PRICING AMERICAN OPTIONS BY SIMULATION

by

Cem Coşkan

B.S., Management Engineering, Istanbul Technical University, 2006

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Industrial Engineering
Boğaziçi University
2008

ACKNOWLEDGEMENTS

It is difficult to state my gratitude to my supervisor, Assoc. Prof. Wolfgang Hörmann. Without his inspiration and great efforts to explain things clearly and simply, I would probably be lost in thesis study. Besides introducing me to the field of computational finance, I am also grateful for his advices and guidance throughout my thesis period.

I am deeply indebted to Assist. Prof. Mehtap Hisarcıklılar from Istanbul Technical University who helped and encouraged me not only while writing this thesis, but also since I have began the graduate program. In addition, I am thankful to TUBITAK for the financial assistance that is provided until my graduation.

During this work, I have collaborated with many colleagues for whom I have great regard and I want to express that I thank much to those who have helped me with my work besides providing a stimulating and fun environment.

Finally, I wish to extend my warmest thanks to my parents for their suppoert and to all who really care about me.

ABSTRACT

PRICING AMERICAN OPTIONS BY SIMULATION

Pricing American options is a difficult problem in mathematical finance as no closed form solution exists. So, many approximations and numerical techniques have been developed. In option pricing, it is advantageous to use Monte Carlo simulation due to its precision and convergence. In this thesis, we implement and analyze the Least Squares Monte Carlo (LSM) algorithm of Longstaff and Schwartz (2001), which is a regression-based method for American option pricing. For one-dimensional cases, we offer an alternative method that requires less computational effort while the accuracy is increased; Simple Regression Approach (SRA), which is based on the LSM algorithm. Moreover, we analyze the options on multi-assets and use the LSM algorithm to price these options. As well as applying variance reduction techniques and reaching a considerable decrease in the estimated variance in multi-dimensional cases, we also investigate how many early exercise opportunities should be considered in the algorithm for the option to be close to American one. Numerical results indicate that the price estimates in the literature are actually low for American options on the maximum of two and five assets. In addition, we determine a useful set of basis functions for pricing American spread options. For all implementations we use R, which is a programming language and environment for statistical computing and graphics.

ÖZET

AMERİKAN OPSİYONLARININ SİMÜLASYONLA FİYATLANDIRILMASI

Matematiksel finansta, Amerikan opsiyonlarının fiyatlandırılması kapalı formül çözümü bulunmadığı için zor bir problemdir. Bu sebeple, birçok yaklaşım yöntemleri ve sayısal teknikler geliştirilmiştir. Opsiyon fiyatlandırmada Monte Carlo simülasyonu kullanmak, hassasiyetinden ve yakınsamasından ötürü avantajlıdır. Bu tezde, Amerikan opsiyonlarının fiyatlandırılması için Longstaff ve Schwartz'ın (2001), regresyona dayalı bir metot olan, En Küçük Kareler Monte Carlo algoritmasını uyguladık ve analiz ettik. Tek boyutlu örnekler için doğruluk artarken daha az hesaplama gücüne ihtiyaç duyan, En Küçük Kareler Monte Carlo algoritmasına dayanan, Basit Regresyon Yaklaşımı adlı alternatif bir yöntem önerdik. Ayrıca, birden fazla varlığa dayalı opsiyonları analiz ettik ve bu opsiyonları fiyatlandırmak için En Küçük Kareler Monte Carlo algoritmasını kullandık. Çok boyutlu örneklerde varyans azaltma teknikleri uygulamanın ve varyansta kayda değer bir düşüş sağlamanın yanı sıra, opsiyonun Amerikan opsiyonuna yakın olabilmesi için kaç adet erken kullanım fırsatının göz önünde bulundurulması gerektiğini araştırdık. Sayısal sonuçlar iki ve beş varlığın maksimum fiyatına bağlı opsiyonların literatürdeki fiyat tahminlerinin aslında düşük olduğuna işaret etmektedir. Ek olarak, Amerikan yayılma opsiyonları için yararlı bir temel fonksiyon kümesi belirledik. Tüm uygulamalar için programlama dili ve istatiksel hesaplama-çizgeleme ortamı olan R'ı kullandık.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/ABBREVIATIONS

b	Cost of carry
d	Number of time steps
h	Payoff function
k	Number of underlying assets
K	Strike price
n	Number of generated paths
Q_i	Quantity of the i^{th} asset
r	Risk-free interest rate
S_0	Initial asset price
S_0^i	Initial price of the i^{th} asset
S_t	Price of the asset at time t
S_t^i	Price of the i^{th} asset at time t
t	Time
T	Maturity time
V_t	Value of an option at time t
β	Regression coefficients
δ	Continous dividend yield
ψ	Basis functions
ρ	Correlation
σ	Volatility
τ	Stopping time
AV	Antithetic variates
BS	Black-Scholes
CI	Confidence interval
CV	Control variates
exp	Exponential power
FD	Finite difference

GBM	Geometric Brownian motion
i.i.d	independent and identically distributed
LSM	Least squares Monte Carlo
max	Maximum of
NR	Newton-Raphson
OPT	Optimization
PDE	Partial differential equation
SDE	Stochastic differential equation
SRA	Simple regression approach
TRF	Time reduction factor
VRF	Variance reduction factor

1. INTRODUCTION

An option is a financial instrument which gives its holder the right to receive a contingent payoff. There are several types of options. One common type is American options, which allow the holder to receive payoff at any time during the lifetime of the option. Hence, as expected, it is not easy to price this type of option. Although some derivatives, such as European options, can be priced analytically, a general closed-form analytical solution for the evaluation of American options does not exist.

The use of Monte Carlo simulation in option pricing is advantageous regarding its precision and convergence, which is generally independent of the number of underlying variables. However, it has the disadvantage of being quite slow. In the last decade, several articles have been published on how to price American options by Monte Carlo simulation (Andersen and Broadie, 2004; Broadie and Glasserman, 2004; Longstaff and Schwartz, 2001; Tsitsiklis and Van Roy, 2001). Monte Carlo methods usually simulate the development of the underlying asset with random numbers, to determine several possible price movements, subject to the known market data.

The aim of this thesis study is to understand and implement the pricing of American options by simulation. We focus on the Least Squares Monte Carlo (LSM) method of Longstaff and Schwartz (2001) to price American options, analyze and try to improve it in terms of computational cost while increasing its accuracy. Also, we apply the LSM algorithm together with variance reduction techniques in pricing multidimensional American options and investigate how many time points should be considered within the simulation. We make all the implementations in R, which is a free software environment for statistical computing and graphics.

The organisation of the thesis is as follows: First, Chapter 2 gives general information about options. Then, fundamentals in option pricing are introduced to reveal the assumptions and to comprehend the use of simulation in option pricing in Chapter 3. In Chapter 4, the literature on different approaches to price American options is summarized. In addition, it describes the issues about the simulation of American options and the Least Squares Monte Carlo (LSM) method of Longstaff and Schwartz (2001). In Chapter 5, we assess the LSM algorithm with a number of numerical experiments and introduce an alternative method to price American options, named simple regression approach (SRA), in order to increase the efficiency of the LSM algorithm. An optimization approach is implemented and its results are discussed in Chapter 6. Then, in Chapter 7, multi-asset American options are explained. Finally, we price American options on the maximum of two or five assets and American spread options, both of which are classified as multi-asset American options, by applying variance reduction techniques and give numerical results in Chapter 8. Conclusions, Appendix and references are added at the end of the thesis.

2. OPTIONS

In this chapter, options are described in general. After that some different types of options are explained. Besides, important issues about early exercise of American options are stated. Hull (2003) and Lyuu (2002) are references commonly used in this chapter.

2.1. Definition

In the world of finance, a financial security whose value is derived from another underlying financial security is called as a derivative (Hull, 2003). Derivatives can be used to hedge risk as well as for speculative purposes.

An option, which is also a financial derivative, represents a contract between two parties, which is sold by one party to another and gives the right, but not the obligation, to buy or to sell a security or an asset at an agreed-upon exercise (strike) price (Hull, 2003).

According to the Options Industry Council (OIC), which is a Chicago-based non-profit industry group, the options marketplace was slack and unregulated until the mid 20th century (OIC, 2008). The Chicago Board Options Exchange (CBOE) began trading options in 1973 (Lyuu, 2002). As stated by the 2006 Market Statistics report of CBOE, more than two billion option contracts changed hands in 2006. In this volume of the market, CBOE had a share of 33 per cent, which is worth almost 311 billion dollars. Furthermore, Clary (2007) argues that the volume of the options has already exceeded the record occurred in 2006. It is also asserted that Wall Street tries to convince people to use options. On the other hand, the marketplace of options in Turkey, which is named as Turkish Derivative Exchange (TurkDEX), started operations in 2001 and the volume of the operations is increasing rapidly year by year (TurkDEX, 2008).

2.2. Types of Options

The general distinction in options is on the right that is gained. If the option owner gains the right to purchase, this option is called a call option. Oppositely, the type of option that gives the right to sell is named put option. The payoff function h is:

$$h_c(S) = \max(S - K, 0) = (S - K)^+ \text{ for calls}$$
$$h_p(S) = \max(K - S, 0) = (K - S)^+ \text{ for puts}$$

where S is the price of the underlying security and K is the exercise (strike) price. Zero indicates that the option is not exercised.

According to when options can be exercised, they are classified into three groups; European, Bermudan and American Options. It is possible to exercise American options at any time within the time horizon whereas European options only are exercised at the pre-defined expiry date. Therefore, when the price of the underlying is high enough for calls and low enough for puts, which results payoff to be high, it can influence its owner to exercise the American option early (Charnes, 2000). Early exercise refers to the exercise of the option before maturity. Bermudan options are similar in style to American options regarding the possibility of early exercise. However, the difference is the fact that a Bermudan option has predetermined discrete exercise dates. So, Bermudan options can be placed between European and American options (Glasserman, 2003).

Additionally, Exotic options differ from common options in terms of the underlying or the calculation of the payoff. Asian, Barrier, Rainbow, Spread, Basket and maximum options are some commonly traded options that are categorized as exotic ones.

2.3. American Options and Early Exercise

The basic feature of American options is that it enables the owner to exercise earlier than maturity date (Hull, 2003). In this section, the early exercise of American calls and puts is explained.

Lyuu (2002) asserts that an American call option on a non-dividend paying stock should not be exercised early. This can be shown by an example. Let us consider two portfolios. Portfolio A is composed of a bond that has an initial value of Ke^{-rT} and an American call on a non-dividend paying stock whereas Portfolio B is only one share of the underlying stock. T is the expiration date and r is the risk-free interest rate. If early exercise occurs at a time t, such that $0 < t < T$, the value of Portfolio A would be:

$$Ke^{-r(T-t)} + S_t - K = S_t - K\left(1 - e^{-r(T-t)}\right) \leq S_t$$

and of Portfolio B would be S_t which denotes the stock price at time t. It can be noticed that if the call is exercised early, worth of Portfolio A is smaller than or equal to the worth of Portfolio B. However, on the maturity date T, Portfolio B would have a value of S_T while Portfolio A would have:

$$K + \max(S_T - K, 0) = \max(K, S_T)$$

Therefore, if the option were held until expiration, Portfolio A would be worth at least as much as Portfolio B. This shows that when the underlying stock is paying no dividend, an American call should not be exercised early. In addition, it implies that an American call has the same value as a European call option. This does not mean that we must keep an American call until the expiration date. Selling it may be a better alternative.

Exercise of an American call option on a dividend paying stock should take place at maturity date or before the stock goes ex-dividend (Hull, 2003). As the right to collect dividend will be obtained, it may be optimal to exercise it before the ex-dividend date. When we consider the same two portfolios mentioned above, it can happen that by the effect of dividend payment at early exercise time t, portfolio A may have a greater value than portfolio B.

On the other hand, early exercise of an American put might be optimal regardless of whether the underlying stock is paying dividend or not (Lyuu, 2002). Hull (2003) explains this fact by considering two portfolios. Let Portfolio C consist of an American put option plus one share of the underlying stock and Portfolio D consist of a bond that has an initial value of $K\exp(-rT)$. At time t, where $0 < t < T$, the worth of Portfolio D amounts to $K\exp(-r(T-t))$ if the option is exercised, Portfolio C is worth K, which will be the worth of Portfolio D at maturity date. If it is not exercised at time t, the worth of Portfolio C is $\max(K, S_T)$ at maturiy. Lyuu emphasizes that early exercise is preferred due to the time value of money because early exercise results in an instant income of K, which can be reinvested.

In short, an American call on a non-dividend paying stock can be evaluated as European option. Therefore, in the one-dimensional case, we consider and price American put options instead of call options. Furthermore, due to the feature of early exercise, it is clear that an American option must have a price at least as high as its European equivalent.

3. FUNDAMENTALS IN OPTION PRICING

This chapter covers the model of Black and Scholes (1973), risk neutral valuation and the use of simulation which are assumed to be elements of fundamentals in option pricing.

3.1. Black-Scholes Model

Black and Scholes (1973) built the derivative pricing theory based on geometric Brownian motion (GBM). The price of an asset at time t, S_t, is said to follow a GBM if it satisfies the stochastic differential equation (SDE):

$$dS_t = \mu S_t dt + \sigma S_t dW_t \tag{3.1}$$

where W_t is the Wiener process or Brownian motion, μ the percentage drift and σ the percentage volatility, are constants. A Brownian motion W_t is characterized by the following facts:

- $W_0 = 0$
- W_t is almost surely continuous
- for $0 \leq s < t$, the independent increments $W_t - W_s$ are normally distributed with mean zero and variance $(t - s)$.

Notice that Equation 3.1 can be written in the following way (the left hand side is the return of the asset):

$$\frac{dS_t}{S_t} = \mu dt + \sigma dW_t$$

By the help of Ito's lemma, the analytic solution of Equation 3.1 is

$$S_t = S_0 \exp\left((\mu - \sigma^2/2)t + \sigma\sqrt{t}Z\right) \tag{3.2}$$

for an arbitrary initial value S_0, where Z is a standard normal variate (Lyuu, 2002). Hence, the log-returns for S_t are normally distributed with mean $(\mu - \sigma^2/2)t$ and variance, $\sigma^2 t$.

Under the assumptions of Black-Scholes, the Black-Scholes PDE, which holds whenever V, the price of the derivative, is twice differentiable with respect to S, is

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0 \tag{3.3}$$

by arbitrage-free pricing. Further details can be seen in Chapter 15 of Lyuu (2002).

The Black-Scholes formula (Black and Scholes, 1973), which is driven by equation 3.3, gives the exact value of a European call or put option, whereas American options do not have any closed form solution. The price of European call V_c^{EU} and European put V_p^{EU} on a non-dividend paying asset, currently trading at price S can be calculated by the Black-Scholes formula:

$$V_c^{EU} = SN(d_1) - Ke^{-rT}N(d_2) \tag{3.4}$$

$$V_p^{EU} = Ke^{-rT}N(-d_2) - SN(-d_1) \tag{3.5}$$

where $N(.)$ denotes the cumulative distribution function of the standard normal distribution, T is the lifetime of the option and

$$d_1 = \frac{\ln(S/K) + (r + \sigma^2/2)T}{\sigma\sqrt{T}} \tag{3.6}$$

$$d_2 = d_1 - \sigma\sqrt{T} \tag{3.7}$$

Implementation of the Black-Scholes formula in R can be seen in Appendix A.1.

3.2. Risk Neutral Valuation

Cox and Ross (1976) suggest risk neutrality in derivative pricing. According to risk neutrality valuation derivatives can be valued under the assumption that investors are risk neutral (Hull, 2003). This means that the risk preferences of the investors have no effect on the value. In a risk-neutral world, there are two simple results:

i. The expected return from all securities is the risk-free interest rate.
ii. The appropriate discount rate for any expected future cash flow is the risk-free interest rate.

Hence, the price of the underlying follows the risk-neutral process,

$$dS_t = rS_t dt + \sigma S_t dW_t$$

where the drift μ in Equation 3.1 is replaced by the constant risk free interest rate r if the underlying is paying no dividend (Lyuu, 2002). Then, the price of this underlying asset at time t, S_t, can be derived by

$$S_t = S_0 \exp\left((r - \sigma^2/2)t + \sigma\sqrt{t}Z\right) \tag{3.8}$$

Furthermore, V^{EU}, which is the price of a European option that expires at time T is given by the discounted expected payoff of the option with the risk-free interest rate,

$$V^{EU} = \mathrm{E}\left[e^{-rT} h(S_T)\right]$$

where S_T is the value of the underlying asset at the option expiry date. In addition, the value of the American option can be written as:

$$V^{US} = \sup_{\tau} \mathrm{E}[e^{-r\tau} h(S_\tau)]$$

where the maximum is taken over all stopping times $0 \leq \tau \leq T$.

3.3. Option Pricing by Monte Carlo Simulation

Monte Carlo methods are a class of computational algorithms that are based on repeated computation and random sampling (Glasserman, 2003). Monte Carlo simulation is used in finance to value and analyze instruments, portfolios and investments by simulating the sources of uncertainity that affects their value. Then the value is determined according to the simulated underlying inputs.

Options can be priced by Monte Carlo simulation (Glasserman, 2003). First, the price of the underlying asset is simulated by random number generation for a number of paths as in Equation 3.2. Then, the value of the option is found by calculating the average of discounted returns over all paths. Since the option is priced under risk neutral measure, the discount rate is the risk-free interest rate.

In order to get a good estimate from simulation (Glasserman, 2003), the variance of the estimator should go to zero and thus the number of samples goes to infinity, which is impossible. Therefore, variance reduction techniques help us to obtain a better estimate in simulation. In the next section, variance reduction techniques that are used throughout the thesis are summarized.

3.3.1. Variance Reduction Techniques

Simulations driven by random inputs will produce random outputs (Glasserman, 2003). If we can somehow reduce the variance of an output random variable of interest without disturbing its expectation, we should have greater precision that means shorter confidence intervals. In Monte Carlo simulation, it is possible to reduce the variance by using a number of variance reduction techniques. The brief explanation of two variance reduction techniques, namely Antithetic Variates (AV) and Control Variates (CV) can be found in this section.

3.3.1.1. Antithetic Variates (AV). The method of antithetic variates (AV) is a variance reduction technique, which introduces negative dependence between pairs of replications (Glasserman, 2003). For instance, pairing a sequence $Z_1, Z_2, \ldots, Z_n$ of i.i.d. $N(0,1)$ variables with the sequence $-Z_1, -Z_2, \ldots, -Z_n$ of i.i.d $N(0,1)$ variables is the implementation of AV for a simulation driven by independent standard normal random variables.

Let us suppose we would like to estimate $\mu = \mathrm{E}[A]$, where A is the output of a simulation experiment, and we have generated two samples, A_1 and A_2. Then an unbiased estimate of μ is given by:

$$\hat{\mu} = \frac{1}{n}\left(\frac{A_1 + A_2}{2}\right)$$

as n, which is the number of paths that each sample has, goes to infinity. Then the variance of the AV estimator is:

$$Var\left(\hat{\mu}\right) = \frac{Var\left(A_1\right) + Var\left(A_2\right) + 2Cov\left(A_1, A_2\right)}{4}$$

Since the variances of the samples equal to each other, the variance of the estimator becomes:

$$Var\left(\hat{\mu}\right) = \frac{Var\left(A_1\right) + Cov\left(A_1, A_2\right)}{2}$$

The variance will be reduced since we generate pairs such that

$$Cov\left(A_1, A_2\right) < 0$$

Further details can be found in Chapter 4 of Glasserman (2003).

3.3.1.2. Control Variates (CV). Suppose again that we wish to estimate $\mu = \mathrm{E}[A]$ where A is the output of a simulation experiment. Thus, our usual estimator is $\hat{\mu} = A$.

Assume that another output of the simulation is X and we know $\mathrm{E}[X]$. Then for any fixed c,

$$\hat{\mu}_c = A - c\,(X - E\,(X)) \tag{3.9}$$

is a control variate (CV) estimator where the observed error $(X - E\,(X))$ serves as a control on the estimate (Glasserman, 2003). In other words, X is a control variable for A. By Equation 3.9, the variance of the CV estimator becomes:

$$Var\,(\hat{\mu}_c) = Var\,(A) + c^2 Var(X) - 2cCov(X, A) \tag{3.10}$$

Notice that if c equals to zero, then

$$Var\,(\hat{\mu}_c) = Var\,(\hat{\mu})$$

Hence, the CV estimator has smaller variance than the standard estimator if

$$c^2 Var\,(X) < 2cCov(X, A)$$

The optimal coefficient c^*, which minimizes the variance of the CV estimator, can be calculated by

$$c^* = \frac{Cov(X, A)}{Var(X)} \tag{3.11}$$

Substituting this value into Equation 3.10, we find that

$$Var\,(\hat{\mu}_c) = Var\,(\hat{\mu}) - \frac{Cov(X, A)^2}{Var(X)}$$

As long as we find an X such that $Cov(X, A)$ is not equal to zero, a variance reduction will be achieved.

Generally $Cov(X, A)$ and $Var(X)$ are not known. It is possible to estimate c* by using the information within the simulation. However, this results in a biased estimate. A small pilot study is enough to remove that bias.

4. PRICING AMERICAN OPTIONS

This section gives a summary of methods in the literature for pricing American options and provides clarification of several important issues in simulation of American options. Also, the simulation method of Longstaff and Schwartz is explained briefly.

4.1. Methods in the Literature

Basic pricing methods for American options are approximation techniques, finite difference (FD) methods, binomial trees and Monte Carlo simulation.

One of the earliest approximation methods is the work of Geske (1979). He approximated the price of the American option by considering a compound option with two exercise dates. In 1984, besides improving the accuracy by considering more exercisable dates, Geske and Johnson introduced the use of Richard extrapolation in option pricing. The quadratic approximation of Barone-Adesi and Whaley (1987), which gives accurate results only for short periods, is another early approach. Several approximation techniques are compared in Broadie and Detemple (1996) and AitSahlia and Carr (1997).

FD method is a traditional method that directly deals with the pricing equations. When the continuous time and space variables are discretized with constant step lengths, a FD grid is formed. Then, the derivative is priced at each node starting with the terminal nodes and proceeding recursively up until the starting time through the FD scheme, which is obtained by discrete approximations to the PDE 3.3. Tavella (2002) emphasizes that the ability to capture early exercise effectively is an advantage of the FD method. An early example to the application of the FD method on pricing derivatives is the work of Brennan and Schwartz (1977).

Binomial tree, which was first proposed by Cox *et al.* (1979), is another method to value options. American options can be valued after approximating the underlying stochastic process as a lattice by this technique. In 1996, Broadie and Detemple suggested improving the binomial method by using the analytic Black Scholes formula and reached a faster rate of convergence. Broadie and Yamamoto (2005) applied the Fast Gauss Transform to speed up calculations to price American options by the binomial method.

Simulation acts as a promising alternative to the methods mentioned previously. An important simulation method for American options was Tilley (1993). This encouraged people to investigate the possibility of pricing American options using Monte Carlo methods. Boyle *et al.* (1997) reviewed this and other early methods such as Barraquand and Martineau (1995), Bossaerts (1989), Broadie and Glasserman (1997).

To price American options, Longstaff and Schwartz (2001) combined Monte Carlo Simulation with the Least Squares method, a regression-based method, by using backwards induction to approximate conditional continuation values. The methods of Tsitsiklis and Van Roy (2001) and Carriere (1996) are similar to least-squares Monte Carlo (LSM) algorithm of Longstaff and Schwartz (2001). Clement *et al.* (2001) analyzed the convergence of LSM. Many of these methods are explained in Glasserman (2003) and Tavella (2002). Throughout this thesis, LSM refers to the algorithm of Longstaff and Schwartz.

Primal-dual representations of the American option problem have been established by Haugh and Kogan (2004), Rogers (2002) and Andersen and Broadie (2004). Duality approach allows both an upper and lower bound to be calculated through a minimization problem over a class of martingales or super martingales. These methods are also described in Glasserman (2003).

FD and tree techniques are traditional methods for pricing (Longstaff and Schwartz, 2001). Most of these methods try to approximate the continuation values by considering the problem in a stochastic dynamic programming framework. For pricing American-style securities, they are efficient in the one-dimensional case whereas simulation methods work well for multi-asset derivatives (Broadie *et al.*, 2000).

4.2. Simulation of American Options

For the simulation of American options, there are a number of basic issues to be explained before we can discuss implementation.

It is emphasized that Bermudan options can be positioned between American and European options. In simulation, due to the inability to simulate in continuous time, an American option is priced under the assumption that it has Bermudan style, and thus only discrete exercise opportunities exist. So, if an American option is exercisable at any time t where $0 \leq t \leq T$, we restrict the option such that it can be exercised only at a fixed set of exercise opportunities $0 < t_1 < t_2 < \ldots < t_d$, where d is the number of exercisable time steps.

In addition, the American option holder compares the payoff from early exercise and expected continuation value to make decision to keep the option alive or not, because the rational behavior for the holder is to maximize the income by this comparison. The payoff function is determined according to the option type. Hence, the key insight in simulation for pricing American options is to estimate the continuation value.

Moreover, the Exercise Boundary is an essential term in pricing American options. It refers to a threshold value for each time step which shows if we should exercise or not. For a put, we should exercise when the price of the underlying is below the exercise boundary while it is opposite if the option is a call. The boundary for expiration date is exactly the strike price. Approximation methods deal with the approximation of the exercise boundary. Simulation techniques try to price the option by determining the stopping time for each path after generating paths and estimating the continuation value. However, in a sense, they look for the exercise boundary in the lifetime of the option. Thus, since we have an idea about threshold values for early exercise, which refer to exercise boundary at any exercisable time step, the simulation of the price will be simple and fast.

Furthermore, it is important to understand the biases in simulation for pricing American options. Most of the methods terminate simulated paths according to their approximation of the optimal exercise boundary (Glasserman, 2003). This can lead to both low and high biases. Using a sub-optimal exercise policy introduces low bias. However, the reason behind high bias is the usage of future information in paths while making the exercise decision.

4.3. LSM algorithm of Longstaff and Schwartz

At any exercise time, the immediate cash flow generated by early exercise and expected payoff from continuation is compared by the American option holder. According to this comparison, the decision to exercise the option or to keep it alive is made. Longstaff and Schwartz use regression to estimate the continuation value.

Longstaff and Schwartz (2001) combine Monte Carlo Simulation with the Least Squares method by using backwards induction to approximate conditional continuation values to price American options. The main idea in their least-squares Monte Carlo (LSM) approach is the usage of least squares to estimate conditional continuation values to decide to exercise the option or not. In the algorithm, the American option is assumed to be of Bermudan style, which has discrete time steps allowing early exercise. After generating paths of the underlying for discrete time steps, the cash flows from early exercise and continuation values are compared to decide to exercise or to continue holding the option at each time step by backward induction. The continuation values at an exercise date are estimated by regressing the discounted subsequent realized cash flows from continuation on a set of basis functions. The regression fitted at each exercisable time step includes only the paths where the option is in the money and results are used only for the comparison. The price of the option is computed by the means of discounted cash flows according to the stopping rule that the algorithm found.[1]

Let a Markov process $\{X(t), 0 \leq t \leq T\}$ denote all necessary information about

[1] See the numerical example in Longstaff and Schwartz (2001) for a better understanding

the current price of the underlying asset. To reduce notation, $X(t_i)$ will be written as X_i, which is the state of the underlying Markov process at the i^{th} time step, where $i = 1, \ldots, d$. The discrete-time process $X_0 = X(0), X_1, \ldots, X_d$ is a Markov chain. $V_i(x)$ and $h_i(x)$ denote the value of the option and the payoff function at time t_i given $X_i = x$. The continuation value in state x at date t_i is

$$C_i(x) = \mathrm{E}\left[V_{i+1}(X_{i+1})\,|X_i = x\right] = \sum_{r=1}^{M} \beta_{ir}\psi_r(x)$$

for some basis functions ψ_r and coefficients $\beta_{ir}, r = 1, \ldots, M$. The coefficients could be estimated by pairs $(X_i, V_{i+1}(X_{i+1}))$. V_{i+1} is unknown in practice. Hence, it must be replaced by the discounted estimated values $\hat{V}_{i+1}$ at downstream nodes. So, regression coefficients define the estimate of the continuation value at an arbitrary point x:

$$\hat{C}_i(x) = \beta_i^T \psi(x) \tag{4.1}$$

with

$$\hat{\beta}_i^T = \left(\hat{\beta}_{i1}, ...\hat{\beta}_{iM}\right), \psi(x) = (\psi_1(x), ..., \psi_M(x))^T$$

Then, the algorithm can be summarized as follows:

i. Simulate n independent paths of the Markov chain $\{X_0, X_1, \ldots, X_d\}$ for d time steps,

ii. At terminal nodes, set $\hat{V}_{dj} = h_d(X_{dj}), j = 1, \ldots, n$,

iii. Apply backward induction: for $i = d-1, \ldots, 1$,
   - Specify the set I, which is composed of paths that are in the money
   - Discount the values $\hat{V}_{i+1,j}, j \in I$ to be input for regression
   - Given estimated values $\hat{V}_{i+1,j}, j \in I$, use regression to calculate $\hat{\beta}_i$,
   - Estimate continuation values by using Equation 4.1,
   - If $h_i(X_{ij}) \geq \hat{C}_i(X_{ij})$, set $\hat{V}_{ij} = h_i(X_{ij})$, else set $\hat{V}_{ij} = \hat{V}_{i+1,j}, j \in I$,

iv. Calculate $\left(\hat{V}_{11} + ... + \hat{V}_{1n}\right)\Big/n$ and discount it to find $\hat{V}_0$

From the perspective of exercise boundary, we know that at the expiration date, the exact boundary is the strike price K. If the option is put, then optimal stopping time τ^* has the form

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

for some optimal exercise boundary $b^*(t)$, where $S(t)$ is the price of the underlying at time t.

Diagnostic of the LSM algorithm has been tested by applying the regression parameters obtained by one set of paths to another set of paths. So, price estimates calculated by using different two sets of paths and same regression functions are approximately similar and accuracy of the approach is supported.

There are several essential issues considered in LSM algorithm of Longstaff and Schwartz. First, only in-the-money paths, where the payoff function is positive, are selected as input to the regression. In other words, paths that do not create any cash flow when exercised at that time step are neglected in regression. In addition, all values are normalized by dividing with the strike price and the conditional expectation function is estimated in the normalized space for American put illustration to avoid computational underflows while using a set of weighted Laguerre polynomials:

$$L_0 = \exp(-X/2) \tag{4.2}$$

$$L_1 = \exp(-X/2)(1 - X) \tag{4.3}$$

$$L_2 = \exp(-X/2)(1 - 2X + X^2/2) \tag{4.4}$$

$$L_n = \exp(-X/2)\frac{e^X}{n!}\frac{d^n}{dX^n}(X^n e^{-X}) \tag{4.5}$$

This process does not affect the option value since the unnormalized values are later transformed back.

LSM method produces a low-biased estimator because for each path, an estimate of optimal exercise time is used (Lemieux and La, 2005). Besides being versatile and easy to implement, the approach is easily adjustable to different cases such as exotic American options and American options with jump diffusions as well as multi-dimensional American options (Laprise *et al.*, 2006). However, Longstaff and Schwartz (2001) claim that the results of LSM are robust with respect to the choice of basis functions. For example, they assert that to price a one-dimensional Amrican put option, it is sufficient to use a constant and three basis functions, which are the first three weighted Laguerre polynomials, as in equations 4.2-4.4. Moreover, as a simulation technique, it is possible to use LSM for parallel computing. In other words, 50000 paths might be generated in one CPU or 5000 paths each on 10 CPUs.

Meanwhile, several variance reduction techniques were suggested for pricing American options via the LSM Algorithm (Bolia *et al.*, 2004; Ehrlichman and Henderson, 2006; Lemieux and La, 2005)

5. ASSESSMENT OF LSM ALGORITHM

The LSM algorithm of Longstaff and Schwartz was explained in the previous chapter. In this chapter, information about how to implement the algorithm in R and results of experiments made by using different basis functions are provided. Then, some problems in the algorithm and ways to improve them are discussed.

5.1. Implementation of LSM in R

R is a free software environment for statistical computing and graphics. Throughout this thesis, R version 2.7.0 has been used. No package in R exists for the LSM algorithm. You can find R codes for the implementation of the LSM algorithm in Appendix A.2. In this section, we simply explain our codes.

Before explaining our implementation, an issue should be cleared. The stock prices for all paths for each time step can be generated by geometric Brownian motion, which is decribed in Section 3.1, through Equation 3.8. However, in this case, a matrix is needed to store the stock prices at all time steps because of the working backwards principle in LSM. The Brownian-Bridge offers a solution to decrease memory requirements (Glasserman, 2003). It allows us to interpolate a value of a Brownian motion given two endpoints. Thus, for any time t, such that $0 < t < T$, where T is the expiration date, S_t can be calculated by a Brownian Bridge construction if S_0 and S_T are known. Hence, it is possible to start with simulating the prices S_T at maturity and continue by simulating S_t. The expectation and the variance can be calculated by the simple formulae:

$$E(\log S_t) = \log S_a + \frac{t-a}{b-a}\left(\log S_b - \log S_a\right) \tag{5.1}$$

$$V(\log S_t) = \sigma^2 \frac{(b-t)\,(t-a)}{b-a} \tag{5.2}$$

where $a < t < b$. So, at each time step, storing two arrays with length of n, the number of paths generated, is enough instead of storing a n x d matrix. Hence, to reduce memory requirements and to be able to simulate a large number of paths, the Brownian Bridge idea is used in our implementation.

The function for pricing is named *simul_US_LSM*. The initialization stage is composed of the normalization of all prices and cash flows by dividing the starting stock price over the exercise price K and finding the time interval throughout simulation.

The first stage, which takes place at the expiration date, includes simulating the stock prices and calculating the cash flows. For this purpose, the stock prices at T are simulated using the Equation 3.8. For the version, where half of the stock prices are simulated by using antithetic variates, generated normal variates are needed to be stored. In this case, all stock prices to apply regression are combined in one array such that the first half is generated from normal variates and the second uses antithetic variates.

In the next stage, a for loop from $d - 1$ to 1 is executed to deal with the calculations in each time step. In this for loop first the stock prices at that time step for all paths are generated by using Brownian Bridge through equations 5.1 and 5.2. By the payoff function, in-the-money paths are selected to be considered in the regression. For this purpose, an array named *idx* that stores the indices of paths that are in the money is assigned. This facilitates carrying out the operations. Before fitting regression, future cash flows are discounted with the risk-free interest rate to be input to the regression. Basis functions for the regression are defined as constant and the first three Laguerre polynomials, which can be seen in equations 4.2-4.4, as indicated in the paper of Longstaff and Schwartz (2001). The coefficients that are calculated by regression enable us to estimate the continuation values for selected paths. The estimated continuation values are saved in a different array. According to the comparison between estimated continuation values and the payoff function, ones and zeros are assigned to exercise flag *EF*.

At this step, the continuation value for the current time step is composed of two parts; one coming from early exercise for the paths where EF contains 1s and continuation values for future time step for the paths where EF contains 0s. The first part might be calculated in two ways; by exercising the paths when EF is equal to 1 or by selecting the maximum stock price where EF is 1 so that below that price there will be early exercise. After computing the value at the current time step, the stock prices at t_i are assigned to stock prices at time t_{i+1} to continue with the previous time step.

The for loop ends when it completes the calculations for t_1. The remaining part is the discount process of the last continuation value to t_0, which represents the initial date. The mean of all paths after discounting gives us the American option price. It is essential to claim that in the case that antithetic variates are used, the average of first half and second half should be taken into consideration before computing the mean. The function prints out the American option price, the European option price (if the option would be European style), the early exercise premium (the value difference between American and European style) and the standard error of estimation.

5.2. How many basis functions are necessary?

The algorithm is run by varying the initial stock price, the volatility and the lifetime. In Section 4.3, it was stated that Longstaff and Schwartz (2001) assert considering three Laguerre polynomials is sufficient. Therefore, experiments were carried out by changing the number of Laguerre polynomials used as basis functions in algorithm, according to the general equation of Laguerre polynomials 4.5. Each case was replicated for 100 times.

In Table 5.1, besides the results of these experiments, the prices from the article of Longstaff and Schwartz (2001), calculated by Finite Difference (FD) method and LSM can be seen. L_k denotes first k number of Laguerre polynomials selected as basis functions. There is not too much difference between the results of FD, LSM and the cases where four and five Laguerre polynomials are used. This is a fact which suits to the statement of Longstaff and Schwartz (2001); only few basis functions are needed

Table 5.1. Price estimates for different number of basis functions and varied initial stock price, volatility and lifetime

			L_1	L_2	L_3	L_4	L_5	LSM	FD
$S_0 = 38$	sigma=0.2	T=1	3.213	3.236	3.246	3.247	3.247	3.244	3.25
		T=2	3.691	3.727	3.733	3.738	3.738	3.735	3.745
	sigma=0.4	T=1	6.116	6.136	6.141	6.141	6.142	6.139	6.148
		T=2	7.613	7.649	7.668	7.667	7.671	7.669	7.67
$S_0 = 40$	sigma=0.2	T=1	2.289	2.307	2.312	2.315	2.315	2.313	2.314
		T=2	2.844	2.872	2.881	2.886	2.884	2.879	2.885
	sigma=0.4	T=1	5.283	5.299	5.312	5.321	5.315	5.308	5.312
		T=2	6.869	6.908	6.918	6.920	6.919	6.921	6.92
$S_0 = 42$	sigma=0.2	T=1	1.601	1.613	1.616	1.616	1.617	1.617	1.617
		T=2	2.18	2.2	2.208	2.208	2.210	2.206	2.212
	sigma=0.4	T=1	4.563	4.574	4.584	4.588	4.587	4.588	4.582
		T=2	6.204	6.232	6.240	6.243	6.245	6.243	6.248

to approximate conditional expectation closely and adding more basis functions has little effect on the value of the option. However, it is obvious that when less than three polynomials are used; price estimates are not very close to the ones of FD.

5.3. Problems in LSM algorithm

In this section, problems in the LSM are discussed. After explaining the influence of the regression on the speed, its accuracy in estimating the early exercise boundary will be checked.

5.3.1. Inefficiency of Regression

It is asserted that continuation values at a time step are estimated through a regression to decide whether to exercise or not. The comparison between these estimated values and immediate cash flows will give us a threshold value, which is the maximum stock price for which early exercise takes place. Thus, it is the value where estimated continuation values and immediate cash flows from early exercise intersect each other. In Figure 5.1, an example of this case taken from a simulation at time t_{d-1}, where d is the number of exercise opportunities, is shown. Notice that the figure is just the part of the whole plot that is close to the intersection point. By fitting the regression to many stock prices that are far from threshold value, LSM algorithm estimates continuation values for stock prices that we are not interested in because it takes all in-the money points as input to regression to find the threshold value at a time step.

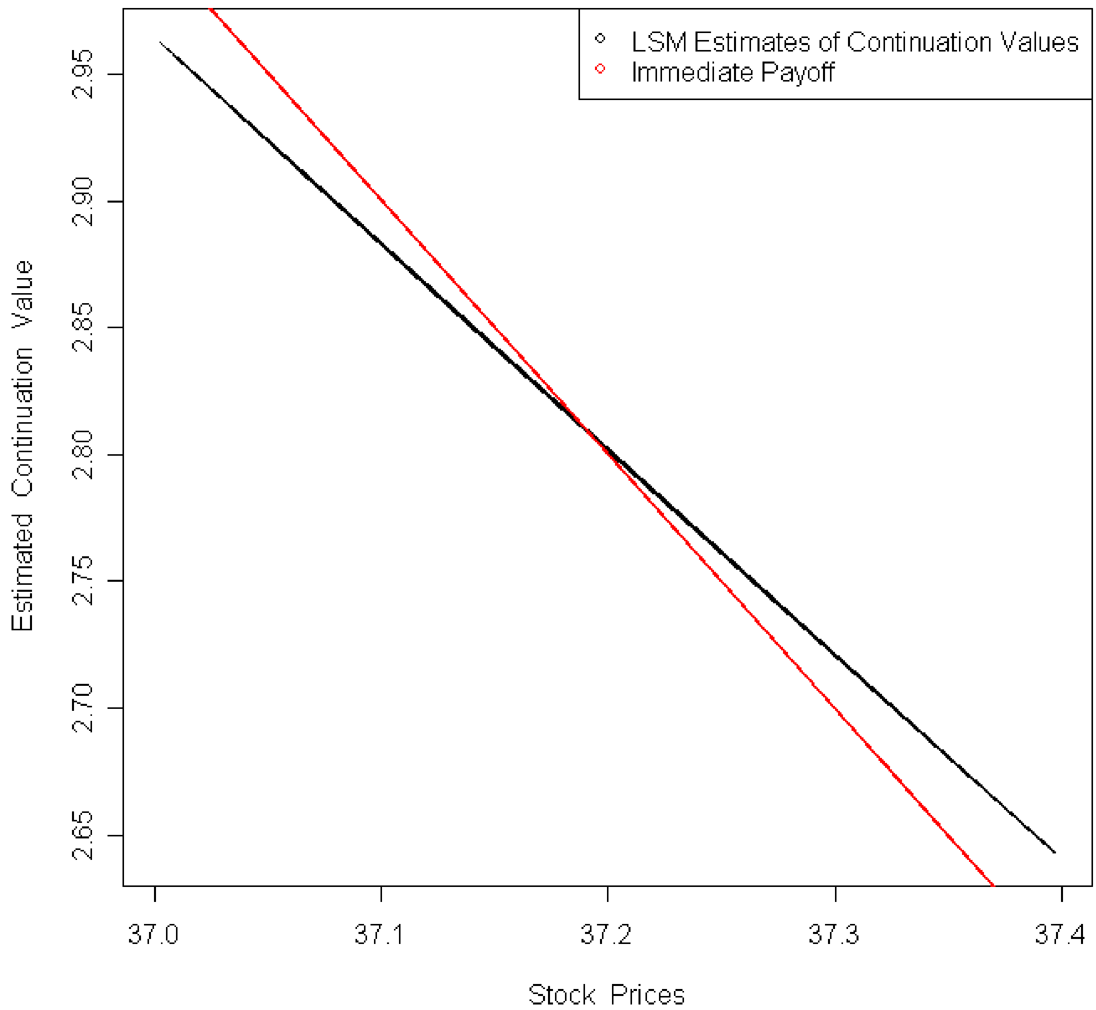

Figure 5.1. Plot of LSM estimates of continuation values versus immediate cash flows from early exercise at time t_{d-1}

Fitting a regression takes significant time. When basis functions are Laguerre polynomials, it is obviously longer than usual. In other words, while there is the interesting region where the two functions intersect each other as in Figure 5.1, the algorithm includes all stock prices that are smaller than the strike price K, instead of focusing on the important region. This leads to inefficiency in computational time and relatively high error in estimation.

It is obvious that the expected continuation values are only estimated to be able to find an exercise boundary, which has influence on the price. However, all generated paths are not considered. Therefore, if a better way to specify the exercise threshold could be found, we would not need to estimate continuation values and would price the option according to the boundary by directly using the stock prices.

5.3.2. Boundary Estimation

Let us suppose there are d time steps that option can be exercised in $(0, T]$, where $t_d = T$. At time t_{d-1}, it is possible to find the exact early exercise boundary by using the Black-Scholes (BS) formula and the Newton-Raphson (NR) method. In Section 3.1, it is indicated that the BS formula gives the exact value of the European call or put option. Also, it is mentioned that in the LSM algorithm, an American option is priced under the assumption that it is Bermudan. Hence, since time t_{d-1}, the option is a European option that has life time in $(t_{d-1}, t_d]$. It is possible to specify an exact boundary at time t_{d-1}, such that there must be a stock price that makes the value of the European option computed from BS formula, which is also the continuation value of the American option, equal to the immediate income from early exercise of American option at t_{d-1}. In other words, at time t_{d-1}, the option does not differ from a European style. So, we can use the price of the European option starting at t_{d-1} and with expiry $t_d = T$ as the exact continuation value for the American option at t_{d-1}. Since we know the threshold is the stock price which equalizes the continuation value to the early exercise value, it can be obtained by equating the price of the European option to the payoff at t_{d-1}. Hence, by the help of BS formula, the threshold for a put is the stock price that satisfies the equation:

$$V_p^{EU} - (K - S)^+ = 0$$

Table 5.2. Early exercise boundary calculated by LSM algorithm for a Bermudan option with two exercise opportunities and comparison with exact values

	Time of 1^{st} Exercise					
	11/12	10/12	9/12	8/12	7/12	6/12
L_1	36.3309	36.06	35.9347	35.8921	35.8621	35.8946
L_2	36.536	36.4974	36.4543	36.4342	36.4301	36.4303
L_3	37.1316	36.9665	36.8417	36.7319	36.6238	36.5559
L_4	37.4576	37.145	36.9341	36.7636	36.6455	36.5483
L_5	37.6021	37.1821	36.9311	36.767	36.642	36.5545
BS+NR	**37.6472**	**37.1941**	**36.9366**	**36.7663**	**36.6457**	**36.5571**

This means that for a specific value of S, which is called early exercise boundary, the continuation value equals the immediate income gained by early exercise. If V_p^{EU} is replaced by its value of the BS formula, equation becomes:

$$Ke^{-rT}N(-d_2) - SN(-d_1) - (K-S)^+ = 0 \tag{5.3}$$

This equation can be solved by the NR method. A root of $f(x) = 0$ can be found with an initial guess of x_0

$$x_{k+1} = x_k - \frac{f\left(x_k\right)}{f'(x_k)}$$

(Acton, 1990). The algorithm might be stopped when

$$\frac{|x_{k+1} - x_k|}{|x_k|} \leq \varepsilon$$

where ε is the acceptable error and can be determined by the user. Since $f(x)$ is the left hand side of Equation 5.3 and x is the asset price S, the root found by the NR method is the element of exercise boundary for t_{d-1}. The R implementation of the algorithm can be seen in Appendix A.3.

To illustrate the idea, let us consider a Bermudan put option on a share of a non-dividend paying stock that has initial value 40 and volatility of 20 percent. The option is exercisable at an exercise price of 40 and has the maturity of one year. The risk-free interest rate is six per cent. For the first example, let us assume that the number of exercisable time steps d is two; one is at maturity and the other one varies. In other words, there is only a single early exercise opportunity. At time $t_2 = T$, the exact boundary is the exercise price, so, the option will be exercised if it is in the money. Thus, the exact boundary at maturity is the strike price. We experimented to find the early exercise threshold value by comparing LSM algorithm and a combination of BS formula and NR method. In the experiments where the early exercise boundary found by BS formula and NR method, ε is taken as 10^{-8}. Results are summarized in Table 5.2 by varying the first exercisable time and the number of Laguerre polynomials, where L_k denotes first k weighted Laguerre polynomials selected as basis functions. Since the threshold values found by BS formula and NR method are the exact boundaries, it can be seen that the performance of LSM decreases when the exercisable time step is close to maturity. In addition, the boundaries computed by LSM are always smaller than the exact boundaries and get closer to the exact ones as we increase the number of basis functions used in the algorithm, however, they are still not identical for the situations in which the early exercise is allowed close to the maturity. This property is a problem for the algorithm because for pricing American options we use Bermudan options with many exercise opportunities.

Altogether, the LSM algorithm has two main problems; the computational load of fitting the regression and the inability to estimate the exercise boundary.

5.4. Improvement of Region of Regression

In Section 5.3.1, the problem of the LSM algorithm with regression, which is the fact including all in-the-money points instead of focusing on the important region is described. In this section, we propose a way to improve this situation, its implementation in R and numerical results.

5.4.1. The Idea

Longstaff and Schwartz use Laguerre polynomials as basis functions in regression. It takes considerable time to fit such a regression to all in-the-money paths. However, if we had an idea about the early exercise threshold, it would be possible to fit a simpler linear regression around that value and modify that pre-estimated threshold value. One possibility to limit the region where regression is applied at a time step is the usage of the exercise boundary at the subsequent time step. The other, which we will focus on, is the usage of the LSM algorithm. Using a relatively small number of paths to estimate the threshold values in LSM algorithm might give an idea about where the thresholds are. After that, those values might be improved by applying simple regressions around that region. Thus, we propose that the LSM algorithm might be enhanced by following these steps;

- First run the LSM algorithm with a specific small number of paths selected from generated ones, to have an idea where the early exercise threshold is.
- At each time step specify a distance, such that, a specific number of paths exists in that region close to the pre-estimated threshold.
- Instead of Laguerre polynomials, consider a simpler basis function, such as stock price at that time step, fit regression and find the modified threshold value.
- Value the option according to the modified threshold values.

By this algorithm, it is aimed to reach a speed-up while getting identical results with the LSM algorithm and the FD method. Throughout this thesis, this method will be called Simple Regression Approach (SRA).

In the pre-estimating stage, we have two choices in selecting paths to be considered; random selection from generated paths or selection of paths by beginning from the first generated one. Since, all paths are randomly generated, we do not think that it will cause a problem in bias to select the first samples. In experiments, the second approach is used. For some time steps the LSM algorithm might have problems to find a threshold value. In these cases, it is assumed the threshold value at time t_i is equal to the one at time t_{i+1} if it cannot be computed by the algorithm.

After commenting how to implement SRA in R, the algorithm will be tested by a number of experiments.

5.4.2. R Implementation of SRA

The implementation of SRA in R can be seen in Appendix A.4. The implementation is not much different from the one of LSM. The difference in the function *simul_US_SRA* is the stage that estimates the threshold value with small number of paths and fits a simple regression to the values closer to the threshold value than a specified distance.

Two new arguments were added to the definition of the previous function; *npilot*, as the percentage of generated paths for the pilot study with the LSM algorithm and *nreg*, as the percentage of generated paths to be considered in the simple regression. There is no difference in commands before the *for* loop except identifying an array for thresholds to save. Since Brownian Bridge is used, it is impossible to have an idea about all the threshold values at each time step at once. Hence, we add this fitting regression with a small number of paths at the beginning of the *for* loop, after generating the stock prices at that time step. We store that threshold value and specify a distance according to the input value of *nreg*. For this purpose, we subtract the threshold value from all asset prices at that time step, take their absolute value and store them in an array. The *nreg*th element of the sorted array is our distance. After that, regression is fitted by using simpler basis functions, a constant, the asset price, and a new threshold value are obtained. Before going to the previous time step, the exercise flag *EF* is assigned by the threshold value and the value of the option at that time step is calculated. In the end of the *for* loop, again we change the current asset price as the subsequent asset price as we use the Brownian Bridge. The function prints American option price, standard error, European option price and early exercise premium.

In addition, the computational time can be taken by using the function *sys-*

tem.time.

5.4.3. Numerical Results

Let us consider the example that we used to show the problem of LSM in estimating the exercise boundary. The exercise boundary calculated by SRA for that example is compared with the exact values obtained by the BS formula and NR method in Table 5.3. *npilot* refers to the percent of all paths to be considered for the pilot study with the LSM algorithm whereas *nreg* refers to the percentage of all paths used for simple regression. In contrast to the LSM algorithm, SRA estimates the boundaries quite well, even when the time interval gets smaller.

Now, the algorithm is run to price the American option and the results are compared with FD and LSM results given in the paper of Longstaff and Schwartz (2001). Since the boundaries are estimated well when *npilot* is 1 percent and *nreg* is 10 percent, those values were selected for further experiments, to reduce the execution time. However, in the cases where the percentage of in-the-money paths is less due to the initial asset price, a higher percentage value would be better to select for *npilot*.

Table 5.4 summarizes the results for the American put example considered previously. According to the table, SRA gives sensible results closer to FD results than the LSM algorithm. Also, time reduction factors (TRFs), which means the fraction of time used by LSM algorithm over time used by simple regression approach, are reported and show that the computational effort is decreased. TRFs are higher when the option begins in-the-money, in other words, when the initial asset price is lower than the strike price. As the life of the option gets longer, TRFs tends to decrease, however, it is not valid for the cases in which the initial asset price is higher than the strike price. Furthermore, it can be said that increased volatility of the asset leads to higher TRFs.

Altogether, SRA is an alternative simulation method that can be used for pricing one-dimensional American options with price estimates that are similar to those

Table 5.3. Early exercise boundary calculated by SRA for a Bermudan option with two exercise opportunities and comparison with exact values

		Time of 1^{st} Exercise					
		11/12	10/12	9/12	8/12	7/12	6/12
npilot=10%	nreg=10%	37.61998	37.16731	36.88797	36.74795	36.63377	36.52508
	nreg=5%	37.54787	37.18704	36.91227	36.73801	36.62277	36.55265
npilot=5%	nreg=10%	37.60546	37.15817	36.87599	36.74097	36.62698	36.53988
	nreg=5%	37.54228	37.17075	36.91736	36.73636	36.61452	36.55383
npilot=1%	nreg=10%	37.60229	37.17075	36.89427	36.75314	36.63698	36.57485
	nreg=5%	37.51084	37.14335	36.87268	36.72641	36.6116	36.51356
BS+NR		**37.6472**	**37.1941**	**36.9366**	**36.7663**	**36.6457**	**36.5571**

calculated by FD and with smaller computational costs than LSM.

Table 5.4. Comparison of prices calculated by SRA, LSM and FD, and TRFs varying initial stock price, volatility and lifetime

			LSM	FD	SRA	TRF
S_0=38	sigma=0.2	T=1	3.244	3.25	3.249	4.61
		T=2	3.735	3.745	3.744	4.03
	sigma=0.4	T=1	6.139	6.148	6.147	4.19
		T=2	7.669	7.67	7.672	3.93
S_0=40	sigma=0.2	T=1	2.313	2.314	2.314	3.06
		T=2	2.879	2.885	2.886	2.66
	sigma=0.4	T=1	5.308	5.312	5.313	3.45
		T=2	6.921	6.92	6.916	3.32
S_0=42	sigma=0.2	T=1	1.617	1.617	1.616	1.77
		T=2	2.206	2.212	2.21	1.85
	sigma=0.4	T=1	4.588	4.582	4.584	2.77
		T=2	6.243	6.248	6.241	2.87

6. OPTIMIZATION APPROACH

An American option holder tries to maximize his income by making a comparison between two choices. From this point of view, optimization might be a useful idea.

The approach used in this chapter is the optimization of the exercise boundary by backwards induction. In other words, it is similar to the LSM algorithm with respect to the working principle but searches the optimum threshold value for the maximization of the total value instead of fitting regression. In addition, the optimization approach directly uses future information in pricing whereas in regression based methods, future information is only used for the regression to estimate future cash flows. As stated in Section 4.2, this fact causes the optimization approach to have high bias, in contrast to the low bias of regression based methods.

6.1. R Implementation of Optimization Approach

R codes for the implementation of the optimization approach can be seen in Appendix A.5. Since we do not use LSM algorithm and Laguerre polynomials, there is no need to normalize the prices and cash flows with the strike price. Then, the initialization stage of the function *simul_US_OPT* is only composed of calculating the length of the time interval, generating stock prices at maturity and defining an array to save threshold values. There are also a few changes inside the codes, especially in the *for* loop.

The content of the *for* loop is different from the one used in LSM or SRA. At each time step before the maturity date, the order of the array which consists of stock prices is stored. By using the same indices for this order, continuation values and early exercise incomes are written into two arrays with length of n, equal to the number of paths generated. Notice that if antithetic variates are used, each will have a length of $2n$. Next, the total value array is formed for each stock price, which is a candidate for the boundary value, according to the fact that total value of a put option is the sum of cumulative sum of early exercise incomes equal to or below the threshold candidate and cumulative sum of continuation values above the threshold candidate. The threshold value is selected as the stock price that maximizes the total value by using the function *which.max*. Therefore, early exercise for put options takes place for the stock prices under the threshold values and estimated continuation values are calculated according to this exercise policy before performing the operations for the next time step in the loop. Stock prices at time t_i are assigned as future stock prices for time t_{i-1}.

After the calculations within the *for* loop, the function calculates the average of discounted expected cash flows of all paths. If antithetic variates are used, before taking the mean, the cash flows of the paths generated by normal variates and the cash flows of paths generated by antithetic variates must be averaged. As an output, the price of the American option and the standard error of the simulation are printed.

6.2. Numerical Results

Similar to the experiments done previously, the accuracy of the optimization approach is tested by pricing the American put option of the example used in the paper of Longstaff and Schwartz (2001). It is possible to compare the results of different methods for a variety of cases with the help of Table 6.1. In the optimization approach, for all cases, price estimates achieved by applying antithetic variates are significantly higher than the estimates of LSM, SRA and FD. The main reason of this difference seems to be the usage of future information.

Besides being high biased, there is another issue about this approach, which might also lead to inefficient results. Figure 6.1 shows a plot of total values versus stock prices at time t_{d-1}, where d is the number of time steps and $t_d = T$ is the expiration date. As seen in Figure 6.1, several other stock prices, when considered as a threshold, conclude in total values that are close to the maximum. However, in this approach, the stock price that leads to maximum total value is taken as the threshold value at that time step. This might affect the choice of the threshold value in the previous exercise opportunity date. In other words, maximizing the total value for all paths at

Table 6.1. Prices calculated by optimization approach, LSM and FD, varying initial stock price, volatility and lifetime

			LSM	FD	OPT	s.e
S_0=38	sigma=0.2	T=1	3.244	3.25	3.263	0.005
		T=2	3.735	3.745	3.76	0.006
	sigma=0.4	T=1	6.139	6.148	6.171	0.008
		T=2	7.669	7.67	7.701	0.01
S_0=40	sigma=0.2	T=1	2.313	2.314	2.326	0.005
		T=2	2.879	2.885	2.9	0.006
	sigma=0.4	T=1	5.308	5.312	5.334	0.009
		T=2	6.921	6.92	6.948	0.01
S_0=42	sigma=0.2	T=1	1.617	1.617	1.626	0.006
		T=2	2.206	2.212	2.228	0.007
	sigma=0.4	T=1	4.588	4.582	4.603	0.01
		T=2	6.243	6.248	6.276	0.011

each time step does not necessarily lead to the maximization of the total value for the entire simulation in the whole lifetime of the option. In spite of this, the estimates for the option price are biased significantly high. Nevertheless, this approach might be used to obtain an upper bound for the option price. Additionally, the threshold value estimates of this approach might be examined by using the two-step Bermudan case.

Altogether, the optimization approach is run to price American put option. However, the results are not identical to the ones calculated by using LSM and FD but too high. So, an upper bound for the option price might be obtained by using the optimization approach.

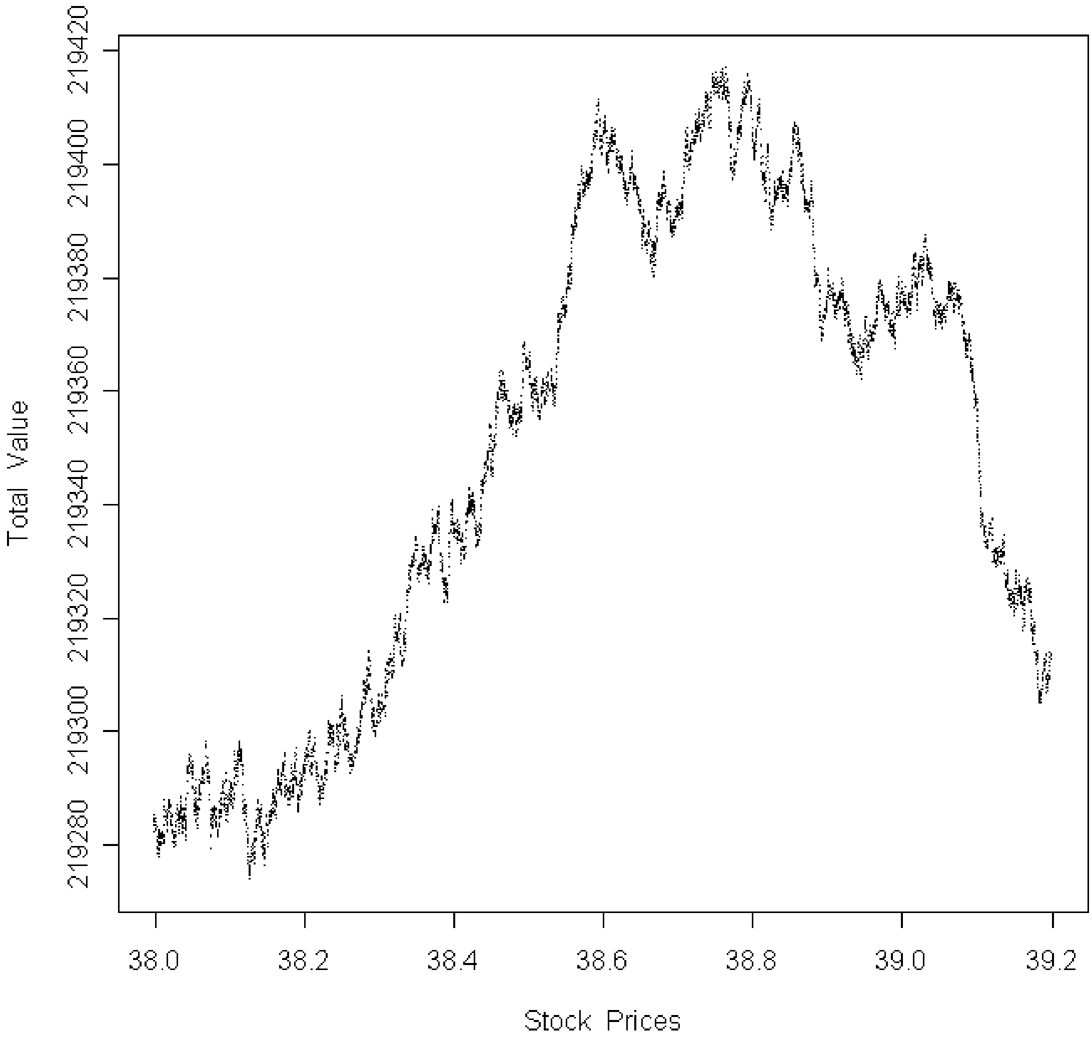

Figure 6.1. Plot of total values for different early exercise thresholds at time t_{d-1}

7. MULTI-ASSET OPTIONS

The rapid pace of innovations in financial markets has resulted in an increasing interest in options written on multiple assets (Detemple *et al.*, 2003).

According to Das (2004), multi-asset options form a subgroup of exotic options. Multi-Asset options are exotic options where the payoff depends on two or more assets. Thus, the correlation between the underlying asset prices must be considered as a factor in the behavior of the option. Options on the maximum of several assets, spread and basket options are examples of multi-asset options.

The primary reason for the use of multi-asset options is to hedge or trade specific types of relative asset price movements (Das, 2004). Variation in risk profile and reduction of option premium are other possible reasons for using multi-asset options.

Before giving a brief summary about the general framework and the formulation of multi-asset options, we define and explain two option types that will be priced in further sections; spread options and options on the maximum of k assets, which are classified as rainbow options by Das (2004).

7.1. Types of Multi-Asset Options

In this section, we briefly explain the spread options and option on the maximum of k assets. In addition, the formulae to calculate the exact price of a European option on the maximum of two assets and the formulae of Kirk's approximation, which is an approximation method for European-style spread options, are given.

7.1.1. Options on the Maximum of k Assets

Options on maximum of k assets, in other words maximum options or best-of options, are options whose payoff depends on the difference between the maximum price of k assets and the strike price (Detemple *et al.*, 2003). Let us assume there is an option on the maximum of two assets. Then, its payoff function for a call is

$$h_c\left(S^1, S^2\right) = \max\left(\max\left(S^1, S^2\right) - K, 0\right)$$

and for a put, the function becomes

$$h_p\left(S^1, S^2\right) = \max\left(K - \max\left(S^1, S^2\right), 0\right)$$

where S^1 and S^2 represent the prices of assets, K denotes the exercise price and Q^1 and Q^2 are the quantities of the asset. In practice, generally quantities are assumed as $Q^1 = Q^2 = 1$.

According to Detemple *et al.* (2003), an option on the maximum of several assets is one of the fundamental building blocks of complex financial products and capital budgeting operations. Das (2004) asserts that asset managers generally use this type of options. Commonly they are written on baskets of equity stocks or equity indexes as well as on the performance of different asset classes such as equity and fixed income/bonds.

7.1.1.1. Exact Price of a European call on the maximum of two assets. The price of a European option on the maximum of k assets has a closed form solution only when the number of underlying assets is two. Haug (2007) gives the formulae to find the exact value of a European option on the maximum of two assets in Chapter 5. Due to the case that will be analyzed, only the formulation for the price of a call is given in this section. The price of a European call option on the maximum of two assets is:

$$\begin{aligned} V_c^{Maximum} = {} & S_0^1 e^{(b_1-r)T} M\left(y_1, d; \rho_1\right) + S_0^2 e^{(b_2-r)T} M\left(y_2, -d + \sigma\sqrt{T}; \rho_2\right) \\ & - K e^{-rT}\left[1 - M\left(-y_1 + \sigma_1\sqrt{T}, -y_2 + \sigma_2\sqrt{T}; \rho\right)\right] \end{aligned} \quad (7.1)$$

where

$$\sigma = \sqrt{\sigma_1^2 + \sigma_2^2 - 2\rho\sigma_1\sigma_2} \tag{7.2}$$

$$d = \frac{\ln\left(S_0^1/S_0^2\right) + \left(b_1 - b_2 + \sigma^2/2\right)T}{\sigma\sqrt{T}} \tag{7.3}$$

$$y_1 = \frac{\ln\left(S_0^1/K\right) + \left(b_1 + \sigma_1^2/2\right)T}{\sigma_1\sqrt{T}} \tag{7.4}$$

$$y_2 = \frac{\ln\left(S_0^2/K\right) + \left(b_2 + \sigma_2^2/2\right)T}{\sigma_2\sqrt{T}} \tag{7.5}$$

$$\rho_1 = \frac{\sigma_1 - \rho\sigma_2}{\sigma} \tag{7.6}$$

$$\rho_2 = \frac{\sigma_2 - \rho\sigma_1}{\sigma} \tag{7.7}$$

and $M(.,.;.)$ is the cumulative distribution function of bivariate standard normal distribution. S_0^i, b_i, σ_i are respectively initial price, cost-of-carry and volatility for the asset i, where $i = 1, 2$. R codes that calculate the exact price of a European call option on the maximum of two assets according to the formulas can be seen in Appendix A.6.

7.1.2. Spread Options

A spread option is an option type whose value depends on the difference between the prices of two or more assets (Haug, 2007). The payoff of a call spread option on two assets is:

$$h_c\left(S^1, S^2\right) = \max\left(\left(Q_1S^1 - Q_2S^2\right) - K, 0\right) \tag{7.8}$$

where S^1 and S^2 are the asset prices. Notice that if the spread option is a put, then the payoff function is:

$$h_p\left(S^1, S^2\right) = \max\left(K - \left(Q_1S^1 - Q_2S^2\right), 0\right)$$

Das (2004) asserts that spread options can be used to capture price differentials between commodities that are related in two ways; demand substitution or transformation potential. Demand substitution refers to the substitution between alternative investments such as bonds or equity indexes. Transformation potential is the term for assets where one asset is a price input into the other.

Underlying assets can be any type of financial products such as equities, bonds and currencies (Haug, 2007). The commodity spreads that are based on the difference between the inputs and outputs of a process enable the trader to gain exposure to the production process of the commodity (Investopedia, 2008). Crack and spark spreads are common instances for this type of options.

Spread options are generally traded in the over-the-counter market. American spread options on oil products are traded in the New York Mercantile Exchange (Haug, 2007). For instance, the heating oil crack is a spread option on the heating oil and crude oil whereas the gasoline crack is a spread option on unleaded gasoline and crude oil. These types of options are used for hedging purposes by oil refineries. In addition, Das (2004) mentions that interest rate spread options are widely used in the US market. He adds that another area of significant activity is the practice of cross-market spread options between the major reserve currencies such as US dollars, Euro and Yen.

7.1.2.1. Kirk's approximation for Spread Options. Kirk (1995) approximated the price of a European spread option by using the Black-Scholes model by a set of transformation. According to Kirk's approximation, the value of a call V_c^{spread} or the value of a put V_p^{spread} can be calculated by:

$$V_c^{spread} \approx \left(Q_2 S_0^2 e^{(b2-r)T} + Ke^{-rT}\right)\left[SN\left(d_1\right) - N\left(d_2\right)\right] \tag{7.9}$$

$$V_p^{spread} \approx \left(Q_2 S_0^2 e^{(b2-r)T} + Ke^{-rT}\right)\left[N\left(-d_2\right) - SN\left(-d_1\right)\right] \tag{7.10}$$

where

$$d_1 = \frac{\ln(S) + (\sigma^2/2)\,T}{\sigma\sqrt{T}}, d_2 = d1 - \sigma\sqrt{T} \quad (7.11)$$

$$S = \frac{Q_1 S_0^1 e^{(b_1 - r)T}}{Q_2 S_0^2 e^{(b_2 - r)T} + K e^{-rT}} \quad (7.12)$$

and the volatility can be approximated by

$$\sigma \approx \sqrt{\sigma_1^2 + (\sigma_2 F)^2 - 2\rho\sigma_1\sigma_2 F} \quad (7.13)$$

where

$$F = \frac{Q_2 S_0^2 e^{(b_2 - r)T}}{Q_2 S_0^2 e^{(b_2 - r)T} + K e^{-rT}} \quad (7.14)$$

and

S_0^1 : Initial Price of Asset One

S_0^2 : Initial Price of Asset Two

Q_1 : Quantity of Asset One

Q_2 : Quantity of Asset Two

K : Strike Price

T : Time to expiration

b_1 : Cost-of-carry of Asset One

b_2 : Cost-of-carry of Asset Two

r : Risk-free interest rate

σ_1 : Volatility of Asset One

σ_2 : Volatility of Asset One

ρ : Correlation

R codes of the Kirk's approximation can be seen in Appendix A.7.

7.2. Simulation of multi-asset options

While simulating multi-asset options, it is obvious that we need to simulate prices of a number of assets according to the dimension. Equation 3.1 is a SDE for the one-dimensional case. For multi asset options, prices of assets at a time t S_t^i, where $i = 1, \ldots, k$ and k is the number of assets, SDE becomes:

$$dS_t^i = \mu_i S_t^i dt + \sigma_i S_t^i W_t^i \tag{7.15}$$

In the examples considered for the multi-dimensional case, generally assets are dividend-paying. Hence, the continuous dividend yield should be included in the drift μ under the risk neutral measure (Wilmott, 2006). Thus, the drift should be replaced by the difference of risk-free interest rate r and continuous dividend yield δ. The Equation 7.15 becomes

$$dS_t^i = (r - \delta_i)\, S_t^i dt + \sigma_i S_t^i W_t^i \tag{7.16}$$

Then, the prices of the assets at a time t can be simulated by the following equation:

$$S_t^i = S_0^i \exp\left(\left(r - \delta_i - \frac{1}{2}\sigma_i^2\right) t + \sigma_i \sqrt{t} Z^i\right)$$

Notice that processes Z^i are correlated due to the correlation between asset prices. Cholesky decomposition helps us to transform the uncorrelated geometric Brownian motions (GBMs) into the correlated ones. It is explained in detail in the next subsection.

7.2.1. Cholesky decomposition

According to Cholesky factorization, a symmetric positive-definite matrix A can be decomposed into a lower triangular matrix L and the transpose of the lower triangular matrix L:

$$A = L.L^T$$

In Monte Carlo simulation, the Cholesky decomposition enables us to generate correlated variates with a transformation of uncorrelated variates (Wilmott, 2006). Let us suppose we are given a correlation matrix A and we have generated matrix $Z = (Z_1, \ldots, Z_k)'$ of k independent standard normal variates. Then, correlated variates Z_c can be obtained by the transformation:

$$Z_c = L.Z$$

where L is the Cholesky decomposition of A with $A = L.L^T$. Thus, for simulating correlated asset prices, we must use Cholesky decomposition.

8. PRICING MULTI-ASSET AMERICAN OPTIONS

In this section, we price three exotic high-dimensional American options that exist in the literature. One of them is a spread option and the other two are basket options on two and five assets. For options on the maximum of k assets, the results calculated via our LSM algorithm are compared with several methods including LSM algorithm used in the relevant reference. For the spread options, we only compare our price estimates with those of the trinomial tree method.

8.1. Simulation of American-style Multi-asset Options

It is obvious that pricing high-dimensional options is more complicated for American versions due to the early exercise opportunity. The pricing methods such as lattices and finite difference methods, which work well for American options on a single asset, require work that is exponential in the number of state variables (Broadie and Glasserman, 1997).

Monte Carlo simulation has the distinct advantage of having a convergence rate, which is independent of state variables (Broadie and Glasserman, 1997). Besides, it is easy to handle different types of models and payoff structures.

In simulation of multi-asset options, correlation between the assets is a fact that must not be neglected. By Cholesky decomposition, it is simple to simulate prices of assets that are correlated and it almost does not require any extra computational effort.

Altogether, due to the pros of Monte Carlo simulation regarding its convergence rate and exponential increase of required work in the other methods, Monte Carlo simulation is especially appropriate for pricing multi-asset derivatives.

8.2. LSM Algorithm for Multi-Asset American Options

As mentioned in Section 4.3, the LSM algorithm can be modified for multi-dimensional cases. Longstaff and Schwartz (2001) price an option on the maximum of five assets by using the LSM algorithm. They have obtained a result within the tightest confidence interval calculated by the stochastic mesh method of Broadie and Glasserman (1997). Also, the LSM algorithm needs one to two minutes while the computation of the stochastic mesh method takes about 20 hours.

It is possible and easy to modify the algorithm for multi-dimensional cases. The addition in the algorithm for multi-asset options is to generate prices of more than one asset and to extend the basis functions as they include prices of all assets. Of course, while generating the asset prices, the correlation should be considered. However, not much work is required to modify the algorithm for the multi-dimensional case.

Let $X_0, X_1, \ldots, X_d$ denote a $\Re^k$ valued Markov chain describing all relevant financial information where X_0 is the fixed initial information, k is the number of assets and d is the number of exercisable time points. The algorithm given in 4.3 is still valid for the multi-dimensional case. The only difference is in the dimension of the Markov chain. To summarize:

i. Simulate prices of assets for n independent paths of the Markov chain $\{X_0, X_1, \ldots, X_d\}$ by using Cholesky decomposition
ii. At terminal nodes, set continuation values to the payoff function,
iii. Apply backward induction: for $i = d-1, \ldots, 1$,
    - Fit regression by using the value of the option at that time steps and basis functions to estimate the coefficients.
    - Estimate the continuation values according to the estimated coefficients by Equation 4.1.
    - Compare the estimated continuation values and the income from early exercise, and decide to exercise or not
iv. Take the average of the incomes in the paths and discount it to find the price of

the option.

8.3. American Options on the Maximum of Two Assets

We gave brief information about American options on the maximum of k assets in Section 7.1.1. This section contains the explanation of the possible control variable and R implementation for pricing an American option on the maximum of two assets. It also includes the numerical results on the example analyzed by Glasserman (2003).

8.3.1. Price of European Option on the Maximum of Two Assets as Control Variable

The price of an American option is related to the price of its European equivalent. Since it is possible to calculate its exact value, the price of a European option can be used as control variable while applying CV to pricing an American option. The equations from 7.1 to 7.7 given in Section 7.1.1.1 enable us to compute the exact price of a European option on the maximum of two assets. Hence, for the American options on the maximum of two assets, European style option can be selected as control variable in simulation. Our R implementation includes the application of CV with this control variable.

8.3.2. R Implementation

In the R implementation of pricing American options on the maximum of two assets, beside the antithetic variates, we also used the control variates. Since we know the closed form solution of the European equivalent of the option we price, it can be used as control variable. Chapter 5 of Haug (2007) includes the formula of the price of a European option on the maximum of two assets. The key in this simulation is the determination of the parameter c. It is possible to calculate c according to Equation 3.11 by using the payoff of the American option and the payoff of the European option, which are the results of a pilot study simulation with a smaller size of sample, for example a simulation with 1000 paths. However, to shorten the R code, we define parameter c as an input to the function instead of calculating it by a pilot study within the function. Appendix A.8 includes the implementation that is explained in this section. The version with the pilot study can be found in Appendix A.9.

The function *simul_US_max_of_2* prices an American call option on the maximum of two assets by using LSM algorithm. In the R implementation, we first calculate the time interval. Since we do not use polynomials, there is no need to normalize all values by dividing with the exercise price. Next, according to the given correlation, the correlation matrix is formed. By using the Cholesky decomposition of the correlation matrix, we convert the uncorrelated random variates into correlated ones. For this purpose, we generate standard normal variates and store them in a matrix. The matrix multiplication of L, which is transpose of Cholesky decomposition of correlation matrix, and the matrix z, which consists of independent variates, gives us the correlated variates. Notice that we use the function chol for Cholesky decomposition and syntax %*% for the multiplication of matrices. Since we do not need the independent variates anymore, the correlated ones are stored in z. After these operations, we generate the terminal prices of assets by using the new matrix z. Different to the one-dimensional case, we include the dividend terms of each asset while simulating the prices. In addition, the stock prices generated by antithetic variates are saved in the same vector for easy calculations. Before the for loop, we assign the payoff to array CC, which will be discounted for being an input for regression at time t_{d-1}.

Within the for loop, we first simulate asset prices at that time step. Therefore, we again generate independent standard normal variates and transform them by using the matrix L. After the simulation of asset prices in the current time step, in-the-money points are determined to fit regression. Similar to the one-dimensional implementation, *discountedCC*, which is composed of discounted cash-flows, and basis functions are formed to be input for the regression to be fitted. The continuation values are estimated according to the regression coefficients and the comparison between estimated continuation values and income from the early exercise is made. CC is updated due to the comparison. To go to the previous time step, asset prices at the current time step are assigned as future prices.

In the end of the *for* loop, the payoff of all paths and the option price are computed. According to the control variable, payoff is controlled by using the parameter c and the exact value of the control variable. Within the code, the function *EU_max_of_2* calculates the exact price of the European equivalent. Then, the price estimate found by using the controlled payoff "payoff.c" and the standard error of the simulation are printed out.

Meanwhile, it should be mentioned that instead of using Cholesky decomposition, a function named mvrnorm, which is defined in the R package MASS, could be used. The function mvrnorm generates samples from the multivariate normal distribution; inputs are the dimension, mean, standard deviation and the variance-covariance matrix. Of course this function uses Cholesky decomposition as well.

8.3.3. Numerical Results

We use Example 8.6.1 from Glasserman (2003), which is also used in Andersen and Broadie (2004) and Broadie and Glasserman (2004). It is a call option on the maximum of two assets, S_1 and S_2. Hence, its payoff function h is;

$$h\left(S^1, S^2\right) = \max\left(\max\left(S^1, S^2\right) - K, 0\right)$$

The initial asset prices are equal to each other and the exercise price K is three. The risk-free interest rate r is 0.05. Volatilities and dividend yields, which are considered as continuous, are same for each asset and are sequentially 0.2 and 0.1. It is assumed that there is no correlation between the assets. This means they are independent of each other. The option has nine equally spaced exercisable dates and expires in three years.

Glasserman (2003) uses three cases by varying the starting asset prices. The summarized results can be seen in Table 8.1. In Table 8.1, the column LSM includes our price estimates by using seven basis functions; a constant, asset prices, squares of asset prices, multiplication of asset prices and the payoff function. According to

Table 8.1. Price estimates for a two asset American maximum call option

S_0	Closed Form European	Binomial Model	95% CI	LSM	VRF (AV)	VRF (AV & CV)
90	6.5551	8.075	$[8.053, 8.082]$	8.0598	2.487066	4.15552
100	11.1957	13.902	$[13.892, 13.934]$	13.9001	2.747369	4.023047
110	16.9286	21.345	$[21.316, 21.359]$	21.320	3.109262	3.938483

Glasserman, this choice of basis functions is the set with the minimum number of elements that gives the best results. Furthermore, closed form European prices are computed by the formula in Haug (2007) and the 95 percent confidence interval (CI) is the interval computed by Andersen and Broadie (2004). Besides these, price estimates of the binomial tree are calculated by using the multi-asset binomial method of Boyle et. al (1989) which has a error of 0.003. Moreover, VRF indicates the variance reduction factor for applying antithetic variates (AV) and control variates (CV) to the LSM algorithm. The closed form European price is used as control variable for each case, assuming that it is the exact value. In the table, VRFs when only AV is applied and when both AV and CV are applied are given separately.

According to Table 8.1, our price estimates with LSM algorithm, which have standard error of 0.002, are in the 95 percent confidence interval computed by Andersen and Broadie (2004). Also, as it can be noticed that AV is a more effective variance reduction method when the option is deeply in the money due to the initial starting asset prices. If the option starts out of the money, then simulation results in a lower variance reduction factor. In addition, CV is not as efficient as AV. However, when it is applied, it decreases the variance as well. In opposition to AV, the efficiency of CV decreases when the option starts deeply in the money.

It is important to stress that in the example Glasserman (2003) priced, there is no correlation between assets. However, our implementation given in Section 8.3.2 is capable of pricing any American option on the maximum of two assets even when the assets are correlated. In addition, the variance reduction factor of the simulation increases as assets are correlated.

<u>8.3.3.1. Early Exercise Region.</u> The early exercise region of an option written on more than one asset is getting smaller as we go backwards in time. However, LSM algorithm fits the regression by using all in-the-money points, which is the exercise region at maturity.

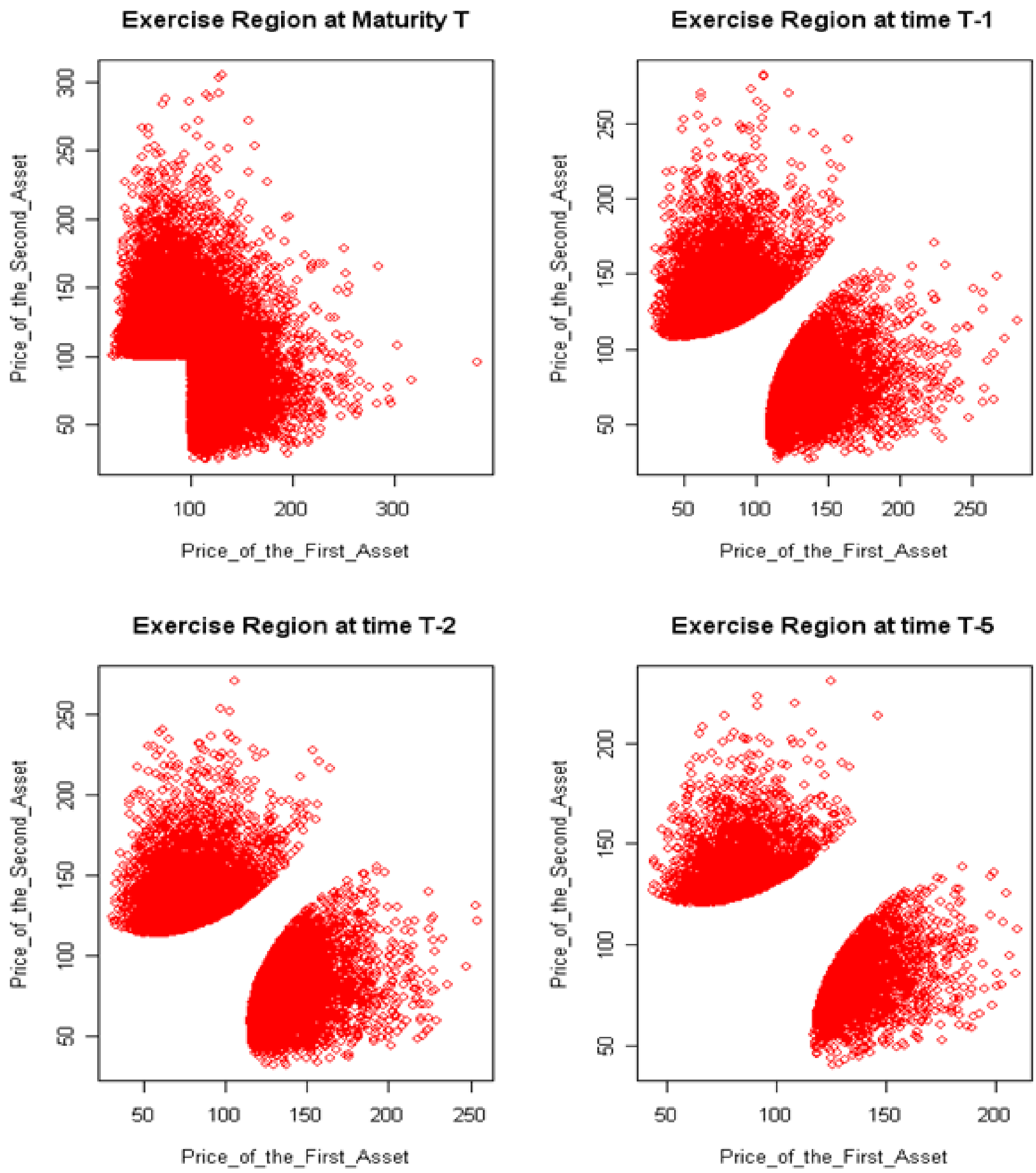

Figure 8.1. Early exercise region of an American call option on the maximum of two assets

Figure 8.1 shows the early exercise region of an American option on the maximum of two assets at different time points. It can be seen that the exercise region is composed of two sub regions. Furthermore, as we get closer to the start, the exercise region is getting smaller and the sub regions are getting farer apart from each other. The inefficiency of the LSM algorithm related to the selection of input points to the regression can also be seen from the plots in Figure 8.1. How to reduce the area used for the regression is an open question.

8.3.3.2. Number of Time Steps versus Early Exercise Premium. It was mentioned earlier that a Bermudan option, which has discrete early exercise opportunities, is placed between American and European options. As we price Bermudan options by simulation, it might be possible to determine the number of time steps necessary that the price of the Bermudan option gets equal to the American price.

Since LSM algorithm estimates the option price low, the number of time steps which results in the maximum early exercise premium might be selected for pricing. So, Bermudan option we priced might be closer to its American equivalent. Also, if we know the early exercise premium regarding to the early exercise property, it will be easy to find the price of the American option by using the closed form solution of the European option and the early exercise premium.

Figure 8.2 shows the early exercise premium of the case we priced in the previous section for different numbers of time steps. The algorithm is run with 50000 antithetic pairs and the estimates have standard errors ranging from 0.006 to 0.008. As it is seen in Figure 8.2, the increase in early exercise premium is not proportional to the increase in the number of time steps. As we consider more time steps in simulation, the pace of increase in the premium decreases and the premium converges to a value. The simulation time is linearly increases with the number of time steps. Hence, we do not want to allow too many exercise opportunities in the algorithm due to the very small changes in the premium. Broadie and Glasserman (2004) price this option by considering nine time steps and call it as American. This experiment shows that there

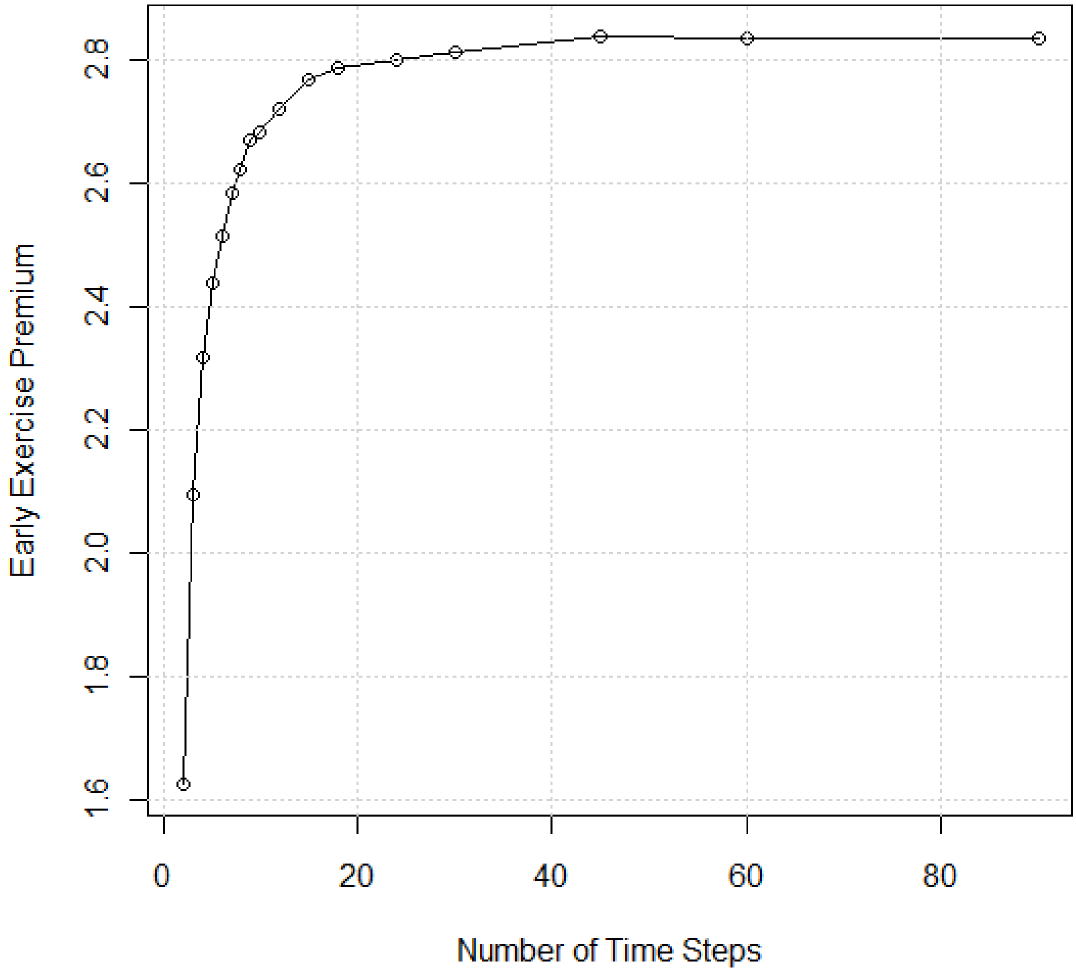

Figure 8.2. Total number of time steps versus early exercise premium for an American call option on the maximum of two assets

is a significant difference for the early exercise premium. Considering nine time steps results in a premium of 2.667936 but as it is 2.838814 when the option is priced with 45 time steps. Notice that more than 45 time steps increases the computational cost without changing the price.

It was mentioned that there is a closed form solution for the European option on the maximum of two assets. As the price of the American option is the sum of the price of its European equivalent and the early exercise premium, it is possible to find the price of the American option by using the premium. Table 8.2 summarizes the

Table 8.2. Price estimates of binomial tree and LSM with different number of time steps for a two asset American maximum call option on two assets

S_0	Binomial Tree	LSM	LSM^*
90	8.075	8.0598	8.152799
100	13.902	13.9001	14.03451
110	21.345	21.32	21.53642

price estimates for the same American maximum call option, which was priced by using nine time steps. The estimates in the column LSM are results calculated earlier (with nine time steps) whereas the ones in the column LSM^* are calculated with 45 time steps. There is a significant differences in the estimated prices whatever the initial asset price is. The values calculated by LSM^* are higher. Since we know that the American option price is larger than the Bermudan one, it shows that there is a better convergence of Bermudan option to its American equivalent if we use 45 time steps. In other words, LSM estimates more precisely when 45 time steps are used.

8.4. American Options on the Maximum of Five Assets

After pricing an American option on the maximum of two assets, we consider the case where the payoff is on the maximum of five assets. Longstaff and Schwartz (2001) compare their results with the results of Broadie and Glasserman (1997) when they assert that their LSM algorithm works well for multi-dimensional cases. The implementation of this type of option in R differs from the implementation of the option on the maximum of two assets only in the number of assets and the basis functions used. Because of its length, the implementation using the basis functions stated by Longstaff and Schwartz (2001) with AV is given in Appendix A.10. Numerical results are included in this section.

8.4.1. Numerical Results

Broadie and Glasserman (2004) apply their stochastic mesh method to an American call option on the maximum of five assets, which has the payoff function:

$$h\left(S^1, S^2, S^3, S^4, S^5\right) = \max\left(\max\left(S^1, S^2, S^3, S^4, S^5\right) - K, 0\right)$$

It is assumed that the assets are not correlated. The option has a three year life and has nine exercise opportunities, in other words, exercisable three times per year. Volatility of each asset is 20 percent. The assets pay 10 percent continuous dividend. The risk free rate is 5 percent and the strike price is 100. The initial stock prices are assumed to be the same for all assets and taken as 90,100 and 110.

Table 8.3 shows the Monte Carlo simulation results for the price of the European option, the confidence interval for the American option, which is computed by Andersen and Broadie (2004), our price estimates for the American option, which are calculated by LSM algorithm using 50000 paths and 19 basis functions as described in the paper of Longstaff and Schwartz (2001). These 19 basis functions are a constant, the first five Hermite polynomials in the maximum values of the five assets, the four asset prices and squares of the prices of the second through fifth highest asset prices, the product of the highest and second highest, second highest and third highest, etc. The price estimates of LSM algorithm have standard errors ranging from 0.004 to 0.007. It is remarkable to notice that LSM estimates are within the 95 percent confidence interval. This was the example used by Longstaff and Schwartz (2001) to show that their algorithm works well for multi-dimensional options. In addition, VRFs reached by using AV can be seen in the table.

It should be remarked that since we do not know the exact price of the European equivalent of the option, we cannot apply CV. In addition, like for the maximum option on two assets, our implementation is also working when the asset prices are correlated.

Table 8.3. Price estimates for an American call option on the maximum of five assets

S_0	Monte Carlo EU	95% CI	LSM	VRF
90	14.581	$[16.602, 16.655]$	16.6159	2.324062
100	23.0493	$[26.109, 26.292]$	26.1015	2.390550
110	32.6993	$[36.704, 36.832]$	36.725	2.39652

Table 8.4. Price estimates for an American call option on the maximum of five assets with different number of time steps

S_0	Monte Carlo EU	95% CI	LSM	LSM^*
90	14.581	$[16.602, 16.655]$	16.6159	16.89765
100	23.0493	$[26.109, 26.292]$	26.1015	26.42993
110	32.6993	$[36.704, 36.832]$	36.725	37.1323

8.4.1.1. Number of Time Steps versus Early Exercise Premium. We question how many exercise opportunities should be considered in pricing American options on the maximum of five assets. Figure 8.3 shows the early exercise premium for different numbers of time steps. According to Figure 8.3, it is clear that, like in maximum option on two assets, as the number of time steps increases, the increase in the premium gets slower and it almost converges to a fixed value when the number of time steps is 45. However, Broadie and Glasserman (2004) and Longstaff and Schwartz (2001) price this American option by using only nine time steps and there is a significant increase in premium between pricing by considering nine time steps and 45 time steps.

For the price of the option to reflect the American property, we price it with 45 exercise opportunities. The results can be seen in Table 8.4, in the column named LSM^*. The values in the column LSM are estimates with nine time steps. Since we price American options in simulation as Bermudan, the results of LSM^* are closer to be American than the estimates of Broadie and Glasserman (2004) and Longstaf and Schwartz (2001).

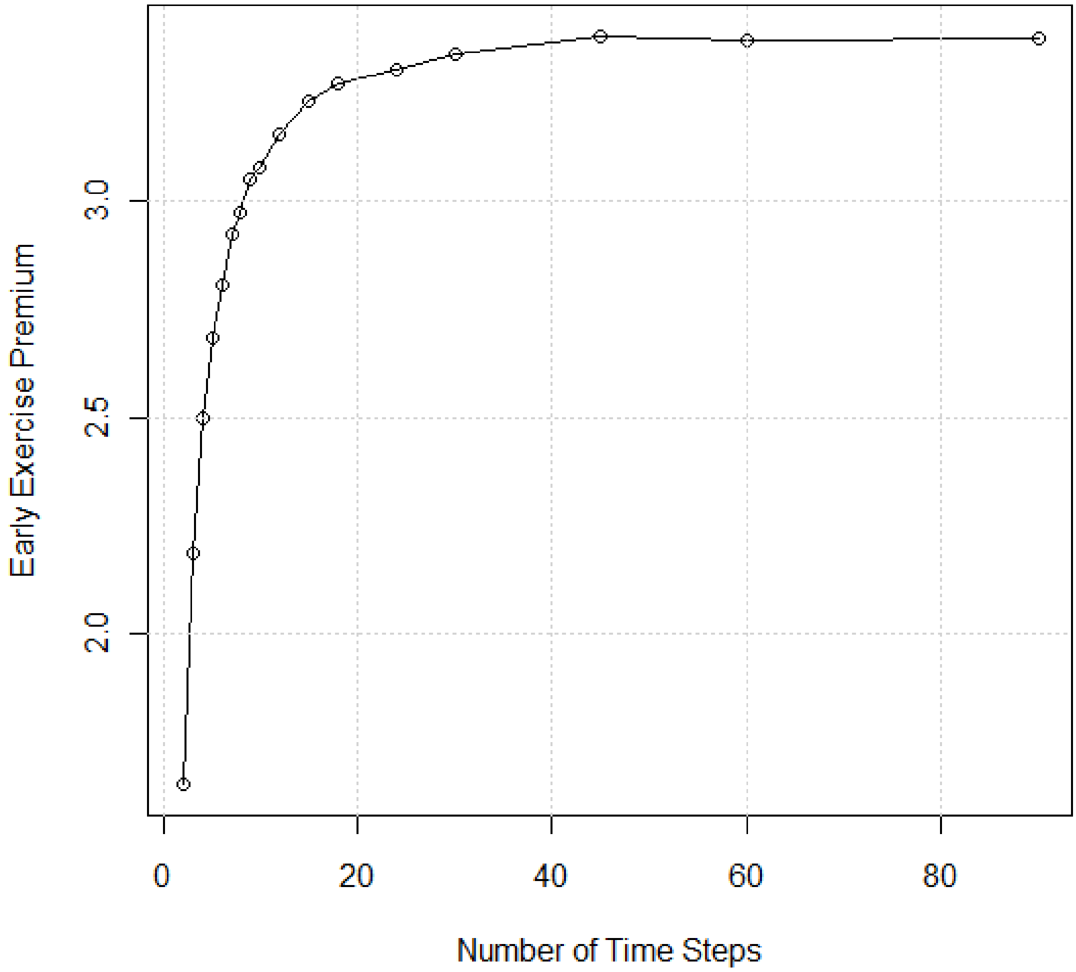

Figure 8.3. Total number of time steps versus early exercise premium for an American call option on the maximum of five assets

8.5. American Spread Options

In Section 7.1.2, we described spread options and gave information about their application area. This section includes the R implementation of the simulation of American spread options and numerical results on a case priced by Haug (2007). We analyze this case to find a set of basis functions that results in good price estimates and to illustrate the early exercise region of an American spread option. In addition, we make experiments to answer the question how many exercisable time steps should be considered in pricing an American spread option to have a close to exact value of the early exercise premium, which is the difference between the European and the American price.

8.5.1. European Spread Option Price as Control Variable

There is no closed form solution for European spread option. However, Kirk (1995) approximated the price of a European spread option. By Monte Carlo simulation with a huge sample size, which is quite fast, it is possible to get a price estimate for a European spread option that is close to the exact price. In this section, we compare price estimates of binomial tree, Kirk's approximation and simulation to discuss whether one of them can be used as control variable in pricing American spread options. The R implementation of pricing a European spread call by simulation is included in Appendix A.11.

Table 8.5 summarizes the price estimates of binomial tree, which is coded by Haug (2007), approximation of Kirk (1995) and Monte Carlo simulation with 10 replication of 1000000 paths for a European spread option with initial stock prices $S_0^1 = 122$ and $S_0^2 = 120$. The quantitaties of the stocks are equal to one. The life of the option, the correlation between assets and the volatilities are varied within the table.

According to Table 8.5, price estimates of Kirk's approximation and Monte Carlo simulation are close to each other for all cases. We know that Monte Carlo simulation converges to the true option value when a huge sample size is used. Thus, Kirk's approximation, which gives almost identical results with the simulation, can be assumed as the exact price of the European spread option and it allows us to use the European spread option price as the control variable in pricing an American spread option. Hence, in the implementation of pricing American spread options with control variates, we will use Kirk's approximation as the exact price of the European spread option.

Table 8.5. Price estimates of European spread call using Binomial tree, Kirk's approximation and Monte Carlo simulation

			T=0.5			T=1		
Method	σ_1	σ_2	$\rho = -0.5$	$\rho = 0$	$\rho = 0.5$	$\rho = -0.5$	$\rho = 0$	$\rho = 0.5$
Tree	0.20	0.20	10.7566	8.708	6.0286	14.6148	11.8706	8.2824
Kirk's Approx.			10.7517	8.70198	6.02572	14.61008	11.86839	8.278906
Simulation			10.751	8.70067	6.02693	14.6068	11.86757	8.276589
Tree	0.25	0.20	12.2031	9.9377	7.0097	16.5453	13.521	9.5992
Kirk's Approx.			12.1941	9.93397	7.00672	16.53485	13.51713	9.595823
Simulation			12.1927	9.93385	7.00648	16.53468	13.51764	9.592516
Tree	0.20	0.25	12.1521	9.8811	6.9323	16.4762	13.4451	9.4954
Kirk's Approx.			12.1483	9.87802	6.92836	16.47387	13.4423	9.490681
Simulation			12.1482	9.87698	6.92588	16.47266	13.44079	9.490336

8.5.2. R Implementation

R codes to price American spread options without a pilot study for CV can be found in Appendix A.12. The function *simul_US_spread* price an American spread option by using the LSM algorithm. This implementation does not differ much from the early implementations of pricing other types of multi-asset options. The antithetic variates method cannot be used due to the payoff structure. However, the price of the European spread option is used as control variable to apply CV and its exact value is assumed to be the result of Kirk's approximation. The function takes the parameter c, for CV, as an input. The version with the pilot study can be seen in Appendix A.13.

We first calculate the time interval before forming the correlation matrix. Using Cholesky decomposition of the correlation matrix, the uncorrelated random variates are converted into correlated. After the generation of the asset prices, the payoff is assigned to the array CC, which will be an input for regression at time t_{d-1} after the discounting. Before the for loop, the payoff of the European style is stored to use it while applying CV.

The content of the *for* loop within the implementation is, except for the payoff structure, similar to the loops in the other implementations. It includes the generation of the asset prices by using the Cholesky decomposition, fitting the regression according to the basis functions, the comparison between the estimated continuation values and the immediate income from payoff and the assignment of the asset prices for the next step of the loop.

In the end of the loop, the payoff is calculated using the parameter c and the exact value of the options European equivalent. The exact value is accepted to be the result of Kirk's approximation. The function *kirk_approximation* computes the price of the European spread option. After CV calculations, the function prints the price estimate and the standard error of the simulation.

8.5.3. Numerical Results

Haug (2007) price an American spread call option, which has payoff structure as in equation 7.8, by using trinomial tree with 100 time steps. The option is on two assets which have initial prices $S_0^1 = 122$, $S_0^2 = 120$ with quantities $Q_1 = Q_2 = 1$ and the exercise price K is three. The risk free interest rate r is 0.1 and the cost of carry of both assets is zero. It is essential to notice that the cost-of-carry is the difference between the risk-free interest rate r and the continuous dividend yield δ. Thus, if the cost-of-carry is zero, it means dividend yield is equal to the risk-free interest rate. The option is priced with 50 exercise opportunities by varying the time to expiry as well as the correlation between assets and their volatilities.

First, we make experiments for the selection of the basis functions. After illustrating the exercise region, we investigate how many exercisable time points should be considered.

<u>8.5.3.1. Basis Functions.</u> In the experiments, we looked for the best set of basis functions that ends with a price estimate close to the result of the trinomial tree in the

book of Haug (2007). The sets of basis functions tried in the experiments are varied by the payoff function, prices of the assets, their squares and cross-products. These sets can be seen in Table 8.6. The sets that include the payoff as a basis function do not contain the price of the second asset because S^1, S^2 and $h\left(S^1, S^2\right)$ are linearly dependent.

Table 8.6. Sets of basis functions used in pricing an American spread call

Sets	Basis Functions
1	$h(S^1, S^2), S^1, S^1S^2$
2	$h(S^1, S^2), S^1, (S^1)^2, S^1S^2$
3	S^1, S^2, S^1S^2
4	$S^1, S^2, (S^1)^2, (S^2)^2, S^1S^2$
5	$S^1, S^2, (S^1)^2, (S^2)^2, S^1S^2, (S^1)^2S^2, S^1(S^2)^2$
6	$S^1, S^2, (S^1)^2, (S^2)^2, S^1S^2, (S^1)^2S^2, S^1(S^2)^2, (S^1)^2(S^2)^2$

The price estimates for the American spread call option achieved by using different sets of basis of functions are given in Table 8.7. It is seen that the sixth set, which is composed of the asset prices, squares of the asset prices and three cross-multiplication of asset prices is the appropriate choice for basis functions because we know that LSM algorithm estimate the price low and this choice of basis functions results in better price estimates. Also, the estimates of the fifth set are close to the those of the sixth. Thus, to reduce the computational time even more, the fifth set may be chosen for valuation, however, the decrease in time consumed by using less basis functions is not significant. As a result of this experiment, we will continue pricing American spread options by using the basis functions of the sixth set. Furthermore, the price estimates of LSM with the sixth set are not identical to those of trinomial tree. However, we do not know whether the results of the trinomial tree are 100 percent exact values of the option. Also, it estimates the price of European option always higher than Kirk's approximation and simulation as shown in Section 8.5.1.

8.5.3.2. Variance Reduction. After explanaining why it is possible to use a European spread option as control variable under the assumption its exact price is the result

Table 8.7. Price estimates for American spread call option by using different sets of basis functions in pricing algorithm

			T=0.5			T=1		
Method	σ_1	σ_2	$\rho=-0.5$	$\rho=0$	$\rho=0.5$	$\rho=-0.5$	$\rho=0$	$\rho=0.5$
LSM - Set1	0.20	0.20	10.83578	8.755928	6.064245	14.88673	12.10921	8.422175
LSM - Set2			10.8104	8.745786	6.056096	14.90784	12.08815	8.449647
LSM - Set3			10.83052	8.756744	6.065066	14.8974	12.10153	8.424005
LSM - Set4			10.85715	8.780862	6.079356	14.96019	12.13907	8.468816
LSM - Set5			10.87056	8.7908	6.08445	14.98109	12.16715	8.487024
LSM - Set6			10.87228	8.798006	6.085901	14.98986	12.170113	8.48866
Tree			10.8754	8.8029	6.0939	14.9967	12.178	8.4952
LSM - Set1	0.25	0.20	12.28299	10.01362	7.05683	16.91463	13.78644	9.77719
LSM - Set2			12.26572	10.00309	7.044334	16.8846	13.77136	9.774053
LSM - Set3			12.27982	10.014409	7.054606	16.91015	13.78192	9.77025
LSM - Set4			12.31974	10.03117	7.072127	16.93783	13.84705	9.82853
LSM - Set5			12.33751	10.04854	7.078977	16.97063	13.85917	9.839035
LSM - Set6			12.33544	10.0432	7.07943	16.97401	13.86069	9.840375
Tree			12.3383	10.0468	7.0858	16.9789	13.8727	9.8463
LSM - Set1	0.20	0.25	12.24264	9.944552	6.972998	16.81626	13.69899	9.671492
LSM - Set2			12.22076	9.903828	6.967923	16.78713	13.68161	9.656747
LSM - Set3			12.24853	9.940646	6.97869	16.82085	13.69681	9.668236
LSM - Set4			12.26254	9.97314	6.98868	16.87373	13.7772	9.711424
LSM - Set5			12.28186	9.981732	6.99179	16.90122	13.77125	9.72888
LSM - Set6			12.28264	9.98341	6.996226	16.90453	13.77939	9.725018
Tree			12.2867	9.9897	7.0082	16.9085	13.7953	9.7415

of Kirk's approximation, we report the variance reduction factors (VRFs) that are achieved by this method. It should be remarked that antithetic variates are not used for spread options as they do not lead to any variance reduction due to the structure of the payoff function.

Table 8.8 includes the VRFs for the American spread option we price. VRFs change according to the life time of the option, the correlation between the underlying assets and the volatilities. The data in Table 8.8 shows that VRF is negatively corre-

Table 8.8. VRFs in pricing American spread options

		T=0.5			T=1		
σ_1	σ_2	$\rho = -0.5$	$\rho = 0$	$\rho = 0.5$	$\rho = -0.5$	$\rho = 0$	$\rho = 0.5$
0.20	0.20	4.125466	3.925507	3.894836	3.416155	3.398603	3.340055
0.25	0.20	4.217281	3.987225	3.963501	3.48033	3.437527	3.4113
0.20	0.25	3.994254	3.759955	3.712589	3.2358	3.214024	3.165177

lated with the life of the option. As the life of the option increases, the reduction in variance decreases. It is also clear that in all cases, the correlation between assets has a decreasing influence on VRF. It means that if the correlation between the assets is greater than zero, the simulation ends in a VRF that is relatively smaller in the case where the assets are not correlated. On the contrary, the reduction in variance tends to increase if the assets are correlated negatively. Furthermore, the results show that the volatilities of the assets affect the VRF. When the volatility of the first asset, which has the positive sign in the payoff function of a call, is smaller than the volatility of the second asset, the simulation results in the lowest VRFs. If it is assumed that the volatility of the first asset is greater, then we get the highest VRFs.

8.5.3.3. Early Exercise Region. After the implementation of the algorithm in R, it is not hard to have a plot of asset prices where the early exercise takes place, in other words, a plot of the exercise region. Figure 8.4 shows exercise region of an American spread call option at time t_{d-1}. Parameters of the option are assumed to be same as in the example Haug (2007) considered. The line in Figure 8.4 represents the exercise boundary at the expiration date. Since the payoff is as in Equation 7.8, the condition for the option to be exercised at time T is

$$S^1 > S^2 + K$$

which makes the payoff positive. Taking this fact into account we can conclude that the exercise region is getting farer away from the line, which shows that the price of the first asset is equal to the sum of the price of the second asset and the exercise price, as the time to expiration increases. In other words, the exercise region is moving to the right hand side. This means the difference between the prices of the assets should be large to exercise early.

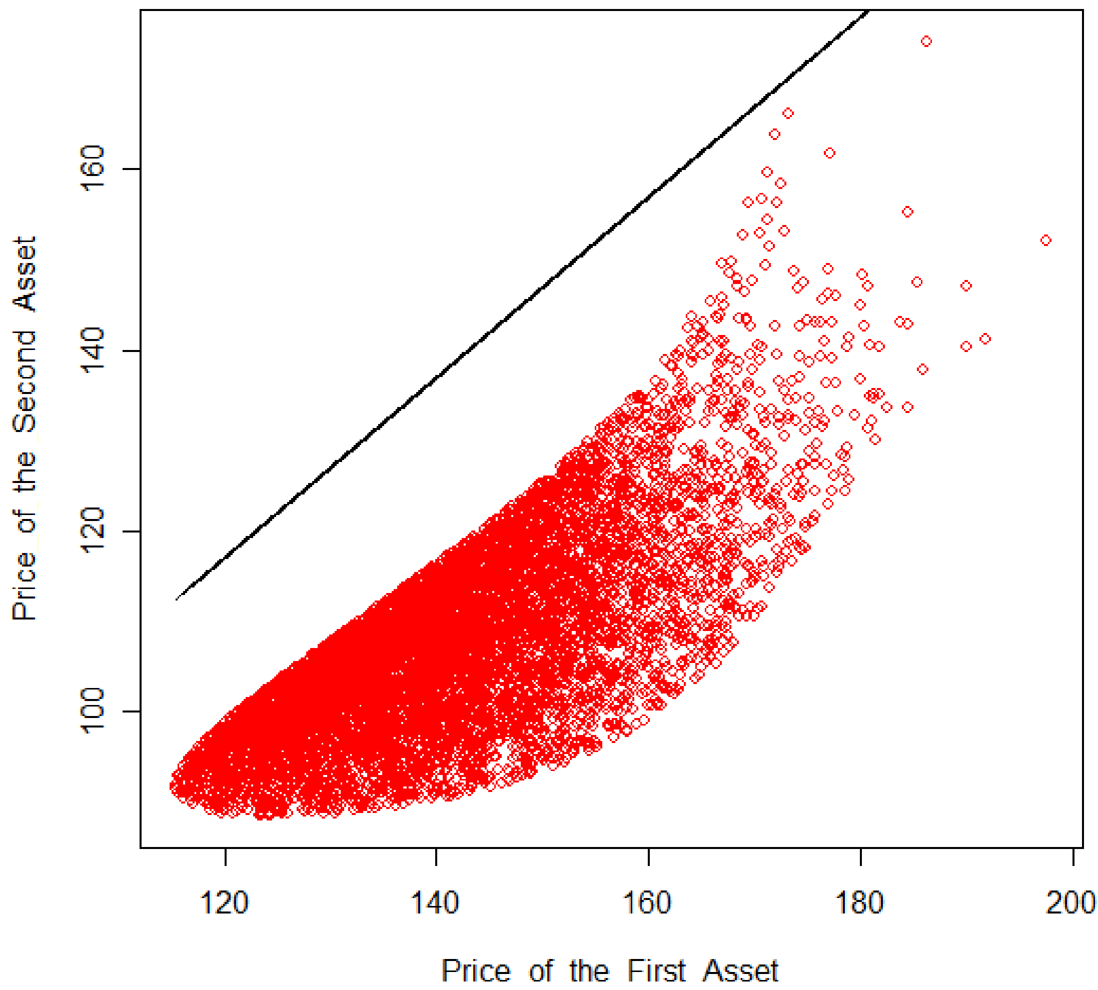

Figure 8.4. Early exercise region of an American spread call option at time t_{d-1}

As we know, LSM algorithm takes all in-the-money points as input for the regression. This is the area to the right hand side of the straight line (for calls) in Figure 8.4. So, LSM algorithm uses all the paths that are to the right hand side of the line. However, it is obvious that for early exercise, the payoff should be larger as we go backwards in time. Thus, to consider all paths in regression leads to computational cost. It might be possible to fit regression in a smaller area to specify the exercise boundary. For calls, at a time step, the minimum difference between the prices of the assets that early exercise takes place can be used as the exercise boundary at that time step. How to reduce the region that is considered as input for regression to determine this boundary is an open question.

8.5.3.4. Number of Time Steps versus Early Exercise Premium. Similarly to the idea for American options on the maximum of two assets, we should consider the number of time steps in the pricing algorithm necessary to obtain price estimates that are sufficiently close to the true value of an American spread option.

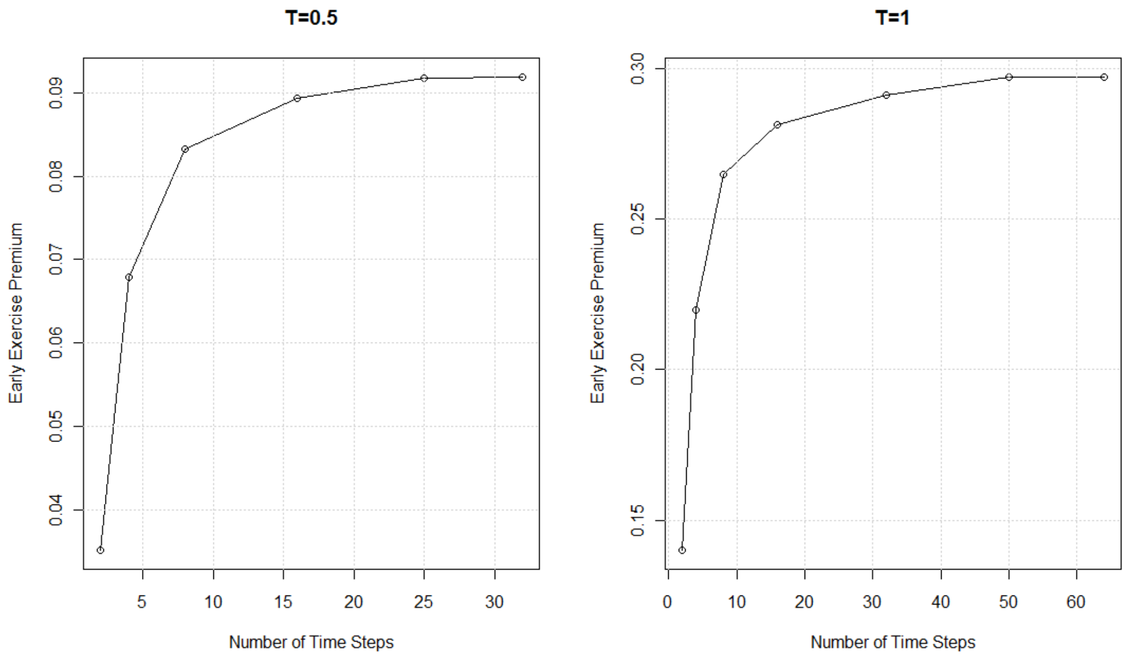

Figure 8.5. Plots of early exercise premium versus number of time steps for American spread call options

Figure 8.5 contains two plots which show early exercise premium for different number of time steps that are powers of two as well as the premium with 50 time steps per year. The left plot is for the life of the option equal to 0.5 years whereas it is one year for the right plot. In both cases, volatilities of the assets are 20 percent and there is no correlation between the asset prices. According to Figure 8.5, it is seen that increasing the total number of time steps considered in pricing algorithm does not always improve the price estimate and that the early exercise premium converges to a certain value. Thus, considering more time steps than a specific number does not significantly improve the price estimate for the American spread option. However, it increases the computational cost. As it can be noticed from the figure, the sufficient number of exercise opportunities per year is 50 under these conditions.

9. CONCLUSIONS

FD and tree techniques are efficient methods to price American options with single underlying security whereas simulation works well better for multi-asset American options. In this study, we focused on the LSM algorithm of Longstaff and Schwartz (2001), which is a regression-based Monte Carlo simulation method for pricing American options.

When we know the early exercise boundary, pricing an American option is quite easy. In the one-dimensional case, the estimate of LSM for the early exercise boundary at the last time step does not match with the result of Black-Scholes formula and Newton Raphson method. This shows that LSM cannot estimate the boundary well. We tried to reduce the inefficiency of LSM algorithm about the input selection for the regression and improved the algorithm. It now estimates the price of an American option with less computational cost and more accurately than the LSM algorithm. Furthermore, we coded an optimization approach, which maximizes the total value at each time step, and we always had results higher than those of FD method. Therefore, it might be possible to use it as an upper bound of the American option price.

For multi-asset American options, we extended the implementation of the LSM algorithm. We analyzed how many early exercise points should be considered for the option to be close to the American price and we showed that the price estimates in the literature are actually too low for American options on the maximum of two or five assets as the number of time steps was selected too small. For American spread options, we tested different sets of basis functions and determined a useful one. We also applied variance reduction techniques for multi-asset options. We showed that it is possible to use Kirk's approximation for European spread option as a control variable on its American equivalent while applying control variates (CV). Considerable variance reduction is achieved by using antithetic variates (AV) and CV for maximum options on two assets, AV for maximum options on five assets and CV for spreads.

APPENDIX A: R CODES

A.1. R Codes for Black-Scholes Formula to Price European Options

```
EU_black_scholes<-function(s0,K,r,sigma,T,flag="c"){
# s0 ... initial asset price
# K ... strike Price
# r ... risk-free interest rate
# sigma ... volatility
# T ... time horizon
# flag ... c for call options

d1<-(log(s0/K)+(r+sigma^2/2)*T)/(sigma*sqrt(T));
d2<-d1-sigma*sqrt(T);
if(flag=="c"){
        s0*pnorm(d1)-K*exp(-r*T)*pnorm(d2)
        }else{
        K*exp(-r*T)*pnorm(-d2)-s0*pnorm(-d1)
        }
}
```

A.2. R Codes for the Implementation of the LSM Algorithm

```
simul_US_LSM<- function(n,d,s0,K,sigma,r,T){
s0<-s0/K
dt <- T/d
z<-rnorm(n)
s.t<-s0*exp((r - 1/2*sigma^2)*T+sigma*z*(T^0.5))
s.t[(n+1):(2*n)]<-s0*exp((r - 1/2*sigma^2)*T-sigma*z*(T^0.5))
CC<-pmax(1-s.t,0)
```

```
payoffeu<-exp(-r*T)*(CC[1:n]+CC[(n+1):(2*n)])/2*K
euprice<-mean(payoffeu)
for(k in (d-1):1){
        z<-rnorm(n)
        mean<-(log(s0)+k*log(s.t[1:n]))/(k+1)
        vol<-(k*dt/(k+1))^0.5*z
        s.t_1<-exp(mean+sigma*vol)
        mean<-(log(s0)+k*log(s.t[(n+1):(2*n)]))/(k+1)
        s.t_1[(n+1):(2*n)]<-exp(mean-sigma*vol)
        CE<-pmax(1-s.t_1,0)
        idx<-(1:(2*n))[CE>0]
        discountedCC<- CC[idx]*exp(-r*dt)
        basis1<-exp(-s.t_1[idx]/2)
        basis2<-basis1*(1-s.t_1[idx])
        basis3<-basis1*(1-2*s.t_1[idx]+(s.t_1[idx]^2)/2)
        p<-glm(discountedCC~basis1+basis2+basis3)$coefficients
        estimatedCC<-p[1]+p[2]*basis1+p[3]*basis2+p[4]*basis3
        EF<-rep(0,2*n)
        EF[idx]<-(CE[idx]>estimatedCC)
        CC<-(EF==0)*CC*exp(-r*dt)+(EF==1)*CE
        s.t<-s.t_1
        }
payoff<-exp(-r*dt)*(CC[1:n]+CC[(n+1):(2*n)])/2
usprice<-mean(payoff*K)
error<-1.96*sd(payoff*K)/sqrt(n)
earlyex<-usprice-euprice
data.frame(usprice,error,euprice,premium)
}
```

A.3. R Codes for the Exact Early Exercise Boundary of an American Put at time t_{d-1} by Newton-Raphson Method

```
newton_raphson<-function(initial.s,K,r,sigma,dt){
# initial ... initial value for the algorithm
# K ... strike Price
# r ... risk-free interest rate
# sigma ... volatility
# dt ... length of time interval

s<-initial.s
repeat{
        d1=(log(s/K)+(r+sigma^2/2)*dt)/(sigma*sqrt(dt));
        d2=d1-sigma*sqrt(dt)
        f<-(K*exp(-r*dt)*pnorm(-d2)-s*pnorm(-d1)-(K-s))
        fderivative<-((s*sigma*sqrt(dt))^(-1))
                *(-K*exp(-r*dt)*dnorm(d2)+s*dnorm(d1))+pnorm(d1)
        new.s<-s-f/fderivative
        conv<-abs(new.s-s)
        if((conv/abs(s))<1e-8) break
        s<-new.s
        }
        s
}
```

A.4. R Codes for the Implementation of the SRA

```
simul_US_SRA<-function(n,npilot,nreg,d,s0,K,sigma,r,T){
s0<-s0/K
dt <- T/d
z<-rnorm(n)
```

```
s.t<-s0*exp((r - 1/2*sigma^2)*T+sigma*rnorm(n)*(T^0.5))
s.t[(n+1):(2*n)]<-s0*exp((r - 1/2*sigma^2)*T-sigma*z*(T^0.5))
CC<-pmax(1-s.t,0)
pthreshold<-NULL
pthreshold[d]<-1
payoffeu<-exp(-r*(d)*dt)*(CC[1:n]+CC[(n+1):(2*n)])/2*K
euprice<-mean(payoffeu)
for(k in (d-1):1){
        z<-rnorm(n)
        mean<-(log(s0)+k*log(s.t[1:n]))/(k+1)
        vol<-(k*dt/(k+1))^0.5*z
        s.t_1<-exp(mean+sigma*vol)
        mean<-(log(s0)+k*log(s.t[(n+1):(2*n)]))/(k+1)
        s.t_1[(n+1):(2*n)]<-exp(mean-sigma*vol)
        CE<-pmax(1-s.t_1,0)
        ###TO HAVE AN IDEA ABOUT THRESHOLD###
        idx<-(1:(2*n))[CE>0]
        idx<-idx[1:npilot]
        discountedCC<- CC[idx]*exp(-r*dt)
        basis1<-exp(-s.t_1[idx]/2)
        basis2<-basis1*(1-s.t_1[idx])
        basis3<-basis1*(1-2*s.t_1[idx]+(s.t_1[idx]^2)/2)
        p<-glm(discountedCC~basis1+basis2+basis3)$coefficients
        estimatedCC<-p[1]+p[2]*basis1+p[3]*basis2+p[4]*basis3
        pilot.EF<-(CE[idx]>estimatedCC)
        pthreshold[k]<-max(s.t_1[idx][pilot.EF==1])
        if(all(pilot.EF==0)){
                pthreshold[k]<-pthreshold[k+1]
                }
        ####REGRESSION AROUND THAT THRESHOLD VALUE###
        distance<-abs(s.t_1-pthreshold[k])
        epsilon<-sort(distance)[nreg]
```

```
          idx<-(1:(2*n))[s.t_1>=(pthreshold[k]-epsilon)&
                          s.t_1<=(pthreshold[k]+epsilon)]
          discountedCC<-CC[idx]*exp(-r*dt)
          p<-lm(discountedCC~s.t_1[idx])$coefficients
          estimatedCC<-p[1]+p[2]*s.t_1[idx]
          EF<-rep(0,2*n)
          EF[s.t_1<(pthreshold[k]-epsilon)]<-1
          EF[idx]<-(CE[idx]>estimatedCC)
          CC<-(EF==0)*CC*exp(-r*dt)+(EF==1)*CE
          s.t<-s.t_1
}
payoffus<-exp(-r*dt)*(CC[1:n]+CC[(n+1):(2*n)])/2*K
usprice <- mean(payoffus)
error <- 1.96*sd(payoffus)/sqrt(n)
data.frame(usprice,error,euprice)
}
```

A.5. R Codes for the Implementation of the Optimization Approach

```
simul_US_OPT<-function(n,d,s0,K,sigma,r,T){
dt <- T/d
z<-rnorm(n)
s.t<-s0*exp((r - 1/2*sigma^2)*T+sigma*z*(T^0.5))
s.t[(n+1):(2*n)]<-s0*exp((r - 1/2*sigma^2)*T-sigma*z*(T^0.5))
CC<-pmax(K-s.t,0)
optimized<-NULL
optimized[d]<-1
for(k in (d-1):1){
          z<-rnorm(n)
          mean<-(log(s0)+k*log(s.t[1:n]))/(k+1)
          vol<-(k*dt/(k+1))^0.5*z*sigma
```

```
        s.t_1<-exp(mean+vol)
        mean<-(log(s0)+k*log(s.t[(n+1):(2*n)]))/(k+1)
        s.t_1[(n+1):(2*n)]<-exp(mean-vol)
        CE<-pmax(K-s.t_1,0)
        o<-order(s.t_1)
        continuation<-CC[o]*exp(-r*dt)
        exercise<-CE[o]
        totalvalue<-cumsum(continuation[(2*n):1])[(2*n):1]
                +cumsum(exercise)-continuation
        optimized[k]<-s.t_1[o][which.max(totalvalue)]
        EF<-rep(0,2*n)
        EF[s.t_1<=optimized[k]]<-1
        CC<-(EF==0)*CC*exp(-r*dt)+(EF==1)*CE
        s.t<-s.t_1
        }
payoff<-exp(-r*dt)*(CC[1:n]+CC[(n+1):(2*n)])/2
data.frame(mean(payoff),sd(payoff)/sqrt(n))
}
```

A.6. R Codes for the Exact Price of a European Call on the Maximum of Two Assets

```
EU_max_of_2<-function(s1,s2,K,sigma1,sigma2,b1,b2,r,T,rho){
# This function requires the package "mnormt"
sigma<-sqrt(sigma1^2+sigma2^2-2*rho*sigma1*sigma2)
d<-(log(s1/s2)+(b1-b2+sigma^2/2)*T)/(sigma*sqrt(T))
y1<-(log(s1/K)+(b1+sigma1^2/2)*T)/(sigma1*sqrt(T))
y2<-(log(s2/K)+(b2+sigma2^2/2)*T)/(sigma2*sqrt(T))
rho1<-(sigma1-rho*sigma2)/sigma
rho2<-(sigma2-rho*sigma1)/sigma
varcov<-matrix(c(1,rep(rho,2),1),nrow=2)
```

```
varcov1<-matrix(c(1,rep(rho1,2),1),nrow=2)
varcov2<-matrix(c(1,rep(rho2,2),1),nrow=2)
s1*exp((b1-r)*T)*pmnorm(c(y1,d),rep(0,2),varcov1)
+s2*exp((b2-r)*T)*pmnorm(c(y2,-d+sigma*sqrt(T)),rep(0,2),varcov2)
-K*exp(-r*T)*(1-pmnorm(c(-y1+sigma1*sqrt(T),
                           -y2+sigma2*sqrt(T)),
                           rep(0,2),varcov))
}
```

A.7. R Codes of the Kirk's approximation for European Spread Options

```
kirk_approximation<-function(S1,S2,Q1=1,Q2=1,K,T,b1,b2,r,
                             sigma1,sigma2,rho,flag="call"){
S<-Q1*S1*exp((b1-r)*T)/(Q2*S2*exp((b2-r)*T)+K*exp(-r*T))
F<-Q2*S2*exp((b2-r)*T)/(Q2*S2*exp((b2-r)*T)+K*exp(-r*T))
sigma<-sqrt(sigma1^2+(sigma2*F)^2-2*rho*sigma1*sigma2*F)
d1<-(log(S)+(sigma^2/2)*T)/(sigma*sqrt(T))
d2<-d1-sigma*sqrt(T)
if(flag=="call"){
    (Q2*S2*exp((b2-r)*T)+K*exp(-r*T))*(S*pnorm(d1)-pnorm(d2))
    }else{
    (Q2*S2*exp((b2-r)*T)+K*exp(-r*T))*(pnorm(-d2)-S*pnorm(-d1))
    }
}
```

A.8. R Codes to Price an American Call on the Maximum of Two Assets without Pilot Study for CV

```
simul_US_max_of_2<-function(n,d,s0,K,sigma1,sigma2,
                    r,dividend,T,rho,c){
dt<-T/d
cormat<-matrix(c(1,rep(rho,2),1),nrow=2)
L<-t(chol(cormat))
z<-matrix(rnorm(n*2),nrow=2)
z<-L%*%z
s1.t<-s0*exp((r-dividend-1/2*sigma1^2)*T+sigma1*(T^0.5)*z[1,])
s1.t[(n+1):(2*n)]<-s0*exp((r-dividend-1/2*sigma1^2)*T
                              -sigma1*(T^0.5)*z[1,])
s2.t<-s0*exp((r-dividend-1/2*sigma2^2)*T+sigma2*(T^0.5)*z[2,])
s2.t[(n+1):(2*n)]<-s0*exp((r-dividend-1/2*sigma2^2)*T
                              -sigma2*(T^0.5)*z[2,])
CC<-pmax(pmax(s1.t, s2.t)-K,0)
payoffeu<-exp(-r*T)*(CC[1:n]+CC[(n+1):(2*n)])/2

for(k in (d-1):1){
        z<-L%*%matrix(rnorm(n*2),nrow=2)
        mean<-(log(s0)+k*log(s1.t[1:n]))/(k+1)
        vol<-(k*dt/(k+1))^0.5*z[1,]*sigma1
        s1.t_1<-exp(mean+vol)
        mean<-(log(s0)+k*log(s1.t[(n+1):(2*n)]))/(k+1)
        s1.t_1[(n+1):(2*n)]<-exp(mean-vol)
        mean<-(log(s0)+k*log(s2.t[1:n]))/(k+1)
        vol<-(k*dt/(k+1))^0.5*z[2,]*sigma1
        s2.t_1<-exp(mean+vol)
        mean<-(log(s0)+k*log(s2.t[(n+1):(2*n)]))/(k+1)
        s2.t_1[(n+1):(2*n)]<-exp(mean-vol)
        CE<- pmax(pmax(s1.t_1, s2.t_1)-K,0)
        idx<-(1:(2*n))[CE>0]
        discountedCC<-CC[idx]*exp(-r*dt)
        basis1<-CE[idx]
```

```
        basis2<-s1.t_1[idx]
        basis3<-s1.t_1[idx]^2
        basis4<-s2.t_1[idx]
        basis5<-s2.t_1[idx]^2
        basis6<-s1.t_1[idx]*s2.t_1[idx]
        p<-glm(discountedCC~basis1+basis2+basis3+basis4
              +basis5+basis6)$coefficients
        estimatedCC<-p[1]+p[2]*basis1+p[3]*basis2+p[4]*basis3
              +p[5]*basis4+p[6]*basis5+p[7]*basis6
        EF<-rep(0,2*n)
        EF[idx]<-(CE[idx]>estimatedCC)
        CC<-(EF==0)*CC*exp(-r*dt)+(EF==1)*CE
        s1.t<-s1.t_1;s2.t<-s2.t_1
        }
payoff<- exp(-r*dt)*(CC[1:n]+CC[(n+1):(2*n)])/2
payoff.c<-payoff-c*(payoffeu-EU_max_of_2(s0,s0,K,sigma1,sigma2,
                                   r-dividend,r-dividend,r,T,rho))
usprice.c<-mean(payoff.c)
s.e.c<-sd(payoff.c)/sqrt(n)
data.frame(usprice.c,s.e.c)
}
```

A.9. R Codes to Price an American Call on the Maximum of Two Assets with Pilot Study for CV

```
simul_US_max_of_2_pilot<-function(n,d,s0,K,sigma1,sigma2,
                    r,dividend,T,rho){
npilot<-1000
dt<-T/d
cormat<-matrix(c(1,rep(rho,2),1),nrow=2)
L<-t(chol(cormat))
```

```
### PILOT STUDY ###
z<-matrix(rnorm(npilot*2),nrow=2)
z<-L%*%z
s1.t<-s0*exp((r-dividend-1/2*sigma1^2)*T+sigma1*(T^0.5)*z[1,])
s1.t[(npilot+1):(2*npilot)]<-s0*exp((r-dividend-1/2*sigma1^2)*T
                               -sigma1*(T^0.5)*z[1,])
s2.t<-s0*exp((r-dividend-1/2*sigma2^2)*T+sigma2*(T^0.5)*z[2,])
s2.t[(npilot+1):(2*npilot)]<-s0*exp((r-dividend-1/2*sigma2^2)*T
                               -sigma2*(T^0.5)*z[2,])
CC<-pmax(pmax(s1.t, s2.t)-K,0)
payoffeu<-exp(-r*T)*(CC[1:npilot]+CC[(npilot+1):(2*npilot)])/2

for(k in (d-1):1){
        z<-L%*%matrix(rnorm(npilot*2),nrow=2)
        mean<-(log(s0)+k*log(s1.t[1:npilot]))/(k+1)
        vol<-(k*dt/(k+1))^0.5*z[1,]*sigma1
        s1.t_1<-exp(mean+vol)
        mean<-(log(s0)+k*log(s1.t[(npilot+1):(2*npilot)]))/(k+1)
        s1.t_1[(npilot+1):(2*npilot)]<-exp(mean-vol)
        mean<-(log(s0)+k*log(s2.t[1:npilot]))/(k+1)
        vol<-(k*dt/(k+1))^0.5*z[2,]*sigma1
        s2.t_1<-exp(mean+vol)
        mean<-(log(s0)+k*log(s2.t[(npilot+1):(2*npilot)]))/(k+1)
        s2.t_1[(npilot+1):(2*npilot)]<-exp(mean-vol)
        CE<- pmax(pmax(s1.t_1, s2.t_1)-K,0)
        idx<-(1:(2*npilot))[CE>0]
        discountedCC<-CC[idx]*exp(-r*dt)
        basis1<-CE[idx]
        basis2<-s1.t_1[idx]
        basis3<-s1.t_1[idx]^2
        basis4<-s2.t_1[idx]
```

```
        basis5<-s2.t_1[idx]^2
        basis6<-s1.t_1[idx]*s2.t_1[idx]
        p<-glm(discountedCC~basis1+basis2+basis3+basis4
                    +basis5+basis6)$coefficients
        estimatedCC<-p[1]+p[2]*basis1+p[3]*basis2++p[4]*basis3
                    +p[5]*basis4+p[6]*basis5+p[7]*basis6
        EF<-rep(0,2*npilot)
        EF[idx]<-(CE[idx]>estimatedCC)
        CC<-(EF==0)*CC*exp(-r*dt)+(EF==1)*CE
        s1.t<-s1.t_1;s2.t<-s2.t_1
        }
payoff<- exp(-r*dt)*(CC[1:npilot]+CC[(npilot+1):(2*npilot)])/2
c<-cov(payoffeu,payoff)/var(payoffeu)

### PRICING WITH CV ###
z<-matrix(rnorm(n*2),nrow=2)
z<-L%*%z
s1.t<-s0*exp((r-dividend-1/2*sigma1^2)*T+sigma1*(T^0.5)*z[1,])
s1.t[(n+1):(2*n)]<-s0*exp((r-dividend-1/2*sigma1^2)*T
                              -sigma1*(T^0.5)*z[1,])
s2.t<-s0*exp((r-dividend-1/2*sigma2^2)*T+sigma2*(T^0.5)*z[2,])
s2.t[(n+1):(2*n)]<-s0*exp((r-dividend-1/2*sigma2^2)*T
                              -sigma2*(T^0.5)*z[2,])
CC<-pmax(pmax(s1.t, s2.t)-K,0)
payoffeu<-exp(-r*T)*(CC[1:n]+CC[(n+1):(2*n)])/2
for(k in (d-1):1){
        z<-L%*%matrix(rnorm(n*2),nrow=2)
        mean<-(log(s0)+k*log(s1.t[1:n]))/(k+1)
        vol<-(k*dt/(k+1))^0.5*z[1,]*sigma1
        s1.t_1<-exp(mean+vol)
        mean<-(log(s0)+k*log(s1.t[(n+1):(2*n)]))/(k+1)
        s1.t_1[(n+1):(2*n)]<-exp(mean-vol)
```

```
        mean<-(log(s0)+k*log(s2.t[1:n]))/(k+1)
        vol<-(k*dt/(k+1))^0.5*z[2,]*sigma1
        s2.t_1<-exp(mean+vol)
        mean<-(log(s0)+k*log(s2.t[(n+1):(2*n)]))/(k+1)
        s2.t_1[(n+1):(2*n)]<-exp(mean-vol)
        CE<- pmax(pmax(s1.t_1, s2.t_1)-K,0)
        idx<-(1:(2*n))[CE>0]
        discountedCC<-CC[idx]*exp(-r*dt)
        basis1<-CE[idx]
        basis2<-s1.t_1[idx]
        basis3<-s1.t_1[idx]^2
        basis4<-s2.t_1[idx]
        basis5<-s2.t_1[idx]^2
        basis6<-s1.t_1[idx]*s2.t_1[idx]
        p<-glm(discountedCC~basis1+basis2+basis3+basis4
                      +basis5+basis6)$coefficients
        estimatedCC<-p[1]+p[2]*basis1+p[3]*basis2+p[4]*basis3
                      +p[5]*basis4+p[6]*basis5+p[7]*basis6
        EF<-rep(0,2*n)
        EF[idx]<-(CE[idx]>estimatedCC)
        CC<-(EF==0)*CC*exp(-r*dt)+(EF==1)*CE
        s1.t<-s1.t_1;s2.t<-s2.t_1
        }
payoff<- exp(-r*dt)*(CC[1:n]+CC[(n+1):(2*n)])/2
payoff.c<-payoff-c*(payoffeu-EU_max_of_2(s0,s0,K,sigma1,sigma2,
                            r-dividend,r-dividend,r,T,rho))
usprice.c<-mean(payoff.c)
s.e.c<-sd(payoff.c)/sqrt(n)
data.frame(usprice.c,s.e.c)
}
```

A.10. R Codes to Price an American Call on the Maximum of Five Assets

```
simul_US_max_of_5<-function(s0,K,n,d,t,sigma,r,dividend,rho){
s0<-s0/K
dt=t/d
cor_matrix<-matrix(0,ncol=5,nrow=5)
diag(cor_matrix)<-1
L<-t(chol(cor_matrix))
z<-matrix(rnorm(n*5),nrow=5)
z<-L%*%z
s1.t<-s0*exp((r-dividend-1/2*sigma^2)*T+sigma*(T^0.5)*z[1,])
s2.t<-s0*exp((r-dividend-1/2*sigma^2)*T+sigma*(T^0.5)*z[2,])
s3.t<-s0*exp((r-dividend-1/2*sigma^2)*T+sigma*(T^0.5)*z[3,])
s4.t<-s0*exp((r-dividend-1/2*sigma^2)*T+sigma*(T^0.5)*z[4,])
s5.t<-s0*exp((r-dividend-1/2*sigma^2)*T+sigma*(T^0.5)*z[5,])
s1.t[(n+1):(2*n)]<-s0*exp((r-dividend-1/2*sigma^2)*T
                                        -sigma*(T^0.5)*z[1,])
s2.t[(n+1):(2*n)]<-s0*exp((r-dividend-1/2*sigma^2)*T
                                        -sigma*(T^0.5)*z[2,])
s3.t[(n+1):(2*n)]<-s0*exp((r-dividend-1/2*sigma^2)*T
                                        -sigma*(T^0.5)*z[3,])
s4.t[(n+1):(2*n)]<-s0*exp((r-dividend-1/2*sigma^2)*T
                                        -sigma*(T^0.5)*z[4,])
s5.t[(n+1):(2*n)]<-s0*exp((r-dividend-1/2*sigma^2)*T
                                        -sigma*(T^0.5)*z[5,])
CC<-pmax(pmax(s1.t,s2.t,s3.t,s4.t,s5.t)-1,0)

for(k in (d-1):1){
z<-L%*%matrix(rnorm(n*5),nrow=5)
s1.t_1<-exp((log(s0)+k*log(s1.t))/(k+1)
                              +sigma*(k*dt/(k+1))^0.5*z[1,])
s2.t_1<-exp((log(s0)+k*log(s2.t))/(k+1)
                              +sigma*(k*dt/(k+1))^0.5*z[2,])
```

```
s3.t_1<-exp((log(s0)+k*log(s3.t))/(k+1)
                              +sigma*(k*dt/(k+1))^0.5*z[3,])
s4.t_1<-exp((log(s0)+k*log(s4.t))/(k+1)
                              +sigma*(k*dt/(k+1))^0.5*z[4,])
s5.t_1<-exp((log(s0)+k*log(s5.t))/(k+1)
                              +sigma*(k*dt/(k+1))^0.5*z[5,])
s1.t_1[(n+1):(2*n)]<-exp((log(s0)+k*log(s1.t[(n+1):(2*n)]))
                         /(k+1)-sigma*(k*dt/(k+1))^0.5*z[1,])
s2.t_1[(n+1):(2*n)]<-exp((log(s0)+k*log(s1.t[(n+1):(2*n)]))
                         /(k+1)-sigma*(k*dt/(k+1))^0.5*z[2,])
s3.t_1[(n+1):(2*n)]<-exp((log(s0)+k*log(s1.t[(n+1):(2*n)]))
                         /(k+1)-sigma*(k*dt/(k+1))^0.5*z[3,])
s4.t_1[(n+1):(2*n)]<-exp((log(s0)+k*log(s1.t[(n+1):(2*n)]))
                         /(k+1)-sigma*(k*dt/(k+1))^0.5*z[4,])
s5.t_1[(n+1):(2*n)]<-exp((log(s0)+k*log(s1.t[(n+1):(2*n)]))
                         /(k+1)-sigma*(k*dt/(k+1))^0.5*z[5,])
st1<-pmax(s1.t_1,s2.t_1,s3.t_1,s4.t_1,s5.t_1)
st2<-pmax(s1.t_1*(s1.t_1<st1),s2.t_1*(s2.t_1<st1),
          s3.t_1*(s3.t_1<st1),s4.t_1*(s4.t_1<st1),
          s5.t_1*(s5.t_1<st1))
st3<-pmax(s1.t_1*(s1.t_1<st2),s2.t_1*(s2.t_1<st2),
          s3.t_1*(s3.t_1<st2),s4.t_1*(s4.t_1<st2),
          s5.t_1*(s5.t_1<st2))
st4<-pmax(s1.t_1*(s1.t_1<st3),s2.t_1*(s2.t_1<st3),
          s3.t_1*(s3.t_1<st3),s4.t_1*(s4.t_1<st3),
          s5.t_1*(s5.t_1<st3))
st5<-pmin(s1.t_1,s2.t_1,s3.t_1,s4.t_1,s5.t_1)
CE<-pmax(st1-1,0)
idx<-(1:(2*n))[CE>0]
discountedCC<-CC[idx]*exp(-r*dt)
basis1<-exp(-st1[idx]/2)
basis2<-basis1*st1[idx]
```

```
basis3<-basis1*(st1[idx]^2-1)
basis4<-basis1*(st1[idx]^3-3*st1[idx])
basis5<-basis1*(st1[idx]^4-6*(st1[idx]^2)+3)
basis6<-st2[idx]
basis7<-st3[idx]
basis8<-st4[idx]
basis9<-st5[idx]
basis10<-basis6^2
basis11<-basis7^2
basis12<-basis8^2
basis13<-basis9^2
basis14<-st1[idx]*basis6
basis15<-basis6*basis7
basis16<-basis7*basis8
basis17<-basis8*basis9
basis18<-st1[idx]*basis6*basis7*basis8*basis9
p<-glm(discountedCC~basis1+basis2+basis3+basis4+basis5+basis6
            +basis7+basis8+basis9+basis10+basis11+basis12
            +basis13+basis14+basis15+basis16+basis17
            +basis18)$coefficients
estimatedCC<-p[1]+p[2]*basis1+p[3]*basis2+p[4]*basis3+p[5]*basis4
            +p[6]*basis5+p[7]*basis6+p[8]*basis7+p[9]*basis8
            +p[10]*basis9+p[11]*basis10+p[12]*basis11
            +p[13]*basis12+p[14]*basis13+p[15]*basis14
            +p[16]*basis15+p[17]*basis16+p[18]*basis17
            +p[19]*basis18
EF<-rep(0,2*n)
EF[idx]=(CE[idx]>estimatedCC)
CC<-(EF==0)*CC*exp(-r*dt)+(EF==1)*CE
s1.t<-s1.t_1;s2.t<-s2.t_1;s3.t<-s3.t_1;s4.t<-s4.t_1;s5.t<-s5.t_1
}
```

```
payoff <- exp(-r*dt)*(CC[1:n]+CC[(n+1):(2*n)])/2*K
usprice<-mean(payoff)
s.e<-sd(payoff)/sqrt(n)
data.frame(usprice,s.e)
}
```

A.11. R Codes to Price a European Spread Call

```
simul_EU_spread_call<- function(n,s1,s2,K,sigma1,sigma2,
                                r,dividend,T,rho){

cormat<-matrix(c(1,rep(rho,2),1),nrow=2)
z<-matrix(rnorm(n*2),nrow=2)
L<-t(chol(cormat))
z<-L%*%z
s1<-s1*exp((r-dividend-1/2*sigma1^2)*T+sigma1*z[1,]*(T^0.5))
s2<-s2*exp((r-dividend-1/2*sigma2^2)*T+sigma2*z[2,]*(T^0.5))

CC<-pmax(s1-s2-K,0)

payoffeu<-exp(-r*T)*CC
euprice<-mean(payoffeu)
s.e<-sd(payoffeu)/sqrt(n)
data.frame(euprice,s.e)
}
```

A.12. R Codes to Price an American Spread Call without Pilot Study for CV

```
simul_US_spread<-function(n,d,s0_1,s0_2,K,sigma1,sigma2,
                          dividend1,dividend2,r,T,rho,c){
dt<-T/d
correlation_matrix<-matrix(c(1,rep(rho,2),1),nrow=2)
L<-t(chol(correlation_matrix))
z<-matrix(rnorm(n*2),nrow=2)
z<-L%*%z
s1.t<-s0_1*exp((r-dividend1-1/2*sigma1^2)*T+sigma1*(T^0.5)*z[1,])
s2.t<-s0_2*exp((r-dividend2-1/2*sigma1^2)*T+sigma2*(T^0.5)*z[2,])
CC<-pmax(s1.t-s2.t-K,0)
payoffeu<-exp(-r*T)*CC
for(k in (d-1):1){
        z<-matrix(rnorm(n*2),nrow=2)
        z<-L%*%z
        mean<-(log(s0_1)+k*log(s1.t))/(k+1)
        vol<-(k*dt/(k+1))^0.5*z[1,]
        s1.t_1<-exp(mean+sigma1*vol)
        mean<-(log(s0_2)+k*log(s2.t))/(k+1)
        vol<-(k*dt/(k+1))^0.5*z[2,]
        s2.t_1<-exp(mean+sigma2*vol)
        CE<-pmax(s1.t_1-s2.t_1-K,0)
        idx<-(1:n)[CE>0]
        discountedCC<-CC[idx]*exp(-r*dt)
        basis1<-s1.t_1[idx]
        basis2<-s2.t_1[idx]
        basis3<-s1.t_1[idx]*s2.t_1[idx]
        basis4<-s1.t_1[idx]^2
        basis5<-s2.t_1[idx]^2
        basis6<-s1.t_1[idx]^2*s2.t_1[idx]
        basis7<-s1.t_1[idx]*s2.t_1[idx]^2
```

```
        basis8<-s1.t_1[idx]^2*s2.t_1[idx]^2
        p<-glm(discountedCC~basis1+basis2+basis3+basis4
                 +basis5+basis6+basis7+basis8)$coefficients
        estimatedCC<-p[1]+p[2]*basis1+p[3]*basis2+p[4]*basis3
                 +p[5]*basis4+p[6]*basis5+p[7]*basis6
                 +p[8]*basis7+p[9]*basis8
        EF<-rep(0,n)
        EF[idx]<-(CE[idx]>estimatedCC)
        CC<-(EF==0)*CC*exp(-r*dt)+(EF==1)*CE
        s1.t<-s1.t_1;s2.t<-s2.t_1
        }
payoff<-CC*exp(-r*dt)
payoff.c<-payoff-c*(payoffeu-kirk_approximation(s0_1,s0_2,
                 Q1=1,Q2=1,K,T,b1=r-dividend1,b2=r-dividend2,
                 r,sigma1,sigma2,rho))
usprice.c <- mean(payoff.c)
se.c<-sd(payoff.c)/sqrt(n)
data.frame(usprice.c,se.c)
}
```

A.13. R Codes to Price an American Spread Call with Pilot Study for CV

```
simul_US_spread_pilot<-function(n,d,s0_1,s0_2,K,sigma1,sigma2,
                            dividend1,dividend2,r,T,rho){
npilot<-1000
dt<-T/d
cormat<-matrix(c(1,rep(rho,2),1),nrow=2)
L<-t(chol(cormat))

### PILOT STUDY ###
z<-matrix(rnorm(npilot*2),nrow=2)
```

```
z<-L%*%z
s1.t<-s0_1*exp((r-dividend1-1/2*sigma1^2)*T+sigma1*(T^0.5)*z[1,])
s2.t<-s0_2*exp((r-dividend2-1/2*sigma1^2)*T+sigma2*(T^0.5)*z[2,])
CC<-pmax(s1.t-s2.t-K,0)
payoffeu<-exp(-r*T)*CC
for(k in (d-1):1){
        z<-matrix(rnorm(npilot*2),nrow=2)
        z<-L%*%z
        mean<-(log(s0_1)+k*log(s1.t))/(k+1)
        vol<-(k*dt/(k+1))^0.5*z[1,]
        s1.t_1<-exp(mean+sigma1*vol)
        mean<-(log(s0_2)+k*log(s2.t))/(k+1)
        vol<-(k*dt/(k+1))^0.5*z[2,]
        s2.t_1<-exp(mean+sigma2*vol)
        CE<-pmax(s1.t_1-s2.t_1-K,0)
        idx<-(1:npilot)[CE>0]
        discountedCC<-CC[idx]*exp(-r*dt)
        basis1<-s1.t_1[idx]
        basis2<-s2.t_1[idx]
        basis3<-s1.t_1[idx]*s2.t_1[idx]
        basis4<-s1.t_1[idx]^2
        basis5<-s2.t_1[idx]^2
        basis6<-s1.t_1[idx]^2*s2.t_1[idx]
        basis7<-s1.t_1[idx]*s2.t_1[idx]^2
        p<-glm(discountedCC~basis1+basis2+basis3+basis4
                +basis5+basis6+basis7)$coefficients
        estimatedCC<-p[1]+p[2]*basis1+p[3]*basis2+p[4]*basis3
                +p[5]*basis4+p[6]*basis5+p[7]*basis6+p[8]*basis7
        EF<-rep(0,npilot)
        EF[idx]<-(CE[idx]>estimatedCC)
        CC<-(EF==0)*CC*exp(-r*dt)+(EF==1)*CE
        s1.t<-s1.t_1;s2.t<-s2.t_1
```

```
        }
payoff<-CC*exp(-r*dt)
c<-cov(payoffeu, payoff)/var(payoffeu)

### PRICING WITH CV ###
z<-matrix(rnorm(n*2),nrow=2)
z<-L%*%z
s1.t<-s0_1*exp((r-dividend1-1/2*sigma1^2)*T+sigma1*(T^0.5)*z[1,])
s2.t<-s0_2*exp((r-dividend2-1/2*sigma1^2)*T+sigma2*(T^0.5)*z[2,])
CC<-pmax(s1.t-s2.t-K,0)
payoffeu<-exp(-r*T)*CC
for(k in (d-1):1){
        z<-matrix(rnorm(n*2),nrow=2)
        z<-L%*%z
        mean<-(log(s0_1)+k*log(s1.t))/(k+1)
        vol<-(k*dt/(k+1))^0.5*z[1,]
        s1.t_1<-exp(mean+sigma1*vol)
        mean<-(log(s0_2)+k*log(s2.t))/(k+1)
        vol<-(k*dt/(k+1))^0.5*z[2,]
        s2.t_1<-exp(mean+sigma2*vol)
        CE<-pmax(s1.t_1-s2.t_1-K,0)
        idx<-(1:n)[CE>0]
        discountedCC<-CC[idx]*exp(-r*dt)
        basis1<-s1.t_1[idx]
        basis2<-s2.t_1[idx]
        basis3<-s1.t_1[idx]*s2.t_1[idx]
        basis4<-s1.t_1[idx]^2
        basis5<-s2.t_1[idx]^2
        basis6<-s1.t_1[idx]^2*s2.t_1[idx]
        basis7<-s1.t_1[idx]*s2.t_1[idx]^2
        p<-glm(discountedCC~basis1+basis2+basis3+basis4
                +basis5+basis6+basis7)$coefficients
```

```
        estimatedCC<-p[1]+p[2]*basis1+p[3]*basis2+p[4]*basis3
                +p[5]*basis4+p[6]*basis5+p[7]*basis6+p[8]*basis7
        EF<-rep(0,n)
        EF[idx]<-(CE[idx]>estimatedCC)
        CC<-(EF==0)*CC*exp(-r*dt)+(EF==1)*CE
        s1.t<-s1.t_1;s2.t<-s2.t_1
        }
payoff<-CC*exp(-r*dt)
payoff.c<-payoff-c*(payoffeu-kirk_approximation(s0_1,s0_2,
                Q1=1,Q2=1,K,T,b1=r-dividend1,b2=r-dividend2,
                r,sigma1,sigma2,rho))
usprice.c <- mean(payoff.c)
se.c<-sd(payoff.c)/sqrt(n)
data.frame(usprice.c,se.c)
}
```

REFERENCES

Acton, F. S., 1990, *Numerical Methods That Work*, Cambridge University Press, Washington.

AitSahlia, F. and P. Carr, 1997, "American Options: A Comparison of Numerical Methods", *Numerical Methods in Finance*, eds. L. C. G. Rogers and D. Talay. Cambridge, Cambridge University Press.

Andersen, L. and M. Broadie, 2004, "Primal-Dual Simulation Algorithm for Pricing Multidimensional American Options", *Management Science*, Vol. 50, No. 9, pp. 1222-1234.

Barone-Adesi, G. and R. E. Whaley, 1987, "Efficient analytic approximation of American option values", *Journal of Finance*, Vol. 42, No. 2, pp. 301-320.

Barraquand, J. and D. Martineau, 1995, "Numerical Valuation of High Dimensional Multivariate American Securities ", *The Journal of Financial and Quantitative Analysis*, Vol. 30, No. 3, pp. 383-405.

Black, F. and M. Scholes, 1973, "The Pricing of Options and Corporate Liabilities", *Journal of Political Economy*, Vol. 81, pp. 637-654.

Bolia, N., S. Juneja and P. Glasserman, 2004, "Function-Approximation Based Importance Sampling for Pricing American Options", *Proceeding of the 2004 Winter Simulation Conference*, Washington, DC, pp. 604-611.

Bossaerts, P., 1989, "Simulation Estimators of Optimal Early Exercise", working paper, Carnegie-Mellon University.

Boyle, P. P., J. Evine and S. Gibbs, 1989, "Numerical Evaluation of Multivariate Contingent Claims", *Review of Financial Studies*, Vol. 2, No. 2, pp. 241-250.

Boyle, P. P., M. Broadie and P. Glasserman, 1997, "Monte Carlo methods for security pricing", *Journal of Economic Dynamics and Control*, Vol. 21, pp. 1267-1321.

Brennan, M. J. and E. S. Schwartz, 1977, "Convertible Bonds: Valuation and Optimal Strategies for Call and Conversion", *Journal of Finance*, Vol. 32, pp. 1699-1715.

Broadie, M. and J. Detemple, 1996, "American Option Valuation: New Bounds, Approximations, and a Comparison of Existing Methods", *Review of Financial Studies*, Vol. 9, No. 4, pp. 1211-1250.

Broadie, M. and P. Glasserman, 1997, "A Stochastic Mesh Method for Pricing High-Dimensional American Options", *Working paper*, Columbia University.

Broadie, M. and P. Glasserman, 2004, "A Stochastic Mesh Method for Pricing High-Dimensional American Options", *Journal of Computational Finance*, Vol.7, No. 4, pp. 35-72.

Broadie, M. and Y. Yamamoto, 2005, "A Double-Exponential Fast Gauss Transform Algorithm for Pricing Discrete Path-dependent Options", *Institute for Operations Research and the Management Sciences*, Vol. 53, No. 5, pp. 764-779.

Brodie, M., P. Glasserman and Z. Ha, 2000, "Pricing American Options by Simulation using a Stochastic Mesh with Optimized Weights", *Probabilistic Constrained Optimization: Methodology and Applications*, S. Uryasev, ed., Kluwer Publishers, Norwell, Mass, pp. 32-50.

Carriere, J. F., 1996, "Valuation of the Early-Exercise price for options using simulations and nonparametric regression", *Insurance: Mathematics and Economics*, Vol. 19, pp. 19-30.

Charnes, J. M., 2000, "Using Simulation for Option Pricing", *Proceedings Proceeding of the 2000 Winter Simulation Conference*, Orlando, FL, pp. 151-157.

Chicago Board Options Exchange (CBOE), 2006, "2006 Market Statistics", http://www.cboe.com/data/marketstats-2006.pdf

Clary, I., 2007, *Wall Street Spreading the Word on Options*, http://coweb.sv.publicus.com/apps/pbcs.dll/article?AID=/20070219/PRINTSUB/702190711

Clément, E., D. Lamberton and P. Protter, 2001, "An analysis of a least squares regression algorithm for American option pricing", *Finance and Stochastics*, Vol. 6, No. 4, pp. 449-471.

Cox, J. and S. Ross, 1976, "The valuation of options for alternative stochastic processes", *Journal of Financial Economics*, Vol. 3, No. 1-2, pp. 145-166.

Cox, J., S. Ross and M. Rubinstein, 1979, "Option pricing: A simplified approach", *Journal of Financial Economics*, Vol. 7, No. 3, pp. 229-263.

Das, S., 2004, *Swaps/Financial Derivatives : Products, Pricing, Applications and Risk Management*, John Wiley & Sons, Singapore.

Detemple, J., S. Feng and W. Tian, 2003, "The Valuation of American Call Options on the Minimum of two dividend paying assets", *The Annals of Applied Probability*, Vol. 13, No. 3, pp. 953-983.

Ehrlichman, S. M. T. and S. G. Henderson, 2006, "American options from MARS", *Proceeding of the 2006 Winter Simulation Conference.*, Monterey, CA, pp. 719-726.

Fu, M. C. and J. Q. Hu, 1995, "Sensitivity Analysis for Monte Carlo Simulation of Option Pricing", *Probability in the Engineering and Information Sciences.*, Vol. 9, No. 3, pp. 417-446.

Geske, R., 1979, "A note on an analytic valuation formula for unprotected American call options on stocks with known dividends", *Journal of Financial Economics*, Vol. 7, pp. 375-380.

Geske, R. and H. E. Johnson, 1984, "The American Put Option Valued Analytically", *Journal of Finance*, Vol. 39, No. 5, pp. 1511-1524.

Glasserman, P., 2003, *Monte Carlo Methods in Financial Engineering*, Cambridge University Press, Cambridge.

Haug, E. G., 2007, *The Complete Guide to Option Pricing Formulas*, McGraw Hill, New York.

Haugh, M. B. and L. Kogan, 2004, "Pricing American options: A duality approach", *Operations Research*, Vol. 52, No. 2, pp. 258-270.

Hull, J., 2003, *Options, Futures and Other Derivative Securities*, 5th ed., Prentice Hall, New Jersey.

Investopedia, 2008, "Spread Option", http://www.investopedia.com/terms/s/spreadoption.asp

Kirk, E., 1995, "Correlation in the Energy Markets", in *Managing Energy Price Risk*, Risk Publications and Enron, pp. 71-78.

Laprise, S. B., M. C. Fu, S. I. Marcus, A. E. B. Lim and H. Zhang, 2006, "Pricing American-Style Derivatives with European Call Options", *Management Science*, Vol. 52, No. 1, pp. 95-110.

Lemieux C. and J. La, 2005, "A Study of Variance Reduction Techniques for American Option Pricing", *Proceeding of the 2005 Winter Simulation Conference*, Orlando, FL, pp. 1884-1991.

Longstaff, F. A. and E. S. Schwartz, 2001, "Valuing American Options by Simulation: A Simple Least-Squares Approach", *Review of Financial Studies*, Vol. 14, No. 1, pp. 113-147.

Lyuu, Y. D., 2002, *Financial Engineering and Computation: Principles, Mathematics, Algorithms*, Cambridge University Press, Corrigenda.

Options Industry Council (OIC), 2008, “Options Basics”, http://www.optionseducation.org/classes/syllabus_ options_ basics.jsp

Rogers, L. C. G., 2002, “Monte Carlo valuation of American options”, *Mathematical Finance*, Vol. 12, No. 3, pp. 271-286.

Tavella, D., 2002, *Quantitative Methods in Derivatives Pricing: An Introduction to Computational Finance*, John Wiley & Sons, New Jersey.

Tilley J. A., 1993, “valuing american options in a simulation method ”, *Transactions of the Society of Actuaries*, Vol. 45, pp. 83-104.

Tsitsiklis, J. N. and B. Van Roy, 2001, “Regression Methods for Pricing Complex American-Style Options ”, *IEEE Transactions on Neural Networks*, Vol. 12, No. 4, pp. 694-703.

Turkish Derivative Exchange (TurkDEX), 2008, “TurkDEX - Introduction”, http://www.turkdex.org.tr/VOBPortalEng/DesktopDefault.aspx?tabid=100

Wilmott, P., 2006, *Paul Wilmott on Quantitative Finance*, 2nd ed., John Wiley & Sons, New Jersey.