Spring 5-15-2020

# American Option Pricing: From PDE Numerical Solutions to Simulation-Based Methods and Reinforcement Learning.

Chenshan Hu
*Washington University in St. Louis*

WASHINGTON UNIVERSITY IN ST. LOUIS

Department of Mathematics

# American Option Pricing: From PDE Numerical Solutions to Simulation-Based Methods and Reinforcement Learning.

by

Chenshan Hu

A thesis presented to
The Graduate School
of Washington University in
partial fulfillment of the
requirements for the degree
of Master of Arts

May 2020

St. Louis, Missouri

# Contents

# Acknowledgements

Hereby, I would like to express my sincere gratitude to Professor José E. Figueroa-López for his continued support, guidance and feedback throughout this research study.

I am also particulary grateful for the invaluable assistance given by the faculty in Department of Mathematics, who have helped me a lot with my master courses in the past two years.

Furthermore, I would like to thank my family members and friends for their unceasing love, care and support.

Last but not least, all kinds of feedback, comments and suggestions for improvement are warmly welcomed.

# Abstract

An American call (put) option is a contract that gives the holder the right, but not the obligation, to buy (sell) one unit of an asset (typically, stock) at a prespecified price (called strike price) at any desired time before a preset expiration time of the contract. The associated option pricing problem plays an important role in modern financial markets and one way to solve this is by searching for the optimal exercise policy, i.e., find the optimal time to exercise so that maximal reward is achieved. In this thesis, we shall discuss the modern Least Square Policy Iteration Method to solve the American option pricing problem based on Reinforcement Learning and compare it to the method of the Longstaff-Schwartz Method and the Finite Difference Method.

# 1 Introduction.

In this section, we will go through the background and theorems first.

## 1.1 Brownian Motion.

The following introduces one of the most fundamental building blocks in modern mathematical finance: Brownian Motion. We will subsequently borrow ideas and theorems in this subsection to simulate stock price paths, which is needed later on in this thesis.

**Definition 1.1.** A continuous-time process $\{B_t\}_{t\geq 0}$ is a standard Brownian motion (BM) if it satisfies the following properties:

(1) $B_0 = 0$, i.e. starts at 0.

(2) $B_{t+s} - B_t \sim N(0, s)$, for any $t \geq 0$, $s > 0$, i.e. stationary increments.

(3) $B_{t_1} - B_{t_0}, B_{t_2} - B_{t_1}, ..., B_{t_n} - B_{t_{n-1}}$ are independent for any $0 \leq t_0 < t_1 < ... < t_n$, i.e. independent increments.

(4) $t \to B_t$ is a continuous function of t, i.e. continuous paths.

**Definition.** From "Arbitrage Theory in Continuous Time' written by Björk: let $t_0 = 0 < t_1 < t_2...$ be such that $t_n \to \infty$ and let $\{f_t\}_{t\geq 0}$ be an adapted process. When the Riemann-Stieltjes sum $I_B^n(f)_t = \sum_{i=0}^{\infty} f_{t_i}(B_{t_{i+1}\wedge t} - B_{t_i\wedge t})$ converges in probability to a unique random variable **I** as the partition mesh $\max\{t_i - t_{i-1}\} \to 0$ and is independent of the partition $t_0 = 0 < t_1 < t_2...t_n \to \infty$, this limiting value **I**

is called the Stochastic or Itô integral of f with respect to the Brownian Motion, B, and is denoted by any of the following notations:

$$I_B(f)_t = B(f)_t = \int_0^t f_u dB_u.$$

*Remark.* An equation of the form: $X_t = x + \int_0^t b(u, X_u)du + \int_0^t f(u, X_u)dB_u$ (or, in the corresponding differential form, $dX_t = b(t, X_t)dt + f(t, X_t)dB_t$ ) is called a Stochastic Differential Equation (SDE). Besides, if a process $\{X_t\}_{t\geq 0}$ satisfies this equation almost surely, then we say X is a (strong) solution of the SDE.

**Definition.** A stochastic process $\{S_t\}_{t\geq 0}$ is a Geometric Brownian motion (GBM) if $S_t$ follows the following Stochastic Differential Equation:

$$dS_t = \mu S_t dt + \sigma S_t dB_t = S_t(\mu dt + \sigma dB_t),$$

where $\mu$ and $\sigma$ are constants, called the drift and volatility of $S_t$, and $B_t$ is a standard Brownian motion.

Next we are going to introduce a key tool, Itô's Lemma, for working with SDE.

**Lemma 1.1.** *Itô's Lemma: Let $\{X_t\}_{t\geq 0}$ be a BM satisfying the following SDE:*

$$dX_t = \mu_t dt + \sigma_t dB_t,$$

*where $\mu_t$ and $\sigma_t$ are adapted processes. Let f(t,x) be a real-valued function whose*

*second-order partial derivatives are continuous. Then* $Y_t = f(t, X_t)$ *admits the following representation:*

$$dY_t = (\frac{\partial f}{\partial t} + \mu\frac{\partial f}{\partial x} + \frac{\sigma^2}{2}\frac{\partial^2 f}{\partial x^2})dt + \sigma\frac{\partial f}{\partial x}dB_t.$$

**Example 1.1.** Now we are able to derive a formula for the simulated stock price.

Let $\{S_t\}_{t\geq 0}$ be a geometric Brownian motion. Set $f(t, s) = log(s)$.

Note that the function $f$ has no explicite dependence on $t$. After applying the Itô's lemma for GBM, we shall have

$$d\log(S_t) = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial S}dS_t + \frac{\sigma^2}{2}\frac{\partial^2 f}{\partial S^2}S^2dt. \tag{1.1}$$

Then, as $\frac{\partial f}{\partial t} = 0, \frac{\partial f}{\partial S} = \frac{1}{S}, \frac{\partial^2 f}{\partial S^2} = -\frac{1}{S^2}$,

$$\begin{aligned} d\log(S) &= \frac{1}{S}(\mu S dt + \sigma S dB_t) - \frac{1}{2}\sigma^2 dt \\ &= \sigma dB_t + (\mu - \frac{\sigma^2}{2})dt. \end{aligned}$$

Integrate both side with limits t=0 to t=T,

$$\begin{aligned} \log S_T - \log S_0 &= \sigma B_T - \sigma B_0 + (\mu - \frac{\sigma^2}{2})(T - 0) \\ S_T &= S_0 \times \exp\{\sigma B_T + (\mu - \frac{\sigma^2}{2})T\}. \end{aligned} \tag{1.2}$$

## 1.2 Least Squares Regression.

The key to the Longstaff-Schartz approach is the use of Least Squares Method to estimate the conditional expectation of the continuation value and thus we will introduce the Least Squares Regression technique in this section.

Given a data set with response variable $\mathbf{Y}$ and variables $\mathbf{X}$, suppose $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, where $\mathbf{X}$ is a fixed $n \times p$ matrix, $\boldsymbol{\beta} = (\beta_1, ...\beta_n)'$ is a $p \times 1$ vector and $\boldsymbol{\epsilon} = (\epsilon_1, ...\epsilon_n)'$ is a $n \times 1$ random vector. Assume that $\epsilon_i$ are i.i.d. with mean 0 and variance $\sigma^2$.

We want to find an estimate of the coefficient vector $\boldsymbol{\beta}$. One canonical approach is to use the Least Squares estimators $\hat{\boldsymbol{\beta}}$, which is defined as the value of $\tilde{\beta}$ that minimizes

$$\boldsymbol{S}(\tilde{\boldsymbol{\beta}}) = \sum_{i=1}^{n} e_i^2 = \boldsymbol{e'e} = (\mathbf{y} - \mathbf{X}\tilde{\boldsymbol{\beta}})'(\mathbf{y} - \mathbf{X}\tilde{\boldsymbol{\beta}}) \quad \text{over all } \tilde{\beta} \in \mathbb{R}^p.$$

Hence, the above equation becomes:

$$\boldsymbol{S}(\beta) = \boldsymbol{y'y} - \beta'\boldsymbol{X'y} - \boldsymbol{y'X}\beta + \beta'\boldsymbol{X'X}\beta = \boldsymbol{y'y} - \mathbf{2}\beta'\boldsymbol{X'y} + \beta'\boldsymbol{X'X}\beta.$$

Then, by applying matrix differentiation:

$$\frac{\partial \boldsymbol{S}}{\partial \boldsymbol{\beta}}\Big|_{\hat{\beta}} = -2(\boldsymbol{X'y} - \boldsymbol{X'X}\hat{\boldsymbol{\beta}}) = \mathbf{0}.$$

Finally, we obtain the so-called normal equations:

$$\boldsymbol{X'y} = \boldsymbol{X'X}\hat{\boldsymbol{\beta}}.$$

*Remark.* When the inverse matrix of $\mathbf{X'X}$ exists, then $\hat{\boldsymbol{\beta}} = (\boldsymbol{X'X})^{-1}\boldsymbol{X'y}$.

## 1.3 Monte Carlo Method.

Monte Carlo Method is used in both Longstaff Schwartz Method and Least Square Policy Iteration to obtain an approximated optimal policy.

**Definition 1.2.** Monte Carlo Method (MC) are handy computational techniques for repeatedly sampling a probability distribution to solve deterministic problems.

Consider the problem of evaluating the integral $E_f[h(X)] \triangleq \int h(x)f(x)dx$, where $f(x)$ is the density of $X$. The MC Solution generates N random replicas $X_1, X_2, ..., X_N$ of $X$ and uses these values to approximate the above integral as:

$$E_f[h(X)] \approx \overline{h}_N = \frac{\sum_{i=1}^{N} h(X_i)}{N}.$$

Besides, if $E_f[h^2(X)] < \infty$, we can assess the MC approximation error by evaluating:

$$\text{var}(\overline{h}_N) = \frac{\int (h(x) - E_f[h(X)])^2 f(x)dx}{N},$$

$$\rightarrow \text{var}(\overline{h}_N) \approx v_N = \frac{\sum_{i=1}^{N} [h(X_i) - \overline{h}_N]^2}{N^2}.$$

By the Central Limit Theorem, as $N \rightarrow \infty$, we have

$$\frac{\overline{h}_N - E_f[h(X)]}{\sqrt{v_N}} \xrightarrow{d} N(0,1).$$

## 1.4 Reinforcement Learning.

We will discuss the background & main theorems of Reinforcement Learning in this subsection so that we can better understand the Least Square Policy Iteration.

**Definition 1.3.** Reinforcement Learning (RL): RL is to learn what to do - how to map situations to actions - so as to maximize a numerical reward.

Generally, RL problems can be formalized as the optimal control of an unknown Markov Decision Processes.

**Introduction to Markovian Decision Process (MDP):**

Roughly, a Markovian system is one where all information about past states is carried by the current state of the system.

Under the MDP setting, we have a sequence of $S_0, A_0, R_1, S_1, A_1, R_2, ...$, where $S_i$ denotes the state at time $t_i$, $A_i$ denotes the action taken at time $t_i$, and $R_i$ denotes the rewards received at time $t_i$.

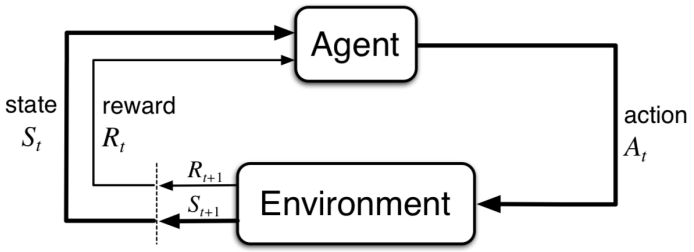



Figure 1: Illustration of the 5-tuple of RL
We can see from the plot, the agent would take action according to the changes and feedbacks from the environment. Actually, the challenging part of RL lies in the fact that the current action would influence both immediate and subsequent states for sequential decision-making problems.

We would link a Markovian structure to some function $p(s', r|s, a)$ :

$$\begin{aligned} p(s', r|s, a) &= Pr\{S_t = s', R_t = r|S_{t-1} = s, A_{t-1} = a\} \\ &= Pr\{S_t = s', R_t = r|S_{t-1} = s, A_{t-1} = a, ..., S_0 = s_0, A_0 = a_0\}. \end{aligned}$$

The function $p(s', r|s, a)$ is called the transition probability of the MDP.

**Definition.** A (stochastic) policy $\pi$ is defined to be a mapping from states to probabilities of selecting each possible action.

**Definition 1.4.** In a typical MDP problem, we would like to find an optimal policy to maximize the expected discounted reward. Thus, define the action-value function for policy $\pi$ by $Q_\pi(s, a) = E_\pi[G_t|S_t = s, A_t = a]$ , where $\gamma \in (0, 1)$ is the discounting factor and $G_t = \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$ is the discounted return.

**Definition.** A policy $\pi$ is defined to be better than or equal to another policy $\pi'$ if its expected return is greater than or equal to that of $\pi'$ for all states. Furthermore, the optimal policy is the policy that is better than or equal to any other policies. We shall denote all the optimal policies by $\pi_*$ and they share same optimal action value function, defined as $Q_*(s,a) = \max_\pi Q_\pi(s,a)$.

**Proposition 1.1.** *By applying properties of conditional expectation to the definition of $Q_\pi$, we shall have*

$$\begin{aligned} Q_\pi(s,a) &= E_\pi[G_t|S_t = s, A_t = a] \\ &= E_\pi[R_{t+1} + \gamma G_{t+1}|S_t = s, A_t = a] \\ &= \sum_{s',r} p(s',r|s,a)[r + \gamma \sum_{a'} \pi(a'|s')Q_\pi(s',a')]. \end{aligned} \tag{1.3}$$

*Equation (1.3) is called the Bellman Equation.*

*Remark.* $\pi(a|s)$ is the probability of taking action $a$ in a given state $s$ under policy $\pi$. When $\pi(a|s) = 1$ for some $a \in A$, policy $\pi$ is said to be 'deterministic'.

### 1.4.1 Relationship between MDP and Markov Chains.

**Definition 1.5.** A discrete-time stochastic process $\{X_n\}_{n\in N=\{0,1,...\}}$, with each $X_n$ taking values in the finite set $S = \{1, ...N\}$, is called a (time-homogeneous) Markov

Chain if, for any time n and states $i_0, ..., i_{n-1}, i, j$,

$$\mathbb{P}\{X_{n+1} = j | X_n = i, X_{n-1} = i_{n-1}, ..., X_0 = i_0\} = \mathbb{P}\{X_{n+1} = j | X_n = i\} \tag{1.4}$$

$$= P\{X_1 = j | X_0 = i\}. \tag{1.5}$$

*Remark.* The Equality in (1.4) is called the markov property while the equality in (1.5) is called the time-homogeneity property.

The Markovian property states that the future and past are independent given the present:

Future event $B = \{X_{n+1} = i_{n+1}, X_{n+2} = i_{n+2}, ..., X_{n+r} = i_{n+r}\}$.

Past event $A = \{X_0 = i_0, ..., X_{n-1} = i_{n-1}\}$.

Present event $C = \{X_n = i_n\}$.

$\rightarrow \mathbb{P}\{A, B|C\} = \mathbb{P}\{A|C\}P\{B|C\}$.

*Remark* 1.1. MDPs are an extension of Markov Chains with addition of actions and rewards. Also, if only one action exists for each state and all rewards are the same, then the MDP would be reduced to a Markov Chain.

## 1.5 Option Pricing.

**Definition.** An American call (put) option is a contract that gives the holder the right, but not the obligation, to buy (sell) one unit of an asset (typically, stock) at a prespecified price (called strike price) at any desired time before a preset expiration time of the contract.

Determining the option pricing is a very popular yet challenging topic in modern mathematical finance. One possible approach is through the search for the Optimal Stopping Policy, i.e optimal choice of the moment to exercise the option, to maximise the expected return. Below are some useful concepts in the Optimal Stopping Problem.

- Let $S_t$ denote the stock price at time t. In the Black Scholes model, $S_t$ is assumed to be a geometric Browinian motion. Then, from Equation (1.2), we shall have:

  $$S_T = S_0 \times \exp\{\sigma B_T + (r - \frac{\sigma^2}{2})T\} \quad \text{where } \mu = r, \text{ i.e. the risk-free interest rate.}$$

*Remark.* For Fig 2, I have simulated three stock price paths for each figures. For all, initial price $S_0 = 100$, and time taken $T = 1$. Besides, the index of time steps run from 0 to 1, i.e. $\delta t = \frac{1}{1000}$. The figure in the left displays the evolution of

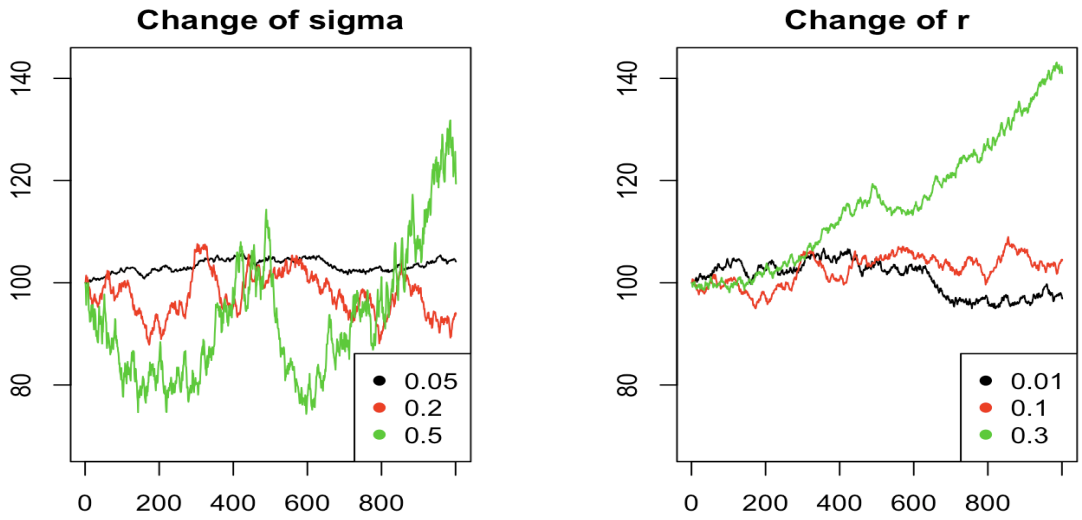



Figure 2: Simulations for stock price paths

the stock price with three different values of $\sigma$, i.e. 0.05, 0.2, 0.5, when r = 0.06 . The figure in the right displays the evolution of stock price with three values of $r$, i.e. 0.01, 0.1, 0.3, when $\sigma = 0.1$.

- Stopping time $\tau$: a 'stopping policy' to decide whether to continue or to stop given the current & past information. Information at time $t$ is represented in terms of a $\sigma - field$ $\{\mathscr{F}_t\}$. So, we want $\tau$ to be such that $\{\tau \leq t\} \in \mathscr{F}_t$ for all $t$; in this way, deciding whether $\tau \leq t$ only depends on infomation up to $t$.

- Hitting time: an example of stopping time, i.e. the first time $X_t$ takes a value within the Borel set A for a process $X_t$,

$$T_{X,A} = min\{t \in R | X_t \in A\}.$$

- The price of an American call option would be:

$$V_t = \sup_{\tau} E[(S_\tau - X)_+ | \mathscr{F}_t], \text{ where X is the strike price.}$$

# 2 Longstaff Schwartz Algorithm.

Longstaff Schwartz Algorithm, also known as the Least Square Monte Carlo (LSM) approach, was first proposed by Francis A. Longstaff and Eduardo S. Schwartz in 2001 to compute the American option prices by solving the Optimal Stopping Problem. LSM is widely-used especially in high dimensions where classic PDE methods are usually futile.

Longstaff Schwartz Algorithm has two aims:

1. Approximate the value of American option through simulation by using least squares regression to estimate the continued conditional expected payoff.

2. Determine the optimal stopping, or exercise strategy from the conditional expectation of continuation value, i.e. value of the option if we decide to continue rather than to exercise it at the current time. Hence LSM approach can be helpful to find the optimal strategy, i.e. maximize the payoff of the option for each simulated path.

Longstaff Schwartz Algorithm has the following assumptions:

1. Option can only be exercised at discrete times $t_i's$ with $0 < t_1 \leq t_2 \leq t_3 \leq ..... \leq t_K = T$. Select K to be sufficiently large so that continuous exercisablity is roughly achieved.

2. No Arbitrage valuation theory.

3. A linear combination of countable set of measurable basis functions, e.g. (weighted) Laguerre polynomial shall be used to approximate the conditional expectation of continuation value.

$$L_0(X) = \exp(-X/2),$$
$$L_1(X) = \exp(-X/2)(1-X),$$
$$L_2(X) = \exp(-X/2)(1-2X+\frac{X^2}{2}),$$
$$L_n(X) = \exp(-X/2)\frac{e^x}{n!}\frac{d^n}{dX^n}(X^n e^{-X}).$$

4. Given a complete probability space $(\Omega, F, P)$ and finite time inverval [0,T], where $\Omega$ is the set of all possible realizations between 0 and T and element $\omega$ denotes a sample path, then the Real Value of Continuation is defined as:

$$CV(\omega, t_k) = E_Q[\sum_{j=k+1}^{K} \exp(-\int_{t_k}^{t_j} r(\omega, s)ds)C(\omega, t_j; t_k, T)|F_{t_k}], \tag{2.1}$$

where $r(\omega, t)$ is the riskless discount rate, Q is the equivalent martingale measure for the economy and $C(\omega, s; t, T)$ is the path of cash flows.

Besides, $CV(\omega, t_{K-1}) = \sum_{j=0}^{\infty} a_j L_j(X)$.

## 2.1 Procedure.

**Steps**: [Backward, start from $t_{K-1}$ and all the way down to $t_1$]

1. Approximate $CV(\omega, t_{K-1})$ by $CV_M(\omega, t_{K-1})$. More specifically, we shall regress the corresponding discounting value of max(strike price - stock price at $t_K$,0) against the first M basis functions of the stock prices for in-the-money paths at $t_{K-1}$. Then, calculate the approximated continuation value ($C\hat{V}_M(\omega, t_{K-1})$) by matrix multiplica-

tion of coefficients from the model and the first M basis functions of the stock prices for in-the-money paths at $t_{K-1}$.

Note:

(i) Only in-the-money paths are used to save computation time.

(ii) The fitted line estimate, i.e. $C\hat{V}_M(\omega, t_{K-1})$ is a best linear unbiased estimator of $CV_M(\omega, t_{K-1})$.]

(iii) 5 basis functions would be a good benchmark in practice.

2. Decide to exercise if:

i) the immediate exercise value is positive.

ii) immediate exercise value$\geq C\hat{V}_M(\omega, t_{K-1})$ .

3. Continue this process for $t_{K-2}$.

## 2.2 A Simple Numerical Example.

Steps:

1. Simulate stock price paths. Consider an American put option on a share of stock with no divident. Assume that it can only be exercised at times 1 and 2 with strike price 10.5. Besides, risk free interest rate $r = 0.06$, and volatility $\sigma = 0.1$.

| Path | t=0 | t=1 | t=2 |
|---|---|---|---|
| 1 | 10 | 9.22 | 8.79 |
| 2 | 10 | 9.98 | 10.48 |
| 3 | 10 | 9.35 | 10.61 |
| 4 | 10 | 10.58 | 11.66 |
| 5 | 10 | 9.93 | 10.11 |
| 6 | 10 | 10.28 | 9.50 |
| 7 | 10 | 9.65 | 9.26 |
| 8 | 10 | 11.95 | 11.55 |

Next, apply the LSM backward algorithm to this set of paths.

2. Compute cash flow at time t = 2. The cashflow matrix below is calculated by $max(\text{stock price at time } 2 - \text{strike price}, 0)$.

| Path | t=0 | t=1 | t=2 |
|---|---|---|---|
| 1 | - | - | 1.71 |
| 2 | - | - | 0.02 |
| 3 | - | - | 0 |
| 4 | - | - | 0 |
| 5 | - | - | 0.39 |
| 6 | - | - | 1.00 |
| 7 | - | - | 1.24 |
| 8 | - | - | 0 |

3. Regress at t = 1. Here, we would regress Y on $X^0, X^1, X^2$ and the obtained conditional expectation function is $E[Y|X] = 217.369 - 44.192X + 2.25X^2$.

| Path | Y | X |
|---|---|---|
| 1 | $1.71 \times 0.9417$ | 9.22 |
| 2 | $0.02 \times 0.9417$ | 9.98 |
| 3 | 0 | 9.35 |
| 4 | - | - |
| 5 | $0.39 \times 0.9417$ | 9.93 |
| 6 | $1.00 \times 0.9417$ | 10.28 |
| 7 | $1.24 \times 0.9417$ | 9.65 |
| 8 | - | - |

4. Derive the optimal early exercise decision at t = 1. Here, use the coefficient obtained from the last step to approximate the continutation value. Besides, immediate exercise values are also listed for further comparison.

| Path | Exercise | Continuation |
|---|---|---|
| 1 | 1.28 | 1.17 |
| 2 | 0.52 | 0.42 |
| 3 | 1.15 | 0.86 |
| 4 | - | - |
| 5 | 0.57 | 0.39 |
| 6 | 0.22 | 0.84 |
| 7 | 0.85 | 0.43 |
| 8 | - | - |

5. Obtain the final stopping rules. The table below is calculated from comparing immediate exercise values with estimated continuation values.

Two actions are available, i.e.{1,0}, while '1' corresponds to 'exercise' and '0' corresponds to 'continue'.

| Path | t=1 | t=2 |
|---|---|---|
| 1 | 1 | 0 |
| 2 | 1 | 0 |
| 3 | 1 | 0 |
| 4 | 0 | 0 |
| 5 | 1 | 0 |
| 6 | 0 | 1 |
| 7 | 1 | 0 |
| 8 | 0 | 0 |

6. Calculate the option price.

| Path | t=1 | t=2 |
|---|---|---|
| 1 | 1.28 | 0.00 |
| 2 | 0.52 | 0.00 |
| 3 | 1.15 | 0.00 |
| 4 | 0.00 | 0.00 |
| 5 | 0.57 | 0.00 |
| 6 | 0.00 | 1.00 |
| 7 | 0.85 | 0.00 |
| 8 | 0.00 | 0.00 |

We can now calculate the value of the American put option:

$$\frac{(1.28 + 0.52 + 1.15 + 0.57 + 0.85) \times e^{-0.06} + 1 \times e^{-0.12}}{8} \approx 0.6253.$$

While the value of the European put option is:

$$\frac{(1.71 + 0.02 + 0.39 + 1 + 1.24) \times e^{-0.12}}{8} \approx 0.4834.$$

Therefore, as we have expected, the American put option value calculated from LSM is larger than the European put option value calculated from averaging the discounted returns at the final time t = 2.

# 3 Least Square Policy Iteration.

Least Square Policy Iteration (LSPI) was first presented by Michail G. Lagoudakis and Ronald Parr to solve control problems by Reinforcement Learning techniques in 2003 and was later applied to approximate the American option pricing by Yuxi Li in 2009. LSPI determines the option pricing by searching for the optimal exercise policy, i.e. find sequential decision such that $Q^*(s,a) = sup_\pi Q_\pi(s,a)$ in every state s.

## 3.1 Terms introduction.

**Terms:**

1. Batch RL: decouples data collections and optimization, i.e. we do not update weights very often. Actually, we are going to adopt this batch updating in LSPI as it is more stable in practice, otherwise quasi-singular matrix might be encountered during implementation.

2. Policy Iteration: Policy Evaluation (improve value function Q) + Policy Improvement (update policy by maximizing the value function).

## 3.2 Set-up.

**Set-up:**

1. MDP scenario:

- m simulated paths indexed by i = 0, 1,..., (m-1).

- (n+1) time steps indexed by j = n, (n-1),..., 1, 0.
- State: $s \in S_t$ is the stock price at time $t$.
- Strike price: $X$.
- Stock price $S_{t+1} \sim P_t\{\cdot|S_t\}$, $t \in \{0, 1, 2, .....T-1\}$.
- Two actions: $A \in \{1, 0\}$, i.e. exercise & continue.
- The payoff for American put option at state $s$ is defined by function $g(s) = max(0, X - s)$.
- $\mathbf{C}(\mathbf{t})$ denotes the two-tuple $(S_t, t)$.
- The rewards $r : \mathbf{C} \times A \to \mathbb{R}$, where $r((s,t),1) = g(s)$ if the option is exercised at $t$, otherwise, $r((s,t),0) = 0$.

2. Use basis function approximation rather than real computation.

3. Essence of this algorithm: Find the desired set of weights from solution of the linear system: $Q(s,a) = (w^{(i)})^T \phi(s,a)$, where $\phi$ is the basis function of the simulated paths, and this is similar to solving the linear system $Ax = b$.

4. LSPI is run to find the policy that selects the action in every state $s$ to maximize $Q^\pi = \phi w^\pi$ .

5. Using the linear approximation and the Bellman equation derived above, we

shall have (fitted value $\approx$ true value):

$$\phi w \approx R + \gamma P^{\pi} \phi w \tag{3.1}$$
$$(\phi - \gamma P^{\pi} \phi) w \approx R.$$

## 3.3 Derivations.

Define the Projection Operation as: $\prod = \phi(\phi^T D \phi)^{-1} \phi^T D$, where D is the $|s| * |s|$ diagonal matrix with $p(S_t = s)$ on diagonal.

Here, deterministic policy is used and thus D is simply the identity matrix.

Suppose that columns of $\phi$ are lineraly independent, $Q^{\pi} = \phi w^{\pi}$ would be invariant under orthogonal projection.

Then, apply the orthogonal projection to Equation (3.1)

$$\rightarrow \phi(\phi^T \phi)^{-1} \phi^T (R + \gamma P^{\pi} \phi w^{\pi}) \approx \phi w^{\pi}$$
$$\rightarrow w^{\pi} \approx (\phi^T (\phi - \gamma P^{\pi} \phi))^{-1} \phi^T R.$$

Thus, $A \leftarrow \phi^T(\phi - \gamma P^{\pi} \phi)$ and $b \leftarrow \phi^T R$.

## 3.4 A Simple Numerical Example.

For better illustration, we are going to use the same stock price paths as in Section 2. However, because the example has limited stock paths, we shall set the batch size

**Algorithm 1** Least Square Policy Iteration for Continuation Value

$A \leftarrow 0, B \leftarrow 0, w \leftarrow 0$.
for $i \leftarrow 0$ to $m-1$
  For $j \leftarrow 0$ to $n-1$
    $Q \leftarrow Payoff(s_{i,j+1})$.
    $P \leftarrow \phi(s_{i,j+1})$ if $j < n-1$ and $Q \leq w \cdot \phi(s_{i,j+1})$ else 0.
    $R \leftarrow Q$ if $Q > w \cdot P$ else 0.
    $A \leftarrow A + \phi(s_{i,j}) \cdot (\phi(s_{i,j}) - e^{-r_{t_j}(t_{j+1}-t_j)} \cdot P)^T$ [Note that here P is equivalent to $P^\pi \phi$ as this is for time (j+1)].
    $B \leftarrow B + e^{-r_{t_j}(t_{j+1}-t_j)} \cdot R \cdot \phi(s_{i,j})$.
  $w \leftarrow A^{-1} \cdot b, A \leftarrow 0, b \leftarrow 0$ if $(i+1)\%BatchSize == 0$. [Batch RI]

to be 3 to better use the data. The stopping rule is shown below:

| Path | t=1 | t=2 |
|---|---|---|
| 1 | 0 | 1 |
| 2 | 1 | 0 |
| 3 | 1 | 0 |
| 4 | 0 | 0 |
| 5 | 1 | 0 |
| 6 | 0 | 1 |
| 7 | 0 | 1 |
| 8 | 0 | 0 |

Then, the undiscounted reward is:

| Path | t=1 | t=2 |
|---|---|---|
| 1 | 0.00 | 1.71 |
| 2 | 0.52 | 0.00 |
| 3 | 1.15 | 0.00 |
| 4 | 0.00 | 0.00 |
| 5 | 0.57 | 0.00 |
| 6 | 0.00 | 1.00 |
| 7 | 0 | 1.24 |
| 8 | 0.00 | 0.00 |

Hence, the corresponding value for American put option is:

$$\frac{(0.52 + 1.15 + 0.57) \times e^{-0.06} + (1.71 + 1 + 1.24) \times e^{-0.12}}{8} \approx 0.7016.$$

After comparison, the value approximated by LSPI is larger than that by LSM.

# 4 Finite Difference Method.

The introduction of Finite Difference Method in this section comes from both Numerical Methods in Finance: An MATLAB-Based Introduction by Paolo Brandimarte and Professor José Figueroa-López's lecture note on Advanced Probability and Options, with Numerical Methods.

The Finite Difference Method (FDM) aims to find a numerical solution to a well-posed differential equation at the points of a regular grid of the equation's domain. In option pricing, we would determine the option pricing by applying FDM to the Black Scholes Equation, i.e. Equation(4.1).

## 4.1 Background.

**Definition 4.1.** Black Scholes Equation:

$$\frac{\partial f}{\partial t} + \frac{\sigma^2 S^2}{2}\frac{\partial^2 f}{\partial S^2} + rS\frac{\partial f}{\partial S} - rf = 0, \tag{4.1}$$

where stock price $S = S(t) \in [0, \infty)$, t$\in [0, T]$, $f = f(t, S)$ is the price of the option, $r$ is the risk-free interest rate and $\sigma$ is the volatility of the stock.

In order to apply finite difference method to solve the PDE, we must set up a discrete grid with respect to time $t$ and stock price $S$:

$$t = 0, \delta t, 2\delta t, ..., N\delta t = T.$$

$$S = 0, \delta S, 2\delta S, ..., M\delta S = S_{max}.$$

For simplicity sake, denote $f_{i.j} = f(i\delta t, j\delta S)$.

By Taylor's Theorem, there are different ways to approximate the partial derivatives:

- Forward Difference:

$$\frac{\partial f}{\partial S} = \frac{f_{i,j+1} - f_{i,j}}{\delta S}, \ \frac{\partial f}{\partial t} = \frac{f_{i+1,j} - f_{i,j}}{\delta t}.$$

- Backward Difference:

$$\frac{\partial f}{\partial S} = \frac{f_{i,j} - f_{i,j-1}}{\delta S}, \frac{\partial f}{\partial t} = \frac{f_{i,j} - f_{i-1,j}}{\delta t}.$$

- Symmetric Difference:

$$\frac{\partial f}{\partial S} = \frac{f_{i.j+1} - f_{i,j-1}}{2\delta S}, \frac{\partial f}{\partial t} = \frac{f_{i+1,j} - f_{i-1,j}}{2\delta t}.$$

- For the second derivative, we have

$$\begin{aligned}\frac{\partial^2 f}{\partial S^2} &= (\frac{f_{i.j+1} - f_{i,j}}{\delta S} - \frac{f_{i,j} - f_{i,j-1}}{\delta S})/\delta S \\ &= \frac{f_{i,j+1} + f_{i,j-1} - 2f_{i,j}}{\delta S^2}.\end{aligned}$$

*Remark.* For European options, Equation (4.1) can be solved analytically.

For call option with boundary condition $f(T, S) = max(S - X, 0)$, we have:

$$C = S_0 N(d_1) - X e^{-rT} N(d_2),$$

where $N$ denotes the cumulative distribution function for standard normal $N(0,1)$ and:

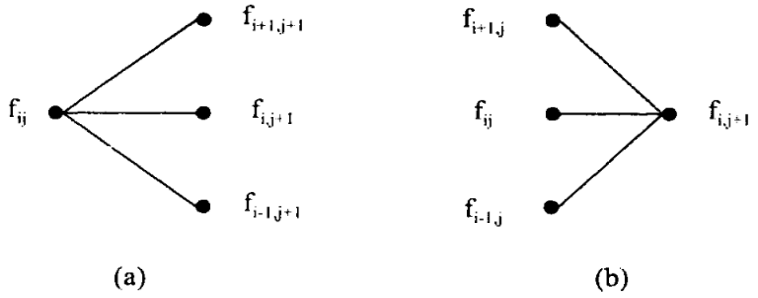
$$d_1 = \frac{log(\frac{S_0}{X}) + (r + \frac{\sigma^2}{2})T}{\sigma\sqrt{T}},$$
$$d_2 = d_1 - \sigma\sqrt{T}.$$

For put option with boundary condition $f(T,S) = max(X - S, 0)$, we have:

$$P = Xe^{-rT}N(-d_2) - S_0N(-d_1).$$

## 4.2 Derivations.

There are three main schemes in general to solve the PDE's numerically, i.e. Explicit Method, Implicit Method and the Crank Nicolson Method.




(a) denotes the explicit scheme while (b) denotes the implicit scheme.

**Definition.** Explicit Method is to apply the Backward Difference to approximate the PDE w.r.t time.

Implicit method is to apply the Forward Difference to approximate the PDE w.r.t time.

The Crank Nicolson Method is a combination of the two methods above.

*Remark.* We will only discuss the Implicit Method and the Crank Nicolson Method in this thesis as Explicit Method will incur possible stability issues.

### 4.2.1 European Put Options.

- **Implicit method.**

Use the Forward Difference method and sub into Equation(4.1), we shall have

$$\frac{f_{i+1,j} - f_{i,j}}{\delta t} + rj\delta S\frac{f_{i,j+1} - f_{i,j-1}}{2\delta S} + \frac{\sigma^2 j^2 \delta S^2}{2} \times \frac{f_{i,j+1} + f_{i,j-1} - 2f_{i,j}}{\delta S^2} = rf_{i,j}.$$

The above equation can be reduced to:

$$f_{i+1,j} = a_j f_{i,j-1} + b_j f_{i,j} + c_j f_{i,j+1}, \tag{4.2}$$

for $i = 0, 1, ...N-1$, and $j = 1, 2, ...M-1$ and

$$a_j = \frac{\delta t(rj - \sigma^2 j^2)}{2},$$
$$b_j = 1 + \delta t(\sigma^2 j^2 + r),$$
$$c_j = -\frac{\delta t(\sigma^2 j^2 + rj)}{2}.$$

Equation(4.2) can be rewritten in the following matrix form:

$$\begin{bmatrix} b_1 & c_1 & & & & \\ a_2 & b_2 & c_2 & & & \\ & a_3 & b_3 & c_3 & & \\ & & ... & ... & ... & \\ & & & a_{M-2} & b_{M-2} & c_{M-2} \\ & & & & a_{M-1} & b_{M-1} \end{bmatrix} \begin{bmatrix} f_{i,1} \\ f_{i,2} \\ f_{i,3} \\ ... \\ f_{i,M-2} \\ f_{i,M-1} \end{bmatrix}$$

$$= \begin{bmatrix} f_{i+1,1} \\ f_{i+1,2} \\ f_{i+1,3} \\ ... \\ f_{i+1,M-2} \\ f_{i+1,M-1} \end{bmatrix} - \begin{bmatrix} a_1 f_{i+1,0} \\ 0 \\ 0 \\ ... \\ 0 \\ c_{M-1} f_{i+1,M} \end{bmatrix}.$$

### 4.2.2 Barrier Option.

We will consider the down-and-out put option, meaning that the option would be void once the asset price falls below the agreed barrier, say $S_{barrier}$. In this case, the boundary conditions would be: $f(t, S_{max}) = 0, f(t, S_{barrier}) = 0$.

- **Method 2: Crank Nicolson Method.**

Apply the Crank Nicolson Method to Equation(4.1), we shall have:

$$\frac{f_{i,j} - f_{i-1,j}}{\delta t} + \frac{rj\delta S}{2} \times \frac{f_{i-1,j+1} - f_{i-1,j-1}}{2\delta S} + \frac{rj\delta S}{2} \times \frac{f_{i.j+1} - f_{i,j}}{2\delta S} + \frac{\sigma^2 j^2 (\delta S)^2}{4} \times$$

$$\frac{f_{i-1,j+1} - 2f_{i-1,j} + f_{i-1,j-1}}{(\delta S)^2} + \frac{\sigma^2 j^2 (\delta S)^2}{4} \times \frac{f_{i,j+1} - 2f_{i,j} + f_{i,j-1}}{(\delta S)^2} = \frac{r}{2} f_{i-1.j} + \frac{r}{2} f_{i,j}. \tag{4.3}$$

The equation above can be reduced to:

$$-\alpha_j f_{i-1,j-1} + (1 - \beta_j) f_{i-1,j} - \gamma_j f_{i-1,j+1} = \alpha_j f_{i,j-1} + (1 + \beta_j) f_{i.j} + \gamma_j f_{i,j+1},$$

where:

$$\alpha_j = \frac{\delta t(\sigma^2 j^2 - rj)}{4},$$
$$\beta_j = -\frac{\delta t(\sigma^2 j^2 + r)}{2},$$
$$\gamma_j = \frac{\delta t(\sigma^2 j^2 + rj)}{4}.$$

We can also rewrite Equation(4.3) in the matrix form:

$$\mathbf{M_1 f_{i-1}} = \mathbf{M_2 f_i},$$

where

$$M_1 = \begin{bmatrix} 1-\beta_1 & -\gamma_1 & & & & \\ -\alpha_2 & 1-\beta_2 & -\gamma_2 & & & \\ & -\alpha_3 & 1-\beta_3 & -\gamma_3 & & \\ & & ... & ... & ... & \\ & & & -\alpha_{M-2} & 1-\beta_{M-2} & -\gamma_{M-2} \\ & & & & -\alpha_{M-1} & 1-\beta_{M-1} \end{bmatrix},$$

$$M_2 = \begin{bmatrix} 1+\beta_1 & \gamma_1 & & & & \\ \alpha_2 & 1+\beta_2 & \gamma_2 & & & \\ & \alpha_3 & 1+\beta_3 & \gamma_3 & & \\ & & ... & ... & ... & \\ & & & \alpha_{M-2} & 1+\beta_{M-2} & \gamma_{M-2} \\ & & & & \alpha_{M-1} & 1+\beta_{M-1} \end{bmatrix},$$

,

$$\mathbf{f}_i = [f_{i,1}, f_{i,2}, ...f_{i,M-1}]^T.$$

### 4.2.3 American Put Options.

*Remark.* For American options, due to its continuous exercisability, or equivalently the free boundary condition $f(S,t) \geq max(X - S(t), 0)$, we cannot find an exact form for the option value, and thus using a numerical algorithm would be recommendable.

For American put options, we should compare the computed $f_{i,j}$ in each iteration with the intrinsic value, i.e. $X - j\delta S$, to see if early exercise could happen, and then update:

$$f_{i,j} = max(f_{i,j}, X - j\delta S).$$

In this scenario, we will use the Crank-Nicolson Method and solve the linear

equations:

$$M_1 f_{i-1} = M_2 f_i + \alpha_1 \begin{bmatrix} f_{i-1,0} + f_{i,0} \\ 0 \\ ... \\ 0 \end{bmatrix} \text{ for i=1,...,N.}$$

Then, for $j = 1, ..., M-1$, define $g_j$ to be the intrinsic value, i.e. $X - j\delta S$. For each time layer i, the iteration scheme is as following:

$$f_{i,1}^{(k+1)} = max\{g_1, f_{i,1}^{(k)} + \frac{\omega}{1-\beta_1}[r_1 - (1-\beta_1)f_{i,1}^{(k)} + \gamma_1 f_{i,2}^{(k)}]\},$$
$$f_{i,2}^{(k+1)} = max\{g_2, f_{i,2}^{(k)} + \frac{\omega}{1-\beta_2}[r_2 + \alpha_2 f_{i,1}^{(k+1)} - (1-\beta_1)f_{i,2}^{(k)} + \gamma_2 f_{i,3}^{(k)}]\},$$

.....

$$f_{i,M-1}^{(k+1)} = max\{g_{M-1}, f_{i,M-1}^{(k)} + \frac{\omega}{1-\beta_{M-1}}[r_{M-1} + \alpha_{M-1} f_{i,M-2}^{(k+1)} - (1-\beta_{M-1})f_{i,M-1}^{(k)}]\},$$

where $\omega$ is the overrelaxation parameter.

In practice, $\omega$ is set to be in the range of $(0, 2)$ to ensure a fast convergence.

# 5 Numerical Results and Comparisons.

In this section, we would present numerical results to evaluate the methods discussed above. Thoughout the implementation, I compared the polynomial basis, i.e. $\{1, x, x^2, x^3, ...\}$, with the suggested weighted Laguerre polynomial basis, i.e. $\{\exp(-X/2), \exp(-X/2)(1-X), \exp(-X/2)(1-2X+\frac{X^2}{2}), ...\}$ and in all the cases, the normal polynomial basis would outweigh the weighted Laguerre polynomial basis in both efficiency and accuracy.

## 5.1 Comparison of approximated option values.

The table below illustrates the results from my own implementation. The strike price is 40 and time to expiration is 1 year. Besides, values for spot price $S$ and volatility $\sigma$ are shown in their corresponding columns. For FDM, the overrelaxation parameter is all set to be 0.9. For LSM and LSPI, results are based on 50,000 simulated stock paths and the option can be exercised 50 times per year. All the results except the Early Exercise Value, i.e. the difference between FDM and Closed Form European, are rounded to 3 decimal places.

| S | $\sigma$ | FDM | Closed Form European | Early Exercise Value | LSM | LSPI |
|---|---|---|---|---|---|---|
| 36 | 0.2 | 4.476 | 3.844 | $\approx 0.63$ | 4.471 | 4.473 |
| 36 | 0.4 | 7.103 | 6.711 | $\approx 0.39$ | 7.101 | 7.101 |
| 38 | 0.2 | 3.242 | 2.852 | $\approx 0.40$ | 3.231 | 3.233 |
| 38 | 0.4 | 6.149 | 5.834 | $\approx 0.31$ | 6.143 | 6.144 |
| 40 | 0.2 | 2.304 | 2.066 | $\approx 0.24$ | 2.294 | 2.313 |
| 40 | 0.4 | 5.312 | 5.060 | $\approx 0.25$ | 5.298 | 5.301 |

The table below aims to further evaluate the relationship between the option pricing and the combined effect of risk free interest rate and volatility, where the other factors remain the same as above.

| r | $\sigma$ | FDM | LSM | LSPI |
|---|---|---|---|---|
| 0.01 | 0.1 | 4.092 | 4.087 | 4.090 |
| 0.01 | 0.6 | 10.851 | 10.781 | 10.781 |
| 0.2 | 0.1 | 4 | 3.838 | 3.957 |
| 0.2 | 0.6 | 7.975 | 7.930 | 7.951 |

By comparison, volatility would be a very influential factor in terms of option pricing. The larger volatility is, the higher the option pricing would be. Besides, there is some negative correlation between the risk free interest rate and the option pricing, ceteris paribus.

Generally, the results from LSM would be lower than those from FDM and LSPI. Roughly speaking, the error for LSM could be attributed to two possible reasons. First is related to the Monte Carlo error. The second is related to the conflict between

the heteroscedasticity of the simulated paths and the homoscedasticity requirement of the Least Square estimator.

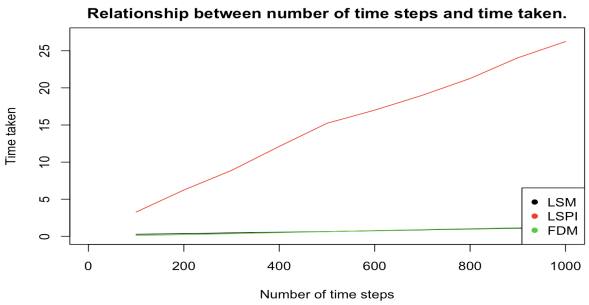
For LSPI, it tends to yield larger values compared with LSM, however, it is far less numerically stable.

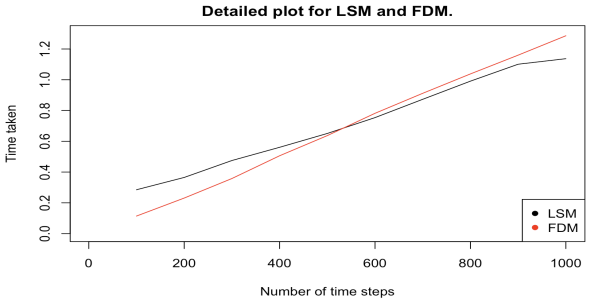
## 5.2 Comparison of execution time.

Now we shall investigate the execution time of LSM, LSPI and FDM under different numbers of time steps. The 'tictoc' package in R is used to calculate the time elapsed.

- Set the initial price $S_0 = 36$, strike price $X = 40$, $\sigma = 0.2$, $T = 1$, r=0.06. Note that for LSM and LSPI, 1000 paths are simulated.

Figure 3




| Number of time steps | LSM | LSPI | FDM |
|---|---|---|---|
| 100 | 0.285s | 3.274s | 0.114s |
| 200 | 0.365s | 6.242s | 0.231s |
| 300 | 0.475s | 8.884s | 0.358s |
| 400 | 0.561s | 12.114s | 0.506s |
| 500 | 0.651s | 15.231s | 0.637s |
| 600 | 0.754s | 17.001s | 0.782s |
| 700 | 0.873s | 19.002s | 0.912s |
| 800 | 0.991s | 21.264s | 1.038s |
| 900 | 1.101s | 24.05s | 1.16s |
| 1000 | 1.137s | 26.237s | 1.286s |
| 10000 | 11.331s | 259.597s | 9.471s |

We can see from both the table and the plot that the time taken for all these methods would grow linearly as number of time steps increases.

- The table below compares the time taken for LSM and LSPI when number

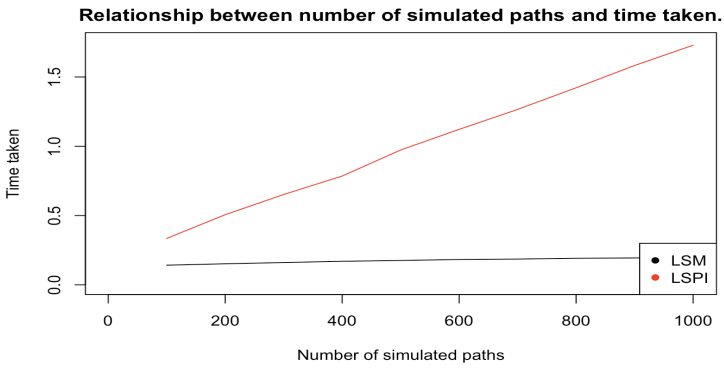
of simulated paths increases. Suppose that the option can be exercised 50 times per year. Set the initial price $S_0 = 36$, strike price $X = 40$, $\sigma = 0.2$, $T = 1$ and $r = 0.06$. Also, batch size is 3 for LSPI.

| Number of simulated paths | LSM | LSPI |
|---|---|---|
| 100 | 0.142s | 0.335s |
| 200 | 0.152s | 0.506s |
| 300 | 0.161s | 0.652s |
| 400 | 0.170s | 0.785s |
| 500 | 0.176s | 0.973s |
| 600 | 0.183s | 1.122s |
| 700 | 0.186s | 1.266s |
| 800 | 0.192s | 1.422s |
| 900 | 0.194s | 1.583s |
| 1000 | 0.198s | 1.729s |
| 10000 | 0.652s | 15.399s |

From both the table and the plot, the time taken for both LSM and LSPI would grow linearly as number of simulated paths increases.

Generally, LSM and FDM tend to be much more time efficient than LSPI. However, it is worth noting that extra cautions about this comparison should be taken into account. First, for LSM, instead of using the closed form solution to compute the Least Squares estimator, I used a built-in function in R (lm) to reduce the execution

Figure 4




time. Secondly, for LSPI, the time taken would vary significantly for different simulated paths. Though I have tried to minimize this difference by taking the average of 50 runs, it is still possible that the averaged time is not very representative.

*Remark.* In the above plots, the results for 10000 are intentionally omitted. Otherwise, the scale of both x-axis & y-axis would expand tenfold and the plot will look 'linear' naturally, leading to a misleading conclusion.

# 6 Conclusions.

In this thesis, we aim to solve the problem of American option pricing by using three different methods. The application of the Finite Difference Method to the option pricing problem is very well-established and one can understand & implement it with ease. For LSM and LSPI, we would apply them to simulated paths first and then search for the optimal policy which determines the option pricing. However, due to the continuous exercisability for American options, we need to make use of a large finite set of exercise time, i.e. Bermuda option in fact, to approximate the American option pricing. Taking the results obtained from FDM as a benchmark, a dilemma exists: LSM is superior to LSPI in terms of time efficiency and numerical stability while LSPI tends to produce larger values than LSM.

In the future, it may be possible to work on an alternative version of the 'Optimal Stopping Problems', i.e. instead of maximizing the payoff of one American option in a given time period, we might savor higher (compound) profit by exercising two or more options, ceteris paribus.

# 7 References.

Papers that have been used:

[1] Francis A. Longstaff, Eduardo S. Schwartz, Valuing American Options by Simulation A Simple Least-Squares Approach. In The Review of Financial Studies (2001) Vol. 14, No 1, pp.113-147.

[2] Yuxi Li, Csaba Szepesvari, Dale Schuurmans, Learning Exercise Policies for American Options. In Proc. of the 12th International Conference on Artificial Intelligence and Statistics (2009), JMLR: W&CP, volume 5, pp.352-359.

[3] Model-free Least-Squares Policy Iteration by Michail G. Lagoudakis and Ronald Parr. In Advances in Neural Information Processing Systems Vol. 14 (2001), Morgan Kaufmann, pp. 1547-1554.

[4] Least-Squares Policy Iteration by Michail G. Lagoudakis and Ronald Parr. In Journal of Machine Learning Research, 4 (2003), pp.1107-1149.