Monte Carlo Pricing of Bermudan-style Derivatives with Lower and Upper Bound Methods

Submitted in partial fulfillment of the requirement for the degree of

Master of Science
in
Applied Mathematics

Author:	*Supervisors:*
Lai Jiang	Prof. Dr. Arun Bagchi
	Dr. Tim Dijkstra
	Dr. Lech Grzelak

Faculty of Electrical Engineering, Mathematics and
Computer Science

University of Twente
Enschede - The Netherlands

October 29, 2012

Abstract

The Longstaff-Schwartz algorithm is widely used for pricing Bermudan options. It allows Monte Carlo simulation to take into account the early-exercise feature of a Bermudan option. The method utilizes multi-linear regression to estimate the continuation value of such options. In this thesis, we study the impact of different regressor configurations on the performance of the Longstaff-Schwartz method. We evaluate pricing result in various model settings including the Black-Scholes world, the Heston model and the lognormal Libor market model. By using an upper bound pricing algorithm proposed by Andersen and Broadie, we demonstrate a reliable measure for evaluating the performance of the Longstaff-Schwartz algorithm. We show that regressor configuration plays a significant role in this method, and give recommendations on how to construct effective regressors.

Acknowledgements

I would like to express my sincere gratitude to Prof. Arunabha Bagchi for his encouragement and valuable feedbacks during the project and for introducing me to this fascinating field. I am very grateful to my daily supervisors at Rabo, Dr. Tim Dijkstra and Dr. Lech Grzelak, for the fruitful discussions and great suggestions. I would like to thank Dr. Fang Fang for generously offering me her excellent pricing tool by the COS method. I would like to thank Dr. Natalia Borovykh for sharing her keen insight in the subject. I would like to thank Dr. Viktor Tchistiakov for recommending me to the group. I would like to thank Erik van Raaij and Sacha van Weeren for hiring me on the research project leading to this thesis. I would like to thank all members from Derivative Research & Validation for making the experience so much more enjoyable. To my family and friends, I can not thank you more for all your love and support along the way.

Contents

1 Introduction

How much would you pay for the right to buy or sell certain asset at some specified price in a predetermined finite set of dates? Academics and practitioners alike face this problem when they want to price a Bermudan option. The difficulty arises because in general we do not know a priori when to exercise the option to maximize the payoff. The associated mathematical problem is a discrete optimal stopping problem, with known solution method by the dynamic programming principle. There is no known exact analytical solution to the above problem even with very simple option payoff type and underlying asset dynamics model. Instead, one can solve it by numerical methods. Depending on the number of state variables involved, different numerical method gains preference. For one to at most three state variables, Tree methods or lattice methods such as Finite Differences produce fast and accurate result. But as the state variables grows in number, "curse of dimensionality" [28] means that these methods are no longer practical in terms of efficiency and implementation complexity. Monte Carlo simulation is by far the most favourable method for solving higher dimensional problems.

Over the years, interest rate derivatives market has expanded tremendously. Apart from the bonds, interest rate caps, floors and swaptions are all traded in huge volume. OTC market for exotic rate products appearing as structured notes was also growing significantly in the hey days. The liquidity of caps and swaptions makes them popular hedging instruments for more complicated interest rate derivatives. Thus one of the most desired features from the market for a successful interest rate model is to efficiently recover the cap and swap market prices, which is called model calibration. The multi-factor BGM model [12], also known as the Libor market model, and its various extensions have gained popularity over the short rate models mainly because of their built-in capability for calibration.

As is mentioned, Monte Carlo simulation is the de facto tool for pricing complex derivatives in these multi-factor models. To price a Bermudan option, one needs to compare the payoff due to exercise and the remaining value of the option if it is not exercised. A problem arises in estimating the option's remaining value based on the information up to the exercise moment. This is a functionality not available in the original Monte Carlo framework. Researchers have proposed a number of regression based methods including the Longstaff-Schwartz algorithm to overcome this difficulty. It is now within the capability of Monte Carlo to evaluate Bermudan options.

Successful implementation of the Longstaff-Schwartz method hinges on the construction of regressors. Hence we will explore using a reliable criterion to evaluate different regressor configurations and improve the performance by identifying more effective ones. We will also give recommendation on how to construct effective regressors with respect to different models and option types. In Chapter 2, we will introduce the Longstaff-Schwartz Monte Carlo, discuss its implementation issues and highlight important aspects that affect its performance. In Chapter 3, we will discuss both the theory and implementation issues of a method for calculating the upper bound for Bermudan option price. Thus we can bound the true option value from above and below (by the Longstaff-Schwartz method). The gap between the two bounds can be used as a performance evaluation measure. In Chapter 4, we will introduce the Black-Scholes world and the Heston model as well as several types of option payoff. These serve as test cases for studying the performance of the Longstaff-Schwartz method. In Chapter 5, we will apply two special regression methods to the Longstaff-Schwartz method. Their effectiveness will be evaluated. In Chapter 6, we will discuss the main model of interest for us, the Libor market model, and present how to build a Monte Carlo engine for it. We will introduce relevant interest rate market products including the Bermudan-style Libor Exotics. We will show how to adapt the Longstaff-Schwartz algorithm to evaluate these complex interest rate derivatives. In Chapter 7, we will present and analyse all the simulation results, evaluate various regressor configurations by the criterion based on the upper bound algorithm, and give recommendation on how to construct effective regressors. In Chapter 8, we will make final conclusions and suggest topics for further research. In the Appendices, we will introduce the basics about pricing derivatives with Tree methods and Finite Difference methods. We also include all the upper bound simulation data for reference.

2 Monte Carlo and the Longstaff-Schwartz Algorithm

In this section we introduce the basics of the Monte Carlo method through an example in the Black-Scholes world. We will review the mathematical problem of pricing a Bermudan option and study the Longstaff-Schwartz algorithm for solving this problem in the Monte Carlo framework. We will also discuss details of this method concerning regressor choice, pricing error and convergence property.

2.1 Pricing Plain Vanilla Option with Monte Carlo

Monte Carlo simulation is widely used in pricing derivative securities. We begin this section by reviewing the application of Monte Carlo in pricing vanilla options. The following simple example helps to clarify the basic ideas underlying Monte Carlo method. Suppose we want to evaluate the expectation $\mathbb{E}[f(X(T))]$, where X denotes a stochastic process $\{X(t), 0 \leq t \leq T\}$ taking value in $\mathbb{R}$. f is an integrable function, $f : \mathbb{R} \rightarrow \mathbb{R}$. Let $X^{(k)}$, for $k = 1, \ldots, N$, be realized paths of process X. By the Strong Law of Large Numbers, the following limit holds almost surely,

$$\lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^{N} f(X^{(k)}(T)) = \mathbb{E}[f(X(T))].$$

This implies that we can obtain an approximation to the expectation with finite number of realized paths.

$$\frac{1}{N} \sum_{k=1}^{N} f(X^{(k)}(T)) \approx \mathbb{E}[f(X(T))].$$

Suppose the dynamics of process X is specified by the following Stochastic Differential Equation (SDE),

$$dX(t) = a(X(t))dt + b(X(t))dW(t), \qquad \text{for } 0 \leq t \leq T.$$

where $a(x)$ and $b(x)$ are deterministic coefficient functions satisfying necessary regularity conditions.

To simulate the above continuous time SDE, we need to discretize it on a time grid. Suppose we choose an equidistant grid with grid size h. Let $X^{(h,k)}$ denote the discretized process along path k with grid size h. $X^{(h,k)}$ converges to $X^{(k)}$ as $h \downarrow 0$, so we have the following approximation

$$\frac{1}{N} \sum_{k=1}^{N} f(X^{(h,k)}(T)) \approx \mathbb{E}[f(X(T))]. \tag{2.1}$$

Approximation (2.1) differs from the true expectation value with error introduced by both finite number of paths N and nonzero time step h. Error introduced by the latter source is commonly called the discretization error.

As an example, we will value a plain vanilla option on an asset following the Geometric Brownian Motion (GBM). The option price at maturity T, $V(T)$, is given by its payoff function. Under the Equivalent Martingale Measure (EMM) $\mathbb{Q}$ where W is a $\mathbb{Q}$ Brownian Motion, the risk-neutral dynamics of the asset price is given by

$$dS(t) = rS(t)dt + \sigma S(t)dW(t), \qquad 0 \leq t \leq T. \tag{2.2}$$

where r is the constant risk-free rate and σ is the constant volatility parameter.

Applying Itô's lemma to $\ln(S(t))$ yields,

$$\begin{aligned} d\ln(S(t)) &= \frac{1}{S(t)} dS(t) + \frac{1}{2}(-\frac{1}{S^2(t)})dS(t)dS(t) \\ &= (r - \frac{1}{2}\sigma^2)dt + \sigma dW(t). \end{aligned} \tag{2.3}$$

Integrating SDE (2.3) over the time interval $[t_i, t_{i+1}]$, where $0 \leq t_i < t_{i+1} \leq T$, gives,

$$\begin{aligned} \ln\frac{S(t_{i+1})}{S(t_i)} &= \ln S(t_{i+1}) - \ln S(t_i) \\ &= \int_{t_i}^{t_{i+1}} (r - \frac{1}{2}\sigma^2)dt + \int_{t_i}^{t_{i+1}} \sigma dW(t) \\ &= (r - \frac{1}{2}\sigma^2)(t_{i+1} - t_i) + \sigma(W(t_{i+1}) - W(t_i)). \end{aligned}$$

This gives us the recursive expression for $S(t_{i+1})$,

$$S(t_{i+1}) = S(t_i)\exp\left[(r - \frac{1}{2}\sigma^2)(t_{i+1} - t_i) + \sigma(W(t_{i+1}) - W(t_i))\right] \tag{2.4}$$

Assuming an equidistant grid, so $t_{i+1} - t_i = \Delta t$, for all i. Let $S_{i+1} = S(t_{i+1})$, $\Delta W_{i+1} = W(t_{i+1}) - W(t_i)$, for $i = 0, 1, \ldots, N_h - 1$. Notice that $\Delta W_{i+1} \sim \mathcal{N}(0, \Delta t)$, so we can replace ΔW_{i+1} by $\sqrt{\Delta t} Z_{i+1}$ where Z_{i+1} are independent standard normal random variables for all $i = 0, 1, \ldots, N_h - 1$. Given initial asset value $S_0 = S(0)$, Equation (2.4) gives the exact formula for simulating a GBM asset price path,

$$S_{i+1} = S_i \exp\left[(r - \frac{1}{2}\sigma^2)\Delta t + \sigma\sqrt{\Delta t} Z_{i+1}\right]. \tag{2.5}$$

for all $i = 0, 1, \ldots, N_h - 1$.

Alternatively, we can discretize SDE (2.2) directly to show the principle of Euler discretization scheme. The time axis is divided by grid $0 = t_0 < t_1 < \ldots < t_{N_h} = T$ with step $\Delta t = h$. Integrating SDE (2.2) over the interval $[t_i, t_{i+1}]$ and using approximations to the two integrals gives

$$\begin{aligned} S(t_{i+1}) &= S(t_i) + r\int_{t_i}^{t_{i+1}} S(x)dx + \sigma\int_{t_i}^{t_{i+1}} S(x)dW(x), \\ &\approx S(t_i) + rS(t_i)(t_{i+1} - t_i) + \sigma S(t_i)(W(t_{i+1}) - W(t_i)). \end{aligned}$$

In the last step, these two Euler approximations are assumed,

$$\begin{aligned} \int_{t_i}^{t_{i+1}} S(x)dx &\approx S(t_i)(t_{i+1} - t_i), \\ \int_{t_i}^{t_{i+1}} S(x)dW(x) &\approx S(t_i)(W(t_{i+1}) - W(t_i)). \end{aligned}$$

Using the same notations as in Equation (2.5), the Euler scheme approximates a GBM asset price path on the grid by

$$S_{i+1} = S_i + rS_i\Delta t + \sigma S_i\sqrt{\Delta t} Z_{i+1}. \tag{2.6}$$

for all $i = 0, 1, \ldots, N_h - 1$.

Equation (2.5), derived from the analytical solution, gives the exact value of realized asset price along each simulated path if the random variables Z_{i+1} are ideal. As Glasserman[28] points out, Equation (2.6) approximates the actual process to an accuracy of $O(\sqrt{\Delta t})$.

Now we can price vanilla options with Monte Carlo simulation. The Risk-Neutral Pricing Principle [51] states that,

$$V(t) = \mathbb{E}\left[e^{-r(T-t)}V(T) \mid \mathcal{F}(t)\right], \qquad 0 \le t \le T. \tag{2.7}$$

where $\mathbb{E}$ is taken under EMM $\mathbb{Q}$. Function $V(t)$ denotes the arbitrage-free value of the option at time t. At maturity T, it is given by the payoff function of the vanilla option.

We generate N sample paths of the asset price according to either Equation (2.5) or Equation (2.6). Given initial asset value $S(0)$, we have the following approximation to the option value (2.7) at $t = 0$,

$$\hat{V}(0) = e^{-rT}\frac{1}{N}\sum_{k=1}^{N} g(S^{h,k}(T)), \tag{2.8}$$

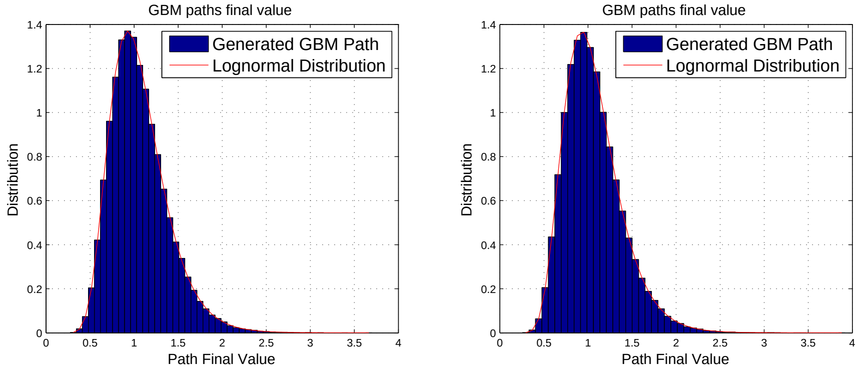
where function $g(s)$ is the time T payoff of the vanilla option with underlying asset value s.

We have compared the GBM paths generated according to Equation(2.5) and the paths generated according to Equation(2.6). The model parameters used are $r = 0.06, \sigma = 0.3, T = 1, S(0) = 1$. We discretize the one year time period into 52 equidistant segments ($\Delta t = T/52$). 10^5 sample paths are generated to make a histogram of the time T value of the GBM paths. We use Matlab throughout the whole thesis. The random number generator is based on Mersenne Twister. Figure 2.1 shows that the computer generated random samples from both equations follow approximately lognormal distribution. Equation (2.5) indicates that the ratio $S(T)/S(0)$ follows a lognormal distribution such that

$\ln[S(T)/S(0)] \sim \mathcal{N}(m, s^2)$, where $m = (r - \frac{1}{2}\sigma^2)T$ and $s = \sigma\sqrt{T}$. We have the following statistics for the lognormal random variable [52]

$$\begin{aligned}
\text{mean} &= e^{m+s^2/2}, \\
\text{variance} &= (e^{s^2} - 1)e^{2m+s^2}, \\
\text{skewness} &= e^{s^2} + 2)\sqrt{e^{s^2} + 1}, \\
\text{kurtosis} &= e^{4s^2} + 2e^{3s^2} + 3e^{2s^2} - 3.
\end{aligned} \tag{2.9}$$

The sample statistics are calculated by Matlab functions. Although these numbers are themselves random, being close to the model statistics brings reassurance to our simulation setup. Some typical values (up to the 4th digit) from simulation are shown in Table 2.1. Both methods give close approximations to the theoretical lognormal random variable with 10^5 sample paths.



(a) GBM by Euler Scheme on SDE

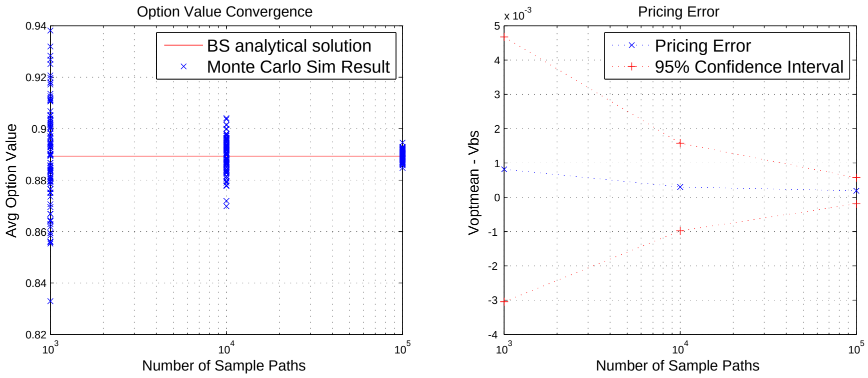
(b) GBM by Analytical Solution

Figure 2.1: Histogram of time T value of GBM paths generated by Euler scheme and by the analytical solution, $\Delta t = T/52$, path number $N_p = 10^5$.

Table 2.1: Sample statistics of computer generated GBM path ratio $S(T)/S(0)$, $\Delta t = T/52$, path number $N_p = 10^5$.

Statistics	Value				
	Theoretical	Euler run1	Solution run1	Euler run 2	Solution run2
Mean	1.0618	1.0635	1.0621	1.0625	1.0618
Variance	0.1062	0.1061	0.1070	0.1067	0.1060
Skewness	0.9495	0.9217	0.9570	0.9555	0.9533
Kurtosis	4.6449	4.5170	4.6620	4.8164	4.7059

We have compared the differences between simulated option value and the Black-Scholes analytical solution. Figure 2.2 shows convergence of simulated option value to the analytical solution when $S(0) = 10$, $K = 10$. We take 10^3 to 10^5 sample paths. For each number of sample paths, we run 100 Monte Carlo simulations. The average of 100 results is reported as the simulation result. The pricing error is then given by the difference between this result and the analytical solution from the Black-Scholes formula. We also report the 95% confidence interval of the pricing error based on t-statistics of the sample mean. The pricing error with GBM paths generated by analytical solution is due to the error in computer generated random samples as well as finite number of paths used in the Monte Carlo simulation. The pricing error with GBM paths generated by Euler Scheme on SDE is caused by both of the above sources as well as the error introduced by the approximations in Equation (2.6).



(a) Convergence of Monte Carlo Result

(b) Pricing Error

Figure 2.2: Convergence of Monte Carlo simulation result according to Equation (2.6) to the analytical solution of a European Put in the Black-Scholes world, $r = 0.06$, $\sigma = 0.3$, $T = 1$, $S(0) = 10$, $K = 10$, $\Delta t = T/52$.

2.2 Pricing American or Bermudan Option with Monte Carlo

2.2.1 Dynamic Programming Formulation of Bermudan Option Pricing

An American option grants its holder the right to exercise the option at any time prior to maturity. A Bermudan option allows its holder to exercise the option at a set of possible exercise moments up to maturity. Bermudan option value can be simulated by Monte Carlo. As we choose finer and finer grid on the time axis and have more and more early exercise moments, the American option value can be approximated by the Bermudan option value. Convergence from discrete to continuous time optimal stopping problem is treated in [22].

Let us consider the following tenor structure on which we will study the problem of pricing Bermudan options. The time interval $[0, T]$ is discretized by the grid $0 = T_0 < T_1 < \ldots < T_N = T$, where $\alpha_i = T_i - T_{i-1} = T/N$ for each $i = 0, 1, \ldots, N$. A Bermudan option with maturity T issued at time T_0 admits the following set of possible exercise moments $\mathcal{T} = \{T_1, \ldots, T_N\}$.

Suppose we have chosen the probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and the natural sigma-algebra filtration, $\{\mathcal{F}_t, t \geq 0\}$. For any $0 \leq a < b < \infty$, Let $\mathcal{S}_{a,b}$ denote the set of stopping times on the interval $[a, b]$, so $\mathcal{S}_{a,b} \subset [a, b] \cap \mathcal{T}$ [44]. For example, the set of all stopping times on the interval $[0, T]$ is $\mathcal{S}_{0,T}$.

In order to simplify the expressions, the quantities below are all in time 0 money, i.e. the quantities are discounted by a proper numeraire back to time 0, but the numeraire is not explicitly expressed. Suppose we have a value process $\{\bar{X}(t), t \geq 0\}$, where $\bar{X}(t)$ represents the value at time t in time t money. Let $\{\bar{N}(t), t \geq 0\}$ be the chosen strictly positive numeraire process. Then $\{X(t), t \geq 0\}$ is the discounted value process, where $X(t) = \bar{X}(t)/\bar{N}(t)$ represents the value at time t in time T_0 money denoted with numeraire $\bar{N}$. By choosing the numeraire $\bar{N}$, the expectations are taken under the corresponding EMM $\mathbb{Q}$. We define the following quantities in the same fashion.

Let Z_k and H_k denote the discounted exercise value and the discounted hold value of the option at time T_k in time 0 money respectively. V_k is the discounted option value, i.e. the arbitrage free price, at time T_k in time 0 money. We adopt the convention that there is no early exercise opportunity at time 0, hence $Z_0 = 0$. Holding the option at its maturity brings no payoff, so $H_N = 0$.

The following approach of Bermudan option pricing [44] takes the option holder's perspective. The value to the holder is the maximal expected payoff from exercising the option. It is stated that there exists an optimal stopping time τ^* such that

$$V_0 = \mathbb{E}[Z_{\tau^*}|\mathcal{F}_0] = \max_{\tau \in \mathcal{S}_{0,T}} \mathbb{E}[Z_\tau|\mathcal{F}_0]. \tag{2.10}$$

τ^* gives the maximum expected payoff to the option holder, hence it defines the true option value to the holder. Any other stopping time $\tau \in \mathcal{S}_{0,T}$ will give a suboptimal expected payoff. Hence any attempt to pricing a Bermudan option by searching for close to optimal stopping time (exercise strategy) will in general end up in a lower bound estimate to the true option value. We use maximum instead of supremum because we are pricing a Bermudan option with finite exercise opportunity.

The stochastic process $\{V_n\}$:

$$\begin{cases} V_N = Z_N, & \\ V_n = \max_{\tau \in \mathcal{S}_{t_n,T}} \mathbb{E}[Z_\tau | \mathcal{F}_n], & \text{for } n = 0, \ldots, N-1. \end{cases} \quad (2.11)$$

is called the Snell Envelop of the payoff process $\{Z_n\}$ [53], which is proved to be a supermartingale. We will come back to this point in Chapter 3.

If we distinguish exercise choice at time T_n and later, the Snell Envelop can be written by the following expression,

$$V_n = \max \left\{ Z_n, \max_{\tau \in \mathcal{S}_{T_{n+1},T}} \mathbb{E}[Z_\tau | \mathcal{F}_n] \right\}$$

for each $n = 0, \ldots, N-1$.

The hold value is defined as $H_n = \mathbb{E}[V_{n+1}|\mathcal{F}_n] = \max_{\tau \in \mathcal{S}_{T_{n+1},T}} \mathbb{E}[Z_\tau|\mathcal{F}_n]$, which gives us the more familiar recursion for the Bermudan option value,

$$\begin{cases} V_N = Z_N, & \\ V_n = \max\left[Z_n, H_n\right], & \text{for } n = 0, \ldots, N-1. \end{cases} \quad (2.12)$$

Backward recursion (2.12) indicates that the holder should exercise the first time $Z_k > H_k$ in order to maximize payoff from the option. Define the exercise region as $\Psi_k = \left\{\omega \in \Omega, Z_k^{(\omega)} > H_k^{(\omega)}\right\}$ for each $k = 1, \ldots, N$. Define the sequence of optimal stopping time $\tau_k : \Omega \to \{k, \ldots, N\}$ for each $k = 1, \ldots, N$ as $\tau_k^{(\omega)} = \inf\{k \le j \le N, \omega \in \Psi_k\}$. τ_k can be interpreted as the exercise time of the Bermudan option given it has not been exercised up to time T_{k-1}. It can also be interpreted as the exercise time of a Bermudan option newly issued on time T_k with expiry at T. The dynamic programming formulation (2.12) can be rewritten in terms of the stopping times, as follows,

$$\begin{cases} \tau_N = N, & \\ \tau_n = n, & \text{if } \omega \in \Psi_n. \\ \tau_n = \tau_{n+1}, & \text{otherwise.} \end{cases} \quad (2.13)$$

for each $n = 0, \ldots, N-1$.

The option value at time T_n is then given by $V_n = \mathbb{E}[Z(\tau_n)|\mathcal{F}_n]$. In particular $V_0 = \mathbb{E}[Z(\tau_0)|\mathcal{F}_0]$. Making early exercise decision requires knowing or being able to approximate Z_k and H_k at each step (in certain situations such as a Bermudan option on a forward contract, both quantities are not known at time T_k). Recursion (2.13) is essential in deriving an algorithm for computer simulation of the lower bound for Bermudan option value. Approximation of Z_k and H_k can be realized by the Longstaff-Schwartz algorithm.

2.2.2 The Longstaff-Schwartz Algorithm

We will outline in this section a regression based method first studied by Carriere [16], Tsitsiklis and Van Roy [56] and further developed by Longstaff and Schwartz [43] for approximating the continuation value function. It allows Bermudan options to be valued by Monte Carlo. Longstaff and Schwartz named their method the Least Square Monte Carlo method (LSM), hence forth we will refer to this name for simplicity. Part of the description below is derived from Glasserman's book [28].

Let us consider again the setup from last section. Let a d dimensional $\mathcal{F}_t$ adapted Markov process $\{X(t), 0 \le t \le T\}$ be the state process, which contains all the observable information needed for option pricing. For simplicity, suppose the exercise value (also known as the intrinsic value) at time T_n can be fully determined by the information up to that moment, i.e. Z_n is $\mathcal{F}_n$ measurable. Our job is to approximate the hold value (also known as the continuation value) H_n at each $n = 1, \ldots, N-1$. We have $H_n = \mathbb{E}[V_{n+1}|\mathcal{F}_n] = \mathbb{E}[V_{n+1}|X_n]$ in this setup. The main idea of the LSM algorithm is to approximate the conditional expectation with linear combination of a set of $R+1$ basis functions, $\Phi_0, \Phi_1, \ldots, \Phi_R$, at each time step. Assume these basis functions are linearly independent. For each $n = 1, \ldots, N-1$, we assume the following multi-linear model,

$$\mathbb{E}[V_{n+1}|X_n] \approx \sum_{r=0}^{R} \beta_{n,r} \Phi_r(X_n). \quad (2.14)$$

We generate N independent sample paths of process X, denoted by X^k, for $k = 1, \dots, N$. The squared residual along path k is,

$$r_k^2 = \left[\mathbb{E}[V_{n+1}|X_n^k] - \sum_{r=0}^{R} \beta_{n,r} \Phi_r(X_n^k) \right]^2 .$$

As is discussed by Longstaff and Schwartz [43], since early exercise is relevant only when the option is in-the-money(ITM) for a put payoff, only ITM paths at time T_n are needed for regression. In our numerical simulations, we observe results that support this argument. We will formulate the expression according to this path-selection criterion. For some other options, for which the exercise value can not be observed explicitly, all the paths should be taken for regression. Assume there are N_{ITM} ITM paths indexed by k_1 to $k_{N_{ITM}}$. Ordinary Least Square (OLS) regression is performed to find the parameter vector $\beta_n = (\beta_{n,0}, \dots, \beta_{n,R})^T$ that minimizes the sum of all squared residuals,

$$\min_{\beta_n} \sum_{j=1}^{N_{ITM}} r_{k_j}^2 = \min_{\beta_n} \sum_{j=1}^{N_{ITM}} \left[\mathbb{E}[V_{n+1}|X_n^{k_j}] - \sum_{r=0}^{R} \beta_{n,r} \Phi_r(X_n^{k_j}) \right]^2 .$$

The OLS estimator [32] of β_n is

$$\hat{\beta}_n = (\mathbf{U}^T \mathbf{U})^{-1} \mathbf{U}^T y. \tag{2.15}$$

where entry (j, r) of matrix $\mathbf{U}$ is $U(k_j, r) = \Phi_r(X_n^{k_j})$, and entry j of column vector y is $y_{k_j} = \mathbb{E}[V_{n+1}|X_n^{k_j}]$. Φ_0 is often chosen to be 1 as a constant regressor. In the LSM backward recursion $\mathbb{E}[V_{n+1}|X_n^{k_j}]$ is replaced by $Z_{\tau_{n+1}}^{k_j}$, the discounted hold value at stopping time τ_{n+1} along path k_j.

Once we have determined the vector $\hat{\beta}_n$, the hold value of the option at time t_n along path k_j is estimated by the fitted value of regression

$$\hat{H}_n^{k_j} = \sum_{r=1}^{R} \hat{\beta}_{n,r} \Phi_r(X_n^{k_j}). \tag{2.16}$$

Thus we can compare the option's exercise value with its hold value and make exercise decision at time T_n, along those ITM Monte Carlo paths. For the out-of-the-money(OTM) Monte Carlo paths, it is clear that there will not be early exercise at the moment. Hence the option's value along those paths equals its hold value, i.e. the option's value at the next exercise moment. By repeating this procedure backwards in time, we can determine the LSM early exercise moment along each path according to recursion (2.13). The option value at the initial moment can be obtained by taking the average of discounted exercise cash flows along all Monte Carlo paths.

For computer simulation, we have implemented the following algorithm. Suppose the option owner has the right to exercise at all of the following moments $0 = T_0 < T_1 < ... < T_N = T$ except T_0. R is the number of nonconstant regressors. We use a vector for the remaining option values $V_{rm} = (V_{rm}(k), k = 1, \dots, N)$ to keep track of the discounted future exercise value, $Z_{\tau_{n+1}}^k$. At each time step, the vector is updated according to the exercise decision. In this way the sequence of stopping times do not need to be explicitly coded. Code efficiency is better than to record the stopping time and use that to find the exercise value from a matrix. The algorithm is as follows:

```
1. At maturity, $V_{rm}$ is evaluated by the option payoff function (with proper discounting back to time zero).
2. for $n = N-1 : -1 : 1$
    Calculate the discounted exercise value $Z_n^k$, for each path $k = 1, \dots, N_p$.
    if $N_{ITM} < R+1$
        Continue to the next recursion.
        (Not enough paths for regression, give up the exercise opportunity)
    else
        Pick all ITM paths at time point $t_n$ for regression:
        (these ITM paths are indexed by $k_1, \dots, k_{N_{ITM}}$.)
        Use Equation (2.15) to calculate the estimator $\hat{\beta}_n$.
        Use Equation (2.16) to calculate the fitted value $\hat{H}_n^{k_j}$, for each $j = 1, \dots, N_{ITM}$.
        If $Z_n^{k_j} > \hat{H}_n^{k_j}$, then $V_{rm}(k_j) = Z_n^{k_j}$, for each $j = 1, \dots, N_{ITM}$.
        end if
    end for
```

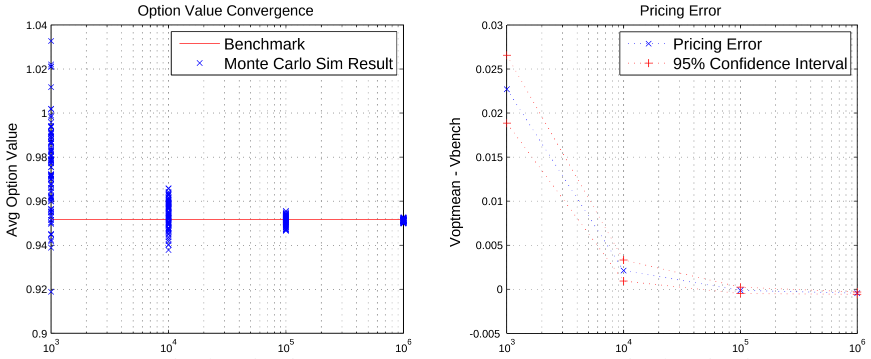
3. Average V_r over all the paths to get the Monte Carlo result of the Bermudan option value at time T_0, $V(T_0) = \frac{1}{N}\sum_{k=1}^{N} V_{rm}(k)$. Notice that there is no early exercise opportunity at time T_0 ■

2.2.3 Implementation Issues and Alternative Benchmarks

We will give some notes about implementing the LSM. In general, with R basis functions (and one constant) as the regressors, we need at least $R+1$ samples to make sure that it is possible for the regressor matrix $\mathbf{Z}$ to have full column rank $R+1$, so that the regressors are linearly independent. Otherwise, when the regressors are highly correlated, which is named multicollinearity [32], $\hat{\beta}$ may change dramatically with respect to data. But as Longstaff and Schwartz pointed out [43], the fitted value $\hat{H}_n$ is not affected by this problem. Since we are interested in the fitted value rather than the regression coefficients themselves, the only concern for us is whether there are enough ITM paths to carry out the regression. In our implementation, we perform the regression as long as there are at least $R+1$ ITM paths at a time step. If not, we simply say that there is not enough ITM paths, so we give up the exercise opportunity at that moment.

We set the first possible exercise moment to be $T_1 = \Delta T$ instead of T_0. At the initial time T_0, the underlying asset prices along all paths are the same. As a result, OLS regression can not be performed at T_0. For a Bermudan option in real life, there is usually a lockout period, i.e. the first possible early exercise date is set to be on a later date than the starting date of that option. So unable to run regression at time T_0 poses no problem. When approximating the value of an American option which can be exercised immediately after its starting time, this will result in a low-biased estimate for the option value especially when the option is initially deep ITM. However, if we make the time grid finer and finer, by the convergence result in [22], we expect the error to approach zero.

In order to characterize the performance of the LSM algorithm on Bermudan options, we need some alternative pricing benchmarks as a gauge of accuracy. The two numerical schemes that we implement are the Tree scheme and the Finite Difference scheme. An outline of pricing European and Bermudan options with these methods in the Black-Scholes world is listed in Appendix A and Appendix B. We realize that these methods can only approximate the true option value to certain error bound within limited computation time. Our goal of establishing alternative benchmarks is to approximate the true option value as close as possible, at least better than using the LSM method. Thus through comparing the pricing result from the LSM method and the benchmarks, we will be able to analyse the pricing error profile from the LSM method, and to obtain some insight of how different configurations of LSM implementation affect the pricing result.



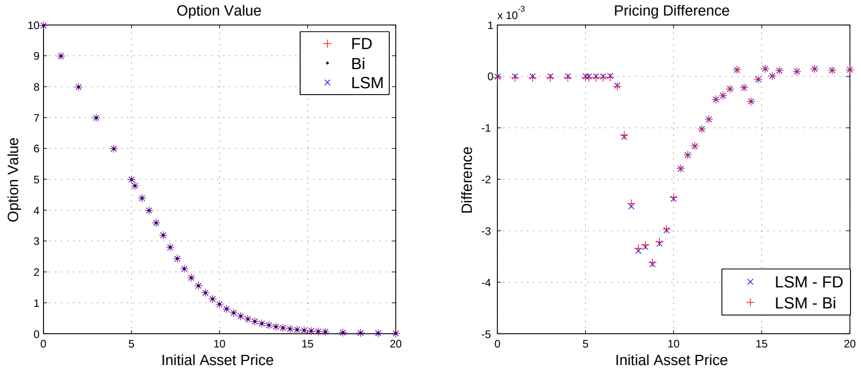
(a) Convergence of the LSM Result

(b) Price Difference

Figure 2.3: Convergence of the LSM result to a Binomial Tree benchmark for a Bermudan Put in the Black-Scholes world, $r = 0.06$, $\sigma = 0.3$, $S(0) = 10$, $K = 10$, $T = 1$, $\Delta t = T/52$. Regressors are 1, S, S^2.

A simple numerical example helps to show how the LSM method works. As Longstaff and Schwartz's first numerical example in Section 1 of [43], we choose the underlying asset price process S as the state process X described before. We choose a constant c, the underlying asset value S and its square S^2 as regressors. Figure 2.3 shows convergence of the LSM algorithm for an ATM Bermudan Put. The 95% confidence bound is calculated by the t-statistics based on 100 Monte Carlo simulation average. The benchmark result is obtained by a binomial tree with 52 × 400 = 20800 steps. In this way, all of the possible early exercise moments are on the time grid of the tree. Subplot (b) shows that the LSM results with 10^5 and 10^6 sample paths are low biased when compared to the benchmark.

We have evaluated prices of the Bermudan Put with different initial asset prices $S(0)$ and the following parameters, $K = 10$, $T = 1$, $r = 0.06$, $\sigma = 0.3$, $\Delta t = T/52$. 10^5 sample paths are generated in the Monte Carlo simulation for each $S(0)$. We take the average of 100 simulated option value as the result. The lockout period in our Monte Carlo simulation is Δt, i.e. the first early exercise opportunity occurs at $t = \Delta t$. We build a binomial tree with 20800 time steps and a finite difference solvers with 20800 time steps as benchmarks.



(a) Pricing Result

(b) Price Difference

Figure 2.4: Simulated price and price difference for a Bermudan Put in the Black-Scholes world by three numerical methods. $r = 0.06$, $\sigma = 0.3$, $K = 10$, $T = 1$, $\Delta t = T/52$, paths number $N_p = 10^5$, 100 run average. Regressors used in the LSM algorithm is 1, S and S^2. The benchmarks are provided by a Binomial Tree and a Finite Difference solver for the same Bermudan Put.

The simulated option prices and pricing differences are shown in Figure 2.4. We only give a brief discussion of the result here, detailed discussion is delegated to later chapters. The left plot shows that three numerical methods give quite close pricing result across different moneyness. The right plot gives the profile of pricing difference (LSM price − benchmark price). LSM price is found to be lower than the benchmark except for deep OTM region. The largest low bias occurs ITM, near the region where the early exercise boundary maybe.

For this simple setting, the LSM algorithm takes much longer computation time than either binomial tree or finite difference. The LSM method does not compete well either in convergence or computation efficiency with the alternative methods in one or two dimensional problems. However, in higher dimensional problems, it soon becomes infeasible to use the alternative methods for efficient pricing. That is when the LSM becomes the de facto tool.

2.2.4 Explanatory Variables and Basis Functions

Equation (2.14) in the last section gives the multi-linear model for the hold value function in LSM method, which we write again here,

$$\mathbb{E}[V_{n+1}|X_n] \approx \sum_{r=1}^{R} \beta_{n,r}\Phi_r(X_n).$$

In the model, X_n gives the value of the state process at time T_n, $X_n : \Omega \to \mathbb{R}^d$, $d \geq 1$. $\Phi_1, \ldots, \Phi_R$ are the basis functions, with $\Phi_r : \mathbb{R}^d \to \mathbb{R}$.

The above division into explanatory variable and basis function is a conceptual one. For actual implementation, boundary between the two is purely technical. We may directly evaluate some basis functions, such as the Laguerre polynomial, at the value of some underlying variables, such as some index value or interest rate value. Alternatively, we may use some function to map the underlying variables to some intermediate variables and then evaluate basis functions so that these value can better approximate the conditional expectation through the OLS regression. Exactly which way to choose depends on the property of the underlying model dynamics as well as the payoff feature. The intuition is to extract the most relevant information from the underlying variables so that the estimated hold value function is close to the true one. As Piterbarg pointed out in [45], success of the LSM depends on how robust and well-behaved the numerical problem of fitting the conditional expectations to simulated value is. Overfitting poses a significant danger, one should avoid using an overly rich set of parametric functions in the fitting. A less robust fitting may find a good fit within the range of fitted values but produces completely unreasonable outcome outside this range. For example, high order polynomials reacts very strongly to out of the range data. Piterbarg proposed to use financially meaningful variables that actually drive the future exercise and/or hold value. For the basis functions, he suggests to use very simple parametric families such as polynomials of degree 2.

We will show in later chapters how we choose proper explanatory variables and basis functions according to each specific model and payoff. We have collected extensive numerical evidence to validate our choice.

It is important to note that one should only use the $\mathcal{F}_n$ measurable variables to construct the regressors, because the conditional expectation is based on information available up to time T_n. Using information in future time will in general lead to high biased estimator for the option price. A related issue is about facilitating the OLS regression method we use. We find that specific choices of the regressor setup has a big impact on the result. This will be dealt with in the section concerning numerical results.

We have experimented with different types of commonly used polynomials as basis functions. These include Power series, Legendre polynomials, (weighted) Laguerre polynomials, (physicists) Hermite polynomials. For details on the properties of different polynomials, one can turn to Abramowitz and Stegun [2]. We show here the recurrence law that we use to construct the above mentioned polynomials. Given the appropriate coefficients and starting value, the polynomials satisfies the following recursion,

$$a_{n+1}f_{n+1}(x) = (a_n + b_n x)f_n(x) - a_{n-1}f_{n-1}(x).$$

where the coefficients and starting values are given in Table 2.2.

Table 2.2: The coefficients and starting value for the recurrence law of the polynomials.

Name	$f_n(x)$	a_{n+1}	a_n	b_n	a_{n-1}	$f_0(x)$	$f_1(x)$
Powers	$W_n(x)$	1	0	1	0	1	x
Legendre	$G_n(x)$	$n+1$	0	$2n+1$	n	1	x
Laguerre	$L_n(x)$	$n+1$	$2n+1$	-1	n	1	$1-x$
(phys)Hermite	$H_n(x)$	1	0	1	1	1	$2x$

The weighted Laguerre polynomials used by Longstaff and Schwartz are constructed by multiplying the Laguerre polynomials with $e^{-x/2}$,

$$Gw_n(x) = e^{-x/2}G_n(x),$$

for each $n = 0, 1, \ldots$. The first 10 Laguerre and weighted Laguerre polynomials are plotted in the Figure 2.5. It is clear that on $(0, +\infty)$ the exponential term scales the polynomial so that its value stays bounded in a narrow range. This property helps to reduce numerical error in the LSM, since we need to invert a matrix constructed by the regressors. We will show in numerical experiments that in several settings, weighted Laguerre polynomials do give better pricing result compared to other unscaled polynomials.

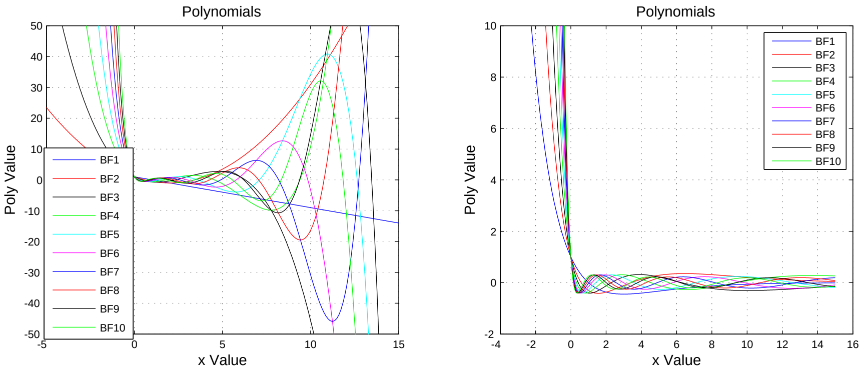
2.2.5 Exercise Boundary, Pricing Error and Convergence Property

In this section, we will discuss pricing error of the LSM and how it is related to the error in approximating the true early exercise boundary. We will also give an overview of the convergence property of the LSM in literatures. We will use the definitions and notations in Section 2.2.1 and Section 2.2.2.

Recall that $\mathcal{F}_t$ measurable functions $Z(t)$, $H(t)$ and $V(t)$ represent the discounted exercise value, the discounted hold value, and the discounted value of the Bermudan option at time t with some chosen numeraire. The following discussion is partly derived from [45].

Dynamic programming recursion 2.12 gives that

$$\begin{aligned} H(T_n) &= \mathbb{E}\left[V(T_{n+1})|\mathcal{F}_n\right], \\ &= \mathbb{E}\left[\max(Z(T_{n+1}), H(T_{n+1}))|\mathcal{F}_n\right], \end{aligned} \tag{2.17}$$



(a) Laguerre polynomials (b) weighted Laguerre polynomials

Figure 2.5: Plot of first 10 Laguerre and weighted Laguerre polynomials

We show explicitly the dependency on the underlying probability space and define the exercise region at time T_n by Ψ_n,

$$\Psi_n = \{\omega \in \Omega : H(T_n, \omega) \leq Z(T_n, \omega)\}, n = 1, \ldots, N. \tag{2.18}$$

The early exercise boundary is defined as the set

$$EB_n = \{\omega \in \Omega : H(T_n, \omega) = Z(T_n, \omega)\}, n = 1, \ldots, N.$$

Let $\eta = \eta(\omega)$ be the index that the exercise region is hit for the first time (or N if it is never hit before T),

$$\eta(\omega) = \min\{n \geq 1 : \omega \in \Psi_n\} \wedge N.$$

The time 0 Bermudan option value can be expressed by,

$$V(0) = \mathbb{E}\left[Z(T_\eta)|\mathcal{F}_0\right]. \tag{2.19}$$

Writing the option pricing formula in this form helps to reveal how the the accuracy in approximating the option's hold value and the accuracy in approximating the early exercise boundary(exercise region) affect the accuracy of the LSM simulated option price.

At each early exercise moment T_n, OLS regression approximates the hold value function by the multilinear model,

$$\hat{H}(T_n, x) = \sum_{r=1}^{R} \hat{\beta}_{n,r} \Phi_r(x),$$

where x is the value of the state process at time T_n. The hold value along each sample path is then calculated using this function. Finite terms of regressors introduce an error in the hold value function. We can view this error as a function of x indexed by time T_n. The error in hold value approximation at time T_n affects the exercise decision at that moment according to (2.18). Moreover, through the backward dynamic programming recursion (2.17), the error affects all the exercise decisions at earlier time steps. In this way, error in hold value approximation at each early exercise moment would have an impact in the time 0 value of the option.

Definition of the exercise region (2.18) and the Bermudan price formulation based on first hit time of the exercise region (2.19) indicate that the error in hold value approximation affects the option value the most when it is made near the true early exercise boundary, i.e. when the hold value and the exercise value are close to each other. That is because the error in $\hat{H}(T_n, x)$ makes an impact on the option price only when it affects the exercise decision. That happens when the error is significant enough to change the approximation of the exercise regions and subsequently the first hit moment η. Since $H(T_n, x)$ and $Z(T_n, x)$ are close to each other near the true exercise boundary, a small error in $\hat{H}(T_n, x)$ may change the approximation of Ψ_n. On contrary, when the error occurs far away from the the true exercise boundary, it takes a lot higher value to influence the approximation of Ψ_n. In view of this, we should provide best approximation accuracy near the true early exercise boundary. For some options where the strike is fixed, such as a Bermudan put, we have a rough idea of where the boundary may be [50]. This information proves vital in improving the approximation accuracy in such cases. We will show relevant examples in the sections about numerical results. For other options such as a Bermudan Swaption with both legs being functions of some market rates, we do not know a priori where the boundary is. The effort to improving pricing accuracy then focus on finding the better regressors, i.e. explanatory variables and basis functions that makes better approximation.

We have qualitatively discussed how error in hold value function affects the LSM Bermudan option price above. Rigorous analysis of the method requires the asymptotic property of the LSM estimator. Finite number of sample paths affects the accuracy in estimating the regression coefficients for approximating the continuation value function, which affects the approximating the hold value function. Finite number of sample paths also introduces a general Monte Carlo error. The type and number of terms of regressors (basis functions) affects the accuracy of the regression coefficients as well. What further complicates the matter is that the algorithm is carried out through a backward recursion, error introduced at one stage propagates in a highly nonlinear way.

Longstaff and Schwartz [43] established convergence of the LSM estimator to the true option value with single state variable under a one-step setting. They chose the Laguerre polynomial as basis function. The number of terms was chosen large enough to satisfy certain error bound in their proof. Clement et. al. [17] gave a more general convergence analysis. They chose a general sequence of basis functions that is total in the L^2 space. They first showed that under certain assumption on the basis functions, at each time step $n = 1, ..., N$, the approximation to the Snell Envelop using m terms of basis functions converges as $m \to \infty$,

$$\lim_{m \to +\infty} \mathbb{E}\left[Z_{\tau_n^m} | \mathcal{F}_n\right] = \mathbb{E}\left[Z_{\tau_n} | \mathcal{F}_n\right],$$

in L^2. Then they fixed the number of basis functions m and showed that for each step $n = 1, \ldots, N$, the LSM estimator using m basis functions and N_p sample paths,

$$\frac{1}{N_p} \sum_{k=1}^{N_p} Z^{(k)}_{\tau_n^{k,m,N_p}},$$

converges to the $\mathbb{E}\left[Z_{\tau_n^m} | \mathcal{F}_n\right]$ as $N_p \to \infty$. They have also given results concerning the rate of convergence in the same paper.

The analysis by Clement et. al. establishes the asymptotic properties of the LSM estimator. In computer Monte Carlo simulation, one can only use finite number of regressors (or basis functions). Actually, the number of basis functions used is often not large at all. For example, in the numerical example provided by Longstaff and Schwartz, they used 4 terms for a Bermudan Put, 8 terms for a Bermudan-Asian option, 9 terms for a Cancellable Index Swap in a two-factor model and 22 terms for a Swaption in a 20-factor string model. In [29], Glasserman and Yu discussed how fast the number of sample paths needed for certain convergence requirement grows when the number of basis functions increases. They chose basis functions to be polynomials and studied the problem in both Brownian Motion and Geometric Brownian Motion setting of the underlying single asset price dynamics. They placed bounds on the Mean Square Error (MSE) of the regression coefficients and derived the number of paths it takes to satisfy those bounds. They claimed that the number of paths grows at least exponentially in both settings and expect this property to also apply in more general settings. Although in several thesis this was found to be not true through numerical experiments [55], it does raise the question about whether the number of sample paths is sufficient for convergence of the LSM price. In the original paper by Longstaff and Schwartz, they showed that for any fixed finite number of basis functions, the LSM price converges in probability to the option price with a suboptimal exercise rule when the number of paths goes to infinity. And, by definition, this value is lower than the true buyer option value given by the optimal exercise rule.

3 Upper Bound for Bermudan Option Price

In Chapter 2, we discussed how to calculate the buyer's price of a Bermudan option. The method is based on a formulation of the problem in terms of optimal exercise strategy. In this chapter, we will study the same option pricing problem but from the writer's or seller's perspective. Calculating the buyer's price is a maximization problem. We will see that calculating the seller's price is a minimization problem. Hence in literatures these two methods are labelled as the primal and dual approaches for pricing Bermudan options respectively.

As mentioned in Chapter 2, in theory, suboptimal exercise policy by the LSM method should result in negatively biased pricing result. A commonly used rule of thumb criterion to evaluate the LSM price can be stated as follows: the higher the LSM price is, the closer it is believed to be near the true option value. What is missing in this argument is that the LSM price should be fully converged. However, in practice, LSM simulations are often carried out with limited number of sample paths. Considering the complexity of this algorithm, we need to gather numerical results to show convergence. Only using the level of LSM price as an evaluation criterion may lead to wrong conclusion. We need a better criterion to evaluate the LSM price by, such that we can compare between different regressor configurations with confidence. It turned out that the upper bound for Bermudan option price, or more precisely the gap between the lower bound price and its associated upper bound price, can be used as such a criterion. In this chapter we will discuss the theory and implementation details of this method.

3.1 Dual Approach: The Upper Bound

Let's recall that according to the primal approach, the buyer's value is defined by the maximal expected payoff from the option. Equation (2.10) gives the mathematical formulation, which we write again here for ease of reference.

$$V_0 = \mathbb{E}[Z_{\tau^*}|\mathcal{F}_0] = \max_{\tau \in \mathcal{S}_{0,T}} \mathbb{E}[Z_\tau|\mathcal{F}_0].$$

The dual approach was developed by Haugh and Kogan [31], Rogers [48], as well as Jamshidian [39]. We will follow Rogers's construction but on a discrete time grid of N exercise moments. Given the grid $0 = T_0 < T_1 < \ldots < T_N = T$, $\mathcal{T} = \{T_1, \ldots, T_N\}$ contains all the possible exercise moments. Using the notations from Section 2.2.1, we start from the primal approach formulation.

Let H represent the space of all adapted martingales π for which $\sup_{t\in\mathcal{T}} |\pi_t| < \infty$ and $\pi_0 = 0$. For a martingale $\pi \in H$, we have:

$$\begin{aligned} V_0 &= \max_{\tau \in \mathcal{S}_{0,T}} \mathbb{E}[Z_\tau - \pi_\tau + \pi_\tau|\mathcal{F}_0] \\ &= \pi_0 + \max_{\tau \in \mathcal{S}_{0,T}} \mathbb{E}[Z_\tau - \pi_\tau|\mathcal{F}_0] \\ &\le \pi_0 + \mathbb{E}[\max_{n\in\mathcal{T}} (Z_n - \pi_n)\,|\mathcal{F}_0]. \end{aligned} \tag{3.1}$$

The second equality in (3.1) is due to the Martingale Optional Sampling theorem [49]. The inequality in (3.1) is due to the fact that $\mathcal{S}_{0,T} \subset \mathcal{T}$. Since the inequality holds for any $\pi \in H$, it also holds when taking the infimum over all $\pi \in H$,

$$V_0 \le \inf_{\pi\in H} \left(\pi_0 + \mathbb{E}[\max_{n\in\mathcal{T}} (Z_n - \pi_n)\,|\mathcal{F}_0] \right). \tag{3.2}$$

Rogers showed that when choosing π to be the martingale part of the supermartingale V (which is the Snell Envelop of the payoff process), according to the Doob decomposition theorem [21], equality holds in (3.1). Since all the other martingales in the space H satisfies the inequality (3.1), holding equality means that the martingale part of V achieves the infimum in (3.2). Now we can formulate the problem of Bermudan option pricing as a minimization problem,

$$V_0 = \inf_{\pi\in H} \left(\pi_0 + \mathbb{E}[\max_{n\in\mathcal{T}} (Z_n - \pi_n)\,|\mathcal{F}_0] \right). \tag{3.3}$$

Interpretation of Equation (3.3) is as follows. The martingale π is regarded as the discounted price process of a hedging portfolio. The option seller has to prepare for any potential buyer exercise during the life time of the option. By following certain strategy that gives a martingale portfolio π, the option seller has a hedge against potential buyer exercise. The expected cost to the seller is the expected maximum shortfall of the hedge. This value is taken as the seller's price of the Bermudan option. The optimal hedging portfolio is the one that minimizes the hedging cost, it's price process should be the martingale part of V.

Equality (3.3) states that when choosing the optimal hedging portfolio, the seller's price coincides with the buyer's price. Doob's decomposition only proves its existence. But in general, we do not know exactly how this martingale evolves over time. In order to get a close upper bound, we may try to construct a martingale π from a good approximation of the Snell Envelop V. The buyer's value process,

$$\{L_n = \mathbb{E}[Z_{\tau_n}|\mathcal{F}_n], n = 1, \ldots, N\}, \tag{3.4}$$

associated with a specific stopping rule is a good candidate, for example the stopping rule defined by the LSM method.

We construct a simplified version of the martingale proposed by Anderson and Broadie [6]. Their numerical results show excellent performance. The martingale is defined as,

$$\begin{cases} \hat{\pi}_0 = L_0, \\ \hat{\pi}_n = \hat{\pi}_{n-1} + L_n - \mathbb{E}\left[L_n|\mathcal{F}_{n-1}\right] \qquad \text{for } n = 1, \ldots, N. \end{cases} \tag{3.5}$$

We can write

$$\begin{aligned} \mathbb{E}\left[\hat{\pi}_n|\mathcal{F}_{n-1}\right] &= \mathbb{E}\left[\hat{\pi}_{n-1} + L_n - \mathbb{E}\left[L_n|\mathcal{F}_{n-1}\right]|\mathcal{F}_{n-1}\right] \\ &= \hat{\pi}_{n-1} + \mathbb{E}\left[L_n|\mathcal{F}_{n-1}\right] - \mathbb{E}\left[L_n|\mathcal{F}_{n-1}\right] \\ &= \hat{\pi}_{n-1}. \end{aligned}$$

So, $\hat{\pi}$ satisfies the Martingale identity. Notice the initial value $\hat{\pi}_0$ equals a positive constant L_0. Since adding or subtracting a constant from a martingale still gives a martingale, $\pi_n = \hat{\pi}_n - \hat{\pi}_0$ is the martingale with zero initial value as required in the dual problem.

In order to construct this martingale, we need the following two conditional expectations,

$$L_n = \mathbb{E}\left[Z_{\tau_n}|\mathcal{F}_n\right], \tag{3.6}$$

and

$$\begin{aligned} \mathbb{E}\left[L_n|\mathcal{F}_{n-1}\right] &= \mathbb{E}\left[\mathbb{E}\left[Z_{\tau_n}|\mathcal{F}_n\right]|\mathcal{F}_{n-1}\right] \\ &= \mathbb{E}\left[Z_{\tau_n}|\mathcal{F}_{n-1}\right], \end{aligned} \tag{3.7}$$

for each $n = 1, \ldots, N$. Once we have the martingale, we can derive from Inequality (3.1) that the gap between the lower and upper bound is

$$G = \mathbb{E}[\max_{n \in \mathcal{T}} \left(Z_n - \hat{\pi}_n\right)|\mathcal{F}_0]. \tag{3.8}$$

The upper bound is then calculated by $U_0 = L_0 + G$ as an approximation to the true option value by the dual problem (3.3).

3.2 The Nested Monte Carlo

Equations (3.6) and (3.7) indicate that at each possible exercise moment, we need to approximate the two conditional expectations. But as Glasserman [28] pointed out, the workload can be reduced due to recursion (2.12). Let's write the backward recursion in terms of the lower bound estimate,

$$L_n = \begin{cases} Z_n, & \text{if } Z_n > H_n, \\ \mathbb{E}[L_{n+1}|\mathcal{F}_n], & \text{otherwise.} \end{cases} \tag{3.9}$$

Plug in Equations (3.6) and (3.7), we have

$$\mathbb{E}\left[Z_{\tau_n}|\mathcal{F}_n\right] = \begin{cases} Z_n, & \text{if } Z_n > H_n, \\ \mathbb{E}\left[Z_{\tau_{n+1}}|\mathcal{F}_n\right], & \text{otherwise.} \end{cases} \tag{3.10}$$

So we only need to evaluate $\mathbb{E}\left[Z_{\tau_{n+1}}|\mathcal{F}_n\right]$ for $n = 0, \ldots, N-1$. In Monte Carlo simulation, we approximate this conditional expectation at each tenor moment before maturity along each sample path. The actual calculation is implemented by a nested Monte Carlo.

Given that we have already obtained estimate of the hold value function (and the exercise value function if necessary) at each step from the LSM method, the upper bound estimation proceeds by repeating steps 1 to 3 and calculating the estimate by step 4:

1. Simulate a path of the underlying Markov chain, $X^q = \{X^q_0, X^q_1, \dots, X^q_N\}$, for the outer Monte Carlo. q denotes the qth outer sample path, $q = 1, \dots, N_{out}$.
2. At each step T_n, for $n = 0, \dots, N-1$,
   Generate N_{in} sample paths for the inner Monte Carlo from time T_n to T_N with initial value $X^{qp}_n = X^q_n$. p denotes the pth inner sample path, $p = 1, \dots, N_{in}$.
   Along each inner Monte Carlo path, follow the LSM exercise rule and calculate the payoff $Z_{\tau_{n+1}}$. Use the average of these payoffs as an estimate of the conditional expectation $\mathbb{E}\left[Z_{\tau_{n+1}}|\mathcal{F}_n\right]$.
   Evaluate H^q_n and Z^q_n by the LSM estimated functions along path X^q at time step T_n. Then use Equation (3.10) to determine $\mathrm{E}[Z_{\tau_n}|\mathcal{F}_n]$.
3. Use Equation (3.5) to construct the martingale along the qth outer path, $\pi^q = \{\pi^q_0, \pi^q_1, \dots, \pi^q_N\}$. Evaluate the maximum gap $G = \max_{n \in \mathcal{T}} (Z^q_n - \pi^q_n)$.
4. Estimate the gap in Equation (3.8) by averaging over all outer paths, then calculate the upper bound by $U_0 = L_0 + G$. ■

3.3 Tightness of the Upper Bound

Andersen and Broadie [6] gave an analysis of the tightness of the upper bound. Let's first introduce the Doob decomposition [21] of a discrete supermartingale. Given a discrete supermartingale $\{Y_n\}$, it admits the following unique decomposition,

$$Y_n = M_n - A_n \tag{3.11}$$

where $\{M_n\}$ is a martingale with $M_0 = Y_0$, $\{A_n\}$ is an increasing process with $A_0 = 0$, A_n is $\mathcal{F}_{n-1}$ measurable, i.e. $\mathbb{E}[A_n|\mathcal{F}_{n-1}] = A_n$.

We start with Equation (2.11), $V_n = \max_{\tau \in \mathcal{S}_{T_n,T}} \mathbb{E}[Z_\tau|\mathcal{F}_n]$. Let L_n be the lower bound price estimate at step n. Define the lower bound error as $e_n = V_n - L_n$, for each $n = 0, 1, \dots, N$. We can write Recursion (3.5) as a summation, plug in $L_n = V_n - e_n$, and apply Doob decomposition to write the hedging martingale in terms of the martingale part of the Snell Envelop and the lower bound errors,

$$\begin{aligned}
\pi_n &= L_0 + \sum_{k=0}^{n-1} (L_{k+1} - \mathbb{E}[L_{k+1}|\mathcal{F}_k]) \\
&= V_0 - e_0 + \sum_{k=0}^{n-1} (V_{k+1} - e_{k+1} - \mathbb{E}[V_{k+1} - e_{k+1}|\mathcal{F}_k]) \\
&= M_0 - e_0 + \sum_{k=0}^{n-1} (M_{k+1} - A_{k+1} - e_{k+1} - \mathbb{E}[M_{k+1} - A_{k+1} - e_{k+1}|\mathcal{F}_k]) \\
&= M_0 + \sum_{k=0}^{n-1} (M_{k+1} - \mathbb{E}[M_{k+1}|\mathcal{F}_k]) - e_0 + \sum_{k=0}^{n-1} (-A_{k+1} - e_{k+1} + \mathbb{E}[A_{k+1} + e_{k+1}|\mathcal{F}_k]) \\
&= M_n - e_0 - \sum_{k=0}^{n-1} (e_{k+1} - \mathbb{E}[e_{k+1}|\mathcal{F}_k]) .
\end{aligned}$$

Now the upper bound can be written as

$$\begin{aligned}
U_0 &= L_0 + \mathbb{E}[\max_{n \in \mathcal{S}_{0,T}} (Z_n - \pi_n) |\mathcal{F}_0] \\
&= L_0 + \mathbb{E}\left[\max_{n \in \mathcal{S}_{0,T}} \left(Z_n - M_n + e_0 + \sum_{k=0}^{n-1} (e_{k+1} - \mathbb{E}\left[e_{k+1}|\mathcal{F}_k\right])\right) |\mathcal{F}_0\right] \\
&\le V_0 + \mathbb{E}\left[\max_{n \in \mathcal{S}_{0,T}} \left(\sum_{k=0}^{n-1} (e_{k+1} - \mathbb{E}\left[e_{k+1}|\mathcal{F}_k\right])\right) |\mathcal{F}_0\right] .
\end{aligned}$$

The last inequality is due to $Z_n \leq V_n \leq M_n$, for each $n = 1, \ldots, N$. If we assume that $e_{n+1} \geq 0$, then $\mathbb{E}\left[e_{n+1}|\mathcal{F}_n\right] \geq 0$. We have the following bound for the dual price,

$$U_0 \leq V_0 + \mathbb{E}\left[\max_{n \in \mathcal{S}_{0,T}} \left(\sum_{k=0}^{n-1} e_{k+1}\right)\right] \leq V_0 + \mathbb{E}\left[\sum_{k=1}^{N} e_k\right]. \tag{3.12}$$

Hence if $e_k \leq \epsilon$ for each $k = 1, \ldots, N$, then $U_0 - V_0 \leq N \times \epsilon$.

The above result states that if the lower bound error is uniformly bounded by ϵ at all possible exercise moments, the corresponding upper bound error should be bounded by $N \times \epsilon$. The key assumption here is that $e_n = V_n - L_n \geq 0$.

3.4 Implementation Issues

In our simulation of the upper bound price, we first take more than 10^6 sample paths for the LSM algorithm in order to get a stable estimate of the regression coefficients. We take a second set of 10^6 to 10^7 sample paths to estimate the LSM price. This is done in order to eliminate the foresight bias according to [25]. Calculation of the upper bound requires nested Monte Carlo simulation, which is computationally expensive. We choose 10^3 outer sample paths and 10^3 inner sample paths as a compromise between result variability and computation cost. We will give numerical result of convergence for the upper bound result in Chapter 7.

Since the LSM regression coefficients will be used for computing the upper bound, simulation configurations of the LSM algorithm have a strong impact on the upper bound result as well. In particular, the choice of paths for the LSM regression plays a big role. If LSM regression is performed with only ITM paths, the resulting gap between the upper and lower bound will be significantly higher when the option is initially OTM as compared to ITM or ATM. When all the paths are used for LSM regression, such difference in the gap value does not appear. This is mainly due to the out-of-sample-range performance of the OLS regression fitted multi-linear model. When only ITM paths are chosen for regression, the resulting multi-linear model might be wildly irregular in OTM region. Hence when using this model to compute the continuation value in the upper bound algorithm, it is not surprising that large error may occur. Such error lead to error in the approximated optimal exercise strategy or stopping rule.

The Anderson-Broadie upper bound is basically expectation of the maximal expected shortfall by the hedger with certain exercise strategy. It is a stochastic variable which is very likely to have a significant tail distribution. This means that its samples are intrinsically volatile. When the approximated stopping rule contains error due to regression configuration, the hedging portfolio can not track the option value closely and the gap between lower and upper bound is due to widen. Fortunately, we use all the sample paths in the LSM algorithm for pricing Bermudan-style Libor Exotics. The problem of moneyness in upper bound value does not exist in that case. We will discuss the details in Chapter 6.

4 The Black-Scholes World and the Heston Model

To get an idea about how the LSM method performs in various models and simulation setups, we start with simpler cases, namely the GBM asset price model in the Black Scholes world and the Heston stochastic volatility model. We will present Monte Carlo schemes for these models, introduce several option payoff types on single asset, and briefly discuss the FD schemes for solving these option values. In Section 4.2.2, we will discuss the source of pricing error in the LSM through a concrete pricing problem. These all serve as the test cases for our study of Bermudan option pricing. Knowledge and experience gained from these cases will be applied to pricing Bermudan-style Libor Exotics in the next chapter. More numerical results and detailed analysis will also be presented in later chapters.

4.1 Models

4.1.1 The Black-Scholes World

In the Black-Scholes world, we assume a risk-free money market account with the dynamics,

$$dB(t) = rB(t)dt$$

where constant r is the risk-free interest rate. Assume there is one underlying asset price process S, whose dynamics under the EMM $\mathbb{Q}$ is given by the GBM SDE,

$$dS(t) = rS(t)dt + \sigma S(t)dW(t), \qquad 0 \leq t \leq T.$$

where σ is the constant volatility. W is a $\mathbb{Q}$ standard Brownian Motion.

Suppose we discretize the time domain $[0, T]$ by grid $0 = t_0 < t_1 < ... < t_N = T$ with step Δt. The above SDE can be solved explicitly, the procedure is presented in Section 2.1. We write the result here again for the ease of reference.

$$S(t + \Delta t) = S(t) \exp \left[(r - \frac{1}{2}\sigma^2)\Delta t + \sigma(W(t + \Delta t) - W(t)) \right].$$

From this equation we find that the log-return is normally distributed,

$$\ln \left(\frac{S(t + \Delta t)}{S(t)} \right) \sim \mathcal{N} \left((r - \frac{1}{2}\sigma^2)\Delta t, \sigma^2 \Delta t \right).$$

Since Brownian Motion W has independent increments on non-overlapping time intervals, the log-return on different intervals are independent normal random variables. Hence by the property of independent normal random variables, we have

$$\ln \left(\frac{S(T)}{S(0)} \right) \sim \mathcal{N} \left((r - \frac{1}{2}\sigma^2)T, \sigma^2 T \right), \tag{4.1}$$

which states that the distribution of the return of a GBM asset price on [0,T] follows a lognormal distribution with parameters $m = (r - \frac{1}{2}\sigma^2)T$ and $s^2 = \sigma^2 T$. The value of a plain vanilla option can be derived from this distribution [35], which gives the famous Black-Scholes formula. In order to price a path dependent option or option with early exercise feature in the Monte Carlo framework, we need to generate the whole sample path of asset price. The procedure is also given in section 2.1.

Since the Black-Scholes world is the starting point of equity dynamics modeling, we are going to investigate the performance of the LSM method in this setting with several types of payoffs. The payoff types and their focus will be discussed in the following sections.

The short coming of the GBM asset price model is that by assuming a constant volatility, the model does not fit the empirical data well. Empirical studies show that the log-return of asset price is not normally distributed [35]. Its distribution is often featured by heavy tail and high peak. Empirical evidence also show that equity return and implied volatility are often negatively correlated. These deviations from normality call for models that can reflect the reality more closely in order to obtain more realistic pricing result.

4.1.2 The Heston Model

In 1993, Heston [33] introduced the Heston stochastic volatility model as an extension to the constant volatility GBM asset price model. He gave a semi-analytical solution for the value of plain vanilla option under the model. This is a big advantage over some other models because it is possible to calibrate the model parameters to market data according to this solution. Besides, the solution allows to benchmark some numerical schemes for the Heston model. We apply the LSM algorithm to option in the Heston model in order to find out how to best utilize the stochastic volatility to improve simulation accuracy. A question to be answered is whether and how the correlation coefficient affects the LSM simulation.

The Heston model under EMM $\mathbb{Q}$ can be written as,

$$dS(t) = rS(t)dt + S(t)\sqrt{v(t)}\left[\rho dW_1(t) + \sqrt{1-\rho^2}dW_2(t)\right], \tag{4.2}$$

$$dv(t) = \kappa(\eta - v(t))dt + \sigma_v\sqrt{v(t)}dW_1(t), \tag{4.3}$$

where S is the asset price process on $t \geq 0$, v is the volatility process on $t \geq 0$, W_1 and W_2 are two independent $\mathbb{Q}$ Brownian Motions on $t \geq 0$. v is a square-root diffusion process with mean-reverting drift that was first proposed by Cox, Ingersoll and Ross [18] as an interest rate model. The constant parameters are the rate of reversion $\kappa > 0$, the long term mean $\eta > 0$ and the volatility of volatility $\sigma_v > 0$. ρ represents the correlation between the stochastic drivers of $S(t)$ and $v(t)$. r is the risk-free constant interest rate. We consider the model when the Feller condition $2\kappa\eta < \sigma_v^2$ is satisfied [4].

The associated PDE for simple payoff on time interval $[0, T]$ is,

$$\frac{\partial U}{\partial \tau} = \frac{1}{2}s^2 v\frac{\partial^2 U}{\partial s^2} + \rho\sigma_v sv\frac{\partial^2 U}{\partial s\partial v} + \frac{1}{2}\sigma_v^2 v\frac{\partial^2 U}{\partial v^2} + rs\frac{\partial U}{\partial s} + \kappa(\eta - v)\frac{\partial U}{\partial v} - rU. \tag{4.4}$$

for $0 \leq \tau \leq T$ and $\tau = T - t$, $s > 0$, $v > 0$. A Finite Difference scheme can be derived from this PDE, for details please refer to [34].

Let's have a look at the asset dynamics used in Monte Carlo simulation. The Feller condition guarantees that for the continuous model if $v(0) > 0$, then $v(t)$ remains strictly positive for all $0 < t < \infty$ almost surely. However, if we discretize the variance process by the Euler scheme, it is still possible that the discretized variance process takes negative value at some step. One possible walk around is to use the following scheme which truncates the variance process when it hits zero.

$$\begin{aligned} ln\hat{S}(t+\Delta t) &= ln\hat{S}(t) - \frac{1}{2}\hat{v}(t)^+\Delta t + \sqrt{\hat{v}(t)^+}Z_s\sqrt{\Delta t}, \\ \hat{v}(t+\Delta t) &= \hat{v}(t) + \kappa(\eta - \hat{v}(t)^+)\Delta t + \sigma_v\sqrt{\hat{v}(t)^+}Z_v\sqrt{\Delta t}. \end{aligned}$$

where Z_s and Z_v are two correlated standard normal random variables. The function $x^+ = \max[x, 0]$. However, this choice is highly empirical and it alters the property that variance will not reach zero. So we need a scheme that preserves the model dynamics as closely as possible, while allowing easy Monte Carlo implementation.

As pointed out by Glasserman [28], Cox et al.[18] noticed that the distribution of the CIR process conditioned on its value at an earlier moment is, up to a scale factor, a noncentral chi-square distribution. In terms of Equation (4.3), this means that given the value of the variance process at time t, $v(t)$, its value at time $t + \Delta t$ is given by,

$$v(t+\Delta t) = \frac{\sigma_v(1-e^{-\kappa\Delta t})}{4\kappa}\chi_d^{'2}\left(\frac{4\kappa e^{-\kappa\Delta t}}{\sigma_v^2(1-e^{-\kappa\Delta t})}v(t)\right), \tag{4.5}$$

where $d = \frac{4\kappa\eta}{\sigma_v^2}$, and $\chi_d^{'2}(\lambda)$ is a noncentral chi-square random variable with d degrees of freedom and λ as the noncentrality parameter. The variance process can thus be sampled by this transition density. A procedure for sampling a noncentral chi-square distribution is given in [28].

To derive a scheme for sampling the price process, we first integrate the SDE 4.3,

$$v(t+\Delta t) = v(t) + \int_t^{t+\Delta t}\kappa(\eta - v(u))du + \sigma_v\int_t^{t+\Delta t}\sqrt{v(u)}dW_1(u).$$

After reshuffling the equation, we have

$$\int_t^{t+\Delta t}\sqrt{v(u)}dW_1(u) = \frac{1}{\sigma_v}\left(v(t+\Delta t) - v(t) - \kappa\eta\Delta t + \kappa\int_t^{t+\Delta t}v(u)du\right). \tag{4.6}$$

Applying Itô's lemma to (4.2) with $f(x) = ln(x)$, we have

$$dlnS(t) = (r - \frac{1}{2}v(t))dt + \rho\sqrt{v(t)}dW_1(t) + \sqrt{1-\rho^2}\sqrt{v(t)}dW_2(t).$$

Integrating it gives us

$$\begin{aligned} lnS(t+\Delta t) &= lnS(t) + r\Delta t - \frac{1}{2}\int_t^{t+\Delta t} v(u)du + \rho\int_t^{t+\Delta t}\sqrt{v(u)}dW_1(u) \\ &+ \sqrt{1-\rho^2}\int_t^{t+\Delta t}\sqrt{v(u)}dW_2(u). \end{aligned} \tag{4.7}$$

Substituting (4.6) into (4.7), we have

$$\begin{aligned} lnS(t+\Delta t) &= lnS(t) + r\Delta t + \frac{\rho}{\sigma_v}\left(v(t+\Delta t) - v(t) - \kappa\eta\Delta t\right) \\ &+ (\frac{\kappa\rho}{\sigma_v} - \frac{1}{2})\int_t^{t+\Delta t} v(u)du + \sqrt{1-\rho^2}\int_t^{t+\Delta t}\sqrt{v(u)}dW_2(u). \end{aligned} \tag{4.8}$$

Equation (4.8) gives an exact solution of the log asset price. But we are still left with two unsolved integrals. We know the distribution of the second integral up to a parameter. Since processes W_2 and v are independent from each other, the integral on $[t, t+\Delta t]$ follows a normal distribution,

$$\int_t^{t+\Delta t}\sqrt{v(u)}dW_2(u) \sim \mathcal{N}\left(0, \int_t^{t+\Delta t} v(u)du\right). \tag{4.9}$$

The first integral is not easily solved. Broadie and Kaya [14] derived the characteristic function of this integral and then numerically Fourier-invert the characteristic function to generate the conditional cumulative distribution function for the integral. Then they can sample a value of the integral from the distribution function. Although being a clear idea, the scheme is quite complex. It takes both attention to the parameter choice and longer computation time to reduce the bias introduced by the numerical discretization in the algorithm.

For our Monte Carlo simulation, we adopt the idea proposed by Andersen [5]. We write

$$\int_t^{t+\Delta t} v(u)du \approx \Delta t[\gamma_1 v(t) + \gamma_2 v(t+\Delta t)] \tag{4.10}$$

for some constant γ_1 and γ_2. By choosing $\gamma_1 = 1, \gamma_2 = 0$, we have an Euler-like scheme for the integral; by choosing $\gamma_1 = \frac{1}{2}, \gamma_2 = \frac{1}{2}$, we approximate the integral by trapezoidal rule.

Using (4.9) and (4.10), Equation (4.8) gives the following discretization scheme for the log asset price,

$$\begin{aligned} lnS(t+\Delta t) &= lnS(t) + r\Delta t + \frac{\rho}{\sigma_v}\left(v(t+\Delta t) - v(t) - \kappa\eta\Delta t\right) \\ &+ \Delta t\left(\frac{\kappa\rho}{\sigma_v} - \frac{1}{2}\right)\left(\gamma_1 v(t) + \gamma_2 v(t+\Delta t)\right) \\ &+ \sqrt{\Delta t}\sqrt{1-\rho^2}\sqrt{\gamma_1 v(t) + \gamma_2 v(t+\Delta t)}Z, \end{aligned} \tag{4.11}$$

where $Z\sim \mathcal{N}(0,1)$.

The above scheme is a tradeoff between accuracy and simulation efficiency. The whole sampling procedure from t to $t+\Delta t$ is outlined by the following steps,

1. At time t, $v(t)$ is known. First sample $v(t+\Delta t)$ from the conditional distribution (4.5).

2. $lnS(t)$ is also known at time t, so conditioned on $v(t)$, $v(t+\Delta t)$ and $lnS(t)$ we approximate $lnS(t+\Delta t)$ with Equation (4.11). ■

4.2 Option Payoffs for Simulation

4.2.1 Simple Put Payoff

The simple put payoff function is

$$f_{put}(t,s) = \max[K - s, 0].$$

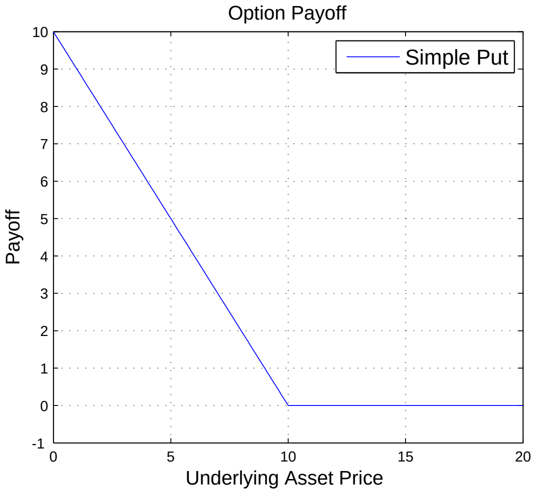


Figure 4.1: Put payoff function with strike $K = 10$.

where K is the strike. This means that the payoff at time t on an asset with price process $\{S(t), t \geq 0\}$ is given by $f(t, S(t))$. Figure 4.1 illustrates the payoff function.

The simple put payoff in the Black-Scholes world is the starting point of our investigation of the LSM method. By studying how the LSM method works on this payoff, we shall make clear a few important choices of the simulation configurations, such as which set of sample paths to select for the least square regression and why. We also want to search for a criterion in order to evaluate different set of regressors by. Besides, the pricing result for the Bermudan put will give us a sense of the order of accuracy we may achieve with the LSM. So when pricing options with more complicated features, we would have some ideas about the boundary.

The arbitrage free price of a European put satisfies the Black-Scholes PDE [10]. Suppose $\bar{U}(t, s)$ represents the value function of a European put option at time t, where $0 \leq t \leq T$. Through change of variable $\tau = T - t$, $\bar{U}(t, s)$ is changed into $U(\tau, s)$, which is the solution to the following initial value problem.

$$\begin{aligned} \frac{\partial U}{\partial \tau} &= \frac{1}{2}\sigma^2 s^2 \frac{\partial^2 U}{\partial s^2} + rs\frac{\partial U}{\partial s} - rU \qquad \text{for } 0 \leq \tau \leq T. \\ U(0, s) &= f_{put}(T, s). \end{aligned} \tag{4.12}$$

The solution $\bar{U}(t, s)$ is given by the Black-Scholes formula,

$$\begin{aligned} \bar{U}(t, s) &= sN[d_1(t, s)] - e^{-r(T-t)}KN[d_2(t, s)] \qquad \text{for } 0 \leq t < T, \\ \text{where } d_1(t, s) &= \frac{1}{\sigma\sqrt{T-t}}\left[ln\left(\frac{s}{K}\right) + \left(r + \frac{1}{2}\sigma^2\right)(T-t)\right], \\ d_2(t, s) &= d_1(t, s) - \sigma\sqrt{T-t}. \end{aligned} \tag{4.13}$$

and K is the strike, T is the time of maturity, σ is the volatility of the underlying asset price, $N(.)$ is the cumulative distribution function for the standard normal random variable $\mathcal{N}(0, 1)$, which is given by $N(y) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{y} e^{-\frac{z^2}{2}} dz$.

4.2.2 Put Spread Payoff

The put spread payoff function is

$$f_{putsprd}(t, s) = \begin{cases} Q & \text{if } s \leq K_1, \\ Q \times \frac{K_2 - s}{K_2 - K_1} & \text{if } K_1 < s < K_2, \\ 0 & \text{if } s \geq K_2. \end{cases} \tag{4.14}$$

where $0 < K_1 < K_2$ are the lower and upper strikes, and Q is the fixed maximum payoff value. Figure 4.2 illustrates the payoff function.

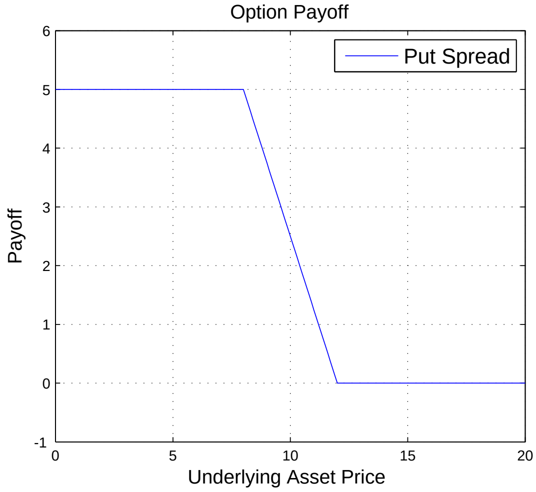


Figure 4.2: Put spread payoff function with strike $K_1 = 8$, $K_2 = 12$, $Q = 5$.

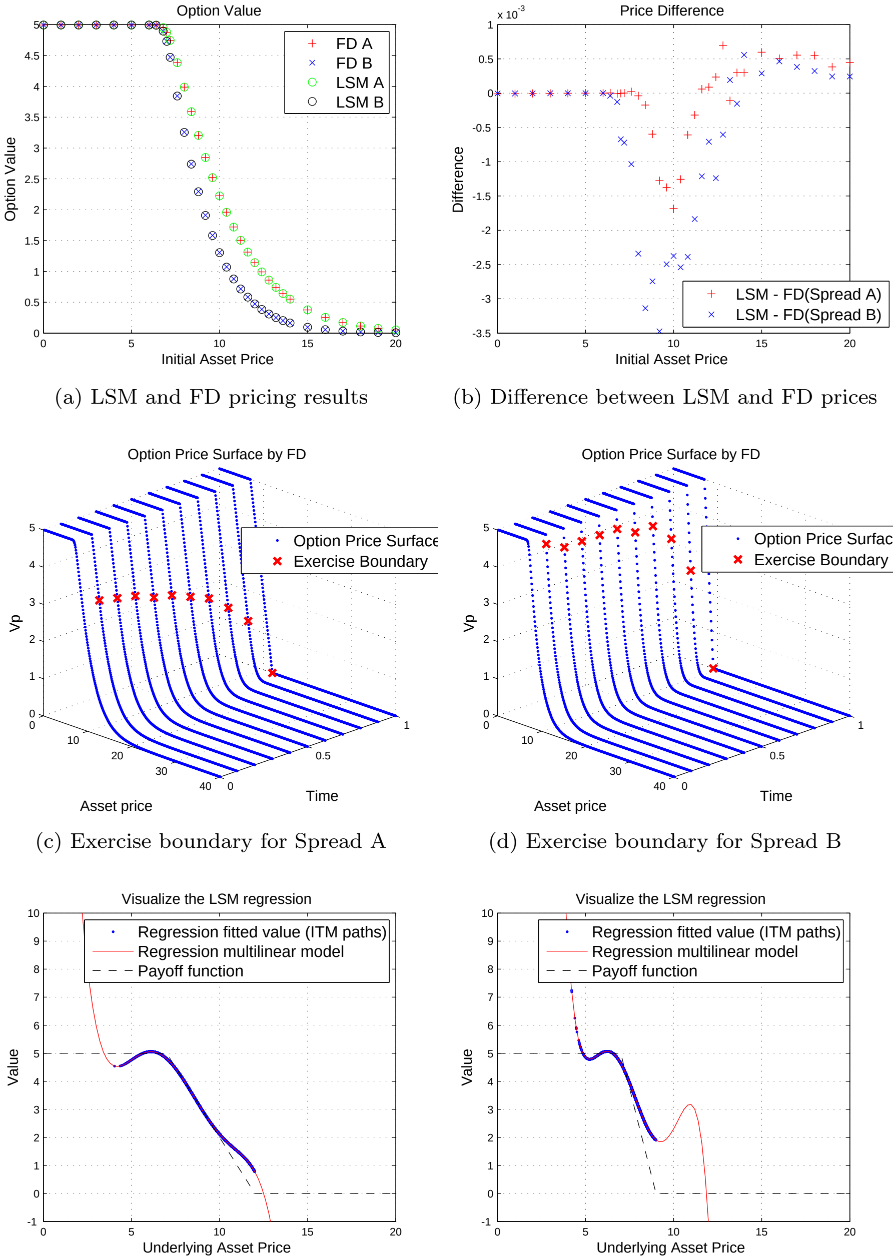
Compared to the put payoff, put spread payoff has a bend at its lower strike. This nonlinear relation w.r.t the underlying asset price will pass onto the option's continuation value function. Hence it is interesting to know how this bend, or the slope of the payoff function would affect the LSM price. Moreover, the position of the bend may be near the true early exercise boundary in the s domain. This strong nonlinearity could make it difficult for the OLS regression to fit the continuation value function. This will be used to illustrate how the error in approximating the continuation value function would affect the pricing error by the LSM method.

In Figure 4.3, we gather simulation results from both LSM and FD schemes for two Bermudan put spreads. Spread A is defined by $K_1 = 7$, $K_2 = 12$, $Q = 5$. Spread B is defined by $K_1 = 7$, $K_2 = 9$, $Q = 5$. The slope of Spread B is steeper than that of Spread A, which also makes the bend around K_1 more pronounced for Spread B. The payoff function value of Spread A is uniformly higher than or equal to that of Spread B. We price Bermudan style option on the two spread payoff function. Maturity $T = 1$, number of exercise moments $N_t = 52$.

Subplot (a) shows the simulated price by LSM and FD for both put spreads at different initial asset prices. Our LSM prices agree well with the FD benchmark. Notice that the price of Spread A is higher than that of Spread B at all initial asset prices. Looking at the pricing difference between the LSM method and the FD method in (b), we find out that the difference for Spread B is more significant than for Spread A. More specifically, the LSM price is more negatively biased as compared to the FD benchmark for Spread B than for Spread A, and the ratio of the two biases is much higher than the ratio of the absolute option prices of the two spreads. The most significant difference occurs when the initial asset price is around 10. Subplots (c) and (d) show that this is the asset price region where the early exercise boundary of the option may lie in. We only marked early exercise boundary at about 10 exercise moments during the options lifetime. We also observe that the early exercise boundary is closer to the bend around K_1 for Spread B than for Spread A. In subplots (e) and (f), we choose the observation time $0.3 \times T$ and plot the regression fitted continuation value function approximation. What appears is that more significant bend around K_1 for Spread B makes it more difficult to approximate the continuation value function by the OLS fitted multi-linear model (in red curve in the plots). We do not know the exact form of the continuation value function. However, the more irregular shape as shown by subplot (f) is not to be expected. As is discussed in Section 2.2.5, error in approximating the continuation value function translates into error in exercise decision and propagates through the backward recursion in the LSM. These errors accumulate and cause the LSM price to be lower than the true option value associated with the optimal exercise strategy. Data in Chapter 7 shows that our FD benchmark converges sufficiently well, hence we believe that price difference in (b) can represent the actual pricing error from the true option value up to the precision necessary for the above discussion.

The arbitrage free price of a European put spread also satisfies the Black-Scholes PDE. It can be solved from Equation (4.12) with initial condition replaced by $U(0, s) = f_{putsprd}(T, s)$.

The payoff function (4.14) can be expressed as the scaled difference between two put payoffs,



(a) LSM and FD pricing results

(b) Difference between LSM and FD prices

(c) Exercise boundary for Spread A

(d) Exercise boundary for Spread B

(e) Continuation value function approximation for Spread A

(f) Continuation value function approximation for Spread B

Figure 4.3: (a) and (b) show the pricing result and difference between LSM and FD for Spread A and B. (c) and (d) show the option price curve and early exercise boundary at time $t = 0$ and several exercise moments. (e) and (f) show the the regression fitted continuation value function. Model and simulation parameters are: $r = 0.06$, $\sigma = 0.3$, $T = 1$, $N_t = 52$. 10^5 paths and 100 run average for the LSM. The regressors are: c, S, $\ldots$, S^5.

$$f_{putsprd}(t,s) = \frac{Q}{K_2 - K_1}[f_{put}(t,s;K_2) - f_{put}(t,s;K_1)].$$

where $f_{put}(t,s;K)$ represents the simple put payoff with strike K. But this means that the arbitrage-free price of a European put spread can be expressed as the scaled difference between two corresponding simple puts,

$$\bar{U}^{Eu}_{putsprd}(t,s) = \frac{Q}{K_2 - K_1}[\bar{U}^{Eu}_{put}(t,s;K_2) - \bar{U}^{Eu}_{put}(t,s;K_1)]. \tag{4.15}$$

Since the European put price $\bar{U}^{Eu}_{put}(t,s;K)$ is given by the Black-Scholes formula, we obtain the analytical solution for the European put spread as well. Equation (4.15) may be used as an explanatory variable for constructing the regressors in the LSM method. Its performance is documented and analysed in the section of numerical results. On the other hand, we do not have similar equation like (4.15) for American or Bermudan option. This is because two American puts with different strikes have different optimal exercise boundaries. If they are not exercised at the same moment, the cash flows generated by the American put spreads and the two corresponding American puts will not match. Hence we do not have the equality in their arbitrage-free prices.

4.2.3 An Asian-style Payoff

The Asian-style payoff that we are going to investigate is the discretely-monitored arithmetic-mean fixed-strike Put. It is a path-dependent payoff in the sense that the payoff depends not only on the underlying's price at the observation moment, but also on its price at some earlier time. We shall study how to take care of this path-dependency in the LSM method.

Suppose $0 \leq t_1 < t_2 < \ldots \leq t_N \leq T$ are the discrete monitoring moments. The payoff at time t_n is given by,

$$f_{dm}(t_n, A) = \max[K - A(t_n), 0] \qquad \text{for } 1 \leq n \leq N. \tag{4.16}$$

where S represents the underlying asset price process $\{S(t), t \geq 0\}$, and $A(t_n) = \frac{1}{n}\sum_{i=1}^{n} S(t_i)$.

A closely related payoff is the continuously-monitored arithmetic-mean fixed-strike put, whose payoff at time t is given by,

$$f_{cm}(t, A) = \max[K - A(t), 0] \qquad \text{for } 0 \leq t \leq T. \tag{4.17}$$

where $A(t) = \frac{1}{t}\int_0^t S(\chi)d\chi$.

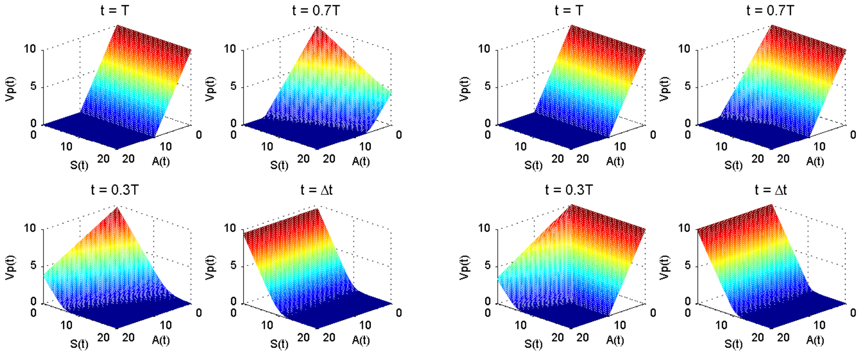
These are two different payoff functions, but as we choose more and more discrete monitoring moments. Payoff (4.16) converges to payoff (4.17). We need the continuously-monitored version in order to build a FD solver based on the PDE that its option value satisfies. The arbitrage free price is a function of time t, underlying asset price $S(t)$ and its running average $A(t) = \frac{1}{t}\int_0^t S(\chi)d\chi$. Let $U(\tau, s, a)$ be the option value at time $t = T - \tau$, with underlying price $S(T-\tau) = s$, its running average $A(T-\tau) = a$. It satisfies the following initial value problem,

$$\begin{aligned} \frac{\partial U}{\partial \tau} = \frac{1}{2}\sigma^2 s^2 \frac{\partial^2 U}{\partial s^2} + rs\frac{\partial U}{\partial s} + \frac{s-a}{T-\tau}\frac{\partial U}{\partial a} - rU, \\ U(0,s,a) = f_{cm}(T,a) \end{aligned} \tag{4.18}$$

for $0 \leq \tau \leq T$ and $\tau = T - t$, $s > 0$, $a > 0$.

We build a FD solver for the Bermudan Asian-style option based on PDE (4.18). Although the solution approximates the continuously-monitored version of the Bermudan Asian option, which differs from the discretely-monitored one, this is the closest match we have. As we will see from the numerical experiment, this allows us to make comparison between different regressor configurations in the LSM method.

The overall procedure for pricing this option with FD is similar to the one we adopt for pricing the Bermudan put. We first build a FD solver for the corresponding European Asian-style option, based on which we build the solver for the Bermudan version. The complication is that we have to deal with the partial derivative to a, $\frac{\partial U}{\partial a}$, which requires special treatment to avoid numerical problem. Unlike the case of floating-strike Asian option, it is not possible to reduce the fixed-strike version to a one dimensional problem through change of variable. The PDE solver uses up-wind scheme in the a direction. In order to improve computation efficiency, we use the Alternate Direction Implicit(ADI) scheme for time stepping, see [3] and [54] for details. The ADI scheme also allows us to improve convergence at early exercise



(a) FD price surfaces for a European-Asian option

(b) FD price surfaces for a Bermudan-Asian option

Figure 4.4: The FD price surfaces for a European continuously-monitored arithmetic-mean fixed-strike Asian-style put and its Bermudan counterpart.

moments by using PSOR to solve the matrix equation iteratively. We can take into account of early exercise decision along each s grid instead of on the whole domain all at once.

Figure 4.4 illustrates the FD approximation to the price surface of a European Asian-style continuously monitored arithmetic-mean fixed-strike put. The strong convection term in (4.18), the term $\frac{s-a}{T-\tau}\frac{\partial U}{\partial a}$, makes the price surface "rotate" above the $S \times A$ plane. We observe that there is a $T-\tau$ term appearing in the denominator part of the convection term. To avoid the explosion of that coefficient, we can only obtain finite solution to (4.18) at time $t = \Delta t$ instead of time $t = 0$. We take the option value along the line $A(\Delta t) = S(\Delta t)$ as an approximation to the initial option value $A(0)$. This approximation introduces error to the FD solution. However, we can reduce this error by using large number of time steps, such that $\Delta t \approx 0$. For calculating the Bermudan Asian option price, we choose 1000 steps in the S direction, 1000 steps in the A direction and 2000 steps in t direction. That makes $\Delta t = 5 \times 10^{-4}$. Figure 4.4 also shows the FD approximation of the price surface for a Bermudan Asian-style continuously monitored arithmetic-mean fixed-strike put. Comparing the price surfaces of the Bermudan option and its European counterpart at $t = 0.3T$, $t = 0.7T$, we observe the effect of early exercise feature. Again, at $t = \Delta t$, strong convection makes the Bermudan option price surface appear to have "rotated" for 90 degrees.

5 Two Special Regression Methods

In this chapter we study two regression methods that will be used to solve the problem of automatically selecting the most relevant set of regressors for the LSM method. These are the Principal Component Regression (PCR) and the Stepwise Regression (SR). We perform numerical simulation to check their usefulness when applied in the LSM method.

5.1 Principal Component Regression

In [24] a good outline is given for the PCR method, we follow their description. Let's begin with a multi-linear regression model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}.$$

where $\mathbf{y} = (y_1, \ldots, y_n)^T$ is the vector of response variable, $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_r)^T$ is the vector of regression coefficients, $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_n)^T$ is the vector of error terms. By assumption $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$. The regressors matrix $\mathbf{X}$ is given by,

$$\mathbf{X} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1r} \\ 1 & x_{21} & \cdots & x_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{nr} \end{bmatrix}_{n\times(r+1).}$$

The ith component in $\mathbf{y}$ is given by

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_r x_{ir} + \epsilon_i. \tag{5.1}$$

The least square estimator of $\boldsymbol{\beta}$ is,

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}.$$

One of the underlying assumptions of the Ordinary Least Square (OLS) regression is that the regressors (column vectors) in matrix $\mathbf{X}$ must be linearly independent. Mathematically this means that the regressor matrix must have full column rank. In the context of the LSM, there will be far more rows representing independent samples of the regressors than columns, so the practical constraint is on the columns. When this assumption is violated, the regressors are called multicollinear. In this situation, the regression coefficient estimates may change erratically in response to small changes in the model or data. However, multicollinearity does not reduce the reliability of the multi-linear model as a whole. In this sense, it does not pose a problem to the LSM method because the goal is to estimate the continuation value function, not to find out the response from a single regressor, see Section 2.2.3. Nevertheless, the least square estimator for the coefficients demand linearly independent regressors, since otherwise the covariance matrix $\mathbf{X}^T\mathbf{X}$ is not full rank hence not invertible.

The PCR is a method for dealing with multicollinearity. In order to introduce the method, let's first rewrite the multi-linear model (5.1) in terms of centered and scaled regressors. Define the centered and scaled regressors according to

$$u_{ij} = \frac{x_{ij} - \bar{x}_j}{s_j},$$

where $\bar{x}_j = \frac{1}{n}\sum_{i=1}^n x_{ij}$ is the sample mean and $s_j = \sqrt{\sum_{i=1}^n (x_{ij} - \bar{x}_j)^2}$ is the scaling factor. We can write the model (5.1) as

$$y_i = \beta_0^* + \beta_1^* \left(\frac{x_{i1} - \bar{x}_1}{s_1}\right) + \beta_2^* \left(\frac{x_{i2} - \bar{x}_2}{s_2}\right) + \ldots + \beta_r^* \left(\frac{x_{ir} - \bar{x}_r}{s_r}\right) + \epsilon_i.$$

Write it in the matrix form, we have

$$\mathbf{y} = \beta_0^*\mathbf{1} + \mathbf{U}\boldsymbol{\beta}^* + \boldsymbol{\epsilon}, \tag{5.2}$$

where $\boldsymbol{\beta}^* = (\beta_1^*, \beta_2^*, \ldots, \beta_r^*)^T$, $\mathbf{1}_{r\times1} = (1, 1, \ldots, 1)^T$ and $\mathbf{U}$ is given by

$$\mathbf{U} = \begin{bmatrix} u_{11} & \cdots & u_{1r} \\ u_{21} & \cdots & u_{2r} \\ \vdots & \ddots & \vdots \\ u_{n1} & \cdots & u_{nr} \end{bmatrix}_{n\times r.}$$

The relation between the regression coefficients in the original model and the centered and scaled model is,

$$\beta_j = \frac{\beta_j^*}{s_j},$$

for $j = 1, \ldots, r$ and

$$\beta_0 = \beta_0^* - \frac{\beta_1^*}{s_1} - \frac{\beta_2^*}{s_2} - \ldots - \frac{\beta_r^*}{s_r},$$

where $\beta_0^* = \bar{y}$.

The $r \times r$ sample correlation matrix of the regressors is $\mathbf{C} = \mathbf{U}^T\mathbf{U}$. Let $\lambda_1, \ldots, \lambda_r$ be the eigenvalues of $\mathbf{C}$, and $\mathbf{V} = (v_1, v_2, \ldots, v_r)$ be the $r \times r$ matrix consisting all the associated normalized eigenvectors. The eigenvalues are the solution to the equation

$$|\mathbf{U}^T\mathbf{U} - \lambda\mathbf{I}| = 0,$$

where $|\cdot|$ denotes the matrix determinant and $\mathbf{I}$ is the $r \times r$ identity matrix. Each pair of eigenvalue and eigenvector satisfies the equation

$$(\mathbf{U}^T\mathbf{U} - \lambda_j\mathbf{I})v_j = \mathbf{0},$$

for $j = 1, \ldots, r$.

Assume $\mathbf{U}$ has full column rank r. First notice that for any $r \times 1$ vector x, $\mathbf{U}^T\mathbf{U}x = 0$ if and only if $\mathbf{U}x = 0$. Hence $\mathbf{U}^T\mathbf{U}$ and $\mathbf{U}$ has the same rank. That is to say $\mathbf{C}$ is a full rank $r \times r$ matrix. Hence all of its eigenvalues are non-zero. Take any nonzero $r \times 1$ vector $x = (x_1, \ldots, x_r)^T$ in $\mathbb{R}^r$, we have

$$x^T(\mathbf{U}^T\mathbf{U})x = (\mathbf{U}x)^T\mathbf{U}x \geq 0.$$

So we conclude that all the eigenvalues are strictly positive, and $\mathbf{C}$ is positive definite. Since $\mathbf{C}$ is a real symmetric matrix, the eigenvectors can be chosen such that they are real, orthogonal to each other and have norm one, i.e. they satisfies $v_j^Tv_j = 1$ and $v_i^Tv_j = 0$ for $i \neq j$. Hence the eigenvector matrix $\mathbf{V}$ is orthonormal, i.e. $\mathbf{V}\mathbf{V}^T = \mathbf{I}$.

Model (5.2) now gives the formula for PCR,

$$\begin{aligned} \mathbf{y} &= \beta_0^*\mathbf{1} + \mathbf{U}\mathbf{V}\mathbf{V}^T\boldsymbol{\beta}^* + \boldsymbol{\epsilon} \\ &= \beta_0^*\mathbf{1} + \mathbf{Z}\boldsymbol{\alpha} + \boldsymbol{\epsilon}, \end{aligned} \tag{5.3}$$

where $\mathbf{Z} = \mathbf{U}\mathbf{V}$ is the $n \times r$ matrix of principal components (PCs), $\boldsymbol{\alpha} = \mathbf{V}^T\boldsymbol{\beta}^*$ can be interpreted as the new regression coefficients. The least square estimators for $\boldsymbol{\alpha}$ is

$$\hat{\boldsymbol{\alpha}} = (\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T(\mathbf{y} - \beta_0^*\mathbf{1}). \tag{5.4}$$

Notice that the principal components (column vectors in $\mathbf{Z}$) are orthogonal to each other,

$$\mathbf{Z}^T\mathbf{Z} = (\mathbf{U}\mathbf{V})^T(\mathbf{U}\mathbf{V}) = \mathbf{V}^T\mathbf{U}^T\mathbf{U}\mathbf{V} = \mathbf{V}^T\mathbf{C}\mathbf{V} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_r).$$

In practice, when all the eigenvalues are strictly positive according to the assumption, $\mathbf{Z}^T\mathbf{Z}$ is positive definite. Hence the least square estimator (5.4) is justified. When $\mathbf{C}$ is not full rank, then some of the eigenvalues are zero, (5.4) can not be used. In this case, one can eliminate the PCs associated with those zero eigenvalues and perform the regression on a reduced set of PCs. Let there be $r - q$ positive eigenvalues $(\lambda_1, \lambda_2, \ldots, \lambda_{r-q})$, $r \times r - q$ matrix $\mathbf{V}_{r-q} = (v_1, v_2, \ldots, v_{r-q})$. We obtain the $n \times r - q$ matrix of remaining principal components $\mathbf{Z}_{r-q} = \mathbf{U}\mathbf{V}_{r-q}$. We have $\mathbf{Z}_{r-q}^T\mathbf{Z}_{r-q} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_{r-q})$. Hence the least square estimator with $\mathbf{Z}_{r-q}$ can be used. Similarly, if some of the eigenvalues are much lower compared to others, the amount of variation in $\mathbf{y}$ that is explained by the associated eigenvectors is low. In that case, we may leave out those eigenvectors in the multi-linear model.

In our numerical simulation, we assemble a large set of regressors for the LSM and then perform PCR on those regressors. We experiment two ideas about how to select the eigenvectors or PCs to improve the LSM price accuracy. The first one is according to the amount of total variance explained by the PCs. We sort the eigenvalues (and eigenvectors) according to their values in decreasing order. We set a threshold for the amount of total variance that is explained by the final model. This number is given by trace of the matrix $\mathbf{Z}_s^T\mathbf{Z}_s$, where $\mathbf{Z}_s$ contains all the selected PCs. The eigenvectors associated with the lowest eigenvalues are eliminated from the final model. Suppose the eigenvalues $\lambda_1, \ldots, \lambda_r$ are already arranged in a decreasing order. We set the threshold to be a, often a value less than but close to 1. We will keep the eigenvectors associated with eigenvalues $\lambda_1, \dots, \lambda_k$, where k is determined by

$$k = \min_{l=1,\dots,r} \left(\frac{\sum_{i=1}^{l} \lambda_i}{\sum_{i=1}^{r} \lambda_i} \geq a \right).$$

The second idea is according to the likelihood or statistical significance of the PCs in the model. After the PCR we have a multi-linear model with all the PCs as regressors. By checking whether the coefficients are statistically nonzero, we can select the more significant PCs to form the final model through another regression. When the $p-$value of the $t-$statistic of a coefficient is higher than a chosen significance level, the associated PC will be eliminated from the final model.

5.2 Stepwise Regression

Depending on the exact procedure and statistics involved, Stepwise Regression (SR) can represent a family of regression methods. It is a systematic method for adding and removing terms from a multilinear model based on their statistical significance in a regression. We have experimented using it in the LSM method.

We use the Matlab function "stepwisefit.m". Here is a brief description of the method [1]. The model starts with an initial model and then compares the explanatory power of incrementally larger and smaller models. At each step, the $p-$value of an $F-$statistic is computed to test models with and without a potential term. If a term is currently not in the model, we have the following hypothesis test,

H_0: This term would have a zero coefficient if added to the model.

H_1: This term would have a non-zero coefficient if added to the model.

If there is enough evidence to reject H_0, i.e. the $p-$value is lower than the inclusion threshold, one should consider adding the term to the model.

On the other hand, if a term is already in the model, we have the following hypothesis test,

H_0: This term has a zero coefficient.

H_1: This term has a non-zero coefficient.

If there is not enough evidence to reject H_0, i.e. the $p-$value is higher than the removal threshold, one should consider removing the term from the model.

The whole algorithm proceeds as follows:

1. Fit the initial model (can be specified by user).

2. If any term not in the model has a $p-$value less than the inclusion threshold, add the one term with the lowest $p-$value and repeat this step; otherwise, go to step 3.

3. If any terms in the model has a $p-$value greater than the removal threshold, remove the one term with the highest $p-$value and go to step 2; otherwise, end.

■

The major problem with the method is that it may produce different multi-linear model depending on the initial model and the order that each term is added or removed. In this sense the method can produce locally optimal result, not the globally optimal one.

5.3 Numerical Results

We have tested the above two methods in both the Black-Scholes world and the Heston model. We choose a large set of regressors to price a Bermudan put in each setting. Applying the above mentioned regression methods to this large set of regressors shall give the evidence of their usefulness. We will consider the following methods: the original LSM method, plain PCR, PCR plus selecting PCs according to the total variance explained (PCR selVar), PCR plus selecting PCs according to their respective likelihood of being in a multi-linear model (PCR selLH), SR according to the Matlab function "stepwise.m" (SR routine). We choose 10^5 paths to calculate the LSM price. An average from100 such simulated prices is reported as the pricing result. The total simulation time is also reported.

5.3.1 In the Black-Scholes World

The simulation setup is the same as the one we use in Section 7.1. We fix the strike at $K = 10$ and 52 possible exercise moments for the maturity $T = 1$. We choose the first 10 powers of the underlying asset price as the set of regressors. The regressor vector at time T_n is $\big(c, S(T_n), S(T_n)^2, \dots, S(T_n)^{10}\big)$.

Table 5.1: Price difference between the LSM and FD benchmark with different regression methods in the Black-Scholes world.

Name	Initial Asset Price			Time
	8(ITM)	10(ATM)	12(OTM)	
Benchmark (FD)	2.1016	0.9517	0.3945	
LSM	-9×10^{-4}	-1.8×10^{-3}	-1.1×10^{-3}	12 min
PCR	7×10^{-4}	1.1×10^{-3}	9×10^{-4}	21 min
PCR(selVar 99.5%)	-1.75×10^{-2}	-7.5×10^{-3}	-1.8×10^{-3}	21 min
PCR(selVar 95%)	-1.13×10^{-1}	-3.79×10^{-1}	-1.57×10^{-1}	21 min
PCR(selLH)	2×10^{-4}	$< 1 \times 10^{-4}$	3×10^{-4}	27 min
SR(routine)	$< 1 \times 10^{-4}$	1×10^{-4}	2×10^{-4}	40 min

Table 5.1 shows price difference between the above mentioned regression methods and the benchmark in the Black-Scholes world. The benchmark prices are the same as those in Section 7.1. The original LSM method with OLS regression produces lower prices compared to the benchmark. Plain PCR gives higher prices compared to the benchmark at all three initial asset values. PCR with PC selection based on variance gives lower prices. The percentage of total variance explained has a big impact on the result. Since we need the price to be as accurate as possible, the variance selection PCR does not perform well in this regard. By comparison, the PCR with PC selection based on likelihood performs better. The results are close to the benchmark, although still high biased. The Matlab SR routine performs similarly to PCR(selLH), but it takes considerably longer simulation time. It is necessary to point out that the above results are simulated with a relatively simple set of regressors $c, S(T_n), S(T_n)^2, \dots, S(T_n)^{10}$. Choosing other set of regressors may result in different effectiveness for these methods.

5.3.2 In the Heston Model

The model and simulation setup is the same as the one we use in Section 7.4. We fix the initial asset price $S_0 = 10$ and 52 possible exercise moments for the maturity $T = 1$. The correlation coefficient is $\rho = -0.6$. We use in total 27 terms of regressors, which are listed in Table 5.2.

Table 5.2: The regressor set used in the Heston model for simulation of various regression methods

Type	Regressors
Constant	c
Asset price	$S, S^2, \dots, S^{10}$
Volatility	$v^{\frac{1}{2}}, v, v^{\frac{3}{2}}, \dots, v^3$
First order cross product	$Sv^{\frac{1}{2}}$
Second order cross product	$Sv, S^2v^{\frac{1}{2}}$
Third order cross product	$Sv^{\frac{3}{2}}, S^3v^{\frac{1}{2}}, S^2v$
Fourth order cross product	$Sv^2, S^4v^{\frac{1}{2}}, S^2v^{\frac{3}{2}}, S^3v$

Table 5.3: Price difference between the LSM and COS benchmark with different regression methods in the Heston model.

Name	Strike			Time
	8(OTM)	10(ATM)	12(ITM)	
Benchmark (COS)	0.3715	1.1038	2.3486	
LSM	6×10^{-4}	1.1×10^{-3}	-1.9×10^{-3}	36 min
PCR	2.7×10^{-3}	3.4×10^{-3}	3.5×10^{-3}	60 min
PCR(selVar 99.5%)	-7×10^{-3}	-2.63×10^{-2}	-7.7×10^{-4}	60 min
PCR(selVar 95%)	-1.05×10^{-1}	-2.75×10^{-1}	-3.62×10^{-1}	60 min
PCR(selLH)	2.3×10^{-3}	3.2×10^{-3}	3.8×10^{-3}	60 min
SR(routine)	4×10^{-4}	3×10^{-4}	-7×10^{-4}	126 min

Table 5.3 contains pricing result with the regression methods in the Heston model. The original LSM with OLS regression does not produce results consistently lower or higher than the COS benchmark. OTM and ATM results are lower, ITM result is higher. Plain PCR gives considerably higher price. PCR with PC selection based on variance criterion gives lower prices. PCR with PC selection based on likelihood does not differ much as compared to plain PCR. The SR routine produces result that lies between the original LSM result and the COS benchmark although still high biased. It takes about 3 times longer to compute than the original LSM. We need to emphasize that this is the result with the set of regressors in Table 5.2. Further investigation is necessary to evaluate the effectiveness of these regression methods in the LSM.

6 The Libor Market Model and Libor Exotics

6.1 Interest Rate Market Products

6.1.1 Bond

There are various types and definitions of Bonds. For our modeling purpose, we consider the most simple form, the zero coupon bond. It is a contract that guarantees the holder to receive 1 unit of currency at maturity T. At time $t < T$ the price of the zero coupon $T-$Bond is $p(t, T)$. It represents the time t value of 1 unit of currency that is to be received at time T. By definition, $p(T, T) = 1$.

The spot rate at time t on the interval $[t, T]$ can be derived from $p(t, T)$. Suppose we invest 1 unit of currency at time t in $T-$Bond, the payoff is $1/p(t, T)$ at time T. Suppose instead we invest 1 unit of currency from time t till T in a risk-free savings account at the simple spot rate $R(t; t, T)$. The time T payoff is $1 + (T - t)R(t; t, T)$. Standard non-arbitrage argument demands that the two payoffs equal. So,

$$R(t; t, T) = \frac{1 - p(t, T)}{(T - t)p(t, T)}.$$

6.1.2 Forward Rate Agreement

A Forward Rate Agreement (FRA) is an interest rate contract that specifies a simple interest rate R on a specified principal N over some future period $[S, T]$. At time S, the borrower receives N from the lender. At time T, he pays back the principal plus interest, $N[1 + (T - S)R]$.

Suppose there exists a liquid bond market for both maturity S and T. The value of the contract for the lender at time $t < S$ is,

$$\text{FRA}(t) = N[p(t, T)(1 + (T - S)R) - p(t, S)].$$

Market quotes are available for the Forward Rate Agreement for standard time intervals according to market definition. The forward rate R is then defined as the rate which makes the value of FRA equals to zero. At time $t < S$, this means

$$R(t; S, T) = \frac{p(t, S) - p(t, T)}{(T - S)p(t, T)}. \tag{6.1}$$

6.1.3 Forward Libor Rate

Let's first define the tenor structure which is consistent with the definition in [13]. Divide the time horizon $[0, T]$ by the sequence of time points,

$$0 = T_{-1} < T_0 < T_1 < \ldots < T_N = T.$$

We define the tenor distance $\alpha_i = T_i - T_{i-1}$, for each $i = 0, \ldots, N$. These can be equal distance or not. In our experiment, we keep equal distance for simplicity. The forward Libor Rate $L_i(t) = L(t, T_{i-1}, T_i)$ is the simple forward rate for the period $[T_{i-1}, T_i]$ contracted at time t for each $i = 1, 2, \ldots, N$. $L_i(t)$ moves according to the bond market up to time T_{i-1}. Afterwards, $L_i(t)$ freezes at $L_i(T_{i-1})$ in the period $(T_{i-1}, T_i]$ and it stops to exist after time T_i. The simple rate on the interval $[T_{-1}, T_0]$ is the spot T_0 rate which fixes at time $T_{-1} = 0$. We call it $L_0(t)$, $L_0(t) = L_0(0)$ for $t \in [T_{-1}, T_0]$.

According to the definition of forward rate in (6.1), $L_i(t)$ is given by,

$$L_i(t) = \frac{p(t, T_{i-1}) - p(t, T_i)}{(T_i - T_{i-1})p(t, T_i)}, \qquad \text{for } t \in [0, T_{i-1}].$$

6.1.4 Cap and Floor

Using the same tenor structure as before, let us define cap and floor. First, a T_i caplet with caplet rate R is a contract that pays its holder the amount

$$X_i^{capl} = \alpha_i \max\left[L_i(T_{i-1}) - R, 0\right]$$

at time T_i. A cap with cap rate R is a contract that gives the holder the amount X_i^{capl} at each T_i for $i = 1, \ldots, N$.

Similarly, a T_i floorlet with floorlet rate R is a contract that pays its holder the amount

$$X_i^{flrl} = \alpha_i \max\left[R - L_i(T_{i-1}), 0\right]$$

at time T_i. A floor with floor rate R is a contract that gives the holder the amount X_i^{flrl} at each T_i for $i = 1, \dots, N$.

One can see a caplet/floorlet as a call/put option on the underlying spot rate, the cap/floor is then a portfolio of caplets/floorlets.

6.1.5 Swap and European Swaption

An interest rate swap is a contract which exchanges one set of rate payments for another set of rate payments. The simplest of which is the fixed-for-floating swap. The terminology always refers to the fixed leg. Holder of a receiver swap will pay floating rate and receive fixed rate, holder of a payer swap will pay fixed rate and receive floating rate. For a $T_n \times (T_N - T_n)$ swap, the exchange of cash flows takes place at time $T_{n+1}, \dots, T_N$.

For a $T_n \times (T_N - T_n)$ payer swap with swap rate K, at each T_i for $i = n+1, \dots, N$, the payer receives the floating leg $\alpha_i L_i(T_{i-1})$ and pays the fixed leg $\alpha_i K$. Then at time $t < T_n$, total value of the floating leg payments in time t money is [9],

$$\begin{aligned}
\text{Value floating leg} &= \sum_{i=n+1}^{N} p(t, T_i)\alpha_i L_i(t) \\
&= \sum_{i=n+1}^{N} p(t, T_{i-1}) - p(t, T_i) \\
&= p(t, T_n) - p(t, T_N) \\
&\equiv D_n^N(t).
\end{aligned} \tag{6.2}$$

Total value of the fixed leg payments up to a scale factor K in time t money is,

$$\begin{aligned}
\text{Value fixed leg} &= \sum_{i=n+1}^{N} p(t, T_i)\alpha_i \\
&= \sum_{i=n+1}^{N} \alpha_i p(t, T_i) \\
&\equiv A_n^N(t).
\end{aligned} \tag{6.3}$$

A_n^N is often called the annuity.

The time t value of the a $T_n \times (T_N - T_n)$ payer swap with swap rate K is then

$$PS_n^N(t; K) = p(t, T_n) - p(t, T_N) - K \sum_{i=n+1}^{N} \alpha_i p(t, T_i) \tag{6.4}$$

The forward swap rate $R_n^N(t)$ of the $T_n \times (T_N - T_n)$ swap is the value of K for which $PS_n^N(t; K) = 0$. So,

$$\begin{aligned}
R_n^N(t) &= \frac{p(t, T_n) - p(t, T_N)}{\sum_{i=n+1}^{N} \alpha_i p(t, T_i)} \\
&= \frac{D_n^N(t)}{A_n^N(t)}.
\end{aligned} \tag{6.5}$$

Then time t value of a $T_n \times (T_N - T_n)$ payer swap with swap rate K can be express as

$$\begin{aligned}
PS_n^N(t; K) &= D_n^N(t) - K A_n^N(t) \\
&= (R_n^N(t) - K) A_n^N(t).
\end{aligned} \tag{6.6}$$

A $T_n \times (T_N - T_n)$ European payer swaption with strike K is a contract with exercise date T_n that at exercise date gives the holder the right but not the obligation to enter into a $T_n \times (T_N - T_n)$ payer swap with fixed swap rate K. Thus the T_n payoff of the payer swaption in time t money is

$$\begin{aligned} X_n^N &= \max[PS_n^N(t;K),0] \\ &= \max[R_n^N(t)-K,0]A_n^N(t). \end{aligned} \tag{6.7}$$

We see that by writing the swap payoff in terms of the accrual factor A_n^N and the forward swap rate R_n^N [38], the payoff of the European payer swaption can be viewed as a call option on $R_n^N(t)$ with strike K multiplied by $A_n^N(t)$.

6.1.6 Bermudan-style Libor Exotics

A Callable Libor Exotic (CLE) is typically defined to be a Bermudan-style option to enter an exotic swap [7]. By exotic, we mean that the coupons of the swap are determined by function of the underlying interest rate, in this case the forward Libor rates. On the other hand, a Cancellable Libor Exotic (CnLE) is a Bermudan-style option to cancel the remaining cash flows of an exotic swap based on Libor rates. Exotic swaps based on a broader category of interest rates are also actively traded in the market, such as Constant Maturity Swaps (CMS) and Spread-based Swaps. We will focus on Libor based exotic swaps with callable or cancellable features in this thesis. Here are some of the common coupon rate functions. Variable x takes the value of the associated Libor rate at time T_{n-1}, $L_n(T_{n-1})$.

Standard swap:

$$C(x) = k,$$

where k is the constant coupon rate.

Capped and floored floaters:

$$C(x) = \max(\min(g \times x - s, c), f)$$

where s is the strike, g is the gearing factor, c is the cap, f is the floor.

Capped and floored inverse floaters:

$$C(x) = \max(\min(s - g \times x, c), f)$$

where all the parameters are defined as above

Digitals:

$$C(x) = k \times 1_{\{x>s\}}$$

where s is the strike.

In a Snowball swap, the coupon not only depends on the Libor rate $L_n(T_{n-1})$, but also the last coupon payment. The most common snowball is of the inverse floating type. In particular, the coupon paid at time T_n is defined by

$$C_n = \max(C_{n-1} + s_n - g_n \times L_n(T_{n-1}), 0) \tag{6.8}$$

where s_n and g_n are the spread and the gearing factor at time T_n.

6.2 The Lognormal Libor Market Model

6.2.1 Market Practice and Market Model

The fixed income market has long used the Black-Scholes formula to price caplets. It is assumed that each forward Libor rate follows a lognormal distribution. For a caplet with time T_i payoff $X_i = \alpha_i \max[L_i(T_{i-1}) - R, 0]$, the Black-Scholes formula for the caplet used by the market is [9],

$$Capl_i^{BS}(t) = \alpha_i p(t,T_i)\left[L_i(t)\Phi(d_1) - R\Phi(d_2)\right], \qquad i = 1,\dots,N \tag{6.9}$$

where

$$\begin{aligned} d_1 &= \frac{1}{\sigma_i^{mkt}\sqrt{T_{i-1}-t}}\left[ln\left(\frac{L_i(t)}{R}\right) + \frac{1}{2}(\sigma_i^{mkt})^2(T_{i-1}-t)\right], \\ d_2 &= d_1 - \sigma_i^{mkt}\sqrt{T_{i-1}-t}. \end{aligned}$$

The volatility term σ_i^{mkt} is the market implied caplet Black volatility. One of its definitions is the volatility that makes $Capl_i^{BS}(0)$ equal to the market caplet price quote $Capl_i^{mkt}(0)$. Actually, the caplet prices are quoted in terms of Black volatility in the market.

More complex fixed income products, such as various kind of swaptions, usually involve cash flows at different times. Valuation of these products can not be carried out in the same way as a cap, because the cash flows can not be valued separately. The model should provide an arbitrage-free dynamics for all the Libor rates. Hence there was a need for a model that on the one hand produces the caplet pricing formula similar to the market conventional Black-Scholes formula, while on the other hand is able to provide arbitrage-free dynamics for pricing complex interest rate derivatives. The Libor Market Model (LMM), also known as the BGM model [12] was the solution. Jamshidian also contributed greatly to the model and the swap market model [38].

The underlying assumptions [9] of the LMM include an arbitrage free bond market for each tenor maturity $T_0, \dots, T_N$; a deterministic vector function of time $\boldsymbol{\sigma}_i(t)$ for each $i = 1, \dots, N$; a set of initial nonnegative forward Libor rates, $L_1(0), \dots, L_N(0)$ and a k-dimensional $\mathbb{Q}^N$ Brownian Motion $\mathbf{W}^N$. For each $i = 1, \dots, N$, we define $\mathbf{W}^i$ as the k-dimensional $\mathbb{Q}^i$ Brownian Motion generated by $\mathbf{W}^N$ according to the Girsanov theorem.

The Libor forward rates are assumed to have the following dynamics,

$$dL_i(t) = L_i(t)\boldsymbol{\sigma}_i(t)d\mathbf{W}^i(t), \qquad i = 1, \dots, N.(6.10)$$

We will omit the model derivation in this thesis, interesting readers may turn to [9] or [13] for excellent reference. We observe that $L_i(T)/L_i(t)$ is a GBM. Solving Equation (6.10), we have

$$ln\left(\frac{L_i(T)}{L_i(t)}\right) = -\frac{1}{2}\int_t^T ||\boldsymbol{\sigma}_i(u)||^2 du + \int_t^T \boldsymbol{\sigma}_i(u)d\mathbf{W}^i(u)$$

So

$$ln\left(\frac{L_i(T)}{L_i(t)}\right) \sim \mathcal{N}\left(m_i(t,T), \Sigma_i^2(t,T)\right).$$

where

$$\begin{aligned} m_i(t,T) &= -\frac{1}{2}\int_t^T ||\boldsymbol{\sigma}_i(u)||^2 du, \\ \Sigma_i^2(t,T) &= \int_t^T ||\boldsymbol{\sigma}_i(u)||^2 du. \end{aligned} \tag{6.11}$$

Following a similar procedure as the one for deriving the single asset Black-Scholes formula [13], one can derive the pricing formula for caplet under the LMM,

$$Capl_i^{Lib}(t) = \alpha_i p(t,T_i)\left[L_i(t)\Phi(d_1) - R\Phi(d_2)\right], \qquad i = 1, \dots, N \tag{6.12}$$

where

$$\begin{aligned} d_1 &= \frac{1}{\Sigma_i(t,T_{i-1})}\left[ln\left(\frac{L_i(t)}{R}\right) + \frac{1}{2}\Sigma_i^2(t,T_{i-1})\right], \\ d_2 &= d_1 - \Sigma_i(t,T_{i-1}). \end{aligned}$$

Equation (6.12) has a similar form to the market formula (6.9). In order to make the two consistent, one should have

$$\sigma_i^{mkt}\sqrt{T_{i-1} - t} = \Sigma_i(t,T_{i-1}). \tag{6.13}$$

where $\Sigma_i(t,T_{i-1})$ is given by Equation (6.11). This allows calibrating the model to market caplet prices.

An alternative formulation of the model assumes a scalar $\mathbb{Q}^N$ Brownian Motion for each Libor rate $L_i(t)$ [9]. These BMs are correlated,

$$dW_i^N(t)dW_j^N(t) = \rho_{ij}(t)dt, \qquad \text{for } i,j = 1, \dots, N. \tag{6.14}$$

The Girsanov theorem is used to change these to $\mathbb{Q}^i$ Brownian Motions. The Libor forward rates are assumed to have the dynamics,

$$dL_i(t) = L_i(t)\sigma_i(t)dW^i(t), \qquad i = 1, \dots, N. \tag{6.15}$$

where for each $i = 1, \dots, N$, $\sigma_i(t)$ is a scalar instantaneous volatility function. Solving SDE (6.15), we have

$$ln\left(\frac{L_i(T)}{L_i(t)}\right) \sim \mathcal{N}\left(m_i(t,T), \Sigma_i^2(t,T)\right).$$

where

$$m_i(t,T) = -\frac{1}{2}\int_t^T \sigma_i^2(u)du,$$
$$\Sigma_i^2(t,T) = \int_t^T \sigma_i^2(u)du. \tag{6.16}$$

Now we have pricing formula (6.12) with the integrated volatility function given by (6.16). By specifying a parametric form of the function $\sigma_i(t)$, the model can be calibrated to the caplet market data. This model form is sometimes easier to implement in computer simulation.

Recall from Equation (6.7) that the payoff of a payer swaption is formally a call option multiplied by the annuity. Similar to the market practice with caplet, in the swap market, a European swaption is quoted by its Black implied volatility. The Black-Scholes formula for a T$_n$ × $(T_N - T_n)$ European payer swaption with strike K is,

$$PSN_n^N(t) = A_n^N(t)\left[R_n^N(t)\Phi(d_1) - K\Phi(d_2)\right] \tag{6.17}$$

where

$$d_1 = \frac{1}{\sigma_{n,N}^{mkt}\sqrt{T_n - t}}\left[ln\left(\frac{R_n^N(t)}{K}\right) + \frac{1}{2}(\sigma_{n,N}^{mkt})^2(T_n - t)\right],$$
$$d_2 = d_1 - \sigma_{n,N}^{mkt}\sqrt{T_n - t}.$$

The volatility term $\sigma_{n,N}^{mkt}$ is the market implied Black volatility for the swaption.

Just as the Libor market model can produce market consistent pricing formula for the caplets, the swap market model [38] can produce pricing formula for European swaptions consistent with the above Black-Scholes type formula. As mentioned by Piterbarg [45], the Libor market model is more commonly used than the Swap market model mainly because it is less straight forward to calibrate the Swap market model to data from both markets. We will investigate the problem of pricing CLE and CnLE in the Libor market model.

6.2.2 Dynamics under Various Measures

For derivation of the arbitrage-free dynamics of the Libor rates under measure $\mathbb{Q}^j$, $j = 1, \ldots, N$, one can turn to [9] or [13] for reference. We only state the result here. $L_i(t)$ dynamics under the EMM $\mathbb{Q}^j$ (with T_j Bond as numeraire) is given by,

$$dL_i(t) = L_i(t)\mu_i^j(\mathbf{L}, t)dt + \sigma_i(t)L_i(t)dW_i^j(t),$$

for $t \leq \min(T_{i-1}, T_j)$. W_i^j is a scalar $\mathbb{Q}^j$ Brownian Motion. $\mathbf{L}$ represents the vector of all (living) Libor rates at time t. The drift term $\mu_i^j(\mathbf{L}, t)$ is given by,

$$\mu_i^j(\mathbf{L}, t) = \begin{cases} \sum_{k=j+1}^{i} \frac{\alpha_k L_k(t)}{1+\alpha_k L_k(t)}\sigma_i(t)\sigma_k(t)\rho_{ik}(t), & \text{if } i > j \\ 0, & \text{if } i = j \\ -\sum_{k=j+1}^{i} \frac{\alpha_k L_k(t)}{1+\alpha_k L_k(t)}\sigma_i(t)\sigma_k(t)\rho_{ik}(t), & \text{if } i < j \end{cases} \tag{6.18}$$

and the $\mathbb{Q}^j$ Brownian Motions are correlated by the instantaneous correlation function $\rho_{ik}(t)$,for $i, k = m(t)+1, \ldots, N$.

$$dW_i^j(t)dW_k^j(t) = \rho_{ik}(t)dt.$$

Since the Libor rates are defined on discrete tenors, the continuously rebalanced risk-neutral bank account $B(t)$ associated to the short rate $r(t)$ is not readily available here. However it is possible to define a risk neutral bank account that is rebalanced in a discrete fashion [38]. Define the next rate reset moment, $m(t) = \min(i, T_i \geq t)$, for $t > 0$. So $T_{m(t)-1} < t \leq T_{m(t)}$. The discretely rebalanced bank-account $B^d(t)$ is defined as,

$$\begin{cases} B^d(0) = 1, \\ B^d(t) = \frac{p(t, T_{m(t)})}{\prod_{j=1}^{m(t)} p(T_{j-1}, T_j)} & \text{for } 0 < t \leq T. \end{cases}$$

It is worth mentioning that the "living" Libor rates $L_i(t)$ at time t are those with $i = m(t)+1, m(t)+2, \ldots, N$. Take process B^d as the numeraire, we have the EMM $\mathbb{Q}^d$, which is named the spot Libor measure. By applying the Girsanov theorem, one can derive $L_i(t)$ dynamics under $\mathbb{Q}^d$.

We realize that in the market, the Libor rate $L_i(t)$ freezes after time T_{i-1}. So we only care about the dynamics of $L_i(t)$ for $0 < t \leq T_{i-1}$. If we fix a moment t, we are sure that $m(t) < i$. Choose $j = m(t)$ in (6.18), we have the drift term of $L_i(t)$ under measure $\mathbb{Q}^{m(t)}$,

$$\mu_i^{m(t)}(\mathbf{L}, t) = \sum_{k=m(t)+1}^{i} \frac{\alpha_k L_k(t)}{1+\alpha_k L_k(t)} \sigma_i(t)\sigma_k(t)\rho_{ik}(t). \tag{6.19}$$

Next we try to find the Girsanov kernel that allows converting a $\mathbb{Q}^{m(t)}$ Brownian Motion to a $\mathbb{Q}^d$ Brownian Motion. According to the Change-of-Numeraire technique proposed by Geman et al. [27], the associated likely ratio process should be

$$\begin{aligned}
\eta^d_{m(t)}(t) &= \frac{P(0,T_{m(t)})}{B^d(0)} \frac{B^d(t)}{P(t,T_{m(t)})} \\
&= \frac{P(0,T_{m(t)})}{B^d(0)} \frac{P(t,T_{m(t)})}{\prod_{k=0}^{m(t)} P(T_{k-1},T_k)} \frac{1}{P(t,T_{m(t)})} \\
&= \frac{P(0,T_{m(t)})}{B^d(0)} \frac{1}{\prod_{k=0}^{m(t)} P(T_{k-1},T_k)}
\end{aligned}$$

Notice that in this final expression, all the quantities are already known before time t, hence $\eta^d_{m(t)}(t)$ is a constant. But this means that the Girsanov kernel is zero if we try to write the likelihood ratio process as an Itô integral. As a result, a $\mathbb{Q}^d$ Brownian Motion has the same drift as a $\mathbb{Q}^{m(t)}$ Brownian Motion, which means that the drift of $L_i(t)$ under $\mathbb{Q}^d$ is the same as the drift under $\mathbb{Q}^{m(t)}$ given by Equation (6.19). Now we have the Libor rate dynamics under $\mathbb{Q}^d$,

$$dL_i(t) = L_i(t)\mu_i^d(\mathbf{L}, t)dt + \sigma_i(t)L_i(t)dW_i^d(t), \tag{6.20}$$

where W_i^d is a scalar $\mathbb{Q}^d$ Brownian Motion. The drift term $\mu_i^d(\mathbf{L}, t)$ is given by,

$$\mu_i^d(\mathbf{L}, t) = \sum_{k=m(t)+1}^{i} \frac{\alpha_k L_k(t)}{1+\alpha_k L_k(t)} \sigma_i(t)\sigma_k(t)\rho_{ik}(t). \tag{6.21}$$

and the $\mathbb{Q}^d$ Brownian Motions are correlated by the instantaneous correlation function $\rho_{ik}(t)$, for $i, k = m(t)+1, \ldots, N$.

$$dW_i^d(t)dW_k^d(t) = \rho_{ik}(t)dt. \tag{6.22}$$

6.2.3 Some Choices of Model Implementation

Before discussing how to build a Monte Carlo engine for the Libor market model and how to price CLE with it, we need to have a look at some details in the model. These are related to the calibration procedure.

The goal of the Libor market model is to develop, under certain assumptions, a rigid mathematical model such that the pricing result can be given in a form consistent with the Black-Scholes style market caplet pricing formula. The assumption of deterministic volatility functions plays a central role to this end. As is shown in Formula (6.12), the price of a caplet depends on the volatility function of each Libor, not the correlation between them. To price other derivatives such as a swaption, the correlation structure needs to be taken into consideration.

Although it is shown that the Libor market model and the swap market model are not consistent, under the lognormal Libor market model, distribution of the forward swap rate is not far from lognormality [37]. It is possible to come up with pretty close lognormal approximation for the swap rate under the Libor market model. This allows efficient simultaneous calibration of the Libor market model to both the cap/floor market data and the swap market data.

We are going to adopt some parametric form of the deterministic instantaneous volatility and correlation functions. To check the correctness of our Monte Carlo engine implementation, we will perform a simple calibration to the market caplet prices. For the correlation structure, we will not perform the calibration, but instead take some parametric form from literature. The chosen functional form will generate a correlation structure that allows us to conduct a study on pricing swaptions.

Various instantaneous volatility functions have been proposed, one can turn to [13] for a review. An important issue here is how the term structure of the volatility evolves over time. The Libor market model does not specify this with deterministic volatility function. One has to impose his own belief by choosing a specific form. This belief can be based on information obtained from historical data, but various empirical studies show that historical volatility is often a poor estimate for the volatility in the future. A more reasonable choice is to assume time homogeneity of the instantaneous volatility function and use today's volatility term structure as an estimation for that in the future. This is achieved by assuming that the volatility is a function of the time to maturity, $T - t$. We choose the following form proposed by Rebonato [46],

$$\sigma_i(t) = \Phi_i(t)\left([a(T_{i-1} - t) + d]e^{-b(T_{i-1}-t)} + c\right) \tag{6.23}$$

where a, b, c, d and Φ_i are constant parameters to be determined by calibration. The procedure of calibration to the caplet data is the following. From the market caplet price formula (6.9) and the Libor market model caplet pricing formula (6.16), we should have

$$\sigma_i^{mkt}(0) = \sqrt{\frac{1}{T_{i-1}}\int_0^{T_{i-1}} \sigma_i^2(u)du}$$

for each $i = 1, \ldots, N$. The integral can be evaluated analytically according to (6.23). Then we perform a nonlinear fit of the model to market implied caplet volatilities $\sigma^{mkt}(0) = [\sigma_1^{mkt}(0), \ldots, \sigma_N^{mkt}(0)]$. We can use least square fit for this. The Matlab function nlinfit.m uses the Levenberg-Marquardt algorithm instead. The parametric form (6.23) allows an exact fit to the market volatilities. We first perform the nonlinear fit by the expression $\frac{\sigma_i(t)}{\Phi_i}$ instead of $\sigma_i(t)$. This shall give us a rough fit of the model and fix the parameters a, b, c, d. Then parameter Φ_i is used to get an exact fit to each $\sigma_i^{mkt}(0)$. Figure 6.1 shows the calibration result. The market data from [36] gives all the ATM caplet volatilities of quarterly tenor spacing up to 20 years out.

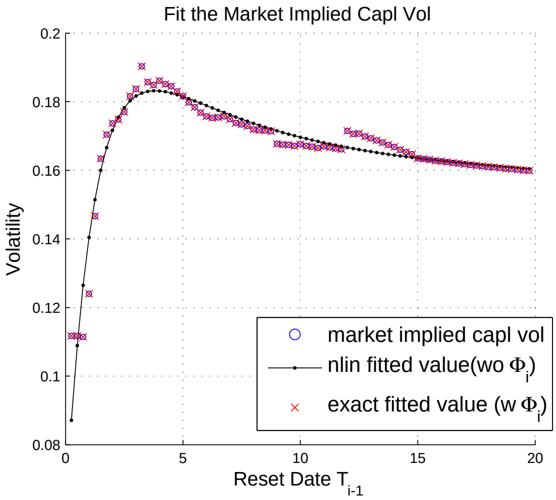


Figure 6.1: The market implied caplet volatility and the calibration of the instantaneous volatility function. The data used is taken from [36].

For the instantaneous correlation function $\rho_{ij}(t)$, we use the following parametric form from [47],

$$\rho_{ij} = \alpha + (1 - \alpha)e^{-\beta|T_i - T_j|}. \tag{6.24}$$

The parameters are chosen to be $\alpha = 0.1, \beta = 0.1$, which give the correlation matrix shown in Figure 6.2. Brigo and Mercurio mentioned in [13] that (6.24) as well as several other correlation parameterizations can automatically produce full rank correlation matrix.

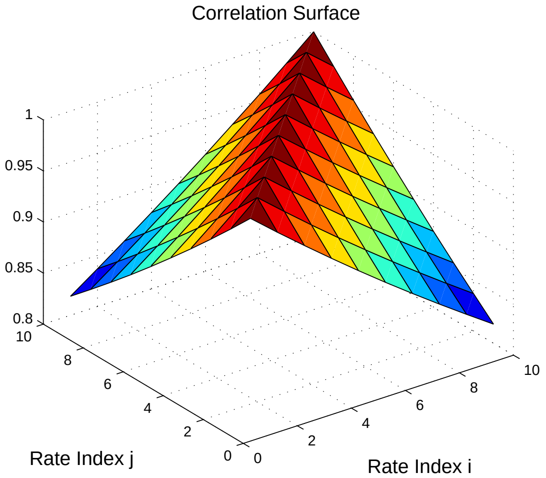


Figure 6.2: The volatility surface generated according to (6.24) for $\alpha = 0.1, \beta = 0.1$.

6.2.4 Discretization for Monte Carlo Simulation

Let $f(x) = ln(x)$. By applying Itô's lemma to $f(L_i(u))$, we have

$$dlnL_i(u) = \left(\mu_i^d(\mathbf{L}, u) - \frac{1}{2}\sigma_i^2(u)\right) du + \sigma_i(u)dW_i^d(u),$$

for $u \leq T_{i-1}$. Integrate it on $[s,t] \in [T_{m(t)-1}, T_{m(t)}]$, we have

$$lnL_i(t) = lnL_i(s) + \underbrace{\int_s^t \mu_i^d(\mathbf{L}, u)du}_{X_i} + \underbrace{\int_s^t -\frac{1}{2}\sigma_i^2(u)du}_{Y_i} + \underbrace{\int_s^t \sigma_i(u)dW_i^d(u)}_{Z_i}. \tag{6.25}$$

for each $i = m(t)+1, \ldots, N$.

Among the three integrals, only Y_i is deterministic. It can be calculated before generating the sample paths.

Write Z_i for all the living Libor rates $L_i(t)$ in vector form, we have $Z = (Z_{m(t)+1}, Z_{m(t)+2}, \ldots, Z_N)^T$. Since $\sigma_i(t)$ are deterministic, one can prove by Itô isometry that the stochastic integral is normal, $Z_i \sim \mathcal{N}(0, \int_s^t \sigma_i^2(u)du)$ and the covariance between two integrals Z_i and Z_k is $\mathbb{E}(Z_iZ_k) = \int_s^t \sigma_i(u)\sigma_k(u)\rho_{ik}(u)du$. This means Z is a multivariate normal vector with zero mean. Let C_Z denote the covariance matrix of Z, it is a real symmetric matrix. Suppose C_Z is also positive definite, it has an orthonormal basis of eigenvectors $(v_1, \ldots, v_{N-m(t)})$ with nonnegative eigenvalues $(\lambda_1, \ldots, \lambda_{N-m(t)})$. C_Z is diagonalizable by the orthogonal matrix $O = (v_1, \ldots, v_{N-m(t)})$,

$$C_Z = ODO^T,$$

where $D = \text{diag}(\lambda_1, \ldots, \lambda_{N-m(t)})$.

If we take the positive root of the eigenvalues, then $D = D^{\frac{1}{2}}D^{\frac{1}{2}}$, where $D^{\frac{1}{2}} = \text{diag}(\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_{N-m(t)}})$. We can write

$$C_Z = OD^{\frac{1}{2}}D^{\frac{1}{2}}O^T = OD^{\frac{1}{2}}O^T(OD^{\frac{1}{2}}O^T)^T \equiv Q_ZQ_Z^T.$$

Q_Z is the square root of matrix C_Z. Let $U = (U_1, \ldots, U_{N-m(t)})^T$, with each element independent standard normals, $U_i \sim \mathcal{N}(0,1)$ and $\mathbb{E}(U_iU_j) = 0$, for $i \neq j$. The covariance matrix of Q_ZU is

$$\text{Cov}(Q_ZU) = Q_Z\text{Cov}(U)Q = Q_ZIQ_Z^T = C_Z.$$

where I is an $N - m(t) \times N - m(t)$ identity matrix. So multivariate normal random vector Q_ZU and Z have the same joint distribution. We can sample vector U and then calculate Q_ZU to get a sample for Z.

We do not necessarily need the square root matrix Q_Z to sample Z. Instead, we can calculate the pseudo square root for matrix C_Z, which can be realized by Cholesky decomposition $C_Z = G_ZG_Z^T$, where G_Z is a lower triangular matrix. G_ZU and Z also share the same joint distribution.

The integral X_i depends on the state of Libor rates, more specifically, Equation (6.21) shows that under the spot Libor measure $\mathbb{Q}^d$, X_i (through $\mu_i^d(\mathbf{L},t)$) depends on the value of L_k for $k = m(t)+1,\ldots,i$ on the time interval $[s,t]$. For efficient Monte Carlo simulation, this integral has to be approximated. Let's write the integral again for ease of reference,

$$X_i = \int_s^t \sum_{k=m(t)+1}^{i} \frac{\alpha_k L_k(u)}{1+\alpha_k L_k(u)} \sigma_i(u)\sigma_k(u)\rho_{i,k}(u)du.$$

We present two alternatives for the approximation. The first one is the so called log-Euler scheme, which takes the initial value on the interval, $L_k(s)$, as an approximation to the stochastic $L_k(u)$.

$$X_i \approx \sum_{k=m(t)+1}^{i} \frac{\alpha_k L_k(s)}{1+\alpha_k L_k(s)} \int_s^t \sigma_i(u)\sigma_k(u)\rho_{i,k}(u)du \tag{6.26}$$

The second choice is the predictor–corrector (PC) scheme by [36], which consists of 2 steps. In the predictor step, an estimate of X_i, named X_i^E is calculated with the log-Euler scheme (6.26). Estimate of the Libor rates, $L_i^E(t)$, are calculated by Equation (6.25) with X_i^E. These estimates are then used in the corrector step,

$$X_i \approx \sum_{k=m(t)+1}^{i} \frac{1}{2}\left[\frac{\alpha_k L_k(s)}{1+\alpha_k L_k(s)} + \frac{\alpha_k L_k^E(t)}{1+\alpha_k L_k^E(t)}\right] \int_s^t \sigma_i(u)\sigma_k(u)\rho_{i,k}(u)du \tag{6.27}$$

This approximation uses a trapezoidal rule like scheme to approximate the integral of the part $\frac{\alpha_k L_k(u)}{1+\alpha_k L_k(u)}$.

Researchers had proposed other approximation schemes for the Libor market model [41]. But considering the extra computation effort required, we only consider the above mentioned two schemes for simulation. Now that we know how to calculate or approximate each term in Equation (6.25), we can evolve the Libor rates from the initial time. If one step calculation between two tenor points does not provide us with satisfying accuracy, we can divide the interval between every two tenor points into subintervals. For example, if we divide interval $[T_{n-1}, T_n]$ into two subintervals, then we use Equation (6.25) to evolve the living Libor rates on $[T_{n-1}, (T_{n-1}+T_n)/2]$ and $[(T_{n-1}+T_n)/2, T_n]$ successively. In this way, we hope to improve the computation efficiency while maintaining the accuracy we need. For numerical results on the accuracy of various discretization schemes see [41].

6.2.5 Pricing Formula and Numerical Results

We will give several pricing formulas for the Libor forward rates, caplet prices and coterminal European swaption prices. An analytical approximation is given for the coterminal European swaption price. The following relations between rates and bond prices will often appear in the pricing formulas. We summarize them here.

By definition of the Libor forward rate,

$$L_i(t) = \frac{p(t,T_{i-1}) - p(t,T_i)}{\alpha_i p(t,T_i)}.$$

So we have

$$\frac{p(t,T_i)}{p(t,T_{i-1})} = \frac{1}{1+\alpha_i L_i(t)},$$

for each $i = 1,\ldots,N$ and $t < T_{i-1}$. We can use this to expand the zero coupon bond price between two tenor moments,

$$\begin{aligned} p(T_n,T_m) &= \frac{p(T_n,T_m)}{p(T_n,T_{m-1})} \times \frac{p(T_n,T_{m-1})}{p(T_n,T_{m-2})} \times \ldots \times \frac{p(T_n,T_{n+1})}{p(T_n,T_n)} \\ &= \frac{1}{1+\alpha_m L_m(T_n)} \times \frac{1}{1+\alpha_{m-1}L_{m-1}(T_n)} \times \ldots \times \frac{1}{1+\alpha_{n+1}L_{n+1}(T_n)} \\ &= \prod_{k=n+1}^{m} \frac{1}{1+\alpha_k L_k(T_n)}, \end{aligned}$$

for $0 \le T_n < T_m \le T$.

Specifically, at time $t = 0$ we have the initial bond prices

$$p(0, T_m) = \prod_{k=0}^{m} \frac{1}{1 + \alpha_k L_k(0)},$$

for $T_0 \leq T_m \leq T$.

Let's write the discretely-rebalanced bank account again,

$$\begin{aligned} B^d(0) &= B^d(T_{-1}) = 1, \\ B^d(t) &= \frac{p(t, T_{m(t)})}{\prod_{k=0}^{m(t)} p(T_{k-1}, T_k)}, \qquad \text{for } t > 0, \end{aligned}$$

where $m(t) = \min\{j : T_j \geq t\}$. In simulation, we often care about the value of $B^d(t)$ when $t = T_n$ for $n = -1, 0, \ldots, N$. It is then simplified to be

$$\begin{aligned} B^d(T_n) &= \frac{1}{\prod_{k=0}^{n} p(T_{k-1}, T_k)} \\ &= \prod_{k=0}^{n} (1 + \alpha_k L_k(T_{k-1})). \end{aligned}$$

Here are the pricing formulas under measure $\mathbb{Q}^N$ or $\mathbb{Q}^d$ that we use in the Monte Carlo simulation. The derivation is based on the Risk-Neutral Pricing Principal and Change-of-Numeraire technique. One can turn to [13] for reference.

Libor rate under $\mathbb{Q}^N$,

$$L_i(0) = \frac{p(0, T_N)}{p(0, T_i)} \mathbb{E}^N \left[\frac{L_i(T_{i-1})}{p(T_i, T_N)} | \mathcal{F}(0) \right].$$

Libor rate under $\mathbb{Q}^d$,

$$L_i(0) = \frac{1}{p(0, T_i)} \mathbb{E}^d \left[\frac{L_i(T_{i-1})}{B^d(T_i)} | \mathcal{F}(0) \right].$$

T_i caplet value under $\mathbb{Q}^N$,

$$Capl_i(0) = p(0, T_N) \mathbb{E}^N \left[\frac{\alpha_i \max[L_i(T_{i-1}) - R, 0]}{p(T_i, T_N)} | \mathcal{F}(0) \right].$$

T_i caplet value under $\mathbb{Q}^d$,

$$Capl_i(0) = B^d(0) \mathbb{E}^d \left[\frac{\alpha_i \max[L_i(T_{i-1}) - R, 0]}{B^d(T_i)} | \mathcal{F}(0) \right].$$

The model parameters used in this chapter is either taken or derived from [36]. They are listed in Appendix D.1. We compare the initial Libor rates directly to the market data. But for all ATM caplets, we will compare the implied Black volatility from the simulated caplet prices to those given by market data. Results for the Libor rates are shown in Figure 6.3. Results for the caplet volatilities are shown in Figure 6.4.

We price co-terminal European swaptions only under $\mathbb{Q}^d$. The arbitrage-free price for a European $T_n \times (T_N - T_n)$ payer swaption is,

$$PSN_n^N(0; K) = B^d(0) \mathbb{E}^d \left[\frac{\max[PS_n^N(T_n; K), 0]}{B^d(T_n)} | \mathcal{F}(0) \right].$$

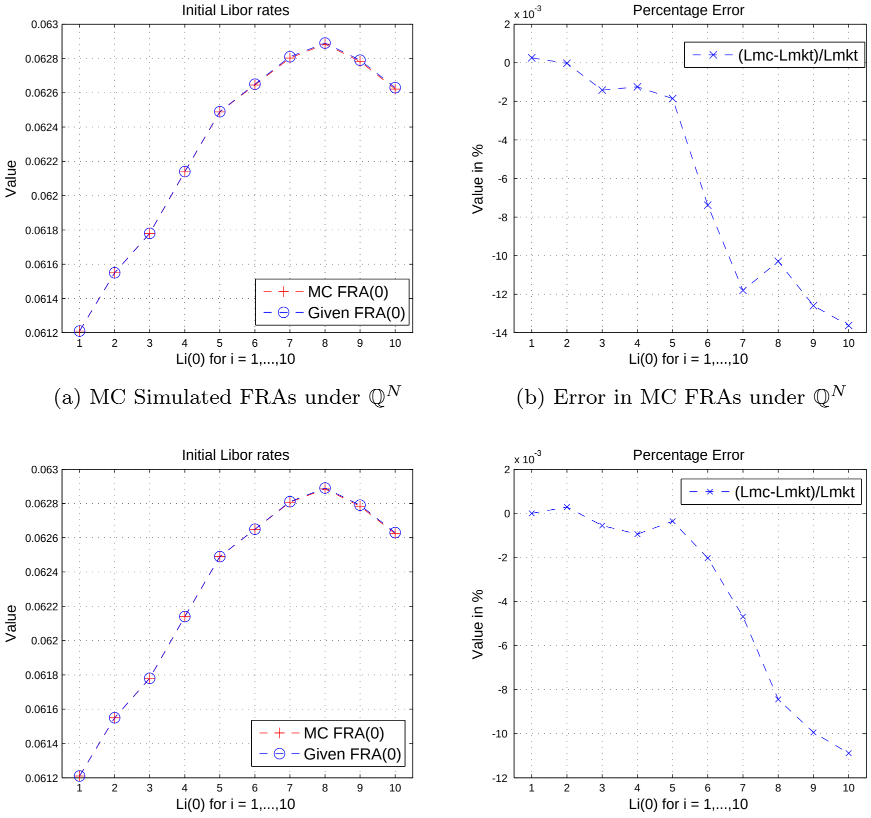
We have the following formula for the price of a European $T_n \times (T_N - T_n)$ payer swaption under $\mathbb{Q}^A$, the swap measure with numeraire $A_n^N(t)$,

$$PSN_n^N(0; K) = A_n^N(0) \mathbb{E}^A \left[\max[R_n^N(T_n) - K, 0] | \mathcal{F}(0) \right]. \tag{6.28}$$

But since we use the Libor market model throughout for simulation, we choose to compute an analytical approximation of the Black volatility for coterminal forward swap rates under the Libor market model

[37]. The derivation can be found in Appendix C. We compute the implied Black volatility from the simulated swaption price and compare it with the analytical approximation. The result is illustrated in Figure 6.5.

The tenor spacing for the data from [36] is pretty short, $\Delta T = 0.25$ to be precise. Our simulation results show that in this case the Monte Carlo engine based on Euler scheme produces comparable accuracy to that based on Predictor-Corrector scheme. In order to save computation time, we choose the one based on Euler scheme to simulate the Bermudan-style Libor Exotics.



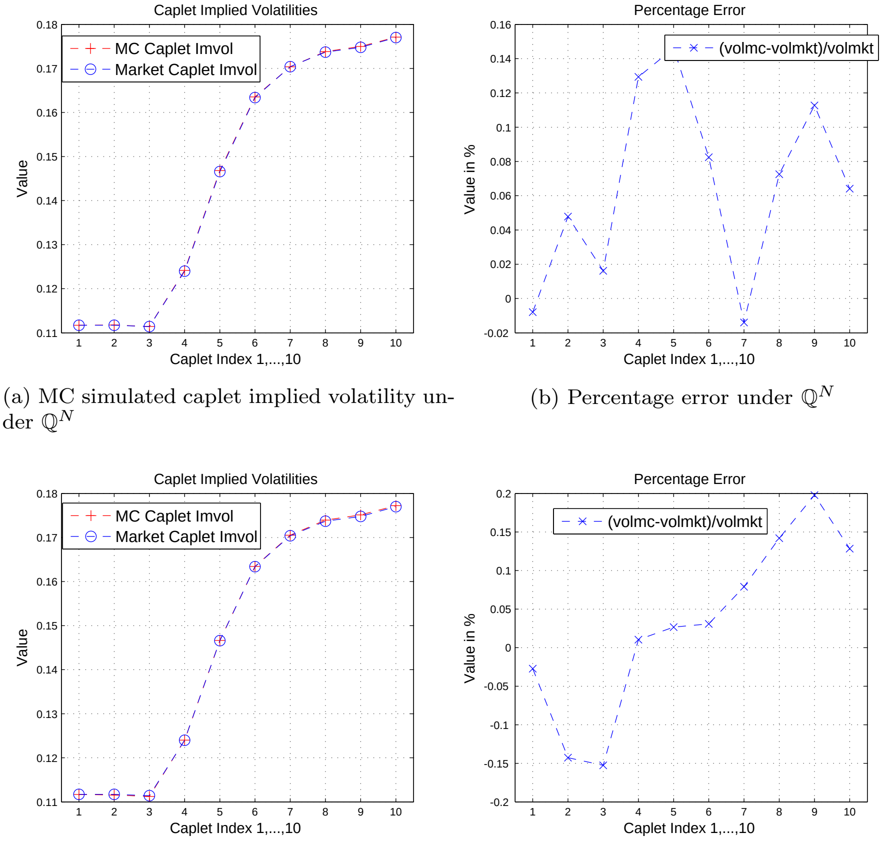
(a) MC Simulated FRAs under $\mathbb{Q}^N$

(b) Error in MC FRAs under $\mathbb{Q}^N$

(c) MC Simulated FRAs under $\mathbb{Q}^d$

(d) Error in MC FRAs under $\mathbb{Q}^d$

Figure 6.3: Comparison of the MC simulated and the given FRAs, MC simulation performed under $\mathbb{Q}^N$ and $\mathbb{Q}^d$. log-Euler scheme for generating the Libors. 10^6 number of paths.



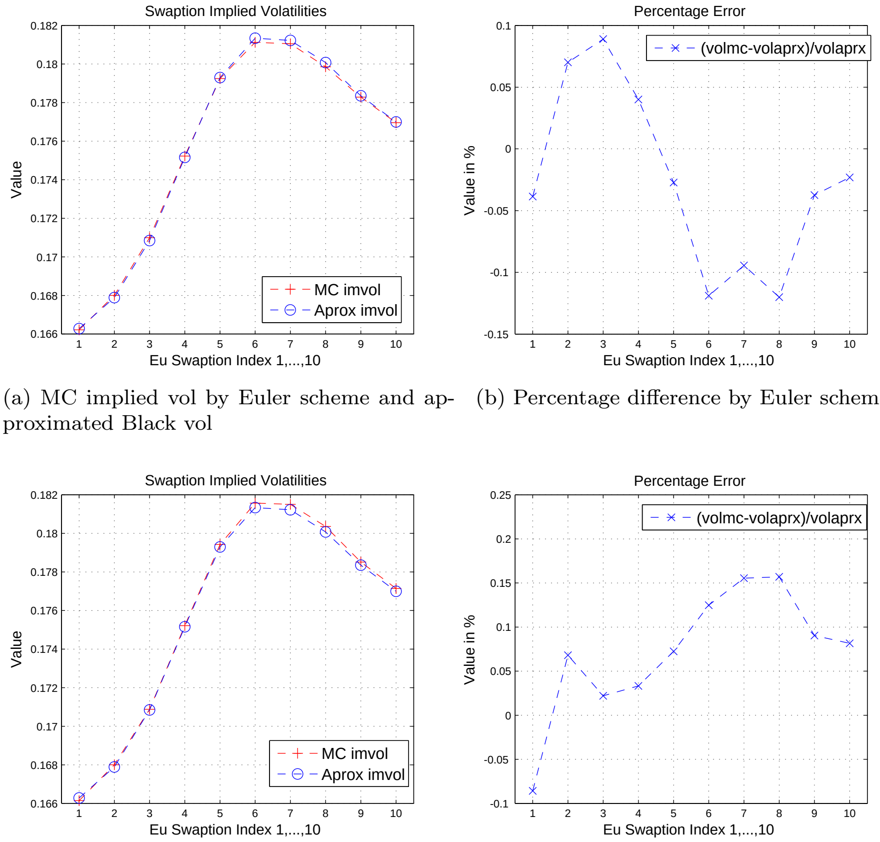
(a) MC simulated caplet implied volatility under $\mathbb{Q}^N$

(b) Percentage error under $\mathbb{Q}^N$

(c) MC simulated caplet implied volatility under $\mathbb{Q}^d$

(d) Percentage error under $\mathbb{Q}^d$

Figure 6.4: Comparison of the MC simulated and the given ATM caplet implied volatility, MC simulation performed under $\mathbb{Q}^N$ and $\mathbb{Q}^d$. Predictor-Corrector scheme for generating the Libors. 10^6 number of paths.



(a) MC implied vol by Euler scheme and approximated Black vol

(b) Percentage difference by Euler scheme

(c) MC implied vol by Predictor-Corrector scheme and approximated Black vol

(d) Percentage difference by Predictor-Corrector scheme

Figure 6.5: Comparison of the MC simulated and the approximated forward swap rate Black implied volatility based on [37], MC simulation performed under $\mathbb{Q}^d$. Euler scheme and Predictor-Corrector scheme are used for generating the Libors. 10^6 number of paths.

6.3 Callable and Cancellable Libor Exotics

Let's recall the dynamic programming formulation of Bermudan option pricing in Section 2.2.1. We will first use the time grid $0 = T_0 < T_1 < \ldots < T_N = T$ to illustrate the procedure. Remember that all the option value quantities are in time 0 money, i.e. discounted by proper numeraire back to time 0. The recursions that we will use for Monte Carlo simulation are (2.12) and (2.13), which we rewrite here.

$$\begin{cases} V_N = Z_N, & \\ V_i = \max\left[Z_i, H_i\right], & \text{for } i = 1, \ldots, N-1. \end{cases}$$

where $H_i = \mathbb{E}(V_{i+1}|\mathcal{F}_i)$. At maturity we have $H_N = 0$.

The sequence of stopping times τ_i can be determined by,

$$\begin{cases} \tau_N = N, & \\ \tau_i = i, & \text{if } \omega \in \Psi_i. \\ \tau_i = \tau_{i+1}, & \text{otherwise.} \end{cases}$$

for each $i = 0, \ldots, N-1$, where $\Psi_i = \left\{\omega \in \Omega, Z_i^{(\omega)} > H_i^{(\omega)}\right\}$ defines the exercise region at each time step.

The option value at time T_i is given by $V_i = \mathbb{E}[Z(\tau_i)|\mathcal{F}_i]$. In particular $V_0 = \mathbb{E}[Z(\tau_0)|\mathcal{F}_0]$. Recursion (2.13) can be used to derive an algorithm for computer simulation of the Bermudan option value. We just need to approximate Z_i and H_i (if they are not known at time T_i) at each step. The approximation of Z_i and H_i is realized by the Longstaff-Schwartz method, i.e. the LSM algorithm.

We can now modify the above procedure to price Bermudan Swaption and Callable and Cancellable Libor Exotics (CLE and CnLE). We use the same time discretization as defined earlier. The interval $[0, T]$ is divided by the equal distant grid $0 = T_{-1} < T_0 < \ldots < T_M = T$, where $\alpha_i = T_i - T_{i-1} = T/(M+1)$ for each $i = 0, 1, \ldots, M$.

6.3.1 LSM for Pricing a CLE

Let CF_i be the discounted net cash flow at time T_i to the option holder. For example, for a payer swaption, CF_i equals the time T_i floating leg payment minus fixed leg payment discounted to time 0 by a proper numeraire. Since we choose $\mathbb{Q}^d$ as the simulation measure, the associated numeraire is the discretely rebalanced bank account $B^d(t)$.

For a callable $T_0 \times (T_N - T_0)$ swaption, the holder has the right to enter a $T_n \times (T_N - T_n)$ swap at any T_n for $n = 0, \ldots, N-1$. Clearly, $H_{N-1} = 0$. We adopt the convention that at exercise decision moment, T_n, $L_{n+1}(T_n)$ is already fixed [45]. That is to say CF_{i+1} is $\mathcal{F}_i$ measurable. Hence, $Z_{N-1} = CF_N$ is $\mathcal{F}_{N-1}$ measurable. At T_n for $n = 0, 1, \ldots, N-2$, $Z_n = \Sigma_{i=n+1}^{N}(CF_i)$ is no longer $\mathcal{F}_n$ measurable. Hence the exercise value at T_n must also be expressed as a conditional expectation. In [57], Versendaal outlined an algorithm for calculating the price of CLE, in which he used the exercise value Z_n at T_n directly. We think this practice introduces foresight bias because it makes exercise decision based on information in the future. Sticking to the double regression scheme, we have from recursion (2.12) that,

$$\begin{cases} V_{N-1} = \max[Z_{N-1}, H_{N-1}], & \\ V_i = \max\left[\mathbb{E}(Z_i|\mathcal{F}_i), H_i\right], & \text{for } i = 0, \ldots, N-2. \\ V_{-1} = V_0. & \end{cases} \quad (6.29)$$

where $H_i = \mathbb{E}(V_{i+1}|\mathcal{F}_i)$.

At each T_i for $i = 0, \ldots, N-2$, we obtain estimation of the expected exercise value Z_i and the expected hold value H_i by the LSM algorithm. Suppose we use the same set of basis functions, $\Phi_1, \ldots, \Phi_R$. The approximations along each path are

$$\mathbb{E}(V_{i+1}|\mathcal{F}_i)^{(k)} \approx \sum_{r=1}^{R} \hat{\beta}_{i,r}\Phi_r(X_i^{(k)}) = \hat{H}_i^{(k)},$$

and

$$\mathbb{E}(Z_i|\mathcal{F}_i)^{(k)} \approx \sum_{r=1}^{R} \hat{\gamma}_{i,r}\Phi_r(X_i^{(k)}) = \hat{Z}_i^{(k)},$$

for each $k = 1, \ldots, N_p$.

We define the sequence of stopping time $\tau_i : \Omega \to \{i, \dots, N\}$ for each $i = 0, \dots, N-1$ as $\tau_i = \inf\{i \leq j \leq N, \hat{Z}_j > \hat{H}_j\}$.

From recursion (2.13), we have the following recursion for pathwise stopping time,

$$\begin{cases} \tau_N = N, & \\ \tau_i^{(k)} = i, & \text{if } \hat{Z}_i^{(k)} > \hat{H}_i^{(k)}, \\ \tau_i^{(k)} = \tau_{i+1}^{(k)}, & \text{otherwise.} \end{cases} \quad (6.30)$$

for each $i = 0, \dots, N-1$. There is no exercise opportunity at time 0, so $\tau_{-1}^{(k)} = \tau_0^{(k)}$. The option value at time $T_{-1} = 0$ is

$$V(T_{-1}) = \mathbb{E}[Z(\tau_{-1})|\mathcal{F}_{-1}] \approx \frac{1}{N_p} \sum_{k=1}^{N_p} \sum_{i=\tau_{-1}^{(k)}+1}^{N} CF_i^{(k)}.$$

We use the definition that $\sum_{N+1}^{N} \dots = 0$.

6.3.2 LSM for Pricing a CnLE

For a Cancellable $T_0 \times (T_N - T_0)$ swaption. The holder enters a $T_0 \times (T_N - T_0)$ swap at time 0. Then at each T_n, for $n = 0, \dots, N-1$, the holder has the right to stop the swap and cancel all the remaining cash flows CF_j for $j = n+1, \dots, N$. We use the convention that when it comes to making the decision at time T_n, CF_{n+1} is already known, i.e. CF_{n+1} is $\mathcal{F}_n$ measurable. If the swaption is cancelled at time T_n, the holder gets nothing from T_{n+1} on. If it is not cancelled, the swaption is worth the discounted remaining (hold) value H_n. Notice that this value has a different definition than the hold value defined before. Now recursion (2.12) becomes

$$\begin{cases} H_{N-1} = \max[CF_N, 0], & \\ H_i = \max\left[\mathbb{E}(H_{i+1}|\mathcal{F}_i), 0\right], & \text{for } i = 0, \dots, N-2, \\ H_{-1} = H_0. & \end{cases} \quad (6.31)$$

At each T_i for $i = 0, \dots, N-2$, we only need to obtain estimation of the expected remaining value H_i by the LSM algorithm. The approximation along each path is

$$\mathbb{E}(H_{i+1}|\mathcal{F}_i)^{(k)} \approx \sum_{r=1}^{R} \hat{\beta}_{i,r} \Phi_r(X_i^{(k)}) = \hat{H}_i^{(k)}.$$

for each $k = 1, \dots, N_p$.

We define the sequence of stopping time $\tau_i : \Omega \to \{i, \dots, N\}$ for each $i = 0, \dots, N-1$ as $\tau_i = \inf\{i \leq j \leq N, \hat{H}_j < 0\}$.

From recursion (2.13), we have the following recursion for pathwise stopping time,

$$\begin{cases} \tau_N = N, & \\ \tau_i^{(k)} = i, & \text{if } \hat{H}_i^{(k)} < 0, \\ \tau_i^{(k)} = \tau_{i+1}^{(k)}, & \text{otherwise.} \end{cases} \quad (6.32)$$

for each $i = 0, \dots, N-1$. There is no exercise opportunity at time 0, so $\tau_{-1}^{(k)} = \tau_0^{(k)}$. The option value at time $T_{-1} = 0$ is

$$V(T_{-1}) = \mathbb{E}[H(\tau_{-1})|\mathcal{F}_{-1}] \approx \frac{1}{N_p} \sum_{k=1}^{N_p} \sum_{i=1}^{\tau_{-1}^{(k)}} CF_i^{(k)}.$$

We use the definition that $\sum_{1}^{0} \dots = 0$.

6.3.3 Parity between CLE and CnLE

There is a parity relationship between the prices of CLE and CnLE. As mentioned by Bender et al. [8], time 0 value of a CLE can be expressed as the sum of the value of the underlying exotic swap and the value of the corresponding CnLE with cash flows of the opposite sign,

$$\begin{aligned} V^{CLE^+}(T_{-1}) &= \sup_{\tau\in\{0,\ldots,N-1\}} \mathbb{E}\left[\sum_{j=\tau+1}^{N} CF_j|\mathcal{F}_{-1}\right] \\ &= \mathbb{E}\left[\sum_{j=1}^{N} CF_j|\mathcal{F}_{-1}\right] + \sup_{\tau\in\{0,\ldots,N-1\}} \mathbb{E}\left[\sum_{j=1}^{\tau} -CF_j|\mathcal{F}_{-1}\right] \\ &= V^{SW^+}(T_{-1}) + V^{CnLE^-}(T_{-1}). \end{aligned}$$

The + and − in the above expression denote the direction of the cash flows. Notice that a payer and receiver swap have exactly the opposite cash flows. We have the following parity relations,

$$\begin{aligned} V^{CPSN}(0;\theta) &= V^{PS}(0;\theta) + V^{CnRSN}(0;\theta), \\ V^{CRSN}(0;\theta) &= V^{RS}(0;\theta) + V^{CnPSN}(0;\theta). \end{aligned} \tag{6.33}$$

θ represents all the parameters defining the swap and the swaptions. The superscripts denotes the swap or swaption type. For example, PS means a payer swap, $CnRSN$ means a cancellable receiver swaption. The following numerical example shows that our Monte Carlo simulation for the CLE and CnLE indeed satisfies the parity relation (6.33). We perform two separate LSM simulations to price a callable payer swaption and a cancellable receiver swaption. From these two LSM simulations we also get the price of the European $T_0 \times (T_N - T_0)$ payer swap. We use the parity (6.33) to calculate the corresponding cancellable receiver swaption or the corresponding callable payer swaption respectively.

Table 6.1: Example of the parity between a callable payer swap and a cancellable receiver swap. All the numbers reported are in basis points (1bp = 0.0001).

Strike	Value					
	Simulated	Simulated	Calculated	Simulated	Simulated	Calculated
K	$V^{PS}(0)$	$V^{CPSN}(0)$	$V^{CnRSN}(0)$	$V^{PS}(0)$	$V^{CnRSN}(0)$	$V^{CPSN}(0)$
823	-453.6	11.2	464.8	-453.6	464.8	11.3
623	-0.66	82.8	83.5	-0.66	83.0	82.4
423	452.2	452.2	≈ 0	452.2	≈ 0	452.2

The simulation shown in Table 2.1 is performed with model quotes taken from [36]. Monte Carlo simulation is performed on swaptions fixed at the first 10 tenor dates. $K = 623$bps is the forward swap rate. We run simulation with $K = 823$bps, 623bps, 423bps for different moneyness. We use the same 10^6 paths to estimate the exercise and hold value functions as well as to perform the Monte Carlo averaging. The regressors used in the simulation consists of a constant, the value of floating leg, the value of fixed leg and the value of a curve tilt. Details about the basis functions will be discussed in the next section. Simulation results show close to parity relation at different moneyness. It seems that either algorithm can be used to compare the relative merit of different explanatory variables and basis functions. Later, when we compute the upper bound of Bermudan swaptions, we found that it is better to implement the algorithm based on valuation of the CnLE instead of CLE.

6.3.4 Explanatory Variable and Basis Function

We have demonstrated previously that the LSM can be adapted to price Bermudan swaptions, be it CLE or CnLE. However, there remains a question to answer, namely what set of regressors gives more accurate price.

Choosing regressors in the LSM is regarded somewhat of an art. Since how close the LSM approximates the true stopping time depends on the detailed option specifications, it is not surprising that one set of regressors may work well with one option but not so much with another. There is in general no optimal set of regressors that gives better prices for all types of options. What we hope to achieve is to identify the criterion according to which we can evaluate the regressors.

Piterbarg put forward a good guideline on this issue [45]. He mentioned that the main difficulty in modeling Bermudan-style LE lies in the strong dependence on unobservable volatility parameters. These are partly taken care of by choices of model implementation for the Libor market model. What remains is how to best utilize the information available at time T_n through the choice of regressors to obtain a tight LSM price. The regressors are constructed by evaluating certain basis functions at the value of some explanatory variables. Piterbarg argues that in order to avoid the trouble of overfitting, one should use a relatively simple parametric family of basis functions and to choose the explanatory variables carefully. Based on our simulation result with the Black-Scholes world, we will not spend much time in comparing different basis functions such as polynomials but focus on the explanatory variables.

Piterbarg argued that the explanatory variables should be financially meaningful. He suggests that the overall level of the rates and the slope of the rate curve are the two most significant factors affecting the option price. The coterminal forward swap rate captures information about the overall level. One can use a short tenor rate to capture the curve slope information, since the regression will automatically take care of the difference between this rate and other variables.

Piterbarg's reasoning is based on market observations. The Libor forward rates often move in accordance with one another resulting in such stylized movements as a parallel shift, a curve tilt or a hump. However, it is doubted that whether these basic movements can provide adequate information in the LSM method for some more complex type of swaptions. Maybe it is better to just feed the algorithm with all the living Libor rates as explanatory variables and let it find the information by itself. We have performed large amount of numerical experiments to compare between different setups of the regressors for different types of swaptions. The results can be found in the Chapter 7. We give some descriptions about the regressors that we think is worth experimenting.

As our base case, we take the regressors used by Buitelaar [15]. He studied control variate method for pricing Bermudan swaptions in the Libor market model. Here is the set of explanatory variables that he used.

$$\begin{aligned}
S_1(T_n) &= 1, \\
S_2(T_n) &= \sum_{j=n+1}^{N} \alpha_j p(T_n, T_j), \\
S_3(T_n) &= \sum_{j=n+1}^{N} \alpha_j L_j(T_n) p(T_n, T_j), \\
S_4(T_n) &= \sum_{j=n+1}^{N} \alpha_j L_j(T_n) p(T_n, T_j)(n - i_{mid}), \\
S_5(T_n) &= \mathrm{CF}_{n+1},
\end{aligned}$$

where $i_{mid} = \frac{n+N}{2}$. S_2 is the time T_n present value of the fixed leg omitting the constant coupon rate K. S_3 is the time T_n present value of the floating leg. S_4 is the time T_n present value of a curve tilt. It is defined in such a way that if the remaining forward rate curve exhibits a counter-clockwise tilt around the center point $T_{i_{mid}}$, S_4 increases. In order to capture the path dependent feature of a Snowball, the immediate next cash flow CF_{n+1} is added to the regressor matrix. Some squared terms as well as cross product terms of the above explanatory variables are also used as regressors. This set of regressors is chosen according to Piterbarg's guideline, although not completely according to his recommendation. The above mentioned set of regressors give information about the overall level as well as the curve slope.

We would like to find out if these regressors give better result than the ones recommended by Piterbarg. Looking at the payoff form of a fixed-for-floating swaption (6.28), it is natural to think that the forward swap rate given by (6.5) should be used as an explanatory variable. We expect it to work even with some more complicated type of swaptions. In terms of the notation here, the forward swap rate is given by,

$$S_6(T_n) = \frac{S_3(T_n)}{S_2(T_n)} = \frac{\sum_{j=n+1}^{N} \alpha_j L_j(T_n) p(T_n, T_j)}{\sum_{j=n+1}^{N} \alpha_j p(T_n, T_j)}.$$

At time T_n the immediate next Libor rate is $L_{n+1}(T_n)$, which we choose as a short tenor rate in order to capture the curve slope information.

$$S_7(T_n) = L_{n+1}(T_n).$$

For swaptions other than the basic fixed-for-floating one, the coupon leg provides more information than the fixed leg. We also use the time T_n present value of the coupon leg as an explanatory variable,

$$S_8(T_n) = \sum_{j=n+1}^{N} \alpha_j C(L_j(T_n))p(T_n, T_j).$$

where $C(x)$ denotes the coupon function depending on rate x. In the above expression, time T_n rates value $L_j(T_n)$ are used to approximate future coupon leg value. For a Snowball swaption, the coupon function is path-dependent. In order to determine C_n one need to know C_{n-1}. This additional path-dependency poses a challenge to the LSM method. What we can do is try approximating the coupon values with the information available at time T_n. One possibility is to freeze the Libor rates at time T_{n-1} and compute the coupon rates according to Equation (6.8). Another more involved approximation will be discussed in Chapter 7.

We would also like to find out whether supplying all the living Libor rates to the LSM algorithm improves the pricing result. In the related experiments, we choose at each step all the living Libor rates, the squared terms and their cross-products as regressors. Thus the regression matrix changes size at each exercise moment, with much larger size at longer time to maturity. One of the concerns we have based on our simulation result with the Black-Scholes world is that when using too many terms of regressors, the simulated LSM price could be high biased as compared to our benchmark. Whether this is due to Monte Carlo error or some other numerical error or convergence issue is unknown. It is natural to ask if this will be an issue when pricing Bermudan-style Libor Exotics with the LSM, especially when all the living Libor rates are used as regressors directly. With the upper bound algorithm, we will be able to check if this ever happens.

6.3.5 Upper Bound for C(n)LE

We do not have any alternative benchmark price for the Bermudan-style LE. The upper bound algorithm makes it possible to evaluate LSM pricing results without an alternative benchmark. As stated in the last section, it allows the evaluation of different regressor configurations in the LSM method.

When trying to implement the upper bound algorithm for the C(n)LE, one should be careful of what information is available at the moment. If the exercise value Z_n is $\mathcal{F}_n$ measurable, then it can be directly computed at time T_n. In our experiment, stocks in the Black-Scholes world or the Heston model both fall into this category. But in case of a swaption with multiple future cash flows to be fixed, Z_n is not known at time T_n. One should approximate $\mathbb{E}[Z_n|\mathcal{F}_n]$ by the LSM regression coefficients, and then make exercise decision based on this value and the hold value H_n. To Compute Z_n directly from the realization of future cash flows implies perfect foresight about future information. This would introduce a significant upward bias. The hold value, H_n, is always approximated by regression fitted multi-linear model.

In our experiment in the Black-Scholes world and the Heston model, we work with Bermudan put on the underlying stock. When experimenting with CLE and CnLE in the Libor market model, we notice that although our pricing algorithm gives the theoretical parity relation between prices of the corresponding CLE and CnLE, it is difficult to produce reasonable upper bound for CLE without resorting to perfect foresight. Due to limited time for this project, we decide to focus on computing the upper bound of CnLE. In this case, we only need to compare expectation of the remaining cash flows until future exercise moment (if this ever happens) to the fixed value 0. Regarding Equation (3.9), Z_n and H_n represents the remaining cash flows value if exercise or hold at time T_n. Exercise means canceling all remaining cash flows, so we have $Z_n = 0$ at each exercise moment.

7 Numerical Results and Discussions

We present in this chapter the main results from our numerical experiment. We give the reasoning for our choices of simulation configuration and discuss the impact on LSM pricing result. We will show results in various models and payoff types. Whenever an alternative benchmark is available, we will give the numerical convergence result for that benchmark. When no alternative benchmark is available, we will present result from upper bound simulation. Upper bound results are also given for the Bermudan put in the Black-Scholes world and the Heston model.

7.1 Bermudan Put in the Black-Scholes World

Unless stated otherwise, we consider the following set of model and simulation parameters.

Parameters for the underlying asset price dynamics under the risk-neutral measure: $r = 0.06$, $\sigma = 0.3$.

Parameters for the Bermudan put option: $K = 10$, $T = 1$. Let $\Delta t = \frac{T}{52}$, the exercise moments are $\left\{\frac{1}{52}, \frac{2}{52}, \ldots, \frac{51}{52}, 1\right\}$.

Parameters for the Monte Carlo simulation: time step $\Delta t = \frac{T}{52}$, 10^5 number of sample paths for each Monte Carlo (LSM) simulation result. An average over 100 such results gives the pricing result. We choose only ITM sample paths at each early exercise moment for regression, i.e. only those paths such that $S(T_n) < K$ at T_n.

7.1.1 Convergence of the Benchmark

We build two benchmarks for comparison. The first one is based on Binomial Tree. We choose $52 \times 400 = 20800$ time steps. The second one is based on Finite Difference with θ scheme and PSOR iterative matrix equation solver. We choose $52 \times 400 = 20800$ time steps and 4000 space steps. The time steps are chosen as a multiple of 52 such that all early exercise moments are on grid. In this way, we can rule out the error caused by choosing different early exercise moments with different numerical schemes.

The Binomial Tree is used to cross validate the FD scheme. Pricing result from the FD scheme is used as the benchmark against which all Monte Carlo simulations are compared.

Table 7.1: Pricing result for a Bermudan Put with Binomial Tree

Index	# Steps	Initial Asset Price $S(0)$				
		6	8	10	12	14
P1	52×4	3.99135	2.10287	0.95161	0.39574	0.15415
P2	52×40	3.98876	2.10162	0.95166	0.39454	0.15439
P3	52×400	3.98850	2.10158	0.95166	0.39450	0.15433

Table 7.2: Convergence for a Bermudan Put with Binomial Tree

Index	Initial Asset Price				
	6	8	10	12	14
P2 − P1	-2.6 $\times 10^{-3}$	-1.2$\times 10^{-3}$	4.84$\times 10^{-5}$	-1.2 $\times 10^{-3}$	2.45$\times 10^{-4}$
P3 − P2	-2.59$\times 10^{-4}$	-3.55$\times 10^{-5}$	3.00$\times 10^{-6}$	-4.87$\times 10^{-5}$	-6.70$\times 10^{-5}$

Table 7.3: Pricing result for a Bermudan Put with Finite Difference

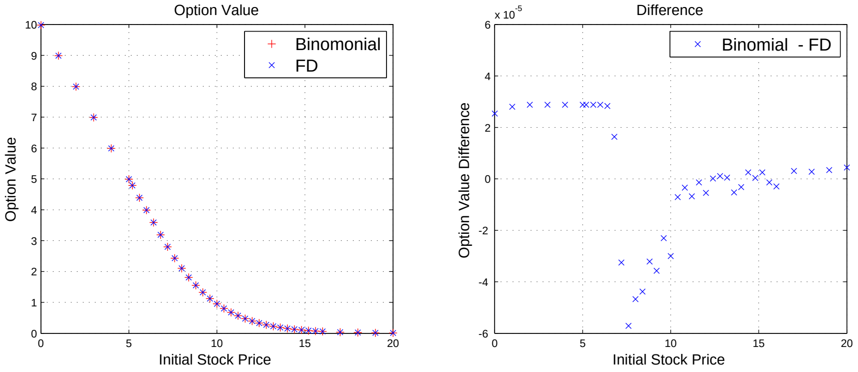
Index	Δs	Initial Asset Price				
		6	8	10	12	14
P1	1	3.98854	2.08569	0.93284	0.38346	0.14989
P2	0.1	3.98847	2.10142	0.95149	0.39436	0.15426
P3	0.01	3.98847	2.10158	0.95167	0.39448	0.15432

Table 7.1 and Table 7.2 shows the convergence of the Binomial Tree with several initial asset price values. We compare three different number of steps for the tree. Table 7.3 and Table 7.4 shows the convergence of the FD scheme with several initial asset price values. We compare three choices of the space step size, Δs. Time step convergence of the FD scheme is not checked, we fix 20800 time steps in

Table 7.4: Convergence for a Bermudan Put with Finite Difference

Index	Initial Asset Price				
	6	8	10	12	14
P2 − P1	-6.67×10^{-5}	1.57×10^{-2}	1.86×10^{-2}	1.09×10^{-2}	4.4×10^{-3}
P3 − P2	-2.9×10^{-7}	1.55×10^{-4}	1.77×10^{-4}	1.23×10^{-4}	6.4×10^{-5}

the simulation. Depending on different moneyness, convergence differs for the two numerical schemes. Figure 7.1 shows the difference of pricing results from the Binomial Tree and the FD scheme. The most significant difference is about 6×10^{-5} where $S(0) = 7.8$, and that the difference in OTM region are all below 1×10^{-5}. Although we can not say much about whether the true Bermudan option value lies above or below the benchmark prices, convergence of both our benchmarks brings confidence that the true option value is nearby. As we will see in later sections, the LSM price varies around our benchmark price when computed using different regressor configurations. Our benchmarks are believed to be accurate enough compared to the LSM result.



(a) Pricing Result

(b) Price Difference

Figure 7.1: Pricing result and difference using Binomial Tree and Finite Difference for a Bermudan Put in the Black-Scholes world

7.1.2 Selection Criteria and Simulation Choices

We try to answer the following two questions:

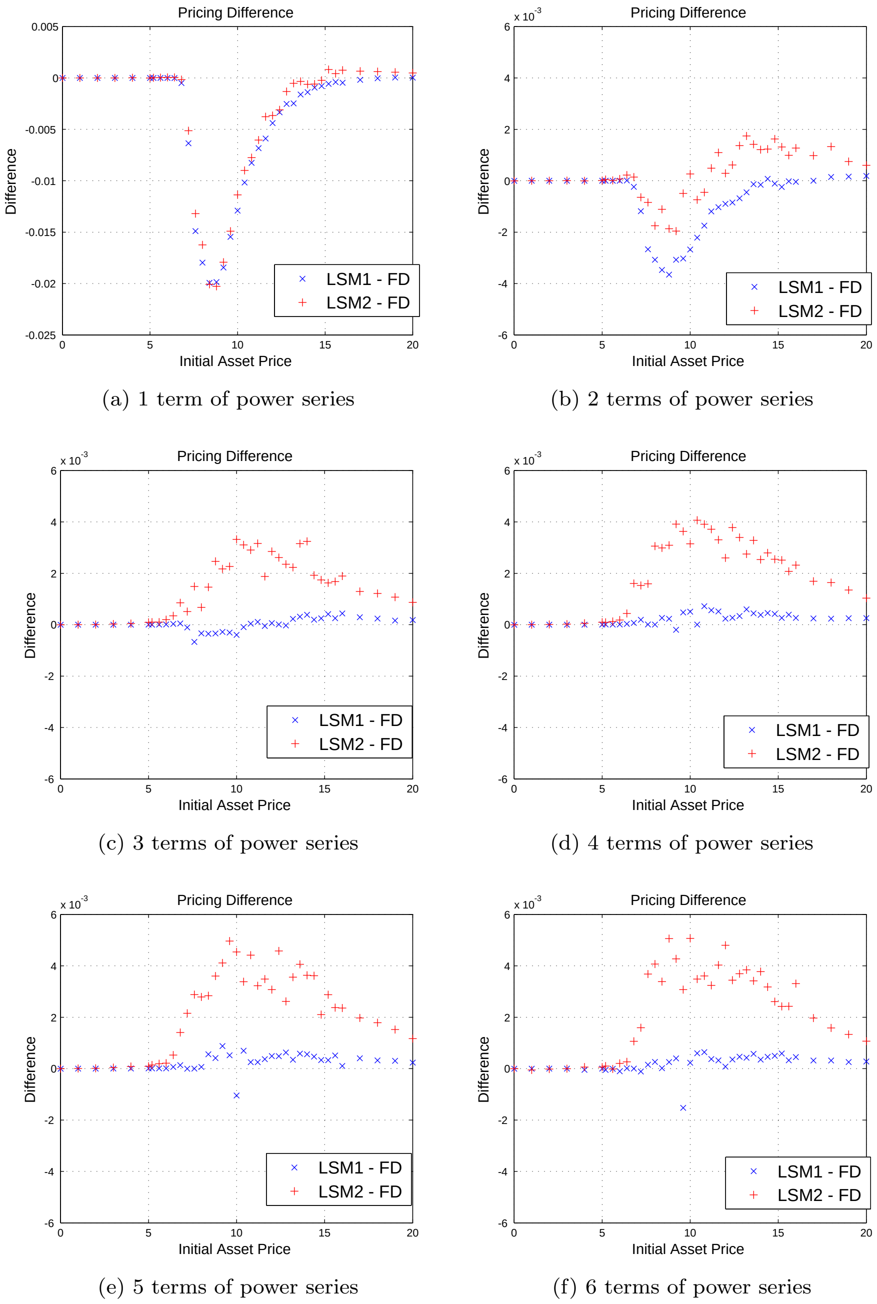
1. How do we judge which set of basis functions gives better pricing result?
2. Which set of sample paths should we choose for regression at each exercise moment?

The first question is closely related to convergence property of the LSM method. Glasserman and Yu [29] pointed out that the number of sample paths necessary for convergence of the LSM method is affected by the number of regressors used. While our goal is to find the most appropriate explanatory variable and basis functions, taking the speed of convergence into consideration would significantly complicate the problem. Hence our approach is to take as many paths as possible to facilitate convergence and to compare the pricing result calculated from that many paths. Although this means that it may take more computation time than necessary for some quick convergence cases, it is necessary to do so for slower convergence cases. Nevertheless, we are limited by the computer hardware in terms of computational power. In our implementation of the LSM algorithm, we need to invert some large matrices. The matrix size can be as large as the number of sample paths. Inverting these large matrices can be handled by the Matlab built-in matrix division operation, which applies certain variant of Gauss elimination. However, Matlab has its limit of the size of matrix that can be generated depending on the amount of memory available. In case of simulation in Black-Scholes world, we find out that sometime it is difficult to generate matrix of size $10^6 \times 10^6$ consistently depending on the amount of memory available to Matlab at a particular moment.

Yet we are still interested in how the number of sample paths affects the pricing result. Table 7.5 and 7.6 shows the simulation results with 10^5 and 10^4 sample paths respectively. We simulated with a group of initial asset price around the strike. Then average price difference from simulation with these initial asset prices are calculated. The price difference is define as

$$\text{Price Difference} = \text{Price by LSM} - \text{Price by Benchmark}.$$

It is not easy to find a pattern from the numbers in these two tables, plots of the price difference would provide more information. Figure 7.2 shows the price difference simulated with the 2 settings, LSM1 with 10^5 sample paths, LSM2 with 10^4 sample paths. We observe that with one term of power series as basis function, both cases give low biased result compared to the FD benchmark. As the number of basis functions increases, the price difference of both cases increases and finally becomes positive, indicating that the LSM price rises higher than the FD benchmark. The result with 10^5 paths agrees with the FD benchmark more closely.



(a) 1 term of power series

(b) 2 terms of power series

(c) 3 terms of power series

(d) 4 terms of power series

(e) 5 terms of power series

(f) 6 terms of power series

Figure 7.2: Price difference for a Bermudan put in the Black-Scholes world, $T = 1$, $\Delta t = T/52$, power series as basis functions. LSM1 simulated with 10^5 sample paths, LSM2 simulated with 10^4 sample paths.

Here are a few comments on the result:

First, as pointed out before, our benchmark also suffers from numerical error. So a pricing result higher than the benchmark does not necessarily mean that it is higher than the true option value based on optimal exercise strategy. But the convergence result from our FD benchmark does suggest that the FD price should lie close to the true option price.

Table 7.5: Price difference for a Bermudan put with Power series as basis function. $T = 1$, $\Delta t = T/52$, 10^5 number of paths, ITM paths for regression.

Polynomial	Term Number	Average Diff $S(0) \in [6,14]$	Most Positive Diff	Most Negative Diff
Power	1	-8.2×10^{-3}	4.2×10^{-5}@19	-1.99×10^{-2}@8.4
Power	2	-1.5×10^{-3}	1.85×10^{-4}@20	-3.7×10^{-3}@8.8
Power	3	-8.33×10^{-5}	4.37×10^{-4}@16	-6.7×10^{-4}@7.6
Power	4	2.69×10^{-4}	7.2×10^{-4}@10.8	-1.97×10^{-4}@9.2
Power	5	2.97×10^{-4}	8.79×10^{-4}@9.2	-1×10^{-3}@10
Power	6	1.79×10^{-4}	6.38×10^{-4}@10.8	-1.5×10^{-3}@9.6
Power	7	-1.72×10^{-4}	4.39×10^{-4}@14.4	-2.1×10^{-3}@8.8
Power	8	-4.9×10^{-4}	4.09×10^{-4}@15.6	-2.1×10^{-3}@10

Table 7.6: Price difference for a Bermudan put with Power series as basis function. $T = 1$, $\Delta t = T/52$, 10^4 number of paths, ITM paths for regression.

Polynomial	Term Number	Average Diff $S(0) \in [6,14]$	Most Positive Diff	Most Negative Diff
Power	1	-7.4×10^{-3}	8.26×10^{-4}@15.2	-2×10^{-2}@8.8
Power	2	-4.5×10^{-5}	1.7×10^{-3}@13.2	-2×10^{-3}@9.2
Power	3	2.1×10^{-3}	3.3×10^{-3}@10	-3.4×10^{-6}@0.01
Power	4	2.8×10^{-3}	4.1×10^{-3}@10.4	-3.4×10^{-6}@0.01
Power	5	3.1×10^{-3}	5×10^{-3}@9.6	-3.4×10^{-6}@0.01
Power	6	3.3×10^{-3}	5.1×10^{-3}@10	-5×10^{-5}@1

Table 7.7: Price difference for a Bermudan put with Power series as basis function. $T = 1$, $\Delta t = T/52$, $N_p = 10^5$ number of paths. All paths for regression.

Polynomial	Term Number	Average Diff $S(0) \in [6,14]$	Most Positive Diff	Most Negative Diff
Power	1	-6.29×10^{-2}	7.1×10^{-8}@5.6	-9.2×10^{-2}@11.6
Power	2	-3.6×10^{-2}	9.9×10^{-5}@6.8	-6.2×10^{-2}@10.8
Power	3	-2.19×10^{-2}	1.35×10^{-5}@6.4	-3.44×10^{-2}@11.6
Power	4	-1.52×10^{-2}	9.86×10^{-5}@6.4	-2.44×10^{-2}@11.2
Power	5	-1.27×10^{-2}	1.59×10^{-4}@6.4	-2.35×10^{-2}@11.2
Power	6	-9.7×10^{-3}	1.48×10^{-4}@6.8	-1.94×10^{-2}@11.2

Second, when using more than 3 terms of power series, LSM2 results are significantly higher than either the benchmark or LSM1. Hence it should lie above the true option value. A rule of thumb criterion for evaluating the performance of the LSM method is that the higher the LSM price is, the closer it is believed to be near the true option value. This argument is based on the assumption that the LSM result is fully converged, and it should be below the true option value as predicted by the suboptimal exercise rule. However, as we observe here, the pricing result with finite number of sample paths may very well be high biased. In order to find an explanation, one need to study the convergence of the LSM method under specific simulation setup. This is a difficult problem considering all the factors involved in the algorithm. We will not investigate further into this convergence topic.

Third, we observe that the LSM result does not keep increasing after adding just a few terms of regressors. Hence when we can only use limited number of sample paths, a practical criterion for selecting the best regressor configuration would be to choose such configuration with the least number of terms reaching this "price saturation". We will see that with 10^5 sample paths, the "saturated" prices are pretty close to our FD benchmark.

Fourth, to get more insight into this matter, one can look into the upper bound of the option value. Corresponding simulation result will be presented in Section 7.1.4.

The question about which set of sample paths to choose is much more straight forward. Comparing the results from Table 7.5 and Table 7.7, we observe that the price result computed by regression on all the sample paths is significantly lower than the result computed from only ITM paths. A put option has a fixed strike, hence moneyness is well defined by only look at the underlying asset value at the exercise moment. That is, if at time t, along path k, $S^k(t) < K$, the option is ITM, meaning its intrinsic value is greater than zero, then this path will be selected for regression. Since the option's continuation value is nonnegative, one only need to approximate the continuation value function and make early exercise decision when the option is ITM.

When using linear least square regression to estimate the multi-linear model for the continuation value function, the estimate of the coefficients are calculated such that the sum of all squared residuals is minimized. This is a global fit of the model to all the data points. One possible explanation of the lower price result when selecting all sample paths is that fitting globally makes the fit to ITM data points worse than if only fitting those ITM data points. This is the drawback of the linear least square regression. Other more advanced regression methods may be considered for improvement.

7.1.3 Performance with Different Regressor Configurations

In the framework of linear least square fitting, we construct the regressor matrix by choosing appropriate explanatory variables and basis functions. In our asset model, the only randomness comes from the GBM asset dynamics. Possible choices of explanatory variables would be the underlying asset value and some quantities related to the underlying asset value. In this section, we give some numerical results regarding different choices of the explanatory variables and basis functions.

For the choice of explanatory variable, we consider the following 3 possibilities:

1. underlying asset value

2. exercise value of the option

3. value of a co-terminal European put based on the Black-Scholes formula using the underlying asset value at exercise moment as the initial value for its underlying.

For the choice of basis function, we consider Power series, Laguerre polynomial and weighted-Laguerre polynomial. Since the payoff function is quite simple, we do not consider other types of basis functions. Table 7.8 gives a list of the regressor configurations we have tested. Figure 7.3 compares the average price difference when using the above mentioned explanatory variables and basis functions.

Table 7.8: The regressor set configurations used for Bermudan Put in the Black-Scholes world

	Explanatory Variable	Basis Function	Terms of Polynomials
Group 1	Asset Price	Power Series	1 to 8
Group 2	Exercise Value	Power Series	1 to 6
Group 3	European Put Value	Power Series	1 to 6
Group 4	Asset price	Laguerre Poly	1 to 6
Group 5	Asset price	weighted-Laguerre	1 to 8

When using only 1 or 2 terms of polynomials, the combination of co-terminal European Put value plus Power series as well as the combination of underlying asset value plus weighted-Laguerre polynomial outperform the rest. The good performance associated with co-terminal European option value is expected since the Bermudan option price can be decomposed into the value of its European counterpart and a relatively small early-exercise premium. In a sense, the European Put value serves as a good control variate for the Bermudan Put value. When using 4 to 6 terms of polynomials, we observe that the average price difference is positive for all combinations. This indicates that for most of the initial asset prices, LSM price rises above the FD benchmark. Using even more terms of Power series or weighted-Laguerre polynomial causes the price difference to reduce again. The exact reason for this positive difference requires further study. But we conjecture that it has something to do with the finite path convergence property of the LSM method w.r.t. the number of regressors used.

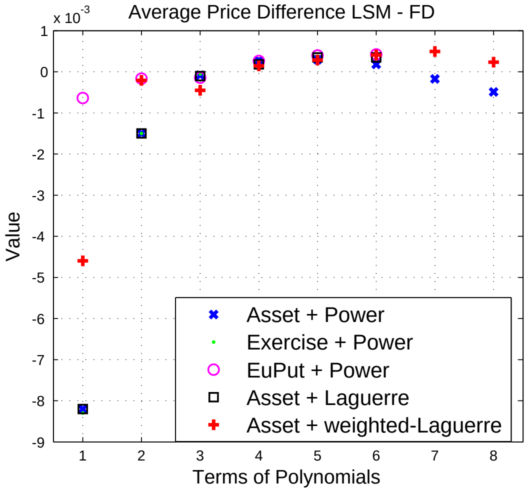


Figure 7.3: Average price difference between LSM and FD when $S(0) \in [6, 14]$ for a Bermudan Put in the Black-Scholes world using various regressor configurations.

7.1.4 Upper Bound Result

We have implemented the upper bound algorithm in Chapter 3 in the setting of the Black-Scholes world. We call the upper bound price the Andersen-Broadie (AB) price. What we try to show with our numerical result here is that the computed gap between an LSM price and its associated upper bound price gives us information about whether a set of regressor performs better than another set.

Since the upper bound algorithm is computationally quite expensive, we choose 12 possible exercise moments for the maturity $T = 1$ in the Black-Scholes world. Notice that in the previous simulation of the LSM price, we choose 52 for $T = 1$. The difference in early exercise moments would cause difference in the option price. As we can see from the benchmark prices in Table 7.1, Table 7.3, and Table 7.9 below, the price difference is observable.

Let us first give some results about convergence of the upper bound price. As a base case, we choose the outer path number $N_{out} = 10^3$, the inner path number $N_{in} = 10^3$. An average of 10 such simulation gives us an estimate of the gap between the LSM price and the upper bound AB price as well as the corresponding 95% t-Confidence Interval (t-CI) of the gap value. We use a constant and the first 3 power terms of the underlying asset value as the regressors. We use 2×10^6 paths to obtain the LSM regression coefficients, then use a separate set of 10^6 paths to obtain the LSM price. These regression coefficients define the exercise rule for the upper bound simulation. Variation in the regression coefficients due to limited number of paths for the LSM algorithm may lead to variation in the upper bound result as well. We choose as many as possible paths for the LSM to minimize such impact. We increase either N_{out} or N_{in} to 10^4 paths to check the variation of the gap value due to upper bound Monte Carlo configuration. Table 7.9 gives the benchmark prices, Table 7.10 till Table 7.12 gives the convergence result of the upper bound algorithm.

Table 7.9: Benchmarks for the Bermudan Put in the Black-Scholes world for $N = 12$, $T = 1$.

Name	Initial Asset Price		
	8(ITM)	10(ATM)	12(OTM)
FD	2.0934	0.9471	0.3923
Binomial	2.0934	0.9471	0.3923

Comparing the numbers in Table 7.10 till Table 7.12, we do not find much difference in variation of the Gap value. Hence choosing $Nout = 10^3$, $N_{in} = 10^3$ is reasonable. The t-CI based on 10 simulations is narrow enough around the mean. There is a big difference in terms of the variation of the Gap value at different moneyness. It seems that the variation is lowest for ITM case, acceptable for ATM case, but excessively high for OTM case. The reason for this is that we only choose ITM paths for regression, hence when the underlying asset starts OTM, a lot of paths would be OTM at different moments before maturity and the exercise rule given by the regression coefficients is basically unable to track the true continuation value closely. So to compare between different sets of regressors in the Black-Scholes world,

Table 7.10: Case 1 of upper bound convergence result for a Bermudan Put in the Black-Scholes world. $N_{out} = 10^3$, $N_{in} = 10^3$, 10 simulation average. Time for a run is about 10 min.

Run Number		Initial Asset Price		
		8(ITM)	10(ATM)	12(OTM)
Run1	LSM price	2.0929	0.9470	0.3922
	Gap	0.0154	0.0158	0.0071
	Gap(t-CI)	(0.0147,0.0160)	(0.0150, 0.0166)	(0.0065, 0.0078)
Run2	LSM price	2.0931	0.9470	0.3923
	Gap	0.0142	0.0140	0.1957
	Gap(t-CI)	(0.0132,0.0152)	(0.0129, 0.0150)	(0.1925, 0.1989)
Run3	LSM price	2.0948	0.9467	0.3927
	Gap	0.0149	0.0140	0.0401
	Gap(t-CI)	(0.0141,0.0157)	(0.0133, 0.0147)	(0.0392, 0.0410)

Table 7.11: Case 2 of upper bound convergence result for a Bermudan Put in the Black-Scholes world. $N_{out} = 10^3$, $N_{in} = 10^4$, 10 simulation average. Time for a rum is about 55 min.

Run Number		Initial Asset Price		
		8(ITM)	10(ATM)	12(OTM)
Run1	LSM price	2.0922	0.9474	0.3914
	Gap	0.0138	0.0144	0.2622
	Gap(t-CI)	(0.0132,0.0144)	(0.0140, 0.0148)	(0.2598, 0.2646)
Run2	LSM price	2.0948	0.9469	0.3925
	Gap	0.0137	0.0157	0.2646
	Gap(t-CI)	(0.0128,0.0146)	(0.0147, 0.0167)	(0.2611, 0.2681)
Run3	LSM price	2.0942	0.9478	0.3928
	Gap	0.0143	0.0125	0.2231
	Gap(t-CI)	(0.0135,0.0152)	(0.0119, 0.0130)	(0.2191, 0.2272)

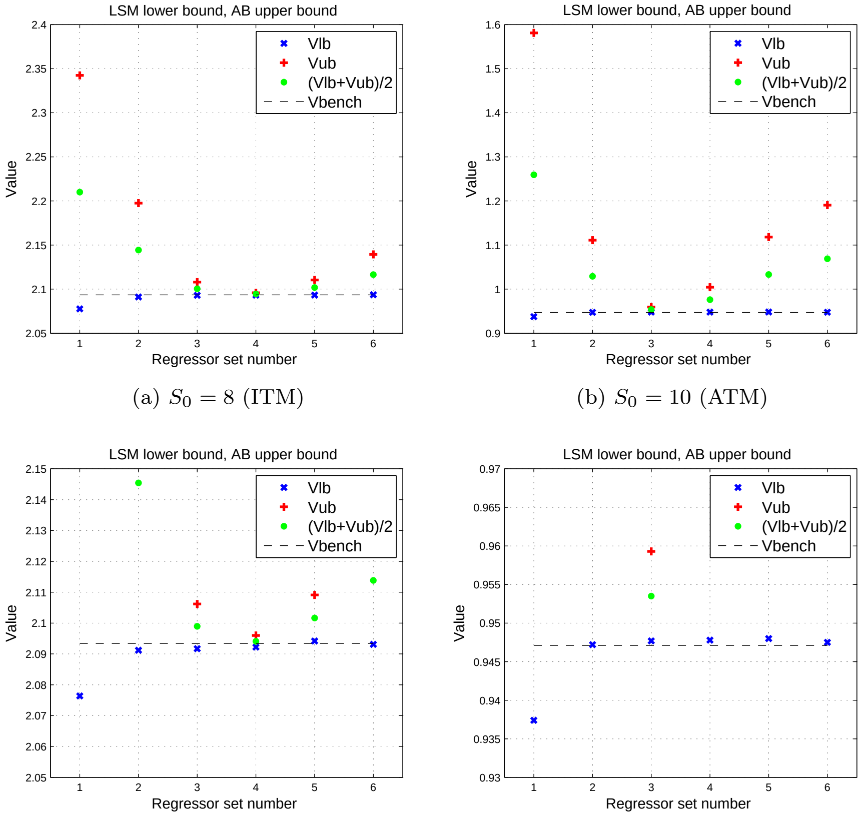
Table 7.12: Case 3 of upper bound convergence result for a Bermudan Put in the Black-Scholes world. $N_{out} = 10^4$, $N_{in} = 10^3$, 10 simulation average. Time for a run is about 125min.

Run Number		Initial Asset Price		
		8(ITM)	10(ATM)	12(OTM)
Run1	LSM price	2.0940	0.9462	0.3921
	Gap	0.0137	0.0098	0.0134
	Gap(t-CI)	(0.0134,0.0139)	(0.0096, 0.0100)	(0.0131, 0.0136)
Run2	LSM price	2.0935	0.9463	0.3924
	Gap	0.0151	0.0128	0.1487
	Gap(t-CI)	(0.0150,0.0153)	(0.0126, 0.0131)	(0.1479, 0.1495)
Run3	LSM price	2.0937	0.9469	0.3918
	Gap	0.0150	0.0139	0.3470
	Gap(t-CI)	(0.0147,0.0154)	(0.0137, 0.0141)	(0.3456, 0.3483)

we will focus on the ITM and ATM gap result. Later, we will see that the OTM problem disappears when we price C(n)LE because all the paths are selected for regression.

Figure 7.4 shows the upper bound simulation results on a Bermudan Put in the Black-Scholes world. All relevant simulation data are listed in Table E.1 to Table E.6 in Appendix E. We choose the underlying asset price as explanatory variable and one to six terms of power series as the basis functions. We will ignore the gap number for OTM case for the reason explained above. The figure shows that the LSM prices are pretty close to the FD benchmark price, although some of them rise slightly above the benchmark. The AB upper bound prices vary much more significantly as we change the number of regressors. The mid point of the lower and upper bound is not a good approximate for the true option value in this case.

Looking at ITM and ATM cases, we have some interesting observations. In both ITM and ATM cases, as the number of basis functions increases the gap value decreases at first and then increases. There is a minimum. For the ITM case, the minimum occurs with 4 terms, and the gap is below 0.2% of our



(a) $S_0 = 8$ (ITM)

(b) $S_0 = 10$ (ATM)

(c) $S_0 = 8$ (ITM) zoomed view

(d) $S_0 = 10$ (ATM) zoomed view

Figure 7.4: Simulated lower bound LSM price and the corresponding upper bound AB price for a Bermudan Put in the Black-Scholes world. The strike is fixed at $K = 10$. Results are simulated with 2 different initial asset prices: $S_0 = 8$, 10.

benchmark price. For the ATM case, the minimum occurs with 3 terms, and the gap is below 2%. Due to different variability at different moneyness, comparison across different moneyness does not make much sense. What appears interesting is that when we choose more than optimal number of basis functions, although the LSM price does not change much, the gap value increases more significantly. The maximum expected shortfall is more sensitive toward different regressor configurations than the LSM price itself. One can say that the relative insensitive LSM price showcases the robustness of the algorithm. But as a measure to compare the relative merit of different regressors, sensitivity of the gap value is a desirable feature.

When the gap value decreases, the LSM price is mostly below our benchmark, but when the gap value increases after it hits the minimum, some of the LSM prices are higher than our benchmark. In Figure 7.2, we have observed that if we keep the same number of sample paths and let the number of basis functions increase, the LSM price raises above our benchmark. Combining the observation from the upper bound result, we think that such increment in price does not imply more accurate pricing. It might be caused by convergence issue, it might be caused by other idiosyncratic numerical error. The exact reason requires further study. But our simulation result shows that one should not blindly follow the rule of thumb criterion for evaluating the LSM price, that is to say, one can not be sure that if an LSM price increases it must be closer to the true option value. Instead, using the upper bound algorithm gives us a much more robust criterion, although the algorithm is quite costly in terms of computation time.

7.2 Bermudan Put Spread in the Black-Scholes World

Unless stated otherwise, we consider the following set of model and simulation parameters:

Parameters for the underlying asset price dynamics under the risk-neutral measure, $r = 0.06$, $\sigma = 0.3$.

We consider two different put spreads,

Spread A: $K_1 = 7, K_2 = 12, Q = 5$.

Spread B: $K_1 = 7, K_2 = 9, Q = 5$.

Parameters for the Bermudan early exercise moments are the same for the two spreads, $T = 1$. Let $\Delta t = \frac{T}{52}$, exercise moments are $\left\{\frac{1}{52}, \frac{2}{52}, \ldots, \frac{51}{52}, 1\right\}$.

Parameters for Monte Carlo simulation, time step is $\Delta t = \frac{T}{52}$, 10^5 number of sample paths for each Monte Carlo (LSM) result. Averaging over 100 such simulation results gives the output. We choose only ITM sample paths at each early exercise moment for regression, i.e. only those paths such that $S(t) < K_2$.

What we would like to find out in this experiment is how the LSM algorithm performs when the payoff function exhibits strong nonlinear relation w.r.t. the underlying. The main difference between a Put and a Put Spread is that the Put Spread payoff function exhibits a bend in the ITM region. The strong nonlinearity in the payoff function happens to be in the region of underlying asset price where the early exercise boundary can be. Studying the Put Spread helps us understand how the error in approximating the continuation value, the error in approximating the true early exercise boundary and the error in the LSM price are related.

7.2.1 Convergence of the Benchmark

As the case with the Bermudan Put, we build two benchmarks for comparison. The first one is based on Binomial Tree. We choose $52 \times 400 = 20800$ time steps. The second one is based on Finite Difference with $\theta-$scheme and PSOR iterative matrix equation solver. We choose $52 \times 400 = 20800$ time steps and 4000 space steps. The Binomial Tree is used to cross validate the FD scheme. Pricing result from the FD scheme is used as the benchmark against which all Monte Carlo simulations are compared. We obtain benchmark prices for the two Put Spreads respectively.

Table 7.13 and Table 7.14 show the convergence for Spread A with Binomial Tree at several initial asset prices. We compare three different number of steps for tree. Table 7.15 and Table 7.16 show the convergence for Spread A with Finite Difference at several initial asset prices. We compare three space step size, Δs. 20800 time steps are used for the FD scheme.

Table 7.17 and Table 7.18 show the convergence for Spread B with Binomial Tree at several initial asset prices. Table 7.21 and Table 7.22 show the convergence for Spread B with Finite Difference at several initial asset prices.

Figure 7.5 shows the difference of pricing results from Binomial Tree and Finite Difference for both Put Spreads. The number shows that the most significant difference lies in the vicinity of $K_1 = 7$, which is less than 1.5×10^{-4} for both spreads. Difference in OTM region is much lower for both Put Spreads.

Table 7.13: Pricing result for Spread A with Binomial Tree

	# Steps	Initial Asset Price				
		6	7	9	11	13
P1	52×4	4.99567	4.88344	3.02286	1.60936	0.79878
P2	52×40	4.99437	4.87501	3.02279	1.60877	0.79839
P3	52×400	4.99424	4.87414	3.02269	1.60858	0.79836

Table 7.14: Convergence for Spread A with Binomial Tree

	Initial Asset Price				
	6	7	9	11	13
P2 − P1	-1.3×10^{-3}	-8.4×10^{-3}	-7.2×10^{-5}	-6.0×10^{-4}	-3.9×10^{-4}
P3 − P2	-1.33×10^{-4}	-8.7×10^{-4}	-9.7×10^{-5}	-1.87×10^{-4}	-2.96×10^{-5}

Table 7.15: Pricing result for Spread A with Finite Difference

	Δs	Initial Asset Price				
		6	7	9	11	13
P1	1	4.99347	4.94954	3.00300	1.59384	0.78592
P2	0.1	4.99422	4.87581	3.02260	1.60843	0.79823
P3	0.01	4.99423	4.87407	3.02269	1.60858	0.79835

Table 7.16: Convergence for Spread A with Finite Difference

	Initial Asset Price				
	6	7	9	11	13
P2 − P1	7.52×10^{-4}	-7.37×10^{-2}	1.96×10^{-2}	1.46×10^{-2}	1.23×10^{-2}
P3 − P2	7.31×10^{-6}	-1.7 ×10^{-3}	8.35×10^{-5}	1.43×10^{-4}	1.24×10^{-4}

Table 7.17: Pricing result for Spread B with Binomial Tree

	# Steps	Initial Asset Price				
		6	7	8	9	11
P1	52×4	4.99567	4.74168	3.25581	2.09834	0.79603
P2	52×40	4.99437	4.72975	3.25533	2.09487	0.79406
P3	52×400	4.99423	4.72981	3.25613	2.09506	0.79378

Table 7.18: Convergence for Spread B with Binomial Tree

	Initial Asset Price				
	6	7	8	9	11
P2 − P1	-1.3×10^{-3}	-1.19×10^{-2}	-4.80×10^{-4}	-3.5 ×10^{-3}	-2×10^{-3}
P3 − P2	-1.39×10^{-4}	5.33×10^{-5}	7.95×10^{-4}	1.89×10^{-4}	-2.8×10^{-4}

Table 7.19: Pricing result for Spread B with Finite Difference

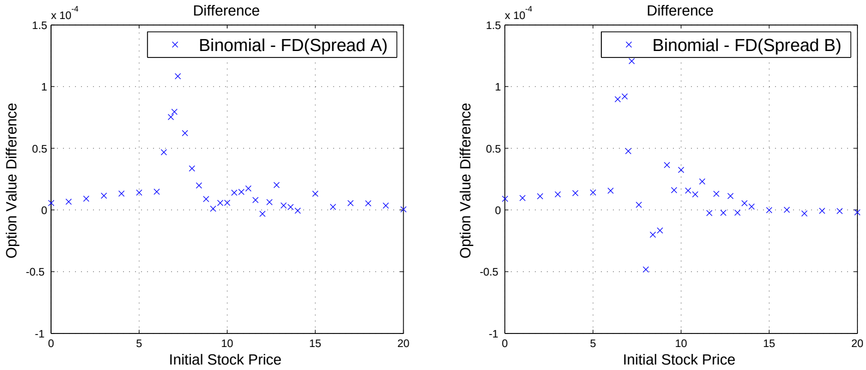
	Δs	Initial Asset Price				
		6	7	8	9	11
P1	1	4.99290	4.91701	3.26701	2.07085	0.76895
P2	0.1	4.99420	4.73451	3.25697	2.09489	0.79347
P3	0.01	4.99422	4.72976	3.25618	2.09502	0.79375

7.2.2 Performance with Different Regressor Configurations

We apply the LSM method for both put spreads, and characterize the performance respectively. We will see that the difference in the slope of the payoff function makes a difference.

Table 7.20: Convergence for Spread B with Finite Difference

	Initial Asset Price				
	6	7	8	9	11
P2 − P1	1.3×10^{-3}	-1.83×10^{-1}	-1.0×10^{-2}	2.4×10^{-2}	2.45×10^{-2}
P3 − P2	1.9×10^{-5}	-4.8×10^{-3}	-8.0×10^{-4}	1.33×10^{-4}	2.86×10^{-4}



(a) Price Difference on Spread A

(b) Price Difference on Spread B

Figure 7.5: Price difference between the Binomial Tree and Finite Difference benchmarks for the two Put Spreads

For the choices of explanatory variable, we consider the following 2:

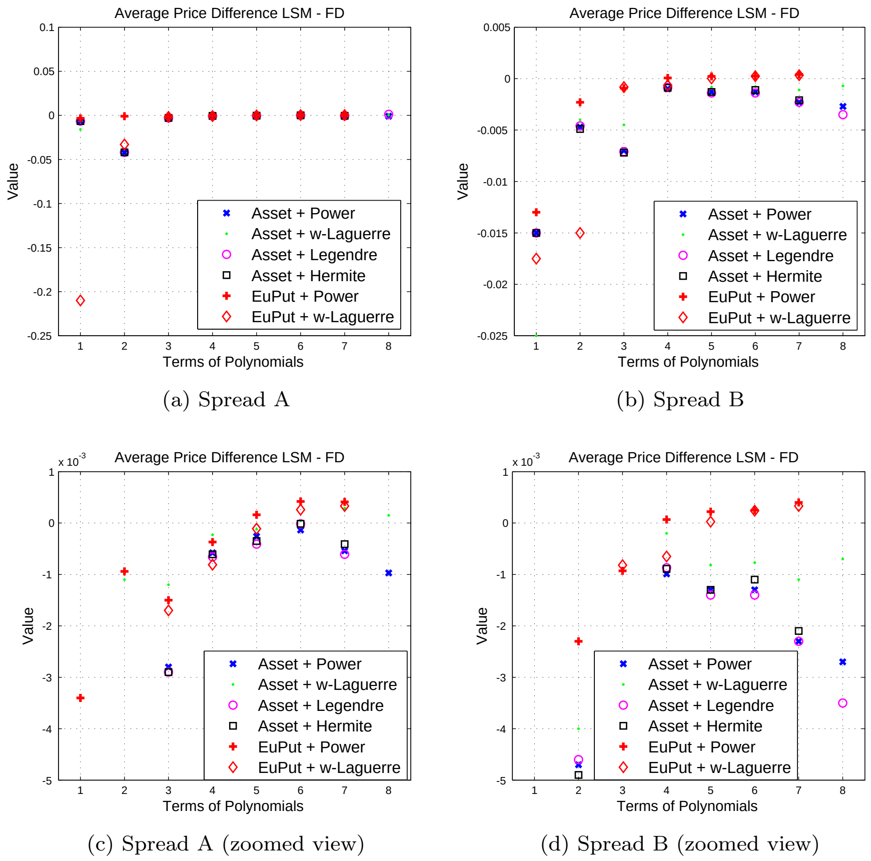
1.the underlying asset value,

2.the value of a co-terminal European put based on the Black-Scholes formula.

For the choice of basis function, since the payoff function of Put Spread has a distinctive bend in ITM region, we consider various types of polynomials, including power series, weighted-Laguerre polynomial, Legendre polynomial and Hermite polynomial.

Figure 7.6 shows the average price difference for $S(0) \in [6, 14]$ for both Put Spreads. Since the two Put Spreads have different payoff functions and hence option values, it does not make sense to compare the average price difference between the two for $S(0) \in [6, 14]$. So, we look at the result separately. Subplot (a) and (c) shows the result with Spread A. (c) shows that the combination of asset price plus weighted-Laguerre polynomial as well as European Put Spread value plus Power series give less negative average price difference before the number becomes positive after using 4 or 5 terms of the polynomials. An interesting case in (a) is that the combination of coterminal European option price and 1 or 2 terms of weighted-Laguerre polynomial produces significantly more negative average difference when compared to other combinations. When we compare this poor result to the good performance of European option price plus Power series, it is not hard to find that a close match of the regressor value to the Bermudan option's exercise or hold value is crucial in determining the performance of the LSM method. Weighted-Laguerre polynomial introduces a scaling factor of $e^{-x/2}$ to bound the polynomials. When x is large, this helps to improve the OLS fitted multi-linear model. On the other hand, when x is small, the scaling factor introduces a nonlinear distortion to the regressor w.r.t. the explanatory variable x. As a result, we see improvement with more terms of weighted-Laguerre polynomials but worse result with one or two terms. As is shown in (b) and (d), we have similar observations for Spread B. Overall, the combination of coterminal European Put value plus Power series as well as the combination of underlying asset price plus weighted-Laguerre polynomial give the best LSM pricing result.

Figure 7.7 compares the price difference (not the average difference) for Spread A and Spread B at different initial asset prices. Underlying asset value is used as explanatory variable, and Power series is used as basis function. Since the payoff function of Spread B lie below that of Spread A uniformly, the option value of Spread B should be lower than that of Spread A. We observe that with 1, 3, 5, 6 terms of Power series, the price difference on Spread B is higher than that on Spread A at most initial asset prices. With 4 terms of power series, the price differences are comparable on the two spreads. It is an anomaly with 2 terms of power series, where the pricing difference on Spread A is much higher. For all six cases, the highest(most negative)price difference of Spread B occurs when $S(0)$ is close to $K_1 = 7$. A discussion about LSM pricing error based on the result in (e) is given in Section 4.2.2. The main conclusion is that



(a) Spread A

(b) Spread B

(c) Spread A (zoomed view)

(d) Spread B (zoomed view)

Figure 7.6: Average price difference between LSM and FD when $S(0) \in [6, 14]$ for the two Put Spreads in the Black-Scholes world using various regressor configurations.

when the OLS fitted multi-linear model approximates the option's continuation value function poorly near the true early exercise boundary, there will be higher LSM pricing error.

7.3 Bermudan with an Asian-style Payoff in the Black-Scholes World

Unless stated otherwise, we consider the following set of model and simulation parameters:

Parameters for the underlying asset price dynamics under risk neutral measure, $r = 0.06$, $\sigma = 0.3$.

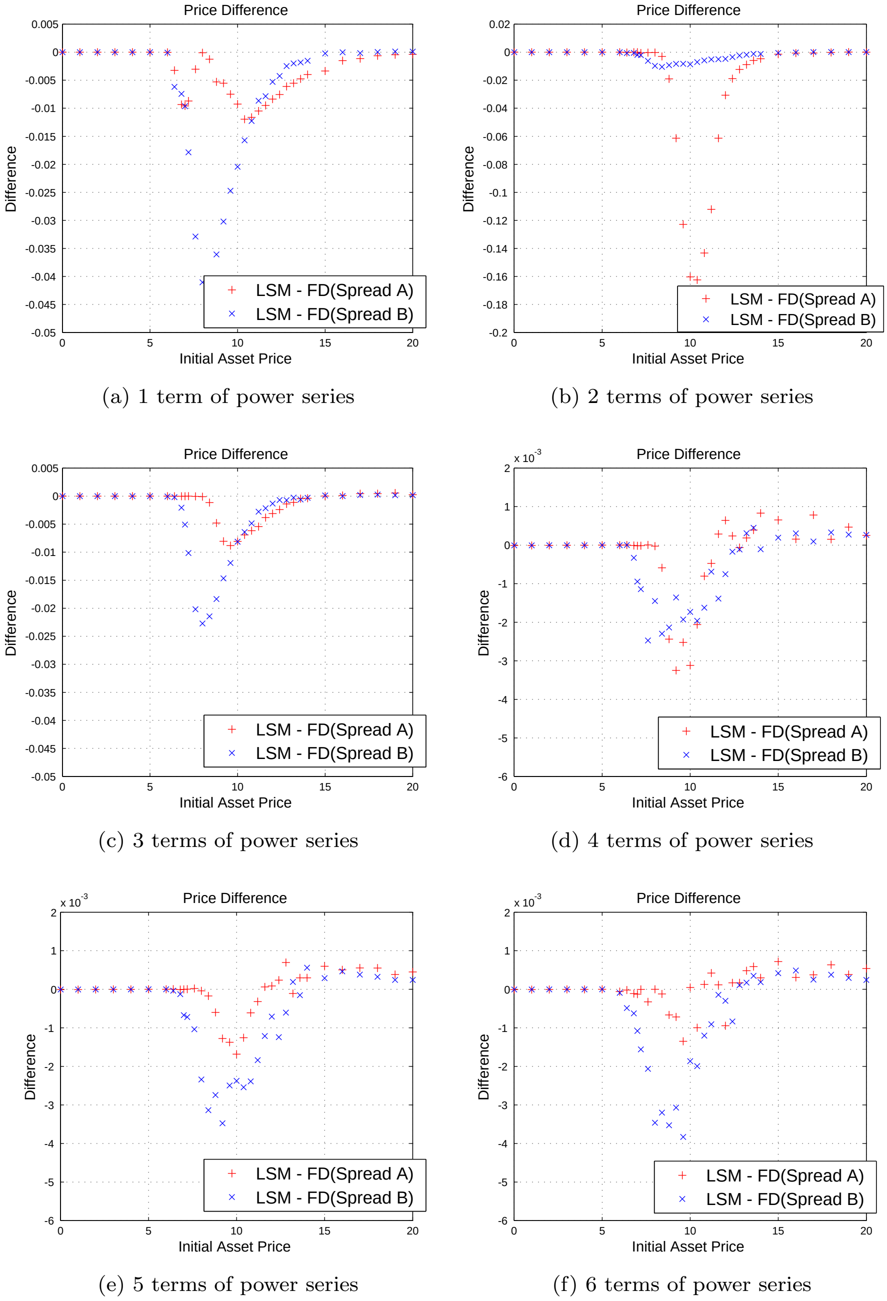
Parameters for the Bermudan put option, $K = 10$, $T = 1$. Let $\Delta t = \frac{T}{52}$, exercise moments are $\left\{\frac{1}{52}, \frac{2}{52}, \ldots, \frac{51}{52}, 1\right\}$.

Parameters for Monte Carlo simulation, time step is $\Delta t = \frac{T}{52}$, 10^5 number of sample paths for each Monte Carlo(LSM) result. Averaging over 100 such simulation results gives the output. We choose only ITM sample paths at each early exercise moment for regression, i.e. only those paths where $A(t) < K$. The Asian-style Bermudan option has a path dependent feature. We would like to understand how to incorporate the path-dependency in the LSM algorithm.

7.3.1 Convergence of the Benchmark

As we have mentioned in Section 4.2.3 , the benchmark FD solver approximates the value of a continuously-monitored arithmetic-mean fixed-strike Bermudan Asian option. It is the closest match to the discretely-monitored Bermudan Asian option that we try to price with the LSM method. We first give some numerical evidence of the convergence of our benchmark price.

We build the FD solver for the Bermudan Asian option. Table 7.21 and 7.22 give the convergence of the Bermudan FD benchmark. For price P3, we use 1000 grid points for asset price S, 1000 grid points for its running average A and 2000 grid points for the time axis. With the strike at $K = 10$, we are mainly interested in option price on the interval $S(0) \in [6, 12]$. The number shows that depending on the moneyness, convergence is about to a few percentage of the price fir P3. This is not so satisfactory



(a) 1 term of power series

(b) 2 terms of power series

(c) 3 terms of power series

(d) 4 terms of power series

(e) 5 terms of power series

(f) 6 terms of power series

Figure 7.7: Price difference for two Bermudan Put Spreads in the Black-Scholes world, $T = 1$, $\Delta t = T/52$, underlying asset value as explanatory variable, power series as basis functions. Simulated with 10^5 sample paths. Spread A, $K_1 = 7, K_2 = 12$. Spread B, $K_1 = 7$, $K_2 = 9$.

from pure numerical point of view. However, despite the effort of using ADI to enhance the numerical efficiency, the PSOR algorithm does not work very efficiently in this case. It takes long time to obtain the price P3. One may need to try some more efficient numerical scheme to build better FD solver for the Bermudan Asian option. However, we shall learn from the next section that this benchmark allows us to gauge the relative performance of the LSM pricing result.

Table 7.21: Pricing result for a Bermudan Asian Put with Finite Difference

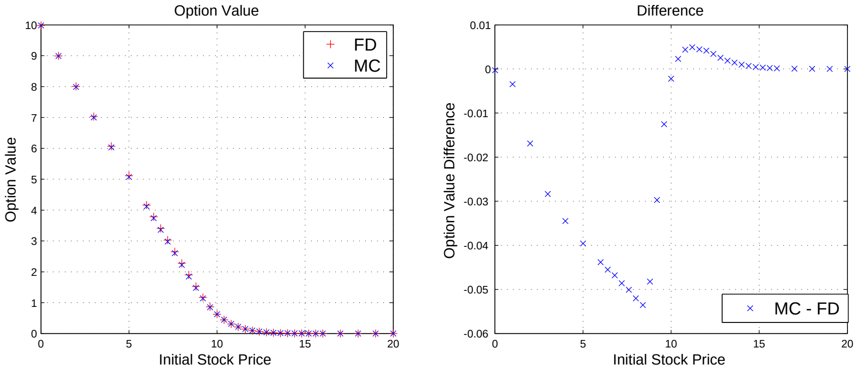
	Δs	Initial Asset Price				
		6	8	10	12	14
P1	2	4.0556	2.2641	0.5636	0.1292	0.0290
P2	0.2	4.1817	2.2967	0.6444	0.1039	0.0123
P3	0.02	4.1612	2.2760	0.6264	0.0965	0.0099

Table 7.22: Convergence for a Bermudan Asian Put with Finite Difference

	Initial Asset Price				
	6	8	10	12	14
P2 − P1	0.1261	0.0327	0.0808	-0.0253	-0.0168
P3 − P2	-0.0205	-0.0207	-0.0179	-0.0075	-0.0023

7.3.2 Performance with Different Regressor Configurations

We have simulated the LSM method with different regressor configurations. Table 7.23 shows the price difference between the LSM price and the FD benchmark. The numbers are higher than those for the Bermudan Put and Put Spread, mainly because the difference in payoff functions for LSM simulation and the FD benchmark. Figure 7.8 illustrates the pricing difference at different initial asset prices. Nevertheless we can still observe change in the price difference numbers when different regressor configurations are used. We do not record the most extreme price difference points, instead, we track the pricing difference at several representative initial asset price points.



(a) Pricing Result

(b) Price Difference

Figure 7.8: Pricing results and difference between the LSM method on a discretely-monitored Bermudan Asian option and the FD scheme on its continuously-monitored counterpart.

The numbers show that using the underlying price S and its running average A together as the explanatory variables helps to reduce the price difference significantly. This is expected, because one of the underlying assumptions of the LSM algorithm is that assume the underlying information process to be a Markov process. When the assumption is satisfied the continuation value function can be expressed as a one step conditional expectation conditioned on the information available up to that moment. In case of this Bermudan Asian option, if we only take S as explanatory variable, then the path-dependent feature means that the continuation value function also depends on S value at earlier moments, which means it is non-Markovian. However, if we augment the explanatory variable by A, then the continuation value

Table 7.23: Price difference for a Bermudan Asian Put with various regressor configurations, $T = 1$, $\Delta t = T/52$, 10^5 number of paths.

Regressors	Average Diff for $S(0) \in [6, 12]$	Initial Asset Price				
		6	8	10	12	14
$c, A.$	-0.1545	-0.16460	-0.27228	-0.09143	-0.00110	0.00085
$c, A, A^2.$	-0.1516	-0.15394	-0.25689	-0.09114	-0.00104	0.00076
$c, A, \ldots, A^3.$	-0.1526	-0.15780	-0.26322	-0.09134	-0.00101	0.00078
$c, A, \ldots, A^4.$	-0.1514	-0.16393	-0.26821	-0.09103	-0.00094	0.00089
$c, S.$	-0.0888	-0.08993	-0.13801	-0.07606	-0.01112	-0.00080
$c, S, S^2.$	-0.0864	-0.08740	-0.13107	-0.07590	-0.01162	-0.00086
$c, S, \ldots, S^3.$	-0.0857	-0.08730	-0.12978	-0.07541	-0.01163	-0.00088
$c, S, \ldots, S^4.$	-0.0857	-0.08750	-0.12952	-0.07552	-0.01162	-0.00086
$c, A, S.$	-0.0280	-0.04715	-0.05457	-0.00481	0.00375	0.00096
$c, A, A^2, S, S^2.$	-0.0258	-0.04381	-0.05202	-0.00222	0.00414	0.00099
$c, A, \ldots, A^3,$ $S, \ldots, S^3.$	-0.0253	-0.04406	-0.05189	-0.00159	0.00393	0.00100
$c, A, \ldots, A^4,$ $S, \ldots, S^4.$	-0.0252	-0.04404	-0.05177	-0.00125	0.00411	0.00099
c, A, A^2,$S, S^2,$ $S \times A.$	-0.0238	-0.04321	-0.04948	0.00038	0.00423	0.00101
$c, A, \ldots, A^3,$ $S, \ldots, S^3,$ $S \times A.$	-0.0233	-0.04246	-0.04942	0.00108	0.00418	0.00099
c, A, A^2,$S, S^2,$ $S \times A$, $S \times A^2,$ $S^2 \times A.$	-0.0232	-0.04266	-0.04932	0.00124	0.00418	0.00102
$c, A, \ldots, A^3,$ $A, \ldots, A^3,$ $S \times A$,$S \times A^2,$ $S^2 \times A.$	-0.0233	-0.04251	-0.04929	0.00089	0.00420	0.00096

function at time t indeed only depends on the time t values of $S(t)$ and $A(t)$. Thus the LSM simulation using both S and A is theoretically justified. The result shows that adding higher order terms such as S^3 or A^3 does not bring much improvement. Adding the cross-product $S \times A$ increases the pricing result slightly further, although not as significant as adding A. Adding higher order cross-product terms such as $S^2 \times A$ or $S \times A^2$ does not make much difference.

We have also tried to use the Black-Scholes price of a co-terminal European-Asian continuously-monitored geometric-mean fixed-strike put as an explanatory variable. The results are shown in Table 7.24. The numbers show that using this variable does not render better pricing result.

Table 7.24: Price difference for a Bermudan Asian Put with various regressor configurations, $T = 1$, $\Delta t = T/52$, 10^5 number of paths. The BS value of a co-terminal geometric-mean European Asian put, V_g, is used as explanatory variable.

Regressors	Average Diff for $S(0) \in [6, 12]$	Initial Asset Price				
		6	8	10	12	14
c, V_g, V_g^2.	-0.0870	-0.08830	-0.13115	-0.07795	-0.01218	0.00095
$c, V_g, \dots, V_g^3$.	-0.0860	-0.08770	-0.12934	-0.07623	-0.01187	0.00092
c, A, A^2, V_g, V_g^2.	-0.0260	-0.04376	-0.05168	-0.00321	-0.00347	-0.00084
$c, A, \dots, A^3, V_g, \dots, V_g^3$.	-0.0254	-0.04397	-0.05178	-0.00189	-0.00363	-0.00083
$c, A, A^2, V_g, V_g^2, V_g \times A$.	-0.0241	-0.04257	-0.04988	-0.00042	-0.00408	-0.00093
$c, A, \dots, A^3, V_g, \dots, V_g^3, V_g \times A, V_g \times A^2, V_g^2 \times A$.	-0.0233	-0.04248	-0.04948	-0.00105	-0.00407	-0.00089

7.4 Bermudan Put in the Heston Model

Unless stated otherwise, we consider the following set of model and simulation parameters:

Parameters for the underlying asset price dynamics under the risk neutral measure:

We consider Heston model with 2 different correlation coefficients.

The one with $\rho = -0.6$ is called Heston A.

The one with $\rho = 0$ is called Heston B.

All the other model parameters are the same, $r = 0.03$, $\sigma_v = 0.3$, $\kappa = 2$, $\eta = 0.1$, $v(0) = \eta$, $S(0) = 10$.

Parameters for the Bermudan put option, $T = 1$. Let $\Delta t = \frac{T}{52}$, exercise moments are $\left\{\frac{1}{52}, \frac{2}{52}, \dots, \frac{51}{52}, 1\right\}$. We evaluate the option price at different strike, K.

Parameters for Monte Carlo simulation, time step is $\Delta t = \frac{T}{52}$, 10^5 number of sample paths for each Monte Carlo(LSM) result. Averaging over 100 such simulation results gives the output. We choose only ITM sample paths at each early exercise moment for regression, i.e. only those paths where $S(t) < K$. For each set of regressors, we generate sample paths from $S(0) = 10$, and simulate by the LSM method option values with different strike K.

In this experiment, we would like to find out how to utilize the stochastic volatility (or variance) variable in the LSM algorithm.

7.4.1 Convergence of the Benchmark

For the Heston model, we use a benchmark based on the COS method [23]. In the algorithm, the one parameter that affects convergence the most is the number of COS terms. We choose all other parameters according to the recommendation in Fang's work [23]. Option values calculated with different number of COS terms are compared to establish convergence. Table 7.25 and 7.26 shows the convergence of the benchmark for Heston A, Table 7.27 and 7.28 shows the convergence of the benchmark for Heston B.

Table 7.25: Pricing result for Heston A with COS

	# COS terms	Strike				
		6	8	10	12	14
P1	32	0.07377	0.37502	1.11034	2.35698	4.05131
P2	62	0.07277	0.37166	1.10414	2.34965	4.03393
P3	128	0.07268	0.37154	1.10376	2.34865	4.03253
P4	256	0.07268	0.37154	1.10376	2.34863	4.03256

Table 7.26: Convergence for Heston A with COS

	Strike				
	6	8	10	12	14
P2 − P1	-1.0×10^{-3}	-3.4×10^{-3}	-6.2×10^{-3}	-7.3×10^{-3}	-1.74×10^{-2}
P3 − P2	-9.2×10^{-5}	-1.2×10^{-4}	-3.8×10^{-4}	1.3×10^{-3}	-1.4×10^{-3}
P4 − P3	-2.9×10^{-6}	-3.9×10^{-6}	-5.1×10^{-6}	-1.9×10^{-5}	-2.8×10^{-5}

Table 7.27: Pricing result for Heston B with COS

	# COS terms	Strike				
		6	8	10	12	14
P1	32	0.04834	0.33838	1.11890	2.41504	4.09708
P2	62	0.04738	0.33517	1.11054	2.40722	4.08645
P3	128	0.04733	0.33483	1.10989	2.40652	4.08630
P4	256	0.04733	0.33483	1.10988	2.40652	4.08629

Table 7.28: Convergence for Heston B with COS

	Strike				
	6	8	10	12	14
P2 − P1	-9.6×10^{-4}	-3.2×10^{-3}	-8.4×10^{-3}	-7.8×10^{-3}	-1.06×10^{-2}
P3 − P2	-5.5×10^{-5}	-3.4×10^{-4}	-6.5×10^{-4}	7.0×10^{-4}	-1.6×10^{-4}
P4 − P3	-2.4×10^{-7}	-2.7×10^{-6}	-6.3×10^{-6}	-4.8×10^{-6}	-8.4×10^{-6}

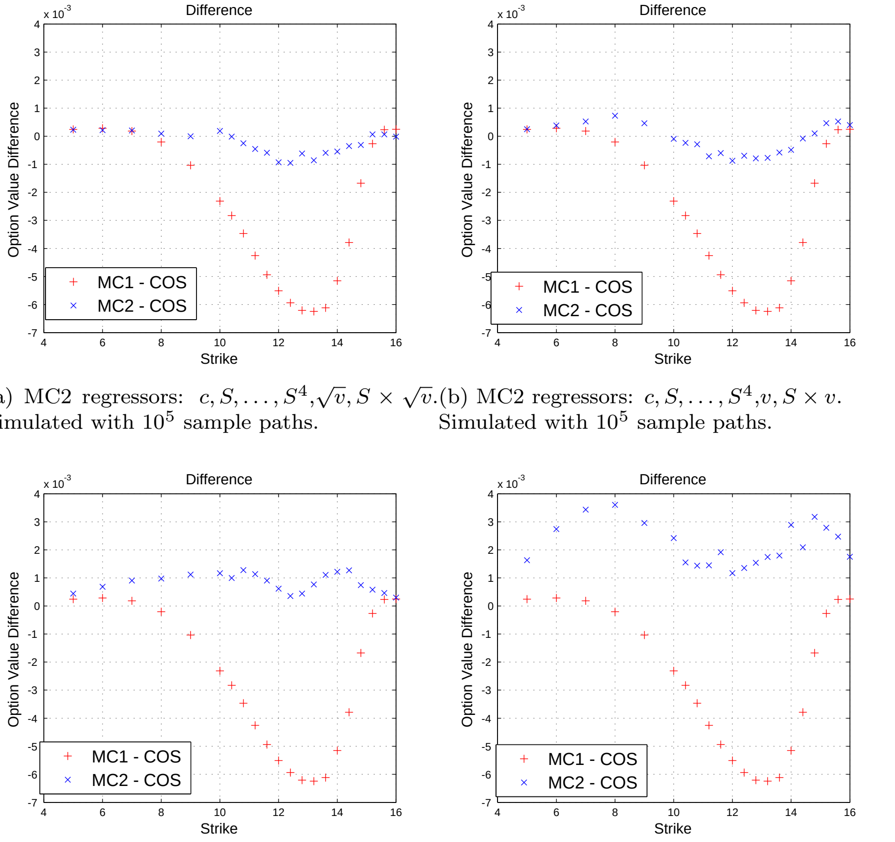
7.4.2 Performance with Different Regressor Configurations

We simulate the LSM method with different regressor configurations. We distinguish between two cases, one with the correlation coefficient $\rho = -0.6$ and the other with $\rho = 0$.

Table 7.29 to 7.31 have the results where $\rho = -0.6$. Table 7.29 shows the price difference with the LSM price simulated from 10^4 sample paths. The figures show that as the number of basis functions increases, the LSM price rises above the COS benchmark for all strike value. We have observed similar phenomenon in the Black-Scholes world. The message here is that we should be more careful to interpret the results.

Despite this ambiguity, we do observe evidence that some configurations of regressors make a difference. Table 7.30 shows that using the underlying asset price S alone does not produce better result regardless how many terms of basis functions to choose. In Table 7.31 we show the price difference with various regressor configurations. Regressors like $c, S, \ldots, S^4, \sqrt{v}, S \times \sqrt{v}$ and $c, S, \ldots, S^4, v, S \times v$ reduce the price difference significantly while keeping the LSM price below the COS benchmark. Adding more terms of cross-product such as $S \times v$ or $S^2 \times \sqrt{v}$ further raises the LSM price, but whether this means better accuracy is still not clear at this point. We will get back to this in the upper bound simulation later. Since in these simulations early exercise is possible at each time step except time 0, running sum of the stochastic variance, $\Sigma_{i=1}^n v(t_n)$, does not contribute to better pricing. Figure 7.9 shows the price differences between the LSM price and the benchmark with several different regressor constructions and simulation setups. In all four subplots, MC1 is simulated with the regressors, $c, S, \ldots, S^4$ and 10^5 paths, which is plotted to illustrate the pricing error profile of low biased LSM results. 4 different error plot with different regressors or simulation configuration are labeled MC2. These plots illustrates the problem that the LSM price would raise above the benchmark under various situations. Subplot (c) shows that adding too many terms of regressors could be a problem. Subplot (d) shows that using low path number can lead to significant high bias.

One question to be answered is whether the significant improvement when adding regressors such as $S \times \sqrt{v}$ or $S \times v$ is related to the negative correlation coefficient $\rho = -0.6$. We thus make a few similar simulations with ρ set to be 0. Table 7.32 shows the results. We again observe that regressors like $c, S, \ldots, S^4, \sqrt{v}, S \times \sqrt{v}$ and $c, S, \ldots, S^4, v, S \times v$ reduce the price difference significantly while keeping the LSM price below the COS benchmark. This illustrates that the improvement is not related to the correlation coefficient. Adding more cross-product terms further raises the LSM price, but the implication requires further study.



(a) MC2 regressors: $c, S, \ldots, S^4$,$\sqrt{v}$, $S \times \sqrt{v}$. Simulated with 10^5 sample paths.

(b) MC2 regressors: $c, S, \ldots, S^4$,v, $S \times v$. Simulated with 10^5 sample paths.

(c) MC2 regressors: $c, S, \ldots, S^4$,$\sqrt{v}, v$, $S \times \sqrt{v}, S \times v, S^2 \times \sqrt{v}$. Simulated with 10^5 sample paths.

(d) MC2 regressors: $c, S, \ldots, S^4$. Simulated with 10^4 sample paths

Figure 7.9: Price difference for a Bermudan Put in the Heston model, $T = 1$, $\Delta t = T/52$, different regressor constructions. All results except MC2 in (d) are simulated with 10^5 sample paths. All the MC1 are simulated with regressors: $c, S, \ldots, S^4$.

Table 7.29: Price difference for a Bermudan Put in the Heston model with underlying price as explanatory variable and Power series as basis functions , $T = 1$, $\Delta t = T/52$, 10^4 number of paths. Correlation coefficient $\rho = -0.6$.

Polynomial	Term Number	Average Diff $S(0) \in [6, 14]$	Most Positive Diff	Most Negative Diff
Power	3	-3.6×10^{-4}	$2.0{\times}10^{-3}$@15.2	-$2.5{\times}10^{-3}$@13.2
Power	4	2.1×10^{-3}	$3.6{\times}10^{-3}$@8	NA
Power	5	2.7×10^{-3}	$6.0{\times}10^{-3}$@9	NA
Power	6	4.2×10^{-3}	$6.6{\times}10^{-3}$@9	NA

Table 7.30: Price difference for a Bermudan Put in the Heston model with underlying price as explanatory variable and Power series as basis function, $T = 1$, $\Delta t = T/52$, 10^5 number of paths. Correlation coefficient $\rho = -0.6$.

Polynomial	Term Number	Average Diff $K \in [8, 16]$	Most Positive Diff	Most Negative Diff
Power	1	-1.4×10^{-2}	2.8×10^{-5}@4	-2.5×10^{-2}@12
Power	2	-6.4×10^{-3}	1.2×10^{-4}@16	-1.15×10^{-2}@12.8
Power	3	-3.9×10^{-3}	2.3×10^{-4}@16	-7.4×10^{-3}@13.2
Power	4	-3.3×10^{-3}	2.8×10^{-4}@6	-6.2×10^{-3}@13.2
Power	5	-3.3×10^{-3}	5.1×10^{-4}@7	-6.2×10^{-3}@13.2
Power	6	-2.9×10^{-3}	5.7×10^{-4}@7	-5.9×10^{-3}@13.6
Power	7	-3.0×10^{-3}	5.9×10^{-4}@6	-6.6×10^{-3}@12.4
Power	8	-3.6×10^{-3}	3.0×10^{-4}@8	-8.7×10^{-3}@12.8

Table 7.31: Price difference for a Bermudan Put in the Heston model with various regressor configurations, using both underlying price and the stochastic variance as explanatory variables. $T = 1$, $\Delta t = T/52$, 10^5 number of paths. Correlation coefficient $\rho = -0.6$. c means a constant.

Regressors	Average Diff $K \in [8, 16]$	Most Positive Diff	Most Negative Diff
$c, S, \ldots, S^4, \sqrt{v}.$	-3.6×10^{-3}	2.2×10^{-4}@15.6	-7.0×10^{-3}@12.4
$c, S, \ldots, S^4, \sqrt{v}, v.$	-2.6×10^{-3}	4.0×10^{-4}@15.2	-5.4×10^{-3}@12.4
$c, S, \ldots, S^4, S \times \sqrt{v}.$	-1.9×10^{-3}	4.7×10^{-4}@6	-3.7×10^{-3}@12.8
$c, S, \ldots, S^4, \sqrt{v}, S \times \sqrt{v}.$	-3.4×10^{-4}	2.3×10^{-4}@5	-9.5×10^{-4}@12.4
$c, S, \ldots, S^4, \sqrt{v}, S \times \sqrt{v}, S \times v.$	-2.0×10^{-4}	5.8×10^{-4}@7	-1.4×10^{-3}@12.8
$c, S, \ldots, S^4, \sqrt{v}, v, S \times \sqrt{v}, S \times v.$	3.8×10^{-4}	8.5×10^{-4}@15.2	-9.1×10^{-5}@13.6
$c, S, \ldots, S^4, \sqrt{v}, v, S \times \sqrt{v},\ S \times v, S^2 \times \sqrt{v}.$	8.6×10^{-4}	1.3×10^{-3}@10.8	NA
$c, S, \sqrt{v}.$	-1.2×10^{-2}	4.4×10^{-5}@4	-2.2×10^{-2}@12
$c, S, S \times \sqrt{v}.$	-1.26×10^{-2}	1.2×10^{-4}@16	-2.3×10^{-2}@12
$c, S, \sqrt{v}, S \times \sqrt{v}.$	-1.15×10^{-2}	7.5×10^{-5}@4	-2.1×10^{-2}@12.4
$c, S, \ldots, S^3, v.$	-3.5×10^{-3}	2.1×10^{-4}@6	-6.7×10^{-3}@12
$c, S, \ldots, S^4, v.$	-3.1×10^{-3}	3.6×10^{-4}@15.2	-6.0×10^{-3}@12.4
$c, S, \ldots, S^5, v.$	-3.7×10^{-3}	2.8×10^{-4}@15.6	-7.6×10^{-3}@12.4
$c, S, \ldots, S^6, v.$	-3.7×10^{-3}	4.1×10^{-4}@10.8	-9.2×10^{-3}@11.6
$c, S, \ldots, S^4, S \times v.$	-3.7×10^{-3}	3.3×10^{-4}@6	-7.5×10^{-3}@12.4
$c, S, \ldots, S^4, v, S \times v.$	-2.0×10^{-4}	7.3×10^{-4}@8	-8.8×10^{-4}@12
$c, S, \ldots, S^4, \Sigma v.$	-4.4×10^{-3}	2.6×10^{-4}@5	-8.5×10^{-3}@12
$c, S, \ldots, S^4, S \times \Sigma v.$	-4.3×10^{-3}	2.6×10^{-4}@6	-8.7×10^{-3}@12
$c, S, \ldots, S^4, \Sigma v, S \times \Sigma v.$	-2.1×10^{-3}	4.0×10^{-4}@6	-4.0×10^{-3}@12.8

Table 7.32: Price difference for a Bermudan Put in the Heston model with various regressor configurations, using both underlying price and the stochastic variance. $T = 1$, $\Delta t = T/52$, 10^5 number of paths. Correlation coefficient $\rho = 0$. c means a constant.

Regressors	Average Diff $K \in [8, 16]$	Most Positive Diff	Most Negative Diff
$c, S, \dots, S^3$.	-6.5×10^{-3}	1.1×10^{-4}@5	-1.0×10^{-2}@13.2
$c, S, \dots, S^4$.	-5.7×10^{-3}	1.9×10^{-4}@6	-9.1×10^{-3}@13.6
$c, S, \dots, S^5$.	-5.6×10^{-3}	2.1×10^{-4}@6	-8.8×10^{-3}@13.6
$c, S, \dots, S^6$.	-5.5×10^{-3}	4.1×10^{-4}@6	-8.8×10^{-3}@14
$c, S, \dots, S^7$.	-5.8×10^{-3}	2.7×10^{-4}@6	-8.8×10^{-3}@12.8
$c, S, \dots, S^8$.	-6.5×10^{-3}	1.8×10^{-4}@5	-1.0×10^{-3}@13.6
$c, S, \dots, S^4, \sqrt{v}$.	-5.3×10^{-3}	1.6×10^{-4}@5	-8.5×10^{-3}@12.4
$c, S, \dots, S^4, \sqrt{v}, v$.	-4.9×10^{-3}	1.7×10^{-4}@5	-7.9×10^{-3}@12.4
$c, S, \dots, S^4, \sqrt{v}, S \times \sqrt{v}$.	-8.2×10^{-4}	5.7×10^{-4}@7	-1.6×10^{-3}@13.6
$c, S, \dots, S^4, v, S \times v$.	-3.6×10^{-4}	5.0×10^{-4}@8	-9.7×10^{-4}@12
$c, S, \dots, S^4, \sqrt{v}, v, S \times \sqrt{v}$, $S \times v, S^2 \times \sqrt{v}$.	1.2×10^{-3}	1.6×10^{-3}@14.8	NA
$c, S, \sqrt{v}$.	-1.39×10^{-2}	4.3×10^{-5}@4	-2.1×10^{-2}@12.4
$c, S, S \times \sqrt{v}$.	-1.36×10^{-2}	3.4×10^{-5}@4	-2.1×10^{-2}@12.4
$c, S, \sqrt{v}, S \times \sqrt{v}$.	-1.41×10^{-2}	6.9×10^{-5}@5	-2.2×10^{-2}@12.4

7.4.3 Upper Bound Result

We compute the upper bound price only for the case where $\rho = -0.6$. The initial asset price is fixed at $S(0) = 10$. Since the upper bound algorithm is computationally quite expensive, we choose $N_t = 12$ possible exercise moments for the maturity $T = 1$ in the Heston model. Notice that in the previous simulation of the LSM price, we choose $N_t = 52$ for $T = 1$. Difference in the option price can be observed from the benchmark prices in Table 7.25 and Table 7.33.

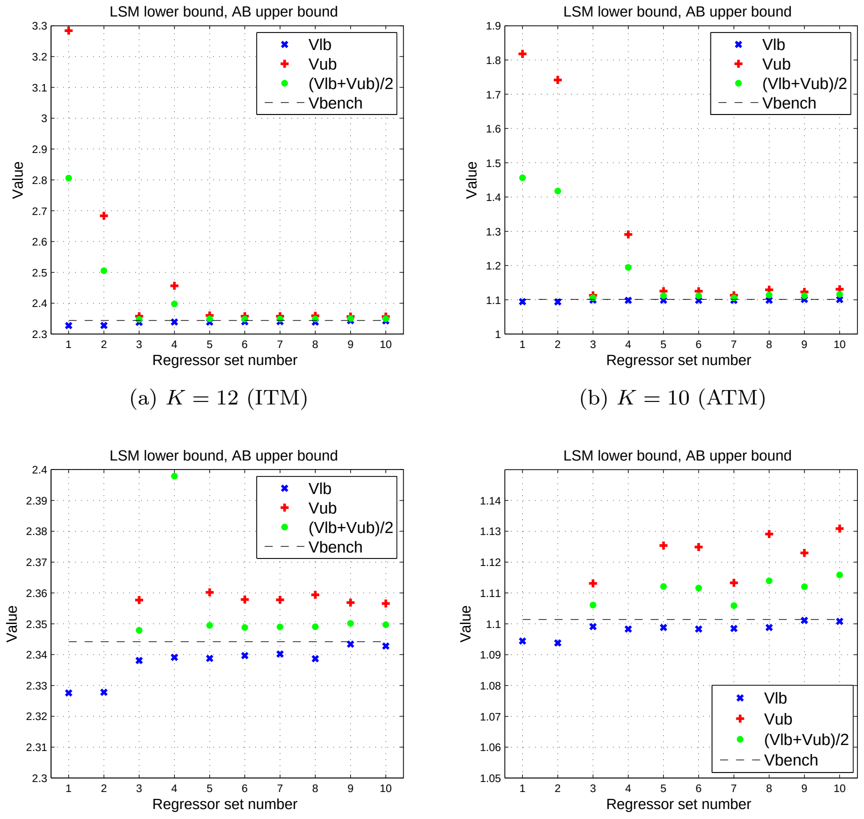
Table 7.33 gives the benchmark prices by COS method. Table E.7 to Table E.9 in Appendix E show the convergence result of the upper bound algorithm. As a base case, we choose the outer path number $N_{out} = 10^3$, the inner path number $N_{in} = 10^3$. Average of 10 such simulation gives us the estimate of the gap between the LSM price and the Upper bound price as well as the corresponding 95% t-Confidence Interval (t-CI) of the gap value. We use the following set of regressors $c, S, \dots, S^4, \sqrt{v}, S \times \sqrt{v}$. We use 10^6 paths to obtain the LSM regression coefficients, then use a separate set of 10^7 paths to obtain the LSM price. These regression coefficients define the stopping rule used in the upper bound simulation. Variation in the regression coefficients due to limited number of paths for the LSM algorithm may lead to variation in the upper bound result as well. We choose as much as possible paths for the LSM to minimize such impact. We increase either N_{out} or N_{in} to 10^4 paths to check the variation of the gap value due to upper bound Monte Carlo configuration.

Figure 7.10 shows the upper bound results for a Bermudan Put in the Heston model. Table 7.34 gives a list of all the 10 sets of regressors used in simulation. The simulation data are recorded in Table E.10 to Table E.19 in Appendix E. We will ignore the gap number for OTM case where the strike $K = 8$.

Table 7.33: Benchmarks for a Bermudan Put in the Heston model for $N = 52$ and $T = 1$.

	Strike		
	8(OTM)	10(ATM)	12(ITM)
COS	0.3707	1.1014	2.3442

We have simulated the upper bound with different set of regressors based on the underlying asset price S and the stochastic variance v. The hand picked "best" set of regressors, Set 9, which consists of a constant c, S, ..., S^4, $\sqrt{v}$ and $S \times \sqrt{v}$ gives a low gap. The resulting gap is below 0.6% in three simulation runs for the ITM case. That is below 3.5% for the ATM case. When choosing less or more terms of the main variable, which is S in this case, the gap increase significantly. Set 1, 2 and 4 uses 1 or 8 power terms of the asset price. Subplot (a) and (b) shows that these regressors give high gap value. The interesting case is Set 4 with 8 power terms of the asset price. The high gap value indicates worse



(a) $K = 12$ (ITM)

(b) $K = 10$ (ATM)

(c) $K = 12$ (ITM) zoomed view

(d) $K = 10$ (ATM) zoomed view

Figure 7.10: Simulated lower bound LSM price and the corresponding upper bound AB price for a Bermudan Put in the Heston model. The initial asset price is fixed at $S(0) = 10$. Results are simulated with 2 different strikes: $K = 12$, 10.

Table 7.34: The regressor set configurations used for a Bermudan Put in the Heston model.

Set Name	Regressor Set Configuration
Set 1	const c, asset price S, square-root of stochastic variance $\sqrt{v}$.
Set 2	$c, S, \sqrt{v}, S \times \sqrt{v}$.
Set 3	$c, S, \ldots, S^4$.
Set 4	$c, S, \ldots, S^8$.
Set 5	$c, S, \ldots, S^4, \sqrt{v}$.
Set 6	$c, S, \ldots, S^4, \sqrt{v}, v$.
Set 7	$c, S, \ldots, S^4, \sqrt{v}, v, v^{\frac{3}{2}}, v^2$.
Set 8	$c, S, \ldots, S^4, S \times \sqrt{v}$.
Set 9	$c, S, \ldots, S^4, \sqrt{v}, S \times \sqrt{v}$.
Set 10	$c, S, \ldots, S^4, \sqrt{v}, v, S \times \sqrt{v}, S \times v, S^2 \times \sqrt{v}$.

approximation due to overfitting caused by using many high order polynomials.

Several other sets of regressors give slightly lower LSM price than Set 9 above, such as c, S, $\ldots$, S^4, $\sqrt{v}$, or c, S, $\ldots$, S^4, $\sqrt{v}$, v. Mainly those with the same number of the main variable S and different number of the secondary variable v or cross-products. These regressor set produce similar upper bound gaps to the "best" set. Subplot (c) and (d) zoom in to give a better view of the gap. Considering the variation of the gap value in separate simulation runs, which can be observed in the data tables in Appendix E, these results are literally indistinguishable. Just using c, S, $\ldots$, S^4 gives a proper gap value. That is less than 1% ITM and less than 2.5% ATM. These results confirms that the LSM price is robust with different choices of the basis function. On the other hand, they shows that due to the variation in the gap value, it is difficult to distinguish between sets of regressors producing very similar pricing results. In Table E.18 and Table E.19, we have the result for Set 9 and Set 10. Results show that with 10^6 sample paths for the LSM algorithm, the additional regressors in Set 10 does not raise the LSM price above our benchmark. Meanwhile, the gap values are quite similar between these two cases.

7.5 Cancellable Bermudan Swaption in the Libor Market Model

We use the model parameters given in Appendix D.2 for all simulations with the Libor market model in this chapter. For each type of coupon specification, we simulate the value of a pair of callable/cancellable Bermudan swaptions according to the algorithms in Section 6.3 at different moneyness. We check the results according to the theoretical parity relation between the simulated option values as suggested by Equation 6.33,

$$V^{CPSN}(0;\theta) = V^{PS}(0;\theta) + V^{CnRSN}(0;\theta).$$

We compute the gap between the LSM price and upper bound price for the cancellable receiver swaptions.

The simulation parameter for the parity result is: 10^6 paths for the LSM regression coefficients and the LSM price.

The simulation parameters for the upper bound result are: 10^6 paths for calculating the LSM regression coefficients, 10^7 paths for calculating the LSM price. 10^3 paths for upper bound outer simulation, 10^3 paths for upper bound inner simulation, 5 upper bound results for calculating average gap value and the 95% t-Confidence Interval. We will report 3 separate simulation runs to give a sense of the overall variability of the gap value. Difference in the results from the 3 runs indicates the impact of the LSM estimated stopping rule on upper bound result.

The first exotic swaption we study is the fixed-for-floating cancellable Bermudan swaption. Its coupon rate is defined by,

$$C_n(L_n(T_{n-1})) = k.$$

where C_n denotes the coupon rate on the interval $[T_{n-1}, T_n]$.

7.5.1 Parity Result

We consider a pair of cancellable receiver swaption and callable payer swaption, which share the same coupon rate and possible exercise moments. We can find the analytical solution of the ATM swaption coupon rate, $k = R_0^N(0)$. We find that $k = 314$bps. The ITM and OTM (with respect to the cancellable receiver swaption) coupon rates are set to be 414bps and 214bps. Table 7.35 gives the parity result.

Table 7.35: Parity value for a callable payer swaption and a cancellable receiver swaption. All the numbers reported are in basis points.

Coupon rate k	Value					
	Simulated $V^{PS}(0)$	Simulated $V^{CPSN}(0)$	Calculated $V^{CnRSN}(0)$	Simulated $V^{PS}(0)$	Simulated $V^{CnRSN}(0)$	Calculated $V^{CPSN}(0)$
214	837.0	901.3	64.3	837.0	64.3	901.3
314	0.3	453.5	453.2	0.3	453.0	453.3
414	-835.2	247.1	1082.3	-835.0	1082.1	247.1

7.5.2 Upper Bound Result

Figure 7.11 shows the simulated lower bound LSM price and the corresponding upper bound Andersen-Broadie(AB) price for the cancellable receiver Bermudan swaption. The middle point between upper bound and lower bound prices is a better estimate of the true option value than either one of them. We simulate the option values at 3 coupon rates mentioned in the last section. We have simulated 10 different sets of regressors, their configurations are listed in Table 7.37. The regressors involved are described in Section 6.3.4. All the numbers reported are in basis points (bps). We show in Figure 7.11 the results from one of three separate simulation runs. Results concerning all 3 runs are listed in Table E.20 till Table E.29 in Appendix E. An example is shown here as Table 7.36. We also report the 95% t-Confidence Interval calculated from 5 upper bound results within a simulation run in Appendix E.

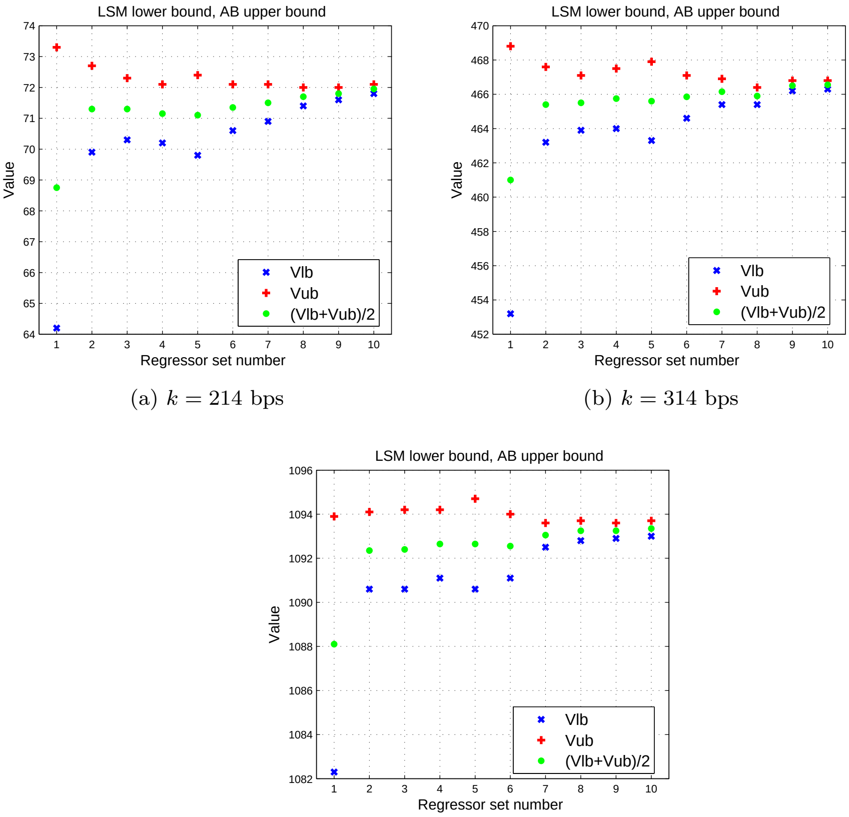
Table 7.36: Upper bound for a cancellable receiver swaption. Regressors: c, fixed leg, floating leg, floating leg tilt. About 13 hours for 3 simulation runs.

Run Number		Coupon rate k		
		214(OTM)	314(ATM)	414(ITM)
Run1	LSM price	64.2	453.2	1082.3
	Gap	9.1	15.6	11.6
	Gap(t-CI)	(8.4, 9.8)	(14.2, 17.1)	(10, 13.2)
Run2	LSM price	64.3	453.1	1082.3
	Gap	9.5	15.8	12.4
	Gap(t-CI)	(8.9, 10)	(15.1, 16.4)	(11.7, 13.1)
Run3	LSM price	64.3	453.2	1082.2
	Gap	9.3	16.5	13
	Gap(t-CI)	(8.8, 9.9)	(14.0, 19.1)	(11.1, 14.9)

Table 7.37: The regressor set configurations used for the cancellable Bermudan swaption in the Libor market model

Set Name	Regressor Set Configuration at T_n
Set 1	const, fixed leg, floating leg, floating leg tilt.
Set 2	const, fixed leg, floating leg, all living Libors.
Set 3	const, fixed leg, floating leg, all living Libors, all living Libors squared.
Set 4	const, fixed leg, floating leg, all living Libors, cross-product of all living Libors.
Set 5	const, fixed leg, floating leg, next Libor rate, forward swap rate $R_n^N(T_n)$.
Set 6	const, fixed leg, floating leg, next Libor rate, $R_n^N(T_n)$, $R_n^N(T_n)^2$.
Set 7	const, fixed leg, floating leg, next Libor rate, $R_n^N(T_n)$, $\ldots$, $R_n^N(T_n)^3$.
Set 8	const, fixed leg, floating leg, next Libor rate, $R_n^N(T_n)$, $\ldots$, $R_n^N(T_n)^4$.
Set 9	const, fixed leg, floating leg, next Libor rate, $R_n^N(T_n)$, $\ldots$, $R_n^N(T_n)^5$.
Set 10	const, fixed leg, floating leg, next Libor rate, $R_n^N(T_n)$, $\ldots$, $R_n^N(T_n)^6$.

Since we choose all the paths for regression, the high gap value caused by picking only ITM paths for regression in the Black-Scholes world and Heston model does not happen here. Even the base case set of regressors, Set 1, can give gap value less than 20 bps across different moneyness. Using all the living Libor rates directly as regressors reduces the gap value to less than 5 bps. Using square or cross-product of the Libor rates gives little improvement. Set 5 to Set 10 features the forward swap rate, $R_n^N(T_n)$. They are able to reduce the gap value further. The gap value is brought down to less than 1 bp with Set 9 and 10, which includes the first 5 and 6 powers of $R_n^N(T_n)$. This term effectively captures the information needed



(a) $k = 214$ bps

(b) $k = 314$ bps

(c) $k = 414$ bps

Figure 7.11: Simulated lower bound LSM price and the corresponding upper bound AB price for a cancellable receiver Bermudan Swaption. Results simulated with 3 different fixed coupon rates: $k =$ 214bps, 314bps, 414bps.

to approximate the continuation value of the fixed-for-floating swaption. We expect that the forward swap rate would also facilitate LSM pricing accuracy for more complicated swaption type, although probably not as effective as what has been demonstrated with the fixed-for-floating swaption.

What is worth mentioning is that we use many terms of regressors in Set 2, 3 and 4. But the resulting gap value appears to be pretty stable. It seems that there is no apparent problem in convergence of the LSM when using many terms of regressors in this setting. Comparing the results associated with Set 5 till Set 10, we observe that as the lower bound LSM price increases, the upper bound reduces as a result of reducing gap value. Considering very similar computational time for all the above mentioned regressor sets, we recommend to use Set 10 for the cancellable fixed-for-floating swaption.

7.6 Cancellable Inverse Floater in the Libor Market Model

We consider a cancellable receiver floored inverse floater with the following coupon function,

$$C_n(L_n(T_{n-1})) = \max(s - L_n(T_{n-1}), f),$$

where s is the strike, f is the floor. Both parameters are set to be time independent constants.

7.6.1 Parity Result

We find the ATM model parameters through a few trail-and-error simulations, and these are $s = 585.2$bps, $f = 20$bps. The parameters makes the simulated swap to worth around ±1bp at time 0. The ITM and OTM (with respect to the cancellable receiver inverse floater) strikes are set to be 685.2bps and 385.2bps. Table 7.38 gives the parity result.

Table 7.38: Parity value for a callable payer floored inverse floater and a cancellable receiver floored inverse floater. All the numbers reported are in basis points.

strike	Value					
	Simulated	Simulated	Calculated	Simulated	Simulated	Calculated
s	$V^{PS}(0)$	$V^{CPSN}(0)$	$V^{CnRSN}(0)$	$V^{PS}(0)$	$V^{CnRSN}(0)$	$V^{CPSN}(0)$
485.2	561.5	832.8	271.3	561.1	270.6	831.7
585.2	1.1	674.3	673.2	1.1	674.1	675.2
685.2	-689.4	533.9	1223.3	-689.4	1223.1	533.7

7.6.2 Upper Bound Result

Figure 7.12 shows the simulated lower bound LSM price and the corresponding upper bound AB price for the cancellable receiver inverse floater. We simulate the option values at 3 strike values mentioned in the last section. We have simulated 10 different sets of regressors, their configurations are listed in Table 7.40. The regressors involved are described in Section 6.3.4. All the numbers reported are in basis points (bps). We show in Figure 7.12 the results from one of the three separate simulation runs. Results concerning all 3 runs are listed in Table E.30 till Table E.39 in Appendix E. An example is shown here as Table 7.39. We also report the 95% t-Confidence Interval calculated from 5 upper bound results within a simulation run in Appendix E.

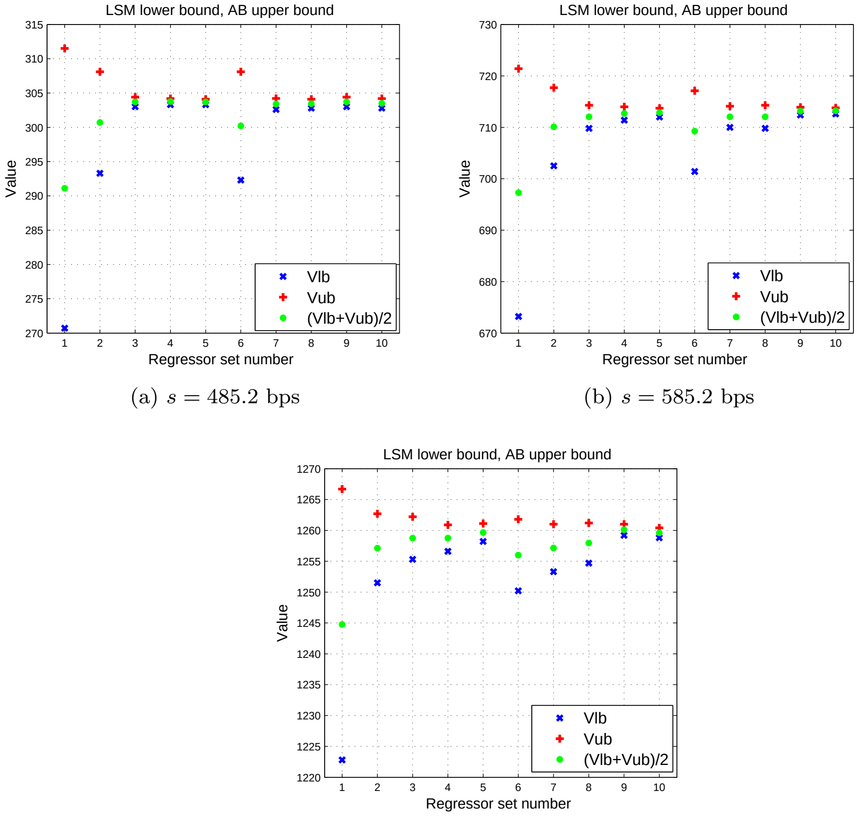
Table 7.39: Upper bound for a cancellable receiver Inverse Floater. Regressors: c, fixed leg, floating leg, floating tilt. About 14 hours for 3 simulation runs.

Run Number		Strike s		
		485.2(OTM)	585.2(ATM)	685.2(ITM)
Run1	LSM price	270.7	673.2	1222.8
	Gap	42.7	48.2	43.9
	Gap(t-CI)	(39.9, 45.4)	(45.3, 51.1)	(40.1, 47.8)
Run2	LSM price	270.8	673.7	1222.4
	Gap	40	49	43.2
	Gap(t-CI)	(38.2, 41.8)	(46.5, 51.5)	(42.3, 44.1)
Run3	LSM price	270.9	673.4	1222.5
	Gap	40.8	47.7	44.8
	Gap(t-CI)	(38.1, 43.5)	(45.0, 50.3)	(42.4, 47.2)

For the floored inverse floater, the base case set of regressors, Set 1, produces a gap value about 48 bps ATM, about twice the number for the ATM Bermudan swaption. The truncation introduced by the floor makes it more difficult for the LSM to approximate the optimal stopping rule, thus resulting in a wider gap. Adding square and cross-product of the floating leg and floating leg tilt to Set 1 gives us Set 2. It reduces the gap to around 15 bps. In this case, adding these terms improves the approximation accuracy.

Table 7.40: The regressor set configurations used for a cancellable inverse floater in the Libor market model

Set Name	Regressor Set Configuration at T_n
Set 1	const, fixed leg, floating leg, floating leg tilt.
Set 2	const, fixed leg, floating leg, floating leg tilt, floating leg squared, floating leg tilt squared, floating leg times floating leg tilt.
Set 3	const, fixed leg, floating leg, all living Libors.
Set 4	const, fixed leg, floating leg, all living Libors, all living Libors squared.
Set 5	const, fixed leg, floating leg, all living Libors, cross-product of all living Libors.
Set 6	const, fixed leg, floating leg, next Libor rate, forward swap rate $R_n^N(T_n)$.
Set 7	const, coupon leg, floating leg, next Libor rate, forward swap rate $R_n^N(T_n)$.
Set 8	const, coupon leg, floating leg, next Libor rate, forward swap rate $R_n^N(T_n)$, $R_n^N(T_n)^2$.
Set 9	const, coupon leg, floating leg, next Libor rate, forward swap rate $R_n^N(T_n)$, coupon leg squared.
Set 10	const, coupon leg, floating leg, next Libor rate, forward swap rate $R_n^N(T_n)$, coupon leg squared, coupon leg third power.



(a) $s = 485.2$ bps

(b) $s = 585.2$ bps

(c) $s = 685.2$ bps

Figure 7.12: Simulated lower bound LSM price and the corresponding upper bound AB price for a cancellable receiver Inverse Floater. Results simulated with 3 different fixed strikes: s = 485.2bps, 585.2bps, 685.2bps.

Using all the living Libors, their squares and cross-products reduces the gap down to less than 2 bps. Set 3, 4 and 5 provide superior result. This rather “crude” approach proves to be quite robust. However, we would like to find out if some more structured regressors can produce similar low gap. Set 6 consists of a const c, forward swap rate $R_n^N(T_n)$, fixed leg, floating leg and the next Libor rate. It gives a gap value similar to that by Set 2. We replace the fixed leg with the coupon leg in Set 7. The gap value immediately drops down to around 4.5 bps. Adding higher order powers of $R_n^N(T_n)$ does not improve the result. Adding higher order powers of the coupon leg further reduces the gap value to less than 2 bps. We understand that the coupon leg contains the most relevant information concerning the payoff of the inverse floater because it takes into account the payoff function. It is not surprising that it proves to be a very effective regressor in this case. Overall, we conclude that Set 5 and 9 are good choices for the cancellable inverse floater.

7.7 Cancellable Snowball in the Libor Market Model

We consider a cancellable receiver Snowball of the inverse floating type. Its coupon rate on the period $[T_n, T_{n+1}]$ is defined by

$$C_n = \max(C_{n-1} + a_n - L_n(T_{n-1}), 0).$$

The spread a_n is defined to be time dependent. A constant virtual coupon rate C_0 is specified to start the Snowball.

7.7.1 Parity Result

The spread a_n is set to increase by time. It is 200bps at T_1 and increases by 20bps at each coupon fixing moment until T_N. The virtual coupon rate C_0 is set to be 300bps for the ATM case. These values are determined through a few trail-and-error simulations. They make the Monte Carlo simulated underlying Snowball swap worth around 7 bps. For ITM and OTM cases, we change C_0 to 400bps and 200bps respectively. Table 7.41 gives the parity result.

Table 7.41: Parity value for a callable payer snowball and a cancellable receiver snowball. All the numbers reported are in basis points.

Virtual	Value					
coupon	Simulated	Simulated	Calculated	Simulated	Simulated	Calculated
C_0	$V^{PS}(0)$	$V^{CPSN}(0)$	$V^{CnRSN}(0)$	$V^{PS}(0)$	$V^{CnRSN}(0)$	$V^{CPSN}(0)$
200	544.0	1339.7	795.7	543.9	795.9	1339.8
300	6.1	1217.0	1210.9	7.0	1209.2	1216.2
400	-593.7	1095.7	1689.4	-593.8	1690.6	1096.8

7.7.2 Upper Bound Result

Figure 7.13 shows the simulated lower bound LSM price and the corresponding upper bound AB price for the cancellable receiver Snowball. We simulate the option values at 3 virtual coupon rates mentioned in the last section. We have simulated 11 different sets of regressors, their configurations are listed in Table 7.43. The regressors involved are described in Section 6.3.4. All the numbers reported are in basis points (bps). We show in Figure 7.13 the results from one of three separate simulation runs. Results concerning all 3 runs are listed in Table E.40 till Table E.50 in Appendix E. An example is shown here as Table 7.42. We also report the 95% t-Confidence Interval calculated from 5 upper bound results within a simulation run in Appendix E.

Table 7.42: Upper bound for a cancellable receiver Snowball. Regressors: c, fixed leg, floating leg, floating leg tilt, next cash flow. About 15 hours for 3 simulation runs.

Run Number		Virtual Coupon Rate C_0		
		200(OTM)	300(ATM)	400(ITM)
Run1	LSM price	795.6	1210.4	1692.0
	Gap	60.2	53.0	51.7
	Gap(t-CI)	(57.5, 62.9)	(50.1, 55.9)	(49.1, 54.2)
Run2	LSM price	795.2	1209.7	1692.6
	Gap	56.3	53.8	51.1
	Gap(t-CI)	(53.9, 58.7)	(49.3, 58.3)	(50.0, 52.2)
Run3	LSM price	795.8	1210.5	1692.1
	Gap	58.4	53.3	52.8
	Gap(t-CI)	(53.9, 63.0)	(47.8, 58.7)	(48.5, 57.2)

For the cancellable Snowball, the base case set of regressors, Set 1, produces a gap value more than 50 bps ATM. Our Snowball payoff function is defined in a path-dependent way. The coupon rates accumulates as time goes on, how fast they accumulate depends on all the relevant Libor rates, $L_n(T_{n-1})$, along the path. Simulation results show that this poses a difficult situation for the LSM with the base

Table 7.43: The regressor set configurations used for the cancellable Snowball in the Libor market model

Set Name	Regressor Set Configuration at T_n
Set 1	const, fixed leg, floating leg, floating leg tilt.
Set 2	const, fixed leg, floating leg, floating leg tilt, floating leg squared, floating leg tilt squared, floating leg times floating leg tilt.
Set 3	const, fixed leg, floating leg, all living Libors.
Set 4	const, fixed leg, floating leg, all living Libors, all living Libors squared.
Set 5	const, fixed leg, floating leg, all living Libors, cross-product of all living Libors.
Set 6	const, fixed leg, floating leg, next Libor rate, forward swap rate $R_n^N(T_n)$.
Set 7	const, coupon leg approximation 1, floating leg, next Libor rate, forward swap rate $R_n^N(T_n)$.
Set 8	const, coupon leg approximation 1, floating leg, next Libor rate, forward swap rate $R_n^N(T_n)$, coupon leg approximation 1 squared.
Set 9	const, coupon leg approximation 2, floating leg, next Libor rate, forward swap rate $R_n^N(T_n)$.
Set 10	const, coupon leg approximation 2, floating leg, next Libor rate, forward swap rate $R_n^N(T_n)$, coupon leg approximation 2 squared.
Set 11	const, coupon leg approximation 2, floating leg, next Libor rate, forward swap rate $R_n^N(T_n)$, coupon leg approximation 2 squared and to the third power.

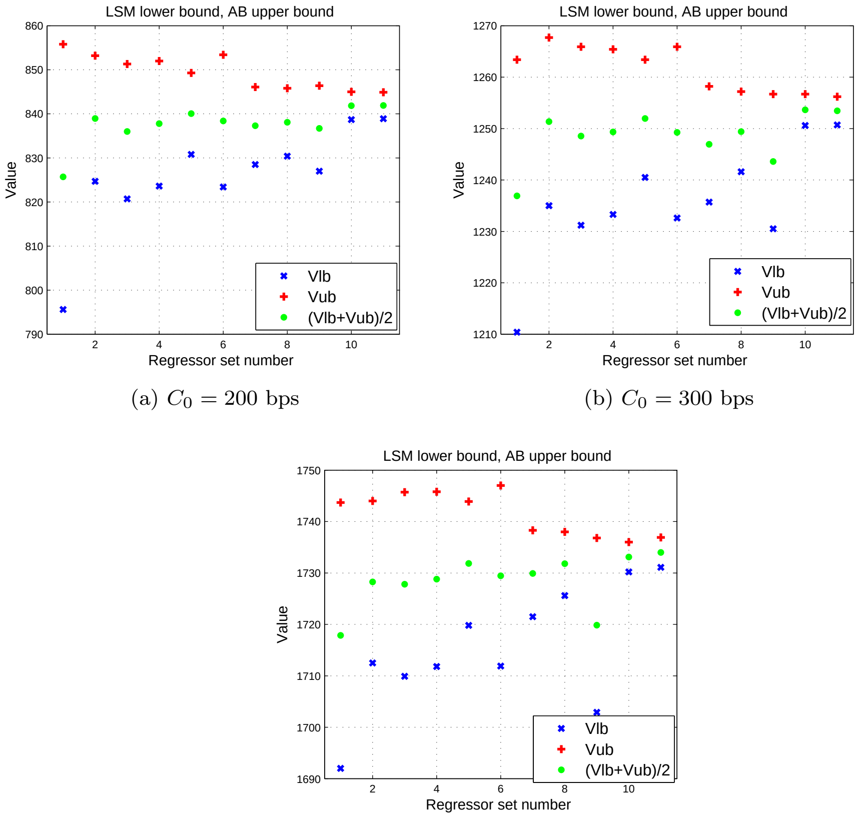
case set of regressors. Set 2 performs surprisingly well even when compared to other better performing sets of regressors. Set 2 consists of all the regressors in Set 1 plus the squares and cross-product of the floating leg and the floating leg tilt. These additional terms are crucial in this case. We believe that the good performance of Set 2 can be traced back to the regressors related to the floating leg tilt. These terms embodies information about the "slope" of the Libor rate curve, that is the relative position of rates at different tenors. The path dependent feature of the payoff function means that the relative position of the rates affects how fast the coupon rate accumulates and thus affects the option value. As a result, using this information in the regressors brings significant improvement.

Using all the living Libors, their squares and cross-products again proves to be quite robust. Set 3, 4 and 5 give quite good results. Set 5 uses the cross-products of the living Libors, which gives higher LSM price compared to the price by Set 2. The cross-products of the living Libors embody information about the joint distribution of these rates, which affects the relative position of the rates. This approach tackles the problem imposed by path dependency from a different angle.

In Set 7, we make a simple approximation for the value of the coupon leg, for which we freeze the living Libor rates at current value. Suppose we are standing at an exercise moment T_n, the Snowball payoff function indicates that we need the rate $L_k(T_{k-1})$ in order to calculate the coupon rate C_k for each $k = n+1, \ldots, N$. Based on the information available at time T_n, we only know $L_{n+1}(T_n)$ among these rates. But we may replace $L_k(T_{k-1})$ by $L_k(T_n)$ for each $k = n+2, \ldots, N$ and get approximation $\widetilde{C}_k$. Then these approximations of the coupon rates are used to calculate an approximation to the value of the coupon leg. In Set 8, we add square of the coupon leg value as a regressor. The results from these two sets are not bad, but they are not much better than the results from Set 5.

The problem with Set 7 and Set 8 is that the frozen Libors are not such a good approximation to future value of the living Libors. A closer approximation may help to improve the pricing result. We devise a second approximation to the coupon leg value base on the following scheme for approximating the future Libor values. We take the vector of living Libor rates at T_n and evolve it in a way similar to our Libor Monte Carlo engine but without stochastic shock. In this way, we hope to obtain closer approximation $\hat{L}_k(T_{k-1})$ for each $k = n+2, \ldots, N$. Depending on the overall level and shape of the volatility term structure, quality of this approximation may vary. Although we do not add any stochastic shock in this construction, the drift of the rate dynamics contains information about the volatility and correlation structure in the future. Hence, we do expect improvement from this future rate approximation.

We adopt the above mentioned scheme in Set 9. Square of the coupon leg value is added as a regressor



(a) $C_0 = 200$ bps

(b) $C_0 = 300$ bps

(c) $C_0 = 400$ bps

Figure 7.13: Simulated lower bound LSM price and the corresponding upper bound AB price for a cancellable receiver Snowball. Results simulated with 3 different virtual coupon rates: $C_0 = 200$bps, 300bps, 400bps.

in Set 10. Square and third power of the coupon leg value are added in Set 11. Simulation results show that Set 9 gives worse result than Set 7, but Set 10 gives by far the best result, which reduces the gap value to about 6bps. Set 11 does not improve over Set 10. The downside of this approach is that it takes roughly twice total simulation time as compared to the total simulation time with other sets of regressors. Preparing the regressors takes most of the computation time in our implementation. If we take computational efficiency into account, Set 5 and 8 are decent choices for the cancellable Snowball.

8 Conclusions and Suggestions

8.1 Conclusions

In this thesis, we have studied the problem of pricing Bermudan options with the Longstaff-Schwartz Monte Carlo in various models and option types. Our goal is to identify better performing regressors for this method in order to reduce pricing error. We have investigated several important issues concerning method implementation and made suggestions with respect to various models and option types. We have shown that using the Andersen-Broadie upper bound algorithm together with the LSM exercise rules produces a better estimate of the true Bermudan option value. We have also shown that the gap between the upper and lower bound prices serves as a good performance measure for evaluating various sets of regressors.

Study of the Bermudan put in the Black-Scholes world shows that for options whose exercise value is observable at the exercise moment, selecting only ITM paths for the LSM regression can give closer option value estimate than selecting all paths. However, gap between the lower and upper bound values can be very high when the option is initially OTM in this case. Yet, we can use the gap to evaluate different regressors when the option is initially ITM or ATM. Lower gap indicates better performing regressors.

Study of the Bermudan put spread in the Black-Scholes world shows that strong nonlinearity present in the payoff function can make it difficult for the LSM to approximate the option's continuation value function, hence resulting in higher pricing error. Choosing the right combination of explanatory variables and basis functions improves the result. Analytical expression for value of the European option with the same payoff function can be used to form an effective explanatory variable.

Study of a Bermudan Asian-style option in the Black-Scholes world shows that one should identify all the observable variables that affects the option's continuation value. For this option, the path dependency property is well captured by the running average of the underlying asset price. Using all these as explanatory variables is consistent with the underlying assumption of the LSM that the continuation value function is a conditional expectation based on information up to the exercise moment.

Study of a Bermudan put in the Heston model shows that one should identify the relative importance of the explanatory variables in estimating the option's continuation value. Assigning an appropriate number of basis functions to the primary variable improves pricing accuracy. Using cross-products between the explanatory variables renders further improvement. These guidelines do work for different correlation coefficient, ρ, in the Heston model.

We then focus on pricing Bermudan-style Libor Exotics in the lognormal Libor market model with the LSM and AB methods. We have adapted the LSM algorithm to pricing both CLE and CnLE. Our results show that these algorithms do produce option values close to a theoretical parity relation between corresponding pairs of CLE and CnLE. We have adapted the AB upper bound algorithm to CnLE. The middle point between the lower and upper bound values is a better estimate of the true option value. The resulting gap value provides a reliable criterion to evaluate different regressor configurations. Careful construction of the regressors helps to extract the most relevant information from the observable variables up to the exercise moment. One should take both future rates evolution and specific option payoff features into consideration.

For a cancellable fixed-for-floating swaption, present values of the fixed leg, the floating leg and the forward swap rates are effective explanatory variables. In our experiment, adding the forward swap rate and its higher order power terms reduces the gap to less than 1bp for a swaption worth over 1000bps.

For a cancellable inverse floater, replacing the fixed leg value by value of the coupon leg and adding higher order terms proves to be very effective. Another effective way is to use all the living Libor rates, their products and cross-products directly as regressors. This approach proves to be both effective and robust.

For a cancellable snowball, the payoff function is itself path dependent. We find that the construction of regressors should reflect the general trend of the rate curve. Regressors that embody the joint distribution information of the rates are also effective. Because of the path dependent payoff function, we do not have a simple expression for the present value of the coupon leg. We propose two effective approximations based on information up to the exercise moment. One of them outperforms all the rest regressor configurations, the other produces decent performance with moderate computation cost.

8.2 Suggestions

During the course of this thesis project, we have come cross several problems that we think are worth the effort of further investigation.

One problem that recurs in multiple occasions is the slightly high biased LSM price when compared to more accurate benchmark price. The exact reason why this happens is not clear. It could be caused by the finite path convergence property of the LSM price estimator, or by certain numerical issue. Our simulation do show that the regressor configuration plays a role in it. Using some proven random number generator other than the Mersenne Twister may help to rule out one possible sources of error.

Throughout the thesis, we estimate the multi-linear model for the continuation value function by OLS regression. We realize the limitation of this approach when trying to fit some more complicated functional form. Our approach for improving the quality of the fit is to construct better explanatory variables. We have experimented using Principal Component Regression and Stepwise Regression in the LSM method, but our simulation results show that their usefulness is not convincing. It would be interesting to explore using other methods such as nonparametric regression [40] or data mining techniques to fit the continuation value function.

The upper bound method for Bermudan option pricing complements the lower bound LSM method well. The drawback is that it takes very long time to compute due to the nested Monte Carlo simulation. Nested Monte Carlo is needed for calculating several conditional expectations. It would be interesting to find more efficient alternatives, although there has been student thesis [57] indicating that this is not easily achievable.

We have studied pricing Bermudan-style LE with the LSM method in the lognormal LMM. It would be interesting to know whether the effectiveness of each regressor configuration would change dramatically or not when other types of initial forward rate curve, volatility term structure, and correlation matrix are used. There have been quite a number of extensions since the original lognormal LMM was introduced in 1997. These extensions are designed to account for the volatility skew as well as its variation over time. One can expect new challenges arising when applying the LSM in these models. Likewise, it is of great interest to find out how the LSM performs with other interest rate derivatives. For example, a spread-based Bermudan-style swaption featuring multiple types of rates or a multi-tranche callable structured note featuring the right for the issuer to increase the size of the note.

A Option Pricing with Tree

Tree methods are commonly used numerical schemes for option pricing. We briefly introduce the theory and address some implementation and performance issues in the Black-Scholes world with the Put payoff. Later, these methods will be used to cross validate the pricing result from other alternative numerical methods.

A.1 Matching Tree Parameters with the Black-Scholes Model

In the Black-Scholes world, the asset price follows a Geometric Brownian Motion(GBM) on the time interval $[0, T]$

$$dS(t) = rS(t)dt + \sigma S(t)dW(t). \tag{A.1}$$

Given the asset price at time t_1, $S(t_1)$, the solution to this SDE is

$$S(t_2) = S(t_1)\exp\left[(r - \frac{1}{2}\sigma^2)(t_2 - t_1) + \sigma W(t_2 - t_1)\right], \text{for } 0 \leq t_1 < t_2 \leq T.$$

Tree methods is a kind of grid method. It discretizes the time and spatial domain to approximate the model dynamics. In order to derive the tree parameters that mimic the GBM model, we need to calculate the first two moments from the GBM model. Later, these expressions will be compared with those consisting of the tree parameters. The ratio $\frac{S(t_2)}{S(t_1)}$ has a lognormal distribution, where $ln\left(\frac{S(t_2)}{S(t_1)}\right) \sim \mathcal{N}\left((r - \frac{1}{2}\sigma^2)(t_2 - t_1), \sigma^2(t_2 - t_1)\right)$. It's mean and second moment are given by,

$$\mathbb{E}\left[\frac{S(t_2)}{S(t_1)}\right] = e^{(r-\frac{1}{2}\sigma^2)(t_2-t_1)+\frac{1}{2}\sigma^2(t_2-t_1)} = e^{r(t_2-t_1)},$$

$$\mathbb{E}\left[\left(\frac{S(t_2)}{S(t_1)}\right)^2\right] = e^{2(r-\frac{1}{2}\sigma^2)(t_2-t_2)+2\sigma^2(t_2-t_1)} = e^{(2r+\sigma^2)(t_2-t_1)}.$$

By choosing $t_1 = t$, $t_2 = t + \Delta t$, the above expressions give,

$$\mathbb{E}\left[\frac{S(t+\Delta t)}{S(t)}\right] = e^{r\Delta t}, \tag{A.2}$$

$$\mathbb{E}\left[\left(\frac{S(t+\Delta t)}{S(t)}\right)^2\right] = e^{(2r+\sigma^2)\Delta t}, \tag{A.3}$$

$$\mathbb{V}ar\left[\frac{S(t+\Delta t)}{S(t)}\right] = (e^{\sigma^2\Delta t} - 1)e^{2r\Delta t}. \tag{A.4}$$

Having derived the first two moments of the return under GBM asset price model, we will match the tree parameters to these value.

The binomial tree scheme discretizes the time interval $[0, T]$ by M steps, with each time step being $\Delta t = \frac{T}{M}$. The time at step i is $t_i = i \cdot \Delta t, i = 0, ..., M$. We call the asset price on a node at t_i to be X_i. At any node with $i < M$ on the tree, assume the following binomial relation under risk-neutral measure $\mathbb{Q}$:

$$\mathbb{Q}(X_{i+1} = u \cdot X_i) = p,$$
$$\mathbb{Q}(X_{i+1} = d \cdot X_i) = 1 - p.$$

The 1st and 2nd moments of $\frac{X_{i+1}}{X_i}$ are

$$\mathbb{E}\left[\frac{X_{i+1}}{X_i}\right] = (1-p)d + pu, \tag{A.5}$$

$$\mathbb{E}\left[\left(\frac{X_{i+1}}{X_i}\right)^2\right] = (1-p)d^2 + pu^2. \tag{A.6}$$

Now fit the tree parameters to the GBM parameters. We align the time points from the above two models by taking $S(t_i) = X_i$ and $S(t_i + \Delta t) = X_{i+1}$. Matching the mean and 2nd moment expressions respectively, we have

$$e^{r\Delta t} = (1-p)d + pu, \tag{A.7}$$
$$e^{(2r+\sigma^2)\Delta t} = (1-p)d^2 + pu^2. \tag{A.8}$$

But (A.7) gives that

$$(u+d)e^{r\Delta t} = (1-p)d^2 + pu^2 + ud. \tag{A.9}$$

Substituting (A.8) into (A.9) eliminates p from the expression. We have,

$$(u+d)e^{r\Delta t} = e^{(2r+\sigma^2)\Delta t} + ud. \tag{A.10}$$

In order to have a recombining tree, we demand that

$$d = 1/u. \tag{A.11}$$

Equation (A.10) and (A.11) gives a quadratic equation on the variable u,

$$u^2 - (e^{-r\Delta t} + e^{(r+\sigma^2)\Delta t)})u + 1 = 0. \tag{A.12}$$

Solving this equation, and taking one of the two solutions, we have

$$\begin{aligned} u &= A + \sqrt{A^2-1}, \qquad \text{where} \\ A &= \frac{1}{2}(e^{-r\Delta t} + e^{(r+\sigma^2)\Delta t}). \end{aligned} \tag{A.13}$$

Besides, Equation (A.7) gives the probability,

$$p = \frac{e^{r\Delta t} - d}{u-d}. \tag{A.14}$$

Now Equations (A.11), (A.13) and (A.14) matches the first two moments of our binomial tree scheme to the GBM model. In order to get a simplified expression for u, we apply Taylor expansion to the exponential terms in (A.12),

$$\begin{aligned} e^{-r\Delta t} &= 1 - r\Delta t + O(\Delta t^2), \\ e^{(r+\sigma^2)\Delta t} &= 1 + (r+\sigma^2)\Delta t + O(\Delta t^2). \end{aligned}$$

Ignoring higher order terms of Δt, we have a new quadratic equation for u,

$$u^2 - (2 + \sigma^2\Delta t)u + 1 = 0.$$

Solving this equation, we have

$$\begin{aligned} u &= \frac{2 + \sigma^2\Delta t \pm \sigma\sqrt{\Delta t}\sqrt{4+\sigma^2\Delta t}}{2} \\ &\approx 1 \pm \sigma\sqrt{\Delta t} + \frac{1}{2}\sigma^2\Delta t \\ &\approx e^{\pm\sigma\sqrt{\Delta t}}. \end{aligned}$$

In the last step, we ignored all higher order terms of Δt. Assuming that $u > d$, we have the following solution that matches the 1st moment, and approximately matches the 2nd moment of our binomial tree scheme to the GBM model,

$$\begin{aligned} u &= e^{\sigma\sqrt{\Delta t}}, \\ d &= 1/u, \\ p &= \frac{e^{r\Delta t} - d}{u-d}. \end{aligned} \tag{A.15}$$

The above solution gives the expression of the Cox, Ross, Rubinstein (CRR) binomial tree. The formal proof and discussion of convergence properties can be found in the pioneering work [19]. Other binomial tree parameterization approaches that match the GBM model have been developed since. [30] gave a literature survey of various alternative binomial and trinomial tree parameterization approaches. The major drawback of the CRR tree is that it loses numerical stability when $\Delta t > \sigma^2/r^2$. In this situation, one of the probability will be negative while the other one larger than one, which is of course nonphysical.

Unlike the binomial tree, a third node is added to each step in trinomial tree. Let u, d and m be the possible asset price rate of change over one step with probability p_u, p_d and p_m respectively. We demand that

$$p_m + p_u + p_d = 1.$$

We discretize time domain in the same way as the binomial tree, We call the asset price on a node at t_i to be X_i. At any node with $i < M$ on the tree, assume the following trinomial relation:

$$\begin{aligned}\mathbb{Q}(X_{i+1} = u \cdot X_i) &= p_u,\\ \mathbb{Q}(X_{i+1} = m \cdot X_i) &= p_m,\\ \mathbb{Q}(X_{i+1} = d \cdot X_i) &= p_d.\end{aligned}$$

In order to derive a recombining tree, we demand that $d = 1/u$ and $m = 1$. Hence the middle branch of the tree would stay at the same asset price value and after two steps, relevant branches of the tree would recombine. Figure **??** illustrates a recombining trinomial tree.

The 1st and 2nd moments of $\frac{X_{i+1}}{X_i}$ are given by

$$\mathbb{E}\left[\frac{X_{i+1}}{X_i}\right] = p_d \cdot 1/u + p_u \cdot u + 1 - p_u - p_d, \tag{A.16}$$

$$\mathbb{E}\left[\left(\frac{X_{i+1}}{X_i}\right)^2\right] = p_d \cdot (1/u)^2 + p_u \cdot u^2 + 1 - p_u - p_d. \tag{A.17}$$

Again we fit the trinomial tree to the GBM in Black-Scholes model. We align the time points from the above two models by taking $S(t_i) = X_i$ and $S(t_i + \Delta t) = X_{i+1}$. Matching the 1st and 2nd moments results in

$$\begin{aligned}M &= p_d \cdot d + p_u \cdot u + 1 - p_u - p_d,\\ Var + M^2 &= p_d \cdot d^2 + p_u \cdot u^2 + 1 - p_u - p_d.\end{aligned}$$

where $M = e^{\Delta t}$, $Var = (\mathrm{e}^{\sigma^2 \Delta t} - 1)e^{2r\Delta t}$.

Boyle[11] solved the probabilities p_u and p_d in terms of M and Var,

$$\begin{aligned}p_u &= \frac{u(Var + M^2 + M) - (M-1)}{(u-1)(u^2-1)},\\ p_d &= \frac{u^2(Var + M^2 + M) - u^3(M-1)}{(u-1)(u^2-1)}.\end{aligned}$$

Boyle suggested to use the parameterization $u = e^{\lambda\sigma\sqrt{t}}$ with $\lambda > 1$, so that the resulting probabilities will be all positive for a range of λ. He investigated this range through trial and error.

Another way to derive a recombining trinomial tree is to combine two steps of a recombining binomial tree. The recombining binomial tree will give 3 different nodes in 2 steps, which makes a single step of a trinomial tree. We set the length of each binomial tree time step to be $\Delta t/2$, so the resulting trinomial tree has a time step of Δt. We use the CRR binomial tree because it has a simple expression, although other forms of recombining binomial trees may also be used.

If we cascade the CRR binomial tree into a trinomial tree with time step Δt, Equation (A.15) leads to the following trinomial tree parameters,

$$\begin{aligned}
u &= (e^{\sigma\sqrt{\Delta t/2}})^2 = e^{\sigma\sqrt{2\Delta t}}, \\
m &= e^{\sigma\sqrt{\Delta t/2}} \cdot e^{-\sigma\sqrt{\Delta t/2}} = 1, \\
d &= (e^{-\sigma\sqrt{\Delta t/2}})^2 = e^{-\sigma\sqrt{2\Delta t}} = 1/u, \\
p_u &= p_u^b \cdot p_u^b = \left(\frac{e^{r\Delta t/2} - e^{-\sigma\sqrt{\Delta t/2}}}{e^{\sigma\sqrt{\Delta t/2}} - e^{-\sigma\sqrt{\Delta t/2}}}\right)^2, \\
p_d &= p_d^b \cdot p_d^b = \left(\frac{e^{\sigma\sqrt{\Delta t/2}} - e^{r\Delta t/2}}{e^{\sigma\sqrt{\Delta t/2}} - e^{-\sigma\sqrt{\Delta t/2}}}\right)^2.
\end{aligned} \tag{A.18}$$

where p_u^b and p_d^b are the probability of going up and down in the CRR binomial tree over time step $\Delta t/2$.

A.2 Pricing European Put with Tree Methods

A European call/put option gives its owner the right to buy/sell the underlying asset at the agreed price K (strike price) at the specified time T (maturity time). To price such an option with tree method, one first calculates the option payoffs at maturity time T,

$$\begin{aligned}
C(T,S) &= \max(S-K,0) && \text{Call Option,} \\
P(T,S) &= \max(K-S,0) && \text{Put Option.}
\end{aligned}$$

and build an asset price tree.

In the Black-Scholes model, we assume a constant volatility σ, and a constant interest rate r. The risk-neutral pricing principle gives the arbitrage free derivative price at time t,

$$V(t) = \mathbb{E}\left[e^{-r(T-t)}V(T) \mid \mathcal{F}(t)\right], 0 \le t \le T, \tag{A.19}$$

where the conditional expectation is taken under the risk-neutral measure.

We can now approximate the righthand side conditional expectation in the above equation, because we have matched our tree parameters to the GBM such that the first two moments match. Since the tree parameters are moment matched to the GBM model under risk neutral measure, the probability parameters in our tree model represents risk neutral probability. For a binomial tree this means

$$V_{i,j} = e^{-r\Delta t}[p_u \cdot V_{i+1,j+1} + p_d \cdot V_{i+1,j-1}], \tag{A.20}$$

where we calculate $V_{i,j}$, the option price at time step i and node j. $V_{i+1,j-1}$ is the option price with an downward asset movement while $V_{i+1,j+1}$ is the option price with a upward asset movement.

For a trinomial tree this means

$$V_{i,j} = e^{-r\Delta t}[p_u \cdot V_{i+1,j+1} + p_m \cdot V_{i+1,j} + p_d \cdot V_{i+1,j-1}], \tag{A.21}$$

with which we can calculate $V_{i,j}$, the option price at time step i and price node j. $V_{i+1,j+1}$ is the option price with an upward price movement, $V_{i+1,j-1}$ is the option price with a downward price movement and $V_{i+1,j}$ is the option price when the asset price is unchanged.

To price with the tree method, we go backwards on the tree. The option value at maturity on the tree nodes are given by the payoff functions. For example, $V_{i,j} = P(S_{i,j}, T)$ for a put option, where $S_{i,j}$ is the asset price on the tree at time step i and price node j. Then we go each time one time step back, calculate option value on all the nodes at that time, and continue the recursion until we reach the initial time point.

A.3 Pricing American or Bermudan Put with Tree Methods

In order to price a Bermudan option with a set of possible early exercise moments while reducing the pricing error, we choose a large number of time steps for the tree, but only check early exercise at those possible moments. The computation is based on the dynamic programming formulation 2.12. Starting at the maturity and going backwards, at each possible exercise moment we calculate the exercise value and hold value of the option. Make the exercise decision and go on. At none-exercise moments, we compute one step backward on the tree like the European option. For a GBM model, since we have a constant risk free interest rate, we discount differently as implied in Equation 2.12. Instead of discounting the values back to zero at each time, we discount one step back at a time by $\mathrm{e}^{-r\Delta t}$.

The exercise value of a vanilla option at a time step prior to the maturity is given by

$$\begin{aligned} C(S_{i,j}) &= \max(S_{i,j} - K, 0), && \text{for a Call Option,} \\ P(S_{i,j}) &= \max(K - S_{i,j}, 0), && \text{for a Put Option.} \end{aligned}$$

This means that for a Bermudan Put, we use the following equations in recursion. For a binomial tree, this is

$$V_{i,j} = \max\left(P(S_{i,j}), e^{-r\Delta t}[p_u \cdot V_{i+1,j+1} + p_d \cdot V_{i+1,j-1}]\right).$$

For a trinomial tree, this is

$$V_{i,j} = \max\left(P(S_{i,j}), e^{-r\Delta t}[p_u \cdot V_{i+1,j+1} + p_m \cdot V_{i+1,j} + p_d \cdot V_{i+1,j-1}]\right).$$

Let $\mathcal{T}$ denote the set of possible early exercise moments. We take Binomial tree as an example, the pricing algorithm becomes,

```
At maturity, $t_N = T$, $V_{N-1,j} = e^{-r\Delta t}[p_u \cdot V_{N,j+1} + p_d \cdot V_{N,j-1}]$.
for $i = N-2 : -1 : 0$
    if $t_i \in \mathbb{T}$, then $V_{i,j} = \max(P(S_{i,j}), e^{-r\Delta t}[p_u \cdot V_{i+1,j+1} + p_d \cdot V_{i+1,j-1}])$.
    else              $V_{i,j} = e^{-r\Delta t}[p_u \cdot V_{i+1,j+1} + p_d \cdot V_{i+1,j-1}]$.
    end if
end for
```

■

with large number of time steps to give alternative pricing benchmark for several different Bermudan option in the Black-Scholes world.

B Option Pricing with Finite Difference

By the Finite Difference (FD) method, one numerically solves a Partial Differential Equation (PDE) for the option value. For American/Bermudan option, it is possible to use FD method to solve an approximation to the true early exercise boundary of the option. This provides valuable insight into the pricing problem. In this section, we first discuss some basics of the FD method by pricing a European vanilla option in the Black-Scholes world. Then we discuss pricing American/Bermudan option with the FD method. We have also worked on pricing some other options with the FD method, such as a Bermudan-Asian option in the Black-Scholes world and a Bermudan-Put in the Heston model. These will be discussed in later chapters where the specific problems are dealt with.

B.1 Pricing European Put with Explicit Scheme

To price a European vanilla option with the FD method, we start with the Black-Scholes PDE and the option payoff as the final condition for the PDE. Let $V(t, S)$ be the option price at time t with underlying asset price process $\{S(t), 0 \leq t \leq T\}$.

$$\frac{\partial \bar{V}}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 \bar{V}}{\partial S^2} + rS\frac{\partial \bar{V}}{\partial S} - r\bar{V} = 0, \tag{B.1}$$

$$\bar{V}(T, S) = \bar{P}(T, S). \tag{B.2}$$

where we $\bar{P}(t, s)$ is the simple Put payoff function.

First we transform the time variable t by $\tau = T - t$ for $0 \leq t \leq T$, and the original option price function $\bar{V}(t, S)$ changes to $\widetilde{V}(\tau, S)$. Thus the final condition $\bar{V}(T, S)$ becomes the initial condition $\widetilde{V}(0, S)$ for the new PDE. The payoff function $\bar{P}(t, s)$ changes to $\widetilde{P}(\tau, s)$. Now the new PDE is accompanied by initial condition, which is common in many physical problems.

$$-\frac{\partial \widetilde{V}}{\partial \tau} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 \widetilde{V}}{\partial S^2} + rS\frac{\partial \widetilde{V}}{\partial S} - r\widetilde{V} = 0, \tag{B.3}$$

$$\widetilde{V}(0, S) = \widetilde{P}(0, S). \tag{B.4}$$

$S \in (0, +\infty)$ in the above equations. But the computation domain for S needs to be cut off at some finite number, so let $S \in [S_{min}, S_{max}]$. We use V to denote the FD approximation to $\widetilde{V}$. The time domain is discretized by grid $0{=}\tau_0 < \tau_1 < ... < \tau_{N_\tau} = T$, and the asset price domain by grid $S_{min} = S_0 < S_1 < ... < S_{N_S} = S_{max}$. Now we have $\tau_n = n \cdot \Delta\tau$, $S_j = j \cdot \Delta S$ and $V_{n,j} = V(\tau_n, S_j)$ for $n = 0, ..., N_\tau$, and $j = 0, ..., N_S$.

For a European put option, we have the following theoretical boundaries [50]:

$$\widetilde{V}(\tau, S) = Ke^{-r\tau} - S, \qquad \text{for } S \approx 0. \tag{B.5}$$

$$\widetilde{V}(\tau, S) = 0, \qquad \text{for } S \to \infty. \tag{B.6}$$

We approximate the Black-Scholes PDE around point (τ_{n+1}, S_j) with the following partial derivative approximations based on Taylor series expansion,

$$\frac{\partial \widetilde{V}}{\partial S} \approx \frac{V_{n,j+1} - V_{n,j-1}}{2\Delta S}, \tag{B.7}$$

$$\frac{\partial^2 \widetilde{V}}{\partial S^2} \approx \frac{V_{n,j+1} - 2V_{n,j} + V_{n,j-1}}{\Delta S^2}, \tag{B.8}$$

$$\frac{\partial \widetilde{V}}{\partial \tau} \approx \frac{V_{n+1,j} - V_{n,j}}{\Delta\tau}. \tag{B.9}$$

Depending on how we choose the approximation for $\widetilde{V}$, we would have two different versions of explicit schemes.

If we choose

$$\widetilde{V} = V_{n,j}. \tag{B.10}$$

then substituting equations (B.7)-(B.10) into equation (B.3) and reshuffling the equation renders

$$V_{n+1,j} = (p \cdot S_j^2 - q \cdot S_j)V_{n,j-1} + (p \cdot S_j^2 + q \cdot S_j)V_{n,j+1} - (1 - r\Delta\tau - 2p \cdot S_j^2)V_{n,j}. \tag{B.11}$$

where

$$p = \frac{\sigma^2}{2}\frac{\Delta\tau}{\Delta S^2},$$
$$q = \frac{r}{2}\frac{\Delta\tau}{\Delta S}.$$

If we choose

$$\widetilde{V} = V_{n+1,j}. \tag{B.12}$$

then substituting equations (B.7)-(B.9), (B.12) into equation (B.3) and reshuffling the equation renders

$$V_{n+1,j} = \frac{1}{1+r\Delta\tau}[(p\cdot S_j^2 - q\cdot S_j)V_{n,j-1} + (p\cdot S_j^2 + q\cdot S_j)V_{n,j+1} - (1-2p\cdot S_j^2)V_{n,j}]. \tag{B.13}$$

where

$$p = \frac{\sigma^2}{2}\frac{\Delta\tau}{\Delta S^2},$$
$$q = \frac{r}{2}\frac{\Delta\tau}{\Delta S}.$$

The initial conditions are also given on the grid $V_{0,j} = \widetilde{P}(0, S_j)$ for all $j = 0, \ldots, N_S$. The boundary conditions are given on the left and right boundaries, $S = S_{\min}$ and $S = S_{\max}$,

$$V_{n,0} = Ke^{-r\tau_n} - S_{\min}, \qquad \text{for all n,} \tag{B.14}$$
$$V_{n,N_s} = 0, \qquad \text{for all n.} \tag{B.15}$$

Both difference equations (B.11) and (B.13) give valid explicit schemes. We can directly calculate the option price $V_{n+1,j}$ at time step $n+1$ if we already know the price on 3 adjacent nodes at time step n, $V_{n,j-1}$, $V_{n,j}$, and $V_{n,j+1}$. So we can start from the initial condition and repeatedly solve option prices over all the grid nodes except those given boundary nodes.

The main drawback of the explicit scheme is that its stability depends on the discretization parameters. For the PDE that we discretize, there is an upper limit of time step size Δt given the step size of the option price such that the FD solution remains stable and converges to the solution of the original PDE. If the parameters are chosen beyond this upper limit, the FD solution might show such behavior as the oscillation in Figure B.1. The discretization parameters in this example are $\Delta S = 0.25$, $\Delta\tau = 0.005$.

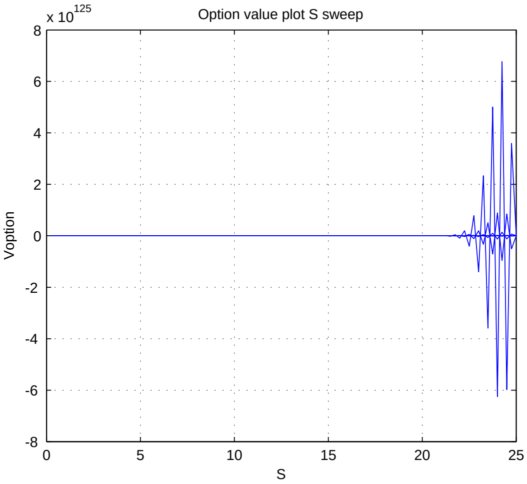


Figure B.1: Oscillation in explicit scheme discretization of the Black-Scholes PDE

We can combine equations (B.11) or (B.13) for all the nodes at one time step into a matrix form and obtain a tridiagnal matrix similar to the one that we show in the next section. Fusai etc. [26] analyzed the stability conditions of various finite difference schemes. They give the following bound for the explicit scheme,

$$\Delta t \le \frac{1}{\frac{r}{2} + (\sigma N_S)^2},$$

which gives us an idea about how large the time step should be prior to each simulation.

B.2 Pricing European Put with $\theta-$Scheme

In order to avoid the stability issue with the explicit scheme, one can use the implicit scheme to discretize the PDE instead. Suppose we have already computed value on all the grid points at time step n, the spatial derivatives in Equation (B.3) are discretized at time step $n+1$. We have the following approximations,

$$\begin{aligned}
\frac{\partial \widetilde{V}}{\partial S} &\approx \frac{V_{n+1,j+1} - V_{n+1,j-1}}{2\Delta S}, \\
\frac{\partial^2 \widetilde{V}}{\partial S^2} &\approx \frac{V_{n+1,j+1} - 2V_{n+1,j} + V_{n+1,j-1}}{\Delta S^2}, \\
\frac{\partial \widetilde{V}}{\partial \tau} &\approx \frac{V_{n+1,j} - V_{n,j}}{\Delta \tau}, \\
\widetilde{V} &\approx V_{n+1,j}.
\end{aligned}$$

Plugging these into Equation (B.3) and reshuffle, we have,

$$\begin{aligned}
\left(-\frac{1}{2}\sigma^2 \Delta\tau \frac{S_j^2}{\Delta S^2} + \frac{1}{2} r\Delta\tau \frac{S_j}{\Delta S}\right) V_{n+1,j-1} + \left(1 + r\Delta\tau + \sigma^2 \Delta\tau \frac{S_j^2}{\Delta S^2}\right) V_{n+1,j} \\
+ \left(-\frac{1}{2}\sigma^2 \Delta\tau \frac{S_j^2}{\Delta S^2} - \frac{1}{2} r\Delta\tau \frac{S_j}{\Delta S}\right) V_{n+1,j+1} = V_{n,j}.
\end{aligned} \tag{B.16}$$

Notice here all the unknowns are on the left hand side of the equation. Hence when we combine all the equations from step n to step $n+1$ and fill in the right boundary conditions, we obtain a system of equations in the matrix form. If this matrix equation can be solved, all the values $V_{n+1,j}$ for $j = 1, \ldots, N_s - 1$ can be obtained at once. It is proved that this indeed can be done and the implicit scheme is free of oscillation when $\sigma^2 > r$ [26]. The main drawback is the relatively costly matrix computation.

One can combine the explicit scheme and the implicit scheme into the so called "θ scheme". Discretization now becomes,

$$\begin{aligned}
\frac{\partial \widetilde{V}}{\partial S} &\approx (1-\theta)\frac{V_{n,j+1} - V_{n,j-1}}{2\Delta S} + \theta \frac{V_{n+1,j+1} - V_{n+1,j-1}}{2\Delta S}, \\
\frac{\partial^2 \widetilde{V}}{\partial S^2} &\approx (1-\theta)\frac{V_{n,j+1} - 2V_{n,j} + V_{n,j-1}}{\Delta S^2} + \theta \frac{V_{n+1,j+1} - 2V_{n+1,j} + V_{n+1,j-1}}{\Delta S^2}, \\
\frac{\partial \widetilde{V}}{\partial \tau} &\approx \frac{V_{n+1,j} - V_{n,j}}{\Delta \tau}, \\
\widetilde{V} &\approx (1-\theta)V_{n,j} + \theta V_{n+1,j}.
\end{aligned}$$

Plugging these into Equation (B.3) and reshuffle, we have,

$$a_{n,j}V_{n+1,j-1} + b_{n,j}V_{n+1,j} + c_{n,j}V_{n+1,j+1} = d_{n,j}V_{n,j-1} + e_{n,j}V_{n,j} + f_{n,j}V_{n,j+1}, \tag{B.17}$$

where

$$
\begin{aligned}
a_{n,j} &= \theta \left(-\frac{1}{2}\sigma^2 \Delta\tau \frac{S_j^2}{\Delta S^2} + \frac{1}{2} r \Delta\tau \frac{S_j}{\Delta S} \right), \\
b_{n,j} &= \left(1 + \theta r \Delta\tau + \theta \sigma^2 \Delta\tau \frac{S_j^2}{\Delta S^2} \right), \\
c_{n,j} &= \theta \left(-\frac{1}{2}\sigma^2 \Delta\tau \frac{S_j^2}{\Delta S^2} - \frac{1}{2} r \Delta\tau \frac{S_j}{\Delta S} \right), \\
d_{n,j} &= (1-\theta) \left(\frac{1}{2}\sigma^2 \Delta\tau \frac{S_j^2}{\Delta S^2} - \frac{1}{2} r \Delta\tau \frac{S_j}{\Delta S} \right), \\
e_{n,j} &= \left(1 - (1-\theta) r \Delta\tau - (1-\theta) \sigma^2 \Delta\tau \frac{S_j^2}{\Delta S^2} \right), \\
f_{n,j} &= (1-\theta) \left(\frac{1}{2}\sigma^2 \Delta\tau \frac{S_j^2}{\Delta S^2} + \frac{1}{2} r \Delta\tau \frac{S_j}{\Delta S} \right).
\end{aligned}
$$

Collecting all the equations connecting step n and step $n+1$, we have the following system,

$$
\mathbf{A} V^{(n+1)} = \mathbf{B} V^{(n)} + d^{(n)},
$$

where $(N_s - 1) \times 1$ vector $\mathbf{V}^{(n)}$ and $\mathbf{V}^{(n+1)}$ represent the option value on grid points at time n and $n+1$, $(N_s - 1) \times 1$ vector $d^{(n)}$ accounts for all the boundary conditions at S_0 and S_{N_s}. Matrix $\mathbf{A}$ and $\mathbf{B}$ store all the coefficients. Notice that all the vectors on the right hand side of the above equation is known. We give the details below. Let's define the vectors at time step n to be

$$
\begin{aligned}
V^{(n)} &= (V_{n,1}, \ldots, V_{n,N_s-1})^T, \\
V^{(n+1)} &= (V_{n+1,1}, \ldots, V_{n+1,N_s-1})^T, \\
d^{(n)} &= (d_{n,1}, 0, \ldots, 0, d_{n,N_s-1})^T,
\end{aligned}
$$

where T denotes transpose and

$$
\begin{aligned}
d_{n,1} &= a_{n,1} V_{n+1,0} + d_{n,1} V_{n,0} \\
d_{n,N_s-1} &= c_{n,N_s-1} V_{n,N_s} + f_{n,N_s-1} V_{n,N_s}.
\end{aligned}
$$

The coefficient matrices are

$$
\mathbf{A} = \begin{pmatrix}
b_{n,1} & c_{n,1} & & & 0 \\
a_{n,2} & b_{n,2} & c_{n,2} & & \\
 & \ddots & \ddots & \ddots & \\
 & & a_{n,N_s-2} & b_{n,N_s-2} & c_{n,N_s-2} \\
0 & & & a_{n,N_s-1} & b_{n,N_s-1}
\end{pmatrix} \in \mathbb{R}^{(N_s-1)\times(N_s-1)},
$$

and

$$
\mathbf{B} = \begin{pmatrix}
e_{n,1} & f_{n,1} & & & 0 \\
d_{n,2} & e_{n,2} & f_{n,2} & & \\
 & \ddots & \ddots & \ddots & \\
 & & d_{n,N_s-2} & e_{n,N_s-2} & f_{n,N_s-2} \\
0 & & & e_{n,N_s-1} & f_{n,N_s-1}
\end{pmatrix} \in \mathbb{R}^{(N_s-1)\times(N_s-1)}.
$$

It is proved that for $\theta \in [0.5, 1]$, the θ scheme produce unconditionally stable solution in the sense that we do not have to restrain size of the time step for stability concern. The time step size still affects numerical accuracy, however that is far more relaxed than the constraint in explicit scheme. As a result, we can choose finer grid in the spatial direction to improve solution accuracy. Notice that when choosing $\theta = 0$, Equation (B.17) degenerates into the explicit scheme (B.11); when choosing $\theta = 1$, Equation (B.17) degenerates into the implicit scheme (B.16). [3] covers the error and stability analysis for these basic FD schemes. For analysis with regards to the Black-Scholes equation, one can turn to [50] and [26].

B.3 Pricing American Put with Explicit Scheme

The explicit scheme can be used to approximate American option price with some adjustments. Since we are working with discretized steps in the time domain, we are actually pricing a Bermudan option with the FD scheme. But as we take finer and finer time steps, the pricing result approximates that of the corresponding American option.

The payoff at maturity is the same for a European option or its American variant, initial condition remains the same for the PDE. The boundary conditions should be modified. Let us consider a put option. On the right boundary, $S \to \infty$ and the put option is deep OTM, its value approaches zero. On the left boundary, $S \approx 0$, so the put option is deep ITM. But for American option this also means that the option is in the immediate exercise region at that moment, so the option value equals its exercise value, $K - S$. The boundary conditions (B.6) and (B.5) now become

$$\widetilde{V}(\tau, S) = K - S, \qquad \text{for } S \approx 0, \tag{B.18}$$

$$\widetilde{V}(\tau, S) = 0, \qquad \text{for } S \to \infty. \tag{B.19}$$

The corresponding boundary conditions on the grid become

$$V_{n,0} = K - S_{\min}, \qquad \text{for all n}, \tag{B.20}$$

$$V_{n,N_s} = 0, \qquad \text{for all n}. \tag{B.21}$$

The dynamic programming formulation (2.12) means that the option price is determined by comparing its exercise value and hold value at each possible exercise moment. Incorporating this feature is straight forward with the explicit scheme. Suppose we already know the option value on the 3 points, $V_{n,j-1}$, $V_{n,j}$, $V_{n,j+1}$, then the value calculated by Equation (B.11) or (B.13) is the hold value at time step $n+1$ and price step j, $H_{n+1,j}$. This value will be compared with the exercise value $Z_{n+1,j} = \max[K - S_{n+1,j}, 0]$. The resulting option price is given by the greater of the two

$$V_{n+1,j} = \max[Z_{n+1,j}, H_{n+1,j}].$$

With these modifications, the explicit scheme can be used for pricing American vanilla options. Our explicit scheme for American option is based on equation (B.11).

B.4 Pricing American Put as Linear Complementarity Problem

Although the explicit scheme can be adjusted to price American options, the stability problem remains. An alternative is to switch to an implicit scheme or the θ-scheme.

To account for early exercise opportunities, we need to compare the immediate exercise value and the hold value of the option at each early exercise moment prior to maturity. But implicit scheme solves the option price on all asset value grid points at a time step in one matrix equation. It is therefore difficult to take into account the early exercise decision in the same way as how the explicit scheme does it.

For an alternative FD method of pricing American options, we follow the procedure as described in chapter 4 of Seydel's work [50]. Let us define a function of t, $S_f(t)$, as the early exercise boundary of an American Put option. for all $t \in [0, T]$. The intuition is that when the asset price $S(t) > S_f(t)$, the option price $\bar{V}(t, S) > (K - S)^+$, its exercise value. In this case, it is in the interest of the option's holder not to exercise early. We call this region the continuation region. On the other hand, when $S(t) \leq S_f(t)$, $\bar{V}(t, S) = K - S$, it is better to exercise the option immediately and maximize the profit. Seydel showed in the appendix that for a put option, on these two regions, the following relations hold,

$$\begin{aligned}
&\text{if } S(t) \leq S_f(t), && \bar{V}(t,S) = (K - S(t))^+ \text{ and} \\
& && \frac{\partial \bar{V}}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 \bar{V}}{\partial S^2} + rS\frac{\partial \bar{V}}{\partial S} - r\bar{V} < 0, \qquad \text{(B.22)} \\
&\text{if } S(t) > S_f(t), && \bar{V}(t,S) > (K - S(t))^+ \text{ and} \\
& && \frac{\partial \bar{V}}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 \bar{V}}{\partial S^2} + rS\frac{\partial \bar{V}}{\partial S} - r\bar{V} = 0. \qquad \text{(B.23)}
\end{aligned}$$

In order to simplify the derivation, we adopt the following transformation to convert the Black-Scholes equation into a heat equation with constant coefficient. Let us take

$$S = Ke^x, t = T - \frac{2\tau}{\sigma^2}, q = \frac{2r}{\sigma^2}. \tag{B.24}$$

The original domain $S \times t \in [0, \infty] \times [0, T]$ turns into the transformed domain $x \times \tau \in [-\infty, +\infty] \times [0, \frac{1}{2}\sigma^2 T]$.The option price $\bar{V}(S, t)$, is transformed into function $y(x, \tau)$ of new variables x and τ by

$$\bar{V}(t, S) = \bar{V}(T - \frac{2\tau}{\sigma^2}, Ke^x) = K\text{exp}\left\{-\frac{1}{2}(q-1)x - (\frac{1}{4}(q-1)^2 + q)\tau\right\} y(\tau, x).$$

Then the equalities and inequalities that hold for $\bar{V}(S, t)$ on the two regions turn into equalities and inequalities for $y(\tau, x)$. With the above transformation, the early exercise boundary turns into $x_f(\tau)$. We have,

$$\begin{aligned}
\text{if } x(\tau) \leq x_f(\tau), \qquad & y(\tau, x) = g(\tau, x) \text{ and} \\
& \frac{\partial y}{\partial \tau} - \frac{\partial^2 y}{\partial x^2} > 0, && \text{(B.25)} \\
\text{if } x(\tau) > x_f(\tau), \qquad & y(\tau, x) > g(\tau, x) \text{ and} \\
& \frac{\partial y}{\partial \tau} - \frac{\partial^2 y}{\partial x^2} = 0. && \text{(B.26)}
\end{aligned}$$

where $g(\tau, x) = \exp\left\{\frac{1}{4}(q-1)^2\tau + 4q\right\} \max\left\{e^{\frac{1}{2}(q-1)x} - e^{\frac{1}{2}(q+1)x}, 0\right\}$.

The equalities and inequalities in (B.25) and (B.27) is equivalent to the following Linear Complementarity Problem (LCP),

$$\begin{aligned}
\left(\frac{\partial y}{\partial \tau} - \frac{\partial^2 y}{\partial x^2}\right)(y - g) = 0, \\
\frac{\partial y}{\partial \tau} - \frac{\partial^2 y}{\partial x^2} \geq 0, \ y - g \geq 0.
\end{aligned} \tag{B.27}$$

In this way the early exercise boundary $x_f(\tau)$ does not show up explicitly in the equations, so we can describe the behavior of the option price on the whole computation grid with one system.

The original final condition, $\bar{V}(T, S) = \max\{K - S, 0\}$, is then transformed into the initial condition,

$$y(0, x) = g(0, x), \qquad \text{for } -\infty \leq x \leq +\infty. \tag{B.28}$$

The original conditions on fixed boundaries for $S \to \pm\infty$ are

$$\begin{aligned}
\lim_{S \to 0} V(t, S) = K - S, \\
\lim_{S \to +\infty} V(t, S) = 0.
\end{aligned}$$

They are transformed into

$$\lim_{x \to \pm\infty} y(\tau, x) = \lim_{x \to \pm\infty} g(\tau, x), \qquad \text{for } 0 \leq \tau \leq \frac{1}{2}\sigma^2 T. \tag{B.29}$$

Now the original put option price conditions on the $S \times t$ domain has been reformulated into the LCP with fixed boundary conditions described by system (B.27), (B.28) and (B.29). System (B.27) gives a nonlinear PDE with 2 inequality constraints, which is still not easy to solve by numerical techniques. However, it is possible to make the discretization at this step and simplify the problem further in latter steps. We cut off the computation domain to $\tau \times x \in [x_{\max}, x_{\min}] \times [0, \frac{1}{2}\sigma^2 T]$. We use w to denote the FD approximation to y. The time domain is discretized by grid $0 = \tau_0 < \tau_1 < ... < \tau_{N_\tau} = \frac{1}{2}\sigma^2 T$, and the w domain by grid $x_{\min} = x_0 < x_1 < ... < x_{N_x} = x_{\max}$. Now we have $\tau_n = n \cdot \Delta\tau$, $x_j = j \cdot \Delta x$ and $w_{n,j} = w(\tau_n, x_j)$ for $n = 0, ..., N_\tau$, and $j = 0, ..., N_x$.

With this grid, we approximate system (B.27) by the θ-scheme. At each time step $0 \leq n < N_\tau$ and each asset price node excluding the boundaries $1 < j < N_x$, we have the following approximations,

$$\begin{aligned}\frac{\partial^2 y}{\partial x^2} &= (1-\theta)\frac{w_{n,j+1} - 2w_{n,j} + w_{n,j-1}}{\Delta x^2} \\ &\quad + \theta\frac{w_{n+1,j+1} - 2w_{n+1,j} + w_{n+1,j-1}}{\Delta x^2}, \\ \frac{\partial y}{\partial \tau} &= \frac{w_{n+1,j} - w_{n,j}}{\Delta \tau}.\end{aligned}$$

Let $\lambda = \frac{\Delta\tau}{\Delta x^2}$, then at grid point (τ_n, x_j), $\frac{\partial y}{\partial \tau} - \frac{\partial^2 y}{\partial x^2} \geq 0$ turns into the difference inequality

$$\begin{aligned}&-\lambda\theta w_{n+1,j+1} + (1-2\lambda\theta)w_{n+1,j} - \lambda\theta w_{n+1,j-1} \\ &\qquad \geq \lambda(1-\theta)w_{n,j+1} + [1-2\lambda(1-\theta)]w_{n,j} + \lambda(1-\theta)w_{n,j-1}\end{aligned} \tag{B.30}$$

The other inequality, $y - g \geq 0$, turns into

$$w_{n,j} \geq g_{n,j}. \tag{B.31}$$

where $g_{n,j} = g(\tau_n, x_j)$.

$\left(\frac{\partial y}{\partial \tau} - \frac{\partial^2 y}{\partial x^2}\right)(y-g) = 0$ turns into

$$\begin{aligned}&[-\lambda\theta w_{n+1,j+1} + (1-2\lambda\theta)w_{n+1,j} - \lambda\theta w_{n+1,j-1} \\ &\quad - \lambda(1-\theta)w_{n,j+1} - [1-2\lambda(1-\theta)]w_{n,j} - \lambda(1-\theta)w_{n,j-1}] \cdot [w_{n,j} - g_{n,j}] = 0\end{aligned} \tag{B.32}$$

We define the vectors at time step n to be

$$\begin{aligned}b^{(n)} &= (b_{n,1}, \ldots, b_{n,N_x-1})^T, \\ w^{(n)} &= (w_{n,1}, \ldots, w_{n,N_x-1})^T, \\ g^{(n)} &= (g_{n,1}, \ldots, g_{n,N_x-1})^T.\end{aligned}$$

where T means transpose.

The coefficient matrices are

$$\mathbf{A} = \begin{pmatrix} 1+2\lambda\theta & -\lambda\theta & & & 0 \\ -\lambda\theta & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -\lambda\theta \\ 0 & & & -\lambda\theta & 1+2\lambda\theta \end{pmatrix} \in \mathbb{R}^{(N_x-1)\times(N_x-1)}$$

$$\mathbf{B} = \begin{pmatrix} 1-2\lambda(1-\theta) & \lambda(1-\theta) & & & 0 \\ \lambda(1-\theta) & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \lambda(1-\theta) \\ 0 & & & \lambda(1-\theta) & 1-2\lambda(1-\theta) \end{pmatrix} \in \mathbb{R}^{(N_x-1)\times(N_x-1)}$$

Then we write equality (B.30) over the asset value nodes $j = 1, \ldots, N_x - 1$ into vector form and have

$$\mathbf{A}w^{(n+1)} \geq b^{(n)} \tag{B.33}$$

where $b^{(n)}$ is given by

$$b^{(n)} = \mathbf{B}w^{(n)} + d^{(n)}$$

and

$$d^{(n)} = \begin{pmatrix} \lambda\theta w_{n+1,1} + \lambda(1-\theta)w_{n,1} \\ 0 \\ \vdots \\ 0 \\ \lambda\theta w_{n+1,N_x} + \lambda(1-\theta)w_{n,N_x} \end{pmatrix} \in \mathbb{R}^{(N_x-1)\times 1}$$

We write inequality (B.31) over the asset value nodes $j = 1, \ldots, N_x - 1$ into a vector form and have

$$w^{(n+1)} \geq g^{(n+1)} \tag{B.34}$$

We write equality (B.32) over the asset value nodes $j = 1, \ldots, N_x - 1$ into a vector form and have

$$\left(\mathbf{A}w^{(n+1)} - b^{(n)}\right)^T \left(w^{(n+1)} - g^{(n+1)}\right) = 0 \tag{B.35}$$

The initial and the boundary conditions are

$$\begin{aligned} w_{0,j} &= g_{0,j}, && \text{for } j = 0, \ldots, N_x, && \text{(B.36)} \\ w_{n,1} = g_{n,1}, \quad w_{n,N_x} &= g_{n,N_x}, && \text{for } n = 1, \ldots, N_\tau. && \text{(B.37)} \end{aligned}$$

Now we can start with the initial condition and repeatedly calculate the vector $w^{(n+1)}$ as solution of system (B.33) - (B.35) under proper boundary conditions.

In order to further simplify our problem, let us make the following substitution

$$\begin{aligned} m &= w^{(n+1)} - g^{(n+1)}, \\ n &= \mathbf{A}w^{(n+1)} - b^{(n)}. \end{aligned}$$

And notice that for $\hat{b} = b - \mathbf{A}g$, $\mathbf{A}m - n \equiv \hat{b}$. Our problem of calculating $w^{(n+1)}$ from (B.33) - (B.35) is equivalent to the problem of finding vectors m and n such that $\mathbf{A}m - n = \hat{b}, m \geq 0, n \geq 0, m^T n$ =0.

Seydel adopted the work of Cryer [20] and showed further that this problem is equivalent to the minimization problem through the Karush-Kuhn-Tucker theorem in convex optimization,

$$\min_{m \geq 0} G(m), \qquad \text{where } \; G(m) = \frac{1}{2}(m^T \mathbf{A} m) - \hat{b}m \text{ is strictly convex.} \tag{B.38}$$

It can be shown that the Hessian matrix $\nabla^2 G = \mathbf{A}$ is positive definite, so $G(m)$ is strictly convex. Thus we can try to solve the above problem by looking for vector m such that the gradient, $\nabla G(m) = 0$ under the constraint that $m \geq 0$. Now this problem can be solved by the so called Projected Successive Over-Relaxation method (PSOR), which is an extension to the Gauss-Seydel method for solving a linear system through iterations [42].

Let k be the recursion counter, ω_R be the the relaxation coefficient, $\|\cdot\|$ be a norm on vector m, and ϵ be the minimum error for the iteration to continue. An approximation to the minimizer of the above problem, vector m, can be solved by the following algorithm:

Set $k = 0$, pick an arbitrary initial value for vector m^0 and continue with the recursion

```
while $\|m^{k+1} - m^k\| > \epsilon$
{
    for $j = 1, 2, \ldots, N_x - 1$
    {
        $m_j^{k+1} = m_j^k + \frac{\omega_R}{1 + 2\lambda\theta}\left[\hat{b}_j + \lambda\theta m_{j-1}^{k+1} - (1 + 2\lambda\theta)m_j^k + \lambda\theta m_{j+1}^k\right]$
        $m_j^{k+1} = \max\left\{g_{n+1,j}, m_j^{k+1}\right\}$
        $k = k + 1$
    }
}
```

■

With this algorithm, we can calculate the vector $w^{(n+1)}$ from system (B.33) - (B.35) for $n = 0, \ldots, N_\tau - 1$. After solving all the w, we can then use equations in (B.24) to transform the grid points back to the $t \times S$ domain. Thus we can solve the whole surface of American put option price.

B.5 Pricing Bermudan Put

In order to price a Bermudan put option with a given set of possible early exercise opportunities, we build the FD solver in the following way. Suppose the whole time interval till the option's maturity is discretized with grid, $0 = t_0 < t_1 < \cdots < t_{N-1} < t_N = T$. Let $\mathcal{T}$ denote the set of all possible early exercise moments. We start at maturity and calculate the option value backwards. The option value follows the Black-Scholes PDE just as a European option between two early exercise moments(as well as in the first and last time interval), so FD method (such as $\theta-$scheme) on the Black-Scholes PDE can be used to calculate the option value between these moments. At an early exercise moment, the pricing problem should be formulated as an LCP, hence the scheme described in the last section for pricing American option should be used instead.

Suppose $\mathcal{T} = \{t_{e_1}, t_{e_2}, \ldots, t_{e_M}\}$ and $0 < t_{e_1} < t_{e_2} < \cdots < t_{e_M} < T$. All the early exercise moments are on grid, i.e. $e_k \in \{0, \ldots, N-1\}$ for all $k = 1, \ldots, M$. Let $t_{e_0} = 0, t_{e_{M+1}} = T$ to simplify the indexing. The algorithm reads,

```
At maturity, the option price satisfies the payoff function, thus $V(t_N, S) = \text{Payoff}(S)$.
for $i = M : -1 : 0$
    Use $V(t_{e_{i+1}}, S)$ as the final condition on the interval $[t_{e_i}, t_{e_{i+1}}]$.
    for $j = e_{i+1} : -1 : e_i+1$
        Solve the option price $V(t_j, S)$ by $\theta-$scheme as a European option
        with lifetime $[t_{e_i}, t_{e_{i+1}}]$.
    end for
    Solve the option value from $V(t_{e_i+1}, S)$ back to $V(t_{e_i}, S)$ as an LCP
    using PSOR (except when $i = 0$).
end for
```

■

C An Approximated Black Volatility for Forward Swap Rates in the LMM

In [37], Jäckel and Rebonato proposed an approximated Black volatility for the forward swap rate $R_0^N(0)$ in the LMM. We will briefly discuss their idea and extend the formula to all coterminal forward swap rates $R_n^N(0)$, for each $n = 0, \dots, N-1$.

We use the same time grid as the one in Chapter 6. The time horizon $[0,T]$ is divided by the sequence of time points, $0 = T_{-1} < T_0 < T_1 < \dots < T_N = T$. Suppose under some pricing EMM $\mathbb{Q}$, the arbitrage-free Libor dynamics is given by

$$\frac{dL_i(t)}{L_i(t)} = \mu_i(\mathbf{L}, t)dt + \sigma_i(t)dW_i(t),$$

for each i =where W_i is a $\mathbb{Q}$ standard Brownian Motion, $\sigma_i(t)$ and $\mu_i(x,t)$ are deterministic functions in $\mathbb{R}$. The Brownian Motions are correlated,

$$dW_i \cdot dW_j = \rho_{ij}(t)dt,$$

where $\rho_{ij}(t)$ is the deterministic time-dependent instantaneous correlation between W_i and W_j.

Equation (6.5) gives the expression for the forward swap rate at time t,

$$\begin{aligned} R_n^N(t) &= \frac{D_n^N(t)}{A_n^N(t)} \\ &= \frac{\sum_{j=n+1}^N p_j(t)\alpha_j L_j(t)}{\sum_{k=n+1}^N \alpha_k p_k(t)} \\ &= \sum_{j=n+1}^N \frac{\alpha_j p_j(t)}{\sum_{k=n+1}^N \alpha_k p_k(t)} L_j(t) \\ &\equiv \sum_{j=n+1}^N w_j(t) L_j(t) \end{aligned} \tag{C.1}$$

Now we will derive the dynamics of $\frac{dR_n^N(t)}{R_n^N(t)}$ for $0 \le t < T_0$. Hence all the Libors are still alive. To simplify the expressions, we will suppress the time index t. By multidimensional Itô's formula, we have

$$\begin{aligned} \frac{dR_n^N}{R_n^N} &= \sum_{j=1}^N \frac{dR_n^N}{dL_j} \frac{dL_j}{R_n^N} \\ &= \sum_{j=1}^N \frac{dR_n^N}{dL_j} \frac{L_j}{R_n^N} \frac{dL_j}{L_j} \end{aligned} \tag{C.2}$$

By Itô's product rule, we have

$$\begin{aligned} \frac{dR_n^N}{R_n^N} \cdot \frac{dR_n^N}{R_n^N} &= \sum_{j,k=1}^N \frac{dR_n^N}{dL_j} \frac{L_j}{R_n^N} \frac{dR_n^N}{dL_k} \frac{L_k}{R_n^N} \left(\frac{dL_j}{L_j} \cdot \frac{dL_k}{L_k} \right) \\ &= \sum_{j,k=1}^N \frac{dR_n^N}{dL_j} \frac{L_j}{R_n^N} \frac{dR_n^N}{dL_k} \frac{L_k}{R_n^N} \rho_{jk}\sigma_j\sigma_k dt \\ &\equiv \sum_{j,k=1}^N \xi_{jk}(t)\rho_{jk}(t)\sigma_j(t)\sigma_k(t)dt. \end{aligned} \tag{C.3}$$

We notice that $\xi_{jk}(t)$ is a stochastic quantity. Thus the righthand side of (C.3) is also stochastic and can only be observed pathwise.

In order to arrive at the Black-Scholes type constant volatility $\sigma_{n,N}^{BK}$ for the forward swap rate $\widetilde{R}_n^N$, one may assume the following dynamics for the forward swap rate under EMM $\mathbb{Q}$,

$$\frac{d\widetilde{R}_n^N(t)}{\widetilde{R}_n^N(t)} = \dots dt + \sigma_{n,N}(t)dW(t), \tag{C.4}$$

where W is a $\mathbb{Q}$ standard Brownian Motion, $\sigma_{n,N}$ is a deterministic time-dependent volatility function. Since the drift of $\frac{d\widetilde{R}_n^N(t)}{\widetilde{R}_n^N(t)}$ is irrelevant to the Black volatility, we omit it in the above expression. Assume that the volatility function satisfies necessary regularity conditions, one can change to an equivalent measure under which $\frac{d\widetilde{R}_n^N(t)}{\widetilde{R}_n^N(t)}$ dynamics has zero drift by the Girsanov theorem. Then one can derive that $\sigma_{n,N}^{BK}$ and $\sigma_{n,N}$ satisfy the following relation,

$$\left[\sigma_{n,N}^{BK}\right]^2 \times T_n = \int_0^{T_n} \left[\sigma_{n,N}(u)\right]^2 du. \tag{C.5}$$

Apply Itô's product rule to dynamics (C.4), we have

$$\frac{d\widetilde{R}_n^N}{\widetilde{R}_n^N} \cdot \frac{d\widetilde{R}_n^N}{\widetilde{R}_n^N} = \left[\sigma_{n,N}(u)\right]^2 dt. \tag{C.6}$$

Jäckel and Rebonato gave an argument in [37] that by freezing $\xi_{jk}(t)$ at time 0, one gets a good approximation to (C.6) by (C.3) from a practical perspective. We can write

$$\left[\sigma_{n,N}(t)\right]^2 \approx \xi_{jk}(0) \sum_{j,k=1}^{N} \rho_{jk}(t)\sigma_j(t)\sigma_k(t). \tag{C.7}$$

In order to calculate $\xi_{jk}(0)$, one needs to find the expression for $\frac{dR_n^N(t)}{dL_j(t)}$, for $0 \le t < T_0$. We start from (C.1). Again we will suppress the time variable t.

For $n = 1, \ldots, N$ and $i = 1, \ldots, N$,

$$\begin{aligned}
\frac{dR_n^N}{dL_i} &= \frac{1}{A_n^N}\frac{dD_n^N}{dL_i} - \frac{D_n^N}{A_n^{N2}}\frac{dA_n^N}{dL_i} \\
&= \frac{1}{A_n^N}\left(\alpha_i p_i 1_{\{i\ge n+1\}} + \sum_{j=n+1}^{N} \alpha_j L_j \frac{dp_j}{dL_i} 1_{\{i\le j\}}\right) - \frac{D_n^N}{A_n^{N2}}\left(\sum_{j=n+1}^{N} \alpha_j \frac{dp_j}{dL_i} 1_{\{i\le j\}}\right)
\end{aligned} \tag{C.8}$$

Notice that

$$\begin{aligned}
\frac{dp_j}{dL_i} 1_{\{i\le j\}} &= \frac{d\prod_{k=0}^{j} \frac{1}{1+\alpha_k L_k}}{dL_i} 1_{\{i\le j\}} \\
&= \prod_{k=0,\ldots,j.k\neq i} \frac{1}{1+\alpha_k L_k}\left(-\frac{\alpha_i}{(1+\alpha_i L_i)^2}\right) 1_{\{i\le j\}} \\
&= -p_j \frac{\alpha_i}{1+\alpha_i L_i} 1_{\{i\le j\}}
\end{aligned} \tag{C.9}$$

Substitute (C.9) into (C.8), we have

$$\begin{aligned}
\frac{dR_n^N}{dL_i} &= \frac{1}{A_n^N}\left(\alpha_i p_i 1_{\{i\ge n+1\}} - \sum_{j=n+1}^{N} \alpha_j L_j p_j \frac{\alpha_i}{1+\alpha_i L_i} 1_{\{i\le j\}}\right) - \frac{D_n^N}{A_n^{N2}}\left(\sum_{j=n+1}^{N} \alpha_j p_j \frac{\alpha_i}{1+\alpha_i L_i} 1_{\{i\le j\}}\right) \\
&= \frac{1}{A_n^N}\left(\alpha_i p_i 1_{\{i\ge n+1\}} - \frac{\alpha_i}{1+\alpha_i L_i} \sum_{j=(i-1)\vee n+1}^{N} \alpha_j L_j p_j\right) - \frac{D_n^N}{A_n^{N2}}\left(\frac{\alpha_i}{1+\alpha_i L_i} \sum_{j=(i-1)\vee n+1}^{N} \alpha_j p_j\right) \\
&= \frac{\alpha_i p_i}{A_n^N} 1_{\{i\ge n+1\}} + \frac{\alpha_i}{1+\alpha_i L_i}\left(\frac{D_n^N A_{(i-1)\vee n}^N - A_n^N D_{(i-1)\vee n}^N}{A_n^{N2}}\right),
\end{aligned} \tag{C.10}$$

where $x \vee y = \max(x, y)$. ξ_{jk} is then given by

$$\begin{aligned}
\xi_{jk} &= \frac{dR_n^N}{dL_j}\frac{L_j}{R_n^N} \times \frac{dR_n^N}{dL_k}\frac{L_k}{R_n^N} \\
&= \left[\frac{\alpha_j p_j L_j}{A_n^N} 1_{\{j\ge n+1\}} + \frac{\alpha_j L_j}{1+\alpha_j L_j}\left(\frac{D_n^N A_{(j-1)\vee n}^N - A_n^N D_{(j-1)\vee n}^N}{A_n^N D_n^N}\right)\right] \\
&\times \left[\frac{\alpha_k p_k L_k}{A_n^N} 1_{\{k\ge n+1\}} + \frac{\alpha_k L_k}{1+\alpha_k L_k}\left(\frac{D_n^N A_{(k-1)\vee n}^N - A_n^N D_{(k-1)\vee n}^N}{A_n^N D_n^N}\right)\right].
\end{aligned} \tag{C.11}$$

Using (C.7) and (C.5), we have the approximated Black volatility, $\sigma_{n,N}^{BK}$, of the forward swap rate, $R_n^N(0)$, in the LMM for each $n = 0, \ldots, N-1$.

D Libor Market Model Parameters

D.1 Parameters Based on the Paper by Hunter et al.

The volatility parameters are either taken from [36] or obtained through model calibration according to the data in [36]. We only show parameters corresponding to the first 10 tenor points.

Tenor spacing: $\Delta T = 0.25$.

Volatility and correlation parameters:

Parameter	a	b	c	d	α	β
Value	0.15717	0.71839	0.15017	-0.08914	0.1	0.1

FRAs and volatility fitting parameters:

Index i	$L_i(0)$	Φ_i
0	6.149%	–
1	6.121%	1.2824
2	6.155%	1.0257
3	6.178%	0.8809
4	6.214%	0.8831
5	6.249%	0.9683
6	6.265%	1.0214
7	6.281%	1.0229
8	6.289%	1.0120
9	6.279%	0.9963
10	6.263%	0.9930
$\vdots$	$\vdots$	$\vdots$

D.2 Parameters Based on the Thesis by Buitelaar

The volatility parameters are taken from [15]. We only show parameters corresponding to the first 10 tenor points.

Tenor spacing: $\Delta T = 1$.

Volatility and correlation parameters:

Parameter	a	b	c	d	α	β
Value	0.976	2.0	1.5	0.5	0.1	0.1

FRAs and volatility fitting parameters:

Index i	$L_i(0)$	Φ_i
0	2.3%	–
1	2.5%	0.153
2	2.7%	0.143
3	2.7%	0.140
4	3.1%	0.140
5	3.1%	0.139
6	3.3%	0.138
7	3.4%	0.137
8	3.6%	0.136
9	3.6%	0.135
10	3.8%	0.134
⋮	⋮	⋮

E Upper Bound Simulation Data

E.1 Bermudan Put in the Black-Scholes World

We present here the simulation results concerning the Bermudan Put in the Black-Scholes world.

Table E.1: Upper bound for a Bermudan Put in the Black-Scholes world. Regressors: c, S.

Run Number		Initial Asset Price		
		8(ITM)	10(ATM)	12(OTM)
Run1	LSM price	2.0764	0.9367	0.3885
	Gap	0.2661	0.6574	0.9346
	Gap(t-CI)	(0.2572,0.2750)	(0.6481, 0.6667)	(0.9228, 0.9463)
Run2	LSM price	2.0775	0.9379	0.3895
	Gap	0.2649	0.6429	0.9666
	Gap(t-CI)	(0.2590,0.2707)	(0.6339, 0.6519)	(0.9497, 0.9836)
Run3	LSM price	2.0771	0.9380	0.3890
	Gap	0.2597	0.6483	0.9346
	Gap(t-CI)	(0.2507,0.2686)	(0.6351,0.6615)	(0.9228,0.9464)

Table E.2: Upper bound for a Bermudan Put in the Black-Scholes world. Regressors: c, S, S^2.

Run Number		Initial Asset Price		
		8(ITM)	10(ATM)	12(OTM)
Run1	LSM price	2.0910	0.9458	0.3920
	Gap	0.1085	0.1593	0.1272
	Gap(t-CI)	(0.1040,0.1131)	(0.1539, 0.1647)	(0.1241, 0.1303)
Run2	LSM price	2.0911	0.9459	0.3925
	Gap	0.1064	0.1618	0.1212
	Gap(t-CI)	(0.1032,0.1097)	(0.1584, 0.1653)	(0.1172, 0.1253)
Run3	LSM price	2.0909	0.9461	0.3909
	Gap	0.1084	0.1637	0.1214
	Gap(t-CI)	(0.1052,0.1116)	(0.1614, 0.1661)	(0.1186, 0.1241)

Table E.3: Upper bound for a Bermudan Put in the Black-Scholes world. Regressors: $c, S, \ldots, S^3$.

Run Number		Initial Asset Price		
		8(ITM)	10(ATM)	12(OTM)
Run1	LSM price	2.0917	0.9479	0.3916
	Gap	0.0145	0.0091	0.3544
	Gap(t-CI)	(0.0137,0.0152)	(0.0083, 0.0099)	(0.3508, 0.3580)
Run2	LSM price	2.0929	0.9473	0.3923
	Gap	0.0150	0.0142	0.2734
	Gap(t-CI)	(0.0142,0.0157)	(0.0133, 0.0151)	(0.2701, 0.2767)
Run3	LSM price	2.0941	0.9469	0.3925
	Gap	0.0131	0.0116	0.1962
	Gap(t-CI)	(0.0123,0.0139)	(0.0108, 0.0123)	(0.1941, 0.1984)

Table E.4: Upper bound for a Bermudan Put in the Black-Scholes world. Regressors: $c, S, \ldots, S^4$.

Run Number		Initial Asset Price		
		8(ITM)	10(ATM)	12(OTM)
Run1	LSM price	2.0922	0.9466	0.3919
	Gap	0.0038	0.0450	0.1950
	Gap(t-CI)	(0.0033,0.0043)	(0.0435, 0.0464)	(0.1917, 0.1983)
Run2	LSM price	2.0931	0.9465	0.3916
	Gap	0.0028	0.0402	0.3519
	Gap(t-CI)	(0.0023,0.0034)	(0.0376, 0.0428)	(0.3479, 0.3559)
Run3	LSM price	2.0921	0.9478	0.3913
	Gap	0.0036	0.0566	0.1811
	Gap(t-CI)	(0.0031,0.0042)	(0.0539, 0.0593)	(0.1767, 0.1855)

Table E.5: Upper bound for a Bermudan Put in the Black-Scholes world. Regressors: $c, S, \ldots, S^5$.

Run Number		Initial Asset Price		
		8(ITM)	10(ATM)	12(OTM)
Run1	LSM price	2.0947	0.9470	0.3924
	Gap	0.0149	0.1924	0.1909
	Gap(t-CI)	(0.0136,0.0161)	(0.1848, 0.1999)	(0.1879, 0.1938)
Run2	LSM price	2.0932	0.9479	0.3923
	Gap	0.0172	0.2371	0.1631
	Gap(t-CI)	(0.0150,0.0194)	(0.2327, 0.2416)	(0.1595, 0.1666)
Run3	LSM price	2.0932	0.9480	0.3925
	Gap	0.0196	0.1700	0.6509
	Gap(t-CI)	(0.0182,0.0210)	(0.1647, 0.1753)	(0.6436, 0.6582)

Table E.6: Upper bound for a Bermudan Put in the Black-Scholes world. Regressors: $c, S, \ldots, S^6$.

Run Number		Initial Asset Price		
		8(ITM)	10(ATM)	12(OTM)
Run1	LSM price	2.0931	0.9472	0.3919
	Gap	0.0415	0.1516	0.7292
	Gap(t-CI)	(0.0388,0.0442)	(0.1489, 0.1543)	(0.7195, 0.7389)
Run2	LSM price	2.0936	0.9473	0.3923
	Gap	0.0457	0.2540	0.1623
	Gap(t-CI)	(0.0444,0.0469)	(0.2497, 0.2584)	(0.1595, 0.1651)
Run3	LSM price	2.0925	0.9471	0.3922
	Gap	0.0494	0.2429	0.5809
	Gap(t-CI)	(0.0468,0.0519)	(0.2394, 0.2464)	(0.5753, 0.5866)

E.2 Bermudan Put in the Heston Model

We present here the simulation results concerning the Bermudan Put in the Heston model.

Table E.7: Upper bound convergence result for a Bermudan Put in the Heston model. $N_{out} = 10^3$, $N_{in} = 10^3$, 10 simulation average. Time for a run is about 80 min.

Run Number		Strike		
		8(OTM)	10(ATM)	12(ITM)
Run1	LSM price	0.3709	1.1024	2.3448
	Gap	0.2266	0.0082	0.0139
	Gap(t-CI)	(0.2230,0.2301)	(0.0070, 0.0093)	(0.0131, 0.0148)
Run2	LSM price	0.3703	1.0993	2.3439
	Gap	0.1779	0.0246	0.0146
	Gap(t-CI)	(0.1755,0.1804)	(0.0231, 0.0260)	(0.0133, 0.0159)
Run3	LSM price	0.3704	1.1021	2.3435
	Gap	0.2426	0.0105	0.0126
	Gap(t-CI)	(0.2396,0.2456)	(0.0097, 0.0113)	(0.0112, 0.0141)

Table E.8: Upper bound convergence result for a Bermudan Put in the Heston model. $N_{out} = 10^3$, $N_{in} = 10^4$, 10 simulation average. Time for a run is about 490 min.

Run Number		Strike		
		8(OTM)	10(ATM)	12(ITM)
Run1	LSM price	0.3699	1.1006	2.3413
	Gap	0.4457	0.0174	0.0089
	Gap(t-CI)	(0.4383,0.4532)	(0.0169, 0.0179)	(0.0086, 0.0093)
Run2	LSM price	0.3711	1.1015	2.3443
	Gap	0.1806	0.0052	0.0087
	Gap(t-CI)	(0.1763,0.1848)	(0.0048, 0.0056)	(0.0078, 0.0096)
Run3	LSM price	0.3702	1.1005	2.3429
	Gap	0.0498	0.0094	0.0079
	Gap(t-CI)	(0.0478,0.0518)	(0.0083, 0.0104)	(0.0075, 0.0084)

Table E.9: Upper bound convergence result for a Bermudan Put in the Heston model. $N_{out} = 10^4$, $N_{in} = 10^3$, 10 simulation average. Time for a run is about 800 min.

Run Number		Strike		
		8(OTM)	10(ATM)	12(ITM)
Run1	LSM price	0.3700	1.1011	2.3423
	Gap	0.2204	0.0148	0.0153
	Gap(t-CI)	(0.2193,0.2214)	(0.0143, 0.0153)	(0.0150, 0.0156)
Run2	LSM price	0.3717	1.0997	2.3421
	Gap	0.0695	0.0063	0.0147
	Gap(t-CI)	(0.069,0.070)	(0.0060, 0.0065)	(0.0144, 0.0150)

Table E.10: Upper bound for a Bermudan put in the Heston model. Regressors: $c, S, \sqrt{v}$.

Run Number		Strike		
		8(OTM)	10(ATM)	12(ITM)
Run1	LSM price	0.3688	1.0939	2.3276
	Gap	0.9565	0.7263	0.3916
	Gap(t-CI)	(0.9439,0.9691)	(0.7218, 0.7309)	(0.3742, 0.4090)
Run2	LSM price	0.3689	1.0939	2.3274
	Gap	0.9772	0.7254	0.3962
	Gap(t-CI)	(0.9651,0.9893)	(0.7116, 0.7392)	(0.3865, 0.4059)
Run3	LSM price	0.3693	1.0935	2.3287
	Gap	0.9681	0.7234	0.3944
	Gap(t-CI)	(0.9564,0.9799)	(0.7125,0.7342)	(0.3814,0.4075)

Table E.11: Upper bound for a Bermudan put in the Heston model. Regressors: $c, S, \sqrt{v}, S \times \sqrt{v}$.

Run Number		Strike		
		8(OTM)	10(ATM)	12(ITM)
Run1	LSM price	0.3689	1.0947	2.3278
	Gap	0.8410	0.6444	0.3557
	Gap(t-CI)	(0.8329,0.8490)	(0.6333, 0.6554)	(0.3451, 0.3663)
Run2	LSM price	0.3691	1.0944	2.3281
	Gap	0.8545	0.6515	0.3671
	Gap(t-CI)	(0.8440,0.8650)	(0.6401, 0.6630)	(0.3572, 0.3770)
Run3	LSM price	0.3689	1.0939	2.3289
	Gap	0.7653	0.6477	0.3632
	Gap(t-CI)	(0.7525,0.7781)	(0.6372,0.6576)	(0.3553,0.3711)

Table E.12: Upper bound for a Bermudan put in the Heston model. Regressors: $c, S, \ldots, S^4$.

Run Number		Strike		
		8(OTM)	10(ATM)	12(ITM)
Run1	LSM price	0.3679	1.0993	2.3381
	Gap	0.4102	0.0133	0.0196
	Gap(t-CI)	(0.4059,0.4145)	(0.0124, 0.0141)	(0.0183, 0.0209)
Run2	LSM price	0.3699	1.0992	2.3395
	Gap	0.1568	0.0274	0.0186
	Gap(t-CI)	(0.1543,0.1593)	(0.0256, 0.0293)	(0.0176, 0.0195)
Run3	LSM price	0.3695	1.0991	2.3387
	Gap	0.4584	0.0140	0.0188
	Gap(t-CI)	(0.4540,0.4628)	(0.0126,0.0154)	(0.0180,0.0195)

Table E.13: Upper bound for a Bermudan put in the Heston model. Regressors: $c, S, \ldots, S^8$.

Run Number		Strike		
		8(OTM)	10(ATM)	12(ITM)
Run1	LSM price	0.3698	1.0986	2.3391
	Gap	0.1666	0.2992	0.1176
	Gap(t-CI)	(0.1641,0.1691)	(0.2917, 0.3067)	(0.1127, 0.1225)
Run2	LSM price	0.3702	1.0991	2.3391
	Gap	0.6808	0.2969	0.1066
	Gap(t-CI)	(0.6755,0.6862)	(0.2930, 0.3007)	(0.1029, 0.1104)
Run3	LSM price	0.3698	1.0983	2.3385
	Gap	0.5362	0.1924	0.1380
	Gap(t-CI)	(0.5294,0.5430)	(0.1865,0.1984)	(0.1320,0.1439)

Table E.14: Upper bound for a Bermudan put in the Heston model. Regressors: $c, S, \ldots, S^4, \sqrt{v}$.

Run Number		Strike		
		8(OTM)	10(ATM)	12(ITM)
Run1	LSM price	0.3696	1.0984	2.3388
	Gap	0.5346	0.0116	0.0214
	Gap(t-CI)	(0.5336,0.5356)	(0.0108, 0.0124)	(0.0202, 0.0225)
Run2	LSM price	0.3700	1.0987	2.3385
	Gap	0.4067	0.0311	0.0202
	Gap(t-CI)	(0.4053,0.4071)	(0.0292, 0.0330)	(0.0192, 0.0212)
Run3	LSM price	0.3701	1.0988	2.3387
	Gap	0.2798	0.0266	0.0205
	Gap(t-CI)	(0.2765,0.2831)	(0.0256,0.0277)	(0.0197,0.0212)

Table E.15: Upper bound with the Heston model. Regressors: $c, S, \ldots, S^4, \sqrt{v}, v$.

Run Number		Strike		
		8(OTM)	10(ATM)	12(ITM)
Run1	LSM price	0.3697	1.0989	2.3397
	Gap	0.0448	0.0324	0.0182
	Gap(t-CI)	(0.0434,0.0462)	(0.0312, 0.0335)	(0.0170, 0.0194)
Run2	LSM price	0.3700	1.0992	2.3393
	Gap	0.4539	0.0134	0.0186
	Gap(t-CI)	(0.4485,0.4592)	(0.0128, 0.0140)	(0.0176, 0.0199)
Run3	LSM price	0.3695	1.0983	2.3394
	Gap	0.1320	0.0266	0.0190
	Gap(t-CI)	(0.1292,0.1349)	(0.0252,0.0281)	(0.0182,0.0198)

Table E.16: Upper bound for a Bermudan put in the Heston model. Regressors: $c, S, \ldots, S^4, \sqrt{v}, \ldots, v^2$.

Run Number		Strike		
		8(OTM)	10(ATM)	12(ITM)
Run1	LSM price	0.3700	1.0979	2.3402
	Gap	0.0960	0.0616	0.0176
	Gap(t-CI)	(0.0936,0.0985)	(0.0591, 0.0641)	(0.0163, 0.0190)
Run2	LSM price	0.3694	1.0987	2.3398
	Gap	0.0801	0.0429	0.0175
	Gap(t-CI)	(0.0781,0.0822)	(0.0404, 0.0455)	(0.0163, 0.0188)
Run3	LSM price	0.3696	1.0985	2.3393
	Gap	0.6389	0.0148	0.0208
	Gap(t-CI)	(0.6338,0.6439)	(0.0137, 0.0158)	(0.0196,0.0220)

Table E.17: Upper bound for a Bermudan put in the Heston model. Regressors: $c, S, \ldots, S^4, S \times \sqrt{v}$.

Run Number		Strike		
		8(OTM)	10(ATM)	12(ITM)
Run1	LSM price	0.3698	1.0988	2.3387
	Gap	0.4425	0.0108	0.0207
	Gap(t-CI)	(0.0437,0.0448)	(0.0102, 0.0115)	(0.0196, 0.0219)
Run2	LSM price	0.3703	1.0991	2.3394
	Gap	0.5511	0.0439	0.0199
	Gap(t-CI)	(0.5466,0.5556)	(0.0420, 0.0458)	(0.0186, 0.0211)
Run3	LSM price	0.3697	1.0988	2.3394
	Gap	0.7018	0.0303	0.0191
	Gap(t-CI)	(0.6956,0.7080)	(0.0288, 0.0319)	(0.0180,0.0202)

Table E.18: Upper bound for a Bermudan put in the Heston model. Regressors: $c, S, \ldots, S^4, \sqrt{v}, S \times \sqrt{v}$.

Run Number		Strike		
		8(OTM)	10(ATM)	12(ITM)
Run1	LSM price	0.3708	1.100	2.3434
	Gap	0.4835	0.0361	0.0135
	Gap(t-CI)	(0.4790,0.4879)	(0.0334, 0.0387)	(0.0123, 0.0148)
Run2	LSM price	0.3704	1.1011	2.3434
	Gap	0.1795	0.0113	0.0139
	Gap(t-CI)	(0.1780,0.1811)	(0.0101, 0.0125)	(0.0130, 0.0148)
Run3	LSM price	0.3704	1.1011	2.3430
	Gap	0.0640	0.0219	0.0139
	Gap(t-CI)	(0.0629,0.0650)	(0.0202, 0.0235)	(0.0129,0.0150)

Table E.19: Upper bound for a Bermudan put in the Heston model. Regressors: $c, S, \ldots, S^4, \sqrt{v}, v, S \times \sqrt{v}, S \times v, S^2 \times \sqrt{v}$.

Run Number		Strike		
		8(OTM)	10(ATM)	12(ITM)
Run1	LSM price	0.3701	1.1007	2.3428
	Gap	0.2308	0.0188	0.0138
	Gap(t-CI)	(0.2287,0.2330)	(0.0176, 0.0199)	(0.0130, 0.0145)
Run2	LSM price	0.3703	1.1014	2.3436
	Gap	0.0487	0.0612	0.0133
	Gap(t-CI)	(0.0468,0.0505)	(0.0585, 0.0638)	(0.0119, 0.0147)
Run3	LSM price	0.3705	1.1008	2.3431
	Gap	0.0259	0.0301	0.0130
	Gap(t-CI)	(0.0255,0.0262)	(0.0283, 0.0319)	(0.0122,0.0138)

E.3 Cancellable Swaption in the Libor Market Model

We present here the simulation results concerning the cancellable receiver swaption in the Libor market model. The simulation configuration is discussed in Section 7.5.

Table E.20: Upper bound for a cancellable receiver swaption. Regressors: c, fixed leg, floating leg, floating leg tilt. About 13 hours for 3 simulation runs.

Run Number		Coupon rate k		
		214(OTM)	314(ATM)	414(ITM)
Run1	LSM price	64.2	453.2	1082.3
	Gap	9.1	15.6	11.6
	Gap(t-CI)	(8.4, 9.8)	(14.2, 17.1)	(10, 13.2)
Run2	LSM price	64.3	453.1	1082.3
	Gap	9.5	15.8	12.4
	Gap(t-CI)	(8.9, 10)	(15.1, 16.4)	(11.7, 13.1)
Run3	LSM price	64.3	453.2	1082.2
	Gap	9.3	16.5	13
	Gap(t-CI)	(8.8, 9.9)	(14.0, 19.1)	(11.1, 14.9)

Table E.21: Upper bound for a cancellable receiver swaption. Regressors: c, fixed leg, floating leg, all living Libors. About 13 hours for 3 simulation runs.

Run Number		Coupon rate k		
		214(OTM)	314(ATM)	414(ITM)
Run1	LSM price	69.9	463.2	1090.6
	Gap	2.8	4.4	3.5
	Gap(t-CI)	(2.2, 3.5)	(4.3, 4.5)	(2.9, 4.1)
Run2	LSM price	69.9	463.3	1090.5
	Gap	2.5	4.5	3.8
	Gap(t-CI)	(2.2, 2.9)	(4.0, 5.1)	(2.4, 5.1)
Run3	LSM price	69.9	463.2	1090.7
	Gap	2.5	5.0	3.7
	Gap(t-CI)	(2.2, 3.0)	(3.5, 6.4)	(2.6, 4.9)

Table E.22: Upper bound for a cancellable receiver swaption. Regressors: c, fixed leg, floating leg, all living Libors, all living Libors squared. About 13 hours for 3 simulation runs.

Run Number		Coupon rate k		
		214(OTM)	314(ATM)	414(ITM)
Run1	LSM price	70.3	463.9	1091.0
	Gap	2.0	3.2	3.6
	Gap(t-CI)	(1.7, 2.2)	(2.6, 3.8)	(2.5, 4.6)
Run2	LSM price	70.2	464.0	1091.0
	Gap	1.8	3.4	3.1
	Gap(t-CI)	(1.7, 2.0)	(2.8, 4.0)	(2.4, 3.9)
Run3	LSM price	70.3	464.0	1091.0
	Gap	1.9	3.6	3.1
	Gap(t-CI)	(1.8, 2.0)	(2.9, 4.3)	(2.4, 3.7)

Table E.23: Upper bound for a cancellable receiver swaption. Regressors: c, fixed leg, floating leg, all living Libors, cross products of all living Libors. About 13 hours for 3 simulation runs.

Run Number		Coupon rate k		
		214(OTM)	314(ATM)	414(ITM)
Run1	LSM price	70.2	464.0	1091.1
	Gap	1.9	3.5	3.1
	Gap(t-CI)	(1.7, 2.1)	(3.2, 3.9)	(2.5, 3.6)
Run2	LSM price	70.2	463.8	1091.1
	Gap	2.0	3.3	2.8
	Gap(t-CI)	(1.7, 2.4)	(3.1, 3.5)	(2.3, 3.3)
Run3	LSM price	71.0	465.6	1092.6
	Gap	1.0	1.4	1.3
	Gap(t-CI)	(0.9, 1.2)	(1.2, 1.6)	(0.8, 1.8)

Table E.24: Upper bound for a cancellable receiver swaption. Regressors: c, forward swap rate $R_n^N(T_n)$, fixed leg, floating leg, next Libor rate. About 13 hours for 3 simulation runs.

Run Number		Coupon rate k		
		214(OTM)	314(ATM)	414(ITM)
Run1	LSM price	69.8	463.3	1090.6
	Gap	2.6	4.6	4.1
	Gap(t-CI)	(2.3, 2.8)	(3.6, 5.6)	(2.7, 5.4)
Run2	LSM price	69.8	463.4	1090.5
	Gap	2.8	4.7	4.4
	Gap(t-CI)	(2.2, 3.5)	(3.6, 5.7)	(2.8, 6.0)
Run3	LSM price	69.9	463.4	1090.7
	Gap	2.6	4.7	3.8
	Gap(t-CI)	(2.3, 2.8)	(3.5, 5.9)	(2.5, 5.2)

Table E.25: Upper bound for a cancellable receiver swaption. Regressors: c, forward swap rate $R_n^N(T_n)$, fixed leg, floating leg, next Libor rate, $R_n^N(T_n)^2$. About 13 hours for 3 simulation runs.

Run Number		Coupon rate k		
		214(OTM)	314(ATM)	414(ITM)
Run1	LSM price	70.6	464.6	1091.1
	Gap	1.5	2.5	2.9
	Gap(t-CI)	(1.3, 1.8)	(1.9, 3.0)	(2.3, 3.5)
Run2	LSM price	70.5	464.7	1091.2
	Gap	1.6	3.1	3.1
	Gap(t-CI)	(1.5, 1.8)	(2.7, 3.4)	(2.2, 4.1)
Run3	LSM price	70.6	464.5	1091.1
	Gap	1.6	2.4	2.6
	Gap(t-CI)	(1.5, 1.8)	(1.9, 2.8)	(2.1, 3.0)

Table E.26: Upper bound for a cancellable receiver swaption. Regressors: c, forward swap rate $R_n^N(T_n)$, fixed leg, floating leg, next Libor rate, $R_n^N(T_n)^2$, $R_n^N(T_n)^3$. About 13 hours for 3 simulation runs.

Run Number		Coupon rate k		
		214(OTM)	314(ATM)	414(ITM)
Run1	LSM price	70.9	465.4	1092.5
	Gap	1.2	1.5	1.1
	Gap(t-CI)	(1.0, 1.4)	(1.4, 1.7)	(0.9, 1.4)
Run2	LSM price	70.9	465.3	1092.5
	Gap	1.3	1.5	1.5
	Gap(t-CI)	(1.0, 1.5)	(1.3, 1.7)	(1.2, 1.7)
Run3	LSM price	71.0	465.4	1092.4
	Gap	1.3	1.5	1.3
	Gap(t-CI)	(1.0, 1.5)	(1.3, 1.7)	(1.1, 1.5)

Table E.27: Upper bound for a cancellable receiver swaption. Regressors: c, forward swap rate $R_n^N(T_n)$, fixed leg, floating leg, next Libor rate, $R_n^N(T_n)^2$,. . . ,$R_n^N(T_n)^4$. About 13 hours for 3 simulation runs.

Run Number		Coupon rate k		
		214(OTM)	314(ATM)	414(ITM)
Run1	LSM price	71.4	465.4	1092.8
	Gap	0.6	1	0.9
	Gap(t-CI)	(0.6, 0.7)	(0.9, 1.2)	(0.8, 1.1)
Run2	LSM price	71.4	466.0	1092.8
	Gap	0.7	0.9	0.7
	Gap(t-CI)	(0.5, 0.8)	(0.7, 1.1)	(0.6, 0.9)
Run3	LSM price	71.3	466.0	1092.9
	Gap	0.9	0.8	0.9
	Gap(t-CI)	(0.7, 1.0)	(0.7, 0.9)	(0.7, 1.2)

Table E.28: Upper bound for a cancellable receiver swaption. Regressors: c, forward swap rate $R_n^N(T_n)$, fixed leg, floating leg, next Libor rate, $R_n^N(T_n)^2$,. . . ,$R_n^N(T_n)^5$. About 13 hours for 3 simulation runs.

Run Number		Coupon rate k		
		214(OTM)	314(ATM)	414(ITM)
Run1	LSM price	71.6	466.2	1092.9
	Gap	0.4	0.6	0.7
	Gap(t-CI)	(0.3, 0.5)	(0.5, 0.7)	(0.5, 1.0)
Run2	LSM price	71.6	466.3	1092.9
	Gap	0.4	0.6	0.8
	Gap(t-CI)	(0.2, 0.5)	(0.5, 0.8)	(0.6, 1.0)
Run3	LSM price	71.7	466.5	1092.9
	Gap	0.5	0.6	0.7
	Gap(t-CI)	(0.4, 0.5)	(0.5, 0.8)	(0.5, 0.8)

Table E.29: Upper bound for a cancellable receiver swaption. Regressors: c, forward swap rate $R_n^N(T_n)$, fixed leg, floating leg, next Libor rate, $R_n^N(T_n)^2$,. . . ,$R_n^N(T_n)^6$. About 13 hours for 3 simulation runs.

Run Number		Coupon rate k		
		214(OTM)	314(ATM)	414(ITM)
Run1	LSM price	71.8	466.3	1093.0
	Gap	0.3	0.5	0.7
	Gap(t-CI)	(0.2, 0.3)	(0.3, 0.6)	(0.5, 1.0)
Run2	LSM price	71.7	466.4	1092.9
	Gap	0.3	0.6	0.7
	Gap(t-CI)	(0.2, 0.3)	(0.5, 0.7)	(0.5, 0.9)
Run3	LSM price	71.7	466.3	1093.0
	Gap	0.2	0.5	0.7
	Gap(t-CI)	(0.2, 0.3)	(0.4, 0.7)	(0.6, 0.8)

E.4 Cancellable Inverse Floater in the Libor Market Model

We present here the simulation results concerning the cancellable receiver Inverse Floater in the Libor market model. The simulation configuration is discussed in Section 7.5.

Table E.30: Upper bound for a cancellable receiver inverse floater. Regressors: c, fixed leg, floating leg, floating tilt. About 14 hours for 3 simulation runs.

Run Number		Strike s		
		485.2(OTM)	585.2(ATM)	685.2(ITM)
Run1	LSM price	270.7	673.2	1222.8
	Gap	42.7	48.2	43.9
	Gap(t-CI)	(39.9, 45.4)	(45.3, 51.1)	(40.1, 47.8)
Run2	LSM price	270.8	673.7	1222.4
	Gap	40	49	43.2
	Gap(t-CI)	(38.2, 41.8)	(46.5, 51.5)	(42.3, 44.1)
Run3	LSM price	270.9	673.4	1222.5
	Gap	40.8	47.7	44.8
	Gap(t-CI)	(38.1, 43.5)	(45.0, 50.3)	(42.4, 47.2)

Table E.31: Upper bound for a cancellable receiver inverse floater. Regressors: c, fixed leg, floating leg, floating leg tilt, floating leg square, floating tilt square, floating leg×floating tilt. About 17 hours for 3 simulation runs.

Run Number		Strike s		
		485.2(OTM)	585.2(ATM)	685.2(ITM)
Run1	LSM price	293.5	702.5	1251.5
	Gap	15.9	15.2	11.2
	Gap(t-CI)	(12.8, 19.0)	(12.8, 17.6)	(9.2, 13.2)
Run2	LSM price	293.4	702.4	1251.2
	Gap	15.6	15.1	12.2
	Gap(t-CI)	(12.9, 18.3)	(13.6, 16.5)	(10.4, 14.0)
Run3	LSM price	293.3	702.6	1251.3
	Gap	14.8	13.9	11.1
	Gap(t-CI)	(13.0, 16.5)	(12.2, 15.6)	(9.7, 12.6)

Table E.32: Upper bound for a cancellable receiver inverse floater. Regressors: c, coupon leg, floating leg, all living Libor rates. About 17 hours for 3 simulation runs.

Run Number		Strike s		
		485.2(OTM)	585.2(ATM)	685.2(ITM)
Run1	LSM price	303.0	709.8	1255.3
	Gap	1.2	4.5	6.9
	Gap(t-CI)	(1.1, 1.3)	(3.6, 4.9)	(4.7, 9.0)
Run2	LSM price	302.9	709.7	1255.1
	Gap	1.2	4.3	5.5
	Gap(t-CI)	(1.0, 1.3)	(4.1, 4.5)	(4.1, 6.9)
Run3	LSM price	303.0	709.8	1255.1
	Gap	1.4	4.2	6.1
	Gap(t-CI)	(0.8, 1.9)	(3.3, 5.2)	(4.9, 7.3)

Table E.33: Upper bound for a cancellable receiver inverse floater. Regressors: c, coupon leg, floating leg, all living Libor rates, all living Libor rates squared. About 17 hours for 3 simulation runs.

Run Number		Strike s		
		485.2(OTM)	585.2(ATM)	685.2(ITM)
Run1	LSM price	303.1	711.4	1256.6
	Gap	1.0	2.6	4.3
	Gap(t-CI)	(0.8, 1.1)	(2.3, 3.0)	(3.1, 5.4)
Run2	LSM price	303.4	711.5	1256.5
	Gap	0.8	2.3	4.7
	Gap(t-CI)	(0.7, 1.0)	(1.6, 3.0)	(3.9, 5.5)
Run3	LSM price	303.3	711.5	1256.6
	Gap	0.9	2.5	5.3
	Gap(t-CI)	(0.7, 1.0)	(1.8, 3.2)	(4.0, 6.5)

Table E.34: Upper bound for a cancellable receiver inverse floater. Regressors: c, coupon leg, floating leg, all living Libor rates, cross-products of all living Libor rates. About 17 hours for 3 simulation runs.

Run Number		Strike s		
		485.2(OTM)	585.2(ATM)	685.2(ITM)
Run1	LSM price	303.3	712.0	1258.2
	Gap	0.8	1.7	2.9
	Gap(t-CI)	(0.6, 0.9)	(1.5, 2.0)	(2.8, 3.0)
Run2	LSM price	303.3	712.1	1258.1
	Gap	0.8	1.7	2.7
	Gap(t-CI)	(0.7, 1.0)	(1.4, 2.1)	(2.1, 3.4)
Run3	LSM price	303.3	712.1	1258.2
	Gap	0.8	2.0	2.9
	Gap(t-CI)	(0.7, 1.0)	(1.8, 2.2)	(2.1, 3.7)

Table E.35: Upper bound for a cancellable receiver inverse floater. Regressors: c, forward swap rate $R_n^N(T_n)$, fixed leg, floating leg, next Libor rate. About 14 hours for 3 simulation runs.

Run Number		Strike s		
		485.2(OTM)	585.2(ATM)	685.2(ITM)
Run1	LSM price	292.7	701.4	1250.2
	Gap	16.7	15.7	11.6
	Gap(t-CI)	(15.6, 17.9)	(13.6, 17.8)	(8.8, 14.4)
Run2	LSM price	292.6	701.1	1250.2
	Gap	15.8	16.5	13.4
	Gap(t-CI)	(13.1, 18.5)	(14.3, 18.7)	(10.3, 16.4)
Run3	LSM price	292.3	701.3	1250.2
	Gap	15.8	15.8	15
	Gap(t-CI)	(12.3, 17.1)	(13.4, 18.2)	(12.8, 17.2)

Table E.36: Upper bound for a cancellable receiver inverse floater. Regressors: c, forward swap rate $R_n^N(T_n)$, coupon leg, floating leg, next Libor rate. About 17 hours for 3 simulation runs.

Run Number		Strike s		
		485.2(OTM)	585.2(ATM)	685.2(ITM)
Run1	LSM price	302.7	709.4	1253.1
	Gap	1.7	4.1	7.7
	Gap(t-CI)	(1.5, 1.9)	(3.7, 4.5)	(6.8, 8.7)
Run2	LSM price	302.5	709.8	1253.0
	Gap	1.7	3.9	8
	Gap(t-CI)	(1.3, 2.0)	(3.1, 4.8)	(6.3, 9.7)
Run3	LSM price	302.6	709.7	1253.3
	Gap	16	4.3	8.2
	Gap(t-CI)	(15.0, 17.0)	(3.8, 4.9)	(7.4, 9.0)

Table E.37: Upper bound for a cancellable receiver inverse floater. Regressors: c, forward swap rate $R_n^N(T_n)$, coupon leg, floating leg, next Libor rate, $R_n^N(T_n)^2$. About 17 hours for 3 simulation runs.

Run Number		Strike s		
		485.2(OTM)	585.2(ATM)	685.2(ITM)
Run1	LSM price	302.8	710.1	1254.6
	Gap	1.3	4.5	6.5
	Gap(t-CI)	(1.1, 1.4)	(3.7, 5.3)	(5.3, 7.8)
Run2	LSM price	302.7	709.8	1254.4
	Gap	1.3	4.1	6.4
	Gap(t-CI)	(0.9, 1.7)	(3.3, 4.8)	(5.3, 7.6)
Run3	LSM price	302.8	709.8	1254.6
	Gap	1.3	4.4	6.6
	Gap(t-CI)	(1.2, 1.4)	(3.9, 5.0)	(5.7, 7.4)

Table E.38: Upper bound for a cancellable receiver inverse floater. Regressors: c, forward swap rate $R_n^N(T_n)$, coupon leg, floating leg, next Libor rate, coupon leg square. About 17 hours for 3 simulation runs.

Run Number		Strike s		
		485.2(OTM)	585.2(ATM)	685.2(ITM)
Run1	LSM price	302.7	712.4	1259.2
	Gap	1.2	1.5	1.8
	Gap(t-CI)	(0.9, 1.5)	(1.3, 1.7)	(1.5, 2.1)
Run2	LSM price	302.8	712.1	1258.8
	Gap	1.3	1.4	1.6
	Gap(t-CI)	(1.0, 1.6)	(1.0, 1.8)	(1.4, 1.7)
Run3	LSM price	303.0	712.1	1259.2
	Gap	1.4	1.5	1.8
	Gap(t-CI)	(1.1, 1.7)	(1.1, 1.8)	(1.4, 2.2)

Table E.39: Upper bound for a cancellable receiver inverse floater. Regressors: c, forward swap rate $R_n^N(T_n)$, coupon leg, floating leg, next Libor rate, coupon leg square, coupon leg cubic. About 17 hours for 3 simulation runs.

Run Number		Strike s		
		485.2(OTM)	585.2(ATM)	685.2(ITM)
Run1	LSM price	302.7	712.6	1258.8
	Gap	1.3	1.2	1.6
	Gap(t-CI)	(1.0, 1.5)	(0.9, 1.5)	(1.0, 2.3)
Run2	LSM price	302.8	712.3	1258.7
	Gap	1.3	1.4	1.8
	Gap(t-CI)	(1.1, 1.4)	(1.2, 1.6)	(1.4, 2.1)
Run3	LSM price	302.8	712.2	1258.9
	Gap	1.4	1.5	1.8
	Gap(t-CI)	(1.2, 1.7)	(1.2, 1.8)	(1.5, 2.0)

E.5 Cancellable Snowball in the Libor Market Model

We present here the simulation results concerning the cancellable receiver snowball in the Libor market model. The simulation configuration is discussed in Section 7.5.

Table E.40: Upper bound for a cancellable snowball. Regressors: c, fixed leg, floating leg, floating leg tilt, next cash flow. About 15 hours for 3 simulation runs.

Run Number		Virtual Coupon Rate C_0		
		200(OTM)	300(ATM)	400(ITM)
Run1	LSM price	795.6	1210.4	1692.0
	Gap	60.2	53.0	51.7
	Gap(t-CI)	(57.5, 62.9)	(50.1, 55.9)	(49.1, 54.2)
Run2	LSM price	795.2	1209.7	1692.6
	Gap	56.3	53.8	51.1
	Gap(t-CI)	(53.9, 58.7)	(49.3, 58.3)	(50.0, 52.2)
Run3	LSM price	795.8	1210.5	1692.1
	Gap	58.4	53.3	52.8
	Gap(t-CI)	(53.9, 63.0)	(47.8, 58.7)	(48.5, 57.2)

Table E.41: Upper bound for a cancellable snowball. Regressors: c, fixed leg, floating leg, floating leg tilt, next cash flow, floating leg squared, floating tilt squared, floating leg×floating tilt. About 15 hours for 3 simulation runs.

Run Number		Virtual Coupon Rate C_0		
		200(OTM)	300(ATM)	400(ITM)
Run1	LSM price	824.7	1235.0	1712.5
	Gap	28.5	32.7	31.5
	Gap(t-CI)	(25.8, 31.1)	(28.3, 37.0)	(26.7, 36.3)
Run2	LSM price	824.3	1235.0	1713.5
	Gap	34	32.3	34.9
	Gap(t-CI)	(28.5, 39.5)	(25.4, 39.1)	(29.2, 40.6)
Run3	LSM price	824.3	1234.5	1712.9
	Gap	32.0	32.4	35.9
	Gap(t-CI)	(29.8, 34.3)	(27.2, 37.6)	(31.8, 39.9)

Table E.42: Upper bound for a cancellable snowball. Regressors: c, fixed leg, floating leg, all living Libors. About 17 hours for 3 simulation runs.

Run Number		Virtual Coupon Rate C_0		
		200(OTM)	300(ATM)	400(ITM)
Run1	LSM price	820.7	1231.2	1709.9
	Gap	30.6	34.7	35.8
	Gap(t-CI)	(28.2, 32.9)	(28.0, 41.4)	(29.6, 42.1)
Run2	LSM price	821.2	1231.6	1710.6
	Gap	32.2	33.5	34.0
	Gap(t-CI)	(30.0, 34.4)	(30.5, 36.5)	(28.9, 39.1)
Run3	LSM price	821.6	1230.6	1710.0
	Gap	31.3	35.0	35.1
	Gap(t-CI)	(29.6, 33.0)	(33.4, 36.6)	(29.6, 40.7)

Table E.43: Upper bound for a cancellable snowball. Regressors: c, fixed leg, floating leg, all living Libors, all living Libors squared. About 17 hours for 3 simulation runs.

Run Number		Virtual Coupon Rate C_0		
		200(OTM)	300(ATM)	400(ITM)
Run1	LSM price	823.6	1233.3	1711.8
	Gap	28.4	32.1	34.0
	Gap(t-CI)	(26.2, 30.6)	(30.7, 33.5)	(29.6, 38.5)
Run2	LSM price	823.6	1232.7	1711.8
	Gap	29.2	32.8	34.7
	Gap(t-CI)	(26.1, 32.2)	(30.8, 34.8)	(29.0, 40.4)
Run3	LSM price	825.1	1234.0	1711.6
	Gap	28.4	34.5	33.9
	Gap(t-CI)	(24.3, 32.4)	(30.7, 38.2)	(32.2, 35.6)

Table E.44: Upper bound for a cancellable snowball. Regressors: c, fixed leg, floating leg, all living Libors, cross product of all living Libors. About 17 hours for 3 simulation runs.

Run Number		Virtual Coupon Rate C_0		
		200(OTM)	300(ATM)	400(ITM)
Run1	LSM price	830.8	1240.5	1719.8
	Gap	18.5	22.9	24.1
	Gap(t-CI)	(16.2, 20.9)	(19.5, 26.5)	(18.8, 29.3)
Run2	LSM price	830.6	1241.9	1719.8
	Gap	18.1	22.8	24.9
	Gap(t-CI)	(16.2, 20.1)	(19.6, 26.0)	(19.6, 30.2)
Run3	LSM price	830.2	1240.9	1719.8
	Gap	19.3	21.7	25.5
	Gap(t-CI)	(17.9, 20.8)	(19.7, 23.7)	(22.4, 28.6)

Table E.45: Upper bound for a cancellable snowball. Regressors: c, forward swap rate $R_n^N(T_n)$, fixed leg, floating leg, next Libor rate, next cash flow. About 15 hours for 3 simulation runs.

Run Number		Virtual Coupon Rate C_0		
		200(OTM)	300(ATM)	400(ITM)
Run1	LSM price	823.4	1232.6	1711.9
	Gap	30.0	33.3	35.1
	Gap(t-CI)	(25.6, 34.4)	(27.1, 39.5)	(28.1, 42.1)
Run2	LSM price	823.2	1233.0	1712.5
	Gap	31.8	30.2	35.4
	Gap(t-CI)	(25.4, 38.2)	(25.8, 34.6)	(31.0, 39.7)
Run3	LSM price	823.3	1232.9	1711.5
	Gap	29.2	30.9	32.2
	Gap(t-CI)	(24.2, 34.1)	(25.4, 36.3)	(26.9, 37.4)

Table E.46: Upper bound for a cancellable snowball. Regressors: c, forward swap rate $R_n^N(T_n)$, coupon leg approximation 1, floating leg, next Libor rate, next cash flow. About 18 hours for 3 simulation runs.

Run Number		Virtual Coupon Rate C_0		
		200(OTM)	300(ATM)	400(ITM)
Run1	LSM price	828.5	1235.7	1721.5
	Gap	17.6	22.5	16.8
	Gap(t-CI)	(15.0, 20.2)	(19.9, 25.2)	(15.2, 18.5)
Run2	LSM price	827.5	1235.2	1720.3
	Gap	17.1	22.5	15.6
	Gap(t-CI)	(15.7, 18.5)	(20.0, 24.9)	(13.2, 18.1)
Run3	LSM price	828.1	1235.3	1720.2
	Gap	16.7	20.8	17.6
	Gap(t-CI)	(15.4, 17.9)	(18.2, 23.4)	(15.4, 19.9)

Table E.47: Upper bound for a cancellable snowball. Regressors: c, forward swap rate $R_n^N(T_n)$, coupon leg approximation 1, floating leg, next Libor rate, next cash flow, coupon leg approximation 1 squared. About 18 hours for 3 simulation runs.

Run Number		Virtual Coupon Rate C_0		
		200(OTM)	300(ATM)	400(ITM)
Run1	LSM price	830.4	1241.6	1725.6
	Gap	15.4	15.6	12.4
	Gap(t-CI)	(12.7, 18.2)	(14.3, 17.0)	(11.3, 13.5)
Run2	LSM price	830.1	1241.9	1724.6
	Gap	15.5	15.7	11.4
	Gap(t-CI)	(14.0, 16.9)	(13.2, 18.2)	(10.3, 12.5)
Run3	LSM price	830.5	1241.4	1725.1
	Gap	16.2	16.1	12.1
	Gap(t-CI)	(14.9, 17.5)	(14.4, 17.7)	(10.9, 13.2)

Table E.48: Upper bound for a cancellable snowball. Regressors: c, forward swap rate $R_n^N(T_n)$, coupon leg approximation 2, floating leg, next Libor rate, next cash flow. About 34 hours for 3 simulation runs.

Run Number		Virtual Coupon Rate C_0		
		200(OTM)	300(ATM)	400(ITM)
Run1	LSM price	827.0	1230.5	1702.9
	Gap	19.4	26.2	33.9
	Gap(t-CI)	(15.9, 22.9)	(23.8, 28.7)	(32.0, 35.7)
Run2	LSM price	826.2	1231.0	1703.8
	Gap	18.0	26.9	34.1
	Gap(t-CI)	(15.4, 20.5)	(25.0, 28.9)	(33.2, 35.1)
Run3	LSM price	827.2	1230.5	1703.1
	Gap	17.9	25.6	36.9
	Gap(t-CI)	(16.2, 19.7)	(23.9, 27.4)	(34.1, 39.8)

Table E.49: Upper bound for a cancellable snowball. Regressors: c, forward swap rate $R_n^N(T_n)$, coupon leg approximation 2, floating leg, next Libor rate, next cash flow, coupon leg approximation 2 squared. About 34 hours for 3 simulation runs.

Run Number		Virtual Coupon Rate C_0		
		200(OTM)	300(ATM)	400(ITM)
Run1	LSM price	838.7	1250.6	1730.2
	Gap	6.3	6.1	5.8
	Gap(t-CI)	(6.0, 6.6)	(5.4, 6.9)	(4.6, 7.1)
Run2	LSM price	839.6	1251.3	1731.4
	Gap	5.4	6.2	5.2
	Gap(t-CI)	(4.1, 6.7)	(5.5, 6.8)	(4.4, 5.9)
Run3	LSM price	839.1	1250.4	1730.6
	Gap	6.6	5.7	5.8
	Gap(t-CI)	(5.5, 7.8)	(4.8, 6.6)	(4.8, 6.8)

Table E.50: Upper bound for a cancellable snowball. Regressors: c, forward swap rate $R_n^N(T_n)$, coupon leg approximation 2, floating leg, next Libor rate, next cash flow, coupon leg approximation 2 squared and to the third power. About 34 hours for 3 simulation runs.

Run Number		Virtual Coupon Rate C_0		
		200(OTM)	300(ATM)	400(ITM)
Run1	LSM price	838.0	1250.5	1730.1
	Gap	5.5	5.6	5.8
	Gap(t-CI)	(4.4, 6.5)	(5.0, 6.1)	(5.0, 6.5)
Run2	LSM price	838.9	1250.7	1731.1
	Gap	6.0	5.5	5.8
	Gap(t-CI)	(3.8, 8.3)	(5.1, 5.8)	(5.3, 6.2)
Run3	LSM price	0	0	0
	Gap	0	0	0
	Gap(t-CI)	(0, 0)	(0, 0)	(0, 0)

References

[1] *Matlab Documentation Center.* www.mathworks.com/help/.

[2] W. Abramowitz and I. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables.* Dover Publication, Inc., New York., 1972.

[3] Y. Achdou and O. Pironneau. *Computational Methods for Option Pricing (Frontiers in Applied Mathematics).* Society for Industrial and Applied Mathematic, 2005.

[4] H. Albrecher, Ph. Mayer, W. Schoutens, and J. Tistaert. The Little Heston Trap. *Wilmott Magazine*, pages 83 – 92, January 2007.

[5] L. Andersen. Simple and Efficient Simulation for the Heston Stochastic Volatility Model. *Journal of Computational Finance*, 11(3):1–42, 2008.

[6] L. Andersen and M. Broadie. Primal-Dual Simulation Algorithm for Pricing Multidimensional American Option. *Management Science*, 50(9):1222 – 1234, 2004.

[7] L. Andersen and V. Piterbarg. *Encyclopedia of Quantitative Finance*, chapter Bermudan Swaptions and Callable Libor Exotics. Wiley Online Library. 2010.

[8] C. Bender, A. Kolodo, and J. Schoenmakers. Iterating Cancellable Snowballs and Related Exotics. *Risk*, 19(9):126 – 131, 2006.

[9] T. Björk. *Arbitrage Theory in Continuous Time.* Oxford University Press, 3 edition, 2009.

[10] F. Black and M. Scholes. The Pricing of Options and Corporate Liabilities. *Journal of Political Economy*, 81:659–683, 1973.

[11] P. Boyle. A Lattice Framework for Option Pricing with Two State Variables. *Journal of Financial and Quantitative Analysis*, 23:1–12, 1988.

[12] A. Brace, D. Gatarek, and M. Musiela. The Market Model of Interest Rate Dynamics. *Mathematical Finance*, 7:127 – 154, 1997.

[13] D. Brigo and F. Mercurio. *Interest Rate Models-Theory and Practice With Smile, Inflation and Credit.* Springer, 2nd edition, 2006.

[14] M. Broadie and O. Kaya. Exact Simulation of Stochastic Volatility and Other Affine Jump Diffusion Processes. *Journal of Operation Research*, 54(2):217–231, 2006.

[15] J. Buitelaar. Control Variates for Callable Libor Exotics. Master's thesis, Delft University of Technology, 2006.

[16] J.F. Carriere. Valuation of the Early-exercise Price for Options Using Simulations and Nonparametric Regression. *Insurance: Mathematics and Economics*, 19(1):19–30, Dec 1996.

[17] Emmanuelle Clement, Damien Lamberton, and Philip Protter. An Analysis of a Least Squares Regression Method for American Option Pricing. *Finance and Stochastics*, 6:449 – 471, 2002.

[18] J.C. Cox, J.E. Ingersoll, and S.A. Ross. A Theory of the Term Structure of Interest Rates. *Econometrica*, 53:129–151, 1985.

[19] J.C. Cox, S.A. Ross, and M. Rubinstein. Option Pricing: A Simplified Approach. *Journal of Financial Economics*, 7:229–263, 1979.

[20] C. Cryer. The Solution of a Quadratic Programming Problem Using Systematic Overrelaxation. *SIAM Journal on Control*, 9:385–392, 1971.

[21] J.L. Doob. *Stochastic Processes.* Wiley, 1953.

[22] P. Dupuis and H. Wang. On the Convergence from Discrete to Continuous Time in an Optimal Stopping Problem. *The Annals of Applied Probability*, 15(2):1339 – 1366, 2005.

[23] F. Fang. *The COS Method: An Efficient Fourier Method for Pricing Financial Derivatives.* PhD thesis, Delft University of Technology, 12 2010.

[24] B.D. Fekedulegn, J.J. Colbert, R.R. Hicks Jr., and M.E. Schuckers. Coping with Multicollinearity: An Example on Application of Principal Components Regression in Dendroecology. *US Department of Agriculture Research Paper*, 2002.

[25] C.P. Fries. Foresight Bias and Suboptimality Correction in Monte-Carlo Pricing of Options with Early Exercise: Classification, Calculation and Removal. *SSRN working paper series*, 2005.

[26] G. Fusai, S. Sanfelici, and A. Tagliani. Practical Problems in the Numerical Solution of PDE's in Finance. *Rendiconti per gli Studi Economici Quantitativi*, 2001(1):105–132.

[27] H. Geman, N. El Karoui, and J.C. Rochet. Changes of Numéraire, Changes of Probability Measure and Option Pricing. *Journal of Applied Probability*, 32:443 – 458, 1995.

[28] P. Glasserman. *Monte Carlo Methods in Financial Engineering.* Springer - Verlag, 2004.

[29] P. Glasserman and B. Yu. Number of Paths Versus Number of Basis Functions in American Option Pricing. *The Annals of Applied Probability*, 14(4):2090 – 2119, 2004.

[30] T.J. Haahtela. Recombining Trinomial Tree for Real Option Valuation with Changing Volatility. In *Real Options Theory Meets Practice, 14th Annual International Conference*, June 2010.

[31] M.B. Haugh and L. Kogan. Pricing American Options: A Duality Approach. *Operations Research*, 52:258 – 270, 2004.

[32] F. Hayashi. *Econometrics.* Princeton University Press, 2000.

[33] S.L. Heston. A Closed-form Solution for Options with Stochastic Volatility with Applicaiton to Bonds and Currency Options. *The Review of Financial Studies*, 6(2):327–343, 1993.

[34] K.J. In'T. Hout and S. Foulon. ADI Finite Difference Schemes for Option Pricing in the Heston Model with Correlation. *International Journal of Numerical Analysis and Modeling*, 7(2):303 – 320, 2010.

[35] J.C. Hull. *Options, Futures Other Derivatives.* Prentice Hall, 7 edition, 2008.

[36] C.J. Hunter, P. Jäckel, and M.S. Joshi. Getting the Drift. *Risk*, 2001.

[37] P. Jäckel and R. Rebonato. The Link between Caplet and Swaption Volatilities in a bgm/j Framework. *Journal of Computational Finance*, 6(4):41 – 59, 2003.

[38] F. Jamshidian. Libor and Swap Market Models and Measures. *Finance and Stochastics*, 1:293 – 330, 1997.

[39] F. Jamshidian. Numeraire-Invariant Option Pricing American, Bermudan, and Trigger Stream Rollover. 2004. Preprint, Versino 1.6.

[40] M.S. Joshi and D.S. Kainth. Enhancements to the Longstaff-Schwartz Algorithm for Bounding Callable Libor Exotics. In *ICBI Risk Management Conference*, 2005.

[41] M.S. Joshi and A.M. Stacey. New and Robust Drift Approximations for the Libor Market Model. *Quantitative Finance*, 8(4):427 – 434, 2008.

[42] E. Kreyszig. *Advanced Engineering Mathematics.* Wiley, 10 edition, 2011.

[43] F.A. Longstaff and E.S. Schwartz. Valuing American Options by Simulation: a Simple Least-Squares Approach. *Review of Financial Studies*, 14(1):113 – 147, 2001.

[44] P.K. Medina and S. Merino. *Mathematical Finance and Probability.* Birkhuser Basel, 2003.

[45] V.V. Piterbarg. A Practitioner's Guide to Pricing and Hedging Callable Libor Exotics in Forward Libor Models. *SSRN working paper series*, 2003.

[46] R. Rebonato. *Interest Rate Option Models.* John Wiley and Sons, 1998.

[47] R. Rebonato. *Modern Pricing of Interest Rate Derivatives.* Princeton University Press, 2002.

[48] L.C.G. Rogers. Monte Carlo Valuation of American Options. *Mathematical Finance*, 12:271 – 286, 2002.

[49] S.M. Ross. *Stochastic Processes.* Wiley, 2nd edition, 1995.

[50] R.U. Seydel. *Tools for Computational Finance.* Springer, 2009.

[51] S.E. Shreve. *Stochastic Calculus for Finance II: Continuous-Time Models*. Springer, 1 edition, 2004.

[52] M.K. Simon. *Probability Distributions Involving Gaussian Random Variables: A Handbook for Engineers and Scientists.* Springer, Nov 2006.

[53] J.L. Snell. Application of Martingale System Theorems. *Transaction of the American Mathematical Society*, 73(2):293 – 312, 1952.

[54] J. Strikwerda. *Finite Difference Schemes and Partial Differential Equations*. Society for Industrial and Applied Mathematics, 2 edition, 2007.

[55] H. Thom. Longstaff Schwartz Pricing of Bermudan Options and their Greeks. Master's thesis, University of Oxford, 2009.

[56] J.N. Tsitsiklis and B. Van Roy. Regression Methods for Pricing Complex American-style Options. *IEEE Transaction on Neural Networks*, 12(4):694–703, Jul 2001.

[57] J. Versendaal. Upper Bounds for the Value of Bermudan Options. Master's thesis, Delft University of Technology, 2008.