delta t}^2 - V_{t+\Delta t}^1} \tag{2.55}$$

The so constructed binomial tree for variance process is a recombining tree, we illustrate it with a concrete example. The input parameters are: $V_0 = 0.01$, $\kappa = 1$, $\theta = 0.01$, $\sigma = 0.1$, $T = 1$, time steps $N = 10$. time step length $\Delta t = T/N = 0.1$. Look at this Figure 2.13, we can find several characteristics of this tree:

1. This variance tree is recombining but not symmetric.
2. At each time step t_i with $i = 0, 1, 2, \ldots, 10$, the number of nodes is not always equal to i, for example, at time t_9, the number of nodes is 8 rather than 9.
3. No variance node is equal to or smaller than 0, due to the stability condition $2\kappa\theta > \sigma^2$ satisfied, the smallest one is 0.0004 at nodes $(5,0)$, $(7,0)$ and $(9,0)$.
4. Small variance gives a positive drift, such that the variance tree below is truncated, for example, at node $(5,0)$, we have $V(5,0) = 0.000438612$, then:

$$\begin{aligned} \mu(5,0) &= V(5,0) + \kappa(\theta - V(5,0))\Delta t \\ &= 0.000438612 + 1 \times (0.01 - 0.000438612) \times 0.1 \\ &= 0.00139475 \\ z(5,0) &= \frac{2}{\sigma}\sqrt{V(5,0)} \\ &= \frac{2}{0.1} \times \sqrt{0.000438612} \\ &= 0.418861 \end{aligned}$$

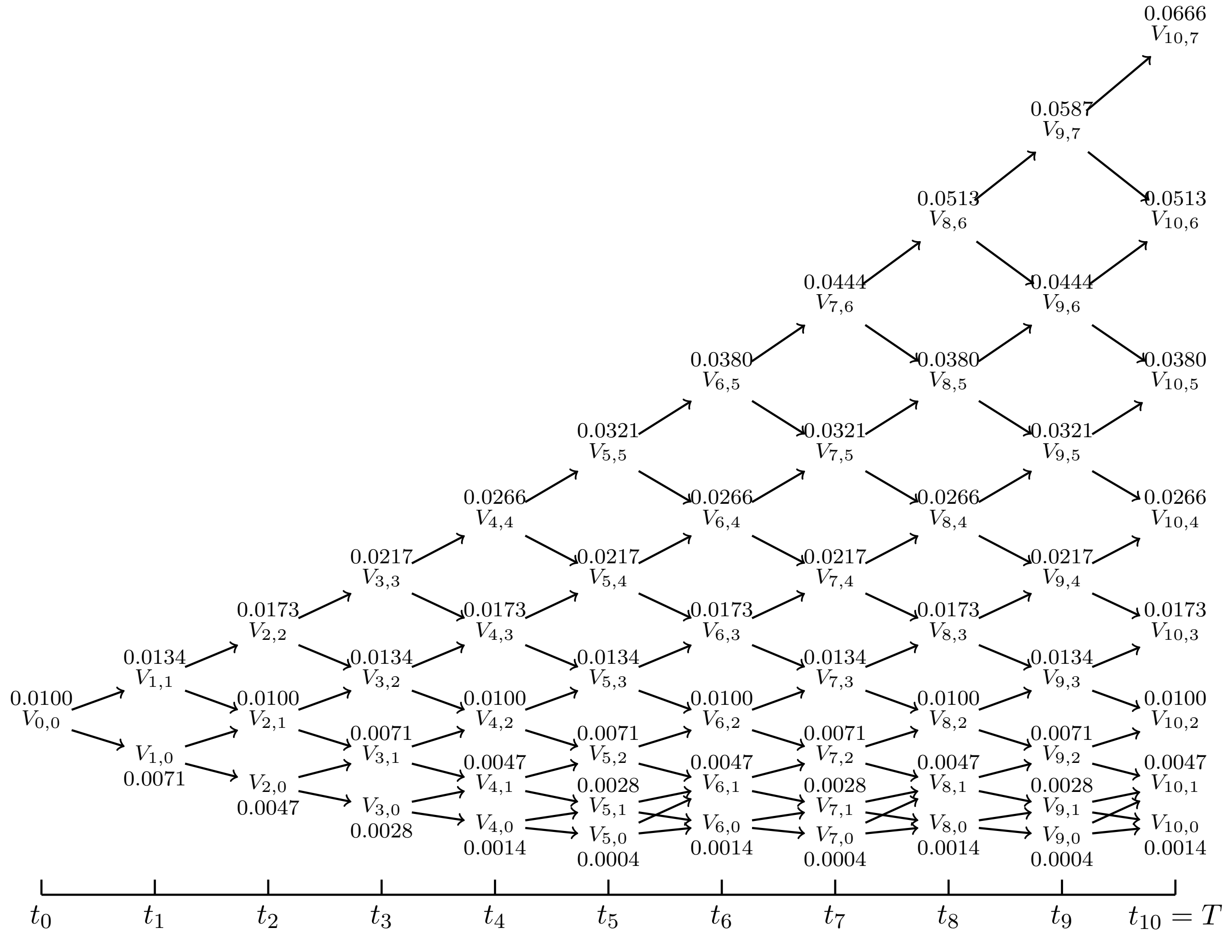


Figure 2.13: Multi-period movement of variance in the RSS tree

Then, the jump size $j^2(5,0)$ and $j^1(5,0)$ are:

$$\begin{aligned}
j^2(5,0) &= \lceil \frac{\frac{2}{\sigma}\sqrt{\mu(5,0)} - z(5,0)}{\sqrt{\Delta t}} \rceil \qquad (+1 \text{ or } +0) \\
&= \lceil \frac{\frac{2}{0.1} \times \sqrt{0.00139475} - 0.418861}{\sqrt{0.1}} \rceil \qquad (+1 \text{ or } +0) \\
&= \lceil 1.03744 \rceil + 1 \\
&= 3 \\
j^1(5,0) &= \lfloor \frac{\frac{2}{\sigma}\sqrt{\mu(5,0)} - z(5,0)}{\sqrt{\Delta t}} \rfloor \qquad (-1 \text{ or } -0) \\
&= \lfloor \frac{\frac{2}{0.1} \times \sqrt{0.00139475} - 0.418861}{\sqrt{0.1}} \rfloor \qquad (-1 \text{ or } -0) \\
&= \lfloor 1.03744 \rfloor - 0 \\
&= 1
\end{aligned}$$

Thus, the successive nodes $V(6,1)$ and $V(6,0)$ can be computed as:

$$
\begin{aligned}
V(6,1) &= \frac{\sigma^2}{4}(z(5,0)+j^2(5,0)\sqrt{\Delta t})^2 \\
&= \frac{0.1^2}{4}(0.418861+3\times\sqrt{0.1})^2 \\
&= 0.00467544 \\
V(6,0) &= \frac{\sigma^2}{4}(z(5,0)+j^1(5,0)\sqrt{\Delta t})^2 \\
&= \frac{0.1^2}{4}(0.418861+1\times\sqrt{0.1})^2 \\
&= 0.00135089
\end{aligned}
$$

2.2.2 Ruckdeschel-Sayer-Szimayer Trinomial Tree for Stock

In this section, we set up a trinomial tree for the logarithmic stock price $X(t)$, see equation (2.45). Since at each time step, the logarithmic stock price $X(t) = X_t$ depends on the variance $V(t) = V_t$, the trinomial tree is generally not recombine. In order to guarantee the characteristic of recombing, a smallest size of movement for the variance process $\hat{V} = V_0$ is introduced and thus also the corresponding smallest size of movement for the logarithmic stock price $\hat{X} = \sqrt{\hat{V}\Delta t}$ is introduced. At time t, conditioned on the variance $V(t) = V_t$, the number of smallest movements for the logarithmic stock price $\kappa(V_t)$ is defined as:

$$
\begin{aligned}
\kappa(V_t) &= \left\lceil \frac{\sqrt{V_t\Delta t + \frac{V_t^2}{4}(\Delta t)^2}}{\hat{X}} \right\rceil \\
&= \left\lceil \frac{\sqrt{V_t(4+V_t\Delta t)}}{4\hat{V}} \right\rceil
\end{aligned}
$$

The three successive nodes $X^3_{t+\Delta t}$, $X^2_{t+\Delta t}$ and $X^1_{t+\Delta t}$ at time $t+\Delta t$ are defined as:

$$X^3_{t+\Delta t} = X_t + \kappa(V_t)\sqrt{\hat{V}\Delta t} \tag{2.56}$$

$$X^2_{t+\Delta t} = X_t \tag{2.57}$$

$$X^1_{t+\Delta t} = X_t - \kappa(V_t)\sqrt{\hat{V}\Delta t} \tag{2.58}$$

The transition probabilities for those three nodes are computed as:

$$P^3_{X_t} = \frac{4V_t + V_t^2\Delta t + 2V_t\kappa(V_t)\sqrt{\hat{V}\Delta t}}{8\kappa(V_t)^2\hat{V}} \tag{2.59}$$

$$P^1_{X_t} = \frac{4V_t + V_t^2\Delta t - 2V_t\kappa(V_t)\sqrt{\hat{V}\Delta t}}{8\kappa(V_t)^2\hat{V}} \tag{2.60}$$

$$P^2_{X_t} = 1 - P^3_{t+\Delta t} - P^1_{t+\Delta t} \tag{2.61}$$

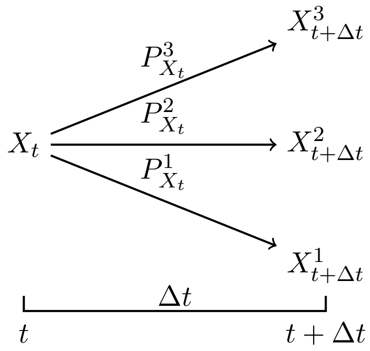


Figure 2.14: One-period movement of logarithmic stock price in the RSS tree

This approach is illustrated in Figure 2.14. We notice that due to the definition of $\kappa(V_t)$, the transition probabilities are guaranteed to be positive. And when the choice of $\hat{V}$ is fixed, a higher variance V_t leads to a higher movement for X_t.

2.2.3 Joint Probability without Correlation

In the previous two sections, we have constructed separate trees for variance process and logarithmic stock price process. In the following two sections, we combine these two trees and define joint transition probabilities. First, we discuss the case of zero correlation, i.e $\rho = 0$. At time t, we assume that the logarithmic stock price is X_t with transition probabilities $P^3_{X_t}$, $P^2_{X_t}$ and $P^1_{X_t}$ for up-, mid- or down- movement and the variance is V_t with transition probabilities $P^2_{V_t}$, $P^1_{V_t}$ for up- or down- movement. Then the $3 \times 2 = 6$ joint probabilities can be defined simply by the product in Table 2.1:

$$P^{i,j}_{(X_t,V_t)} = P^i_{X_t} \cdot P^j_{V_t} \tag{2.62}$$

where $i = 1, 2, 3$ and $j = 1, 2$. Take a look at the definitions of $P^i_{X_t}$ and $P^j_{V_t}$ in the equations (2.59), (2.60), (2.61) and (2.54), (2.55), we notice that the 6 probabilities are surprisingly the same for different values of X_t in case of zero correlation.

	$V^1_{t+\Delta t}$	$V^2_{t+\Delta t}$
$X^1_{t+\Delta t}$	$P^{1,1}_{(X_t,V_t)} = P^1_{X_t} \cdot P^1_{V_t}$	$P^{1,2}_{(X_t,V_t)} = P^1_{X_t} \cdot P^2_{V_t}$
$X^2_{t+\Delta t}$	$P^{2,1}_{(X_t,V_t)} = P^2_{X_t} \cdot P^1_{V_t}$	$P^{2,2}_{(X_t,V_t)} = P^2_{X_t} \cdot P^2_{V_t}$
$X^3_{t+\Delta t}$	$P^{3,1}_{(X_t,V_t)} = P^3_{X_t} \cdot P^1_{V_t}$	$P^{3,2}_{(X_t,V_t)} = P^3_{X_t} \cdot P^2_{V_t}$

Table 2.1: Joint probability without correlation

2.2.4 Joint Probability with Correlation

In the case of $\rho \neq 0$, the transition probabilities need to be adjusted to match the correlation parameter ρ. We define new transition probabilities $\tilde{P}^{i,j}_{(X_t,V_t)}$ by:

$$\tilde{P}^{i,j}_{(X_t,V_t)} = P^{i,j}_{(X_t,V_t)} + \theta^{i,j}_t$$

with $i = 1, 2, 3$ and $j = 1, 2$ in Table 2.2.

	$V^1_{t+\Delta t}$	$V^2_{t+\Delta t}$
$X^1_{t+\Delta t}$	$\tilde{P}^{1,1}_{(X_t,V_t)} = P^{1,1}_{(X_t,V_t)} + \theta^{1,1}_t$	$\tilde{P}^{1,2}_{(X_t,V_t)} = P^{1,2}_{(X_t,V_t)} - \theta^{1,1}_t$
$X^2_{t+\Delta t}$	$\tilde{P}^{2,1}_{(X_t,V_t)} = P^{2,1}_{(X_t,V_t)} + \theta^{2,1}_t$	$\tilde{P}^{2,2}_{(X_t,V_t)} = P^{2,2}_{(X_t,V_t)} - \theta^{2,1}_t$
$X^3_{t+\Delta t}$	$\tilde{P}^{3,1}_{(X_t,V_t)} = P^{3,1}_{(X_t,V_t)} - (\theta^{1,1}_t + \theta^{2,1}_t)$	$\tilde{P}^{3,2}_{(X_t,V_t)} = P^{3,2}_{(X_t,V_t)} + (\theta^{1,1}_t + \theta^{2,1}_t)$

Table 2.2: Joint probability with correlation

From this table, we notice that:

$$\theta^{1,2}_t = -\theta^{1,1}_t \tag{2.63}$$

$$\theta^{2,2}_t = -\theta^{2,1}_t \tag{2.64}$$

$$\theta^{3,1}_t = -(\theta^{1,1}_t + \theta^{2,1}_t) \tag{2.65}$$

$$\theta^{3,2}_t = (\theta^{1,1}_t + \theta^{2,1}_t) \tag{2.66}$$

Thus, in order to determine $\tilde{P}^{i,j}_{(X_t,V_t)}$, we only need to determine two unknowns $\theta^{1,1}_t$ and $\theta^{2,1}_t$ firstly. Sayer [39] points out that we face an optimization problem with six constraints. Because we have to guarantee the non-negative property of the probabilities, we have:

$$0 \leq P^{1,1}_{(X_t,V_t)} + \theta^{1,1}_t \leq P^1_{X_t} \equiv P^{1,1}_{(X_t,V_t)} + P^{1,2}_{(X_t,V_t)}$$

$$0 \leq P^{2,1}_{(X_t,V_t)} + \theta^{2,1}_t \leq P^2_{X_t} \equiv P^{2,1}_{(X_t,V_t)} + P^{2,2}_{(X_t,V_t)}$$

$$0 \leq P^{3,1}_{(X_t,V_t)} - (\theta^{1,1}_t + \theta^{2,1}_t) \leq P^3_{X_t} \equiv P^{3,1}_{(X_t,V_t)} + P^{3,2}_{(X_t,V_t)}$$

From the upper inequalities, we can derive six constraints:

$$-P^{1,1}_{(X_t,V_t)} \leq \theta^{1,1}_t \leq P^{1,2}_{(X_t,V_t)} \tag{2.67}$$

$$-P^{2,1}_{(X_t,V_t)} \leq \theta^{2,1}_t \leq P^{2,2}_{(X_t,V_t)} \tag{2.68}$$

$$-P^{3,2}_{(X_t,V_t)} \leq \theta^{1,1}_t + \theta^{2,1}_t \leq P^{3,1}_{(X_t,V_t)} \tag{2.69}$$

All the values $(\theta^{1,1}_t, \theta^{2,1}_t)$ which satisfy the upper constraints are denoted by $\mathcal{A}$.

The covariance between logarithmic stock price and variance at time $t + \Delta t$ according to old probability measure P and new probability measure $\tilde{P}$ are computed respectively in the following:

$$\mathbb{C}ov_P(X_{t+\Delta t}, V_{t+\Delta t}) \approx \mathbb{C}ov_P(X_t, V_t) \stackrel{eq.(2.50)}{=} \sigma\rho V_t \Delta t \tag{2.70}$$

$$\begin{aligned}
\mathbb{C}ov_{\tilde{P}}(X_{t+\Delta t}, V_{t+\Delta t}) &= \mathbb{E}_{\tilde{P}}(X_{t+\Delta t}V_{t+\Delta t}) - \mathbb{E}_{\tilde{P}}(X_{t+\Delta t})\mathbb{E}_{\tilde{P}}(V_{t+\Delta t}) \\
&\approx \mathbb{E}_{\tilde{P}}(X_{t+\Delta t}V_{t+\Delta t}) - \mathbb{E}_{P}(X_{t+\Delta t})\mathbb{E}_{P}(V_{t+\Delta t}) \\
&= V^1_{t+\Delta t}(\theta^{1,1}_t X^1_{t+\Delta t} + \theta^{2,1}_t X^2_{t+\Delta t} + \theta^{3,1}_t X^3_{t+\Delta t}) \\
&\quad + V^2_{t+\Delta t}(\theta^{1,2}_t X^1_{t+\Delta t} + \theta^{2,2}_t X^2_{t+\Delta t} + \theta^{3,2}_t X^3_{t+\Delta t}) \\
&\overset{eq.(2.63)-(2.66)}{=} (V^1_{t+\Delta t} - V^2_{t+\Delta t})[\theta^{1,1}_t(X^1_{t+\Delta t} - X^3_{t+\Delta t}) + \theta^{2,1}_t(X^2_{t+\Delta t} - X^3_{t+\Delta t})]
\end{aligned} \tag{2.71}$$

In order to obtain the best $(\theta^{1,1}_t, \theta^{2,1}_t)$, we need to minimize the squared Euclidean distance between $\mathbb{C}ov_P(X_{t+\Delta t}, V_{t+\Delta t})$ and $\mathbb{C}ov_{\tilde{P}}(X_{t+\Delta t}, V_{t+\Delta t})$. Thus we solve an optimization problem under the constraints described in the equations (2.67)-(2.69):

$$\min_{(\theta^{1,1}_t, \theta^{2,1}_t)\in\mathcal{A}} \tilde{d}(\theta^{1,1}_t, \theta^{2,1}_t) := \left(\mathbb{C}ov_P(X_{t+\Delta t}, V_{t+\Delta t}) - \mathbb{C}ov_{\tilde{P}}(X_{t+\Delta t}, V_{t+\Delta t})\right)^2 \tag{2.72}$$

The objective function can be simplified further by defining:

$$\begin{aligned}
m &:= (V^1_{t+\Delta t} - V^2_{t+\Delta t})(X^1_{t+\Delta t} - X^3_{t+\Delta t}) \\
n &:= (V^1_{t+\Delta t} - V^2_{t+\Delta t})(X^2_{t+\Delta t} - X^3_{t+\Delta t}) \\
c &:= -\sigma\rho V_t \Delta t
\end{aligned}$$

Then we have:

$$\begin{aligned}
\tilde{d}(\theta^{1,1}_t, \theta^{2,1}_t) &= (m\theta^{1,1}_t + n\theta^{2,1}_t + c)^2 \\
&= \frac{1}{2}\underbrace{\begin{pmatrix}\theta^{1,1}_t & \theta^{2,1}_t\end{pmatrix}}_{x^\intercal:=}\underbrace{\begin{pmatrix}2m^2 & 2mn \\ 2mn & 2n^2\end{pmatrix}}_{H:=}\underbrace{\begin{pmatrix}\theta^{1,1}_t \\ \theta^{2,1}_t\end{pmatrix}}_{x:=} + \underbrace{\begin{pmatrix}2mc & 2nc\end{pmatrix}}_{f^\intercal:=}\underbrace{\begin{pmatrix}\theta^{1,1}_t \\ \theta^{2,1}_t\end{pmatrix}}_{x:=} + c^2 \\
\Rightarrow \quad \tilde{d}(x) &= \frac{1}{2}x^\intercal H x + f^\intercal x + c^2
\end{aligned} \tag{2.73}$$

The six constrains (2.67)-(2.69) can be rewritten as:

$$\begin{aligned}
\underbrace{\begin{pmatrix}1 & 1 \\ -1 & -1\end{pmatrix}}_{A:=} x &\leq \underbrace{\begin{pmatrix}P^{3,1}_{X_t,V_t} \\ P^{3,2}_{X_t,V_t}\end{pmatrix}}_{b:=} \\
\Rightarrow \quad Ax &\leq b
\end{aligned} \tag{2.74}$$

$$\begin{aligned}
\underbrace{\begin{pmatrix}-P^{1,1}_{X_t,V_t} \\ -P^{2,1}_{X_t,V_t}\end{pmatrix}}_{lb:=} \leq x &\leq \underbrace{\begin{pmatrix}P^{1,2}_{X_t,V_t} \\ P^{2,2}_{X_t,V_t}\end{pmatrix}}_{ub:=} \\
\Rightarrow \quad lb \leq x &\leq ub
\end{aligned} \tag{2.75}$$

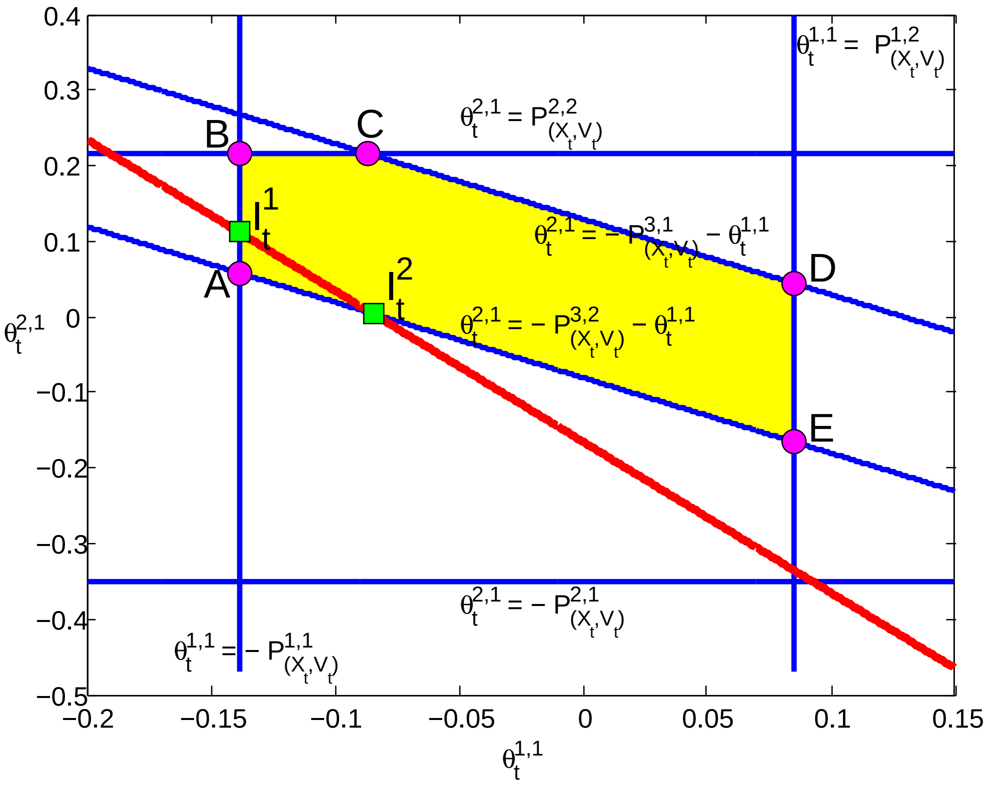


Figure 2.15: Second case: the minimizers lie in a line segment

Finding the minimum of the objective function specified by:

$$\min_x \tilde{d}(x) = \frac{1}{2}x^\intercal Hx + f^\intercal x + c^2 \quad \text{such that} \quad \begin{cases} Ax \leq b \\ lb \leq x \leq ub \end{cases}$$

is to solve a quadratic programming problem, which can be solved using the function *"quadprog"* in MATLAB.

Further, Sayer [39] indicates that there are two cases for the minimum.

- First case, the minimum $\tilde{d}(x^\star)$ is greater than 0, which means: $\mathbb{C}ov_P(X_{t+\Delta t}, V_{t+\Delta t}) \neq \mathbb{C}ov_{\tilde{P}}(X_{t+\Delta t}, V_{t+\Delta t})$, which leads to the sole minimizer $x^\star$ being one of the vertices of the admissible set $\mathcal{A}$.

- Second case, the minimum $\tilde{d}(x^\star)$ is equal to 0, which means: $\mathbb{C}ov_P(X_{t+\Delta t}, V_{t+\Delta t}) = \mathbb{C}ov_{\tilde{P}}(X_{t+\Delta t}, V_{t+\Delta t})$, which leads to set of the minimizers being a line segment I_t in $\mathcal{A}$, see Figure (2.15), where the polygon $\overline{ABCDE}$ is the admissible set $\mathcal{A}$ and $\overline{I_t^1 I_t^2}$ is the line segment I_t. In this case, the covariance is matched exactly. Among those exact matches, we need to choose the best one by matching a higher moment.

Now consider the second case. Denote $I_t^1 = (\theta_{t,1}^{1,1}, \theta_{t,1}^{2,1})$ and $I_t^2 = (\theta_{t,2}^{1,1}, \theta_{t,2}^{2,1})$ as the two boundary points of the line segment I_t comprised of all the minimizers in $\mathcal{A}$. Further, I_t^1 is assumed to be the upper left boundary, I_t^2 is assumed to be the lower right boundary. Then each point in the line segment I_t can be written as:

$$I_t(\lambda_t) \quad = \quad I_t^1 + \lambda_t(I_t^2 - I_t^1), \qquad \lambda_t \in [0,1] \tag{2.76}$$

λ_t can be chosen optimally to match the third moment:

$$\mathbb{E}((X_{t+\Delta t})^3) \quad \text{or} \quad \mathbb{E}((X_{t+\Delta t})^2 V_{t+\Delta t}) \quad \text{or} \quad \mathbb{E}(X_{t+\Delta t}(V_{t+\Delta t})^2) \quad \text{or} \quad \mathbb{E}((V_{t+\Delta t})^3)$$

In practice, we choose $\mathbb{E}((X_{t+\Delta t})^2 V_{t+\Delta t})$ to match, since option payoff depends closer on the logarithmic stock price $X_{t+\Delta t}$ than on the variance $V_{t+\Delta t}$. Thus we want to minimize the squared Euclidean distance between $\mathbb{E}_P((X_{t+\Delta t})^2 V_{t+\Delta t})$ and $\mathbb{E}_{\tilde{P}}((X_{t+\Delta t})^2 V_{t+\Delta t})$.

$$\begin{aligned}
\mathbb{E}_P((X_{t+\Delta t})^2 V_{t+\Delta t}) &\approx \mathbb{E}_P((X_t)^2 V_t) \\
&\overset{eq.(2.51)}{=} (V_t)^2\Delta t + (X_t)^2(V_t + \kappa\theta\Delta t - \kappa V_t \Delta t) - V_t X_t (V_t - 2\sigma\rho)\Delta t
\end{aligned} \tag{2.77}$$

$$\begin{aligned}
\mathbb{E}_{\tilde{P}}((X_{t+\Delta t})^2 V_{t+\Delta t}) = \; & V_{t+\Delta t}^1 \left((X_{t+\Delta t}^1)^2 \tilde{P}_{(X_t,V_t)}^{1,1} + (X_{t+\Delta t}^2)^2 \tilde{P}_{(X_t,V_t)}^{2,1} + (X_{t+\Delta t}^3)^2 \tilde{P}_{(X_t,V_t)}^{3,1}\right) \\
& + V_{t+\Delta t}^2 \left((X_{t+\Delta t}^1)^2 \tilde{P}_{(X_t,V_t)}^{1,2} + (X_{t+\Delta t}^2)^2 \tilde{P}_{(X_t,V_t)}^{2,2} + (X_{t+\Delta t}^3)^2 \tilde{P}_{(X_t,V_t)}^{3,2}\right)
\end{aligned} \tag{2.78}$$

Because:

$$\begin{aligned}
\mathbb{E}_P((X_{t+\Delta t})^2)\mathbb{E}_P(V_{t+\Delta t}) = \; & V_{t+\Delta t}^1 \left((X_{t+\Delta t}^1)^2 P_{(X_t,V_t)}^{1,1} + (X_{t+\Delta t}^2)^2 P_{(X_t,V_t)}^{2,1} + (X_{t+\Delta t}^3)^2 P_{(X_t,V_t)}^{3,1}\right) \\
& + V_{t+\Delta t}^2 \left((X_{t+\Delta t}^1)^2 P_{(X_t,V_t)}^{1,2} + (X_{t+\Delta t}^2)^2 P_{(X_t,V_t)}^{2,2} + (X_{t+\Delta t}^3)^2 P_{(X_t,V_t)}^{3,2}\right)
\end{aligned} \tag{2.79}$$

we have:

$$\mathbb{E}_{\tilde{P}}((X_{t+\Delta t})^2 V_{t+\Delta t}) \overset{eq.(2.79)}{=} \mathbb{E}_P((X_{t+\Delta t})^2)\mathbb{E}_P(V_{t+\Delta t}) + G_1 + \lambda_t G_2 \tag{2.80}$$

where:

$$\begin{aligned}
G_1 &= \theta_{t,1}^{1,1}((X_{t+\Delta t}^1)^2 - (X_{t+\Delta t}^3)^2)(V_{t+\Delta t}^1 - V_{t+\Delta t}^2) + \theta_{t,1}^{2,1}((X_{t+\Delta t}^2)^2 - (X_{t+\Delta t}^3)^2)(V_{t+\Delta t}^1 - V_{t+\Delta t}^2) \\
G_2 &= (V_{t+\Delta t}^1 - V_{t+\Delta t}^2)\left[(\theta_{t,2}^{1,1} - \theta_{t,1}^{1,1})((X_{t+\Delta t}^1)^2 - (X_{t+\Delta t}^3)^2) + (\theta_{t,2}^{2,1} - \theta_{t,1}^{2,1})((X_{t+\Delta t}^2)^2 - (X_{t+\Delta t}^3)^2)\right]
\end{aligned}$$

According to Sayer [39], the optimal $\lambda_t^\star$ is computed as:

$$\begin{aligned}
\lambda_t^\star &= \min(\max(0, \lambda_{t^0}, 1)) \\
\lambda_t^0 &= \frac{\mathbb{E}_P((X_{t+\Delta t})^2 V_{t+\Delta t}) - \mathbb{E}_P((X_{t+\Delta t})^2)\mathbb{E}_P(V_{t+\Delta t}) - G_1}{G_2}
\end{aligned} \tag{2.81}$$

where $\mathbb{E}_P((X_{t+\Delta t})^2 V_{t+\Delta t})$ is given in the equation (2.77), $\mathbb{E}_P((X_{t+\Delta t})^2)\mathbb{E}_P(V_{t+\Delta t})$ is given in the equation (2.79).

In this way, we obtain quite reliable benchmarks of various American-style options and we collect all of them in the Appendix (see the section 7.1).

3 Monte Carlo Methods for Pricing American-style Options

In this section, we study regression-based Monte Carlo methods to price American-style options. First we formulate the problem, then we study backward dynamic programming principle. After that, we present the source of low bias and high bias. Then we study the Tsitsiklis-Roy method, the Longstaff-Schwartz method and the Andersen-Broadie method. This section is mainly based on Korn [26], Glasserman [15] and Wendel [45].

3.1 Theory Study

3.1.1 Problem Formulation

First we give the exact mathematical definition of American contingent claim and Bermudan contingent claim and their fair prices.

Definition 3.1 (American Contingent Claim). An *American contingent claim* consists of a progressively measurable stochastic process $B = \{(B(t), \mathcal{F}_t)\}_{t\in[0,T]}$ with $B(t) \geq 0$ and a final payment $B(\tau)$ at the exercise date $\tau \in [0, T]$ chosen by the holder of the contingent claim.We assume in addition that τ is a stopping time, that $\{(B(t), F(t))\}_{t\in[0,T]}$ possesses continuous paths, and that

$$\mathbb{E}^{\mathbb{Q}}\left(\sup_{0\leq s\leq T}(B(s))^{\mu}\right) < \infty$$

for some $\mu > 1$.

Theorem 3.2 (Fair Price of American Contingent Claim). The fair price $\hat{p}$ of an American contingent claim B is given by:

$$\hat{p} = \sup_{\tau\in\mathcal{T}[0,T]} \mathbb{E}^{\mathbb{Q}}(e^{-r\tau}B(\tau))$$

where $\mathcal{T}[0, T]$ is the set of all stopping times (adapted to the filtration corresponding to the market model) with values in $[0, T]$ almost surely. There exists a stopping time τ^* such that the supremum will be attained for the hedging strategy π^* corresponding to τ^*.

Algorithm 3.1 Monte-Carlo method to price American / Bermudan options

1: Determine the optimal exercise strategy τ^* for the contingent claim B.
2: Determine the option price $\mathbb{E}^{\mathbb{Q}}(e^{-r\tau^*}B(\tau^*))$.

Definition 3.3 (Bermudan Contingent Claim). Consider time instants $0 \leq t_1 \leq \ldots \leq t_m = T$. A *Bermudan contingent claim* consists of a set of $F(t_i)$ - measurable random variables $B(\tau) \geq 0$ and a final payment $B(\tau)$ at the exercise time $\tau \in \{t_1, \ldots, t_m\}$ chosen by the holder of the option. Here, τ is assumed to be a stopping time and that

$$\mathbb{E}^{\mathbb{Q}}\left(\sup_{s\in\{t_1,\ldots,t_m\}} (B(s))^{\mu}\right) < \infty$$

for some $\mu > 1$.

Theorem 3.4 (Fair Price of Bermudan Contingent Claim). The fair price $\hat{p}$ of a Bermudan contingent claim is given by

$$\hat{p} = \sup_{\tau\in\mathcal{T}\{t_1,\ldots,t_m\}} \mathbb{E}^{\mathbb{Q}}(e^{-r\tau}B(\tau))$$

where $\mathcal{T}\{t_1, \ldots, t_m\}$ is the set of stopping times with values in $\{t_1, \ldots, t_m\}$ and there exists a stopping time τ^* such that the supremum will be attained for the hedging strategy π^* corresponding to τ^*.

Further we denote the stock process $S(t)$ as a Markov process in $\mathbb{R}^d$, denote f as the payoff function, e.g $f = (K - S(t))^+$ for simple American put, denote $B(t_i) = f(S(t_i))$ as the time-t_i value of the payoff if the option holder decides to exercise the option at time t_i, denote $g(S(t_i))$ as the discounted time-t_0 value of the payoff if the option holder decides to exercise the option at time t_i, i.e $g(S(t_i)) = e^{-rt_i}B(t_i) = e^{-rt_i}f(S(t_i))$.

Notice that the fair price of the simple American put is achieved by using an optimal stopping time τ^* which has the form:

$$\tau^* \quad = \quad \inf\{t \geq 0 : S(t) \leq b^*(t)\} \tag{3.1}$$

for some *optimal exercise boundary* $b^*(t)$, which is shown in Figure 3.1: Thus, when we price American or Bermudan options, we need to know whether we should exercise the option or not at each potential exercise opportunity $\{t_1, \ldots, t_m\}$[1], this can only be done if we know the optimal exercise strategy τ^* in the equation (3.1) in advance, which leads to the Monte-Carlo framework to price American / Bermudan options, see Algorithm 3.1.

[1] for American option, $m \longrightarrow \infty$

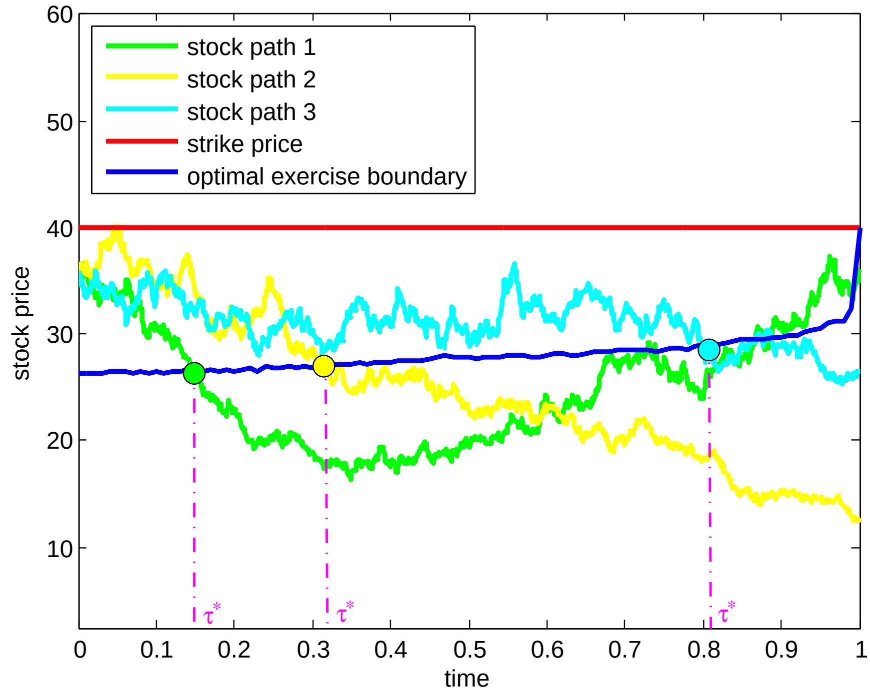


Figure 3.1: Optimal exercise boundary for simple American put option with payoff $(K - S(t))^+$. The green, yellow and cyan curve are three simulated paths for the stock price, the red line is the strike, the blue curve is the calculated optimal exercise boundary $b^*(t)$. The option is optimally exercised at time τ^*, the first time the stock price reaches the optimal exercise boundary.

3.1.2 Backward Dynamic Programming Principle

In the following we only talk "American" to include both American option and Bermduan option, since we can let the number of potential exercise dates of Bermudan option "m" increase to infinity, such that continuous exercise dates for American option can be approximated by a finite set of exercise dates of Bermudan option. This approximation will be examined in the section 3.2.1.

The basic idea of backward dynamic programming principle is: starting at the maturity where the exercise decision is known, one computes one by one time step backwards until the initial time is reached and updates the optimal exercise decision. There are two ways to present this principle to price American options, one is based on time-t_i value, the other is based on discounted time-t_0 value. We study both of them and then show the equality of each other.

Denote $\tilde{V}(S(t_i))$ as the time-t_i value for American option at time t_i, assuming that the option has not been exercised before t_i, where $i \in \{1, 2, \ldots, m\}$, $t_i = \frac{i}{m}T$. $\tilde{V}(S(t_i))$ can be interpreted as the value of a at time t_i newly issued option starting from state $S(t_i)$ and ending at maturity T. The value process $\tilde{V} = \left(\tilde{V}(S(t_i))\right)_{i=1,\dots,m}$ of the American option with the payoff function f satisfies the following time-t_i based backward dynamic programming principle:

$$\tilde{V}(S(t_m)) = f(S(t_m)) \tag{3.2}$$
$$\tilde{V}(S(t_i)) = \max\left(f(S(t_i)), \mathbb{E}[e^{-r(t_{i+1}-t_i)}\tilde{V}(S(t_{i+1}))|S(t_i)]\right) \tag{3.3}$$

for $i = m-1, m-2, \dots, 1$. And the fair price of the American option, which we are interested in, is $\tilde{V}(S(t_0)) = e^{-rt_1}\tilde{V}(S(t_1))$. Notice in the equation (3.3), at the exercise date t_i, the option value is the maximum of the immediate exercise value $f(S(t_i))$ and the expected time$-t_i$ value of continuing the option $\mathbb{E}[e^{-r(t_{i+1}-t_i)}\tilde{V}(S(t_{i+1}))|S(t_i)]$, which is main difficulty in pricing American options by simulation. We also notice that we use the equality $\mathbb{E}[e^{-r(t_{i+1}-t_i)}\tilde{V}(S(t_{i+1}))|\mathcal{F}_{t_i}] = \mathbb{E}[e^{-rt_{(t_{i+1}-t_i)}}\tilde{V}(S(t_{i+1}))|S(t_i)]$ as we assume $S(t)$ to be a Markov process.

Denote $V(S(t_i))$ as the discounted time-t_0 value for American option at time t_i, assuming the option has not been exercised before t_i. The value process $V = (V(S(t_i)))_{i=1,\dots,m}$ of the American option with the discounted payoff function $g = e^{-rt_i}f$ satisfies the following time-t_0 based backward dynamic programming principle:

$$V(S(t_m)) = g(S(t_m)) \tag{3.4}$$
$$V(S(t_i)) = \max\left(g(S(t_i)), \mathbb{E}[V(S(t_{i+1}))|S(t_i)]\right) \tag{3.5}$$

for $i = m-1, m-2, \dots, 1$. The fair price of the American option, which we are interested in, is $V(S(t_0)) = V(S(t_1))$.

This time-t_0 based principle is the same as the previous time-t_i based principle. We proof this as follows. For $i = 1, 2, \dots, m$, we have:

$$\begin{aligned} g(S(t_i)) &= e^{-rt_i}f(S(t_i)) \\ V(S(t_i)) &= e^{-rt_i}\tilde{V}(S(t_i)) \end{aligned} \tag{3.6}$$

Thus we can prove for $i = 0$ and $i = m$:

$$V(S(t_0)) = \tilde{V}(S(t_0)) = e^{-rt_1}\tilde{V}(S(t_1)) = V(S(t_1))$$
$$V(S(t_m)) = e^{-rt_m}V(S(t_m)) = e^{-rt_m}f(S(t_m)) = g(S(t_m))$$

Further, for $i = m-1, m-2, \dots, 1$, $V(S(t_i))$ satisfies:

$$\begin{aligned} V(S(t_i)) &= e^{-rt_i}\tilde{V}(S(t_i)) \\ &\overset{\text{eq.(3.3)}}{=} e^{-rt_i}\max\left(f(S(t_i)), \mathbb{E}[e^{-r(t_{i+1}-t_i)}\tilde{V}(S(t_{i+1}))|S(t_i)]\right) \\ &= \max\left(e^{-rt_i}f(S(t_i)), \mathbb{E}[e^{-rt_i}e^{-r(t_{i+1}-t_i)}\tilde{V}(S(t_{i+1}))|S(t_i)]\right) \\ &= \max\left(g(S(t_i)), \mathbb{E}[e^{-rt_{i+1}}\tilde{V}(S(t_{i+1}))|S(t_i)]\right) \\ &\overset{\text{eq.(3.6)}}{=} \max\left(g(S(t_i)), \mathbb{E}[V(S(t_{i+1}))|S(t_i)]\right) \end{aligned}$$

□

At each potential exercise date, the American option holder must decide whether to exercise the option or to hold the option. The value of holding the option is called the *continuation value*, which is defined in terms of time-t_0 value as follows:

$$C(S(t_i)) = \mathbb{E}[V(S(t_{i+1}))|S(t_i)], \quad i = 1, 2, \dots, m-1 \tag{3.7}$$

Thus the time-t_0 value based backward dynamic programming principle (3.4)-(3.5) can be rewritten as:

$$C(S(t_m)) = 0 \tag{3.8}$$
$$C(S(t_i)) = \mathbb{E}[\max(g(S(t_{i+1})), C(S(t_{i+1})))|S(t_i)] \tag{3.9}$$

for $i = m-1, m-2, \dots, 0$. The fair price of the American option, which we are interested in, is $C(S(t_0))$.

From the equation (3.5) and the equation (3.7), we notice the fact that the discounted process $V(S(t_i))$ determine the continuation value by:

$$V(S(t_i)) = \max(g(S(t_i)), C(S(t_i)))$$

for $i = 1, 2, \dots, m$.

The backward dynamic principle (3.2)-(3.3), (3.4)-(3.5) and (3.8)-(3.9) respectively focus on time-t_i option value, time-t_0 option value and time-t_0 continuation value. It can also be rewritten in terms of stopping rules and optimal exercise region as follows:

$$\tau^*(m) = t_m \tag{3.10}$$
$$\tau^*(i) = \begin{cases} t_i, & g(S(t_i)) \geq \mathbb{E}[g(S(\tau^*(i+1)))|S(t_i)] \\ \tau^*(i+1), & g(S(t_i)) < \mathbb{E}[g(S(\tau^*(i+1)))|S(t_i)] \end{cases} \tag{3.11}$$

for $i = m-1, m-2, \dots, 1$.

The region for exercising the option optimally at each potential exercise time t_i is the set:

$$\{S(t_i) : g(S(t_i)) \geq \mathbb{E}[g(S(\tau^*(i+1)))|S(t_i)]\}$$

while the region for holding the option optimally is the set:

$$\{S(t_i) : g(S(t_i)) < \mathbb{E}[g(S(\tau^*(i+1)))|S(t_i)]\}$$

The stopping rule τ^* can be understood as the first time the stock price $S(t_i)$ enters the optimal exercise region.

3.1.3 Longstaff-Schwartz Method and Tsitsiklis-Roy Method

In this section, we study two regression-based Monte Carlo Methods to price American-style options, which are the Longstaff-Schwartz Method [33] and the Tsitsiklis-Roy Method [43]. Both of them make use of the backward dynamic programming principle presented in the previous section. At each potential exercise date, we have to decide whether to exercise or to hold the option. The discounted time-t_0 value of exercising the option $g(S(t_i))$ in equation (3.5) can be calculated easily while the discounted time-t_0 value of holding the option $C(S(t_i)) = \mathbb{E}[V(S(t_{i+1}))|S(t_i)]$ in equation (3.7) is difficult to compute, because they are nested conditional expectations. Both methods use an approach of a least-square linear regression for selected simulated paths to compute the nested conditional expectations.

Since the stock price $S(t_i)$ is assumed to be a Markov process, we have the following relations:

$$\mathbb{E}[V(S(t_{i+1}))|S(t_i)] \quad = \quad u(S(t_i))$$

for some measurable function u. We set up a regression model and approximate this conditional expectation by minimizing the sum of the squares of errors for selected paths $n = 1, 2, \ldots, \hat{N}$ among all simulated paths $n = 1, 2, \ldots, N$ with $\hat{N} \leq N$:

$$\begin{aligned} & \min_{u \in U} \mathbb{E}\left[\mathbb{E}[V(S(t_{i+1}))|S(t_i)] - u(S(t_i))\right]^2 \\ \approx \quad & \min_{u \in U} \frac{1}{\hat{N}} \sum_{n=1}^{\hat{N}} \left[V(S^{(n)}(t_{i+1})) - u(S^{(n)}(t_i))\right]^2 \end{aligned} \tag{3.12}$$

where U is a parametric family of functions u. We specify the function space U by a linear combination of basis functions:

$$\begin{aligned} U: \quad & \mathbb{R}^d \to \mathbb{R} \\ u: \quad & x \mapsto u(x) = \sum_{l=1}^{k} a_l H_l(x) \quad \text{with} \quad a_i \in \mathbb{R} \end{aligned}$$

Popular choices for U are monomial polynomials, Laguerre polynomials, Legendre polynomials, Hermite polynomials and Chebyshev polynomials. We will test the performance of these basis functions in the section 3.2.5. The simplest choice is monomials with the form of $H_l(x) = x^{l-1}$. Note that although the form of $u(x) = \sum_{l=1}^{k} a_l H_l(x)$ is nonlinear in the input x, it is linear in the coefficients a_l, thus the equation (3.12) is indeed a least-squares linear regression problem, which can be solved explicitly.

$$\begin{aligned}
&\min_{u\in U}\mathbb{E}\left[\mathbb{E}[V(S(t_{i+1}))|S(t_i)]-u(S(t_i))\right]^2 \\
=\;&\min_{a\in\mathbb{R}^k}\frac{1}{\hat{N}}\sum_{n=1}^{\hat{N}}\left[V(S^{(n)}(t_{i+1}))-\sum_{l=1}^{k}a_l H_l(S^{(n)}(t_i))\right]^2 \qquad (3.13)
\end{aligned}$$

where $a=[a_1,a_2,\ldots,a_k]^\intercal\in\mathbb{R}^k$. The solution of (3.13) is the optimal coefficient $a^*=[a_1^*,a_2^*,\ldots,a_k^*]^\intercal$ by:

$$\begin{aligned}
a^* &:= [a_1^*,a_2^*,\ldots,a_k^*]^\intercal \\
&= (X^\intercal X)^{-1}X^\intercal Y\in\mathbb{R}^{k\times 1} \qquad (3.14)
\end{aligned}$$

with $Y:=[V(S^{(1)}(t_i)),\ldots,V(S^{(\hat{N})}(t_i))]^\intercal\in\mathbb{R}^{\hat{N}\times 1}$ and

$$X:=\begin{pmatrix}H^\intercal(S^{(1)}(t_i))\\ \vdots\\ H^\intercal(S^{(\hat{N})}(t_i))\end{pmatrix}=\begin{pmatrix}H_1(S^{(1)}(t_i)) & \ldots & H_k(S^{(1)}(t_i))\\ \vdots & \ldots & \vdots\\ H_1(S^{(\hat{N})}(t_i)) & \ldots & H_k(S^{(\hat{N})}(t_i))\end{pmatrix}\in\mathbb{R}^{\hat{N}\times k}$$

The solution of this regression problem generates an estimate $C^*(S(t_i))$ for the continuation value $C(S(t_i))=\mathbb{E}[V(S(t_{i+1}))|S(t_i)]$ by:

$$\begin{aligned}
C(S(t_i)) &\approx C^*(S(t_i)) \\
&= \sum_{l=1}^{k}a_l^* H_l(S(t_i)) \qquad (3.15)
\end{aligned}$$

In the paper of Longstaff and Schwartz [33], they choose the subset of all paths $\Theta_{\hat{N}}\subset\{1,\ldots,N\}$ for which the option is in-the-money, i.e. $f(S^{(n)}(t_m))>0$ holds for $n\in\Theta_{\hat{N}}$, to do regression. Of course we can also choose all paths to do regression. We compare the performance of these two strategies in the section 3.2.3.

If the estimate of continuation value $C^*(S^n(t_i))$ is bigger than the discounted exercising value $g(S^n(t_i))$ for some certain path $n\in\Theta_{\hat{N}}$ at some certain time step t_i, we should hold the option; otherwise we should exercise it. At this step, the Longstaff-Schwartz method and the Tsitsiklis-Roy method differ here when updating the option value backwards:

- Longstaff-Schwartz

$$V(S^{(n)}(t_i))=\begin{cases}g(S^{(n)}(t_i)), & \text{if } n\in\Theta_{\hat{N}} \text{ and } g(S^{(n)}(t_i))\geq C^*(S^{(n)}(t_i))\\ V(S^{(n)}(t_{i+1})), & \text{otherwise}\end{cases}$$

- Tsitsiklis-Roy

$$V(S^{(n)}(t_i)) = \begin{cases} g(S^{(n)}(t_i)), & g(S^{(n)}(t_i)) \geq C^*(S^{(n)}(t_i)) \\ C^*(S^{(n)}(t_i)), & \text{otherwise} \end{cases}$$

Besides this difference, in the paper of Tsitiklis and Roy [43], they use all paths to do regression instead of in-the-money paths as mentioned in the paper of Longstaff and Schwartz. Now we can summarize the Longstaff-Schwartz algorithm 3.2 and Tsitsiklis-Roy algorithm 3.3 here to price American-style options. Our numerical experiment in the section 3.2.4 will show that the Longstaff-Schwartz method performs better than the Tsitsiklis-Roy method.

Algorithm 3.2 Longstaff-Schwartz method

1. Generate N independent paths for stock at all possible exercise dates: $\{S^{(n)}(t_1), S^{(n)}(t_2), \dots, S^{(n)}(t_m)\}$ with $n = 1, \dots, N$, $t_i = \frac{T}{m} \times i$, $i = 1, \dots, m$.

2. At maturity $t_m = T$, fix the discounted terminal values of the American option for each path $n = 1, \dots, N$: $V(S^{(n)}(t_m)) = g(S^{(n)}(t_m))$.

3. Compute backward at each potential exercise date t_i for $i = m-1, \dots, 1$:
   1) Choose k basis functions: $\{H_1, \dots, H_k\}$.
   2) Consider the subset of paths $\Theta_{\hat{N}} \subset \{1, \dots, N\}$ for which the option is in-the-money, i.e. $g(S^{(n)}(t_i)) > 0$ holds for $n \in \Theta_{\hat{N}}$.
   3) Solve the least-squares linear regression problem:

   $$\min_{a_l \in \mathbb{R}} \frac{1}{\hat{N}} \sum_{n=1}^{\hat{N}} (V(S^{(n)}(t_i)) - \sum_{l=1}^{k} a_l H_l(S^{(n)}(t_i)))^2$$

   and obtain the optimal coefficient a^*:

   $$a^* \;:=\; [a_1^*, \dots, a_k^*]^\intercal = (X^\intercal X)^{-1} X^\intercal Y \in \mathbb{R}^{k \times 1}$$

   with $Y := [V(S^{(1)}(t_i)), \dots, V(S^{(\hat{N})}(t_i))]^\intercal \in \mathbb{R}^{\hat{N} \times 1}$ and

   $$X := \begin{pmatrix} H_1(S^{(1)}(t_i)) & \dots & H_k(S^{(1)}(t_i)) \\ \vdots & \dots & \vdots \\ H_1(S^{(\hat{N})}(t_i)) & \dots & H_k(S^{(\hat{N})}(t_i)) \end{pmatrix} \in \mathbb{R}^{\hat{N} \times k}$$

   4) Calculate the estimated continuation value $C^*(S^{(n)}(t_i))$ and the discounted exercising value $g(S^{(n)}(t_i))$ for each path $n \in \Theta_{\hat{N}}$:

   $$C^*(S^{(n)}(t_i)) \;=\; \sum_{l=1}^{k} a_l^* H_l(S^{(n)}(t_i))$$

   5) Compare $C^*(S^{(n)}(t_i))$ and $g(S^{(n)}(t_i))$ to decide whether to exercise or to continue the option:

   $$V(S^{(n)}(t_i)) = \begin{cases} g(S^{(n)}(t_i)), & \text{if } n \in \Theta_{\hat{N}} \text{ and } g(S^{(n)}(t_i)) \geq C^*(S^{(n)}(t_i)) \\ V(S^{(n)}(t_{i+1})), & \text{otherwise} \end{cases}$$

4. Compute $V_k^N(S(t_0)) = \left(\frac{1}{N} \sum_{n=1}^{N} V(S^{(n)}(t_1))\right)$ as the American option price.

Algorithm 3.3 Tsitsiklis-Roy method

1. Generate N independent paths for stock at all possible exercise dates: $\{S^{(n)}(t_1), S^{(n)}(t_2), \ldots, S^{(n)}(t_m)\}$ with $n = 1, \ldots, N$, $t_i = \frac{T}{m} \times i$, $i = 1, \ldots, m$.
2. At maturity $t_m = T$, fix the discounted terminal values of the American option for each path $n = 1, \ldots, N$: $V(S^{(n)}(t_m)) = g(S^n(t_m))$.
3. Compute backward at each potential exercise date t_i for $i = m-1, \ldots, 1$:
   1) Choose k basis functions: $\{H_1, \ldots, H_k\}$.
   2) Consider all paths $\Theta_{\hat{N}} = \{1, \ldots, N\}$ for which the option is either in-the-money, or out-of-the-money or at-the-money, i.e. $g(S^{(n)}(t_i)) \lesseqqgtr 0$ holds for $n \in \Theta_{\hat{N}}$.
   3) Solve the least-squares linear regression problem:
   $$\min_{a_l \in \mathbb{R}} \frac{1}{\hat{N}} \sum_{n=1}^{\hat{N}} (V(S^{(n)}(t_i)) - \sum_{l=1}^{k} a_l H_l(S^{(n)}(t_i)))^2$$
   and obtain the optimal coefficient a^*:
   $$a^* \quad := \quad [a_1^*, \ldots, a_k^*]^\intercal = (X^\intercal X)^{-1} X^\intercal Y \in \mathbb{R}^{k \times 1}$$
   with $Y := [V(S^{(1)}(t_i)), \ldots, V(S^{(\hat{N})}(t_i))]^\intercal \in \mathbb{R}^{\hat{N} \times 1}$ and
   $$X := \begin{pmatrix} H_1(S^{(1)}(t_i)) & \ldots & H_k(S^{(1)}(t_i)) \\ \vdots & \ldots & \vdots \\ H_1(S^{(\hat{N})}(t_i)) & \ldots & H_k(S^{(\hat{N})}(t_i)) \end{pmatrix} \in \mathbb{R}^{\hat{N} \times k}$$
   4) Calculate the estimated continuation value $C^*(S^{(n)}(t_i))$ and the discounted exercising value $g(S^{(n)}(t_i))$ for each path $n \in \Theta_{\hat{N}}$:
   $$C^*(S^{(n)}(t_i)) \quad = \quad \sum_{l=1}^{k} a_l^* H_l(S^{(n)}(t_i))$$
   5) Compare $C^*(S^{(n)}(t_i))$ and $g(S^{(n)}(t_i))$ to decide whether to exercise or to continue the option:
   $$V(S^{(n)}(t_i)) = \begin{cases} g(S^{(n)}(t_i)), & g(S^{(n)}(t_i)) \geq C^*(S^{(n)}(t_i)) \\ C^*(S^{(n)}(t_i)), & \text{otherwise} \end{cases}$$
4. Compute $V_k^N(S(t_0)) = \left(\frac{1}{N} \sum_{n=1}^{N} V(S^{(n)}(t_1)) \right)$ as the American option price.

3.1.4 Convergence Properties

Notice that the value $V_k^N(S(t_0))$ computed by the Longstaff-Schwartz method or the Tsitsiklis-Roy method is only an approximation of the true value $V(S(t_0))$ of an American-style option. Since $V_k^N(S(t_0))$ depends on the number of simulated paths N and the number and form of basis functions k, there are two sources of errors leading to the difference between $V_k^N(S(t_0))$ and $V(S(t_0))$.

1. The first error by using the Monte Carlo simulation, since the value of option is an expectation, estimated by an arithmetic mean. This error can be decreased by increasing the number of simulated paths N.

2. The second error by using a certain set of basis functions $\{H_1, H_2, \ldots, H_k\}$ to estimate of the continuation value $C(S(t_i)) = \mathbb{E}[V(S(t_{i+1}))|S(t_i)]$. This error can be decreased by choosing proper number and form of basis functions according to the specific payoff form of the option to make the projection better.

Clément, Lamberton and Protter [12] have proved the convergence properties of the Longstaff-Schwartz method in the following way. They define one term $V_k(S(t_0))$ by:

$$V_k(S(t_0)) = \sup_{\tau \in \Gamma(H_1,\ldots,H_k)} \mathbb{E}(e^{-r\tau} g(S(\tau))) \tag{3.16}$$

where τ denotes as a stopping time contained in the set of all stopping times (exercise strategies) $\Gamma(H_1, \ldots, H_k)$ based on solving the regression problem by using the basis functions $\{H_1, H_2, \ldots, H_k\}$. Then they prove the following properties:

- When the number of simulated paths N increases to infinity while fixing the number of basis functions k, the option price computed by the Longstaff-Schwartz method $V_k^N(S(t_0))$ converges to the supremum of all option prices based on the whole functional space spanned by the same basis functions $V_k(S(t_0))$.

  $$V_k^N(S(t_0)) \xrightarrow{N \longrightarrow \infty} V_k(S(t_0)) \quad \text{almost surely}$$

  if the sequence of basis functions is total in a suitable L^2- function space.

- When the number of basis functions k increases to infinity, the supremum of all option prices computed based on the whole functional space spanned by the same basis functions $V_k(S(t_0))$ converges to the actual option price $V(S(t_0))$.

  $$V_k(S(t_0)) \xrightarrow{k \longrightarrow \infty} V(S(t_0))$$

Remark 3.5. 1. If the number of basis functions k is fixed, the option price computed by the Longstaff-Schwartz method $V_k^N(S(t_0))$ only converges to the solution of the optimal stopping problem $V_k(S(t_0))$ (equation (3.16)) by increasing the number of simulated paths N, rather to the true option price $V(S(t_0))$, which means that the Longstaff-Schwartz method uses a suboptimal exercise strategy and underestimate the option price.

2. However, they didn't prove similar convergence result for the number of basis functions k, thus it's hard to say how many basis functions should be used. Glasserman and Yu [16] points out that in many examples we might need exponential growth in the number of simulated path N when the number of basis functions k increases.

3.1.5 Source of Bias

All simulation methods to price American-style options may contain two sources of bias, one is high bias and the other is low bias. Some methods only give high bias, some only show low bias and others may have a mixture of these two sources of bias.

- **High bias** results from applying the backward programming principle and using the same information to decide whether to exercise the option as to estimate the continuation value, however in real life future information is not available.
- **Low bias** comes from using a suboptimal exercising strategy to price American options, while the true fair value of an American option is computed by using an optimal stopping strategy.

Glasserman [15] points out that the Longstaff-Schwartz method and the Tsitsiklis-Roy method both mix high bias and low bias. In order to ensure an estimator only with low bias, we have to add another step at the end of both algorithms by resampling new out-of-sample independent paths and use the calculated optimal coefficient a^* to determine the new continuation value, see Algorithm 3.4 and Algorithm 3.5.

Algorithm 3.4 Modified Longstaff-Schwartz method with low bias

- Step 1 - Step 3.5: Same as in the Longstaff-Schwartz method (Algorithm 3.2) and save the computed optimal coefficient a^*.

- Step 4: Regenerate N_{new} new independent paths for stock at all potential exercise dates: $\{S^{(n)}(t_1), S^{(n)}(t_2), \dots, S^{(n)}(t_n)\}$ with $n = 1, \dots, N_{\text{new}}$.

- Step 5: Define the stopping rule $\tau^{(n)} = t_1$ for each path n and compute forward at t_i for $i = 1, \dots, m$:
  1) Choose same basis functions as before: $\{H_1, \dots, H_k\}$.
  2) Calculate the estimated continuation value $C^*(S^{(n)}(t_i))$ and the discounted exercising value $g(S^{(n)}(t_i))$ for each path $n \in N_{\text{new}}$: $C^*(S^{(n)}(t_i)) = \sum_{l=1}^{k} a_l^* H_l(S^{(n)}(t_i))$
  3) If $\tau^{(n)} = t_1$ and $g(S^{(n)}(t_i)) > 0$ and $g(S^{(n)}(t_i)) > C^*(S^{(n)}(t_i))$: exercise the option at t_i, set $\tau^{(n)} = t_i$ and $V_{\text{new}}(S^{(n)}(t_1)) = g(S^{(n)}(t_i))$, stop;
     Else if $t_i < t_{m-1}$: continue the option at t_i;
     Else: exercise the option at t_m and set $\tau^{(n)} = t_m$ and $V_{\text{new}}(S^{(n)}(t_1)) = g(S^{(n)}(t_m))$, stop.

- Step 6: Compute $V_k^{N_{\text{new}}}(S(t_0)) = \left(\frac{1}{N_{\text{new}}} \sum_{n=1}^{N_{\text{new}}} V_{\text{new}}(S^{(n)}(t_1)) \right)$ as the American option price.

Algorithm 3.5 Modified Tsitsiklis-Roy method with low bias

- Step 1 - Step 3.5: Same as in the Tsitsiklis-Roy method (Algorithm 3.3).
- Step 4: Regenerate N_{new} new independent paths for stock at all potential exercise dates: $\{S^{(n)}(t_1), S^{(n)}(t_2), \ldots, S^{(n)}(t_n)\}$ with $n = 1, \ldots, N_{\text{new}}$.
- Step 5: Define the stopping rule $\tau^{(n)} = t_1$ for each path n and compute forward at t_i for $i = 1, \ldots, m$:
  1) Choose same basis functions as before: $\{H_1, \ldots, H_k\}$.
  2) Calculate the estimated continuation value $C^*(S^{(n)}(t_i))$ and the discounted exercising value $g(S^{(n)}(t_i))$ for each path $n \in N_{\text{new}}$: $C^*(S^{(n)}(t_i)) = \sum_{l=1}^{k} a_l^* H_l(S^{(n)}(t_i))$
  3) If $\tau^{(n)} = t_1$ and $g(S^{(n)}(t_i)) > 0$ and $g(S^{(n)}(t_i)) > C^*(S^{(n)}(t_i))$: exercise the option at t_i, set $\tau^{(n)} = t_i$ and $V_{\text{new}}(S^{(n)}(t_1)) = g(S^{(n)}(t_i))$, stop;
     Else if $t_i < t_{m-1}$: continue the option at t_i;
     Else: exercise the option at t_m and set $\tau^{(n)} = t_m$ and $V_{\text{new}}(S^{(n)}(t_1)) = g(S^{(n)}(t_m))$, stop.
- Step 6: Compute $V_k^{N_{\text{new}}}(S(t_0)) = \left(\frac{1}{N_{\text{new}}} \sum_{n=1}^{N_{\text{new}}} V_{\text{new}}(S^{(n)}(t_1)) \right)$ as the American option price.

3.1.6 Snell Envelope and Doob-Meyer Decomposition

In this section, we study the powerful concepts of *Snell envelope* and *Doob-Meyer* Decomposition, which are used in the next section 3.1.7 and show that the discounted value process $V(S(t_i))$ is the Snell envelope of the discounted payoff process $g(S(t_i))$ in the equation (3.5). This section is based on Wendel [45].

Definition 3.6 (Snell Envelope). Let $Z = (Z(t_i))_{i=0,\ldots,m}$ with filtration $(\mathcal{F}_{t_i})_{i=0,\ldots,m}$ be an adapted process and $\mathbb{E}[\max_{i=0,\ldots,m} Z(t_i)] < \infty$, we define $U = (U(t_i))_{i=0,\ldots,m}$ as follows:

$$\begin{aligned} U(t_m) &= Z(t_m) \\ U(t_i) &= \max\left(Z(t_m), \mathbb{E}[U(t_{i+1})|\mathcal{F}(t_i)]\right), i = m-1, \ldots, 0 \end{aligned}$$

We call U the Snell envelope of Z.

Theorem 3.7. Let $Z = (Z(t_i))_{i=0,\ldots,m}$ with filtration $(\mathcal{F}_{t_i})_{i=0,\ldots,m}$ be an adapted process and $\mathbb{E}[\max_{i=0,\ldots,m} Z(t_i)] < \infty$, then the Snell envelope $U = (U(t_i))_{i=0,\ldots,m}$ of Z is the smallest supermartingale dominating Z.

Proof:
From the definition of U, we have:

$$U(t_i) \geq \mathbb{E}(U(t_i)|\mathcal{F}(t_i)), i = 0, \ldots, m$$

which shows directly that U is a supermartingale.

Define $(\tilde{U}(t_i))_{i=0,\ldots,m}$ be another supermartingale dominating Z, we have:

$$\tilde{U}(t_m) \geq Z(t_m) = U(t_m)$$

Assume that $\tilde{U}(t_i) \geq U(t_i)$, then:

$$\begin{aligned} \tilde{U}(t_{i-1}) &\geq \mathbb{E}[\tilde{U}(t_i)|\mathcal{F}_{t_{i-1}}] \\ &\geq \mathbb{E}[U(t_i)|\mathcal{F}_{t_{i-1}}] \end{aligned}$$

On the other hand, according to the definition of $\tilde{U}$, we have:

$$\tilde{U}(t_{i-1}) \geq Z(t_{i-1})$$

Together with the previous equation, we have:

$$\begin{aligned} \tilde{U}(t_{i-1}) &\geq \max(Z(t_{i-1}), \mathbb{E}[U(t_i)|\mathcal{F}_{t_{i-1}}]) \\ &= U(t_{i-1}) \end{aligned}$$

□

Theorem 3.8 (Doob-Meyer Decomposition). Denote $U = (U(t_i))_{i=0,\dots,m}$ as a supermartingale, then there exists a unique decomposition, which is called Doob-Meyer Decomposition:

$$U(t_i) \quad = \quad U(t_0) + M(t_i) + A(t_i), i = 0, \dots, m$$

where $M = (M(t_i))_{i=0,\dots,m}$ is a martingale with $M(t_0) = 0$ and $A = A(t_i)_{i=0,\dots,m}$ is a predictable nonincreasing process with $A(t_0) = 0$.

Proof:
We define A by recursion as follows:

$$\begin{aligned} A(t_0) &= 0 \\ A(t_i) - A(t_{i-1}) &= \mathbb{E}[U(t_i) - U(t_{i-1})|\mathcal{F}_{t_{i-1}}], i = 1, \dots, m \end{aligned} \tag{3.17}$$

Since U is a supermartingale, $\mathbb{E}[U(t_i) - U(t_{i-1})|\mathcal{F}_{t_{i-1}}] \le 0$, $A(t_i) - A(t_{i-1}) \le 0$, thus A is predictable and nonincreasing.

We further define M as follows:

$$\begin{aligned} M(t_0) &= 0 \\ M(t_i) &= U(t_i) - U(t_0) - A(t_i), i = 1, \dots, m \end{aligned}$$

We have:

$$\begin{aligned} \mathbb{E}[M(t_i) - M(t_{i-1})|\mathcal{F}_{t_{i-1}}] &= \mathbb{E}[U(t_i) - U(t_{i-1}) - (A(t_i) - A(t_{i-1}))|\mathcal{F}_{t_{i-1}}] \\ &= \mathbb{E}[U(t_i) - U(t_{i-1})|\mathcal{F}_{t_{i-1}}] - (A(t_i) - A(t_{i-1})) \\ &\overset{eq.(3.17)}{=} 0 \end{aligned}$$

Thus M is a martingale with $M(t_0) = 0$. Any process A satisfying the required properties must satisfy equation (3.17), thus we can prove the uniqueness of the decomposition.

□

3.1.7 Dual Upper Bound and Andersen-Broadie Method

From the section 3.1.5, we know that the modified Longstaff-Schwartz method (Algorithm 3.4) and the modified Tsitsiklis-Roy method (Algorithm 3.5) give only low bias, although the Longstaff-Schwartz method (Algorithm 3.2) and the Tsitsiklis-Roy method (Algorithm 3.3) contain both high and low biases. But we also have to assess the lower estimates by the modified algorithms.

Since we need to know how lower than the true option price the result is, we need an upper bound to pair the lower bound, in order to judge the quality of the modified algorithms. If the difference between the upper bound and the lower bound is small, we can conclude that the algorithm can price the option price accurately; otherwise, we have to consider choosing other forms or numbers of basis functions to estimate the continuation value of the option.

Rogers [38], Haugh and Kogan [18], Andersen and Broadie [2], Belomestny, Bender and Schoenmakers [4] have looked at the dual optimization problem of the optimal stopping problem and proposed different procedures to give upper bounds to pair the lower bounds. In this section, we focus only on the Andersen-Broadie method [2].

Here we show the time-t_0 based backward dynamic programming principle (eq. (3.4) and (3.5)) again:

$$\begin{aligned} V(S(t_m)) &= g(S(t_m)) \\ V(S(t_i)) &= \max\left(g(S(t_i)), \mathbb{E}[V(S(t_{i+1}))|S(t_i)]\right), \quad i = m-1, 1 \end{aligned}$$

We notice that at time t_i the decision of not exercising the option may not be the optimal exercise strategy at this moment, in other words, the option value $V(S(t_i))$ assuming not exercised at time $t_1, \ldots, t_{i-1}$ may be bigger than the option value $\mathbb{E}[V(S(t_{i+1}))|S(t_i)]$ assuming not exercised at time $t_1, \ldots, t_i$, i.e:

$$V(S(t_i)) \geq \mathbb{E}[V(S(t_{i+1}))|S(t_i)]$$

which means the discounted option value process $V(S(t_i))$ is a supermartingale. Besides of this, we also have:

$$V(S(t_i)) \geq g(S(t_i)), \quad i = 1, \ldots, m \tag{3.18}$$

From the theorem 3.7, we know that $V(S(t_i))$ is the Snell envelope of the discounted payoff process $g(S(t_i))$, which is the smallest supermartingale dominating $g(S(t_i))$. However the Snell envelope is difficult to compute. In order to compute the upper bound of the option price, we have to find another supermartingale dominating $g(S(t_i))$.

Before we give another computable supermartingale, we first define $M = (M(t_i))_{i=0,\ldots,m}$ be a discrete martingale with $M(t_0) = 0$ where $t_1, \ldots, t_m$ are potential exercise dates of the American option. Then for any stopping time τ taking values in $\{t_1, \ldots, t_m\}$, we have:

$$\begin{aligned} \mathbb{E}[g(S(\tau))] &= \mathbb{E}[g(S(\tau)) - M(\tau)] \\ &\leq \mathbb{E}[\max_{i=1,\ldots,m}(g(S(t_i)) - M(t_i))] \end{aligned} \tag{3.19}$$

Since the equation (3.19) holds for all martingales M with $M(t_0) = 0$ and for all stopping times τ, it follows:

$$\begin{aligned} V(S(t_0)) &= \sup_{\tau} \mathbb{E}[g(S(\tau))] \\ &\leq \inf_{M} \mathbb{E}[\max_{i=1,\ldots,m}(g(S(t_i)) - M(t_i))] \end{aligned} \tag{3.20}$$

Thus the right side of the inequality (3.20) provides the upper bound for the left side of the equation $V(S(t_0))$, which is true option price. Further more, the inequality (3.20) is indeed an equality, which is the key part of duality.

Theorem 3.9. With V, g and M defined as before, we have

$$V(S(t_0)) = \inf_M \mathbb{E}[\max_{i=1,\dots,m}(g(S(t_i)) - M(t_i))]$$

Particularly, the infimum is obtained by the martingale component M_{Doob} of the Doob-Meyer decomposition of V.

Proof:
Since $V(S(t_i))$ is a supermartingale dominating $g(S(t_i))$, we can apply the theorem of Doob-Meyer Decomposition 3.8:

$$V(S(t_i)) = V(S(t_0)) + M_{\text{Doob}}(t_i) + A(t_i)$$

where $M_{\text{Doob}}(t_i)$ is a martingale with $M_{\text{Doob}}(t_0) = 0$ and $A(t_i)$ is a predictable nonincreasing process with $A(t_0) = 0$. Thus:

$$M_{\text{Doob}}(t_i) = V(S(t_i)) - V(S(t_0)) - A(t_i)$$

Hence:

$$\begin{aligned}
\mathbb{E}[\max_{i=1,\dots,m}(g(S(t_i)) - M_{\text{Doob}}(t_i))] &= \mathbb{E}[\max_{i=1,\dots,m}(g(S(t_i)) - (V(S(t_i)) - V(S(t_0)) - A(t_i)))] \\
&= \mathbb{E}[\max_{i=1,\dots,m}(\{g(S(t_i)) - V(S(t_i))\} + V(S(t_0)) + A(t_i))] \\
&\overset{eq.(3.18)}{\leq} \mathbb{E}[\max_{i=1,\dots,m}(V(S(t_0)) + A(t_i))] \\
&\leq \mathbb{E}[V(S(t_0)) + A(t_0)] \\
&= \mathbb{E}[V(S(t_0))]
\end{aligned}$$

□

According to the theorem 3.9, an upper bound can be constructed via duality by taking the martingale component of a supermartingale as follows: suppose τ is a good approximation of the optimal exercise strategy τ^*, e.g by the modified Longstaff-Schwartz method (Algorithm 3.4) or the modified Tsitsiklis-Roy method (Algorithm 3.5), $V^{\tau}_{\text{Low}}(S(t_i))$ is defined as the corresponding discounted option value process by using the exercise strategy τ from time t_i forwards assuming not exercised before t_i, which is a supermartingal:

$$V^{\tau}_{\text{Low}}(S(t_i)) = \mathbb{E}[g(S(\tau(i)))|S(t_i)], \quad i = 1, \dots, m \tag{3.21}$$

$V^{\tau}_{\text{Low}}(S(t_0))$ is the option value at time t_0 computed by this exercise strategy, which is a lower bound for the true option price $V(S(t_0))$.

The Doob-Meyer decomosition of $V^{\tau}_{\text{Low}}(S(t_i))$ is:

$$V^{\tau}_{\text{Low}}(S(t_i)) = V^{\tau}_{\text{Low}}(S(t_0)) + M^{\tau}_{\text{Doob}}(t_i) + A(t_i), \quad i = 1, \dots, m$$

with M^{τ}_{Doob} being the martingale component, A being the nonincreasing predictable component and $M^{\tau}_{\text{Doob}}(t_0) = A(t_0) = 0$.

Further, we define $\Delta(t_i)$ as the difference of $M^{\tau}_{\text{Doob}}(t_i)$ and $M^{\tau}_{\text{Doob}}(t_{i-1})$:

$$\begin{aligned}\Delta(t_i) &:= M^{\tau}_{\text{Doob}}(t_i) - M^{\tau}_{\text{Doob}}(t_{i-1}) \\ &= V^{\tau}_{\text{Low}}(S(t_i)) - \mathbb{E}[V^{\tau}_{\text{Low}}(S(t_i))|S(t_{i-1})] \\ &= \mathbb{E}[g(S(\tau(i)))|S(t_i)] - \mathbb{E}[g(S(\tau(i)))|S(t_{i-1})] \qquad (3.22)\end{aligned}$$

which measures the quality of the lower bound for the option price at each time t_i. Thus the upper bound $V^{\tau}_{\text{bound}}(S(t_0))$ is according to the theorem 3.9 given by:

$$\begin{aligned}V^{\tau}_{\text{upper}}(S(t_0)) &= \mathbb{E}[\max_{i=1,\ldots,m}(g(S(t_i)) - M^{\tau}_{\text{Doob}}(t_i))] \\ &= \mathbb{E}[\max_{i=1,\ldots,m}(g(S(t_i)) - \sum_{i=1}^{k}\Delta(t_i))] \qquad (3.23)\end{aligned}$$

The duality gap of the exercise strategy τ is then computed as:

$$\begin{aligned}\Delta^{\tau} &= V^{\tau}_{\text{upper}}(S(t_0)) - V^{\tau}_{\text{low}}(S(t_0)) \\ &= \mathbb{E}\left[\max_{i=1,\ldots,m}\left(g(S(t_i)) - \sum_{i=1}^{k}\Delta(t_i)\right)\right] - V^{\tau}_{\text{low}}(S(t_0)) \\ &= \mathbb{E}\left[\max_{i=1,\ldots,m}\left(g(S(t_i)) - \sum_{i=1}^{k}(V^{\tau}_{\text{Low}}(S(t_i)) - \mathbb{E}[V^{\tau}_{\text{Low}}(S(t_i))|S(t_{i-1})])\right)\right] - V^{\tau}_{\text{low}}(S(t_0)) \\ &= \mathbb{E}\left[\max_{i=1,\ldots,m}\left(g(S(t_i)) - \sum_{i=1}^{k}V^{\tau}_{\text{Low}}(S(t_i)) + \sum_{i=1}^{k}\mathbb{E}[V^{\tau}_{\text{Low}}(S(t_i))|S(t_{i-1})]\right)\right] - V^{\tau}_{\text{low}}(S(t_0)) \\ &= \mathbb{E}\left[\max_{i=1,\ldots,m}\left(g(S(t_i)) - V^{\tau}_{\text{Low}}(S(t_i)) + \sum_{i=1}^{k}\mathbb{E}[V^{\tau}_{\text{Low}}(S(t_i)) - V^{\tau}_{\text{Low}}(S(t_{i-1}))|S(t_{i-1})]\right)\right]\end{aligned}$$

If the exercise strategy τ is the optimal exercise strategy τ^*, then $\sum_{i=1}^{k}\mathbb{E}[V^{\tau^*}_{\text{Low}}(S(t_i)) - V^{\tau^*}_{\text{Low}}(S(t_{i-1}))|S(t_{i-1})] = 0$, thus according to the theorem 3.9, the duality gap $\Delta^{\tau^*} = 0$.

We notice that estimation of the upper bound $V^{\tau}_{\text{upper}}(S(t_0))$ requires determination of the martingale M^{τ}_{Doob} in the equation (3.23), which furthermore requires computation of the conditional expectations $\mathbb{E}[g(S(\tau(i)))|S(t_i)]$ and $\mathbb{E}[g(S(\tau(i)))|S(t_{i-1})]$ in the equation (3.22), which is but computationally expensive.

Andersen and Broadie [2] set up a *primal-dual algorithm* to compute the conditional expectations. Their approach can be conjuncted with any algorithms generating a lower bound, e.g the modified Tsitsiklis-Roy method (Algorithm 3.5) and the Longstaff-Schwartz method (Algorithm 3.4). For instance, denote

$(\tau(1), \ldots, \tau(m))$ as the estimator of the optimal stopping time by the Longstaff-Schwartz method. The estimation of the continuation value function at each potential exercise date t_i is:

$$C^*(S^{(n)}(t_i)) \quad = \quad \sum_{l=1}^{k} a^*_{li} H_l(S^{(n)}(t_i))$$

The exercise rule is determined by comparing the continuation value $C^*(S^{(n)}(t_i))$ and the discounted exercising value $g(S^{(n)}(t_i))$.

Andersen and Broadie use a nested simulation to compute the conditional expectations $V^{\tau}_{\text{Low}}(S(t_i))$ and $\mathbb{E}[V^{\tau}_{\text{Low}}(S(t_i))|S(t_{i-1})]$, and then $\Delta(t_i)$ in the equation (3.22). They use a straightforward Monte Carlo to estimate $\mathbb{E}[V^{\tau}_{\text{Low}}(S(t_i))|S(t_{i-1})]$ and rewrite $V^{\tau}_{\text{Low}}(S(t_i))$ in terms of $\mathbb{E}[V^{\tau}_{\text{Low}}(S(t_i))|S(t_{i-1})]$:

$$\begin{aligned}
V^{\tau}_{\text{Low}}(S(t_i)) &= \mathbb{E}[V^{\tau}_{\text{Low}}(S(t_i))|S(t_i)] \\
&= \begin{cases} g(S(t_i)), & \text{if} \quad g(S(t_i)) \geq C^*(S^{(n)}(t_i)) \\ \mathbb{E}(g(S(\tau(i+1)))|S(t_i)), & \text{if} \quad g(S(t_i)) < C^*(S^{(n)}(t_i)) \end{cases} \\
&= \begin{cases} g(S(t_i)), & \text{if} \quad g(S(t_i)) \geq C^*(S^{(n)}(t_i)) \\ \mathbb{E}[V^{\tau}_{\text{Low}}(S(t_{i+1}))|S(t_i)], & \text{if} \quad g(S(t_i)) < C^*(S^{(n)}(t_i)) \end{cases}
\end{aligned}$$

Now we can present the algorithm of Andersen-Broadie 3.6.

Algorithm 3.6 Andersen-Broadie algorithm

Generate N_1 independent paths (out-of-samples) for stock at all possible exercise dates: $\{S^{(n)}(t_1), S^{(n)}(t_2), \ldots, S^{(n)}(t_m)\}$ with $n = 1, \ldots, N_1$, $t_i = \frac{T}{m} \times i$, $i = 1, \ldots, m$.
Repeat the following iteration for each path $n = 1, \ldots, N_1$:

1. Set $M^{(n)}(t_0) = 0$, $g(S^{(n)}(t_0)) = 0$.
2. For each $i = 0, \ldots, m$

   1) If $0 \leq i \leq m-1$, simulate N_2 subpaths for stock $S^{(\hat{n})}_{\text{Sub}}(t_i), \ldots, S^{(\hat{n})}_{\text{Sub}}(\tau(i+1))$ with $\hat{n} = 1, \ldots, N_2$ and $S^{(n)}_{\text{Sub}}(t_i) = S^{(n)}(t_i)$.
   Estimate $\mathbb{E}(V^{\tau}_{\text{Low}}(S^{(n)}(t_{i+1}))|S^{(n)}(t_i))$ by:

   $$\mathbb{E}(V^{\tau}_{\text{Low}}(S^{(n)}(t_{i+1}))|S^{(n)}(t_i)) = \mathbb{E}(g(S^{(n)}(\tau(i{+}1)))|S^{(n)}(t_i)) = \frac{1}{N_2}\sum_{n=1}^{N_2} g(S^{(\hat{n})}_{\text{Sub}}(\tau(i{+}1)))$$

   2) If $1 \leq i \leq m-1$, evaluate $g(S^{(n)}(t_i))$ and $C^*(S^{(n)}(t_i))$, check which is larger and determine $V^{\tau}_{\text{Low}}(S(t_i))$ by:

   $$\begin{aligned} V^{\tau}_{\text{Low}}(S^{(n)}(t_i)) &= \mathbb{E}[V^{\tau}_{\text{Low}}(S^{(n)}(t_i))|S^{(n)}(t_i)] \\ &= \begin{cases} g(S^{(n)}(t_i)), & \text{if } \; g(S^{(n)}(t_i)) > C^*(S^{(n)}(t_i)) \\ & \&\& \;\; g(S^{(n)}(t_i)) > 0 \\ \mathbb{E}[V^{\tau}_{\text{Low}}(S^{(n)}(t_{i+1}))|S^{(n)}(t_i)], & \text{else} \end{cases} \end{aligned}$$

   If $i = m$, $V^{\tau}_{\text{Low}}(S^{(n)}(t_m)) = g(S^{(n)}(t_m))$.

   3) If $1 \leq i \leq m$, set:

   $$\begin{aligned} \Delta^{(n)}(t_i) &= V^{\tau}_{\text{Low}}(S^{(n)}(t_i)) - \mathbb{E}[V^{\tau}_{\text{Low}}(S^{(n)}(t_i))|S^{(n)}(t_{i-1})] \\ M^{(n)}(t_i) &= M^{(n)}(t_{i-1}) + \Delta^{(n)}(t_i) \end{aligned}$$

3. Compute $V^{\tau}_{\text{Upper}}(S^{(n)}(t_0)) = \max_{i=1,\ldots,m}(g(S^{(n)}(t_i)) - M^{(n)}(t_i))$

The upper bound for the option price is thus: $V^{\tau}_{\text{Upper}}(S(t_0)) = \frac{1}{N_1}(V^{\tau}_{\text{Upper}}(S^{(n)}(t_0)))$, which can be paired with the lower bound $V^{\tau}_{\text{Low}}(S(t_0))$.

3.2 Numerical Studies

3.2.1 Approximation of American Option by Bermudan Counterpart

As mentioned in the section 3.1.2, the price of an American option, which can be exercised at any time $t \in [0, T]$, can be approximated by its corresponding Bermudan counterpart, which can be exercised only at discrete times, e.g. at time $t_i = i\frac{T}{m}, i = 1, \dots, m$.

Approximation without Extrapolation

According to the research of Bally and Pàges [3], if the payoff function is Lipschitz continuous, the rate of convergence is $\frac{1}{\sqrt{m}}$ and if the payoff function is semi-convex, the rate of convergence is $\frac{1}{m}$. Since most payoff functions include positive part of extrema (e.g $\max((S(t) - K), 0)$) or linear combinations of the components of the underlings (e.g $\frac{1}{2}(S_1(t) + S_2(t))$), they belong to the family of semi-convex functions.

Approximation with Extrapolation

In order to further improve the convergence rate, we notice that methods like the Longstaff-Schwartz algorithm leads to approximately monotone convergence of the option price, therefore we can apply Richardson extrapolation techniques. If the payoff function is Lipschitz continuous, Richardson extrapolation leads to the aggregated option price estimate as follows:

$$P_A(2m) = \frac{\sqrt{2}P_B(2m) - P_B(m)}{\sqrt{2} - 1} \tag{3.24}$$

where $P_B(m), P_B(2m)$ denotes the Bermudan option price using m and $2m$ exercise dates and $P_A(2m)$ denotes the aggregated American option price estimate using $2m$ exercise dates. If the payoff function is semi-convex, Richardson extrapolation gives the following formula:

$$P_A(2m) = 2P_B(2m) - P_B(m) \tag{3.25}$$

We test the performance for approximation for 1-D American put option, where the payoff, parameters and the benchmark value are given in Test Case 1 of the section 7.1. A series of 1-D Bermudan put option with the same parameters are computed to approximate the American one. The number of potential exercise times for the Bermudan options are $m \in \{2, 4, 6, \dots, 50\}$. The algorithm we use is the Longstaff-Schwartz method in algorithm 3.2 with number of paths 1000000. The basis functions chosen in the algorithm are $\{1, S, S^2, f(S)\}$, where S is the stock price and $f(S) = (K - S)^+$ is the payoff, which is semi-convex, hence the corresponding extrapolation formula (3.25) can be used. The selected values of the Bermudan options are showed in table 3.1.

We plot the Bermudan option values of Table 3.1 against the number of potential exercise date m. Figure 3.2 makes use of approximation without extrapolation while Figure 3.3 makes use of approximation with extrapolation. From the pictures, we see that when using extrapolation to approximate an American option by its Bermudan counterparty, exercise times $m \geq 20$ gives good approximation, while exercise times $m \geq 10$ already approximate well with the help of extrapolation formula (3.25).

Test Case 1, Benchmark: 7.11			
Exercise times m	Option price	Standard error	95%-confidence interval
2	6.9269	0.0066	[6.9140, 6.9398]
4	7.0197	0.0064	[7.0072, 7.0322]
6	7.0449	0.0063	[7.0326, 7.0572]
8	7.0444	0.0062	[7.0322, 7.0566]
10	7.0601	0.0062	[7.0479, 7.0723]
12	7.0657	0.0062	[7.0535, 7.0779]
14	7.0926	0.0062	[7.0804, 7.1048]
16	7.0735	0.0061	[7.0615, 7.0855]
18	7.0825	0.0061	[7.0705, 7.0945]
20	7.0862	0.0061	[7.0742, 7.0982]
22	7.0831	0.0061	[7.0711, 7.0951]
24	7.0860	0.0061	[7.0740, 7.0980]
26	7.0895	0.0061	[7.0775, 7.1015]
28	7.0885	0.0061	[7.0765, 7.1005]
30	7.0781	0.0060	[7.0663, 7.0899]
32	7.0947	0.0061	[7.0827, 7.1067]
34	7.0897	0.0060	[7.0779, 7.1015]
36	7.0907	0.0060	[7.0789, 7.1025]
38	7.0824	0.0060	[7.0706, 7.0942]
40	7.0923	0.0060	[7.0805, 7.1041]
42	7.0892	0.0060	[7.0774, 7.1010]
44	7.1012	0.0060	[7.0894, 7.1130]
46	7.0819	0.0060	[7.0701, 7.0937]
48	7.0921	0.0060	[7.0803, 7.1039]
50	7.0971	0.0060	[7.0853, 7.1089]

Table 3.1: Approximation of an American option by its Bermudan counterpart for Test Case 1

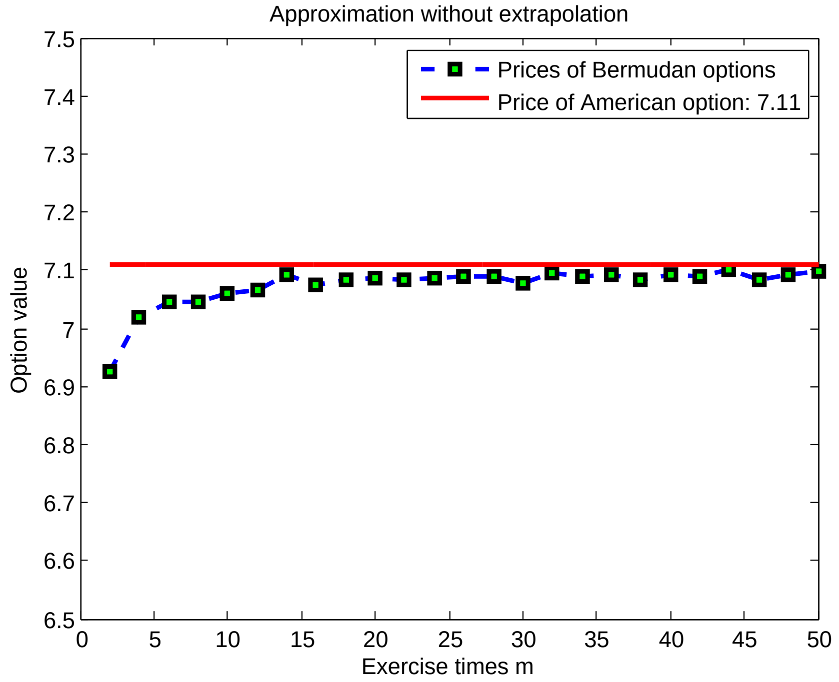


Figure 3.2: Approximation without extrapolation for Test Case 1

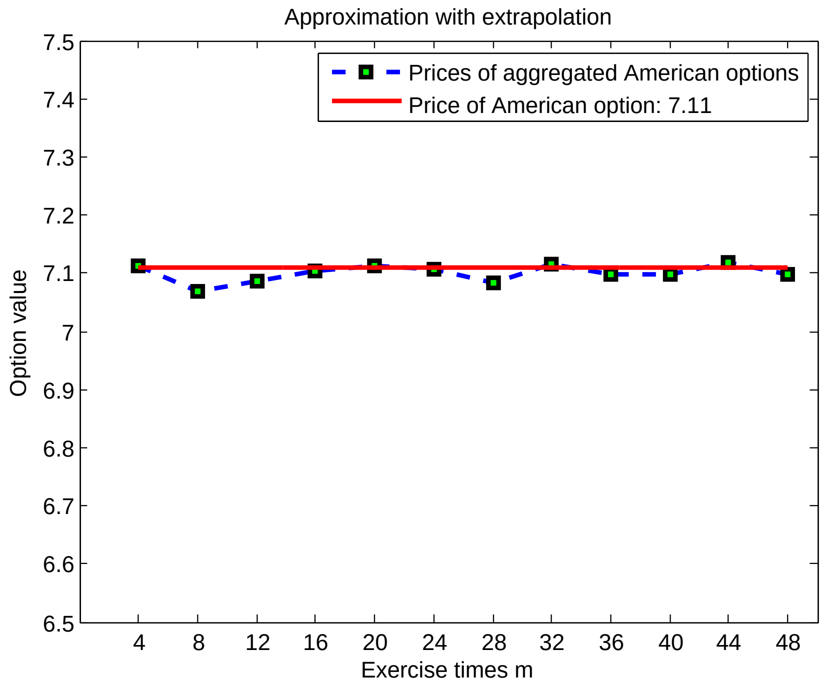


Figure 3.3: Approximation with extrapolation for Test Case 1

3.2.2 Low Bias, High Bias and Mixture of Bias

As mentioned before, the Longstaff-Schwartz method and the Tsitsiklis-Roy method mix high bias and low bias. The modified Longstaff-Schwartz method and the modified Tsitsiklis-Roy method estimate conditional expectation regression coefficients from an in-sample set of paths and apply this stopping rule to an out-of-sample set of paths, thus they should show only low bias compared with the benchmark. However Longstaff and Schwartz [33] pointed that their method almost give very similar result as the modified one. We will test this conclusion in this section. 1-D American option for Test Case 2, 2-D American minimum option for Test Case 12 and 3-D American minimum option for Test Case 16 in section 7.1 are chosen as showcase settings.

Numerical results are collected in Table 3.2, where "ATM-Benchmark", "ITM-Benchmark" and "OTM-Benchmark" mean respectively benchmark prices for American At-the-Money option, American In-the-Money option and American Out-of-the-Money option and "TR" and "LS" are abbreviation for Tsitsiklis-Roy method and Longstaff-Schwartz method. The number of paths for in-sample set to obtain stopping rule is 1000000. The number of new paths as out-of-sample set to apply stopping rule is also 1000000. The basis functions chosen in the algorithms are $\{1, S, S^2\}$ for Test Case 2, $\{1, S_1, S_2, S_1^2, S_2^2\}$ for Test Case 12 and $\{1, S_1, S_2, S_3, S_1^2, S_2^2, S_3^2\}$ for Test Case 16. The standard errors for all simulations are less than 0.01.

From this table, we notice that the difference between the Longstaff-Schwartz method and the modified Longstaff-Schwartz method with out-of-sample set of paths is very small while the difference between the Tsitsiklis-Roy method and the modified Tsitsiklis-Roy method with out-of-sample set of paths are much bigger. The option prices by the Longstaff-Schwartz method are in practice slightly smaller than benchmarks while the ones by the Tsitsiklis-Roy method are significantly bigger than benchmarks when using the set of chosen basis functions. Both the modified Longstaff-Schwartz method and the modified Tsitsiklis-Roy method give very accurate result compared with benchmarks and are both slightly smaller than the actual option prices. However the modified Longstaff-Schwartz method seems to deliver even better outputs than the modified Tsitsiklis-Roy method, as its outputs are more close to the benchmarks. Whether this observation results from the nature of both algorithms or just from the choice of certain basis functions, we will keep on testing in the section 3.2.4 by making use of more choices of basis functions.

Test Case 2				
ATM-Benchmark	TR	Modified TR	LS	Modified LS
11.28	11.9771	11.2119	11.2819	11.2683
ITM-Benchmark	TR	Modified TR	LS	Modified LS
22.74	23.5046	22.6474	22.7099	22.7235
OTM-Benchmark	TR	Modified TR	LS	Modified LS
4.97	5.3404	4.9241	4.9591	4.9622
Test Case 11				
ATM-Benchmark	TR	Modified TR	LS	Modified LS
2.28	4.2995	2.2026	2.2229	2.2241
ITM-Benchmark	TR	Modified TR	LS	Modified LS
5.97	8.5706	5.6159	5.8464	5.8437
OTM-Benchmark	TR	Modified TR	LS	Modified LS
0.029	0.1169	0.0260	0.0288	0.0291
Test Case 16				
ATM-Benchmark	TR	Modified TR	LS	Modified LS
0.81	1.3873	0.7942	0.7947	0.7926
ITM-Benchmark	TR	Modified TR	LS	Modified LS
2.82	3.8875	2.7476	2.7661	2.7638
OTM-Benchmark	TR	Modified TR	LS	Modified LS
0.0022	0.0060	0.0024	0.0023	0.0021

Table 3.2: Low bias, high bias and mixture of bias

3.2.3 In-the-Money Paths vs All Paths

Longstaff and Schwartz include only in-the-money paths in the regression to estimate the continuation value and demonstrate that this increases the efficiency of the algorithm than using all paths. However Glasserman gives an example on page 463 of his book [15] and points out that results with in-the-money paths are even inferior than results with all paths.

In this section, we also use all paths in the regression part of the Longstaff-Schwartz method and the modified Longstaff-Schwartz method and compare the numerical results with the original one using in-then-money paths. We choose 3-D American maximum outperformance option for Test Case 15 in the section 7.1 as showcase setting. We don't make use of one single set of basis functions as in the section 3.2.2, but multiple sets of basis functions to avoid that the choice of basis functions to affect the reliability of the test.

Type	Number	Basis Functions Forms
I	4	$1, f(S_1,S_2,S_3), f^2(S_1,S_2,S_3), f^3(S_1,S_2,S_3)$
II	7	$1, S_1, S_2, S_3, S_1^2, S_2^2, S_3^2$
III	10	$1, S_1, S_2, S_3, S_1^2, S_2^2, S_3^2, S_1^3, S_2^3, S_3^3$
IV	12	$1, S_1, S_2, S_3, S_1^2, S_2^2, S_3^2, S_1^3, S_2^3, S_3^3, S_1S_2S_3, f(S_1,S_2,S_3)$
V	17	$1, S_1, S_2, S_3, S_1^2, S_2^2, S_3^2, S_1^3, S_2^3, S_3^3, S_1S_2, S_1S_3, S_2S_3, S_1S_2S_3,$ $f(S_1,S_2,f_3), f^2(S_1,S_2,f_3), f^3(S_1,S_2,f_3)$
VI	20	$1, S_1, S_2, S_3, S_1^2, S_2^2, S_3^2, S_1S_2, S_1S_3, S_2S_3,$ $S_1^3, S_2^3, S_3^3, S_1^2S_2, S_1^2S_3, S_2^2S_1, S_2^2S_3, S_3^2S_1, S_3^2S_2, S_1S_2S_3$
VII	22	$1, S_1, S_2, S_3, S_1^2, S_2^2, S_3^2, S_1^3, S_2^3, S_3^3, S_1^4, S_2^4, S_3^4,$ $S_1^5, S_2^5, S_3^5, S_1^6, S_2^6, S_3^6, S_1^7, S_2^7, S_3^7$
VIII	35	$1, S_1, S_2, S_3, S_1^2, S_2^2, S_3^2, S_1S_2, S_1S_3, S_2S_3,$ $S_1^3, S_2^3, S_3^3, S_1^2S_2, S_1^2S_3, S_2^2S_1, S_2^2S_3, S_3^2S_1, S_3^2S_2, S_1S_2S_3,$ $S_1^4, S_2^4, S_3^4, S_1^3S_2, S_1^3S_3, S_2^3S_1, S_2^3S_3, S_3^3S_1, S_3^3S_2,$ $S_1^2S_2^2, S_1^2S_3^2, S_2^2S_3^2, S_1^2S_2S_3, S_2^2S_1S_3, S_3^2S_1S_2$

Table 3.3: Sets of basis functions for the test of "In-the-Money Paths vs All Paths"

The sets of basis functions are presented in Table 3.3. They are noted as from Type I to Type III. In Type I, we use 4 basis functions up to polynomial degree 3, where $f(S_1,S_2,S_3)$ is the payoff function defined as: $f(S_1,S_2,S_3) := (\max\{S_1(t),S_2(t),S_3(t)\} - K)^+$. Type II consists of 10 basis functions with monomial polynomials up to degree 2. Type III consists of 7 basis functions with monomial polynomials up to degree 3. In Type IV, We add two terms $S_1S_2S_2$ and $f(S_1,S_2,S_3)$ to Type III. Type V contains 17 basis functions including 14 monomial polynomials up to degree 3 and 3 payoff functions up to degree 3. In Type VI, we use 20 basis functions purely being monomial polynomials up to degree 3. Type VII consists of 22 monomial polynomials up to degree 7 as basis functions. In Type VIII we make use of the most basis functions, namely 35 monomial polynomials up to degree 4.

All numerical results are collected in Table 3.4, 3.5 and 3.6. "LS" and "Modified LS" abbreviate the original algorithm of Longstaff-Schwartz method using only in-the-money paths for regression and its modified one. "LS-All" and "Modified LS-All" mean the usage of all paths for regression and its modified one.

We test not only at-the-money (ATM) option, but also in-the-money (ITM) and out-of-the-money (OTM) option. From the tables, we see that the "LS" algorithm using in-the-money paths present better result than the "LS-All" algorithm using all paths and are not so sensible to the choice of basis functions. The "Modified LS" and the "Modified LS-All" algorithm both use out-of-sample paths and therefore display a low bias compared with the benchmarks. However the difference between the "Modified LS" and the corresponding benchmark is much smaller than the difference between the "Modified LS-All" and the benchmark, which also shows that the usage of all paths for regression is worse than the usage of only in-the-money paths.

The reason can be seen intuitively from Figure 3.4 and 3.5. The blue circle is a sample with stock price as x-coordinates and the value of option as y-coordinates assuming that the option has not been exercised before at this time. Here we use 10000 paths for samples. The red curve is the discounted payoff function. The yellow curve is the real continuation value function. In Figure 3.4, we use all paths to do regression. In Figure 3.5, we use only in-the-money paths to do regression. The green curve is the estimated continuation value function, either using all paths or using in-the-money paths for regression. We see clearly that the green curve in Figure 3.4 is much closer to the red curve than the one in Figure 3.5 especially within the area where the stock prices are less than the strike price 100, i.e out-of-the-money, while the green curve in Figure 3.5 is much closer to the red curve than the one in Figure 3.4 within the area where the stock prices are more than 100, i.e in-the-money. Within the out-of-the-money area, according to the Longstaff-Schwartz algorithm, holding the option and keeping the option value the same as in the previous time step is a clear decision. Thus a good fit in Figure 3.4 is not necessary for the whole algorithm. We are only interested in the estimated continuation value function for the in-the-money area, a good fit in this area is very important for the accuracy of the whole algorithm. Especially when the estimated continuation function (the green curve) is lower than the discounted payoff function (the red curve), the option holder should make a decision to exercise the option immediately rather than holding the option. Thus a good fit within the in-the-money area in Figure 3.5 is necessary.

Since the difference and the corresponding error of two kinds of regression is propagated backwards through time, we see that regression using in-the-money paths gives more accurate result than the one using all paths.

Test Case 15, ATM-Benchmark: 17.50					
Basis Function	Number	LS	Modified LS	LS-All	Modified LS-All
I	4	17.1102	17.1499	17.0524	17.0985
II	7	17.4061	17.4208	17.3844	17.3694
III	10	17.3665	17.4053	17.3041	17.3547
IV	12	17.4501	17.4761	17.4642	17.4568
V	17	17.4569	17.4997	17.5049	17.4656
VI	20	17.4625	17.4811	17.3529	17.4047
VII	22	17.3785	17.3963	17.3555	17.3520
VIII	35	17.4626	17.5036	17.4851	17.4482

Table 3.4: In-the-money paths for regression vs all paths for regression for 3-D American maximum ATM option. Each estimate has a standard error of approximately 0.03.

Test Case 15, ITM-Benchmark: 25.98					
Basis Function	Number	LS	Modified LS	LS-All	Modified LS-All
I	4	25.2939	25.3114	25.2306	25.2735
II	7	25.7998	25.8376	25.8075	25.8026
III	10	25.7648	25.8063	25.8081	25.7758
IV	12	25.9109	25.9713	25.8950	25.9218
V	17	25.9410	25.9633	25.9400	25.9750
VI	20	25.9189	25.9731	25.8792	25.9018
VII	22	25.7760	25.8130	25.7861	25.8227
VIII	35	25.9645	25.9288	25.9250	25.9258

Table 3.5: In-the-money paths vs all paths for 3-D American maximum ITM option. Each estimate has a standard error of approximately 0.035.

Test Case 15, OTM-Benchmark: 2.27					
Basis Function	Number	LS	Modified LS	LS-All	Modified LS-All
I	4	2.2688	2.2824	2.2691	2.2720
II	7	2.2685	2.2615	2.1733	2.1682
III	10	2.2862	2.2821	2.2488	2.2483
IV	12	2.2853	2.2819	2.2639	2.2526
V	17	2.2931	2.2741	2.2650	2.2668
VI	20	2.2922	2.2879	2.2465	2.2492
VII	22	2.2879	2.2783	2.2367	2.2495
VIII	35	2.2905	2.2869	2.2466	2.2411

Table 3.6: In-the-money paths vs all paths for 3-D American maximum OTM option. Each estimate has a standard error of approximately 0.01.

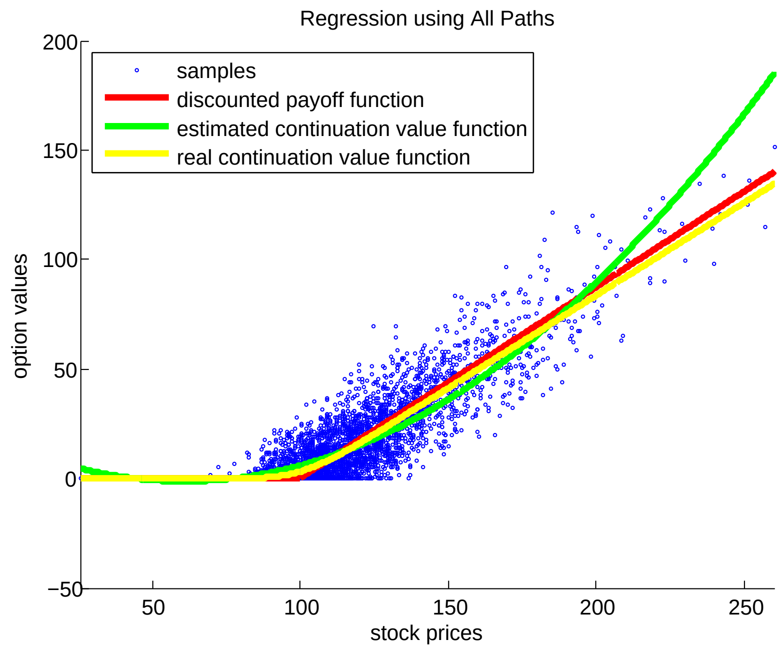


Figure 3.4: Regression part of the Longstaff-Schwartz Method using all paths

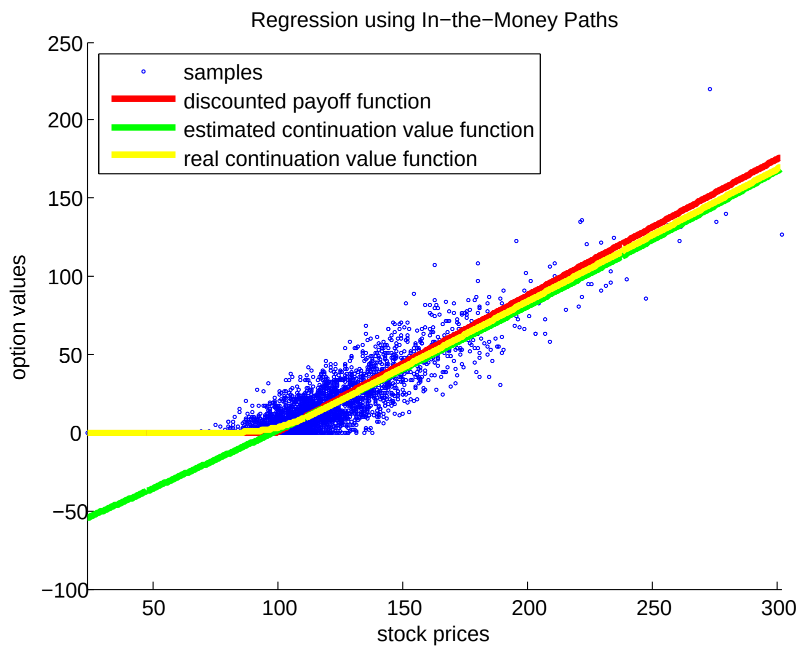


Figure 3.5: Regression part of the Longstaff-Schwartz Method using only in-the-money paths

3.2.4 Longstaff-Schwartz Method vs Tsitsiklis-Roy Method

In this section, we test the accuracy of the Longstaff-Schwartz method and the Tsitsiklis-Roy method. We test 2-D American maximum option in Test Case 10 in the section 7.1 by these two methods and their modified versions with low bias.

Type	Number	Basis Functions Forms
I	7	$1, S_1, S_2, S_1^2, S_2^2, S_1^3, S_2^3$
II	8	$1, S_1, S_2, S_1^2, S_2^2, S_1^3, S_2^3, S_1S_2$
III	9	$1, S_1, S_2, S_1^2, S_2^2, S_1^3, S_2^3, S_1S_2, \max(S_1, S_2)$
IV	10	$1, S_1, S_2, S_1^2, S_2^2, S_1^3, S_2^3, S_1S_2, S_1^2S_2, S_1S_2^2$
V	7	$1, S_1, S_2, S_1^2, S_2^2, S_1S_2, f(S_1, S_2)$
VI	11	$1, S_1, S_2, S_1^2, S_2^2, S_1^3, S_2^3, S_1S_2, S_1^2S_2, S_1S_2^2, f(S_1, S_2)$

Table 3.7: Sets of basis functions for the test of "Longstaff-Schwartz Method vs Tsitsiklis-Roy Method"

Different choices of basis functions haven been tested, see Table 3.7. In Type I, we include 7 basis functions, i.e monomial polynomials with degree up to 3. Type II contains all basis functions of Type I and plus the the term of S_1S_2. Type III contains all basis functions of Type II and plus the new one $\max(S_1, S_2)$. As Type IV, we take all basis functions of Type II and add two new ones $S_1^2S_2$ and $S_1S_2^2$. Type V makes use of all basis functions of Type IV and adds the payoff function $f(S_1, S_2) := \max(\max(S_1, S_2) - K, 0)$. Type VI also use the payoff function but with fewer monomial polynomials.

Again, we test not only 2-D American at-the-money (ATM) option, but also in-the-money (ITM) and out-of-the-money (OTM) option. Numerical results are collected in Table 3.8, 3.9 and 3.10. "LS", "Modified LS", "TR" and "Modified TR" are respectively abbreviations of the Longstaff-Schwartz method, the modified Longstaff-Schwartz method with out-of-sample paths, the Tsitsiklis-Roy Method and the modified Tsitsiklis-Roy Method with out-of-sample paths. For "LS" and "TR", 1000000 stock price paths are simulated to calculate the estimated continuation value and the corresponding stopping rule. For "Modified LS" and "Modified TR", again 1000000 new out-of-sample paths are simulated to compute the modified option price based on the previously obtained stopping rule.

The bechmarks computed by the binomial-tree method are respectively 13,90 for the ATM option, 21,34 for the ITM option and 1,64 for the OTM option.

From the table, we see clearly that the estimator by the Tsitsiklis-Roy Method have significant high bias than the benchmarks, either in the ATM case, or in the ITM case or in the OTM case. The choice of basis functions affects the bias. In Type I, Type II, Type III and Type IV, choices of basis functions all show bad results for the Tsitsiklis-Roy Method. However including the payoff function $f(S_1, S)$ can improve the estimator in Type V and Type VI. Especially in Type VI, additionally adding the interaction term S_1S_2, $S_1^2S_2$ and $S_1S_2^2$ can decrease the bias to a low level.

Compared with the Tsitsiklis-Roy method, the price estimator by the Longstaff-Schwartz method gives much smaller bias, and practically it shows often low bias, either in the ATM case, or in the ITM case or in the OTM case, using any tested type of basis functions. And the choice of basis functions in the Longstaff-Schwartz method is not so sensitive as in the Tsitsiklis-Roy method.

When we compared the modified Longstaff-Schwartz method and the modified Tsitsiklis-Roy method with out-of-sample paths, we notice that both methods show low bias compared with the benchmarks. And the price estimator by the modified Longstaff-Schwartz method is closer to the benchmark than the price estimator by the modified Tsitsiklis-Roy method, which means that the Longstaff-Schwartz method estimate the continuation value more accurately than the Tsitsiklis-Roy method.

The reason is threefold:

1. The Longstaff-Schwartz method only use in-the-money stock prices to do regression while the Tsitsiklis-Roy method use all paths to regress. As shown in the previous section 3.2.3, the regression using in-the-money stocks gives a better fit to the continuation value of holding the option within the in-the-money area than the regression using all stocks. And the fit of the continuation value within the in-the-money area is very crucial while the fit within the out-of-the-money area is not so important. Thus the Longstaff-Schwartz method gives better regression of the continuation value than the Tsitsiklis-Roy method does.

2. When the discounted payoff value is larger than the estimated continuation vale, according to the Tsitsiklis-Roy method, we should exercise the option immediately. However there is one situation that this method doesn't think of. When the estimated continuation value is negative and the discounted payoff value is zero, we should exercise the option according to the Tsitsiklis-Roy method and clearly this decision is wrong, since the payoff is zero and the option is out-of-the-money and we should definitely hold it. The reason why we get a negative continuation value comes from regressing with improper number or form of basis functions. The Longstaff-Schwartz method takes account of this situation and recommends the option holder to exercise the option only when the discounted payoff value is bigger than the estimated continuation value and at the same time the discounted payoff should also be positive, which means the option is in-the-money. In this sense, the Longstaff-Schwartz method is better than the Tsitsiklis-Roy method.

3. When the discounted payoff value is lower than the estimated continuation value, the option holder should hold the option. According to the Tsitsiklis-

Roy method, he should update the option value equaling the estimated continuation value, while according to the Longstaff-Schwartz method, he should keep the option value the same as in the previous time step. Since the strategy of the Longstaff-Schwartz method incorporates all future time steps up to the maturity of the option, the Longstaff-Schwartz method reduces the bias resulting from estimating the continuation value while the Tsitsiklis-Roy method introduce this bias to the whole algorithm.

We plot the regression process for estimating the continuation value of 1-D American call option at different exercise dates respectively by the Longstaff-Schwartz method in Figure 3.6 and by the Tsitsiklis-Roy method in Figure 3.7 using 10000 paths of stocks. Each figure contains two parts. The upper part shows regression at one time step before maturity of the option, where the real continuation value function (yellow curve) can be exactly computed by the Black-Scholes Formula for the European option (see equation (1.10)). The lower part shows regression at nine time step before maturity, where the real continuation value function is not clear.

Comparing the first part of Figure 3.6 and Figure 3.7, we notice that the fit of continuation value within the in-the-money area (stock price > 100) by the Lonstaff-Schwartz method is better than by the Tsitsiklis-Roy method while the fit of continuation value within the out-of-the-money area (stock price < 100) by the Tsitsiklis-Roy method is better than by the Lonstaff-Schwartz method, because we use only in-the-money paths to regress in the Longstaff-Schwartz method but use all paths to do regression in the Tsitsiklis-Roy method.

After regressing at 9 potential exercise dates, we reach the lower part of Figure 3.6 and Figure 3.7. Comparing both of them, we find that the estimated continuation value function in each figure seems similar, which leads to that the numerical result by both methods are nearly identical. However if we watch the figures in details and take account of all the previously mentioned bias by the Tsitsiklis-Roy method, we see that the intersecting point between the estimated continuation value function (green curve) and the discounted payoff function (red curve) by both methods are not the same.

For the Longstaff-Schwartz method, the intersecting point is a little bit smaller than 120. On the other hand, the intersecting point by the Tsitsiklis-Roy method is slightly bigger than 120. That means, the optimal exercise region and the corresponding stopping rule by both methods are similar but not the same and the option value calculated by the Tsitsiklis-Roy method shows practically a high bias than the benchmark while the option value calculated by the Longstaff-Schwartz method shows a low bias.

Test Case 10, ATM-Benchmark: 13.90					
Basis Function	Number	LS	Modified LS	TR	Modified TR
I	7	13.7551	13.7698	15.6548	13.6228
II	8	13.8511	13.8273	15.1730	13.6562
III	9	13.8402	13.8796	15.1644	13.6935
IV	10	13.8417	13.8443	15.0044	13.7458
V	7	13.8649	13.8374	14.0039	13.8069
VI	11	13.8385	13.8659	13.9189	13.8427

Table 3.8: Longstaff-Schwartz Method vs Tsitsiklis-Roy Method for the Test Case 10: 2-D American maximum ATM option. Each estimate has a standard error of approximately 0.03.

Test Case 10, ITM-Benchmark: 21.34					
Basis Function	Number	LS	Modified LS	TR	Modified TR
I	7	21.1850	21.1601	23.8391	20.9041
II	8	21.2650	21.2955	22.5281	21.0806
III	9	21.2573	21.2895	22.3556	21.0447
IV	10	21.2881	21.2779	22.4270	21.1398
V	7	21.2686	21.2836	21.3361	21.2460
VI	11	21.2655	21.3018	21.3402	21.2932

Table 3.9: Longstaff-Schwartz Method vs Tsitsiklis-Roy Method for the Test Case 10: 2-D American maximum ITM option. Each estimate has a standard error of approximately 0.03.

Test Case 10, OTM-Benchmark: 1.64					
Basis Function	Number	LS	Modified LS	TR	Modified TR
I	7	1.6370	1.6290	2.3824	1.6380
II	8	1.6494	1.6305	2.2408	1.6441
III	9	1.6386	1.6382	2.2077	1.6395
IV	10	1.6299	1.6237	2.2328	1.6455
V	7	1.6300	1.6402	1.8231	1.6393
VI	11	1.6460	1.6369	1.7818	1.6354

Table 3.10: Longstaff-Schwartz Method vs Tsitsiklis-Roy Method for the Test Case 10: 2-D American maximum OTM option. Each estimate has a standard error of approximately 0.01.

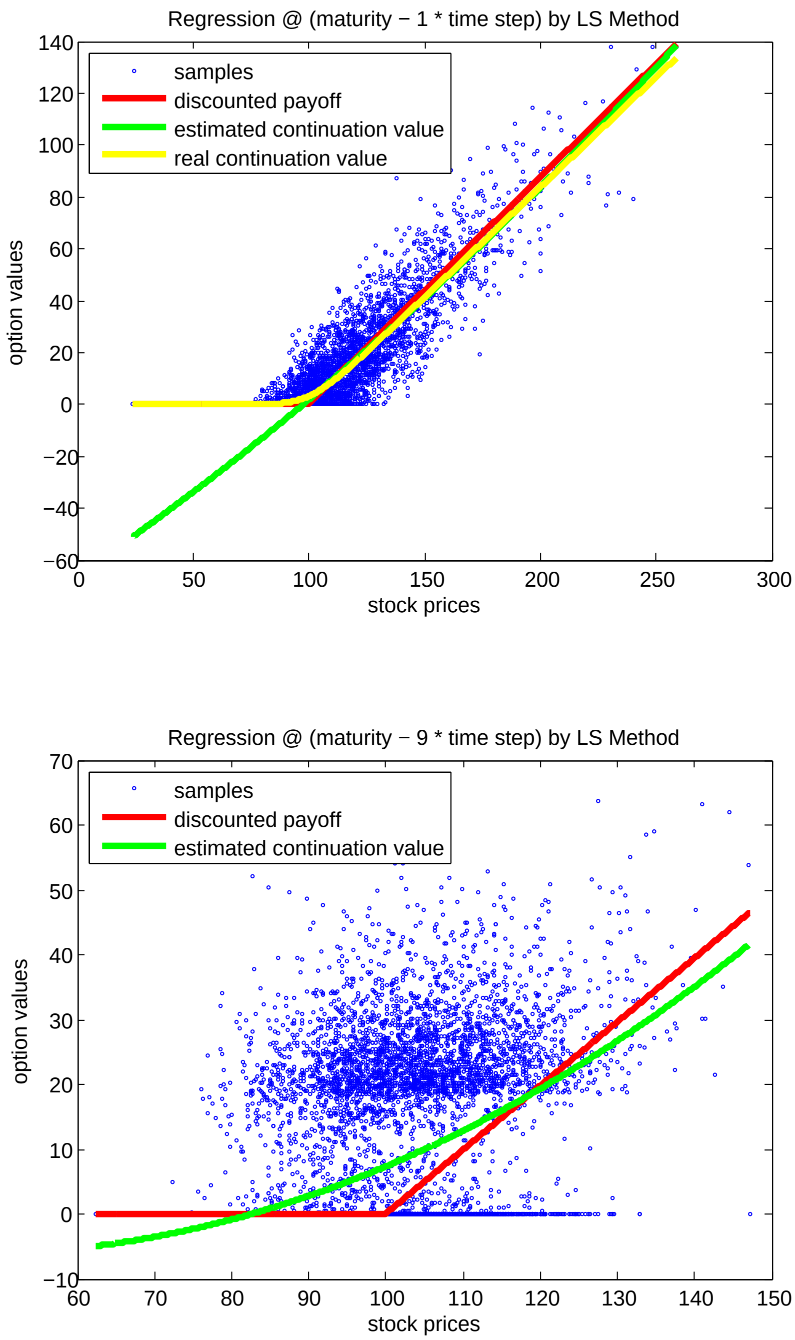


Figure 3.6: Regression using the Longstaff-Schwartz method

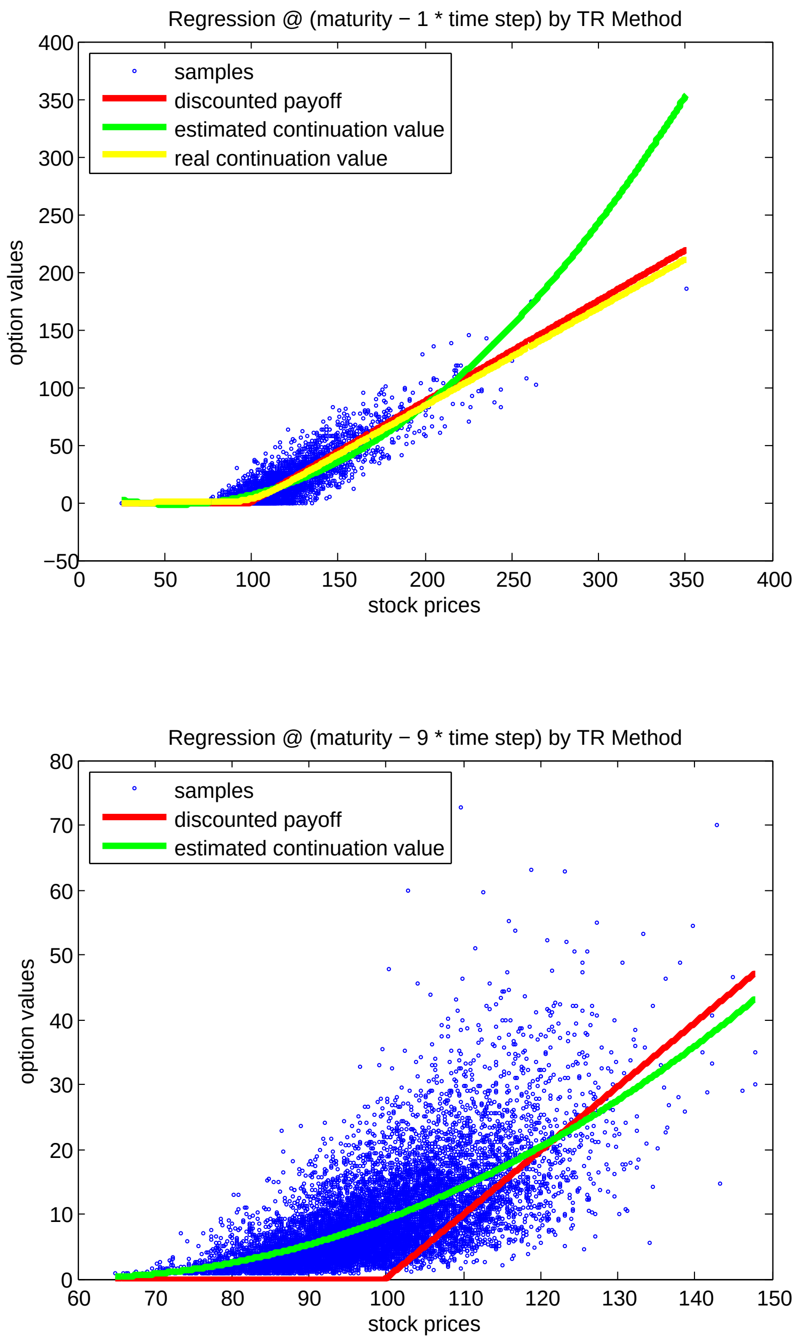


Figure 3.7: Regression using the Tsitsiklis-Roy method

3.2.5 Choice of Orthogonal Polynomials

Recall the section 3.1.3, we set up a linear regression model to estimate the conditional expectation. For simplicity, the basis functions we choose in this model is monomial polynomials. There are of course other choices, such as Legendre polynomials, Laguerre polynomials, Hermite polynomials and Chebyshev polynomials, see Table 3.11. In this section, we study the impact of choice of different polynomials as basis functions on American option prices. This section is mainly based on Abramowitz [1] and Moreno [35].

Polynomial Name	$f_n(x)$	Polynomial Name	$f_n(x)$
Monomial	$W_n(x)$	physicists' Hermite	$H_n(x)$
Legendre	$P_n(x)$	probabilists' Hermite	$H_{e_n}(x)$
Laguerre	$L_n(x)$	Chebyshev 1st kind	$T_n(x)$
		Chebyshev 2nd kind	$U_n(x)$

Table 3.11: Examples of orthogonal polynomials

Definition 3.10 (Orthogonal Polynomials). A system of polynomials $\{f_n(x)\}$ with degree $[f_n(x)] = n$ is called *orthogonal* on the interval $a \leq x \leq b$ with respect to the weight function $w(x)$ if

$$\int_a^b w(x) f_n(x) f_m(x) \mathrm{d}x = 0$$

The weight function $w(x)$ controls the system $f_n(x)$ up to a constant factor in each polynomial. The specification of these factors is referred to as standardization.

These polynomials satisfy a number of relationships of the same general form, such as *explicit expression*, *differential equation*, *recurrence relation* and *rodrigues' formula*, see Abramowitz [1] and Moreno [35].

Here we only study the recurrence relation:

$$a_{1n} f_{n+1}(x) = (a_{2n} + a_{3n} x) f_n(x) - a_{4n} f_{n-1}(x)$$

We collect all coefficients of the recurrence relation for the selected polynomials in Table 3.12.

$f_n(x)$	a_{1n}	a_{2n}	a_{3n}	a_{4n}	$f_0(x)$	$f_1(x)$
$W_n(x)$	1	0	1	0	1	x
$P_n(x)$	$n+1$	0	$2n+1$	n	1	x
$L_n(x)$	$n+1$	$2n+1$	-1	n	1	$1-x$
$H_n(x)$	1	0	2	$2n$	1	$2x$
$H_{e_n}(x)$	1	0	1	n	1	x
$T_n(x)$	1	0	2	1	1	x
$U_n(x)$	1	0	2	1	1	$2x$

Table 3.12: Recurrence relation for the selected polynomials

For example, the first few terms of the selected orthogonal polynomials are presented in Table 3.13 and drawn in Figure 3.8:

n	$P_n(x)$	$L_n(x)$
0	1	1
1	x	$-x+1$
2	$\frac{1}{2}(3x^2-1)$	$\frac{1}{2}(x^2-4x+2)$
3	$\frac{1}{2}(5x^3-3x)$	$\frac{1}{6}(-x^3+9x^2-18x+6)$
4	$\frac{1}{8}(35x^4-30x^2+3)$	$\frac{1}{24}(x^4-16x^3+72x^2-96x+24)$
5	$\frac{1}{8}(63x^5-70x^3+15x)$	$\frac{1}{120}(-x^5+25x^4-200x^3+600x^2-600x+120)$
n	$H_n(x)$	$H_{e_n}(x)$
0	1	1
1	$2x$	x
2	$4x^2-2$	x^2-1
3	$8x^3-12x$	x^3-3x
4	$16x^4-48x^2+12$	x^4-6x^2+3
5	$32x^5-160x^3+120x$	x^5-10x^3+15x
n	$T_n(x)$	$U_n(x)$
0	1	1
1	x	$2x$
2	$2x^2-1$	$4x^2-1$
3	$4x^3-3x$	$8x^3-4x$
4	$8x^4-8x^2+1$	$16x^4-12x^2+1$
5	$16x^5-20x^3+5x$	$32x^5-32x^3+6x$

Table 3.13: First few terms of the selected polynomials

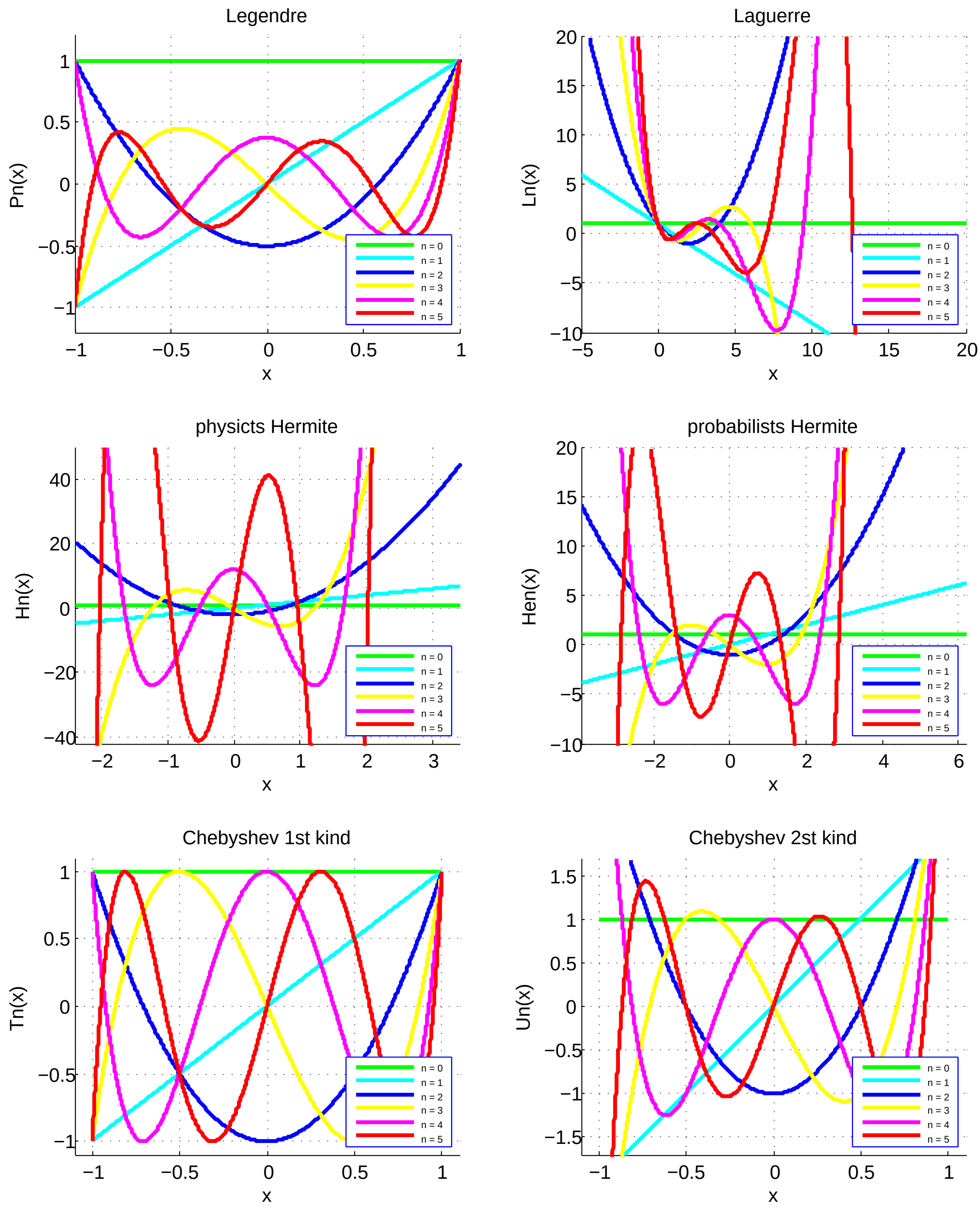


Figure 3.8: First few terms of the selected polynomials

Here we test two options, namely 7-D American geometric-average basket option in Test Case 21 and 7-D American geometric-average basket option with strangle-spread payoff in Test Case 23.

For Test Case 21 and Test Case 23 we only simulate 100000 paths due to very long run time. In each test, we consider the impact of different orthogonal polynomials on the Longstaff-Schwartz method and the Tsisiklis-Roy Method respectively. Numerical results are presented in Table 3.14 and Table 3.15. In each cell of the table, we show not only the option price, but also the corresponding standard error under the option price within the bracket.

The number of basis functions which are used are chosen from 1 to 6, which also means that the degree of polynomials n ranges from 0 to 5. For more basis functions, numerical problems might happen since solving the least-squares linear regression might involve singular matrices. For each class of polynomial, we highlight the option price which is the most close to the benchmark.

Firstly, we test the 7-D American geometric-average basket option in Table 3.14. Since it is a multi-dimensional option, we choose the average product of these seven stocks $((\prod_{i=1}^{7} S_i(T_{ex}))^{\frac{1}{7}})^n$ as the base for the basis functions, where n is polynomial degree and T_{ex} are potential exercise dates. Clearly, the Longstaff-Schwartz method delivers much more accurate result than the Tsitsiklis-Roy method, regardless of the type of the orthogonal polynomials. When n increases, the option prices calculated by the Tsitsiklis-Roy method changes heavily while the option prices calculated by the Longstaff-Schwartz method changes slightly.

On the side of the Longstaff-Schwartz method, the influence of different type of polynomials on option prices are not notable. This is not strange, since according to Abramowitz [1], the coefficients of each class of orthogonal polynomial regarding to monomial polynomial forms a non-singular matrix, which implies all class of other polynomials generate the same span as the monomial polynomial. All option prices for each class of polynomials range from 4.6763 to 4.8275 (without taking account of the case of $n = 0$) with similar standard error around 0.0215. The best choice of polynomial degree seems to be $n = 3$ or $n = 4$. However we suggest to use the Laguerre polynomial $L_n(x)$, since the option prices calculated by different degree of Laguerre polynomials range from 4.7211 to 4.7797, whose interval is the smallest among all classes of polynomials.

On the side of the Tsitsiklis-Roy method, all calculated option prices by different type and number of polynomials are larger than benchmarks and the standard error is around 0.0085, which is almost half of the standard error by the Longstaff-Schwartz method. The suggested polynomial degree is $n = 4$ or $n = 5$, which means 5 or 6 basis functions should be used. The best choice of polynomial is again the Laguerre polynomial $L_n(x)$, since in each row with the same degree n, the Laguerre polynomial always gives the smallest option price compared with other polynomials. That is also not strange, since according to Figure 3.8, Laguerre functions seem most like the payoff function of the option.

Secondly, we test the 7-D American geometric-average basket option with strangle-spread-payoff with result in Table 3.15. As before, we still use the averaged product of these stocks as the base for the basis functions. Since the payoff function is not the same as before, the best polynomial degree has also changed. On the side of the Longstaff-Schwartz method, we suggest using 3 basis functions, namely $n = 2$; on the side of the Tsitsiklis-Roy method, we suggest using 6 basis functions, namely $n = 5$. The best type of polynomial is still the Laguerre polynomials $L_n(x)$.

In summary, the Longstaff-Schwartz method is quite robust with respect to the type of the polynomials, while the Tsitsiklis-Roy method is not. The suggested number of polynomial degree n can range from 3 to 5. The suggested type of polynomial is the Laguerre polynomial.

Test Case 21, Benchmark: 4.77							
Longstaff-Schwartz Method							
n	$W_n(x)$	$P_n(x)$	$L_n(x)$	$H_n(x)$	$H_{e_n}(x)$	$T_n(x)$	$U_n(x)$
0	4.6204 (0.0198)	4.6011 (0.0196)	4.6670 (0.0198)	4.6456 (0.0197)	4.6771 (0.0198)	4.6447 (0.0198)	4.6408 (0.0199)
1	4.7155 (0.0234)	4.7004 (0.0234)	4.7211 (0.0231)	4.7155 (0.0231)	4.7079 (0.0233)	4.7076 (0.0233)	4.6763 (0.0231)
2	4.7149 (0.0219)	4.7197 (0.0217)	4.7797 (0.0223)	4.8275 (0.0221)	4.7203 (0.0218)	4.7286 (0.0218)	4.7479 (0.0218)
3	4.7033 (0.0213)	**4.7808** (0.0216)	**4.7698** (0.0216)	4.7856 (0.0214)	**4.7648** (0.0215)	**4.7813** (0.0218)	**4.7625** (0.0215)
4	**4.7710** (0.0216)	4.8101 (0.0218)	4.7590 (0.0213)	**4.7611** (0.0217)	4.7891 (0.0218)	4.7404 (0.0215)	4.7465 (0.0216)
5	4.7577 (0.0216)	4.7313 (0.0217)	4.7577 (0.0213)	4.7337 (0.0216)	4.8034 (0.0219)	4.7592 (0.0218)	4.7412 (0.0214)
Tsitsiklis-Roy Method							
n	$W_n(x)$	$P_n(x)$	$L_n(x)$	$H_n(x)$	$H_{e_n}(x)$	$T_n(x)$	$U_n(x)$
0	13.7221 (0.0010)	13.8161 (0.0011)	13.6866 (0.0011)	13.6761 (0.0010)	13.6707 (0.0012)	13.7580 (0.0010)	13.7670 (0.0010)
1	8.9316 (0.0092)	8.8306 (0.0092)	8.7523 (0.0092)	8.8326 (0.0093)	8.8761 (0.0092)	8.9054 (0.0093)	8.8345 (0.0092)
2	5.3555 (0.0086)	5.3601 (0.0084)	5.3193 (0.0085)	5.3121 (0.0085)	5.3766 (0.0085)	5.3747 (0.0085)	5.3447 (0.0085)
3	5.3492 (0.0084)	5.3562 (0.0084)	5.3129 (0.0083)	5.3626 (0.0085)	5.3384 (0.0083)	5.3395 (0.0084)	5.3461 (0.0084)
4	**5.1145** (0.0085)	5.1918 (0.0083)	5.1390 (0.0085)	**5.2637** (0.0085)	**5.1120** (0.0086)	5.2149 (0.0084)	5.2548 (0.0084)
5	5.1628 (0.0084)	**5.1347** (0.0085)	**5.1047** (0.0085)	5.3422 (0.0086)	5.2270 (0.0085)	**5.2119** (0.0086)	**5.1872** (0.0086)

Table 3.14: Test the effect of different choice of orthogonal polynomials on option prices in Test Case 21

Test Case 23, Benchmark: 8.31							
Longstaff-Schwartz Method							
n	$W_n(x)$	$P_n(x)$	$L_n(x)$	$H_n(x)$	$H_{e_n}(x)$	$T_n(x)$	$U_n(x)$
0	8.3631 (0.0097)	8.3852 (0.0096)	8.3647 (0.0097)	8.3733 (0.0096)	8.3707 (0.0096)	8.3631 (0.0097)	8.3697 (0.0096)
1	8.3954 (0.0094)	8.4168 (0.0094)	8.3900 (0.0094)	8.4066 (0.0094)	8.4093 (0.0094)	**8.3849** (0.0094)	8.4044 (0.0094)
2	8.4108 (0.0092)	**8.3994** (0.0092)	**8.3771** (0.0093)	**8.3894** (0.0093)	**8.3814** (0.0093)	8.3949 (0.0093)	**8.3905** (0.0093)
3	8.4041 (0.0092)	8.4067 (0.0093)	8.4270 (0.0092)	8.4137 (0.0092)	8.3927 (0.0093)	8.3948 (0.0093)	8.4246 (0.0092)
4	**8.3992** (0.0091)	8.4020 (0.0092)	8.4119 (0.0091)	8.4047 (0.0092)	8.3962 (0.0091)	8.4124 (0.0092)	8.4086 (0.0092)
5	8.4188 (0.0091)	8.4015 (0.0092)	8.3859 (0.0092)	8.3899 (0.0092)	8.3986 (0.0091)	8.4022 (0.0092)	8.3964 (0.0092)
Tsitsiklis-Roy Method							
n	$W_n(x)$	$P_n(x)$	$L_n(x)$	$H_n(x)$	$H_{e_n}(x)$	$T_n(x)$	$U_n(x)$
0	9.8361 (0.0001)	9.8370 (0.0001)	9.8359 (0.0001)	9.8363 (0.0001)	9.8363 (0.0001)	9.8364 (0.0001)	9.8359 (0.0001)
1	9.6687 (0.0001)	9.6727 (0.0003)	9.6750 (0.0003)	9.6728 (0.0003)	9.6719 (0.0003)	9.6692 (0.0003)	9.6716 (0.0003)
2	9.2864 (0.0013)	9.2900 (0.0013)	9.2805 (0.0014)	9.2924 (0.0013)	9.2943 0.0013	9.2771 (0.0013)	9.2800 (0.0013)
3	9.1456 (0.0012)	9.1678 (0.0012)	9.1501 (0.0012)	9.1553 (0.0012)	9.1507 (0.0012)	9.1426 (0.0012)	9.1590 (0.0011)
4	9.0470 (0.0014)	9.1789 (0.0013)	**8.9218** (0.0014)	9.1861 (0.0014)	9.0453 (0.0013)	9.1858 (0.0013)	9.1760 (0.0014)
5	**9.0078** (0.0013)	**9.0710** (0.0014)	8.9407 (0.0014)	**9.0759** (0.0013)	**9.0156** (0.0013)	**9.0787** (0.0014)	**9.0738** (0.0014)

Table 3.15: Test the effect of different choice of orthogonal polynomials on option prices in Test Case 23

3.2.6 Lower Bound vs Upper Bound

In this section, we use the Andersen-Broadie method (Algorithm 3.6) to yield the upper bound for the Longstaff-Schwartz method and the Tsitsiklis-Roy method. The corresponding lower bounds are computed by the modified Longstaff-Schwartz method (Algorithm 3.4) and the modified Tsitsiklis-Roy method (Algorithm 3.5). The difference between the upper bound and the lower bound shows the accuracy of different algorithms when pricing American options.

First we consider a toy exmple – a Bermudan put option on a single stock with only two exercise date $(t_1 = \frac{T}{2}, t_1 = T)$ with all input parameters shown in Test Case 3. This example is also examined on page 254-255 of Korn [26]. They claimed that if we make use of monomial polynomials up to degree 3 as basis functions $(1, S, S^2, S^3)$, the variation of Longstaff-Schwartz method with all paths doing regression (studied in the section 3.2.3) gives very bad lower bounds but acceptable upper bounds. However we drew different conclusions: the lower bounds and upper bounds are both acceptable even in this case.

We use 100000 paths to approximate the regression coefficients which can determine the exercise strategy. For the lower bound, we use again 100000 paths as out-of-samples. For the upper bound, we simulate $N_1 = 1000$ paths as out-of-samples and $N_2 = 1000$ subpaths at t_1 to compute the inner conditional expectations.

The option prices were respectively computed for 100 times via the Longstaff-Schwartz method with all paths doing regression ("LSAll") using monomial polynomials up to degree 3, via its lower bound ("LSALLLower") and via its upper bound ("LSAllUpper"). The benchmark value is 4.313. We show the median of these 100 option prices, the corresponding standard error and the 95% - confidence interval in Table 3.16 and produce a box plot for them in Figure 3.9. From the Table and Figure, we see clearly that the "LSAll" method with monomial polynomials up to degree 3 delievers very good result. We also notice one intersting thing which is the standard error for the "LSAllUpper" method is much smaller than the "LSAll" and the "LSAllLower" methods.

We keep on using the same example - Test Case 3 with same paths numbers, same subpath numbers and same basis functions as before to test the original Longstaff-Schwartz method using only in-the-money paths doing regression ("LSITM") (Algorithm 3.2), its variation with lower bound ("LSITMLower") and its variation with upper bound ("LSITMUpper") and the original Tsitsiklis-Roy method (Algorithm 3.3) using all paths doing regression ("TRAll"), its variation with lower bound ("TRAllLower") and its variation with upper bound ("TRAllUpper"). The simulations run again for 100 times. The box-plot for the 100 option prices are presented in Figure 3.10. From this figure, we notice that the Longstaff-Schwartz method either using in-the-money paths or using all paths approximating regression coefficients delivers good results and the difference between the lower bounds and the upper bounds for both methods is very small. However, the result by the Tsitsiklis-Roy method is not very accurate.

Test Case 3, Benchmark: 4.313			
Name	Option Price	Standard Error	95% - Confidence Interval
LSAll	4.3089	0.0206	[4.2685, 4.3493]
LSAllLower	4.3108	0.0207	[4.2702, 4.3514]
LSAllUpper	4.3138	0.0067	[4.3007, 4.3269]

Table 3.16: Lower and upper bounds for the option price using "LSAll" method with monomial polynomials up to degree 3

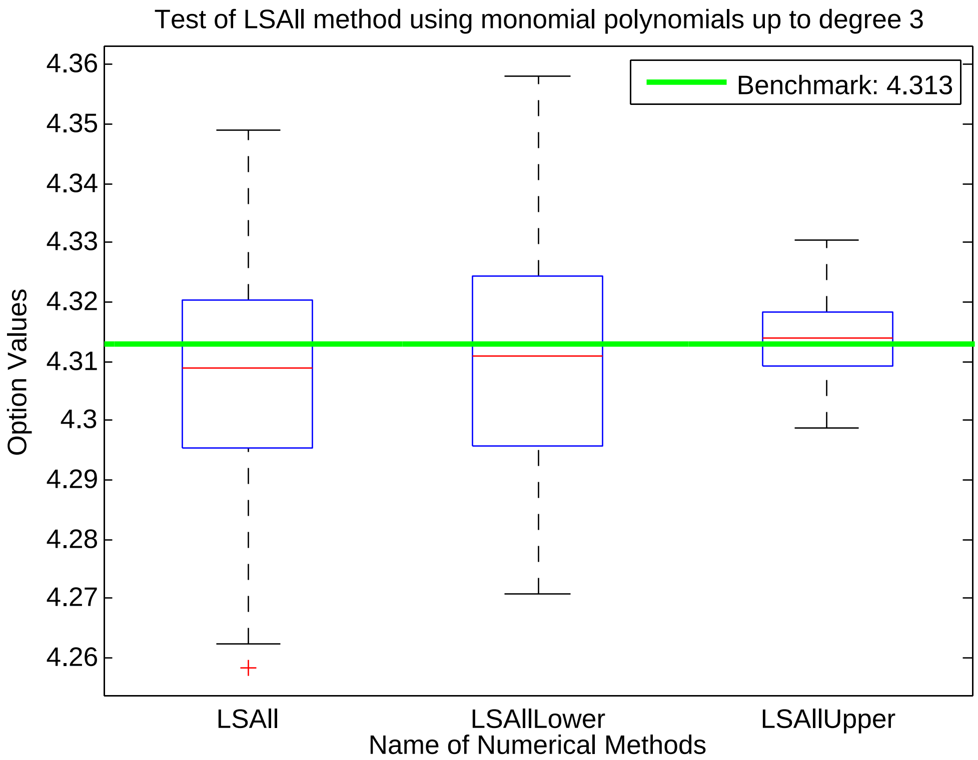


Figure 3.9: Lower and upper bounds for the option price in Test Case 3 using "LSAll" method with monomial polynomials up to degree 3

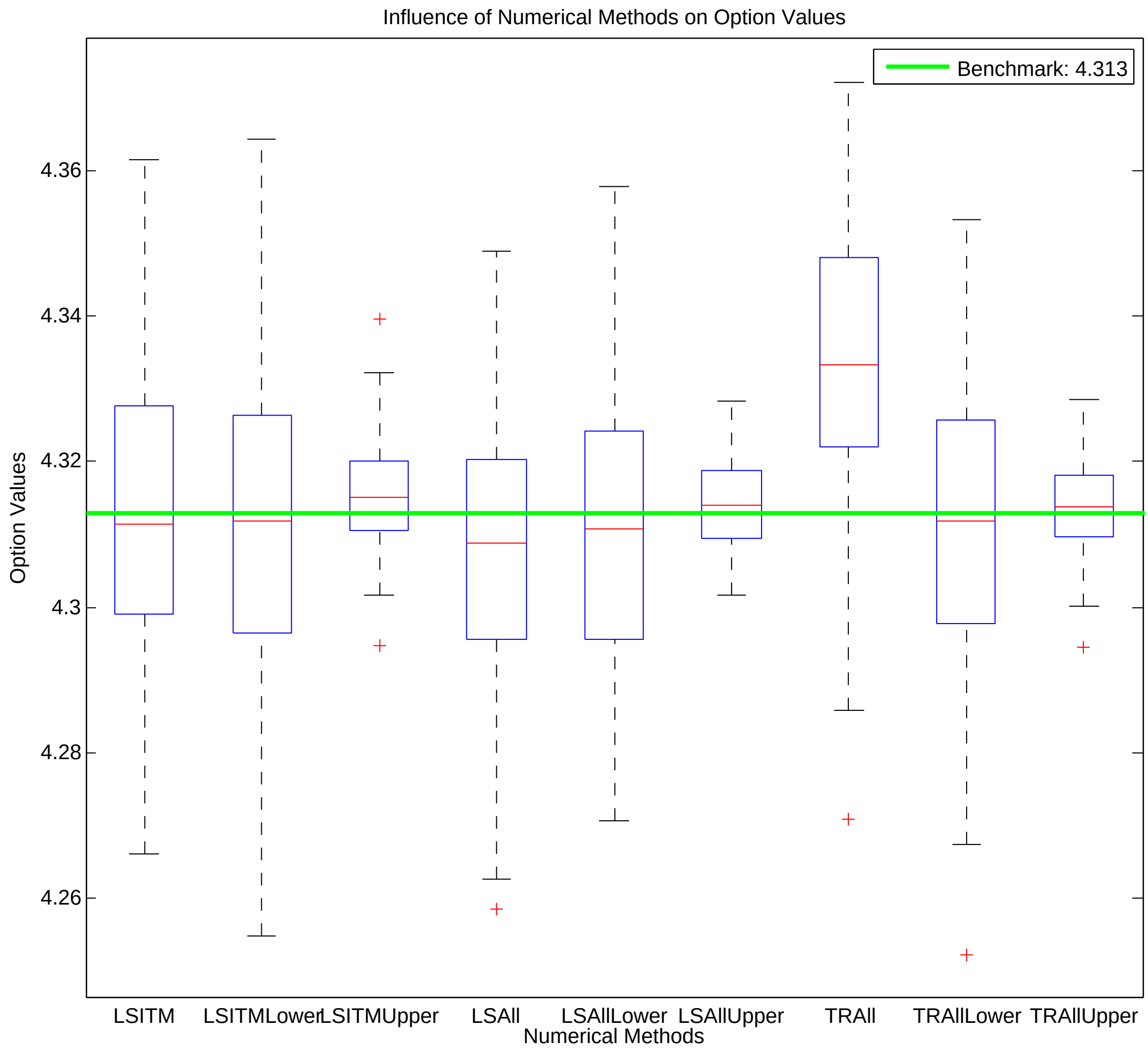


Figure 3.10: Lower and upper bounds for the option price in Test Case 3 using "LSITM", "LSAll" and "TRAll" methods

4 Improvement of the Regression Part by Machine Learning Techniques

Notice that the Longstaff-Schwartz method (Algorithm 3.2) and the Tsitsiklis-Roy method (Algorithm 3.3) use least-squares linear regression to estimate the continuation value (equation 3.7). According to Kohler [22], there are four paradigms of nonparametric regression to estimate: local averaging, local modeling, global modeling (or least squares estimation) and penalized modeling. While they are studied theoretically with focus on consistency or convergence rate in the book of Györfi [17], they are also overall studied practically as machine learning techniques in the book of Bishop [6].

Consider X_i to be the observation data, Y_i to be the value of the regression function $m(x)$ at X_i, $\epsilon_i = Y_i - m(X_i)$ to be the error, where the expectation of the error $\mathbb{E}(\epsilon|X_i)$ is equal to 0:

$$Y_i = m(X_i) + \epsilon_i \quad i = 1, \ldots, n \tag{4.1}$$

The idea of the **local averaging** is to estimate $m(x)$ by the average of those Y_i where X_i is close to x. The corresponding estimate is:

$$m_n(x) = \sum_{i=1}^{n} W_{n,i}(x) \cdot Y_i \tag{4.2}$$

with weights $W_{n,i}(x) = W_{n,i}(x, X_1, \ldots, X_n) \in R$ depending on $X_1, \ldots, X_n$. Some popular choices of local averaging are *partitioning estimate*, *Nadaraya-Watson kernel estimate* and *k-nearst neighbor estimate*.

The idea of the **local modeling** is to fit each data with a general function depending on several parameters. Define $g(\cdot, \{a_k\}_{k=1}^{l}) : \mathcal{R}^d \rightarrow \mathcal{R}$ as a function depending on parameters $\{a_k\}$. For each $x \in \mathcal{R}^d$, the optimal local parameters are obtained by:

$$\{a_k^*(x)\} = \arg\min_{\{a_k\}} \frac{1}{n} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right) (Y_i - g(X_i, \{a_k\}))^2 \tag{4.3}$$

with $K : \mathcal{R}^d \longrightarrow R^+$ being kernel function, where the weight of Y_i depends on the distance between X_i and x, and $h > 0$ being bandwidth. The estimate of $m(x)$ is then obtained as:

$$m_n(x) = g(x, \{a_k^*(x)\})$$

The most popular choice of local modeling is *local polynomial kernel estimate*.

The idea of the **global modeling** is to choose a function space $\mathcal{F}_n$ with functions $f \in \mathcal{F}_n : \mathcal{R}^d \longrightarrow \mathcal{R}$. The estimate is defined as:

$$m_n(\cdot) = \arg\min_{f \in \mathcal{F}_n} \left\{ \frac{1}{n} \sum_{i=1}^{n} |f(X_i) - Y_i|^2 \right\} \tag{4.4}$$

If $\mathcal{F}_n$ is a linear vector space, the minimum can be obtained by solving a linear equation system, which is indeed solved in the Longstaff-Schwartz Method and the Tsitsiklis-Roy Method, see the equation 3.13. If $\mathcal{F}_n$ is a nonlinear vector space, the most popular choice is *neural networks*.

The idea of the **penalized modeling** is to add a penalty term $J_n(f) \geq 0$ to penalize the "roughness" of a function f. The corresponding estimate is defined as:

$$m_n(\cdot) = \arg\min_{f \in \mathcal{F}_n} \left\{ \frac{1}{n} \sum_{i=1}^{n} |f(X_i) - Y_i|^2 + J_n(f) \right\} \tag{4.5}$$

The most popular choice is *smoothing spline estimates*.

Each estimate above contains a parameter, which can control the smoothness of the estimate. In order to use these estimates efficiently, we have to choose proper parameter, which should be data-dependent. The chosen process can be proceeded either using *splitting of the sample* or *cross validation* (see Chapter 7 and 8 in Györfi [17]).

Egloff [13] was the first one who used nonparametric regression with least squares estimation to approximate continuation value of American option. He examined rate of convergence for smooth continuation value function. However his estimate is too hard to implement in practice. Egloff, Kohler and Todorovic [14] used linear vector space for the least squares spline estimates and make the implementation much easier than before, for this estimate can be solved by a linear equation system. Again consistency and rate of convergence was derived in this paper. Kohler [24] investigated smoothing spline estimates and Kohler, Krzyzak and Todorovic [23] considered least squares neural network estimates. Both papers also showed proof for consistency and rate of convergence.

Lee [29] [30] investigated the numerical performance of the kernel method to price American option within the Black-Scholes model and a jump-diffusion model. However his papers didn't give the input of the bandwidth for the kernel estimate, which is crucial for the option price, since it determines the smoothness of the estimation function. Second, the path numbers of Monte Carlo simulation were not given in his paper, thus the corresponding confidence intervals were also not clear. Third, for the kernel method only in-samples paths were simulated to determine the optimal exercise strategy, no additional out-of-samples paths were generated to give results with low-bias. Thus their results were mixed with high bias and low bias. Based on Lee's previous work, we improve the least squares linear regression part of the Longstaff-Schwartz method and the Tsitsiklis-Roy method by the kernel method and the support vector machines. The content is mainly based on Kohler [22], Lee [29] [30], Todorovic [42], Györfi [17] and Bishop [6].

4.1 Kernel Methods

4.1.1 Fixed Bandwidth

The Nadaraya-Watson kernel method is the most popular choice of local averaging (equation (4.2)). The kernel estimate takes the form as:

$$m_n(x) \quad = \quad \sum_{i=1}^{n} \frac{K(\frac{x-X_i}{h_n})}{\sum_{j=1}^{n} K(\frac{x-X_j}{h_n})} Y_i \tag{4.6}$$

where $h_n > 0$ is the bandwidth and depends on the sample size n and $K : \mathcal{R}^d \longrightarrow \mathcal{R}$ is a kernel function. If $||x||$ is smaller, usually $K(x)$ is large. Typical choices of a kernel function are: naive kernel ($K(x) = \mathbb{1}_{||x|| \leq 1}$), Epanechnikov kernel ($K(x) = (1 - ||x||^2)^+$) and Gaussian kernel $K(x) = \exp(-||x||^2/2)$, see Figure 4.1.

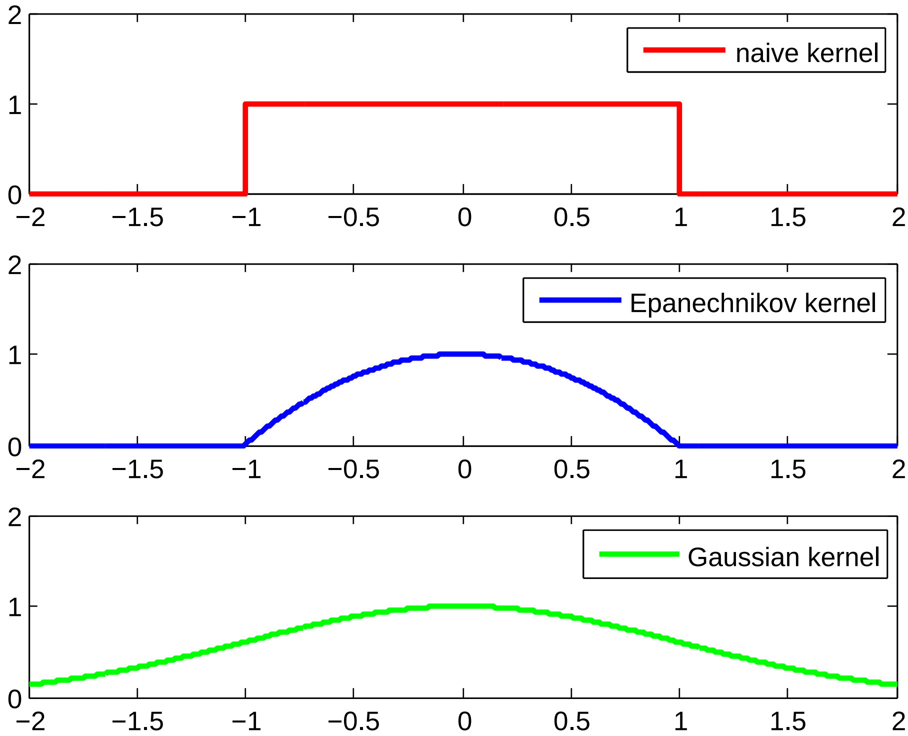


Figure 4.1: Examples of kernel functions

Recall the continuation value function using least-squares linear regression (see equations (3.13), (3.14) and (3.15)):

$$
\begin{aligned}
C^*(S(t_i)) &= \sum_{l=1}^{k} a_l^* H_l(S(t_i)) \\
&= H(S(t_i))^\top a^* \\
&= H(S(t_i))^\intercal (X^\top X)^{-1} X^\top Y
\end{aligned}
$$

where $t_i = t_1, \ldots, t_m$ are potential exercise dates. Thus the continuation value for each path $j = 1, \ldots, N$ is:

$$
\begin{aligned}
C^*(S^{(j)}(t_i)) &= H(S^{(j)}(t_i))^\intercal (X^\intercal X)^{-1} X^\intercal Y \\
&= H(S^{(j)}(t_i))^\intercal (X^\intercal X)^{-1} \sum_{k=1}^{N} H(S^{(k)}(t_i)) V(S^{(k)}(t_i)) \\
&= \frac{1}{N} \sum_{k=1}^{N} H(S^{(j)}(t_i))^\intercal (\frac{1}{N} X^\intercal X)^{-1} H(S^{(k)}(t_i)) V(S^{(k)}(t_i)) \\
&= \frac{1}{N} \sum_{k=1}^{N} K(S^{(j)}(t_i), S^{(k)}(t_i)) V(S^{(k)}(t_i)) \qquad (4.7)
\end{aligned}
$$

where we define a kernel function K as:

$$
\begin{aligned}
K(S^{(j)}(t_i), S^{(k)}(t_i)) &= H(S^{(j)}(t_i))^\intercal (\frac{1}{N} X^\intercal X)^{-1} H(S^{(k)}(t_i)) \\
&= H(S^{(j)}(t_i))^\intercal \left[\frac{1}{N} \sum_{l=1}^{N} H(S^{(l)}(t_i)) H(S^{(l)}(t_i))^\intercal \right]^{-1} H(S^{(k)}(t_i))
\end{aligned}
$$

Thus, we see that the solution to the least-squares linear regression problem in the Longstaff-Schwartz method or in the Tsitsiklis-Roy method can be entirely expressed in terms of the kernel function using basis functions.

However we can choose another kernel function instead of the kernel above. Here we apply Gaussian kernel with bandwidth h to estimate the continuation value function of an American option. The kernel function is:

$$
K_{Gaus}(S^{(j)}(t_i), S^{(k)}(t_i)) = \exp\left(-\frac{||S^{(j)}(t_i) - S^{(k)}(t_i)||^2}{2h^2} \right) \qquad (4.8)
$$

where $||\cdot||$ is the Euclidean norm, $||x|| = \sqrt{x_1^2 + x_2^2 + \ldots, x_n^2}$, where $x = (x_1, x_2, \ldots, x_n)$.

The corresponding kernel estimate for the continuation value $C(S^{(j)}(t_i))$ for the $j - th$ path at exercise date t_i is:

$$
C^*(S^{(j)}(t_i)) = \sum_{k=1}^{N} \frac{K_{Gaus}(S^{(j)}(t_i), S^{(k)}(t_i))}{\sum_{l=1}^{N} K_{Gaus}(S^{(j)}(t_i), S^{(l)}(t_i))} V(S^{(k)}(t_i)) \qquad (4.9)
$$

The kernel method by Lee [29] using Gaussian kernel with bandwith h is presented in Algorithm 4.1. However this method uses same paths to estimate the continuation value and to compute the option vale. Thus it mixes the low and high bias. Its modified version using out-of-samples is presented in Algorithm 4.2, which only gives low bias and hence can be used to compare with algorithms.

Lee's kernel method performs well when suitable bandwidth h is chosen, see Figure 4.2. The blue circles are samples consisting of stock prices at potential exercise date t_{m-1} as x-coordinate and corresponding option values assuming that the option has not been exercised before t_{m-1} as y-coordinate. The red curve is the discounted payoff function. The yellow curve is the real continuation value. The green curve is the estimated continuation value by the Longstaff-Schwartz method and the cyan curve is the estimated continuation value by Lee's kernel method. We notice that the cyan curve is closer to the yellow curve than the green curve, especially within the in-the-money area.

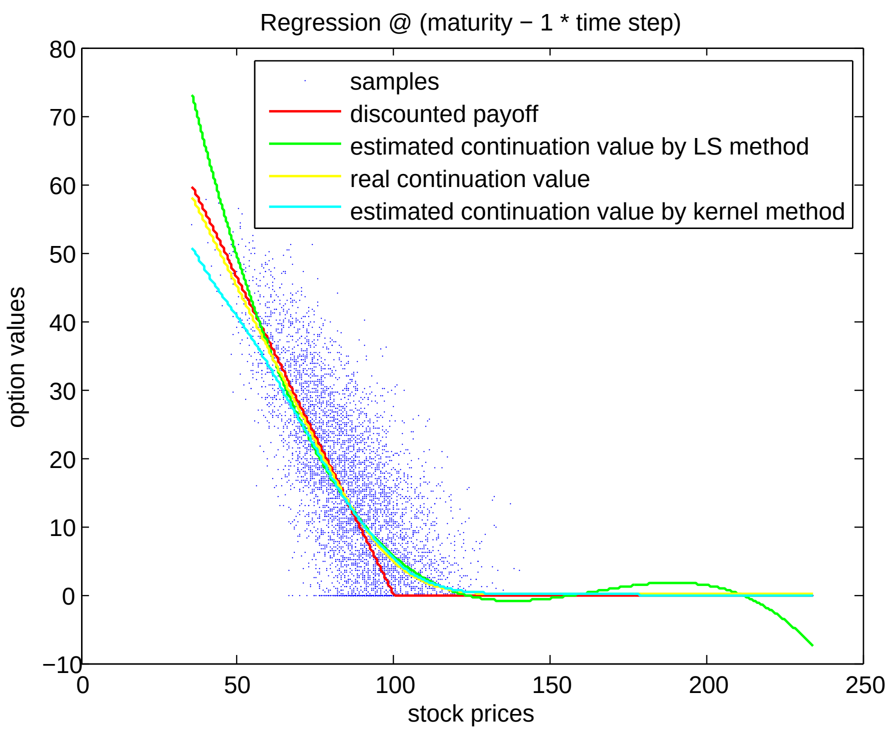


Figure 4.2: Performance of Lee's kernel method

Algorithm 4.1 Lee's kernel method

1. Generate N independent paths for stock at all possible exercise dates: $\{S^{(n)}(t_1), S^{(n)}(t_2), \ldots, S^{(n)}(t_m)\}$ with $n = 1, \ldots, N$, $t_i = \frac{T}{m} \times i$, $i = 1, \ldots, m$.

2. At maturity $t_m = T$, fix the discounted terminal values of the American option for each path $n = 1, \ldots, N$: $V(S^{(n)}(t_m)) = g(S^n(t_m))$.

3. Compute backward at each potential exercise date t_i for $i = m-1, \ldots, 1$:

   1) Calculate the estimated continuation value $C^*(S^{(n)}(t_i))$ and the discounted exercising value $g(S^{(n)}(t_i))$ for each path:

   $$C^*(S^{(n)}(t_i)) = \sum_{k=1}^{N} \frac{K_{Gaus}(S^{(n)}(t_i), S^{(k)}(t_i))}{\sum\limits_{l=1}^{N} K_{Gaus}(S^{(n)}(t_i), S^{(l)}(t_i))} V(S^{(k)}(t_i))$$

   2) Compare $C^*(S^{(n)}(t_i))$ and $g(S^{(n)}(t_i))$ to decide whether to exercise or to continue the option:

   $$V(S^{(n)}(t_i)) = \begin{cases} g(S^{(n)}(t_i)), & g(S^{(n)}(t_i)) > C^*(S^{(n)}(t_i)) \quad \&\& \quad g(S^{(n)}(t_i)) > 0 \\ C^*(S^{(n)}(t_i)), & \text{otherwise} \end{cases}$$

4. Compute $V_k^N(S(t_0)) = \left(\frac{1}{N} \sum\limits_{n=1}^{N} V(S^{(n)}(t_1)) \right)$ as the American option price.

Algorithm 4.2 Modified Lee's kernel method with low bias

- Step 1 - Step 3.2: Same as in Lee's kernel method (Algorithm 4.1). Save $S^{(n)}(t_i)$ and $V(S^{(n)}(t_i))$ as $S^{(n)}_{\text{old}}(t_i)$ and $V(S^{(n)}_{\text{old}}(t_i))$ for $n = 1, \ldots, N$, $n = 1, \ldots, m$.
- Step 4: Regenerate N_{new} new independent paths for stock at all potential exercise dates: $\left\{S^{(n)}(t_1), S^{(n)}(t_2), \ldots, S^{(n)}(t_n)\right\}$ with $n = 1, \ldots, N_{\text{new}}$.
- Step 5: Define the stopping rule $\tau^{(n)} = t_1$ for each path $n = 1, \ldots, N_{\text{new}}$ and compute forward at t_i for $i = 1, \ldots, m$:
  1) Calculate the estimated continuation value $C^*(S^{(n)}(t_i))$ and the discounted exercising value $g(S^{(n)}(t_i))$ for each path $n = 1, \ldots, N_{\text{new}}$:
  $$C^*(S^{(n)}(t_i)) = \sum_{k=1}^{N} \frac{K_{Gaus}(S^{(n)}(t_i), S^{(k)}_{\text{old}}(t_i))}{\sum_{l=1}^{N} K_{Gaus}(S^{(n)}(t_i), S^{(l)}_{\text{old}}(t_i))} V(S^{(k)}_{\text{old}}(t_i))$$
  2) If $\tau^{(n)} = t_1$ and $g(S^{(n)}(t_i)) > 0$ and $g(S^{(n)}(t_i)) > C^*(S^{(n)}(t_i))$: exercise the option at t_i, set $\tau^{(n)} = t_i$ and $V_{\text{new}}(S^{(n)}(t_1)) = g(S^{(n)}(t_i))$, stop;
  Else if $t_i < t_{m-1}$: continue the option at t_i;
  Else: exercise the option at t_m and set $\tau^{(n)} = t_m$ and $V_{\text{new}}(S^{(n)}(t_1)) = g(S^{(n)}(t_m))$, stop.
- Step 6: Compute $V_k^{N_{\text{new}}}(S(t_0)) = \left(\frac{1}{N_{\text{new}}} \sum_{n=1}^{N_{\text{new}}} V_{\text{new}}(S^{(n)}(t_1))\right)$ as the American option price.

4.1.2 Global Optimal Bandwidth

However whether the bandwidth h is given as input or it is chosen as data-dependent is unknown in his paper. If h is too small, the undersmoothing problem might happen, see the first plot of Figure 4.3; if h is too big, the oversmoothing problem might occur, see the third plot. The optimal data-dependent (here we simulate 1000 paths of stock prices) bandwidth is 3.6598, see the second plot.

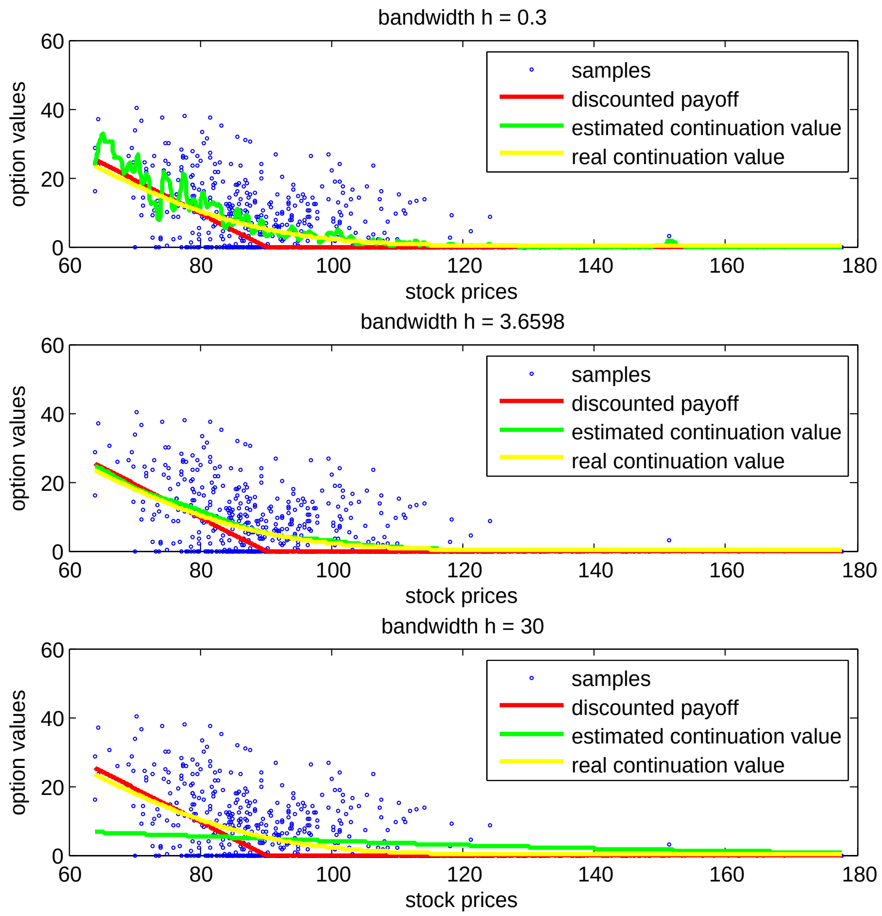


Figure 4.3: Effect of bandwidth for kernel method

Here we obtain the optimal bandwidth by splitting the sample. According to Györfi [17] and Kohler [22], the idea of splitting the sample is that the sample is divided artificially into two parts, the first part is called the training data set, the second part is called the testing data set. The training data set is used to compute the estimate for different smoothing parameters. The testing data set is used to compute the error of each of these estimates. The optimal estimate is obtained by minimizing the error. The kernel method with optimal bandwidth by splitting the sample is presented in Algorithm 4.3 and its modified version using new paths with low bias is shown in Algorithm 4.4. Here we make use of the *Global Search* class of MATLAB, along with the *run* method and the *interior-point* algorithm to find the global minimum for the bandwidth.

Take Test Case 2: 1-D Bermudan option with 12 potential exercise dates as an example for the global search of optimal bandwidth. Figure 4.4 shows the mean squared error (L_2 risk) against the bandwidth at some exercise date. All optimal bandwidths at $11 (= 12 - 1)$ exercise dates are collected in Table 4.1.

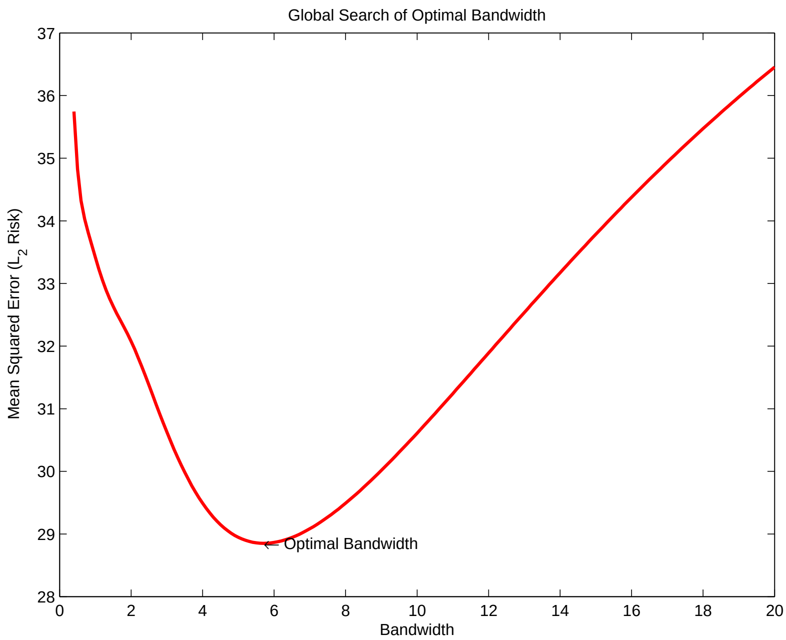


Figure 4.4: Global search of optimal bandwidth for the kernel method

Results for Test Case 2			
Exercise Date	Optimal Bandwidth	Exercise Date	Optimal Bandwidth
t_1	1.0354	t_7	1.2364
t_2	4.7922	t_8	2.5405
t_3	1.8282	t_9	4.0521
t_4	4.4152	t_{10}	4.2550
t_5	1.2956	t_{11}	1.7357
t_6	2.1157		

Table 4.1: Optimal bandwidths for potential exercise dates for Test Case 2

Since the modified Longstaff-Schwartz method (Algorithm 3.4), the modified Tsitsiklis-Roy method (Algorithm 3.5), the modified Lee's kernel method (Algorithm 4.2) and the modified kernel method with optimal bandwidth (Algorithm

4.4) evaluate the approximative optimal stopping rule via newly generated paths, all these four algorithms provide lower bounds. Thus a higher option price implies a better performance of the algorithm.

First we compare Algorithm 4.2 and Algorithm 4.3. We simulate $N = 1000$ paths of stock prices to obtain the kernel estimate. For Algorithm 4.2, we test bandwidth $h = 1, 5, 10$ respectively. For Algorithm 4.3, $N_{\text{train}} = 500$ is used for training the optimal bandwidth and $N_{\text{test}} = 500$ is used for testing the optimal bandwidth. Based on these kernel estimates, we generate $N_{\text{New}} = 4000$ new paths and compute the option prices.

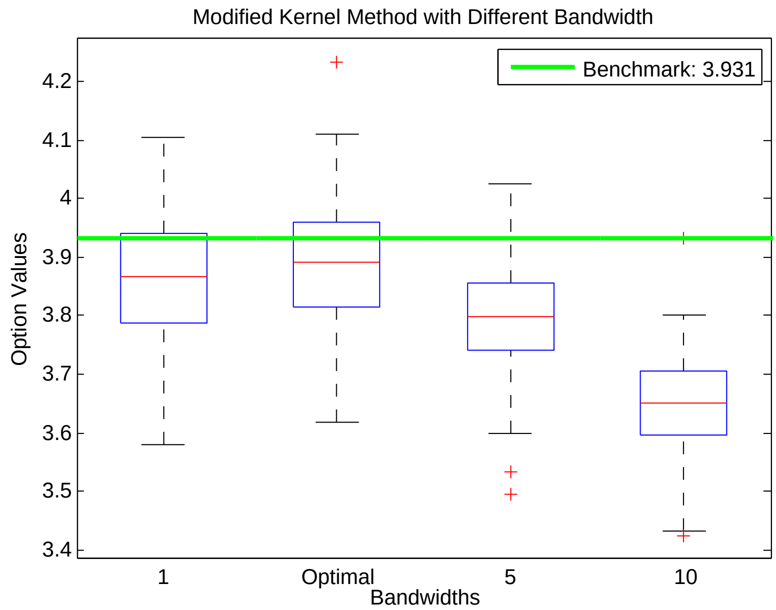


Figure 4.5: Comparison of the modified Lee's kernel method with different bandwidth and the one with optimal bandwidth for Test Case 2: 1-D Bermudan option with 12 potential exercise dates

We run 100 independent Monte Carlo simulations for each test case of bandwidth and generate the box-plot for these 4×100 option prices in Figure 4.5. In the box-plot the median is shown as a red line across the box and the box stretches from the 25th percentile to the 75th percentile. We notice that the modified kernel method with optimal bandwidth performs the best. However it also has a shortcoming, that is, it is very time consuming when searching globally for the optimal bandwidth, especially for large sample size N. That is also the reason that we only simulate $N = 1000$ paths here.

Algorithm 4.3 Kernel method with optimal bandwidth

1. Generate N independent paths for stock at all possible exercise dates: $\{S^{(n)}(t_1), S^{(n)}(t_2), \ldots, S^{(n)}(t_m)\}$ with $n = 1, \ldots, N$, $t_i = \frac{T}{m} \times i$, $i = 1, \ldots, m$.

2. At maturity $t_m = T$, fix the discounted terminal values of the American option for each path $n = 1, \ldots, N$: $V(S^{(n)}(t_m)) = g(S^n(t_m))$.

3. Compute backward at each potential exercise date t_i for $i = m-1, \ldots, 1$:

   1) Define $D_N = \{(S^{(1)}(t_i), V(S^{(1)}(t_i))), \ldots, (S^{(N)}(t_i), V(S^{(N)}(t_i)))\}$. Set the initial bandwidth as h_0. Randomly sample 50% of D_N as training data set D_{Train}, put the remaining part into the testing data set D_{Test}.

   2) Use the training data set D_{Train} and the bandwidth h, build a kernel estimate (eq. 4.9). Based on this estimate, predict the outputs $C^h(S^{(n)}(t_i))$ of the testing data set D_{Test}.

   3) Compare the predicted output $C^h(S^{(n)}(t_i))$ and the actual output $V(S^{(n)}(t_i))$ of D_{Test} and find the best bandwidth h^* to minimize the L_2 risk (mean squared error) between the predicted and actual outputs.

   4) Use this optimal bandwidth h^* and calculate the estimated continuation value $C^*(S^{(n)}(t_i))$ and the discounted exercising value $g(S^{(n)}(t_i))$ for each path of D_N:

$$C^*(S^{(n)}(t_i)) = \sum_{k=1}^{N} \frac{K_{Gaus}(S^{(n)}(t_i), S^{(k)}(t_i))}{\sum_{l=1}^{N} K_{Gaus}(S^{(n)}(t_i), S^{(l)}(t_i))} V(S^{(k)}(t_i))$$

   5) Compare $C^*(S^{(n)}(t_i))$ and $g(S^{(n)}(t_i))$ to decide whether to exercise or to continue the option:

$$V(S^{(n)}(t_i)) = \begin{cases} g(S^{(n)}(t_i)), & g(S^{(n)}(t_i)) > C^*(S^{(n)}(t_i)) \quad \&\& \quad g(S^{(n)}(t_i)) > 0 \\ C^*(S^{(n)}(t_i)), & \text{otherwise} \end{cases}$$

4. Compute $V_k^N(S(t_0)) = \left(\frac{1}{N}\sum_{n=1}^{N} V(S^{(n)}(t_1))\right)$ for the American option price.

Algorithm 4.4 Modified kernel method using optimal bandwidth with low bias

1. Step 1 - Step 3.5: Same as Algorithm 4.3.

2. Step 4 - Step 6: Same as Algorithm 4.2.

4.1.3 Scaling, Parameter Selection and Suboptimal Bandwidth

We now take a closer look at the Gaussion kernel function (equation (4.8)), the kernel values depends on the Euclidean distance between two stocks $S^{(j)}(t_i)$ and $S^{(k)}(t_i)$. If we simulate a lot of stocks, we are likely to find that the distance between certain two stock prices is very large, which leads to that the kernel values is very close to 0. When we calculate the continuation value $C(S^{(j)}(t_i))$ via the formula (4.9), the sum of all kernel values lies in the denominator. If this denominator is too small, we might meet numerical difficulty. One way to avoid this is to scale the stock prices $\{S^1(t_i), \ldots, S^N(t_i)\}$ to the range $[0,1]^N$ while scaling the option values $\{V(S^{(1)}(t_i)), \ldots, V(S^{(N)}(t_i))\}$ also via the same ratio κ, which is the maximum of the stocks:

$$\kappa := \max(S^{(1)}(t_i), \ldots, S^{(N)}(t_i)) \tag{4.10}$$

$$\{S^{(1)}(t_i), \ldots, S^{(N)}(t_i)\} \implies \{\frac{S^{(1)}(t_i)}{\kappa}, \ldots, \frac{S^{(N)}(t_i)}{\kappa}\} \in [0,1]^N \tag{4.11}$$

$$\{V(S^{(1)}(t_i)), \ldots, V(S^{(N)}(t_i))\} \implies \{\frac{V(S^{(1)}(t_i))}{\kappa}, \ldots, \frac{V(S^{(N)}(t_i))}{\kappa}\} \tag{4.12}$$

After obtaining the calculated continuation value $C^*(S^{(j)}(t_i))$ using the kernel function via the formula (4.9), we scale it back:

$$\{C^*(S^{(1)}(t_i)), \ldots, C^*(S^{(N)}(t_i))\} \implies \{C^*(S^{(1)}(t_i)) \cdot \kappa, \ldots, C^*(S^{(N)}(t_i)) \cdot \kappa\} \tag{4.13}$$

After the data are scaled, the optimal bandwidth is also different from before - without scaling, see Figure 4.6. We can set a finite set of parameters $\mathcal{Q}$ according to Györfi [17]:

$$\mathcal{Q} = \{2^{-10}, 2^{-9}, \ldots, 2^0, \ldots, 2^9, 2^{10}\} \tag{4.14}$$

then select the suboptimal bandwidth h^* from $\mathcal{Q}$. In this case, the optimal bandwidth lies practically in the interval of $[0,1]$. However, the difference between each neighboring element of $\mathcal{Q}$ is still large. For accuracy, we can use a *coarse* search first and then use a *fine* search later, according to Chang [9]. Assume that we find $h^*_{\text{coarse}} = 2^{-5}$ as the suboptimal bandwidth from $\mathcal{Q}$. Then we can identify a better region $[2^{-6}, 2^{-4}]$ and set another finite set of parameters $\mathcal{Q}_{\text{fine}}$ as:

$$\mathcal{Q}_{fine} = \{2^{-6}, 2^{-5.8}, \ldots, 2^{-5}, \ldots, 2^{-4.2}, 2^{-4}\} \tag{4.15}$$

After that a fine search for the suboptimal bandwidth $h^* = h^*_{\text{fine}}$ can be processed in $\mathcal{Q}_{\text{fine}}$.

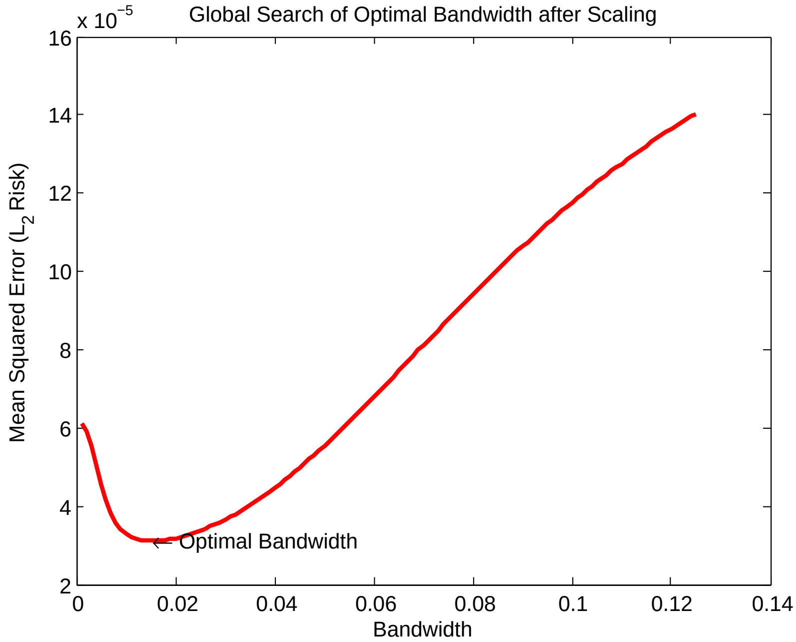


Figure 4.6: Global search of optimal bandwidth after scaling

In this way, the huge computational cost for searching for the global optimal bandwidth can be heavily reduced, for example, from 3960 seconds to 32 seconds, see Table 4.2. However the performance is still guaranteed, see Figure 4.7. The suboptimal bandwidths at different potential exercise dates using coarse search h^*_{coarse} and fine search h^*_{fine} are collected in Table 4.3. Here we use monomial polynomials with degree 3 as basis functions for the modified Longstaff-Schwartz method and the modified Tsitsiklis-Roy method, where $N = 10000$, $N_{\text{train}} = 5000$, $N_{\text{test}} = 5000$ and $N_{\text{New}} = 4000$. We notice that the modified Longstaff-Schwartz method and the modified Tsitsiklis-Roy method both work well. That is not surprising, since both methods perform well for simple payoff options.

Results for Test Case 2			
	Fixed Bandwidth	Global Optimal Bandwidth	Suboptimal Bandwidth
Time	25 s	3960 s	32 s

Table 4.2: Comparison of averaged run time for the kernel method with fixed bandwidth (Algorithm 4.2), the global optimal bandwidth (Algorithm 4.4) and the suboptimal bandwidth from a finite set for Test Case 2.

Results for Test Case 2					
	Suboptimal Bandwidth			Suboptimal Bandwidth	
Exercise Date	Coarse	Fine	Exercise Date	Coarse	Fine
t_1	0.0078	0.0078	t_7	0.0039	0.0034
t_2	0.0078	0.0068	t_8	0.0078	0.0068
t_3	0.0078	0.0059	t_9	0.0039	0.0052
t_4	0.0156	0.0136	t_{10}	0.0078	0.0059
t_5	0.0078	0.0068	t_{11}	0.0039	0.0052
t_6	0.0156	0.0136			

Table 4.3: Suboptimal bandwidths after scaling and parameter selection at potential exercise dates for Test Case 2

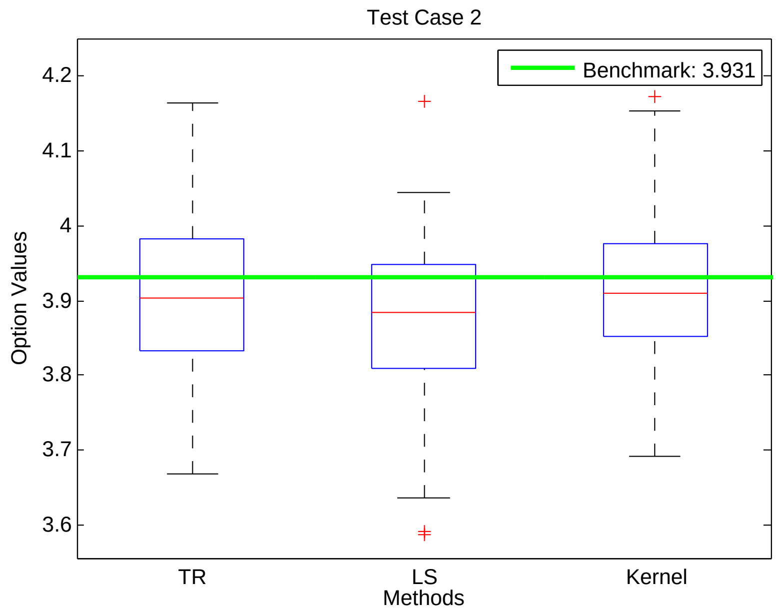


Figure 4.7: Comparison of the modified Tsitsiklis-Roy method (TR), the modified Longstaff-Schwartz method (LS) and the modified kernel method with suboptimal bandwidth from a finite set (Kernel) to price a Bermudan option in Test Case 2.

In our second example, the pricing problem is more difficult. We consider Test Case 4: 1-D American option with strangle-spread-payoff. We choose monomial polynomial with degree 3 as basis functions for the modified Longstaff-Schwartz method and the modified Tsitsiklis-Roy method in this case, where $N = 10000$, $N_{\text{train}} = 5000$, $N_{\text{test}} = 5000$ and $N_{\text{New}} = 10000$. The benchmark is 26.32, denoted as the green line in Figure 4.8. In this case, the modified kernel method delivers much higher option prices than the modified Longstaff-Schwartz method and the modified Tsitsiklis-Roy method.

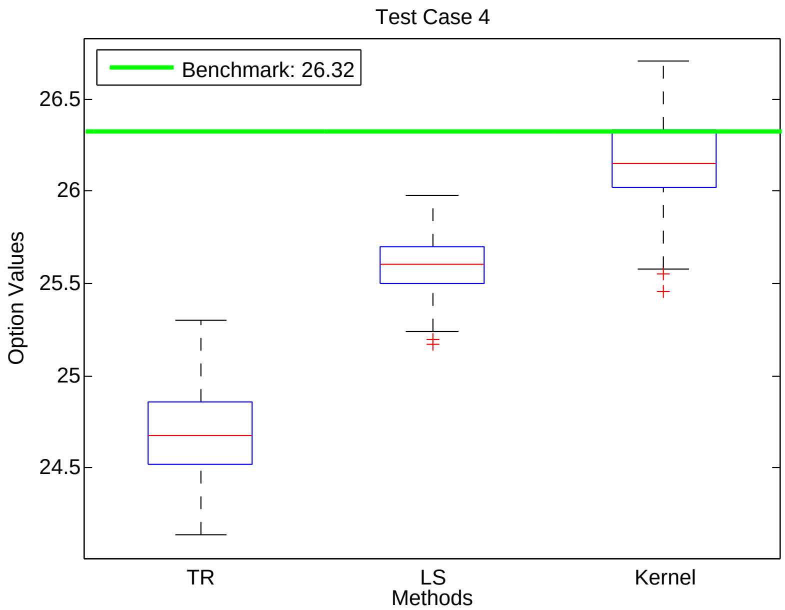


Figure 4.8: Comparison of the modified Tsitsiklis-Roy method (TR), the modified Longstaff-Schwartz method (LS) and the modified kernel method with suboptimal bandwidth from a finite set (Kernel) to price 1-D American option with strangle-spread-payoff in Test Case 4.

In our third example, we consider the high dimensional case, namely the 3-D American geometric-average option with strangle-spread-payoff in Test Case 19. The benchmark option price is 8.934(± 0.001). For the modified Longstaff-Schwartz method and the modified Tsitsiklis-Roy method, monomial polynomials with degree 1 and payoff function are included in the basis functions. The simulated paths for estimating continuation value and for obtaining the lower bound of option price are respectively $N = 10000$, $N_{\text{New}} = 10000$. For the modified kernel method, $N_{\text{train}} = 5000$ is used for training the suboptimal bandwidth, $N_{\text{test}} = 5000$ is used for testing. From Figure 4.9, we notice again that the modified kernel method is superior to modified Tsitsiklis-Roy method and the modified Longstaff-Schwartz method.

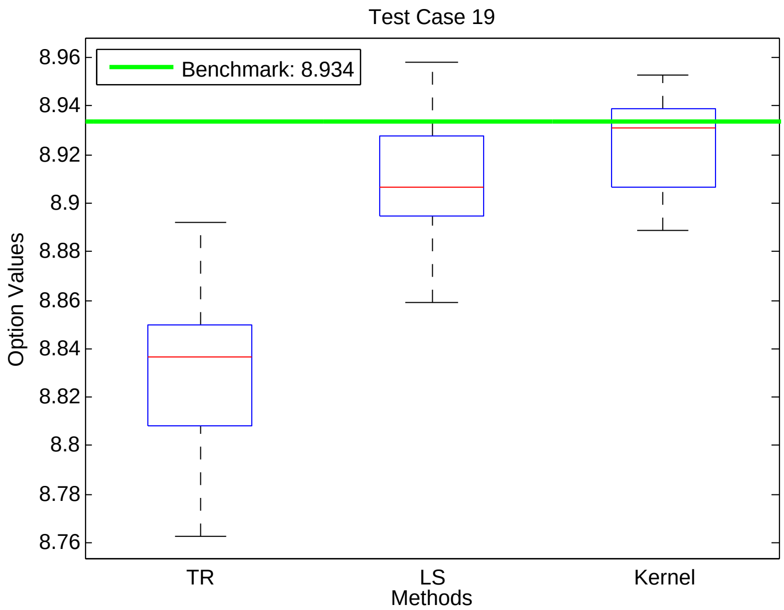


Figure 4.9: Comparison of the modified Tsitsiklis-Roy method (TR), the modified Longstaff-Schwartz method (LS) and the modified kernel method with suboptimal bandwidth from a finite set (Kernel) to price 3-D American geometric-average option with strangle-spread-payoff in Test Case 19.

Finally, we comprehensively test the robustness of the modified kernel method with suboptimal bandwidth selecting from a finite set of parameters in all our test cases from Test Case 1 to Test Case 24. We simulated $N = 10000$ paths to estimate continuation value (stopping rule) and $N_{\text{New}} = 10000$ new paths to compute the lower bound of the option price based on the stopping rule. We use $N_{\text{train}} = 5000$ as training set and $N_{\text{test}} = 5000$ as testing set to obtain the suboptimal bandwidth. We use monomial polynomial with degree up to 3 as basis functions in the modified Longstaff-Schwartz method (LS) and the modified Tsitsiklis-Roy method (TR). We run 100 independent Monte Carlo simulations and obtain 3×100 option prices for these three methods. The median of each 100 option prices in each Test Case are collected in Table 4.4. Notice that Test Case 1, 4, 5, 6, 7, 8 and 24 are real American-style options with infinite exercise dates. According to section 3.2.1, approximation by their Bermudan counterparts with $m = 50$ potential exercise dates usually works fine and thus 50 potential exercise dates are simulated here. The other test cases are all Bermudan option with finite exercise dates and hence finite m for each case is used.

We see clearly that the modified kernel method performs robust in all test cases both within the Black-Scholes model and within the Heston model (Test Case 24) and are even superior than the modified Longstaff-Schwartz method and the modified Tsitsiklis-Roy method in several cases.

Test Case	Benchmark	Modified TR	Modified LS	Modified Kernel
1	7.11	7.0064	7.0738	7.0844
2	3.931	3.9030	3.8849	3.9090
3	4.313	4.2800	4.3005	4.3108
4	26.32	24.6784	25.6032	26.1463
5	7.81	7.7717	7.7869	7.8016
6	23.77	22.7103	22.4499	23.1376
7	4.01	3.9525	3.9307	3.9654
8	3.25	3.1858	3.1944	3.1931
9 (ATM)	11.40	11.0163	11.2541	11.2758
9 (ITM)	15.78	15.3611	15.6366	15.6303
9 (OTM)	5.20	5.0521	5.1607	5.1903
10 (ATM)	13.90	13.2687	13.6046	13.8278
10 (ITM)	21.34	20.3205	20.8598	21.3246
10 (OTM)	1.64	1.6108	1.6179	1.6267
11 (ATM)	2.28	2.1818	2.2033	2.2415
11 (TTM)	5.97	5.6962	5.8311	5.9635
11 (OTM)	0.029	0.0248	0.0253	0.0279
12	1.55	1.5335	1.5441	1.5391
13	1.48	1.4744	1.4731	1.4814
14	1.46	1.4056	1.4428	1.4435
15 (ATM)	17.50	17.1719	17.1779	17.3830
15 (ITM)	25.98	25.2877	25.2603	25.6922
15 (OTM)	2.27	2.2253	2.2282	2.2356
16 (ATM)	0.81	0.7932	0.7943	0.8048
16 (ITM)	2.82	2.7579	2.7422	2.7943
16 (OTM)	0.0022	0.0019	0.0017	0.0020
17	1.77	1.7600	1.7651	1.7654
18	0.97	0.9654	0.9656	0.9683
19	8.934	8.8367	8.9066	8.9310
20	3.27	3.2343	3.2153	3.2491
21	4.77	4.7212	4.7072	4.7287
22	4.32	4.2163	4.2934	4.2843
23	8.42	8.3045	8.3944	8.4003
24 (ATM)	4.65	4.5127	4.5858	4.6145
24 (ITM)	10.65	10.4810	10.6086	10.6274
24 (OTM)	1.68	1.6629	1.6590	1.6611

Table 4.4: Performance of the modified kernel method with suboptimal bandwidth compared with the modified Longstaff-Schwartz method and the modified Tsitsiklis-Roy method in all test cases

4.2 Support Vector Machine

If we simulate N paths of stock prices and use formulas (4.8) and (4.9) of the kernel method to obtain the continuation value $C^*(S^{(j)}(t_i))$, we have to store all N stock prices as training points and compute the kernel function $K_{Gaus}(S^{(j)}(t_i), S^{(k)}(t_i))$ for all possible pairs $S^{(j)}(t_i)$ and $S^{(k)}(t_i)$. And it also leads to excessive computational cost when making prediction of continuation value for new stock price.

This is a significant limitation for the kernel method. The storage requirement and computational cost during prediction is huge. According to Bishop [6], one possible improvement is to use the support vector machine (*SVM*) such that prediction of continuation value for a new stock price depending on the kernel function is only evaluated at a subset of old stock prices ($N_{\text{svm}} < N$). Thus we have a kernel-based algorithm with *sparse* solutions. Chang and Lin [8] developed a free software *LIBSVM* as a library for support vector machines, which can be applied in the scenario of American option pricing.

At some potential exercise date t_i, assume that we have a set of training data comprising N stock prices $\{S^{(1)}(t_i), \ldots, S^{(N)}(t_i)\}$ as training input vectors and corresponding option values (assuming that the option has not been exercised before t_i) $\{V(S^{(1)}(t_i)), \ldots, V(S^{(N)}(t_i))\}$ as training target values.

For simplicity, we define $S = \{S_1, \ldots, S_N\}^\intercal = \{S^{(1)}(t_i), \ldots, S^{(N)}(t_i)\}^\intercal$ and $V = \{V_1, \ldots, V_N\}^\intercal = \{V(S^{(1)}(t_i)), \ldots, V(S^{(N)}(t_i))\}^\intercal$.

According to Bishop [6], we set up a linear regression model to approximate the continuation value $C(S)$:

$$
\begin{aligned}
C(S) &= \omega^\intercal \Phi(S) + b && (4.16)\\
&= \sum_{j=1}^{m} \omega_j \varphi_j(S) + b && (4.17)
\end{aligned}
$$

where $\varphi_j(S)_{j=1,\ldots,m}$ denotes a set of nonlinear functions, ω_j is the corresponding correlation, which needs to be determined and b is a bias parameter. Note that a Gaussian kernel function will be introduced later so that here we do not have to solve explicitly for $\Phi(S)$. Our aim is to minimize a regularized error function:

$$
\frac{1}{2}\sum_{n=1}^{N}\{C(S_n) - V_n\}^2 + \frac{\lambda}{2}||\omega||^2 \tag{4.18}
$$

Further we define ϵ-*insensitive error function* ($\epsilon > 0$) as:

$$
\mathbb{E}_\epsilon(C(S_n) - V_n) = \begin{cases} 0, & \text{if } |C(S_n) - V_n| < \epsilon \\ |C(S_n) - V_n| - \epsilon, & \text{otherwise} \end{cases} \tag{4.19}
$$

Then via a new regularization parameter $c > 0$ by convention the regularized error function (4.18) is changed to be:

$$c\sum_{n=1}^{N}\mathbb{E}_{\epsilon}(C(S_n)-V_n)+\frac{1}{2}||\omega||^2 \tag{4.20}$$

4.2.1 Standard Form

For each stock price S_n, we define two non-negative slack variables $\xi_n \geq 0$ and $\hat{\xi}_n \geq 0$ which should satisfy the following conditions:

- $\xi_n > 0$ and $\hat{\xi}_n = 0$ means that $V_n > C(S_n) + \epsilon$ and S_n lies above the $\epsilon-$ tube.
- $\xi_n = 0$ and $\hat{\xi}_n > 0$ means that $V_n < C(S_n) - \epsilon$ and S_n lies under the $\epsilon-$ tube.
- $\xi_n = 0$ and $\hat{\xi}_n = 0$ means that $C(S_n) - \epsilon \leq V_n \leq C(S_n) + \epsilon$ and S_n lies inside the $\epsilon-$ tube.
  - $V_n = C(S_n)+\epsilon$ means that S_n lies on the upper boundary of the $\epsilon-$ tube.
  - $V_n = C(S_n)-\epsilon$ means that S_n lies on the lower boundary of the $\epsilon-$ tube.
  - $C(S_n)-\epsilon < V_n < C(S_n)+\epsilon$ means that S_n lies within the $\epsilon-$ tube but not on the boundaries.

Denote $\xi = \{\xi_1,\ldots,\xi_N\}^{\intercal}$ and $\hat{\xi} = \{\hat{\xi}_1,\ldots,\hat{\xi}_N\}^{\intercal}$, we obtain the standard form of support vector regression by minimizing the regularized $\epsilon-$insensitive error function:

$$\begin{aligned} \min_{\omega,b,\xi,\hat{\xi}} \quad & c\sum_{n=1}^{N}(\xi_n+\hat{\xi}_n)+\frac{1}{2}||\omega||^2 \\ \text{subject to} \quad & \xi_n \geq 0, i=1,\ldots,n \\ & \hat{\xi}_n \geq 0, i=1,\ldots,n \\ & V_n \leq C(S_n)+\epsilon+\xi_n, i=1,\ldots,n \\ & V_n \geq C(S_n)-\epsilon-\hat{\xi}_n, i=1,\ldots,n \end{aligned} \tag{4.21}$$

From the optimization theory, we know that for a convex minimization problem with convex constraints there is an equivalent dual unconstrained maximization problem by using nonnegative Lagrange multipliers. Thus we can introduce Lagrange multipliers $a_n \geq 0$, $\hat{a}_n \geq 0$, $\mu_n \geq 0$ and $\hat{\mu}_n \geq 0$ and maximize the Lagrangian function:

$$
\begin{aligned}
L &= (c\sum_{n=1}^{N}(\xi_n+\hat{\xi}_n)+\frac{1}{2}||\omega||^2)-\sum_{n=1}^{N}(\mu_n\xi_n+\hat{\mu}_n\hat{\xi}_n) \\
&\quad -\sum_{n=1}^{N}a_n(\epsilon+\xi_n+C(S_n)-V_n)-\sum_{n=1}^{N}\hat{a}_n(\epsilon+\hat{\xi}_n-C(S_n)+V_n) \qquad (4.22)\\
&\overset{eq.(4.16)}{=} (c\sum_{n=1}^{N}(\xi_n+\hat{\xi}_n)+\frac{1}{2}||\omega||^2)-\sum_{n=1}^{N}(\mu_n\xi_n+\hat{\mu}_n\hat{\xi}_n) \\
&\quad -\sum_{n=1}^{N}a_n(\epsilon+\xi_n+(\omega^{\intercal}\Phi(S_n)+b)-V_n)-\sum_{n=1}^{N}\hat{a}_n(\epsilon+\hat{\xi}_n-(\omega^{\intercal}\Phi(S_n)+b)+V_n)
\end{aligned}
$$

In order to obtain the optimum, we set the derivatives of L with respect to $\omega, b, \xi_n, \hat{\xi}_n$:

$$
\frac{\partial L}{\partial \omega}=0 \qquad \frac{\partial L}{\partial b}=0
$$
$$
\frac{\partial L}{\partial \xi_n}=0 \qquad \frac{\partial L}{\partial \hat{\xi}_n}=0
$$

Thus we have:

$$\sum_{n=1}^{N}(a_n-\hat{a}_n)\Phi(S_n) = \omega \qquad (4.23)$$

$$\sum_{n=1}^{N}(a_n-\hat{a}_n) = 0 \qquad (4.24)$$

$$a_n+\mu_n = c \qquad (4.25)$$

$$\hat{a}_n+\hat{\mu}_n = c \qquad (4.26)$$

Put equations (4.23) - (4.26) into the Lagrangian function (4.22) and eliminate variables $\xi_n, \hat{\xi}_n, \mu_n, \hat{\mu}_n, \omega, b$ and define $a=\{a_1,\ldots,a_N\}^{\intercal}$ and $\hat{a}=\{\hat{a}_1,\ldots,\hat{a}_N\}^{\intercal}$, we obtain:

$$
\begin{aligned}
\tilde{L}(a,\hat{a}) \;:=\;& L \\
=\;& \underbrace{\sum_{n=1}^{N}((c-\mu_n-a_n)\xi_n)}_{=0} + \underbrace{\sum_{n=1}^{N}((c-\hat{\mu}_n-\hat{a}_n)\hat{\xi}_n)}_{=0} - \underbrace{\sum_{n=1}^{N}((a_n-\hat{a}_n)b)}_{=0} \\
& +\frac{1}{2}\omega^{\mathsf{T}}\omega - \omega^{\mathsf{T}}\underbrace{\sum_{n=1}^{N}(a_n\Phi(S_n)-\hat{a}_n\Phi(S_n))}_{=\omega} - \epsilon\sum_{n=1}^{N}(a_n+\hat{a}_n) + \sum_{n=1}^{N}(a_n-\hat{a}_n)V_n \\
=\;& -\frac{1}{2}\omega^{\mathsf{T}}\omega - \epsilon\sum_{n=1}^{N}(a_n+\hat{a}_n) + \sum_{n=1}^{N}(a_n-\hat{a}_n)V_n \\
=\;& -\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}(a_n-\hat{a}_n)(a_m-\hat{a}_m)K(S_n,S_m) - \epsilon\sum_{n=1}^{N}(a_n+\hat{a}_n) + \sum_{n=1}^{N}(a_n-\hat{a}_n)V_n
\end{aligned}
\tag{4.27}
$$

where we introduce a kernel function $K(S_n, S_m) = \Phi(S_n)^{\mathsf{T}}\Phi(S_m)$, for example a Gaussian kernel function $K_{\text{Gauss}}(S_n, S_m)$ (see equation (4.8)).

4.2.2 Dual Problem

The standard form of support vector regression (equation (4.21)) is switched to be a dual optimization problem:

$$
\begin{aligned}
\min_{a,\hat{a}} \quad & \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}(a_n-\hat{a}_n)(a_m-\hat{a}_m)K(S_n,S_m) \\
& +\epsilon\sum_{n=1}^{N}(a_n+\hat{a}_n) - \sum_{n=1}^{N}(a_n-\hat{a}_n)V_n \\
\text{subject to} \quad & 0 \le a_n \le c, \quad n = 1,\dots,N \\
& 0 \le \hat{a}_n \le c, \quad n = 1,\dots,N \\
& \sum_{n=1}^{N}(a_n-\hat{a}_n) = 0
\end{aligned}
\tag{4.28}
$$

A quadratic optimization problem with one linear constraint has the general form as:

$$
\begin{aligned}
\min_{x} \quad & f(x) \equiv \frac{1}{2}x^{\mathsf{T}}Qx + p^{\mathsf{T}}x \\
\text{subject to} \quad & 0 \le x_i \le C, i = 1,\dots,n \\
& y^{\mathsf{T}}x = \Delta
\end{aligned}
$$

where $y_i = \pm 1, i = 1, \dots, n$. The constraint $y^\intercal x = \Delta$ is called a linear constraint.

We notice that the dual optimization problem (equation (4.28)) can also be written in the form of quadratic optimization:

$$\min_{a,\hat{a}} \quad \frac{1}{2}[a^\intercal,\ \hat{a}^\intercal] \begin{bmatrix} K & -K \\ -K & K \end{bmatrix} \begin{bmatrix} a \\ \hat{a} \end{bmatrix} + [\epsilon e^\intercal - V^\intercal,\ \epsilon e^\intercal + V^\intercal] \begin{bmatrix} a \\ \hat{a} \end{bmatrix} \tag{4.29}$$

$$\text{subject to} \quad 0 \le a_n, \hat{a}_n \le c, \quad n = 1, \dots, N \tag{4.30}$$

$$y^\intercal \begin{bmatrix} a \\ \hat{a} \end{bmatrix} = 0 \tag{4.31}$$

$$\text{where} \quad y := [\underbrace{1, \dots, 1}_{N}, \underbrace{-1, \dots, -1}_{N}]^\intercal$$

$$e := [1, \dots, 1]^\intercal$$

$$K := (K(S_n, S_m))_{n,m} \quad n, m = 1, \dots, N$$

Since the matrix $K(S_n, S_m)$ is positive definite, we notice that the dual problem (equation (4.28)) is a quadratic optimization problem with positive definite objective function matrix, thus it is a convex optimization problem. We know that any local optimum of a convex optimization problem must be a global optimum. Thus the support vector regression has a very important property which is that solving the optimization is equivalent to solve its dual convex quadratic optimization and thus any local minimum here is also a global minimum. And this is the reason why we solve the dual problem not the primary one.

After solving this dual optimization problem and putting the equation (4.23) into the equation (4.16), the continuation value $C(S)$ for a new stock price S can been computed as:

$$C(S) = \sum_{n=1}^{N} (a_n - \hat{a}_n) K(S, S_n) + b \tag{4.32}$$

Support vectors are defined as the stock prices S_n which have a contribution to predict the continuation value, which means $a_n - \hat{a}_n \neq 0$. Those stock prices, which are not support vectors and have $a_n - \hat{a}_n = 0$, are not necessary to be stored and can be discarded. In this way we can reduce the storage requirement and computational cost compared with traditional kernel method.

According to the corresponding Karush-Kuhn-Tucker (KKT) conditions, the product of the dual variables and the constraints is equal to zero:

$$a_n(C(S_n) + \epsilon + \xi_n - V_n) = 0 \tag{4.33}$$

$$\hat{a}_n(V_n + \epsilon + \hat{\xi}_n - C(S_n)) = 0 \tag{4.34}$$

$$\xi_n(c - a_n) = 0 \tag{4.35}$$

$$\hat{\xi}_n(c - \hat{a}_n) = 0 \tag{4.36}$$

From these equations, we see that only when $(C(S_n) + \epsilon + \xi_n - V_n)$ is equal to zero, a_n can be nonzero and only when $(V_n + \epsilon + \hat{\xi}_n - C(S_n))$ is equal to zero,

$\hat{a}$ can be nonzero. a_n and $\hat{a}_n$ cannot be nonzero at the same time, otherwise $(C(S_n) + \epsilon + \xi_n - V_n)$ and $(V_n + \epsilon + \hat{\xi}_n - C(S_n))$ is equal to zero at the same time, then the sum of these two terms $(2\epsilon + \xi_n + \hat{\xi}_n)$ is equal to zero, which contradicts to the nonnegativity of ϵ, ξ_n and $\hat{\xi}_n$.

According to the definition of ξ_n and $\hat{\xi}_n$, $(C(S_n) + \epsilon + \xi_n - V_n)$ is equal to zero, which means that the stock price S_n lies either on the upper boundary of the ϵ-tube ($\xi_n = 0, \hat{\xi}_n = 0$) or above the upper boundary ($\xi_n > 0, \hat{\xi}_n = 0$); $(V_n + \epsilon + \xi_n - C(S_n))$ is equal to zero, means that the stock price S_n lies either on the lower boundary of the ϵ-tube ($\xi_n = 0, \hat{\xi}_n = 0$) or under the lower boundary ($\xi_n = 0, \hat{\xi}_n > 0$). $C(S_n) - \epsilon - \hat{\xi}_n < V_n < C(S_n) + \epsilon + \xi_n$ means that the stock price S_n lies within the ϵ-tube ($\xi_n = 0, \hat{\xi}_n = 0$), in this situation both a_n and $\hat{a}_n$ are equal to zero. We summarize all cases in Table 4.5, where we denote C_n as $C(S_n)$ for simplicity.

Thus the support vectors are those stock prices, which are either above the upper boundary, under the lower boundary or on the boundaries.

Case	Explaination	ξ_n	$\hat{\xi}_n$	V_n	V_n	a_n	$\hat{a}_n$
I	above upper boundary	> 0	$= 0$	$= C_n + \epsilon + \xi_n$	$> C_n - \epsilon - \hat{\xi}_n$	$= c$	$= 0$
II	on upper boundary	$= 0$	$= 0$	$= C_n + \epsilon + \xi_n$	$> C_n - \epsilon - \hat{\xi}_n$	$\in (0, c)$	$= 0$
III	within upper ϵ-tube	$= 0$	$= 0$	$< C_n + \epsilon + \xi_n$	$> C_n - \epsilon - \hat{\xi}_n$	$= 0$	$= 0$
IV	within lower ϵ-tube	$= 0$	$= 0$	$< C_n + \epsilon + \xi_n$	$> C_n - \epsilon - \hat{\xi}_n$	$= 0$	$= 0$
V	on lower boundary	$= 0$	$= 0$	$< C_n + \epsilon + \xi_n$	$= C_n - \epsilon - \hat{\xi}_n$	$= 0$	$\in (0, c)$
VI	under lower boundary	$= 0$	> 0	$< C_n + \epsilon + \xi_n$	$= C_n - \epsilon - \hat{\xi}_n$	$= 0$	$= c$

Table 4.5: All cases for the relation between stock prices and ϵ-tube

The bias parameter b can be estimated as follows. Consider that a stock S_m lies on the upper boundary of the ϵ-tube, we have $\xi_m = 0$, $0 < a_m < c$ and $(C(S_m) + \epsilon + \xi_m - V_m) = 0$

$$
\begin{aligned}
b &\overset{eq.(4.32)}{=} C(S_m) - \sum_{n=1}^{N} (a_n - \hat{a}_n) K(S_m, S_n) \\
&= V_m - \epsilon - \sum_{n=1}^{N} (a_n - \hat{a}_n) K(S_m, S_n)
\end{aligned} \tag{4.37}
$$

Practically we can average all estimates of b where stocks lie on the boundaries.

4.2.3 ν-SVM

Schölkopf [41] introduced a new parameter $\nu \in (0, 1]$ which limits the ratio of stock prices lying outside the ϵ-tube instead of using the fixed width ϵ of ϵ-tube. That means, at least νN stock prices lie outside the $\epsilon-$tube. Thus the number of support vectors is at least νN, which lie either outside or on the $\epsilon-$tube. In this way, the standard form of support vector regression (equation (4.21)) has an alternative formulation, which is called ν-support vector regression (ν-SVM):

$$\begin{aligned}
\min_{\omega,b,\xi,\hat{\xi}} \quad & c(\nu\epsilon + \frac{1}{N}\sum_{n=1}^{N}(\xi_n + \hat{\xi}_n)) + \frac{1}{2}||\omega||^2 \qquad (4.38)\\
\text{subject to} \quad & \xi_n \geq 0, i = 1, \ldots, n \\
& \hat{\xi}_n \geq 0, i = 1, \ldots, n \\
& V_n \leq C(S_n) + \epsilon + \xi_n, i = 1, \ldots, n \\
& V_n \geq C(S_n) - \epsilon - \hat{\xi}_n, i = 1, \ldots, n
\end{aligned}$$

The dual problem is:

$$\begin{aligned}
\min_{a,\hat{a}} \quad & \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}(a_n - \hat{a}_n)(a_m - \hat{a}_m)K(S_n, S_m) \\
& -\sum_{n=1}^{N}(a_n - \hat{a}_n)V_n \qquad (4.39)\\
\text{subject to} \quad & 0 \leq a_n \leq \frac{c}{N}, n = 1, \ldots, N \\
& 0 \leq \hat{a}_n \leq \frac{c}{N}, n = 1, \ldots, N \\
& \sum_{n=1}^{N}(a_n - \hat{a}_n) = 0 \\
& \sum_{n=1}^{N}(a_n + \hat{a}_n) \leq \nu c
\end{aligned}$$

After obtaining a and $\hat{a}$, the approximation function for the continuation value for a new stock price S is the same as before:

$$C(S) = \sum_{n=1}^{N}(a_n - \hat{a}_n)K(S, S_n) + b \qquad (4.40)$$

Since we use the $\epsilon-$insensitive function to penalize errors which are bigger than ϵ, we usually have sparse representation of the prediction (equation (4.40)). In this way the support vector regression leads to significant representational and algorithmic advantages. In MATLAB, a function *quadprog* is implemented to solve quadratic programming problems. Besides of this, the free software *LIBSVM* designed by Chang and Lin [8] includes functions to solve support vector regression problems. An introduction of this software can be read in Chang [9].

4.2.4 Grid-Search and Analytic Parameter Selection

In order to avoid numerical difficulty, we should firstly use scaling technique before using the support vector machine method as what we do for the kernel method in the section 4.1.3.

When using $\epsilon-$SVM, we should determine three parameters, namely ϵ, c and h, where h is the bandwidth of the Gaussian kernel function, see equation (4.8).

Notice that the parameter c controls the balance between two terms of the regularized error function (equation (4.20)). The first term is about the degree to which errors larger than ϵ can be tolerated. The second term is about the complexity of the model. If c is too big, the objective function is only to minimize the sum of the errors, without considering the model complexity, which leads to the flatness of the regression function. The parameter ϵ defines the width of the $\epsilon-$tube and controls the number of support vectors which are used to form the regression function. If ϵ is very large, we only select very few support vectors, which also leads to the flatness of the regression function. Thus, both c and ϵ controls the model complexity, but in different ways.

According to Chang [9], ϵ, c and h can be chosen via a grid-search process. In practice for pricing American options, we can choose c from a finite set of parameters $\{2^{-5}, \ldots, 2^{15}\}$, ϵ from a finite set of parameters $\{2^{-15}, \ldots, 2^{5}\}$ and h from $\{2^{-5}, \ldots, 2^{5}\}$. Each pair of (c, h) is tried and the one with the smallest mean squared error is chosen as the suboptimal parameter pair. Similar as in the section 4.1.3, we can first use a coarse search and identify a good region and then use a fine search in this region to obtain the best parameter pair.

However, the gird-search process for three parameters is very computation and data-intensive. Cherkassky [10] proposed a rule-of-thumb analytic strategy for selecting the regularization parameter c and the tube width parameter ϵ as follows:

$$c = \max(|\bar{V} + 3\sigma_V|, |\bar{V} - 3\sigma_V|) \tag{4.41}$$

where $\bar{V} = \frac{1}{N}\sum_{n=1}^{N} V_n$ is the mean of the option values $V = \{V_1, \ldots, V_N\}^\intercal$ and σ_V is the corresponding standard deviation of V.

$$\epsilon = 3\sigma\sqrt{\frac{\log N}{N}} \tag{4.42}$$

where σ is the standard deviation of the input noise level and can be estimated by $\hat{\sigma}$:

$$\hat{\sigma}^2 = \frac{k \cdot N^{\frac{1}{5}}}{k \cdot N^{\frac{1}{5}} - 1} \frac{1}{N} \sum_{n=1}^{N} (V_n - \hat{V}_n)^2 \tag{4.43}$$

where $\hat{V}_n$ is estimated via k-nearest-neighbors regression (see Györfi [17]) by taking a local average of k option values from the set V:

$$\hat{V}_n = \frac{1}{k}\sum_{i=1}^{k} V_i \tag{4.44}$$

where the corresponding stock prices $\{S_1,\dots,S_k\}^\intercal \subsetneqq \{S_1,\dots,S_N\}^\intercal$ are k nearest ones from the estimation stock price S_n in terms of the increasing Euclidean distance $||S_n - S_1|| \leq ||S_n - S_2|| \leq \dots \leq ||S_n - S_k||$.

Furthermore, Cherkassky [10] pointed out that the specific k-value does not affect the estimation of the noise σ very much. Thus he suggested using $k = 3$ and we obtain:

$$\hat{\sigma}^2 = \frac{3 \cdot N^{\frac{1}{5}}}{2 \cdot N^{\frac{1}{5}} - 1} \frac{1}{N} \sum_{n=1}^{N} (V_n - \hat{V}_n)^2 \tag{4.45}$$

After c and ϵ is determined, the kernel bandwidth h is chosen firstly by a coarse search from the set $\{2^{-5},\dots,2^5\}$ and then by a fine search as in the section 4.1.3. In this way, the time for searching parameters can be hugely reduced.

Now we test the $\epsilon-$SVM algorithm to price American-style options in Test Case 4: 1-D American option with strangle-spread-payoff. As before, we simulate $N = 10000$ as in-sample paths for obtaining the stopping rule and $N_{new} = 10000$ as out-of-sample paths for computing the option value using this stopping rule. $N_{\text{train}} = 5000$ and $N_{\text{test}} = 5000$ are used respectively for training and testing the parameter kernel bandwidth h. In Test Case 4, the number of potential exercise dates $m = 12$. We collect the sample mean value of each parameter at the last but one exercise date t_{m-1} in Table 4.6, where ν is the ratio of stock prices lying outside the $\epsilon-$tube, nSV is the practical number of support vectors and $nBSV$ is the practical number of bounded support vectors ($\alpha_n = c$, see Table 4.5). We run 100 independent simulations by the $\epsilon-$SVM method to obtain 100 option prices and illustrate them in Figure 4.10.

$\bar{V}$	σ_V	c	$\hat{\sigma}$	ϵ	h	ν	nSV	$nBSV$
0.0335	0.0241	0.1057	0.0139	0.0013	0.1127	0.8548	8565	8532

Table 4.6: The sample mean value of each parameter at the last but one exercise date for Test Case 4 using support vector machine method.

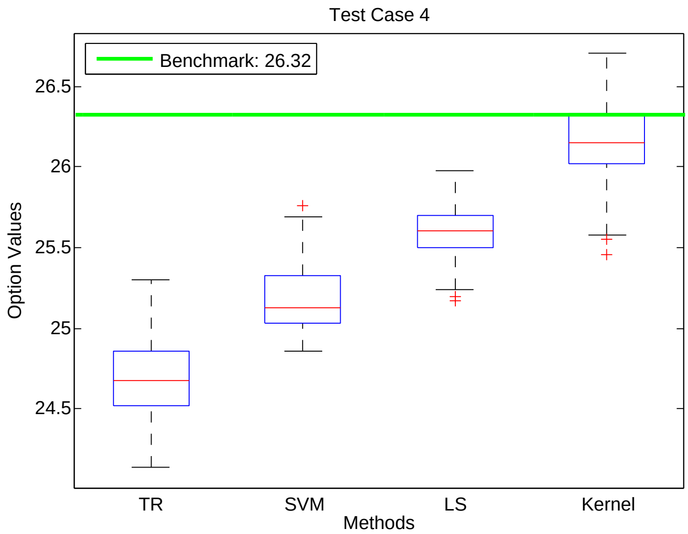


Figure 4.10: Comparison of the modified Tsitsiklis-Roy method (TR), the modified support vector machine method (SVM), the modified Longstaff-Schwartz method (LS) and the modified kernel method with sub-optimal bandwidth from a finite set (Kernel) to price 1-D American option with strangle-spread-payoff in Test Case 4.

From the figure, we notice that the result by the SVM method is superior to the TR method, but is still inferior to the LS method and the Kernel method. The reason is that while in the Kernel method, only one parameter h has to be optimized, in the SVM method, three parameters ϵ, c and h need to be determined. Although the SVM method reduces the storage of the number of stock prices by storing only the support vectors to predict continuation value of a new stock price (here in Table 4.6 storage is reduced from 10000 to 8565), it loses some extend of accuracy compared with its counterpart the Kernel method, which storage all old stock prices. Thus we recommend using the Kernel method instead of the SVM method.

5 Reversion Formula for Implementation of the Longstaff-Schwartz Method on FPGA

In the previous sections, all discussed algorithms or their potential improvements to price American-style options are based on implementation on central processing units (CPU), which can be considered as a fixed hardware with general purpose. On the contrary, the field programmable gate array (FPGA) can be considered as a flexible hardware which can be adjusted according to the application. The use of FPGA can run more efficiently and reduce the energy consumption enormously. Although we can map many numeric algorithms to FPGA directly, there remain a lot of other algorithms which are better to be executed on CPU. We choose the Xilinx Zynq-7000 hybrid CPU/FPGA device to implement the Longstaff-Schwartz method (Algorithm 3.2) such that the best of two worlds - hardware and software can be exploited. For the Black-Scholes model, we propose a novel *Reverse Longstaff-Schwartz algorithm*, which does not require to store the full intermediate stock prices and reduce the requirements on external memory. Our result is 16x faster and 268x more energy-efficient than an optimized Intel CPU implementation, more details can be found in Varela, Brugger, Wehn, Korn and Tang [44]. For the Heston model, we also propose a reversion formula for the stock price and volatility. However its implementation on FPGA and the corresponding test hasn't been finished yet by the end of my PhD, for the embedded architecture is too difficult.

5.1 Black-Scholes Model

5.1.1 Reversion Formula

In this work we apply the *Euler discretization* to discretize the stochastic differential equation of the Black-Scholes model (equation (1.7)) into m steps with equal step sizes $\Delta t = \frac{T}{m}$:

$$\hat{S}(t_{i+1}) \quad = \quad \hat{S}(t_i) \exp\left(\left(r - \delta - \frac{\sigma^2}{2}\right)\Delta t + \sigma\sqrt{\Delta t}\, W(t_i)\right), \qquad (5.1)$$

with $W(t_i)$ being independent standard normal random variables.

In the Longstaff-Schwartz method (Algorithm 3.2), all paths are generated firstly in step (1) and then go back from maturity to initial date in step (3). This means the value of each stock price at each potential exercise date for all paths has to be stored, which totals dmN values, d being the dimension of the option, N being the number of paths, m being the number of potential exercise dates. We call this standard approach the *path storage solutions*. However, this approach leads to the requirement of several external high-speed memory devices for FPGA, because FPGA only has limited internal storage of a few MB. And this design is very complex and time consuming. Now we present a novel idea based on recomputation to avoid storing so many data.

Instead of storing the stock prices at each potential exercise date, we only store the final stock prices at maturity $\hat{S}(T) \equiv \hat{S}(t_m)$ and then recompute all the other stock prices alongside step (3) of the Longstaff-Schwartz algorithm. For that to work we need to find a way to compute the stock price $\hat{S}(t_i)$ based on the future price $\hat{S}(t_{i+1})$:

$$\hat{S}(t_m) \to \hat{S}(t_{m-1}) \ldots \to \hat{S}(t_1) \to \hat{S}(t_0). \tag{5.2}$$

The discretized Black-Scholes equation (equation (5.1)) is reversible if we supply the same random numbers. Thus we obtain the reversion formula:

$$\hat{S}(t_i) \quad = \quad \hat{S}(t_{i+1}) \exp\left(\left(\frac{\sigma^2}{2} - (r-\delta)\right)\Delta t - \sigma\sqrt{\Delta t}\, W(t_i)\right). \tag{5.3}$$

In our work, we use the Mersenne twister (MT) 19937 algorithm (see Matsumoto [34]) to generate a sequence of random numbers. Instead of storing the random numbers, the idea is to build a random number generator that generates exactly the opposite sequence, starting from the last one. Fortunately, the Mersenne twister is a linear random number generator, meaning that its state transition function is invertible. Based on this a reversed Mersenne twister can be built. In fact, while the tempering function is kept unchanged, only the internal states are to be recomputed. As a result, the Reverse Longstaff-Schwartz method only needs to store and communicate dN values.

5.1.2 Test

The architecture of this design contains three steps:

- Step 1: It consists of the paths generation process until maturity, fully implemented on FPGA.
- Step 2: FPGA reconfiguration takes places, instantiating all modules related to the option pricing.
- Step 3: Paths are traversed step by step, backwards from maturity until the initial date in order to obtain the option price. To increase flexibility to easily adapt the calculation of the regression, parts of it are done on CPU.

Once the initial date is reached, the values in the cash flow matrix are averaged, which constitutes the option price.

Paths can be either stored in an external memory chip, following the traditional approach and shown in orange color in Figure 5.1, or they can be recomputed based on our novel Reverse Longstaff-Schwartz approach, shown in green color.

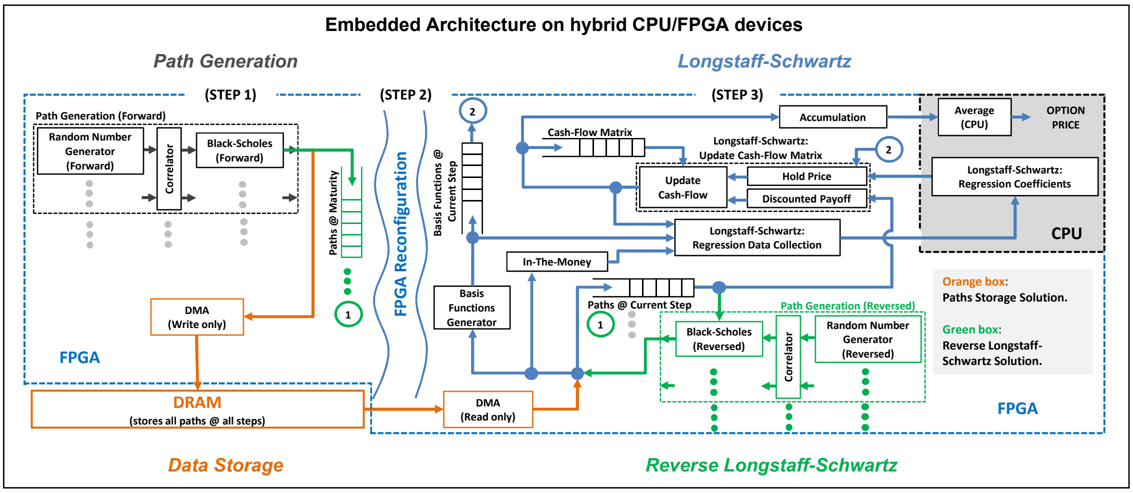


Figure 5.1: Design architecture including both solutions: Paths Storage vs Reverse Longstaff-Schwartz for pricing high-dimensional American options on hybrid CPU/FPGA systems

To evaluate the runtime and energy consumption we price an American maximum call option on two correlated stocks with 365 time steps and 10K paths per stock. Figure 5.2 presents the energy consumption breakdown of the whole architecture when the novel Reverse LS approach is implemented. When comparing the recomputation of the paths in FPGA against the storage of all paths in DRAM (both when writing and reading data), there is a reduction in energy consumption of 2x, as depicted in Figure 5.3.

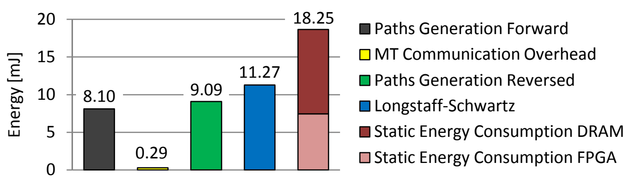


Figure 5.2: Energy consumption breakdown of the Longstaff-Schwartz architecture on Zynq.

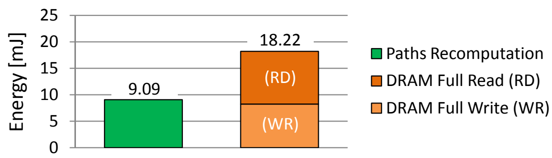


Figure 5.3: Energy required to read and write the paths from DRAM is two times as high as recomputing it on the FPGA alongside the later part of the Longstaff-Schwartz algorithm.

5.2 Heston Model

First we use the native Euler scheme to the stochastic differential equations in the Heston model (equations (1.11)-(1.12)):

$$V(t_{i+1}) = V(t_i) + \kappa(\theta - V(t_i))\Delta t + \sigma\sqrt{V(t_i)}\sqrt{\Delta t}W(t_i) \quad (5.4)$$

$$X(t_{i+1}) = X(t_i) + (r - \frac{1}{2}V(t_i))\Delta t + \sqrt{V(t_i)}\sqrt{\Delta t}Z(t_i) \quad (5.5)$$

where $W(t_i)$ and $Z(t_i)$ are two correlated standard normal random variables and $X(t_i) = \log(S(t_i))$ is the log-stock price, $V(t_i)$ is the volatility (see page 226 of Korn [26]).

While the reversion idea for the Black-Scholes model is intuitive and straightforward (equation (5.3)), the reversion formula for the Heston model is much more complicated. Since in reality the volatility can not be negative, we have several ways to avoid this. The most popular choices are reflection technique and full truncation.

5.2.1 Reversion Formula by Reflection Technique

First we introduce the Euler scheme using reflection technique for the Heston model:

$$V(t_{i+1}) = |V(t_i)| + \kappa(\theta - |V(t_i)|)\Delta t + \sigma\sqrt{|V(t_i)|}\sqrt{\Delta t}W(t_i) \quad (5.6)$$

$$X(t_{i+1}) = X(t_i) + (r - \frac{1}{2}|V(t_i)|)\Delta t + \sqrt{|V(t_i)|}\sqrt{\Delta t}Z(t_i) \quad (5.7)$$

As long as we have the volatility $V(t_i)$, the reversion formula for the log-stock price is straightforward :

$$X(t_i) = X(t_{i+1}) - (r - \frac{1}{2}|V(t_i)|)\Delta t - \sqrt{|V(t_i)|}\sqrt{\Delta t}Z(t_i) \quad (5.8)$$

We focus on discussion of the reversion formula for the volatility.

Positive Reversion Formula

If $V(t_i) \geq 0$, we have $|V(t_i)| = V(t_i)$, thus equations (5.6) and (5.7) become:

$$V(t_{i+1}) = V(t_i) + \kappa(\theta - V(t_i))\Delta t + \sigma\sqrt{V(t_i)}\sqrt{\Delta t}W(t_i) \quad (5.9)$$

$$X(t_{i+1}) = X(t_i) + (r - \frac{1}{2}V(t_i))\Delta t + \sqrt{V(t_i)}\sqrt{\Delta t}Z(t_i) \quad (5.10)$$

The equation (5.9) becomes:

$$V(t_{i+1}) = (1 - \kappa\Delta t)V(t_i) + \sigma\sqrt{\Delta t}W(t_i)\sqrt{V(t_i)} + \kappa\theta\Delta t \quad (5.11)$$

Define h, a, b and c as follows:

$$\begin{aligned} h &:= \sqrt{V(t_i)} \geq 0 \\ a &:= 1 - \kappa\Delta t \\ b &:= \sigma\sqrt{\Delta t}W(t_i) \\ c &:= \kappa\theta\Delta t \end{aligned}$$

Thus the equation (5.11) becomes:

$$V(t_{i+1}) = a \cdot h^2 + b \cdot h + c \quad (5.12)$$

Define $\delta := b^2 - 4a(c - V(t_{i+1}))$, the solutions of the equation (5.12) are:

$$\begin{aligned} h_1 &= \frac{-b - \sqrt{\delta}}{2a} \\ h_2 &= \frac{-b + \sqrt{\delta}}{2a} \end{aligned}$$

There are five cases to be considered:

1. $\delta < 0$, store time step t_i and variance $V(t_i)$.
2. $\delta \geq 0$, $h_1 \geq 0$ and $h_2 \geq 0$, store time step t_i and variance $V(t_i)$.
3. $\delta \geq 0$, $h_1 \leq 0$ and $h_2 \leq 0$, store time step t_i and variance $V(t_i)$.
4. $\delta \geq 0$, $h_1 \geq 0$ and $h_2 \leq 0$, store time step t_i and variance $V(t_i)$.
5. $\delta \geq 0$, $h_1 < 0$ and $h_2 > 0$, which occurs most often, we obtain the **positive reversion formula**:

$$V(t_i) = h_2^2 = (\frac{-b + \sqrt{\delta}}{2a})^2 \quad (5.13)$$

Negative Reversion Formula

If $V(t_i) < 0$, we have $|V(t_i)| = -V(t_i)$, thus equations (5.6) and (5.7) become:

$$V(t_{i+1}) = -V(t_i) + \kappa(\theta + V(t_i))\Delta t + \sigma\sqrt{-V(t_i)}\sqrt{\Delta t}W(t_i) \quad (5.14)$$

$$X(t_{i+1}) = X(t_i) + (r + \frac{1}{2}V(t_i))\Delta t + \sqrt{-V(t_i)}\sqrt{\Delta t}Z(t_i) \quad (5.15)$$

Define $\hat{V}(t_i) := -V(t_i)$, the equation (5.14) becomes:

$$\begin{aligned} V(t_{i+1}) &= \hat{V}(t_i) + \kappa(\theta - \hat{V}(t_i))\Delta t + \sigma\sqrt{\hat{V}(t_i)}\sqrt{\Delta t}W(t_i) \\ &= (1 - \kappa\Delta t)\hat{V}(t_i) + \sigma\sqrt{\Delta t}W(t_i)\sqrt{\hat{V}(t_i)} + \kappa\theta\Delta t \end{aligned} \quad (5.16)$$

Similarly, define h, a, b and c as follows:

$$\begin{aligned} h &:= \sqrt{\hat{V}(t_i)} \geq 0 \\ a &:= 1 - \kappa\Delta t \\ b &:= \sigma\sqrt{\Delta t}W(t_i) \\ c &:= \kappa\theta\Delta t \end{aligned}$$

Thus the equation (5.16) becomes:

$$V(t_{i+1}) = a \cdot h^2 + b \cdot h + c \quad (5.17)$$

Define $\delta := b^2 - 4a(c - V(t_{i+1}))$, the solutions of the equation (5.17) are:

$$\begin{aligned} h_1 &= \frac{-b - \sqrt{\delta}}{2a} \\ h_2 &= \frac{-b + \sqrt{\delta}}{2a} \end{aligned}$$

There are also five cases to be considered:

1. $\delta < 0$, store time step t_i and variance $V(t_i)$.
2. $\delta \geq 0$, $h_1 \geq 0$ and $h_2 \geq 0$, store time step t_i and variance $V(t_i)$.
3. $\delta \geq 0$, $h_1 \leq 0$ and $h_2 \leq 0$, store time step t_i and variance $V(t_i)$.
4. $\delta \geq 0$, $h_1 \geq 0$ and $h_2 \leq 0$, store time step t_i and variance $V(t_i)$.
5. $\delta \geq 0$, $h_1 < 0$ and $h_2 > 0$, we obtain the **negative reversion formula**:

$$\begin{aligned} \hat{V}(t_i) &= h_2^2 \\ \Longrightarrow V(t_i) &= -\hat{V}(t_i) \\ &= -h_2^2 \\ &= -(\frac{-b + \sqrt{\delta}}{2a})^2 \end{aligned} \quad (5.18)$$

5.2.2 Reversion Formula by Full Truncation Technique

Secondly, we introduce the Euler scheme using full truncation technique for the Heston model, see page 227 of Korn [26]:

$$\begin{aligned} V(t_{i+1}) &= V(t_i) + \kappa(\theta - (V(t_i))^+)\Delta t + \sigma\sqrt{(V(t_i))^+}\sqrt{\Delta t}W(t_i) \qquad (5.19) \\ X(t_{i+1}) &= X(t_i) + (r - \frac{1}{2}(V(t_i))^+)\Delta t + \sqrt{(V(t_i))^+}\sqrt{\Delta t}Z(t_i) \qquad (5.20) \end{aligned}$$

Again, as long as we have the volatility $V(t_i)$, the reversion formula for the log-stock price is simple :

$$X(t_i) = X(t_{i+1}) - (r - \frac{1}{2}(V(t_i))^+)\Delta t - \sqrt{|V(t_i)|}\sqrt{\Delta t}Z(t_i) \qquad (5.21)$$

Positive Reversion Formula

If $V(t_i) \geq 0$, we have $(V(t_i))^+ = V(t_i)$, thus equations (5.19) and (5.20) become:

$$\begin{aligned} V(t_{i+1}) &= V(t_i) + \kappa(\theta - V(t_i))\Delta t + \sigma\sqrt{V(t_i)}\sqrt{\Delta t}W(t_i) \qquad (5.22) \\ X(t_{i+1}) &= X(t_i) + (r - \frac{1}{2}V(t_i))\Delta t + \sqrt{V(t_i)}\sqrt{\Delta t}Z(t_i) \qquad (5.23) \end{aligned}$$

The equation (5.22) becomes:

$$V(t_{i+1}) = (1 - \kappa\Delta t)V(t_i) + \sigma\sqrt{\Delta t}W(t_i)\sqrt{V(t_i)} + \kappa\theta\Delta t \qquad (5.24)$$

Define h, a, b and c as follows:

$$\begin{aligned} h &:= \sqrt{V(t_i)} \geq 0 \\ a &:= 1 - \kappa\Delta t \\ b &:= \sigma\sqrt{\Delta t}W(t_i) \\ c &:= \kappa\theta\Delta t \end{aligned}$$

Thus equation (5.24) becomes:

$$V(t_{i+1}) = a \cdot h^2 + b \cdot h + c \qquad (5.25)$$

Define $\delta := b^2 - 4a(c - V(t_{i+1}))$, the solutions of the equation (5.25) are:

$$\begin{aligned} h_1 &= \frac{-b - \sqrt{\delta}}{2a} \\ h_2 &= \frac{-b + \sqrt{\delta}}{2a} \end{aligned}$$

There are five cases to be considered:

1. $\delta < 0$, store time step t_i and variance $V(t_i)$.

2. $\delta \geq 0$, $h_1 \geq 0$ and $h_2 \geq 0$, store time step t_i and variance $V(t_i)$.
3. $\delta \geq 0$, $h_1 \leq 0$ and $h_2 \leq 0$, store time step t_i and variance $V(t_i)$.
4. $\delta \geq 0$, $h_1 \geq 0$ and $h_2 \leq 0$, store time step t_i and variance $V(t_i)$.
5. $\delta \geq 0$, $h_1 < 0$ and $h_2 > 0$, which occurs most often, we obtain the **positive reversion formula**:

$$V(t_i) = h_2^2 \tag{5.26}$$

Negative Reversion Formula

If $V(t_i) < 0$, we have $(V(t_i))^+ = 0$, thus equations (5.19) and (5.20) become:

$$V(t_{i+1}) = V(t_i) + \kappa\theta\Delta t \tag{5.27}$$
$$X(t_{i+1}) = X(t_i) + r\Delta t \tag{5.28}$$

From the equation (5.27), we obtain the negative reversion formula for volatility:

$$V(t_i) = V(t_{i+1}) - \kappa\theta\Delta t \tag{5.29}$$

From the equation (5.28), we obtain the negative reversion formula for log-stock price:

$$X(t_i) = X(t_{i+1}) - r\Delta t \tag{5.30}$$

5.2.3 Test

In this section we test the validity of our reversion formulas within the Heston model either using reflection technique or full truncation technique. We also see how often both positive and negative reversion formulas don't work and variance must be stored.

Input parameters: initial stock price $S_0 = 90$, strike price $K = 100$, maturity $T = 1$, interest rate $r = 0.05$, initial variance $V_0 = 0.04$, speed of mean reversion $\kappa = 3$, long term variance level $\theta = 0.04$, volatility of variance $\sigma \in [0.05, 0.10, 0.15, \ldots, 0.95, 1.00]$, correlation $\rho = -0.1$, number of simulated paths $pathsMC = 10000$, number of time steps per path $stepsMC = 365$.

According to Broadie and Kaya [7], if the parameters obey the stability condition $\frac{2\kappa\theta}{\sigma^2} > 1$, then the variance process V_t is strictly positive.

We denote *HZR:=HiteZeroRate* as the number of paths, when at any certain time step t, the variance process hit zero $V_t \leq 0$, divided by the total simulated paths. Denote *mDS0:=maxDiffStock0* as the maximum of absolute difference between the initial stock price S_0 and $\hat{S}_0$ computed backwards by using the reversion formula, among all simulated paths. Denote *mDV0:=maxDiffVariance0* as maximum of the absolute difference between the initial stock price V_0 and

$\hat{V}_0$ computed backwards by using the reversion formula, among all simulated paths. Denote *mIF:=maxInvFailCount* as the maximum of the counting number, when the reversion formulas fail to work, among all simulated paths. Denote *sIF:=sumInvFailCount* as the sum of all the counting numbers, when the reversion formulas fail to work, among all simulated paths. Denote *mNI:=maxNegInv* as the maximum of the counting number, when the negative reversion formulas works, among all simulated paths. Denote *sNI:=sumNegInv* as the sum of all the counting numbers, when the negative reversion formulas works, among all simulated paths. Denote *sR:=storageRate* how often we must store:

$$storageRate = \frac{sumInvFailCount + sumNegInvCount}{pathsMC \cdot stepsMC} \times 100\%$$

From the Table Table 5.1 and 5.2, we conclude:

1. In practice, even $\frac{2\kappa\theta}{\sigma^2} > 1$, there could still exist variance path which can go down below zero.

2. Inversion formulas both in the case of using reflection technique and using full truncation technique perform well for any case of input parameters, even when the rate of hitting zero is very high.

3. Since Korn [26] points that the full truncation method performs better than the reflection method and numerical results show that *storageRate* for both methods are similar and negative reversion formula for the full truncation technique is even much simpler than for the reflection technique, we suggest to use the full truncation method and the corresponding reversion formulas to reduce the memory of storing variances and stock prices.

σ	$\frac{2\kappa\theta}{\sigma^2}$	HZR	mDS0	mDV0	mIF	sIF	mNI	sNI	sR
0.05	96.00	0%	3.51×10^{-12}	1.58×10^{-15}	0	0	0	0	0%
0.10	24.00	0%	3.60×10^{-12}	3.27×10^{-15}	0	0	0	0	0%
0.15	10.67	0%	4.06×10^{-12}	4.79×10^{-15}	0	0	0	0	0%
0.20	6.00	0%	5.91×10^{-12}	5.99×10^{-15}	0	0	0	0	0%
0.25	3.84	0%	9.22×10^{-11}	6.28×10^{-14}	0	0	0	0	0%
0.30	2.67	0.11%	3.15×10^{-10}	3.79×10^{-13}	3	60	1	10	0.0019%
0.35	1.96	1.57%	1.53×10^{-10}	2.38×10^{-13}	8	789	2	120	0.025%
0.40	1.50	8.27%	1.40×10^{-9}	1.43×10^{-12}	15	4152	4	876	0.14%
0.45	1.19	22.03%	3.76×10^{-11}	1.18×10^{-13}	27	11876	7	2956	0.41%
0.50	0.96	40.01%	4.03×10^{-11}	2.42×10^{-13}	30	25876	13	7455	0.91%
0.55	0.79	56.30%	4.55×10^{-12}	1.52×10^{-14}	55	45421	20	14700	1.65%
0.60	0.67	68.79%	4.39×10^{-12}	4.15×10^{-14}	50	69766	19	24698	2.59%
0.65	0.57	77.18%	1.11×10^{-12}	3.43×10^{-15}	56	94113	23	36241	3.57%
0.70	0.49	83.31%	1.35×10^{-12}	4.60×10^{-15}	72	122813	26	50266	4.74%
0.75	0.43	87.61%	1.19×10^{-12}	1.98×10^{-15}	74	149373	31	64851	5.87%
0.80	0.38	90.66%	8.81×10^{-13}	1.75×10^{-15}	71	175200	32	79630	6.98%
0.85	0.33	92.75%	9.52×10^{-13}	1.94×10^{-15}	73	196378	33	93117	7.93%
0.90	0.30	93.97%	1.11×10^{-12}	1.62×10^{-15}	86	220877	41	108400	9.02%
0.95	0.27	95.14%	1.04×10^{-12}	1.41×10^{-15}	88	241513	43	122041	9.96%
1.00	0.24	96.07%	1.76×10^{-12}	1.70×10^{-15}	87	260510	43	134845	10.83%

Table 5.1: Test of validity for reversion formulas using the reflection technique

σ	$\frac{2\kappa\theta}{\sigma^2}$	HZR	mDS0	mDV0	mIF	sIF	mNI	sNI	sR
0.05	96.00	0%	2.78×10^{-12}	1.67×10^{-15}	0	0	0	0	0%
0.10	24.00	0%	3.27×10^{-12}	1.82×10^{-15}	0	0	0	0	0%
0.15	10.67	0%	3.75×10^{-12}	5.59×10^{-15}	0	0	0	0	0%
0.20	6.00	0%	4.63×10^{-12}	7.57×10^{-15}	0	0	0	0	0%
0.25	3.84	0%	3.05×10^{-11}	6.52×10^{-14}	0	0	0	0	0%
0.30	2.67	0.12%	4.75×10^{-11}	2.82×10^{-13}	6	84	2	11	0.0026%
0.35	1.96	1.65%	3.03×10^{-10}	9.31×10^{-14}	8	832	5	228	0.029%
0.40	1.50	7.99%	6.25×10^{-10}	1.47×10^{-12}	15	3853	10	144	0.15%
0.45	1.19	22.65%	4.85×10^{-11}	7.10×10^{-13}	24	12439	17	6105	0.51%
0.50	0.96	39.45%	8.26×10^{-12}	6.97×10^{-14}	33	26156	34	16456	1.17%
0.55	0.79	57.00%	4.32×10^{-12}	5.57×10^{-14}	40	48241	36	37168	2.34%
0.60	0.67	68.12%	2.62×10^{-12}	8.92×10^{-15}	52	72413	52	66413	3.80%
0.65	0.57	77.82%	1.68×10^{-12}	2.57×10^{-15}	57	102531	64	109824	5.82%
0.70	0.49	83.23%	1.19×10^{-12}	2.89×10^{-15}	63	132299	87	163645	8.11%
0.75	0.43	87.26%	9.52×10^{-13}	1.94×10^{-15}	72	161189	104	224570	10.57%
0.80	0.38	90.63%	1.44×10^{-12}	1.73×10^{-15}	76	192482	123	300668	13.51%
0.85	0.33	92.59%	1.35×10^{-12}	1.69×10^{-15}	73	216284	135	376743	16.25%
0.90	0.30	94.19%	1.19×10^{-12}	1.70×10^{-15}	74	239963	154	459367	19.16%
0.95	0.27	95.02%	1.03×10^{-12}	1.88×10^{-15}	81	261153	156	547256	22.15%
1.00	0.24	95.68%	9.52×10^{-13}	1.19×10^{-15}	76	275502	172	631484	24.85%

Table 5.2: Test of validity for reversion formulas using the full truncation technique

6 Conclusion

The contribution of this dissertation is fourfold:

1. We comprehensively study the JR tree, CRR tree, BEG tree, KM tree and RSS tree for pricing one-dimensional American options, multidimensional American options, American-style -lookback options, -Asian options, -barrier options, -basket options, -strangle-spread-payoff options within the Black-Scholes model and the Heston model and deliver good benchmarks by increasing the number of tree steps to a very huge number. We collect all benchmarks for 24 test cases in the Appendix and make use of these benchmarks to compare with the results by Monte Carlo methods. Our benchmarks can be definitely valuable for other researchers when investigating efficiency of numerical methods for valuing American-style options.

2. We investigate systematically the regression-based Monte Carlo methods. We compare the Longstaff-Schwartz method, the Tsitsiklis-Roy method, their modified variations using all-paths for regression or using only in-the-money paths for regression, their modified variations using out-of-samples new paths to value option presenting lower bound, their modified variations using the Andersen-Broadie method presenting upper bound, by a variety of number and form of basis functions in a lot of test cases. We also test the stability of orthogonal polynomials as basis functions for several multidimensional American options.

3. We study two machine learning techniques to improve the regression part of the Monte Carlo methods: the kernel method and the support vector machine. For the kernel method, we test its variations of fixed bandwidth, global optimal bandwidth and suboptimal bandwidth by data scaling and parameter selection techniques. The kernel method with suboptimal bandwidth works much quicker than the one with global searching and performs robust in all 24 test cases and sometimes even better than the Longstaff-Schwartz method and the Tsitsiklis-Roy method, especially when the payoff is strange and the dimension of the option is high. The support vector machine can improve the kernel method by selecting only a subset of all old stock prices to predict for the option continuation value for the new stock price and thus can reduce the storage of stock prices during training and decrease the run time during prediction.

4. We also work with the electronic engineering group to design the embedded architecture for the Longstaff-Schwartz method for pricing high dimen-

sional American options on FPGA. Based on our novel reverse formula for the stock prices, we don't have to store the intermediate stock prices in external memories with lower speed and more energy consumption and can make full use of FPGA , which only has a limited memory but works much quicker and consumes less energy.

7 Appendix

7.1 Benchmarks

In this section, we present numerical experiments using different trees to price various American-style options. Denote $\mathbb{Q}$ as the risk-neutral measure, S_0 as the initial stock price, K as the strike price, T as the maturity, r as the interest rate, δ as the dividend, σ as the volatility, B as the barrier, $\mathcal{T}[0,T]$ as the set of stopping times taking values in $[0,T]$, T_{ex} as a potential exercise date.

For high dimensional options, we denote Σ as the variance-covariance matrix and ρ as the correlation coefficient of the Brownian motions.

Further, we denote *inputs* as values for input parameters above, *output* as the American-style option prices and *reference* as the corresponding European-style option prices.

When we price one-dimensional American geometric-average Asian options or high-dimensional American geometric-average basket options, we notice that we can always simplify the trees to normal 1-D CRR tree using formulas (2.21) - (2.23) or using formulas (2.26) - (2.28), in order to achieve higher accuracy.

7.1.1 1-D Examples in the Black-Scholes Model

Test Case 1: 1-D American option
Optimal expected discounted payoff is:

$$\sup_{T_{ex}\in\mathcal{T}[0,T]} \mathbb{E}^{\mathbb{Q}}\left[e^{-rT_{ex}}\left(K - S(T_{ex})\right)^{+}\right]$$

Inputs: $S_0 = 36$, $K = 40$, $T = 1$, $r = 0.06$, $\delta = 0$, $\sigma = 0.4$.
Output: 7.11 Reference: 6.71
See selected option prices in Table 7.1 and convergence behaviour in Figure 7.1.

number of steps	100	101	200	201	500	501
option price	7.1190	7.1109	7.1091	7.1144	7.1082	7.1114
number of steps	1000	1001	5000	5001	10000	10001
option price	7.1094	7.1099	7.1092	7.1091	7.1090	7.1091
number of steps	20000	20001	40000	40001	100000	100001
option price	7.1090	7.1090	7.1090	7.1090	7.1090	7.1090

Table 7.1: Selected option prices for 1-D American option

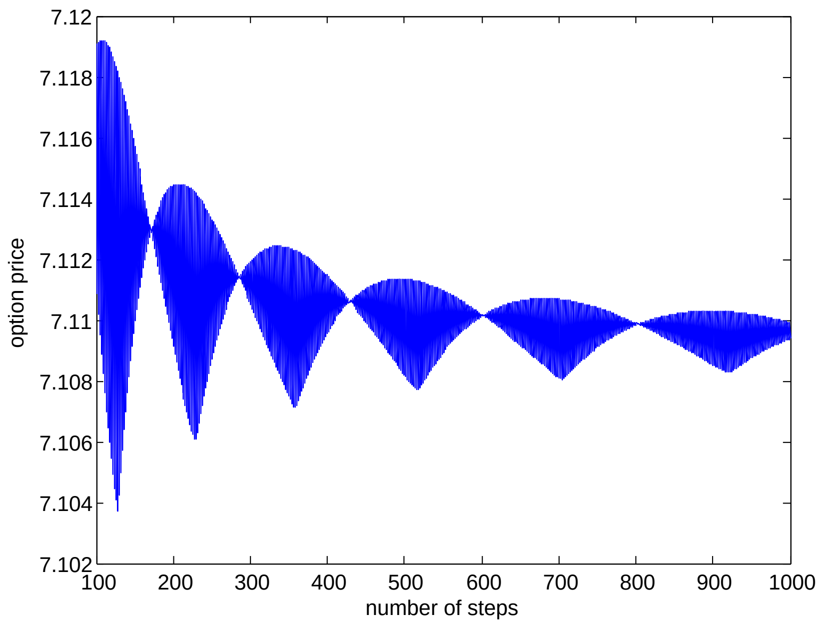


Figure 7.1: CRR tree for 1-D American option

Test Case 2: 1-D Bermudan option
Optimal expected discounted payoff is:

$$\sup_{T_{ex}\in\mathcal{T}} \mathbb{E}^{\mathbb{Q}}\left[e^{-rT_{ex}}\left(K - S(T_{ex})\right)^{+}\right]$$

Inputs: $S_0 = 100$, $K = 90$, $T = 1$, $r = 0.05$, $\delta = 0$, $\sigma = 0.25$, $\mathcal{T} = \mathcal{T}\{\frac{T}{12} \times 1, \frac{T}{12} \times 2, \ldots \frac{T}{12} \times 12\}$
Output: 3.931 Reference: 3.75
See selected option prices in Table 7.2 and convergence behaviour in Figure 7.2.

number of steps	100	101	200	201	500	501
option price	3.9279	3.9475	3.9253	3.9415	3.9340	3.9327
number of steps	1000	1001	5000	5001	10000	10001
option price	3.9327	3.9318	3.9313	3.9318	3.9313	3.9316
number of steps	20000	20001	40000	40001	100000	100001
option price	3.9314	3.9315	3.9314	3.9315	3.9314	3.9314

Table 7.2: Selected option prices for 1-D Bermudan option

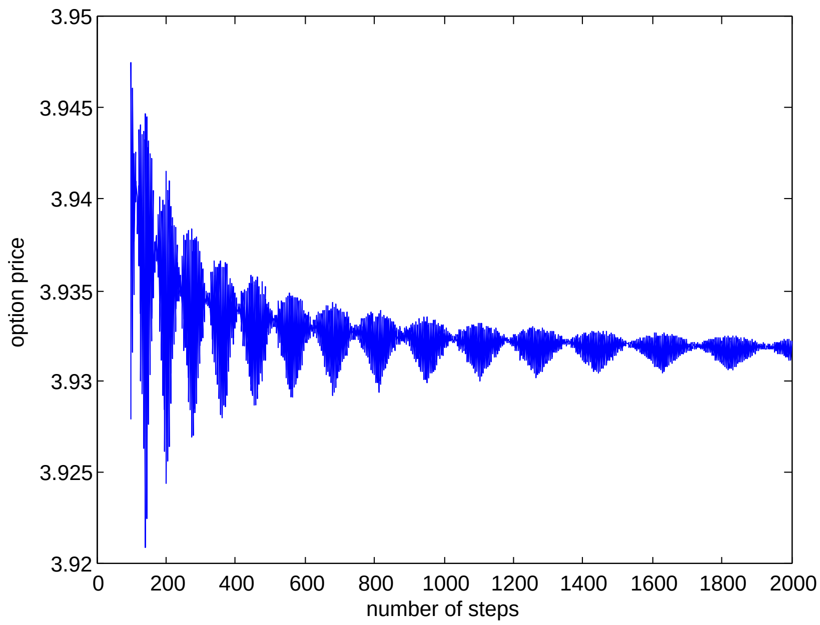


Figure 7.2: CRR tree for 1-D Bermudan option

Test Case 3: 1-D Bermudan option with only two exercise dates
Optimal expected discounted payoff is:

$$\sup_{T_{ex}\in\mathcal{T}} \mathbb{E}^{\mathbb{Q}}\left[e^{-rT_{ex}}\left(K - S(T_{ex})\right)^{+}\right]$$

Inputs: $S_0 = 100$, $K = 100$, $T = 1$, $r = 0.1$, $\delta = 0$, $\sigma = 0.2$, $\mathcal{T} = \mathcal{T}\{\frac{T}{2} \times 1, \frac{T}{2} \times 2\}$
Output: 4.313 Reference: 3.75
See selected option prices in Table 7.3 and convergence behaviour in Figure 7.3.

number of steps	100	101	200	201	500	501
option price	4.3084	4.3126	4.3046	4.3216	4.3120	4.3149
number of steps	1000	1001	5000	5001	10000	10001
option price	4.3125	4.3146	4.3133	4.3135	4.3132	4.3136
number of steps	20000	20001	40000	40001	100000	100001
option price	4.3133	4.3135	4.3133	4.3134	4.3134	4.3134

Table 7.3: Selected option prices for 1-D Bermudan option with only two exercise dates

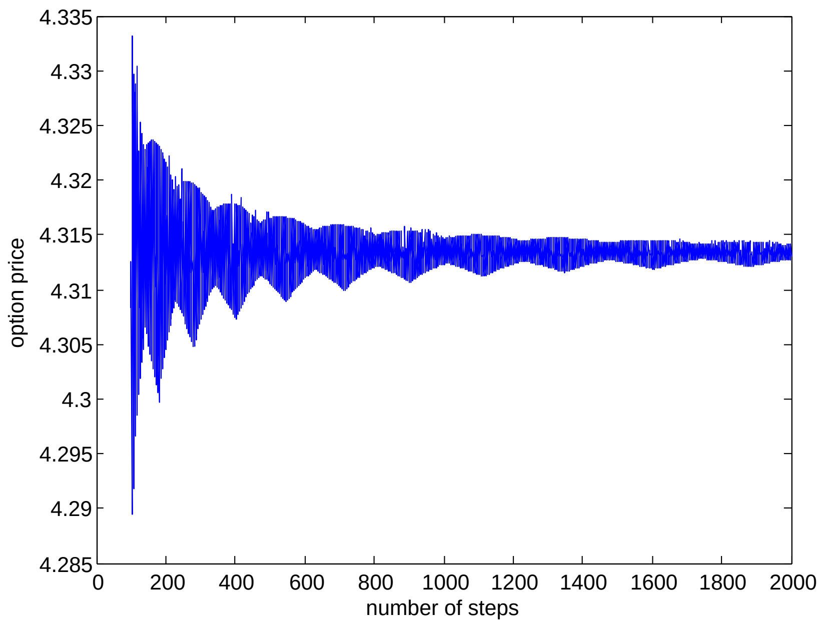


Figure 7.3: CRR tree for 1-D Bermudan option with only two exercise dates

Test Case 4: 1-D American option with strangle-spread-payoff
Optimal expected discounted payoff is:

$$\sup_{T_{ex}} \mathbb{E}^{\mathbb{Q}} \left[e^{-rT_{ex}} \left((K_2 - K_1)^+ \mathbb{1}_{\{S(T_{ex})<K_1\}} + (K_2 - S(T_{ex}))^+ \mathbb{1}_{\{K_1 \le S(T_{ex}) \le K_2\}} \right.\right.$$
$$\left.\left. +0 \cdot \mathbb{1}_{\{K_2<S(T_{ex})<K_3\}} + (S(T_{ex}) - K_3)^+ \mathbb{1}_{\{K_3 \le S(T_{ex}) \le K_4\}} + (K_4 - K_3)^+ \mathbb{1}_{\{S(T_{ex})>K_4\}} \right) \right]$$

Inputs: $S_0 = 100$, $T = 1$, $r = 0.05$, $\delta = 0$, $\sigma = 0.5$, $K_1 = 50$, $K_2 = 90$, $K_3 = 110$, $K_4 = 150$. $\mathcal{T} = \mathcal{T}\{\frac{T}{48} \times 1, \frac{T}{48} \times 2, \ldots, \frac{T}{48} \times 48\}$
Output: 26.32 Reference: 20.70
See selected option prices in Table 7.4 and convergence behaviour in Figure 7.4.

number of steps	48	96	240	480	720	960
option price	26.5336	26.8897	26.3631	26.3762	26.3380	26.3074
number of steps	4800	4848	7200	7248	9600	9648
option price	26.3278	26.3197	26.3164	26.3200	26.3186	26.3189
number of steps	48000	48048	72000	72048	96000	96048
option price	26.3179	26.3177	26.3178	26.3178	26.3176	26.3177

Table 7.4: Selected option prices for 1-D American option with strangle-spread-payoff

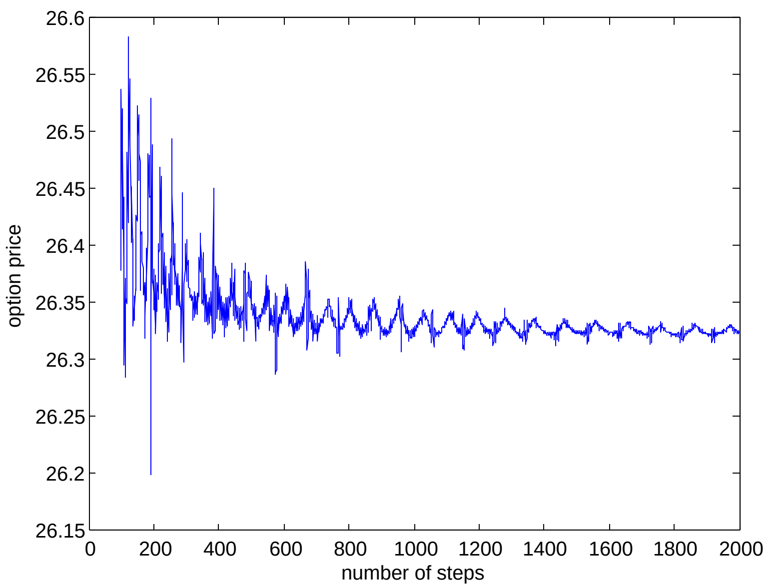


Figure 7.4: CRR tree for 1-D American option with strangle-spread-payoff

Test Case 5: 1-D American lookback option with floating strike
Optimal expected discounted payoff is:

$$\sup_{T_{ex}\in\mathcal{T}[0,T]} \mathbb{E}^{\mathbb{Q}}\left[e^{-rT_{ex}}\left(\max_{t\in[0,T_{ex}]} S(t) - S(T_{ex})\right)^{+}\right]$$

Inputs: $S_0 = 50,\ T = \frac{1}{4},\ r = 0.10,\ \delta = 0,\ \sigma = 0.4.$
Output: 7.81 Reference: 7.61
See selected option prices in Table 7.5 and convergence behaviour in Figure 7.5.

number of steps	5	10	20	50	80	90
option price	5.9186	6.4333	6.8369	7.2288	7.3786	7.4115
number of steps	100	101	200	201	400	401
option price	7.4396	7.4422	7.5943	7.5953	7.7067	7.7071
number of steps	600	601	800	801	1000	1001
option price	7.7574	7.7575	7.7878	7.7879	7.8086	7.8087

Table 7.5: Selected option prices for 1-D American lookback option with floating strike

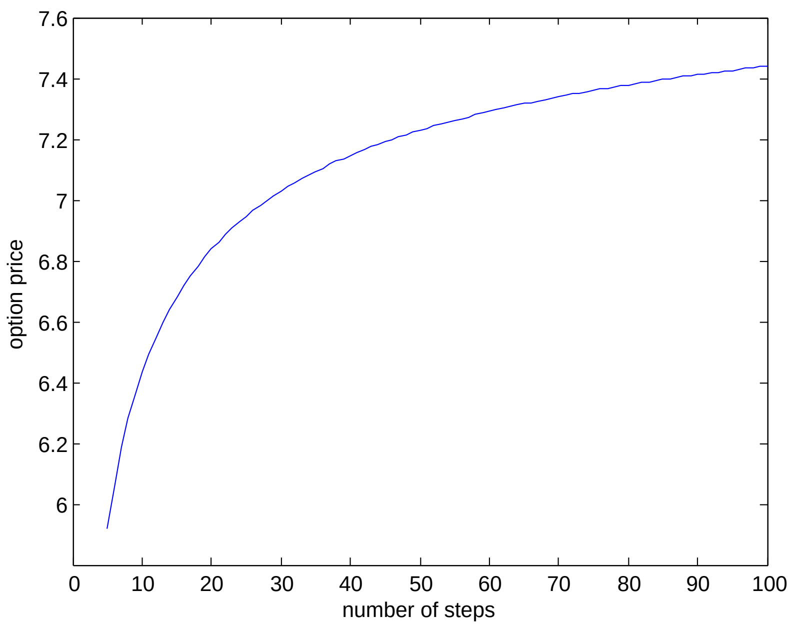


Figure 7.5: CRR tree for 1-D American lookback option with floating strike

Test Case 6: 1-D American knock-out barrier option
Optimal expected discounted payoff is:

$$\sup_{T_{ex}\in\mathcal{T}[0,T]} \mathbb{E}^{\mathbb{Q}}\left[e^{-rT_{ex}}\left(S(T_{ex})-K\right)^{+}\mathbb{1}_{\{\max_{t\in[0,T_{ex}]} S(t)<B\}}\right]$$

Inputs: $S_0 = 100$, $K = 80$, $T = 1$, $r = 0.05$, $\delta = 0$, $\sigma = 0.2$, $B = 120$.
Output: 23.77 Reference: 7.74
See selected option prices in Table 7.6 and convergence behaviour in Figure 7.6.

number of steps	3	10	20	50	80	90
option price	23.3371	23.2426	23.7925	23.6636	23.7433	23.6633
number of steps	100	101	1000	1001	2000	2001
option price	23.7663	23.7503	23.7335	23.7322	23.7482	23.7483
number of steps	4000	4001	6000	6001	8000	8001
option price	23.7600	23.7598	23.7641	23.7640	23.7672	23.7672

Table 7.6: Selected option prices for 1-D American knock-out barrier option

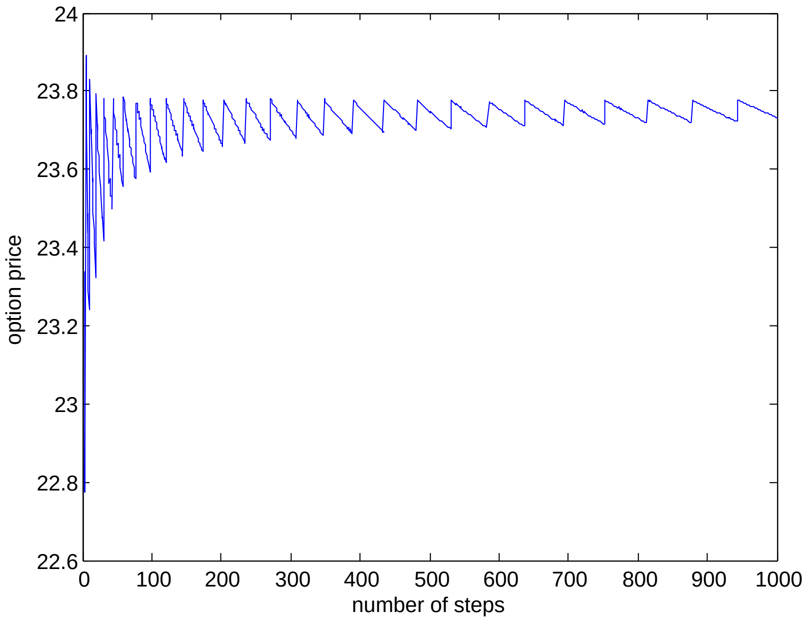


Figure 7.6: CRR tree for 1-D American knock-out barrier option

Test Case 7: 1-D American knock-in barrier option
Optimal expected discounted payoff is:

$$\sup_{T_{ex}\in\mathcal{T}[0,T]} \mathbb{E}^{\mathbb{Q}}\left[e^{-rT_{ex}}\left(K-S(T_{ex})\right)^{+}\mathbb{1}_{\{\max\limits_{t\in[0,T_{ex}]} S(t)\geq B\}}\right]$$

Inputs: $S_0 = 100$, $K = 95$, $T = 1$, $r = 0.05$, $\delta = 0$, $\sigma = 0.2$, $B = 80$.
Output: 4.01 Reference: 3.71
See selected option prices in Table 7.7 and convergence behaviour in Figure 7.7.

number of steps	5	10	20	50	80	90
option price	4.1834	4.1152	4.0817	4.0135	4.0121	4.0170
number of steps	100	101	200	201	400	401
option price	4.0202	4.0209	4.0163	4.0177	4.0166	4.0120
number of steps	600	601	800	801	1000	1001
option price	4.0132	4.0152	4.0147	4.0131	4.0126	4.0145

Table 7.7: Selected option prices for 1-D American knock-in barrier option

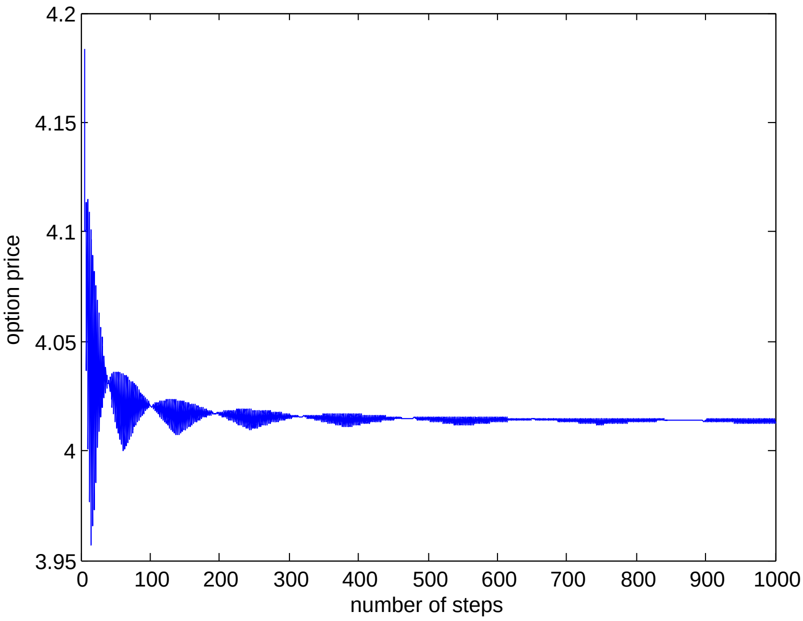


Figure 7.7: CRR tree for 1-D American knock-in barrier option

Test Case 8: 1-D American geometric-average Asian option
Optimal expected discounted payoff is:

$$\sup_{T_{ex}\in\mathcal{T}[0,T]} \mathbb{E}^{\mathbb{Q}}\left[e^{-rT_{ex}}((\prod_{\substack{i=1\\t_n=T_{ex}}}^{n} S(t_i))^{\frac{1}{n}} - K)^{+}\right]$$

Inputs: $S_0 = 50$, $K = 50$, $T = 1$, $r = 0.1$, $\delta = 0.2$, $\sigma = 0.4$.
Output: 3.25 Reference: 2.79
See selected option prices in Table 7.8 and convergence behaviour in Figure 7.8.

number of steps	5	10	20	50	80	90
option price	3.3594	3.1974	3.2272	3.2431	3.2461	3.2470
number of steps	100	101	200	201	400	401
option price	3.2473	3.2574	3.2498	3.2549	3.2510	3.2535
number of steps	600	601	800	801	1000	1001
option price	3.2514	3.2530	3.2516	3.2528	3.2517	3.2527

Table 7.8: Selected option prices for 1-D American geometric-average Asian option

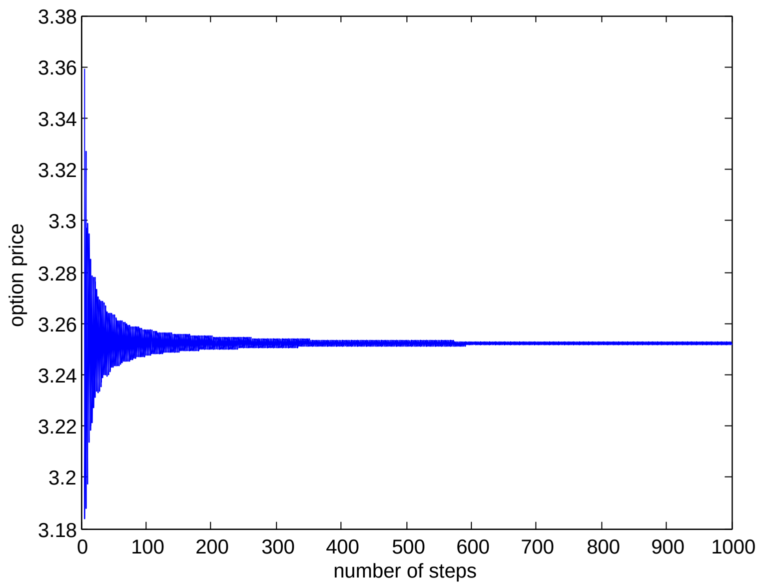


Figure 7.8: CRR tree for 1-D American geometric-average Asian option

7.1.2 2-D Examples in the Black-Scholes Model

Test Case 9: 2-D American spread option
Optimal expected discounted payoff is:

$$\sup_{T_{ex}\in\mathcal{T}} \mathbb{E}^{\mathbb{Q}}\left[e^{-rT_{ex}}([S_1(T_{ex})-S_2(T_{ex})]-K)^+\right]$$

Inputs: $S_1(0) = 100, S_2(0) = 90,\ T = 3,\ r = 0.05,\ \delta = 0.1,\ \rho = (1\ 0.1;\ 0.1\ 1)$, $\Sigma = (0.04\ 0.002;\ 0.002\ 0.01)$, $\mathcal{T} = \mathcal{T}\{\frac{T}{9}\times 1, \frac{T}{9}\times 2, \ldots, \frac{T}{9}\times 9\}$
In this example, we discuss the option prices for three cases: at the money, in the money and out of the money.
See selected option prices using KM tree in Table 7.9.

At the Money: $K = 10$						
number of steps	5	10	20	50	80	90
option price	10.8037	11.4481	11.4110	11.4063	11.4057	11.4048
number of steps	100	101	200	201	400	401
option price	11.4041	11.4038	11.4022	11.4025	11.4015	11.4017
Output: 11.40 Reference: 10.21						
In the Money: $K = 1$						
number of steps	5	10	20	50	80	90
option price	14.7841	15.8455	15.7972	15.7909	15.7916	15.7915
number of steps	100	101	200	201	400	401
option price	15.7895	15.7856	15.7853	15.7863	15.7845	15.7854
Output: 15.78 Reference: 14.00						
Out of the Money: $K = 30$						
number of steps	5	10	20	50	80	90
option price	4.9680	5.1723	5.1850	5.1930	5.1956	5.1957
number of steps	100	101	200	201	400	401
option price	5.1960	5.1962	5.1978	5.1979	5.1986	5.1986
Output: 5.20 Reference: 4.75						

Table 7.9: Selected option prices for 2-D American spread option

Test Case 10: 2-D American maximum outperformance option
Optimal expected discounted payoff is:

$$\sup_{T_{ex}\in\mathcal{T}} \mathbb{E}^{\mathbb{Q}}\left[e^{-rT_{ex}}(\max\{S_1(T_{ex}), S_2(T_{ex})\} - K)^+\right]$$

Inputs: $S_1(0) = S_2(0) = S$, $K = 100$, $T = 3$, $r = 0.05$, $\delta = 0.1$, $\rho = (1\ 0;\ 0\ 1)$, $\Sigma = (0.04\ 0;\ 0\ 0.04)$, $\mathcal{T} = \mathcal{T}\{\frac{T}{9}\times 1, \frac{T}{9}\times 2, \ldots, \frac{T}{9}\times 9\}$
In this example, we discuss the option prices for three cases: at the money, in the money and out of the money.
See selected option prices using KM tree in Table 7.10.

At the Money: $S = 100$						
number of steps	5	10	20	50	80	90
option price	11.9666	13.6121	13.8928	13.8686	13.9011	13.8852
number of steps	100	101	200	201	400	401
option price	13.8708	13.9045	13.8994	13.8982	13.8953	13.9027
Output: 13.90 Reference: 11.19						
In the Money: $S = 110$						
number of steps	5	10	20	50	80	90
option price	18.5304	21.3772	21.3213	21.3436	21.3629	21.3666
number of steps	100	101	200	201	400	401
option price	21.3530	21.3401	21.3476	21.3423	21.3444	21.3452
Output: 21.34 Reference: 16.93						
Out of the Money: $S = 70$						
number of steps	5	10	20	50	80	90
option price	1.2238	1.5455	1.5876	1.6093	1.6262	1.6303
number of steps	100	101	200	201	400	401
option price	1.6236	1.6306	1.6380	1.6303	1.6408	1.6388
Output: 1.64 Reference: 1.44						

Table 7.10: Selected option prices for 2-D American maximum outperformance option

Test Case 11: 2-D American minimum outperformance option
Optimal expected discounted payoff is:

$$\sup_{T_{ex}\in\mathcal{T}} \mathbb{E}^{\mathbb{Q}}\left[e^{-rT_{ex}}(\min\{S_1(T_{ex}), S_2(T_{ex})\} - K)^+\right]$$

Inputs: $S_1(0) = S_2(0) = S$, $K = 100$, $T = 3$, $r = 0.05$, $\delta = 0.1$, $\rho = (1\ 0;\ 0\ 1)$, $\Sigma = (0.04\ 0;\ 0\ 0.04)$, $\mathcal{T} = \mathcal{T}\{\frac{T}{9}\times 1, \frac{T}{9}\times 2, \ldots, \frac{T}{9}\times 9\}$
In this example, we discuss the option prices for three cases: at the money, in the money and out of the money.
See selected option prices using KM tree in Table 7.11.

At the Money: $S = 100$						
number of steps	5	10	20	50	80	90
option price	1.4649	3.2708	2.8430	2.4518	2.3687	2.3554
number of steps	100	101	200	201	400	401
option price	2.3504	2.3642	2.3108	2.3082	2.2870	2.2868
Output: 2.28			Reference: 0.85			
In the Money: $S = 110$						
number of steps	5	10	20	50	80	90
option price	3.0158	7.6047	6.7272	6.1439	6.0969	6.1213
number of steps	100	101	200	201	400	401
option price	6.0957	6.0929	6.0349	6.0347	5.9896	5.9749
Output: 5.97			Reference: 1.82			
Out of the Money: $S = 70$						
number of steps	5	10	20	50	80	90
option price	0.0345	0.0363	0.0373	0.0305	0.0300	0.0301
number of steps	100	101	200	201	400	401
option price	0.0293	0.0299	0.0295	0.0290	0.0291	0.0290
Output: 0.029			Reference: 0.019			

Table 7.11: Selected option prices for 2-D American minimum outperformance option

Test Case 12: 2-D American geometric-average basket option
Optimal expected discounted payoff is:

$$\sup_{T_{ex}\in\mathcal{T}} \mathbb{E}^{\mathbb{Q}}\left[e^{-rT_{ex}}((\prod_{i=1}^{2} S_i(T_{ex}))^{\frac{1}{2}} - K)^{+}\right]$$

Inputs: $S_1(0) = 22, S_2(0) = 20,\ K = 20,\ T = 1,\ r = 0.1,\ \delta = 0.15,\ \rho = (1\ 0.5;\ 0.5\ 1)$, $\Sigma = (0.04\ 0.025;\ 0.025\ 0.0625)$, $\mathcal{T} = \mathcal{T}\{\frac{T}{5} \times 1, \frac{T}{5} \times 2, \ldots, \frac{T}{5} \times 5\}$
Output: 1.55 Reference: 1.32
See selected option prices in Table 7.12 and convergence behaviour in Figure 7.9.

number of steps	100	101	200	201	500	501
option price	1.5506	1.5504	1.5496	1.5494	1.5484	1.5483
number of steps	1000	1001	5000	5001	10000	10001
option price	1.5480	1.5482	1.5479	1.5479	1.5479	1.5480
number of steps	20000	20001	40000	40001	100000	100001
option price	1.5479	1.5479	1.5479	1.5479	1.5479	1.5479

Table 7.12: Selected option prices for 2-D American geometric-average basket option

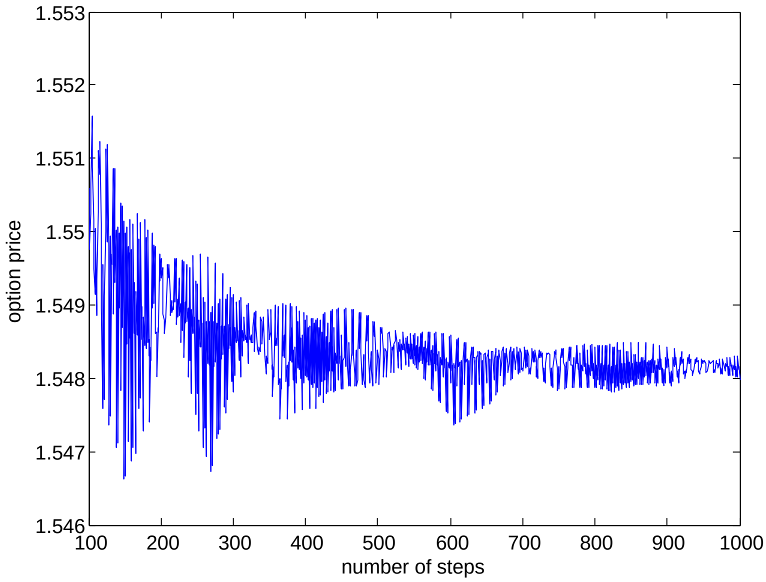


Figure 7.9: CRR tree for 2-D American geometric-average basket option

Test Case 13: 2-D American geometric-average basket option with discontinue payoff
Optimal expected discounted payoff is:

$$\sup_{T_{ex}\in\mathcal{T}} \mathbb{E}^{\mathbb{Q}}\left[e^{-rT_{ex}}(G-K)^{+}\mathbb{1}_{\{G\leq B_1 \text{ or } G\geq B_2\}}\right] \text{ with } G=(\prod_{i=1}^{2} S_i(T_{ex}))^{\frac{1}{2}}$$

Inputs: $S_1(0)=22, S_2(0)=20,\ K=20,\ B_1=25,\ B_2=30$, $T=1,\ r=0.1,\ \delta=0.15,$ $\rho=(1\ 0.5;\ 0.5\ 1),\ \Sigma=(0.04\ 0.025;\ 0.025\ 0.0625),\ \mathcal{T}=\mathcal{T}\{\frac{T}{5}\times 1, \frac{T}{5}\times 2, \ldots, \frac{T}{5}\times 5\}$
Output: 1.48   Reference: 1.32
See selected option prices in Table 7.13 and convergence behaviour in Figure 7.10.

number of steps	100	101	200	201	500	501
option price	1.4820	1.5028	1.4951	1.4814	1.4960	1.4871
number of steps	1000	1001	5000	5001	10000	10001
option price	1.4900	1.4839	1.4802	1.4815	1.4793	1.4803
number of steps	20000	20001	40000	40001	100000	100001
option price	1.4820	1.4827	1.4808	1.4818	1.4832	1.4825

Table 7.13: Selected option prices for 2-D American geometric-average basket option with discontinue payoff

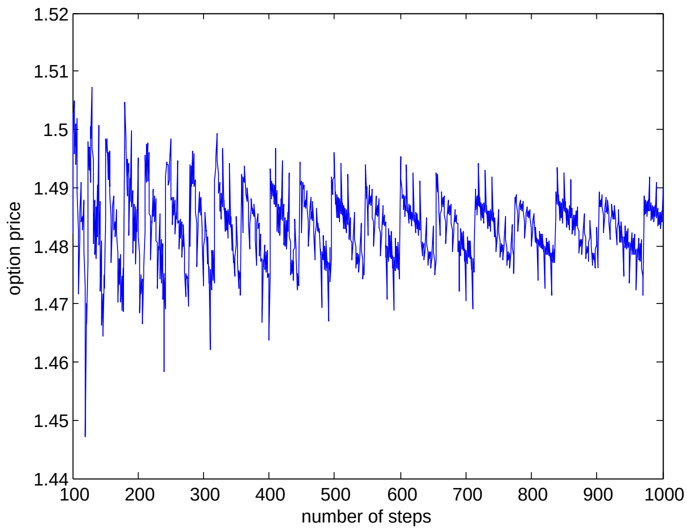


Figure 7.10: CRR tree for 2-D American geometric-average basket option with discontinue payoff

Test Case 14: 2-D American geometric-average basket option with strangle-spread-payoff

Optimal expected discounted payoff is:

$$\sup_{T_{ex}\in\mathcal{T}} \mathbb{E}^{\mathbb{Q}}\left[e^{-rT_{ex}}\left((K_2-K_1)^+\mathbb{1}_{\{G<K_1\}}+(K_2-G)^+\mathbb{1}_{\{K_1\leq G\leq K_2\}}+0\cdot\mathbb{1}_{\{K_2<G<K_3\}}\right.\right.$$
$$\left.\left.+(G-K_3)^+\mathbb{1}_{\{K_3\leq G\leq K_4\}}+(K_4-K_3)^+\mathbb{1}_{\{G>K_4\}}\right)\right] \text{ with } G=\left(\prod_{i=1}^{2}S_i(T_{ex})\right)^{\frac{1}{2}}$$

Inputs: $S_1(0)=22, S_2(0)=20,\ K_1=15,\ K_2=20,\ K_3=30,\ K_4=50,\ T=1,\ r=0.1,$ $\delta=0.15,\ \rho=(1\ 0.5;\ 0.5\ 1),\ \Sigma=(0.04\ 0.025;\ 0.025\ 0.0625),\ \mathcal{T}=\mathcal{T}\{\frac{T}{5}\times1,\frac{T}{5}\times2,\ldots,\frac{T}{5}\times5\}$

Output: 1.46 Reference: 1.40

See selected option prices in Table 7.14 and convergence behaviour in Figure 7.11.

number of steps	100	101	200	201	500	501
option price	1.4613	1.4636	1.4622	1.4614	1.4613	1.4608
number of steps	1000	1001	5000	5001	10000	10001
option price	1.4607	1.4609	1.4607	1.4606	1.4607	1.4607
number of steps	20000	20001	40000	40001	100000	100001
option price	1.4607	1.4607	1.4606	1.4606	1.4606	1.4606

Table 7.14: Selected option prices for 2-D American geometric-average basket option with strangle-spread-payoff

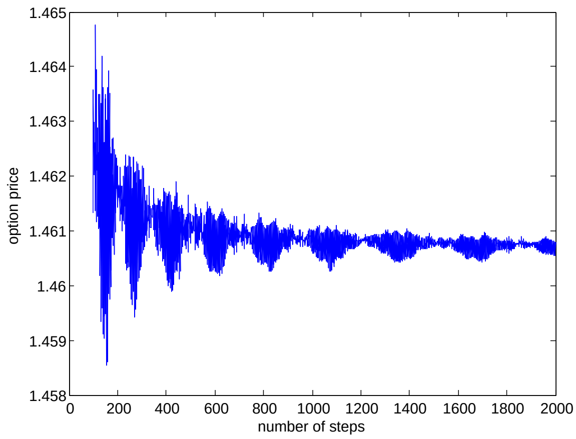


Figure 7.11: CRR tree for 2-D American geometric-average basket option with strangle-spread-payoff

7.1.3 3-D Examples in the Black-Scholes Model

Test Case 15: 3-D American maximum outperformance option
Optimal expected discounted payoff is:

$$\sup_{T_{ex}\in\mathcal{T}} \mathbb{E}^{\mathbb{Q}}\left[e^{-rT_{ex}}(\max\{S_1(T_{ex}),S_2(T_{ex}),S_3(T_{ex})\}-K)^+\right]$$

Inputs: $S_1(0) = S_2(0) = S_3(0) = S$, $K = 100$, $T = 3$, $r = 0.05$, $\delta = 0.1$, $\sigma_1 = \sigma_2 = \sigma_3 = 0.2$, $\rho = (1\ \ -0.25\ \ 0.25;\ \ -0.25\ \ 1\ \ 0.3;\ \ 0.25\ \ 0.3\ \ 1)$, $\Sigma = (0.04\ \ -0.01\ 0.01;\ -0.01\ 0.04\ 0.012;\ 0.01\ 0.012\ 0.04)$, $\mathcal{T} = \mathcal{T}\{\frac{T}{5}\times 1, \frac{T}{5}\times 2, \ldots, \frac{T}{5}\times 5\}$
In this example, we discuss the option prices for three cases: at the money, in the money and out of the money.
See selected option prices using KM tree in Table 7.15.

At the Money: $S = 100$						
number of steps	5	10	20	30	40	50
option price	17.0709	17.2680	17.4709	17.4879	17.4785	17.4881
number of steps	60	70	80	90	100	101
option price	17.5016	17.5073	17.5062	17.5004	17.4965	17.5101
Output: 17.50 Reference: 14.92						
In the Money: $S = 110$						
number of steps	5	10	20	30	40	50
option price	25.7732	25.9850	25.9712	25.9576	25.9746	25.9778
number of steps	60	70	80	90	100	101
option price	25.9701	25.9852	25.9900	25.9873	25.9833	25.9716
Output: 25.98 Reference: 22.06						
Out of the Money: $S = 70$						
number of steps	5	10	20	30	40	50
option price	1.8183	2.1189	2.1889	2.2270	2.2325	2.2464
number of steps	60	70	80	90	100	101
option price	2.2532	2.2565	2.2611	2.2658	2.2635	2.2698
Output: 2.27 Reference: 2.05						

Table 7.15: Selected option prices for 3-D American maximum outperformance option

Test Case 16: 3-D American minimum outperformance option
Optimal expected discounted payoff is:

$$\sup_{T_{ex}\in\mathcal{T}} \mathbb{E}^{\mathbb{Q}}\left[e^{-rT_{ex}}(\min\{S_1(T_{ex}), S_2(T_{ex}), S_3(T_{ex})\} - K)^+\right]$$

Inputs: $S_1(0) = S_2(0) = S_3(0) = S$, $K = 100$, $T = 3$, $r = 0.05$, $\delta = 0.1$, $\sigma_1 = \sigma_2 = \sigma_3 = 0.2$, $\rho = (1\ -0.25\ 0.25;\ -0.25\ 1\ 0.3;\ 0.25\ 0.3\ 1)$, $\Sigma = (0.04\ -0.01\ 0.01;\ -0.01\ 0.04\ 0.012;\ 0.01\ 0.012\ 0.04)$, $\mathcal{T} = \mathcal{T}\{\frac{T}{5}\times 1, \frac{T}{5}\times 2, \ldots, \frac{T}{5}\times 5\}$
In this example, we discuss the option prices for three cases: at the money, in the money and out of the money.
See selected option prices using KM tree in Table 7.11.

At the Money: $S = 100$						
number of steps	5	10	20	30	40	50
option price	1.1991	0.9966	0.8853	0.8530	0.8309	0.8180
number of steps	60	70	80	90	100	101
option price	0.8142	0.8113	0.8086	0.8060	0.8042	0.8095
Output: 0.81 Reference: 0.24						
In the Money: $S = 110$						
number of steps	5	10	20	30	40	50
option price	3.2312	3.2485	2.9919	2.8994	2.8590	2.8440
number of steps	60	70	80	90	100	101
option price	2.8364	2.8316	2.8275	2.8232	2.8189	2.8279
Output: 2.82 Reference: 0.65						
Out of the Money: $S = 70$						
number of steps	5	10	20	30	40	50
option price	0	0.0021	0.0023	0.0020	0.0019	0.0021
number of steps	60	70	80	90	100	101
option price	0.0022	0.0022	0.0022	0.0022	0.0021	0.0022
Output: 0.0022 Reference: 0.0015						

Table 7.16: Selected option prices for 3-D American minimum outperformance option

Test Case 17: 3-D American geometric-average basket option
Optimal expected discounted payoff is:

$$\sup_{T_{ex}\in\mathcal{T}} \mathbb{E}^{\mathbb{Q}}\left[e^{-rT_{ex}}((\prod_{i=1}^{3} S_i(T_{ex}))^{\frac{1}{3}} - K)^{+}\right]$$

Inputs: $S_1(0) = 22$, $S_2(0) = 20$, $S_3(0) = 25$, $K = 20$, $T = 1$, $r = 0.1$, $\delta = 0.2$, $\sigma_1 = 0.2$, $\sigma_2 = 0.25$, $\sigma_3 = 0.15$, $\rho = (1\ 0.5\ -0.2;\ 0.5\ 1\ -0.4;\ -0.2\ -0.4\ 1)$, $\Sigma = (0.04\ 0.025\ -0.006;\ 0.025\ 0.0625\ -0.015;\ -0.006\ -0.015\ 0.025)$, $\mathcal{T} = \mathcal{T}\{\frac{T}{5}\times 1, \frac{T}{5}\times 2, \ldots, \frac{T}{5}\times 5\}$
Output: 1.77 Reference: 0.81
See selected option prices in Table 7.17 and convergence behaviour in Figure 7.12.

number of steps	100	101	200	201	500	501
option price	1.7630	1.7667	1.7653	1.7673	1.7651	1.7659
number of steps	1000	1001	5000	5001	10000	10001
option price	1.7655	1.7659	1.7659	1.7660	1.7659	1.7660
number of steps	20000	20001	40000	40001	100000	100001
option price	1.7660	1.7660	1.7659	1.7660	1.7660	1.7660

Table 7.17: Selected option prices for 3-D American geometric-average basket option

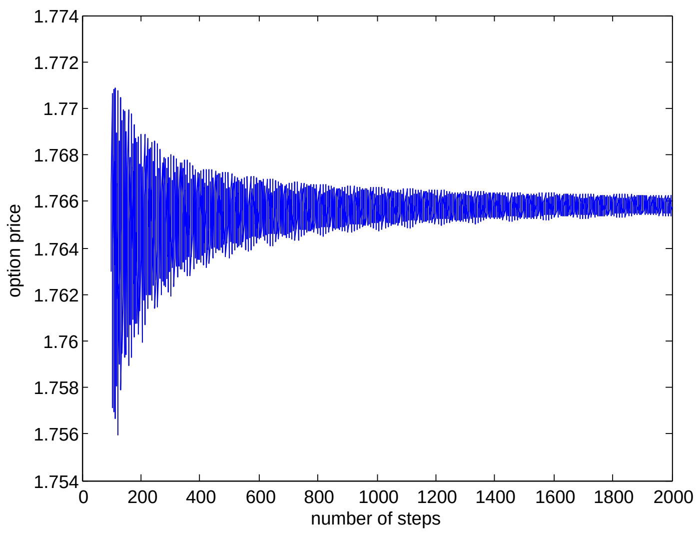


Figure 7.12: CRR tree for 3-D American geometric-average basket option

Test Case 18: 3-D American geometric-average basket option with discontinue payoff
Optimal expected discounted payoff is:

$$\sup_{T_{ex}\in\mathcal{T}} \mathbb{E}^{\mathbb{Q}}\left[e^{-rT_{ex}}(G-K)^{+}\mathbb{1}_{\{G\le B_1 \text{ or } G\ge B_2\}}\right] \text{ with } G=(\prod_{i=1}^{3} S_i(T_{ex}))^{\frac{1}{3}}$$

Inputs: $S_1(0)=22$, $S_2(0)=20$, $S_3(0)=25$, $K=20$, $T=1$, $r=0.1$, $\delta=0.2$, $\sigma_1=0.2$, $\sigma_2=0.25$, $\sigma_3=0.15$, $\rho=(1\ 0.5\ \ -0.2;\ 0.5\ 1\ \ -0.4;\ -0.2\ \ -0.4\ 1)$, $\Sigma=(0.04\ 0.025\ \ -0.006;\ 0.025\ 0.0625\ \ -0.015;\ -0.006\ \ -0.015\ 0.025)$, $\mathcal{T}=\mathcal{T}\{\frac{T}{5}\times 1, \frac{T}{5}\times 2, \ldots, \frac{T}{5}\times 5\}$, $B_1=22$, $B_2=30$
Output: 0.97 Reference: 0.22
See selected option prices in Table 7.18 and convergence behaviour in Figure 7.13.

number of steps	100	101	200	201	500	501
option price	0.9495	0.9987	0.9998	0.9732	1.0461	1.0285
number of steps	1000	1001	5000	5001	10000	10001
option price	0.9616	0.9757	0.9547	0.9612	0.9702	0.9748
number of steps	20000	20001	40000	40001	100000	100001
option price	0.9670	0.9702	0.9789	0.9768	0.9674	0.9688

Table 7.18: Selected option prices for 3-D American geometric-average basket option

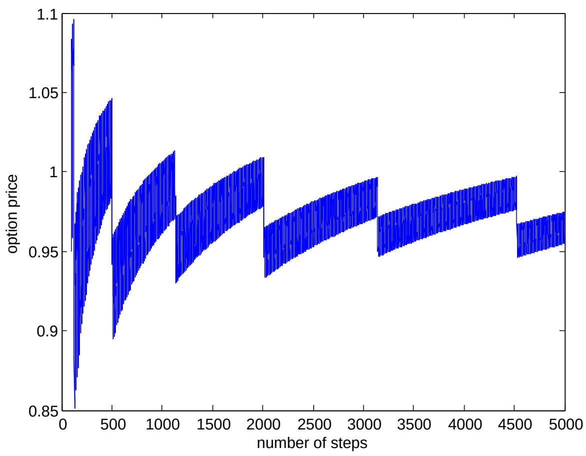


Figure 7.13: CRR tree for 3-D American geometric-average basket option with discontinue payoff

Test Case 19: 3-D American geometric-average basket option with strangle-spread-payoff
Optimal expected discounted payoff is:

$$\sup_{T_{ex}\in\mathcal{T}} \mathbb{E}^{\mathbb{Q}}\Big[e^{-rT_{ex}}\big((K_2-K_1)^+\mathbb{1}_{\{G<K_1\}}+(K_2-G)^+\mathbb{1}_{\{K_1\le G\le K_2\}}+0\cdot\mathbb{1}_{\{K_2<G<K_3\}}$$
$$+(G-K_3)^+\mathbb{1}_{\{K_3\le G\le K_4\}}+(K_4-K_3)^+\mathbb{1}_{\{G>K_4\}}\big)\Big] \text{ with } G=(\prod_{i=1}^{3}S_i(T_{ex}))^{\frac{1}{3}}$$

Inputs: $S_1(0) = 100$, $S_2(0) = 100$, $S_3(0) = 100$, $T = 1$, $r = 0.05$, $\delta = 0$, $\Sigma = (0.1150\ 0.0761\ 0.0353;\ 0.0761\ 0.0736\ 0.0281;\ 0.0353\ 0.0281\ 0.0141)$, $\mathcal{T} = \mathcal{T}\{\frac{T}{48}\times 1, \frac{T}{48}\times 2, \ldots, \frac{T}{48}\times 48\}$, $K_1 = 85$, $K_2 = 95$, $K_3 = 105$, $K_4 = 115$
Output: 8.934 Reference: 6.34
See selected option prices in Table 7.19 and convergence behaviour in Figure 7.14.

number of steps	48	96	240	480	720	960
option price	9.0445	9.0275	8.9454	8.9227	8.9420	8.9239
number of steps	4800	4848	7200	7248	9600	9648
option price	8.9404	8.9368	8.9310	8.9343	8.9315	8.9347
number of steps	48000	48048	72000	72048	96000	96048
option price	8.9346	8.9342	8.9346	8.9342	8.9341	8.9342

Table 7.19: Selected option prices for 3-D American geometric-average basket option with strangle-spread-payoff

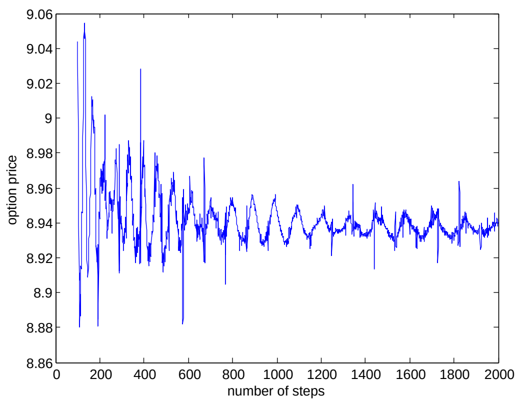


Figure 7.14: CRR tree for 3-D American geometric-average basket option with strangle-spread-payoff

7.1.4 7-D Examples in the Black-Scholes Model

Test Case 20: 7-D American geometric-average basket option with zero correlation

Optimal expected discounted payoff is:

$$\sup_{T_{ex} \in \mathcal{T}} \mathbb{E}^{\mathbb{Q}} \left[e^{-rT_{ex}} ((\prod_{i=1}^{7} S_i(T_{ex}))^{\frac{1}{7}} - K)^{+} \right]$$

Inputs: $S_1(0) = \cdots = S_7(0) = 100$, $K = 100$, $T = 1$, $r = 0.03$, $\delta = 0.05$, $\sigma_1 = \cdots = \sigma_7 = 0.4$, $\mathcal{T} = \mathcal{T}\{\frac{T}{10} \times 1, \frac{T}{10} \times 2, \ldots, \frac{T}{10} \times 10\}$, $\rho = (\rho_{ij})^{\top}$, $\Sigma = (\Sigma_{ij})^{\top}$, $\rho_{ii} = 1$, $\rho_{ij} = 0$, $\Sigma_{ii} = 0.16$, $\Sigma_{ij} = 0$ with $i, j = 1, \ldots, 7, i \neq j$.

Output: 3.27 Reference: 2.42

See selected option prices in Table 7.20 and convergence behaviour in Figure 7.15.

number of steps	100	101	200	201	500	501
option price	3.2668	3.2826	3.2692	3.2744	3.2696	3.2711
number of steps	1000	1001	5000	5001	10000	10001
option price	3.2702	3.2709	3.2699	3.2701	3.2700	3.2701
number of steps	20000	20001	40000	40001	100000	100001
option price	3.2700	3.2700	3.2700	3.2700	3.2700	3.2700

Table 7.20: Selected option prices for 7-D American geometric-average basket option with zero correlation

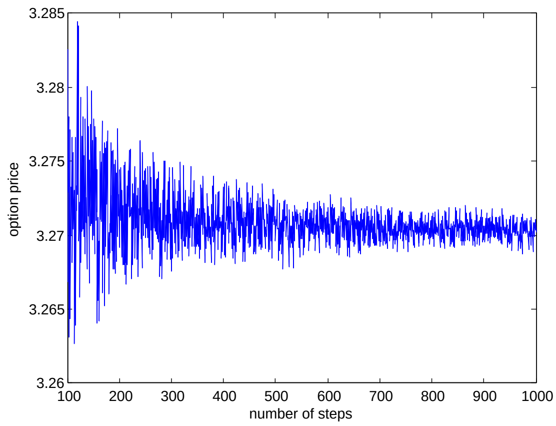


Figure 7.15: CRR tree for 7-D American geometric-average basket option with zero correlation

Test Case 21: 7-D American geometric-average basket option with non-zero correlation

Optimal expected discounted payoff is:

$$\sup_{T_{ex}\in\mathcal{T}} \mathbb{E}^{\mathbb{Q}}\left[e^{-rT_{ex}}((\prod_{i=1}^{7} S_i(T_{ex}))^{\frac{1}{7}} - K)^+\right]$$

Inputs: $S_1(0) = \cdots = S_7(0) = 100$, $K = 100$, $T = 1$, $r = 0.03$, $\delta = 0.05$, $\sigma_1 = \cdots = \sigma_7 = 0.4$, $\mathcal{T} = \mathcal{T}\{\frac{T}{10}\times 1, \frac{T}{10}\times 2, \ldots, \frac{T}{10}\times 10\}$, $\rho = (\rho_{ij})^\top$, $\Sigma = (\Sigma_{ij})^\top$, $\rho_{ii} = 1$, $\rho_{ij} = 0.1$, $\Sigma_{ii} = 0.16$, $\Sigma_{ij} = 0.016$ with $i, j = 1, \ldots, 7, i \neq j$.

Output: 4.77 Reference: 3.93

See selected option prices in Table 7.21 and convergence behaviour in Figure 7.16.

number of steps	100	101	200	201	500	501
option price	4.7618	4.7760	4.7631	4.7712	4.7662	4.7704
number of steps	1000	1001	5000	5001	10000	10001
option price	4.7668	4.7686	4.7670	4.7675	4.7671	4.7673
number of steps	20000	20001	40000	40001	100000	100001
option price	4.7672	4.7673	4.7672	4.7672	4.7672	4.7672

Table 7.21: Selected option prices for 7-D American geometric-average basket option with non-zero correlation

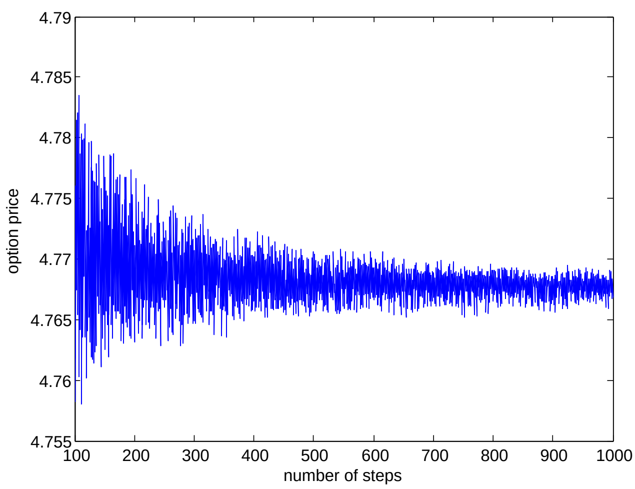


Figure 7.16: CRR tree for 7-D American geometric-average basket option with non-zero correlation

Test Case 22: 7-D American geometric-average basket option with discontinue payoff
Optimal expected discounted payoff is:

$$\sup_{T_{ex}\in\mathcal{T}} \mathbb{E}^{\mathbb{Q}}\left[e^{-rT_{ex}}(G-K)^{+}\mathbb{1}_{\{G\le B_1 \text{ or } G\ge B_2\}}\right] \text{ with } G=(\prod_{i=1}^{7} S_i(T_{ex}))^{\frac{1}{7}}$$

Inputs: $S_1(0)=\cdots=S_7(0)=100$, $K=100$, $B_1=110$, $B_2=120$, $T=1$, $r=0.03$, $\delta=0.05$, $\sigma_1=\cdots=\sigma_7=0.4$, $\mathcal{T}=\mathcal{T}\{\frac{T}{10}\times 1, \frac{T}{10}\times 2, \ldots, \frac{T}{10}\times 10\}$, $\rho=(\rho_{ij})^{\top}$, $\Sigma=(\Sigma_{ij})^{\top}$, $\rho_{ii}=1$, $\rho_{ij}=0.1$, $\Sigma_{ii}=0.16$, $\Sigma_{ij}=0.016$ with $i,j=1,\ldots,7, i\neq j$.
Output: 4.32 Reference: 2.76
See selected option prices in Table 7.22 and convergence behaviour in Figure 7.17.

number of steps	100	101	200	201	500	501
option price	4.3615	4.4479	4.3418	4.3945	4.3176	4.3553
number of steps	1000	1001	5000	5001	10000	10001
option price	4.2277	4.3109	4.3184	4.3292	4.3046	4.3123
number of steps	20000	20001	40000	40001	100000	100001
option price	4.3260	4.3208	4.3099	4.3221	4.3170	4.3191

Table 7.22: Selected option prices for 7-D American geometric-average basket option with discontinue payoff

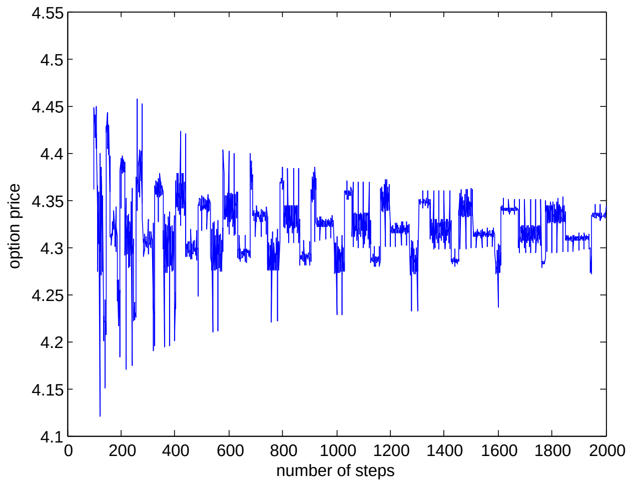


Figure 7.17: CRR tree for 7-D American geometric-average basket option with discontinue payoff

Test Case 23: 7-D American geometric-average basket option with strangle-spread-payoff
Optimal expected discounted payoff is:

$$\sup_{T_{ex}\in\mathcal{T}} \mathbb{E}^{\mathbb{Q}}\Big[e^{-rT_{ex}}\big((K_2-K_1)^+\mathbb{1}_{\{G<K_1\}}+(K_2-G)^+\mathbb{1}_{\{K_1\le G\le K_2\}}+0\cdot\mathbb{1}_{\{K_2<G<K_3\}}$$
$$+(G-K_3)^+\mathbb{1}_{\{K_3\le G\le K_4\}}+(K_4-K_3)^+\mathbb{1}_{\{G>K_4\}}\big)\Big] \text{ with } G=(\prod_{i=1}^{7}S_i(T_{ex}))^{\frac{1}{7}}$$

Inputs: $S_1(0)=\cdots=S_7(0)=100$, $K_1=90$, $K_2=100$, $K_3=110$, $K_4=120$, $T=1$, $r=0.03$, $\delta=0.05$, $\sigma_1=\cdots=\sigma_7=0.4$, $\mathcal{T}=\mathcal{T}\{\frac{T}{10}\times 1, \frac{T}{10}\times 2,\ldots,\frac{T}{10}\times 10\}$, $\rho=(\rho_{ij})^\top$, $\Sigma=(\Sigma_{ij})^\top$, $\rho_{ii}=1$, $\rho_{ij}=0.1$, $\Sigma_{ii}=0.16$, $\Sigma_{ij}=0.016$ with $i,j=1,\ldots,7, i\neq j$.
Output: 8.42 Reference: 6.85
See selected option prices in Table 7.23 and convergence behaviour in Figure 7.18.

number of steps	100	101	200	201	500	501
option price	8.4396	8.4677	8.4632	8.4540	8.4309	8.4274
number of steps	1000	1001	5000	5001	10000	10001
option price	8.4141	8.4133	8.4158	8.4175	8.4165	8.4177
number of steps	20000	20001	40000	40001	100000	100001
option price	8.4177	8.4177	8.4174	8.4174	8.4174	8.4174

Table 7.23: Selected option prices for 7-D American geometric-average basket option with strangle-spread-payoff

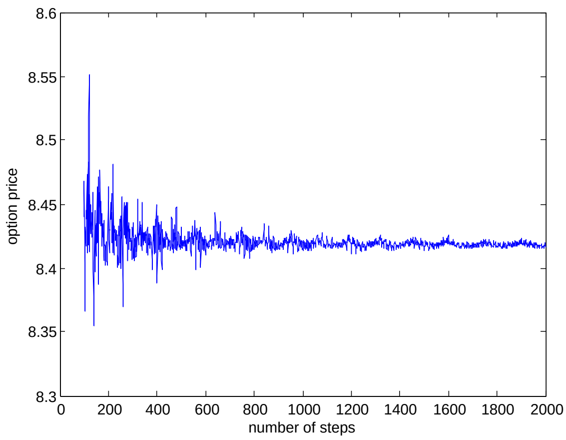


Figure 7.18: CRR tree for 7-D American geometric-average basket option with strangle-spread-payoff

7.1.5 1-D Examples in the Heston Model

Test Case 24: 1-D American option in Heston Model
Optimal expected discounted payoff is:

$$\sup_{T_{ex}\in\mathcal{T}[0,T]} \mathbb{E}^{\mathbb{Q}}\left[e^{-rT_{ex}}\left(K - S(T_{ex})\right)^{+}\right]$$

Inputs: $K = 100$, $T = \frac{1}{2}$, $r = 0.05$, $\delta = 0$, $\kappa = 3$, $\theta = 0.04$, $\sigma = 0.1$, $\rho = -0.1$, $V_0 = 0.04$, $\hat{V} = 0.02$
In this example, we discuss the option prices for three cases: at the money, in the money and out of the money.
See selected option prices using RSS tree in Table 7.24.

At the Money: $S = 100$						
number of steps	5	10	20	30	40	50
option price	4.6660	4.6794	4.6698	4.6607	4.6528	4.6470
number of steps	60	70	80	90	100	200
option price	4.6424	4.6432	4.6431	4.6486	4.6502	4.6532
Output: 4.65 Reference: 4.41						
In the Money: $S = 90$						
number of steps	5	10	20	30	40	50
option price	10.5509	10.6547	10.6334	10.6452	10.6505	10.6443
number of steps	60	70	80	90	100	200
option price	10.6451	10.6499	10.6498	10.6477	10.6458	10.6461
Output: 10.65 Reference: 9.86						
Out of the Money: $S = 110$						
number of steps	5	10	20	30	40	50
option price	1.7093	1.6644	1.6965	1.68 81	1.6858	1.6854
number of steps	60	70	80	90	100	200
option price	1.6887	1.6782	1.6871	1.6866	1.6796	1.6829
Output: 1.68 Reference: 1.62						

Table 7.24: Selected option prices for 1-D American option in Heston Model

Bibliography

[1] **Abramowitz, W.**, **Stegun, I.** (1972). *Handbook of mathematical functions with formulas, graphs and mathematical tables*. Dover Publications.

[2] **Andersen, L.**, **Broadie, M.** (2004). *A primal-dual simulation algorithm for pricing multidimensional American options*. Management Sciences, 50:1222-1234.

[3] **Bally, V.** and **Pàges, G.** (2003). *Error analysis of a quantization algorithm for obstacle problems*. Stochastic Processes and their Applications, 106:1-40.

[4] **Belomestny, D.**, **Bender, C.** and **Schoenmakers, J.** (2009). *True upper bounds for Bermudan products via non-nested Monte Carlo*. Mathematical Finance, 19(1):53-71.

[5] **Boyle, P. P.**, **Evnine, J.** and **Gibbs, S.** (1989). *Numerical evaluation of multivariate contingent claims*. The Review of Financial Studies, Volume 2(2), 241-250.

[6] **Bishop, C. M.** (1979). *Pattern recognition and machine learning*. Springer.

[7] **Broadie, M.** and **Kaya, O.** (2006). *Exact simulation of stochastic volatility and other affine jump diffusion processes*. Operations Research, 54:2, 217-231.

[8] **Chang, C.** and **Lin, C.** (2011). *LIBSVM: A library for support vector machines*. Journal of ACM Transactions on Intelligent Systems and Technology, Volume 2(3), 27:1-27:27.

[9] **Chang, C.**, **Hsu, C.** and **Lin, C.** (2010). *A practical guide to support vector classification*.

[10] **Cherkassky, V.** and **Ma, Y.** (2004). *Practical selection of SVM parameters and noise estimation for SVM regression*. Neural Networks, Volume 17(1), 113-126.

[11] **Cox, J. C.**, **Ross, S. A.** and **Rubinstein, M.** (1979). *Option pricing: A simplified approach*. Journal of Financial Economics 7, 229-263.

[12] **Clément, E.**, **Lamberton, D.** and **Protter, P.** (2002). *An analysis of a least squares regression method for American option pricing*. Finance and Stochastics 6, 449-471.

[13] **Egloff, D.** (2005). *Monte Carlo algorithms for optimal stopping and statistical learning*. Annals of Applied Probability 15, 1-37.

[14] **Egloff, D.**, **Kohler, M.** and **Todorovic, N.** (2007). *A dynamic look-ahead Monte Carlo algorithm for pricing American options*. Annals of Applied Probability 17, 1138-1171.

[15] **Glasserman, P.** (2003). *Monte Carlo methods in financial engineering*. Springer.

[16] **Glasserman, P.** and **Yu** (2004). *Number of paths versus basis functions in American option pricing*. The Annals of Applied Probability, 14(4):2090-2119.

[17] **Györfi, L.**, **Kohler, M**, **Krzyzak, A** and **Walk, H** (2002). *A distribution-free theory of nonparametric regression*. Springer Series in Statistics, Springer.

[18] **Haugh, M.** and **Kogan, L.** (2004). *A duality approach*. Operations Research, 52:258-270.

[19] **Hoek, J.** and **Elliott, R. J.** (2006). *Binomial models in finance*. Springer.

[20] **Hull, J. C.** (2008). *Options, futures, and other derivatives*. 7 edition, Prentice Hall.

[21] **Jarrow, R. A.** and **Rudd, A.** (1983). *Option pricing*. Irwin Professional Publishing.

[22] **Kohler, M.**. *A review on regression-based Monte Carlo methods for pricing American options* (2010). Recent Developments in Applied Probability and Statistics, 37-58.

[23] **Kohler, M.**, **Krzyzak, A.** and **Todorovic, N.** (2010) *Pricing of high-dimensional American options by neural networks*. Mathematical Finance, Volume 20, No. 3, 381-410.

[24] **Kohler, M.** (2008) *A regression based smoothing spline Monte Carlo algorithm for pricing American options*. AStA Advances in Statistical Analysis 92, 153-178.

[25] **Korn, R.** and **Korn, E.** (2001). *Option pricing and portfolio optimization: modern methods of financial mathematics*. 1 edition, American Mathematical Society.

[26] **Korn, R.**, **Korn, E.** and **Kroisandt, G.** (2010). *Monte Carlo methods and models in finance and insurance*. CRC Press.

[27] **Korn, R.** and **Müller, S.** (2009). *Getting multi-dimensional trees into a new shape*. Wilmott Journal, Volume 1(3), 145-153.

[28] **Korn, R.** and **Müller, S.** (2009). *The decoupling approach to binomial pricing of multi-asset options*. The Journal of Computational Finance, Volume 12(3), 1-30.

[29] **Lee, H.-J.**, **Yang, S.-H.**, **Han, G.-S.** and **Lee, J.** (2008). *Simulations for American option pricing under a jump-diffusion model: comparison study between kernel-based and regression-based methods*. Advances in Neural Networks - ISNN 2008. Volume 5263, 655-662.

[30] **Lee, J.** (2008). *Recent advances in American option pricing using simulations.*

[31] **Liang, Q.** (2012). *Innovative Techniken und Algorithmen im Bereich Computational-Finance und Risikomanagement*. Dissertation at the University of Kaiserslautern.

[32] **London, J.** (2004). *Modeling derivatives in C++*. 1 edition, Wiley.

[33] **Longstaff, F. A.** and **Schwartz, E. S.** (2001). *Valuing American options by simulation: A simple least-squares approach*. The Review of Financial Studies, Volume 14(1), 113-147.

[34] **Matsumoto, M.** and **Nishimura, T.** (1998). *Mersenne Twister: a 623-dimensionally equidistributed uniform pseudo-random number generator*. ACM transactions on modeling and computer simulations, Volume 8(1), 3-30.

[35] **Moreno, M.** and **Navas, J. F.** (2003). *On the robustness of least-squares Monte Carlo (LSM) for pricing American derivatives*. Review of Derivatives Research 6, 107-128.

[36] **Müller, S.** (2009). *The binomial approach to option valuation: getting binomial trees into shape*. Dissertation at the University of Kaiserslautern.

[37] **Quecke, S.** (2007). *Efficient numerical methods for pricing American options under Lévy models*. Dissertation at the University of Cologne.

[38] **Rogers, L.C.G.** (2001). *Monte Carlo valuation of American options*. Mathematical Finance, 12:271-286.

[39] **Sayer, T.** (2011). *Pricing American options in the Heston model: a close look on incorporating correlation*. Report of Fraunhofer ITWM, Nr. 204

[40] **Sayer, T.** (2012). *Valuation of American-style derivatives within the stochastic volatility model of Heston*. Dissertation at the University of Kaiserslautern.

[41] **Schölkopf, A. S.**, **Williamson, R. C.** and **Bartlett, P. L.** (2000). *New support vector algorithms*. Neural Computation, 12:1207-1245.

[42] **Todorovic, N.** (2007). *Bewertung Amerikanischer Optionen mit Hilfe von regressionbasierten Monte-Carlo-Verfahren.* PhD Dissertation, Universität Saarland.

[43] **Tsitsiklis, J. N.** and **Van Roy, B.** (2001). *Regression methods for pricing complex American style options.* IEEE Transactions on Neural Networks, Vol.12, No.4, 694-701.

[44] **Varela, J. A.**, **Brugger, C.**, **Wehn, N.**, **Korn, R.** and **Tang, S.** (2015). *Reverse Longstaff-Schwartz American Option Pricing on hybrid CPU/FPGA Systems.* IEEE Date 2015 conference.

[45] **Wendel, S.** (2009) *Monte Carlo methods for pricing American options.* Diploma thesis at the University of Kaiserslautern.

[46] **Zhang, P. G.** (1998). *Exotic options - a guide to second generation options.* 2 edition, World Scientific.

Selbständigkeitserklärung

Ich erkläre hiermit, dass ich die Dissertation mit dem Thema

"American-style Option Pricing and Improvement of Regression-based Monte Carlo Methods by Machine Learning Techniques"

ohne fremde Hilfe angefertigt habe und nur die im Literaturverzeichnis angeführten Quellen und Hilfsmittel benutzt habe.

Wissenschaftlicher Werdegang

Wissenschaftlicher und beruflicher Werdegang

09.2004 – 07.2006	Studium der Werkstoffwissenschaft an der Tongji Universität in Shanghai
10.2006 – 12.2011	Studium der Mathematik an der Technischen Universität Kaiserslautern, Schwerpunkt Finanzmathematik Abschluss: Diplom (Dipl.-Math.)
02.2011 – 10.2011	Diplomand/Praktikant bei der UniCredit Bank in München Abteilung: Structured Fixed Income Trading
01.2012 – 03.2015	Promotionsstudium der Mathematik an der Technischen Universität Kaiserslautern, Schwerpunkt Finanzmathematik Abschluss: Doktor (Dr. rer. nat.)
seit 05.2015	Financial Engineer bei dem BearingPoint/RiValue in Frankfurt am Main

Wissenschaftliche Veröffentlichungen

12.2011	*SABR Model in Interest-Rate World and Correction of Labordere's Option Pricing Formula in the Normal SABR Case*, Diplomarbeit
09.2013	*Exact Analytical Solution for the Normal SABR Model*, Wilmott
03.2015	*Reverse Longstaff-Schwartz American Option Pricing on hybrid CPU/FPGA Systems*, IEEE Date-Conference 2015
08.2015	*American-style Option Pricing and Improvement of Regression-based Monte Carlo Methods by Machine Learning Techniques*, Doktorarbeit
09.2015	*FPGA Based Accelerators for Financial Applications*, Springer
2015	*Brownian Bridge based Longstaff-Schwartz Method for Pricing American Options on FPGA*, vorbereitet

Scientific Career

Scientific and professional career

09.2004 – 07.2006	Study of Material Science at the Tongji University in Shanghai
10.2006 – 12.2011	Study of Mathematics at the University of Kaiserslautern, specialization: Financial Mathematics Degree: Diploma (Dipl.-Math.)
02.2011 – 10.2011	Diplomate/intern at the UniCredit Bank in Munich Department: Structured Fixed Income Trading
01.2012 – 03.2015	Doctoral study of Mathematics at the University of Kaiserslautern, specialization: Financial Mathematics Degree: Doctor (Dr. rer. nat.)
since 05.2015	Financial Engineer at the BearingPoint/RiValue in Frankfurt am Main

Scientific Publications

12.2011	*SABR Model in Interest-Rate World and Correction of Labordere's Option Pricing Formula in the Normal SABR Case*, Diploma thesis
09.2013	*Exact Analytical Solution for the Normal SABR Model*, Wilmott
03.2015	*Reverse Longstaff-Schwartz American Option Pricing on hybrid CPU/FPGA Systems*, IEEE Date-Conference 2015
08.2015	*American-style Option Pricing and Improvement of Regression-based Monte Carlo Methods by Machine Learning Techniques*, PhD thesis
09.2015	*FPGA Based Accelerators for Financial Applications*, Springer
2015	*Brownian Bridge based Longstaff-Schwartz Method for Pricing American Options on FPGA*, prepared