A SMOOTHING FRAMEWORK FOR STOCHASTIC CONTINUOUS-TIME REINFORCEMENT LEARNING PROBLEM

by

Bowen Hu

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Statistics)

at the

UNIVERSITY OF WISCONSIN–MADISON

2021

Date of final oral examination: 04/29/2021

The dissertation is approved by the following members of the Final Oral Committee:

Yazhen Wang, Professor, Statistics
Anru Zhang, Professor, Statistics
Sebastian Raschka, Professor, Statistics
Nicolás García Trillos, Professor, Statistics
Yingyu Liang, Professor, Computer Science

ACKNOWLEDGMENTS

First of all, I would like to express my deepest gratitude to my advisor, Professor Yazhen Wang who has been continuously supporting my research and guiding the direction of this thesis. He always provides me with encouragement and valuable suggestions in my PhD. The completion of my dissertation would not have been possible without his guidance and contribution.

I would also like to offer my special thanks to my dissertation committee members, Professor Anru Zhang, Professor Sebastian Raschka, Professor Nicolás García Trillos, Professor Yingyu Liang. They have provided insightful comments and constructive suggestions to improve this thesis. Thanks should also go to all my peers, faculties and staffs in the Department of Statistics at the University of Wisconsin-Madison for their helpful discussion and relentless support. This research was supported by NSF grants DMS-1707605 and DMS-1913149

Last but not least, my extreme and sincere gratitude to my family. I very much appreciate my husband Jurijs's profound belief in my abilities and offering to discuss all of my ideas with invaluable insight. I'm deeply grateful to my parents, Dong Wang and Enhua Hu, for their unwavering support, patience and encouragement throughout my graduate studies and my whole life.

— Bowen Hu

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Reinforcement learning problem embraces many breakthroughs in stochastic discrete-time and deterministic continuous-time systems. Stochastic continuous-time reinforcement learning is an important yet under studied area. In this dissertation, I present a framework to adapt deterministic continuous time temporal difference learning method to stochastic continuous time systems.

I first review the temporal difference methods of discrete time and deterministic continuous time. Then I discuss a popular method that solves optimal control problem and verify its accuracy with Merton's problem. Motivated by the fact that the stochastic system and corresponding deterministic system can be as close as possible as the variance term decreases to zero, I introduce a new nonparametric smoothing method that generalizes deterministic continuous time method to stochastic problem by shrinking the variance term of the stochastic process. I demonstrate that the smoothing method outperforms traditional deterministic continuous time temporal difference method in our numerical study of the stochastic pendulum. In the end, I provide the proof of the convergence of the solution of the proposed framework to a corresponding deterministic continuous time solution. If the optimal value function and optimal policy can be obtained by traditional deterministic algorithms, then applying kernel smoothing framework with continuous TD guarantees convergence to the optimal value or policy for stochastic process.

1 INTRODUCTION

Reinforcement learning (RL) has recently drawn many attentions with its great applications such as Alpha Go in Silver et al. (2017) and Deep Q-Network (DQN) in Mnih et al. (2015). Most of RL methods are built on processes of discrete time. However, there are many continuous-time RL problems such as stock price and physical process that are not well studied. A common way of dealing with such problems would be discretizing the processes to fit into the discrete-time framework. However, it is stated in Tallec et al. (2019) that the state of the art algorithm such as DQN and Deep Deterministic Policy Gradient from Lillicrap et al. (2015) collapse with small time steps when discretizing continuous-time process.

The direct approaches based on the continuous-time framework is beneficial because it eliminates the error caused by partitioning the state, action, and time in discretization, and potentially accelerates algorithm convergence based on our simulation results. This dissertation proposed a framework that deals with stochastic continuous-time reinforcement learning problem. It utilizes the nonparametric kernel smoothing method in Friedman et al. (2001) in learning the system and establishes convergence result.

In this chapter, I will introduce the reinforcement learning (RL) problems in discrete time and continuous-time, their mathematical foundation and Temporal Difference (TD) learning. In Chapter 2, I will discuss continuous-time optimal control and Markov Chain Approximation (MCA). I will introduce our proposed method for stochastic continuous-time system in Chapter 3, and the simulation results in Chapter 4. In chapter 5, I will give proofs of all theorems and lemmas proposed in this research.

1.1 Reinforcement Learning Process

Reinforcement learning problem is important in many fields and has been studied by many communities. In engineer community, there is an ample body of research in Markov Decision Process (MDP) and dynamic programming especially for discrete-time process. In mathematical finance, people often times consider continuous-time process and refer it to optimal control theories. There has been an increasing interest to combine the findings in all those communities to prevent from reinventing wheels, see reviews of Recht (2019) and Powell and Ma (2011). The goal of reinforcement learning is to learn controlling a system in the way of maximizing a numerical value function which describes a long-term objective, from Szepesvári (2010). Discrete time reinforcement learning problems can usually be described as a Markov Decision Process (MDP). A MDP is consist of two parts, a Markov system and a controller. The controller interacts with the system by receiving signals of states that the system generates, and sending an action accordingly back to the system. The system generates the next state based on previous state and the action, and rewards the controller based on the action (Figure 1.1). This cycle repeats until the process is terminated. The controller is usually called policy, control or agent in literature and I will use them interchangeably. It is a map that takes input of a state from the system and output an action to feed into the system.

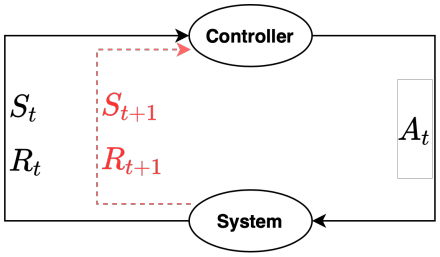


Figure 1.1: The controller-system interaction in a MDP.

More specifically, consider a system with state space $\mathcal{S}$, a control with action space $\mathcal{A}$ and discrete sequence of time steps $t = 0, 1, 2, \dots$. At time step t, the control receives the signal of the system's state, $S_t \in \mathcal{S}$. Then the control calculates an action $A_t = \pi(S_t) \in \mathcal{A}$, where $\pi(\cdot)$ denotes the policy of the control. After one time step, the control receives a new state S_{t+1} from the system and a numerical reward $R_{t+1} = r(S_t, A_t)$ as a consequence of the action, where $r(\cdot, \cdot)$ is the reward function. $S_0, A_0, S_1, R_1, A_1, S_2, \dots$ is called a trajectory or an episode. The interest is to learn the optimal control which maximize the discounted cumulated total rewards until the process is terminated, i.e. $\sum_{t=0}^{\infty} \gamma^t r(S_t, A_t)$. This objective function to optimize in reinforcement learning is called the value function, and I will give the rigorous definition in Section 1.3. The difference between reinforcement learning and optimization problem is that reinforcement learning optimizes a long term unknown objective while optimization problem usually considers a short term and known objective.

Above discrete time set up can be generalized to continuous-time by using a diffusion process to describe the MDP system. I will introduce diffusion process in next section. In continuous-time, the value function is generalized to an integration of discounted rewards.

1.2 Introduction to Itô's Integral

Let's first review Itô's integral which is fundamental to continuous-time reinforcement learning. It is a notion of integration for stochastic process. In this section, I will define Itô's integral for $\mathcal{H}^2[0, T]$ that consist of all measurable adapted functions

$f(w,t)$ that satisfy the integrability constraint

$$\mathbb{E}\left[\int_0^t f^2(w,t)dt\right] < \infty$$

Like Lebesgue-Stieltjes integral, let's start with defining integration for step functions. Let $\mathcal{F}_t^W$ be the filtration of standard Brownian motion W_t. Denote $\mathcal{H}_0^2[0,T]$ as the set of functions of the form $f(w,t) = \sum_{i=0}^{n-1} a_i(w)1(t_i < t < t_{t+1})$ where $a_i \in \mathcal{F}_{t_i}$, $\mathbb{E}a_i^2 < \infty$ and $0 = t_0 < t_1 \cdots < t_n <= T$, define

$$I(f)(w) = \sum_{i=0}^{n-1} a_i(w)(W_{t_{i+1}} - W(t_{t_i})$$

Lemma 1.1. *Given any $f \in \mathcal{H}^2[0,T]$, there is a sequence $\{f_n\} \in \mathcal{H}_0^2[0,T]$ such that f_n converges to f in $\mathcal{L}^2(dP \times dt)$.*

With Lemma 1.1, for each fixed $t > 0$ and $f \in \mathcal{H}^2[0,T]$, let's define $I(f)$ as the limit of $\{I(f_n)\}_n$ in $L^2(\mathbb{P})$, where $\{f_n\}$ is an arbitrary sequence in $\mathcal{H}_0^2[0,T]$ which converges to f in $\mathcal{H}^2[0,T]$. This defines a random variable $I(f) \in \mathcal{L}^2(dP)$ such that $\|I(f_n) - I(f)\|_{\mathcal{L}^2(dP)} \to 0$ as $n \to \infty$. We can further generalize the random variable to a random process. For each $f \in \mathcal{H}^2[0,T]$, we have $f1_{[0,t]} \in \mathcal{H}^2[0,t]$, so we have random variable $I(f1_{[0,t]})$.

Definition 1.2. *For $f \in \mathcal{H}^2[0,T]$, Itô's integral $\int_0^t f_s dW_s$ is defined as the continuous square-integrable martingale M (w.r.t. $\mathcal{F}_t^W$) such that $\mathbb{P}\left(M_t = I(f1_{[0,t]})\right) = 1$ for all $t \in [0,T]$.*

See Steele (2012) for details about Itô's integral and proof of Lemma 1.1 and Theorem 1.3. Similar to fundamental theorem of Calculus, there is a "fundamental" theory in Itô's integral. It is called Itô's lemma or Itô's formula.

Theorem 1.3 (Itô's lemma)**.** *If f has a continuous second derivative, we have*

$$f(W_t) = f(0) + \int_0^t f'(W_s)dW_s + \frac{1}{2}\int_0^t f''(W_s)ds$$

for all $t \geq 0$, a.s. It can be written in short as

$$df(W_t) = f'(W_t)dW_t + \frac{1}{2}f''(W_t)dt$$

We will use Itô's lemma to prove HJB equation in Section 1.3. Now we can define Itô diffusion that is critical to continuous-time reinforcement learning problem.

Definition 1.4. *An Itô diffusion in n-dimensional Euclidean space $\mathbb{R}^n$ is a process $X : [0, +\infty) \times \Omega \to \mathbb{R}^n$ defined on a probability space $(\Omega, \Sigma, \mathbb{P})$ and satisfying a stochastic differential equation of the form*

$$\mathrm{d}X_t = b\left(X_t\right)\mathrm{d}t + \sigma\left(X_t\right)\mathrm{d}W_t$$

where W_t is an m-dimensional Brownian motion and $b : \mathbb{R}^n \to \mathbb{R}^n$ and $\sigma : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ satisfy the usual Lipschitz continuity condition $|b(x) - b(y)| + |\sigma(x) - \sigma(y)| \leq C|x - y|$

b is known as the drift coefficient and σ is known as the diffusion coefficient or variance of X_t. Itô diffusion is usually called in short as diffusion. In optimal control and continuous-time reinforcement learning, it is often assumed that the MDP system is a diffusion process.

1.3 Mathematical Foundations of Reinforcement Learning

The theories behind reinforcement learning is Bellman equations in discrete-time process and Hamilton-Jacobi-Bellman (HJB) equations in continuous-time process. These two equations are usually hard to solve analytically, and we have to develop algorithms to solve them numerically. There are many reinforcement learning algorithms motivated by Bellman equation, such as dynamic programming which directly applies the Bellman equation and Q-learning in Mnih et al. (2015) which approximate the Bellman equation, etc. The HJB equation is vital to optimal control theory and I will introduce Markov Chain Approximation (MCA) method which solves the HJB equation numerically in Chapter 2. It also motivates the algorithm that I propose in Chapter 3.

Bellman equation for discrete-time systems

Definition 1.5. *Assume the system is Markovian with transition probability $P(S_{t+1}|S_t, A_t)$. The **value function** for discrete-time process is defined as the expected cumulative reward of one trajectory,*

$$V^\pi(s) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(S_t, A_t) \middle| S_0 = s \right] \tag{1.1}$$

where γ is the discount factor.

Let optimal value function $V^*(s) = \max_\pi V^\pi(s)$. Look one step further in value function definition, we have the Bellman equation,

$$V^\pi(s) = \mathbb{E}_\pi[R_0 + \gamma V_\pi(S_1) \mid S_0 = s] \tag{1.2}$$

and Bellman optimality equation or dynamic programming equation,

$$V^*(s) = \mathbb{E}[R_0 + \gamma V^*(S_1) \mid S_0 = s] \tag{1.3}$$

The Bellman equation describes the recursive relation of the value function of current state and future state. It provides a way of finding the value function and motivates the field of dynamic programing, starting with Bellman (1956). Howard (1960) provides a summary of research progress in MDP with dynamic programming. See Bertsekas et al. (1995) for detailed review of dynamic programming algorithms in solving MDPs.

HJB equation for continuous-time systems

The HJB equation is a continuous generalization of Bellman equation. It takes the form of stochastic PDE as the generalization of discrete-time recursion.

Definition 1.6. *For a stochastic process system X_t, the **value function** for continuous-time is defined as the expected integrated reward of one trajectory,*

$$V^{\pi}(x) = E\left[\int_0^{\infty} e^{-\beta t} r(X_t, \pi(X_t))dt \middle| X_0 = x\right] \tag{1.4}$$

where β is the continuous discounted factor

Let optimal value function $V^*(s) = \sup_{\pi} V^{\pi}(s)$. If the system X_t is a diffusion process with $dX_t = b(X_t, \pi(X_t))dt + \sigma(X_t, \pi(X_t))dW_t$, the value functions satisfies HJB equation,

$$\beta V^{\pi}(x) = b\frac{\partial V^{\pi}}{\partial x} + \frac{1}{2} tr(\sigma\sigma' \frac{\partial^2 V^{\pi}}{\partial x^2}) + r(x, \pi(x)) \tag{1.5}$$

and HJB optimality equation,

$$\beta V^*(x) = \sup_{\pi} \left\{ b\frac{\partial V^*}{\partial x} + \frac{1}{2} tr(\sigma\sigma' \frac{\partial^2 V^*}{\partial x^2}) + r(x, \pi(x)) \right\} \tag{1.6}$$

The proof of (1.5) is in Section 5.1 and (1.6) can be proved by the verification theorem, see Pham (2009).

The solution to HJB optimality equation is the optimal value function in which we are interested. However, It is usually hard to find an analytical solution especially when the equation is not linear. Numerical PDE solvers also don't immediately apply because of the supremum in the equation. The Merton's problem of one of the few examples when we can find the analytical solution which is included in our numerical study in Section 2.3.

1.4 Introduction to Discrete-Time Reinforcement Learning Algorithms

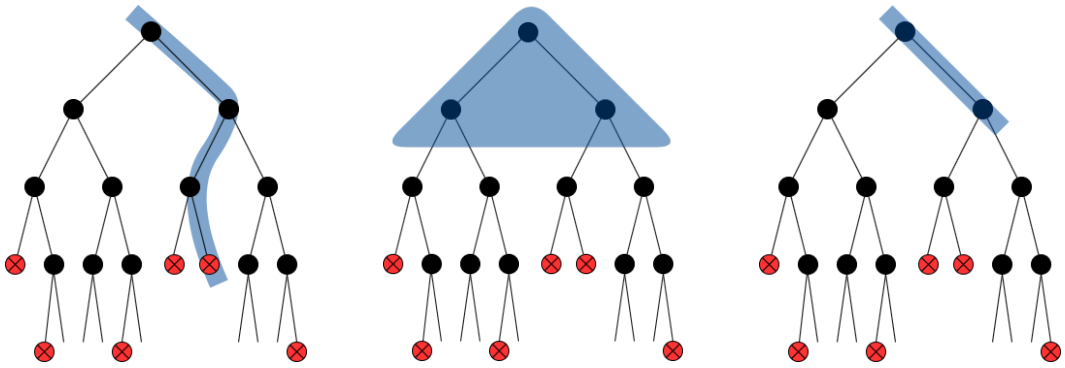
The key to RL is to approximate value function $V^\pi(\cdot)$. Then the optimal policy is the one that maximize the value function. There are three main methods to learn the value function, Monte-Carlo, dynamic programming, and TD learning (Sutton et al. (1998)).

Monte-Carlo and dynamic programming

The goal of Monte-Carlo methods is to approximate $V^\pi(s)$ from n episodes of experience sampled from policy π, e.g., episode $E_i = (S_1, A_1, R_2, S_2, A_2 \dots) \sim \pi$. Denote G_i as the total rewards of E_i. The main idea is to use Monte-Carlo method

$V_n = \frac{1}{n}\sum_{i=1}^{n} G_i$ to estimate the value function $V^{\pi}(s) = \mathbb{E}_{\pi}(G_t \mid S_0 = s)$. There are some algorithmic technique such as the incremental trick to improve efficiency.

Dynamic programming methods take advantage of the Bellman optimality equation (1.3) to learn the value function which minimizes the Bellman error, $\mathbb{E}_{\pi}[R_0 + \gamma V(S_1) \mid S_0 = s] - V(s)$.



(a) Monte Carlo (b) Dynamic Programing (c) TD

Figure 1.2: Learning diagram in Markov decision process. Each node is a state in Markov chain. The roots are initial states of Markov process. Red nodes are the terminated states. States used in one learning update are highlighted in blue.

TD(0)

The successful implementations of RL, for example DQN, are motivated by temporal difference (TD) error proposed in Sutton et al. (1998), which is motivated by Bellman equation (1.2). The TD error (also called TD(0) error) is defined as,

$$\delta_t = R_{t+1} + \gamma V\left(S_{t+1}\right) - V\left(S_t\right)$$

where $V(\cdot)$ is the estimation of value function. It is the difference between value function of current state and its one step ahead estimation using the value of next

state. Now we can derive the simplest TD learning method:

$$V(S_t) \leftarrow V(S_t) + \alpha \left[R_{t+1} + \gamma V(S_{t+1}) - V(S_t) \right] \tag{1.7}$$

We can define TD(k) error by generalizing TD(0), the difference between the value and its 1-step ahead estimation, to the value and its $k+1$ steps ahead estimation, i.e. $\delta_t^k = R_{t+1} + \gamma V(S_{t+1}) + \cdots + \gamma^k V(S_{t+k+1}) - V(S_t)$. Because an estimation $V(\cdot)$ is involved in TD error, TD is also called a bootstrapping method, which update estimates from other estimates. In contrast, Monte Carlo estimates values for each state independently. TD methods are more efficient than Monte Carlo. Because one has to wait till the end of an episode and get the final reward to learn the value function in Monte Carlo methods. However, TD methods updates at every step, and there is no need to wait for final outcome. There are mainly three types of TD algorithms. First is gradient based method that optimizes the cost function by stochastic gradient descent, such as Sutton et al. (2009a) and Sutton et al. (2009b), Second is least squared based method that uses the closed form of least-square solution, such as Bradtke and Barto (1996) and Boyan (2002). Third is probabilistic based methods, such as Engel et al. (2003). Dann et al. (2014) gives a thorough overview and performance comparison of all TD based leaning algorithms.

TD(λ)

TD(λ) in Sutton et al. (1998) builds a bridge from TD to Monte Carlo methods. It minimizes the weighted sum of TD(k) error with weights $\lambda^{k-1}(1-\lambda)$, $k = 1, \ldots, \infty$, which is controlled by the parameter $\lambda \in [0, 1]$. When $\lambda = 1$, it is equivalent to a Monte Carlo method and when $\lambda = 0$, it is equivalent to the TD(0) method. Intermediate λ in between usually outperforms the either end point 0 or 1. The

TD(λ) algorithm is stated as following. Define eligibility trace z_t as

$$
\begin{aligned}
&z_{-1} = 0 \\
&z_t = \gamma\lambda z_{t-1} + \nabla\hat{V}\left(S_t, w_t\right), \quad 0 \leq t \leq T
\end{aligned}
$$

to keep track of which components of the weight vector have contributed. And update weights w as,

$$w_{t+1} = w_t + \eta\delta_t z_t$$

1.5 Introduction to Deterministic Continuous-Time Reinforcement Learning Algorithms

Let's focus on deterministic continuous-time process in this section. The continuous TD learning algorithms were first proposed in Doya (2000). I will introduce the algorithm and theories for stochastic continuous-time process I propose in chapter 3.

Continuous TD(0)

I have discussed in last section that discrete TD error is derived from Bellman equations. To generalize it to continuous-time cases, we should consider HJB equation (1.5), which is the continuous-time generalization of Bellman equations. Consider a deterministic continuous-time system,

$$dX_t = b(X_t, \pi(X_t))dt \tag{1.8}$$

with reward function $r(X_t, \pi(X_t))$. Plug terms in (1.5) and move left hand side to right hand side. We have the continuous TD error,

$$\delta_t = r(X_t, \pi(X_t)) + b(X_t, \pi(X_t))\frac{dV(X_t)}{dx} - \beta V(X_t) \tag{1.9}$$

for an estimation $V(X_t)$ of true value function. Notice that $V(X_t)$ is the true value function if and only if $\delta_t = 0$ in deterministic systems. We can use gradient descent algorithm to minimize the loss $\frac{1}{2}\delta_t^2$. Denote the approximator as $\hat{V}(X_t, w)$ where w is the parameter to learn, then w can be updated by,

$$\Delta w = -\eta \delta_t \frac{d\delta_t}{dw}$$

Connections between continuous and discrete TD(0) errors

We can approximate $dV(X_t)/dt$ in continuous TD error in (1.9) with backward Euler method as $\dot{V}(X_t) = \frac{V(X_t) - V(X_{t-\Delta t})}{\Delta t}$. Plug it back to the continuous TD(0) error, we have

$$\begin{aligned}
\delta_t &= r(X_t, \pi(X_t)) + \frac{V(X_t) - V(X_{t-\Delta t})}{\Delta t} - \frac{1}{\tau}V(X_{t-\Delta t}) \\
&= r(X_t, \pi(X_t))\frac{1}{\Delta t}\left[\left(1 - \frac{\Delta t}{\tau}\right)V(X_t) - V(X_{t-\Delta t})\right] \\
&\approx r(X_t, \pi(X_t)) + \frac{1}{\Delta t}\left[e^{-\frac{\Delta t}{\tau}}V(X_t) - V(X_{t-\Delta t})\right]
\end{aligned}$$

The last expression is the same as discrete TD(0) with time step Δt if we allow a scaling factor $\frac{1}{\Delta t}$.

Continuous TD(λ)

The continuous TD(λ) is the continuous-time generalization of discrete TD(λ). The learning algorithms for $\hat{V}(X_t, w)$ is derived in Doya (2000) as

$$w_{t+1} = w_t + \eta\delta_t e(t)$$
$$\dot{e}_i(t) = -\frac{1}{\kappa}e_i(t) + \frac{\partial V(X_t; w_i)}{\partial w_i}$$

where $e(t) = (e_1(t), \ldots, e_p(t))$ is the continuous eligibility trace and $0 < \kappa \leq \tau$ is the time constant of the eligibility trace. Similar to discrete-time reinforcement learning process, the continuous TD(λ) performs better than the vanilla continuous TD(0).

Both methods I introduced in this section for deterministic continuous-time reinforcement learning problems perform poorly on stochastic continuous-time problems. I will show this numerically in chapter 4. There is a numerical method in optimal control community to solve stochastic continuous-time problems called Markov chain approximation. I will discuss it in chapter 2. It requires some mathematical derivation beforehand in order to apply the algorithm. This dissertation proposes a nonparametric kernel based method to generalize deterministic method to stochastic cases. It doesn't require any mathematical derivation and learns the optimal value function and control function through training. The method is introduced in chapter 3.

2 CONTINUOUS TIME OPTIMAL CONTROL AND MARKOV CHAIN APPROXIMATION

2.1 Continuous Time Optimal Control

Optimal control studies the same the problem as continuous time reinforcement learning. Given diffusion process

$$dX_t = b(X_t, \pi(X_t))dt + \sigma(X_t)dW_t \tag{2.1}$$

with value function (1.4), we want to find the optimal value function $V(x) = \sup_\pi V(x, \pi(x))$, and the optimal control π^* by solving the HJB equation (1.6). Here we only consider uncontrolled variance σ in (2.1) for simplicity. Discussion about controlled variance process can be found in Kushner et al. (1990). Conventional PDE solvers doesn't apply to HJB equation because of the supremum in the equation. In optimal control, Markov Chain Approximation (MCA) is one of the popular methods to solve it numerically.

2.2 Markov Chain Approximation

MCA is proposed in Kushner et al. (1990). The idea is to approximate the value function by the discrete value function of a discrete Markov chain $\{\xi_n\}$ that approximates X_t. If the Markov chain satisfies local consistency condition, then its value function converges to the value function of X_t as the step size of Markov chain goes to zero. It breaks down the problem of continuous time optimal control into two sub-problem, find the Markov chain with local consistency and solve the

optimal value function of the Markov chain. The control problem of Markov chain corresponds to discrete time reinforcement learning problem. It can be solved by methods introduced in section 1.4 such as dynamic programing, etc. In the rest of this section, we will introduce finite difference method to approximate the diffusion process X_t with a Markov chain with local consistency.

Finite Difference Method

Finite difference method is used to obtain a local consistent Markov chain to approximate the diffusion process $X_t \in \mathbb{R}^d$ defined in (2.1). Denote $\sigma = (\sigma_1, \ldots, \sigma_d)$. Assume,

$$\sigma_i^2(x) - \sum_{j:j\neq i} |\sigma_i(x)\sigma_j(x)| \geq 0 \tag{2.2}$$

then the transition probability of the Markov chain can be found as below. Condition (2.2) is necessary so that the transition probabilities of the chain from finite difference are non-negative. The finite difference method is consist of two steps.

Step 1. Approximate derivatives with finite difference method with step size h,

$$f_{x_i}(x) \to \frac{f(x+e_ih)-f(x)}{h} \qquad \text{if } b(x) \geq 0 \tag{2.3a}$$

$$f_{x_i}(x) \to \frac{f(x)-f(x-e_ih)}{h} \qquad \text{if } b(x) < 0 \tag{2.3b}$$

$$f_{x_ix_i}(x) \to \frac{f(x+e_ih)+f(x-e_ih)-2f(x)}{h^2} \tag{2.3c}$$

$$\begin{aligned} f_{x_ix_j}(x) \to & \left[2f(x)+f\left(x+e_ih+e_jh\right)+f\left(x-e_ih-e_jh\right)\right]/2h^2 \\ & -\left[f\left(x+e_ih\right)+f\left(x-e_ih\right)+f\left(x+e_jh\right)\right. \\ & \left.+f\left(x-e_jh\right)\right]/2h^2 \qquad \text{if } \sigma_i(x)\sigma_j(x) \geq 0 \end{aligned} \tag{2.3d}$$

$$\begin{aligned} f_{x_ix_j}(x) \to & -\left[2f(x)+f\left(x+e_ih-e_jh\right)+f\left(x-e_ih+e_jh\right)\right]/2h^2 \\ & +\left[f\left(x+e_ih\right)+f\left(x-e_ih\right)+f\left(x+e_jh\right)\right. \\ & \left.+f\left(x-e_jh\right)\right]/2h^2 \qquad \text{if } \sigma_i(x)\sigma_j(x) < 0 \end{aligned} \tag{2.3e}$$

where e_i is the standard basis vector of i-th dimension. Rewrite HJB equation (1.5) by substituting derivatives with above finite difference approximations.

Step 2. Rearrange terms in finite difference approximated HJB equation to match the following expression,

$$V_h(x,\pi(x)) = \sum_y t_y V_h(y,\pi(x)) + d_y r(x,\pi(x))$$

Then we match the transition probability and interpolation intervals as:

$$p^h(x,y \mid \pi(x)) = t_y$$
$$\Delta t^h(x,\pi(x)) = d_y$$

In this way, we can identify the expression for transition matrix for the ap-

proximated Markov chain, and the interpolation interval. Denote the positive and negative parts of a real number as $r^+ = \max[0, r]$ and $r^- = -\max[0, -r]$. We have the transition probability of Markov chain $\xi_n^h \in \mathbb{R}^d$ as,

$$p^h\left(x, x \pm e_i h \mid \pi(x) = a\right) = \frac{\left[\sigma_i^2(x)/2 - \sum_{j:j\neq i} \left|\sigma_i(x)\sigma_j(x)\right|/2 + h b_i^{\pm}(x, a)\right]}{Q^h(x, a)} \tag{2.4a}$$

$$\begin{aligned} p^h\left(x, x + e_i h + e_j h \mid \pi(x) = a\right) &= p^h\left(x, x - e_i h - e_j h \mid \pi(x) = a\right) \\ &= \frac{\left(\sigma_i(x)\sigma_j(x)\right)^+}{2Q^h(x, a)} \end{aligned} \tag{2.4b}$$

$$\begin{aligned} p^h\left(x, x - e_i h + e_j h \mid \pi(x) = a\right) &= p^h\left(x, x + e_i h - e_j h \mid \pi(x) = a\right) \\ &= \frac{\left(\sigma_i(x)\sigma_j(x)\right)^-}{2Q^h(x, a)} \end{aligned} \tag{2.4c}$$

where

$$Q^h(x, \alpha) = \sum_i \sigma_i^2(x) - \sum_{i,j:i\neq j} \left|\sigma_i(x)\sigma_j(x)\right|/2 + h\sum_i \left|b_i(x, \alpha)\right|$$

The interpolation interval is

$$\Delta t^h(x, \alpha) = \frac{h^2}{Q^h(x, \alpha)} \tag{2.5}$$

Local consistency is the necessary condition for $V_h(x, \pi(x)) \to V(x, \pi(x))$ as $h \to 0$. Now let's give the formal definition of local consistency.

Definition 2.1. *Given interpolation interval $\Delta t^h(x, \alpha)$ and $\{\xi_n^h\}$, let $\Delta\xi_n^h = \xi_{n+1}^h - \xi_n^h$.*

Then ξ_n^h is local consistent to X if it satisfies,

$$\begin{cases} \mathbf{E}\left[\Delta\xi_n^h \mid \xi_i^h, u_i^h, i \leq n, \xi_n^h = x, u_n^h = a\right] = \Delta t^h(x,a)b(x,a) + O\left(\Delta t^h(x,a)\right) \\ \mathbf{Cov}\left(\Delta\xi_n^h \mid \xi_i^h, u_i^h, i \leq n, \xi_n^h = x, u_n^h = a\right) = \Delta t^h(x,a)\sigma(x)\sigma'(x) + O\left(\Delta t^h(x,a)\right) \\ \sup_n |\xi_{n+1}^h - \xi_n^h| = O(h) \end{cases}$$

Local consistency means that, locally, the conditional mean and variance of the change of Markov chain state ξ_n^h is proportional to the local mean and variance of the original process X_t, from Kushner and Dupuis (1992). It is proven that the approximated Markov chain from finite different method is local consistent and its value function converge to the value function of the original process in Kushner and Dupuis (1992) for uncontrolled variance process (i.e. $\sigma(x)$ is free of control). The finite difference method is still valid for controlled variance process and the convergence of approximated Markov chain is proven in Kushner et al. (1990) Section 8. Detailed discussion can be found in Kushner (1999).

There is one simple and common way to to improve accuracy and computation speed using "splitting the operator" if we know the signs of some parameters. Details see section 2.3.

2.3 Use MCA to Solve Merton's Problem

Merton's problem is a well-known finance problem. It models an investment problem as maximize the expected utility of a portfolio of stock and risk-free bond. It is an optimal control problem and corresponding HJB equation can be solved analytically. We will use the MCA method to solve Merton's problem numerically and compare with the true analytical solution. In this section, we first introduce

the Merton's problem and show the procedure to solve it with MCA method, and then we show that MCA solution is closed to the true solution of HJB equation.

Merton's problem

The Merton's problem is described by the following diffusion process

$$\begin{cases} \mathrm{d}X_t = (a(\mu - r) + rX_t - c)\,dt + a\sigma dW_t \\ X_0 = x \end{cases}$$

with utility function $u(x) = x^p/p$, where a and c are control, (μ, r, p, σ) are known parameters. The value function is,

$$V^*(x) = \sup_{a \in \mathscr{A}(x)} E\left[\int_0^\infty e^{-\beta t} u(c) dt \middle| X_0 = x\right]$$

The optimal HJB equation for value function is

$$\beta V^*(x) = \sup_{a \in \mathbb{R}, c \geqslant 0} \left\{ \frac{1}{2} a^2 \sigma^2 (V^*)''(x) + (a\mu + (1-a)r - c)(V^*)'(x) + u(c) \right\} \tag{2.6}$$

And the HJB equation for value function is

$$\beta V^{a,c}(x) = \frac{1}{2} a^2 \sigma^2 (V^{a,c})''(x) + (a\mu + (1-a)r - c)(V^{a,c})'(x) + u(c) \tag{2.7}$$

There exists analytical solution to (2.7). The optimal control function a^* and c^* are

$$\begin{aligned} a^* &= \frac{u - r}{\sigma^2(1-p)} x \\ c^* &= \left[\frac{\beta - rp}{1-p} - \frac{p(\mu - r)^2}{2(1-p)^2\sigma^2} \right] x \end{aligned}$$

Optimal value function is

$$V^*(x) = \left(\frac{\beta - rp}{1-p} - \frac{p(b-r)^2}{2(1-p)^2\sigma^2} \right)^{(p-1)} x^p$$

Complete derivation see Pham (2009) section 3.6.2.

Standard MCA

Assume the time range of X_t is $(0, T)$, and the number of discretization step is N. Let step size $h = T/N$. We can apply finite difference method in (2.3) and split the operator and simplify the terms like $(a(\mu - r) + rX_t - c)^{\pm}$ as following. In Merton's problem, we know that r, x, c are non-negative, and $\mu > r$. Then $(a(\mu - r))^{\pm} = a^{\pm}(\mu - r)$, $c^+ = c$, $c^- = 0$, $(rx)^+ = rx$, $(rx)^- = 0$.

Plug terms into (2.4), X_t can be approximated by the Markov chain S_n with following transition probabilities,

$$\begin{aligned} p(s, s+h|a, c) &= \frac{(1/2)\sigma^2 a^2 + h(rs + a(\mu - r))}{Q^h(s, a, c)} \\ p(s, s-h|a, c) &= \frac{(1/2)\sigma^2 a^2 + hc}{Q^h(s, a, c)} \\ p(s, y|a, c) &= 0 \quad \text{for } y \notin \{s-h, s+h\} \end{aligned}$$

for $s \in \{h, 2h, \ldots, (N-1)h\}$, where,

$$Q^h(s, a, c) = \sigma^2 a^2 + h(rs + a(\mu - r) + c)$$

and interpolation interval:

$$\Delta t^h(s,a,c) = \frac{h^2}{Q^h(s,a,c)}$$

Value function of the Markov chain S_n is

$$(V^h)^*(s) = \sup_{a,c} E\left[\sum_{n=0}^{\infty} e^{-\beta t_n} u\left(c_{t_n} S_n\right) \Delta t^h(S_n, a_{t_n}, c_{t_n}) \middle| S_0 = s\right],$$

where $t_n = \sum_{i=0}^{n-1} \Delta t^h(S_i, a_{t_i}, c_{t_i})$. It can be obtained by solving the Bellman equation,

$$(V^h)^*(s) = \sup_{a,c}\left\{u(cs)\Delta t^h(s,a,c) + e^{-\beta \Delta t^h(s,a,c)} \sum_{s'} p^h(s,s'|a,c)(V^h)^*(s')\right\} \tag{2.8}$$

(2.8) can be solved with dynamic programming type of methods. In our simulation, we use policy iteration in below to solve (2.8).

Algorithm 1: Policy Iteration

Result: Optimal control (a, c) and optimal value V_i.

Randomly initialize with control (a, c);

Initialize V as the largest number possible;

Initialize err as the largest number possible;

Initialize e_tol as the tolerant of convergence;

while *err >e_tol* **do**

$V_{old} = V$;

Solve the system of equations

$V^h(s) = u(cs)\Delta t^h(s,a,c) + e^{-\beta\Delta t^h(s,a,c)} \sum_{s'} p^h(s,s'|a,c)V^h(s')$

and save its solution as a list V;

a, c = maximizer of $u(cs)\Delta t^h(x,a,c) + \mathrm{e}^{-\beta\Delta^h(x,a,c)} \sum_y p^h(x,y|a,c)V^h(y)$;

err = $\|V^{old} - V\|$;

end

When policy iteration converge, the control (a, c) and value vector V converge to optimal control (a^*, c^*) and value function $V^*(x)$ as $h \to 0$.

Simplified MCA

The maximization step in the while loop in policy iteration can be optimized analytically if we replace Q^h with constant $Q^{h*}(x) = \sup_{a,c} Q^h(x,a,c)$. We know that $\sup_{a,c} Q^h(x,a,c)$ exists if we assume a and c are bounded by M. It can be proved that the chain with $Q^{h*}(x)$is still local consistent. Then the transition probabilities

of the approximated Markov chain becomes:

$$\begin{aligned}
\tilde{p}^h(x, x+h|a, c) &= \frac{(1/2)\sigma^2 a^2 + h(rx + a(\mu - r))}{Q^{h*}(x)} \\
\tilde{p}^h(x, x-h|a, c) &= \frac{(1/2)\sigma^2 a^2 + hc}{Q^{h*}(x)} \\
\tilde{p}^h(x, x|a, c) &= 1 - \tilde{p}^h(x, x+h|a, c) - \tilde{p}^h(x, x-h|a, c) \\
&= \frac{Q^{h*}(x) - \sigma^2 a^2 - h(rx + a(\mu - r) + c)}{Q^{h*}(x)}
\end{aligned}$$

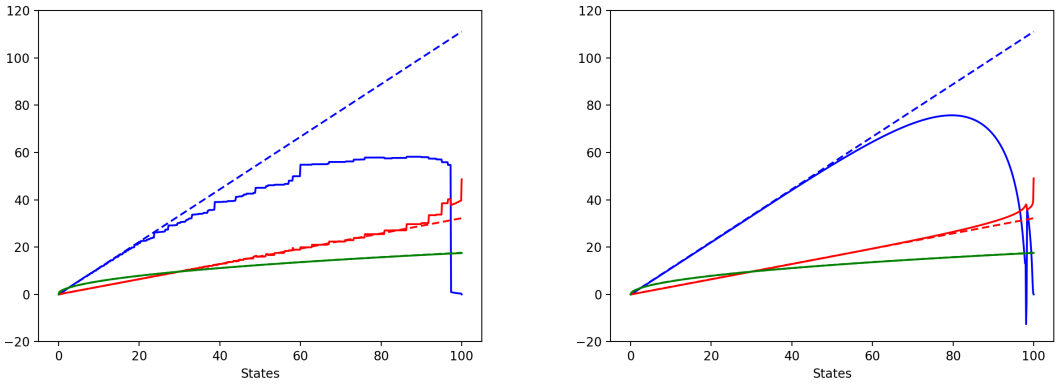
And there exists an closed form solution in the maximize step of policy iteration in step k as:

$$a_{k+1}(ih) = \min\left\{-\frac{b-r}{\sigma^2}\frac{[V_k^n((i+1)h) - V_k^n(ih)]h}{[V_k^h((i+1)h) - 2V_k^h(ih) + V_k^h((i-1)h)]^2}, Mih\right\}, \quad i = 1, 2, \ldots, N-1$$

$$c_{k+1}(ih) = \min\left\{(u')^{-1}\left(e^{-\beta h^2/Q^{h*}(ih)}\frac{V_k^n(ih) - V_k^n((i-1)h)}{h}\right), Mih\right\}, \quad i = 1, 2, \ldots, N$$

$$a_{k+1}(Nh) = 0$$

Simulation Results of MCA



(a) Standard MCA (b) Simplified MCA

Figure 2.1: Numerical results of merton's problem with $N = 1000$ and $T = 100$. In both figures. The dashed blue line is true a function, and the solid blue line is MCA approximated a function. The dashed red line is true c function, and the solid red line is MCA approximated c function. The dashed green line is the true value function, and the solid green line is MCA approximated value function.

The performance of Standard and Simplified MCA are presented in Figure 2.1. We can see the error increases as the state value getting closer to the artificial bound 100 in both Figure 2.1a and Figure 2.1b. That is because the original problem is defined on infinite horizon, and we introduced the artificial bound only for feasibility of computation. It is suggested that the artificial bound should be as large as possible so that it does not influence the accuracy in Kushner and Dupuis (1992).

Standard MCA is easy to implement for finite difference, but Figure 2.1a shows that the performance is worse than the simplified MCA, and it is not smooth. That is caused by the maximization step and its performance depends on the accuracy of optimization method used in the maximization step and is sensitive to different initial point. Figure 2.1a shows that the results of simplified MCA is closer to

the true optimal control and value function. It is also faster and more stable than Standard MCA because it doesn't rely on optimization. However, the tradeoff is that the simplified MCA chain is not always local consistent for all optimal control problem. And it requires extra caution to prove local consistency and calculation for closed form solution in maximization step.

3 CONTINUOUS TEMPORAL DIFFERENCE LEARNING FOR STOCHASTIC CONTINUOUS-TIME SYSTEM

We introduced continuous TD learning from Doya (2000) in Section 1.5. However, it is designed for deterministic continuous-time process and can't handle stochastic process. We know that there is almost always randomness in real world examples. It is necessary to generalize continuous TD learning for stochastic continuous-time reinforcement learning problem. In this chapter, we introduce the framework that we proposed to learn stochastic continuous-time process.

3.1 Preliminaries

Consider two continuous-time process starting with $X_0 = x$,

$$\text{Deterministic process: } X_t = x + \int_0^t b(X_s, \beta(X_s))ds \tag{3.1}$$

$$\text{Stochastic process: } X_t(\sigma) = x + \int_0^t b(X_s(\sigma), a(X_s(\sigma)))ds + \int_0^t \sigma(X_s(\sigma))dW_s \tag{3.2}$$

where W_t is a standard Brownian motion representing the stochastic term.

To properly define value function for discrete trajectories, we introduce step functions as following. For a trajectory U_{t_k}, define its corresponding continuous-time process as a step process

$$U_t = U_{t_0} 1\{t \leq t_0\} + \sum_{k=1}^{\infty} U_{t_k} 1\{t_{k-1} < t \leq t_k\},$$

where $1\{\cdot\}$ is the indicator function.

Denote the set of Lipschitz continuous function with Lipschitz constant M by

Lip_M. Assume we only consider control functions $\pi \in Lip_M$. We'd like to note that for any function Lipschitz function $h(\cdot,\cdot)$ with Lipschitz constant l,

$$\begin{aligned}|h(x,\pi(x)) - h(y,\pi(y))| \leq & l\left[|x-y| + |\pi(x)-\pi(y)|\right] \\ \leq & l(M+1)|x-y|\end{aligned}$$

We will have following **assumptions** for the theories in this chapter.

(A1) $\sigma(\cdot)$ is bounded.

(A2) $b(x,\alpha)$ is Lipschitz continuous with Lipschitz constant L, i.e. $|b(x,\alpha)-b(y,\tau)| \leq L(|x-y|+|\alpha-\tau|), \exists L > 0$.

(A3) Var $\left(\tilde{X}^h_{t_k}(\sigma)\right)$, Var $\left(b\left(\tilde{X}^h_{t_k}(\sigma), \pi(\tilde{X}^h_{t_k}(\sigma))\right)\right)$ are bounded, where $\pi \in Lip_M$. ($\tilde{X}^h_{t_k}(\sigma)$ is defined later in (3.10))

(A4) Corr $\left(\tilde{X}^h_{t_k}(\sigma), b(\check{X}_{t_k,i}(\sigma), \pi(\tilde{X}^h_{t_k}(\sigma)))\right) \to 0$ as $i \to \infty$, where $\pi \in Lip_M$. ($\check{X}_{t_k,i}(\sigma)$ is defined later in (3.10))

(A5) The reward function $r(x,\pi(x))$ is bounded by M_r and Lipschitz continuous with Lipschitz constant L_r.

3.2 Nonparametric smoothing based Policy evaluation and improvement methods

Theorem 3.1. *Suppose that $\sigma = \varepsilon\zeta$, where ζ could be a function of x. Under Assumptions (A1) - (A2), we have*

$$\sup_{0\leq t\leq T, \pi\in Lip_M} \mathbb{E}\left[|X_t(\varepsilon\zeta) - X_t(0)|^2 \mid X_t(\varepsilon\zeta) = x, X_t(0) = x\right] \leq \varepsilon^2 M_T^x$$

where $M_T^x = 2\left(g_\zeta(T,x) + 2(L(M+1))^2 T \int_0^T g_\zeta(s,x)\exp(2(L(M+1))^2 T(T-s))ds\right)$, *and* $g_\zeta(t,x) = \mathbb{E}\left[\int_0^t \zeta^2(X_s(\zeta))ds \mid X_0(\zeta) = x\right]$.

Proof of Theorem 3.1 is in Section 5.2. It indicates that as the diffusion coefficient goes to zero, the difference between diffusion process $X_t(\sigma)$ and deterministic process $X_t(0)$ converges in mean-square to zero. That is, $X_t(\sigma)$ and $X_t(0)$ have negligible difference when σ is small enough. This implies that for very small σ, the effect of random noise in SDE (3.2) is negligible, and stochastic process $X_t(\sigma)$ generated from SDE (3.2) is close to deterministic $X_t(0)$ generated from ODE (3.1). Thus, when σ is negligibly small, we may practically treat $X_t(\sigma)$ as $X_t(0)$, and the continuous policy evaluation and improvement developed for the deterministic model (3.1) can be effectively applied to the stochastic model (3.2).

However, σ in model (3.2) is often not negligibly small. Nonparametric smoothing comes to our rescue. We first apply kernel smoothing (Friedman et al. (2001)) to data observed from (3.2) and then apply the smoothed data to the continuous TD developed for the deterministic model 3.1. The proposed procedure is motivated from Theorem 3.1 and the fact that smoothing reduces the random effect and can make the smoothed $X_t(\sigma)$ close to $X_t(0)$.

To be specific, given current state X_t, assume we produce an action α_t and use the same action to generate m states from (3.2) and denote them by $\check{X}_{t_i}$, where $t_i \in [t, t+h]$, h is a bandwidth in the kernel smoothing, and $m = \#\{t_i : t \le t_i \le t+h\}$. Denote time step by Δ_t. Let $K(x)$ be a kernel function with support on $[0,1]$. We apply kernel smoothing to X_{t_i} to obtain a kernel estimator, $\tilde{X}_t^h$, of X_t, as illustrated in Figure 3.1. For each t, define

$$\overline{\Delta X_t^h} := \frac{1}{h}\sum_{t \le t_i \le t+h} K\left(\frac{t_i - t}{h}\right)\left(\check{X}_{t_i} - X_t\right) \tag{3.3}$$

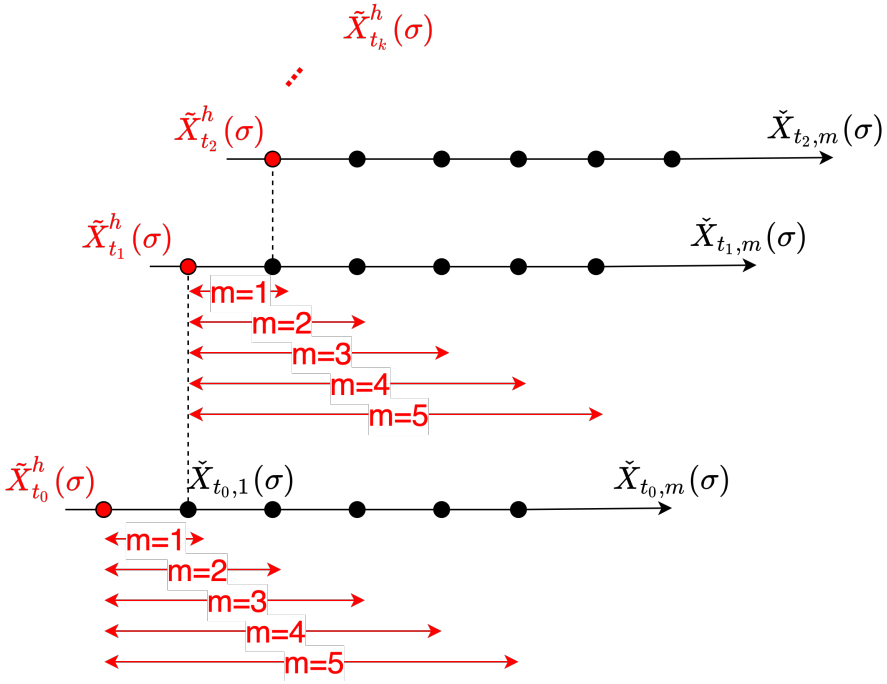


Figure 3.1: Window layout.

The new process $\tilde{X}^h_t$ is obtained by the following procedure: given the estimator $\tilde{X}^h_t$ at current step, we compute next step estimator $\tilde{X}^h_{t+\Delta t}$ by

$$\tilde{X}^h_{t+\Delta t} = \tilde{X}^h_t + \overline{\Delta X^h_t} \tag{3.4}$$

With the obtained smoothed process $\tilde{X}^h_t$, we can carry out any deterministic continuous-time reinforcement learning method to learn the optimal value function and control.

3.3 Implementation

First, let's review 4th order Runge-Kutta method that simulates solution of process,

$$\dot{X}_t = f(X_t, \alpha(X_t), t), \qquad t \in [0, T] \tag{3.5}$$

where f and α are appropriate functions and f can be either deterministic or stochastic. The 4th order Runge-Kutta method is stated as,

$$Y_{t_\ell} = Y_{t_{\ell-1}} + \frac{T}{6N}\left(\Upsilon_1^{\ell-1} + 2\Upsilon_2^{\ell-1} + 2\Upsilon_3^{\ell-1} + \Upsilon_4^{\ell-1}\right), \qquad \ell = 1, \ldots, N, \tag{3.6}$$

where Y_{t_0} is an initial value,

$$\begin{cases} \Upsilon_1^{\ell-1} = f\left(Y_{t_{\ell-1}}, \alpha(Y_t), t_{\ell-1}\right) \\ \Upsilon_2^{\ell-1} = f\left(Y_{t_{\ell-1}} + \Upsilon_1^{\ell-1}/2, \alpha(Y_t), t_{\ell-1} + T/(2N)\right) \\ \Upsilon_3^{\ell-1} = f\left(Y_{t_{\ell-1}} + \Upsilon_2^{\ell-1}/2, \alpha(Y_t), t_{\ell-1} + T/(2N)\right) \\ \Upsilon_4^{\ell-1} = f\left(Y_{t_{\ell-1}} + \Upsilon_3^{\ell-1}, \alpha(Y_t), t_{\ell-1} + T/N\right) \end{cases}$$

N denotes the number of iterations, and $t_\ell = \ell T/N$ with step size T/N. Thus, we can derive the 4th order Runge-Kutta method for deterministic and stochastic processes, respectively, with the same N and t_ℓ as above.

1. Deterministic process with $f_\sigma(x, \alpha(x), t) = b(x, \pi(x))$

$$X_{t_\ell}(0) = X_{t_{\ell-1}}(0) + \frac{T}{6N}\left(\Upsilon_{1,0}^{\ell-1} + 2\Upsilon_{2,0}^{\ell-1} + 2\Upsilon_{3,0}^{\ell-1} + \Upsilon_{4,0}^{\ell-1}\right) \tag{3.7}$$

where

$$\begin{cases} \Upsilon_{1,0}^{\ell-1} & = f\left(X_{t_{\ell-1}}(0), \pi(X_{t_{\ell-1}}(0)), t_{\ell-1}\right) \\ & = b(X_{t_{\ell-1}}(0), \pi(X_{t_{\ell-1}}(0))) \\ \Upsilon_{2,0}^{\ell-1} & = f\left(X_{t_{\ell-1}}(0) + \Upsilon_{1,0}^{\ell-1}/2, \pi(X_{t_{\ell-1}}(0)), t_{\ell-1} + T/(2N)\right) \\ & = b(X_{t_{\ell-1}}(0) + \Upsilon_{1,0}^{\ell-1}/2, \pi(X_{t_{\ell-1}}(0))) \\ \Upsilon_{3,0}^{\ell-1} & = f\left(X_{t_{\ell-1}}(0) + \Upsilon_{2,0}^{\ell-1}/2, \pi(X_{t_{\ell-1}}(0)), t_{\ell-1} + T/(2N)\right) \\ & = b(X_{t_{\ell-1}}(0) + \Upsilon_{2,0}^{\ell-1}/2, \pi(X_{t_{\ell-1}}(0))) \\ \Upsilon_{4,0}^{\ell-1} & = f\left(X_{t_{\ell-1}}(0) + \Upsilon_{3,0}^{\ell-1}, \pi(X_{t_{\ell-1}}(0)), t_{\ell-1} + T/N\right) \\ & = b(X_{t_{\ell-1}}(0) + \Upsilon_{3,0}^{\ell-1}, \pi(X_{t_{\ell-1}}(0))) \end{cases}$$

2. Stochastic process with $f_\sigma(x, \alpha(x), t) = b(x, \pi(x)) + \sigma\dot{W}(t)$, where $\dot{W}(t)$ is white noise (i.e. the derivative of Brownian motion Wt in the generalized function sense)

$$X_{t_\ell}(\sigma) = X_{t_{\ell-1}}(\sigma) + \frac{T}{6N}\left(\Upsilon_{1,\sigma}^{\ell-1} + 2\Upsilon_{2,\sigma}^{\ell-1} + 2\Upsilon_{3,\sigma}^{\ell-1} + \Upsilon_{4,\sigma}^{\ell-1}\right) \tag{3.8}$$

where

$$\begin{cases} \Upsilon_{1,\sigma}^{\ell-1} = f_\sigma\left(X_{t_{\ell-1}}(\sigma), a(X_{t_{\ell-1}}(\sigma)), t_{\ell-1}\right) \\ \qquad\quad = b(X_{t_{\ell-1}}(\sigma), a(X_{t_{\ell-1}}(\sigma))) + \sigma\check{W}(t_{\ell-1}) \\ \Upsilon_{2,\sigma}^{\ell-1} = f_\sigma\left(X_{t_{\ell-1}}(\sigma) + \Upsilon_{1,\sigma}^{\ell-1}/2, a(X_{t_{\ell-1}}(\sigma)), t_{\ell-1} + T/(2N)\right) \\ \qquad\quad = b(X_{t_{\ell-1}}(\sigma) + \Upsilon_{1,\sigma}^{\ell-1}/2, a(X_{t_{\ell-1}}(\sigma))) + \sigma\check{W}(t_{\ell-1} + T/(2N)) \\ \Upsilon_{3,\sigma}^{\ell-1} = f_\sigma\left(X_{t_{\ell-1}}(\sigma) + \Upsilon_{2,\sigma}^{\ell-1}/2, a(X_{t_{\ell-1}}(\sigma)), t_{\ell-1} + T/(2N)\right) \\ \qquad\quad = b(X_{t_{\ell-1}}(\sigma) + \Upsilon_{2,\sigma}^{\ell-1}/2, a(X_{t_{\ell-1}}(\sigma))) + \sigma\check{W}(t_{\ell-1} + T/(2N)) \\ \Upsilon_{4,\sigma}^{\ell-1} = f_\sigma\left(X_{t_{\ell-1}}(\sigma) + \Upsilon_{3,\sigma}^{\ell-1}, a(X_{t_{\ell-1}}(\sigma)), t_{\ell-1} + T/N\right) \\ \qquad\quad = b(X_{t_{\ell-1}}(\sigma) + \Upsilon_{3,\sigma}^{\ell-1}, a(X_{t_{\ell-1}}(\sigma))) + \sigma\check{W}(t_{\ell-1} + T/N) \end{cases} \tag{3.9}$$

and $\check{W}(\cdot)$ in (3.9) are discrete white noises corresponding to $\dot{W}$ and independent from $X_{t_{\ell-1}}(\sigma)$.

Applying kernel smoothing to $X_{t_k}(\sigma)$ and using (3.3) and (3.4), we obtain the smoothed process $\tilde{X}_t^h(\sigma)$ at discrete points $t_k = kT/N$, $k = 1, \ldots, N$, that is iteratively defined as follows,

$$\begin{aligned} \tilde{X}_{t_k}^h(\sigma) &= \tilde{X}_{t_{k-1}}^h(\sigma) + \overline{\Delta X_{t_{k-1}}^h(\sigma)} \\ &= \tilde{X}_{t_{k-1}}^h(\sigma) + \frac{1}{h}\sum_{j=1}^{m} K\left(\frac{t_{k-1+j} - t}{h}\right)\left[\check{X}_{t_{k-1},j}(\sigma) - \tilde{X}_{t_{k-1}}^h(\sigma)\right], \end{aligned} \tag{3.10}$$

where $\check{X}_{t_{k-1},j}(\sigma), j = 1, 2, \ldots, m$, are assumed to be generated from the same action $\alpha_{t_{k-1}} = \pi(\tilde{X}^h_{t_{k-1}}(\sigma))$ by (3.8) and (3.9). Specifically, we have

$$\begin{cases} \check{X}_{t_{k-1},1}(\sigma) = \tilde{X}^h_{t_{k-1}}(\sigma) + \dfrac{T}{6N}\left(\Upsilon^{k-1,0}_{1,\sigma} + 2\Upsilon^{k-1,0}_{2,\sigma} + 2\Upsilon^{k-1,0}_{3,\sigma} + \Upsilon^{k-1,0}_{4,\sigma}\right) \\ \check{X}_{t_{k-1},2}(\sigma) = \check{X}_{t_k,1}(\sigma) + \dfrac{T}{6N}\left(\Upsilon^{k-1,1}_{1,\sigma} + 2\Upsilon^{k-1,1}_{2,\sigma} + 2\Upsilon^{k-1,1}_{3,\sigma} + \Upsilon^{k-1,1}_{4,\sigma}\right) \\ \vdots \\ \tilde{X}_{t_{k-1},m}(\sigma) = \check{X}_{t_{k-1},m-1}(\sigma) \\ \quad + \dfrac{T}{6N}\left(\Upsilon^{k-1,m-1}_{1,\sigma} + 2\Upsilon^{k-1,m-1}_{2,\sigma} + 2\Upsilon^{k-1,m-1}_{3,\sigma} + \Upsilon^{k-1,m-1}_{4,\sigma}\right) \end{cases} \tag{3.11}$$

where,

$$\begin{cases} \Upsilon^{k-1,i}_{1,\sigma} = & b\left(\check{X}_{t_{k-1},i}(\sigma), \pi\left(\tilde{X}^h_{t_{k-1}}(\sigma)\right)\right) + \sigma\left(\check{X}_{t_{k-1},i}(\sigma), \pi\left(\tilde{X}^h_{t_{k-1}}(\sigma)\right)\right)\check{W}_i\left(t_{k-1}\right) \\ \Upsilon^{k-1,i}_{2,\sigma} = & b\left(\check{X}_{t_{k-1},i}(\sigma) + \Upsilon^{k-1,i}_{1,\sigma}/2, \pi\left(\tilde{X}^h_{t_{k-1}}(\sigma)\right)\right) \\ & + \sigma\left(\check{X}_{t_{k-1},i}(\sigma) + \Upsilon^{k-1,i}_{1,\sigma}/2, \pi\left(\tilde{X}^h_{t_{k-1}}(\sigma)\right)\right)\check{W}_i\left(t_{k-1} + T/(2N)\right) \\ \Upsilon^{k-1,i}_{3,\sigma} = & b\left(\check{X}_{t_{k-1},i}(\sigma) + \Upsilon^{k-1,i}_{2,\sigma}/2, \pi\left(\tilde{X}^h_{t_{k-1}}(\sigma)\right)\right) \\ & + \sigma\left(\check{X}_{t_{k-1},i}(\sigma) + \Upsilon^{k-1,i}_{2,\sigma}/2, \pi\left(\tilde{X}^h_{t_{k-1}}(\sigma)\right)\right)\check{W}_i\left(t_{k-1} + T/(2N)\right) \\ \Upsilon^{k-1,i}_{4,\sigma} = & b\left(\check{X}_{t_{k-1},i}(\sigma) + \Upsilon^{k-1,i}_{3,\sigma}, \pi\left(\tilde{X}^h_{t_{k-1}}(\sigma)\right)\right) \\ & + \sigma\left(\check{X}_{t_{k-1},i}(\sigma) + \Upsilon^{k-1,i}_{3,\sigma}, \pi\left(\tilde{X}^h_{t_{k-1}}(\sigma)\right)\right)\check{W}_i\left(t_{k-1} + T/N\right) \end{cases} \tag{3.12}$$

for $i = 0, 1, \ldots, m-1$ and $\check{W}(\cdot)$ in (3.9) are discrete white noises corresponding to $\dot{W}$ and independent from $\tilde{X}^h_{t_{k-1}}(\sigma)$. In particular, if K is chosen to be a constant

kernel function, we obtain,

$$\begin{aligned}\tilde{X}^h_{t_k}(\sigma) &= \tilde{X}^h_{t_{k-1}}(\sigma) + \frac{1}{m}\left[\check{X}_{t_{k-1},m}(\sigma) - \tilde{X}^h_{\sigma}\left(t_{k-1}\right)\right] \\ &= \tilde{X}^h_{t_{k-1}}(\sigma) + \frac{T}{6mN}\sum_{i=0}^{m-1}\left(\Upsilon^{k-1,i}_{1,\sigma} + 2\Upsilon^{k-1,i}_{2,\sigma} + 2\Upsilon^{k-1,i}_{3,\sigma} + \Upsilon^{k-1,i}_{4,\sigma}\right)\end{aligned} \tag{3.13}$$

Notations

Denote the continuous-time process and corresponding optimal value function as following,

1. $V^{\pi}_0(x)$ as the value function for $X_t(0)$
   (i.e. $V^{\pi}_0(x) = \int_0^{\infty} e^{-\beta t} r(X_t(0), \pi(X_t(0)))dt \Big| X_0(0) = x$)
2. $V^{\pi}_{\sigma}(x)$ as the value function for $X_t(\sigma)$
   (i.e. $V^{\pi}_{\sigma}(x) = E\left[\int_0^{\infty} e^{-\beta t} r(X_t(\sigma), \pi(X_t(\sigma))dt \Big| X_0(\sigma) = x\right]$)
3. $V^{*,M}_0(x)$ as optimal value function among control $\pi \in Lip_M$ for $X_t(0)$
4. $V^{*,M}_{\sigma}(x)$ as optimal value function among control $\pi \in Lip_M$ for $X_t(\sigma)$
5. $X_{t_k}(\sigma)$ as the trajectory sampled from $X_t(\sigma)$
6. $\check{X}_t(\sigma)$ and $\tilde{X}^h_t(\sigma)$ as continuous-time step process of $\check{X}_{t_k}(\sigma)$ and $\tilde{X}^h_{t_k}(\sigma)$, respectively.
7. $\tilde{V}^{0,\pi}_0(x) = \int_0^{\infty} e^{-\beta t} r\left(\check{X}_t(0), \pi\left(\check{X}_t(0)\right)\right) dt \mid \check{X}_0(0) = x$ as the value function of $\check{X}_{t_k}(0)$.
8. $\tilde{V}^{h,\pi}_{\sigma}(x) = \mathbb{E}\left[\int_0^{\infty} e^{-\beta t} r\left(\tilde{X}^h_t(\sigma), \pi\left(\tilde{X}^h_t(\sigma)\right)\right) dt \mid \tilde{X}^h_0(\sigma) = x\right]$ as the value function of $\tilde{X}^h_{t_k}(\sigma)$.

9. $\tilde{V}_{\sigma}^{h,*,M}(x) = \sup_{\pi \in Lip_M} \tilde{V}_{\sigma}^{h,\pi}(x)$ and $\tilde{V}_{0}^{0,*,M}(x) = \sup_{\pi \in Lip_M} \tilde{V}_{0}^{0,\pi}(x)$ as the optimal value function with policy restricted in Lip_M

	Deterministic	Stochastic
Continuous	$X_t(0)$	$X_t(\sigma)$
	$\check{X}_t(0)$	$\check{X}_t^h(\sigma)$
	$\tilde{X}_t^h(0)$	$\tilde{X}_t^h(\sigma)$
Discrete	$X_{t_k}(0)$	$X_{t_k}(\sigma)$
	$\check{X}_{t_k}(0)$	$\check{X}_{t_k}^h(\sigma)$
	$\tilde{X}_{t_k}^h(0)$	$\tilde{X}_{t_k}^h(\sigma)$

Table 3.1: Processes notations

Denote the optimal control in Lip_M as following,

1. $\pi_0^{*,M}(x)$ as optimal control function for X_t, i.e. $\pi_0^{*,M}(x) = \arg\max_{\pi \in Lip_M} V_0^{\pi}(x)$
2. $\pi_{\sigma}^{*,M}(x)$ as optimal control function for $X_t(\sigma)$, i.e. $\pi_{\sigma}^{*,M}(x) = \arg\max_{\pi \in Lip_M} V_{\sigma}^{\pi}(x)$
3. $\tilde{\pi}_{\sigma}^{h,*,M}(x)$ as optimal control function for $\tilde{X}_{t_k}^h(\sigma)$, i.e. $\tilde{\pi}_{\sigma}^{h,*,M}(x) = \arg\max_{\pi \in Lip_M} \tilde{V}_{\sigma}^{h,\pi}(x)$

	Deterministic			Stochastic		
	Control π	Optimal Value in Lip_M	Optimal Control in Lip_M	Control π	Optimal Value in Lip_M	Optimal Control in Lip_M
Continuous	$V_0^{\pi}(x)$	$V_0^{*,M}(x)$	$\pi_0^{*,M}(x)$	$V_{\sigma}^{\pi}(x)$	$V_{\sigma}^{*,M}(x)$	$\pi_{\sigma}^{*,M}(x)$
Discrete	$\tilde{V}_0^{0,\pi}(x)$			$\tilde{V}_{\sigma}^{h,\pi}(x)$	$\tilde{V}_{\sigma}^{h,*,M}(x)$	$\tilde{\pi}_{\sigma}^{h,*,M}(x)$

Table 3.2: Value and control functions

The value-gradient based policy algorithm

In simulation, we use the value-gradient based method from Doya (2000) to learn the value function. Assume the system drift $b(x, \alpha)$ is linear with respect to the action $\alpha = (\alpha_1, \ldots, \alpha_d) \in \mathcal{A}$, and the reward function can be written in the form as,

$$r(x, \alpha) = R(x) - \sum_{j=1}^{d} S_j(a_j),$$

where $S = (S_1, \ldots, S_d)$ is the penalty function for actions. We use an approximation function $V(x, w)$ to approximate the value function, where w is the parameter to learn. Then for state x, with HJB optimality equation, the greedy action is taken as:

$$\begin{aligned}\pi(x) &= \arg\max_{a \in \mathcal{A}} \left[r(x, \alpha) + \frac{\partial V(x, w)}{\partial x} b(x, \alpha) \right] \\ &= S'^{-1} \left(\frac{\partial b(x, \alpha)^T}{\partial \alpha} \frac{\partial V(x, w)^T}{\partial x} \right)\end{aligned}$$

A small perturbation $\sigma_n n(t)$ is added into the greedy action to increase exploration. Let the noise process $\dot{n}(t) = -n(t) + N(t)$, where $N(t)$ denotes normal gaussian noise. Let the scaling factor $\sigma_n = \sigma_0 \min\left[1, \max\left[0, \frac{V_{\max} - V(X_t, w)}{V_{\max} - V_{\min}}\right]\right]$, where $V_{\min}$ and $V_{\max}$ are the minimal and maximal levels of the value function. Finally, we have the policy taken as is:

$$\pi(x) = S'^{-1} \left(\frac{\partial b(x, \alpha)^T}{\partial \alpha} \frac{\partial V(x, w)^T}{\partial x} + \sigma_n n \right) \tag{3.14}$$

Algorithm 2: The value-gradient based policy algorithm

Result: Approximated value function $V(x, w)$ with weight w to learn.

initialize state X, generate action a from (3.14) ;

Initialize weight w randomly ;

Initialize eligibility trace $e = 0$;

Initialize learning rate η and time constant κ ;

Initialize e_tol as the tolerant of convergence;

Initialize err as the largest number possible;

Initialize Δt as the desired step size and k as the window step size;

while *err* > *e_tol* **do**

$w_{old} = w$;

Use action a to generate next m states $X_{\Delta(t)}, \ldots, X_{m\Delta(t)}$;

Calculate $\tilde{X}^h_{t+\Delta t}$ as (3.4). Calculate reward $R = r(\tilde{X}^h_{t+\Delta t}, a)$;

Calculate continuous TD error with $(X, \tilde{X}^h_{t+\Delta t}, R)$ as,

$\delta = R + \frac{V(X,w) - V(X^h_{t+\Delta t}, w)}{\Delta t} - \beta V(X, w)$;

Updated eligibility trace $e = e + \left(-\frac{1}{\kappa}e + \frac{\partial V(X,w)}{\partial w}\right)\Delta t$;

update weight $w = w + \eta\delta e$;

err = $\|w - w_{old}\|$;

$X = \tilde{X}^h_{t+\Delta t}$;

end

3.4 Asymptotic theory

In this section, we will prove the convergence of the optimal value function and control function of the discrete processes of (3.1) and (3.2) simulated with the 4th order Runge-Kutta method. We will also prove the convergence of the process from

nonparametric smoothing based method assuming we use constant kernel function in (3.13) and its value and control functions.

Theorem 3.2. *Under assumptions (A1) - (A4), we have that as* $T/N \to 0$, $h \to 0$ *and* $m = h/(T/N) \to \infty$,

$$\sup_{1\le k\le N} \mathbb{E}\left[\tilde{X}^h_{t_k}(\sigma) - X_{t_k}(0)\right]^2 = O\left(h^2(T/N) + \sigma^2(T/N)^2/h\right)$$

See Section 5.3 for proof of Theorem 3.2. It indicates that the difference between $\tilde{X}^h_{t_k}(\sigma)$ and $X(t_k)$ converges in mean square to 0.

Value Function Uniform Convergence

Theorem 3.3. *Suppose that* $\sigma = \varepsilon\zeta$. *Under assumption (A1) - (A5),*

$$V^\pi_{\varepsilon\zeta}(x) \to V^\pi_0(x)$$

uniformly $\forall \pi \in Lip_M$ *as* $\varepsilon \to 0$.

Proof see Section 5.4.

Corollary 3.4. *Under assumption (A1) - (A5),* $V^{*,M}_{\varepsilon\zeta}(x) \to V^{*,M}_0(x)$ *as* $\varepsilon \to 0$.

Proof. Given $V^\pi_{\varepsilon\zeta}(x) \to V^\pi_0(x)$ uniformly $\forall \pi \in Lip_M$ from Proposition 3.3, We have,

$$\lim_{\varepsilon\to 0} \sup_{\pi\in Lip_M} |V^\pi_{\varepsilon\zeta}(x) - V^\pi_0(x)| = 0$$

If $V_{\varepsilon\zeta}^{*,M}(x) \geq V^{*,M}(x)$,

$$\begin{aligned}\left|V_{\varepsilon\zeta}^{*,M}(x) - V_0^{*,M}(x)\right| &= \sup_{\pi\in Lip_M} V_{\varepsilon\zeta}^{\pi}(x) - \sup_{\pi\in Lip_M} V_0^{\pi}(x) \\ &\leq \sup_{\pi\in Lip_M} \left(V_{\varepsilon\zeta}^{\pi}(x) - V_0^{\pi}(x)\right) \\ &\leq \sup_{\pi\in Lip_M} \left|\left(V_{\varepsilon\zeta}^{\pi}(x) - V_0^{\pi}(x)\right)\right| \\ &\to 0\end{aligned}$$

and we have the same argument if $V_{\varepsilon\zeta}^{*,M}(x) < V^{*,M}(x)$.
Thus, we have $\left|V_{\varepsilon\zeta}^{*,M}(x) - V_0^{*,M}(x)\right| \to 0$ as $\varepsilon \to 0$. □

Optimal Control Convergence

Lemma 3.5. *Under assumption (A5), if* $\frac{\partial g_\theta(t,x)}{\partial x}$ *exists and* $\exists \mu > 0$, *s.t.* $\sup_{\theta\in[0,\mu],t\geq 0} \frac{\partial g_\theta(t,x)}{\partial x} < \infty$, *we have* $\frac{\partial V_{\varepsilon\zeta}^{\pi}(x)}{\partial x} \to \frac{\partial V_0^{\pi}(x)}{\partial x}$ *uniformly* $\forall \pi \in Lip_M$ *as* $\varepsilon \to 0$.

Proof see in Section 5.5.

Lemma 3.6. *Under assumption (A1) - (A5),* $\frac{\partial V_{\varepsilon\zeta}^{*,M}(x)}{\partial x} \to \frac{\partial V_0^{*,M}(x)}{\partial x}$ *as* $\xi \to 0$.

Proof is similar to Corollary 3.4.

Theorem 3.7. *Assume the system drift* $b(x,\alpha)$ *is linear with respect to the action* α, *and the reward function can be written in the form as* $r(x,\alpha) = R(x) - \sum_{j=1}^{m} S_j(\alpha_j)$. *Denote* $S = (S_1, \ldots S_m)^T$. *Under assumption (A1) - (A5), suppose there exist the optimal controls* $\pi_0^{*,M}(x)$ *and* $\pi_\sigma^{*,M}(x)$ *measurable, then* $\pi_{\varepsilon\zeta}^{*,M}(x) \to \pi_0^{*,M}(x)$ *as* $\varepsilon \to 0$.

Proof. The optimal controls satisfy the following equation,

$$\begin{aligned}
\pi_0^{*,M}(x) &= \arg\max_{\pi \in Lip_M} \left\{ r(x,\pi(x)) + b(x,\pi(x))^T \frac{\partial V_0^{*,M}(x)}{\partial x} \right\} \\
&= S'^{-1} \left(\frac{\partial b(x,a)}{\partial a}^T \frac{\partial V_0^{*,M}(x)}{\partial x}^T \right) \\
\pi_\sigma^{*,M}(x) &= \arg\max_{\pi \in Lip_M} \left\{ r(x,\pi(x)) + b(x,\pi(x))^T \frac{\partial V_\sigma^{*,M}(x)}{\partial x} + \frac{1}{2} tr \left(\sigma(x)\sigma^T(x) \frac{\partial^2 V_\sigma^{*,M}(x)}{\partial x^2} \right) \right\} \\
&= \arg\max_{\pi \in Lip_M} \left\{ r(x,\pi(x)) + b(x,\pi(x))^T \frac{\partial V_\sigma^{*,M}(x)}{\partial x} \right\} \\
&= S'^{-1} \left(\frac{\partial b(x,a)}{\partial a}^T \frac{\partial V_\sigma^{*,M}(x)}{\partial x}^T \right)
\end{aligned}$$

With lemma 3.6, we have $\pi_{\varepsilon\sigma}^{*,M}(x) \to \pi_0^{*,M}(x)$ as $\varepsilon \to 0$. □

Window Method Value Function Convergence

Lemma 3.8. *Under assumptions (A1) - (A5), we have that as $T/N \to 0$, $h \to 0$ and $m = h/(T/N) \to \infty$,*

$$|\tilde{V}_\sigma^{h,\pi}(x) - \tilde{V}_0^{0,\pi}(x)| \to 0,$$

uniformly for all $\pi \in Lip_M$.

Proof see Section 5.6

Lemma 3.9. *Under assumptions (A1) - (A5), we have that as $T/N \to 0$, $h \to 0$ and $m = h/(T/N) \to \infty$,*

$$|V_0^{\pi}(x) - \tilde{V}_0^{0,\pi}(x)| \to 0,$$

uniformly for all $\pi \in Lip_M$.

Proof see Section 5.7

Theorem 3.10. *Under assumptions (A1) - (A5), we have that as* $T/N \to 0$, $h \to 0$ *and* $m = h/(T/N) \to \infty$,

$$|\tilde{V}_\sigma^{h,\pi}(x) - V_0^\pi(x)| \to 0,$$

uniformly for all $\pi \in Lip_M$.

Proof. The proof is immediate from Lemma 3.8 and Lemma 3.9.

□

Corollary 3.11. *Under assumptions (A1) - (A5), we have that as* $T/N \to 0$, $h \to 0$ *and* $h/(T/N) \to \infty$,

$$\tilde{V}_\sigma^{h,*,M}(x) \to V_0^{*,M}(x),$$

Proof is similar to Corollary 3.4.

Window Method Optimal Control Convergence

Lemma 3.12. *Under assumptions (A1) - (A5), if* $\frac{\partial g_\theta(t,x)}{\partial x}$ *exists and* $\exists \mu > 0$, *s.t.* $\sup_{\theta\in[0,\mu],t\geq 0} \frac{\partial g_\theta(t,x)}{\partial x} < \infty$, *we have that as* $T/N \to 0$, $h \to 0$ *and* $h/(T/N) \to \infty$, $\frac{\partial \tilde{V}_\sigma^{h,\pi}(x)}{\partial x} \to \frac{\partial V_0^\pi(x)}{\partial x}$ *uniformly* $\forall \pi \in Lip_M$.

Proof see Section 5.8

Lemma 3.13. *Under assumptions (A1) - (A5), we have that as* $T/N \to 0$, $h \to 0$ *and* $h/(T/N) \to \infty$, $\frac{\partial \tilde{V}_\sigma^{h,*,M}(x)}{\partial x} \to \frac{\partial V_0^{*,M}(x)}{\partial x}$.

Proof is similar to Corollary 3.4.

Theorem 3.14. *Under the same assumptions as Theorem 3.7, suppose there exist optimal controls* $\tilde{\pi}^{0,*,M}(x)$ *and* $\pi_\sigma^{*,M}(x)$ *measurable. we have that as* $T/N \to 0$, $h \to 0$ *and* $h/(T/N) \to \infty$, $\tilde{\pi}_\sigma^{h,*,M}(x) \to \pi_0^{*,M}(x)$

Proof. The result is immediate from Lemma 3.13. Proof is similar to Theorem 3.7. □

4 NUMERICAL STUDIES

The performance of TD based window smoothing method is tested on the problem of pendulum swing-up with limited torque Doya (2000).

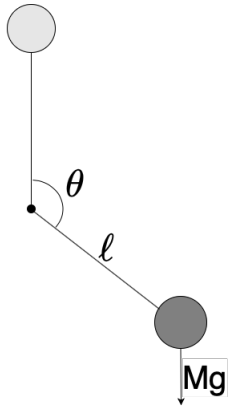


Figure 4.1: Pendulum

4.1 Pendulum Problem

The state variable of pendulum can be considered as $x = (\theta, \omega)$ satisfying stochastic PDE,

$$\begin{cases} d\theta = \omega dt \\ d\omega = \frac{1}{Ml^2}(-\mu\omega + Mgl\sin\theta + \alpha)dt + \sigma dW_t \end{cases} \tag{4.1}$$

where $M = l = 1,\ g = 9.8,\ \alpha = 0.01$. α denotes the control variable and $\alpha^{max} = 5.0$.

The process is simulated by fourth-order Runge-Kutta method, In the simulation, consider $T = 20$ and $N = 10000$.

We adopt the reward definition in Doya (2000): Each trial starts from initial state $x_0 = (\theta_0, 0)$, where θ_0 is randomly selected in $[-\pi, \pi]$. A trial lasts for 20 seconds

T	N	σ	m	Method
20	10000	5, 7, 9, 11	1	continuous TD
			2	nonparametric smoothing based continuous TD
			3	
			4	
			5	

Table 4.1: Experiments settings with 4th order Runge-Kutta method.

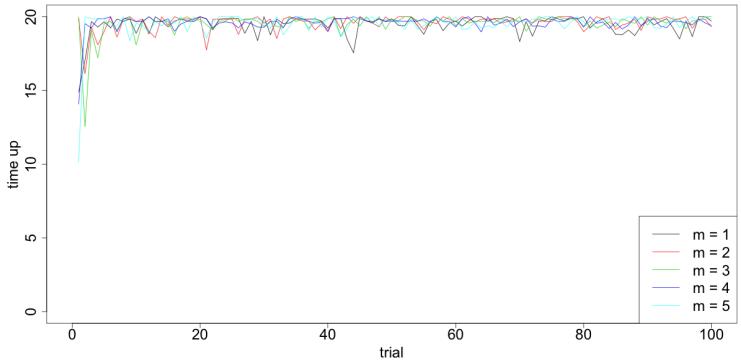
unless the pendulum was over-rotated ($|\theta| > 5\pi$). If a trail fails because of over rotation, the trial will be terminated with a reward $r(t) = -1$ for 1 second. As a measure of the swing-up performance, we defined the up position the pendulum staying up with $(|\theta| < \pi/4)$, and let the time of the pendulum stays in the up position as t_{up}. The reward function is defined as,

$$\begin{aligned} r(x,u) =& R(\mathbf{x}) - \sum_{j=1}^{m} S_j\left(u_j\right) \\ =& \cos(\theta) - cu_{\max}(\frac{2}{\pi})^2 \log(\cos(\frac{\pi u}{2u_{\max}})) \end{aligned}$$

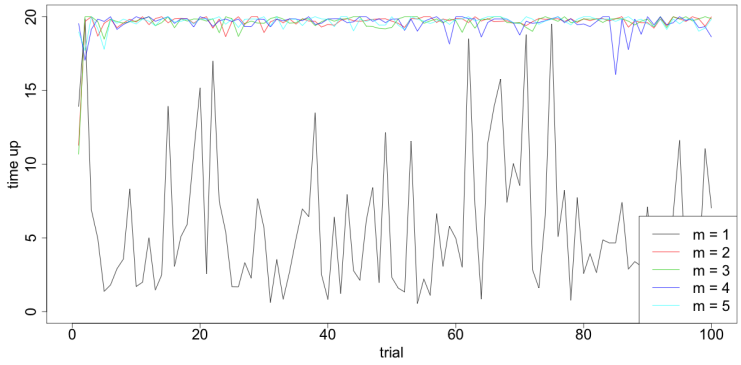
4.2 Simulation Results

We apply kernel smoothing framework with continuous TD(λ) algorithm on stochastic pendulum processes with parameter set up in Table 4.1. In each testing experiment, 100 trials of 20-second stochastic pendulum run were used to train. The scale of noise process added in (3.14) is $\sigma_0 = 0.5$. The results are shown in Figure 4.2. We can see that as the variance σ increases, the performance of method with small window size is becoming worse. When $\sigma = 5$ (Figure 4.2a), all methods including deterministic continuous TD(λ) (i.e. when m = 1) are stable and can maintain the pendulum at up position. When $\sigma = 7$ (Figure 4.2b), the deterministic continuous TD(λ) method cannot learn the optimal control function correctly and

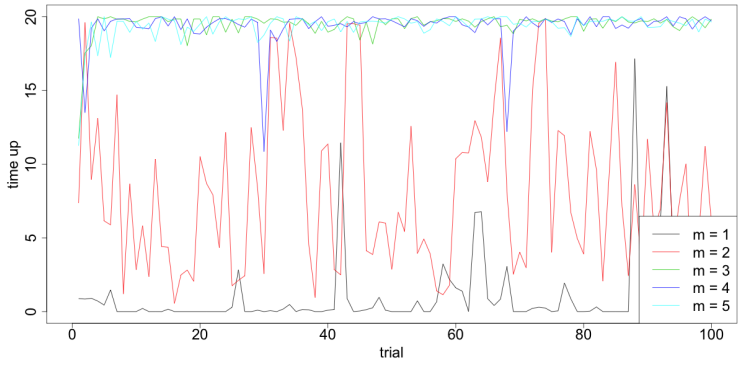
fails to maintain the pendulum in up position. When we further increase σ to 9, (Figure 4.2c), the smoothing method with smallest window size ($m = 2$) also starts to fail. And when $\sigma = 11$, only smoothing method with largest window size in our experiment ($m = 5$) is able to learning the optimal control and maintain the pendulum at up position.



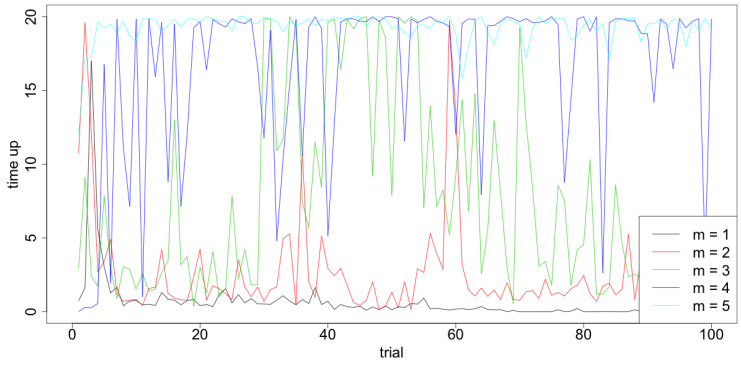
(a) $\sigma = 5$



(b) $\sigma = 7$



(c) $\sigma = 9$



(d) $\sigma = 11$

Figure 4.2: Comparison of performance with different variance σ.

5 PROOFS

5.1 Proof of (1.5) - HJB equation

Proof. By definition,

$$\begin{aligned} V^\pi(x) &= \mathbb{E}\left[\int_0^\infty e^{-\beta t} r(X_t, \pi(X_t))dt \mid x_0 = x\right] \\ &= \mathbb{E}\left[\int_h^\infty e^{-\beta t} r(X_t, \pi(X_t))dt + \int_0^h e^{-\beta t} r(X_t, \pi(X_t))dt \mid x_0 = x\right] \\ &= \mathbb{E}\left[e^{-\beta h} V^\pi(X_h) + \int_0^h e^{-\beta t} r(X_t, \pi(X_t))dt \mid X_0 = x\right] \end{aligned}$$

Apply Ito's formula on $V^\pi(x(h))$, we have

$$\begin{aligned} V^\pi(X_h) = V^\pi(X_0) \\ + \int_0^h b(X_t, \pi(X_t)) \frac{\partial V^\pi}{\partial x}(X_t) + \frac{1}{2} tr(\sigma(X_t, \pi(X_t))\sigma'(X_t, \pi(X_t)) \frac{\partial^2 V^\pi}{\partial x^2}(X_t))dt \\ + \int_0^h \sigma(X_t, \pi(X_t))' \frac{\partial V^\pi}{\partial x}(X_t) dW_s \end{aligned}$$

Denote $\mathcal{L}^\pi(V^\pi) = b(X_t, \pi(X_t)) \frac{\partial V^\pi}{\partial x}(X_t) + \frac{1}{2} tr(\sigma(X_t, \pi(X_t))\sigma'(X_t, \pi(X_t)) \frac{\partial^2 V^\pi}{\partial x^2}(X_t)$.

Plug $V^\pi(x(h))$ back into the first equation, we have,

$$\begin{aligned} V^\pi(x) &= \mathbb{E}\Big[e^{-\beta h}\left(V^\pi(X_0) + \int_0^h \mathcal{L}^\pi(V^\pi)dt + \int_0^h \sigma(X_t, \pi(X_t))' \frac{\partial V^\pi}{\partial x}(X_t) dW_s\right) \\ &\quad + \int_0^h e^{-\beta t} r(X_t, \pi(X_t))dt \mid X_0 = x\Big] \\ &= e^{-\beta h} V^\pi(x) + \mathbb{E}\Big[\int_0^h e^{-\beta h} \mathcal{L}^\pi(V^\pi)dt + e^{-\beta t} r(X_t, \pi(X_t))dt \mid X_0 = x\Big] \end{aligned}$$

Move $e^{-\beta h}V^\pi(x)$ to LHS and divide both side of equation by h, we have,

$$\frac{1-e^{-\beta h}}{h}V^\pi(x) = \frac{\mathbb{E}\Big[\int_0^h e^{-\beta h}\mathcal{L}^\pi(V^\pi)dt + e^{-\beta t}r(X_t,\pi(X_t))dt \mid X_0 = x\Big]}{h}$$

Let $h \to 0$, we have

$$\begin{aligned}\beta V^\pi(x) &= \mathbb{E}\Big[\mathcal{L}^\pi V^\pi(X_0)dt + r(X_0,\pi(X_0)) \mid X_0 = x\Big] \\ &= \mathcal{L}^\pi V^\pi(x)dt + r(x,\pi(x))\end{aligned}$$

□

5.2 Proof of Theorem 3.1

Lemma 5.1 (Grönwall's lemma). *Let α, β and u be real-valued functions defined on I. Assume that β and u are continuous and that the negative part of α is integrable on every closed and bounded subinterval of I. If β is non-negative and if u satisfies the integral inequality, $u(t) \le \alpha(t) + \int_a^t \beta(s)u(s)\mathrm{d}s,\ \forall t \in I$, then*

$$u(t) \le \alpha(t) + \int_a^t \alpha(s)\beta(s)\exp\left(\int_s^t \beta(r)\mathrm{d}r\right)\mathrm{d}s, \quad t \in I.$$

The proof of Grönwall's lemma is omitted.

Proof of Theorem 3.1

Proof. $\forall \pi \in Lip_M$,

$$\left[\int_0^t b(X_s(\varepsilon\zeta), \pi(X_s(\varepsilon\zeta))) - b(X_s(0), \pi(X_s(0)))ds\right]^2 \leq \left[\int_0^t L(M+1)\Big|X_s(\varepsilon\zeta) - X_s(0)\Big|ds\right]^2$$
$$\leq (L(M+1))^2 t \int_0^t \Big|X_s(\varepsilon\zeta) - X_s(0)\Big|^2 ds$$

The second inequality comes from Cauchy-Schwarz inequality. For simplicity, denote $\mathbb{E}^x\left[f(X_t(0), X_t(\sigma)\right] = \mathbb{E}\left[f(X_t(0), X_t(\sigma)\Big|X_0(0) = x, X_0(\sigma) = x\right]$, for any function f.

$$\begin{aligned}
&\mathbb{E}^x|X_t(\varepsilon\zeta) - X_t(0)|^2 \\
=&\mathbb{E}^x\left[\int_0^t b(X_s(\varepsilon\zeta), \pi(X_s(\varepsilon\zeta))) - b(X_s(0), \pi(X_s(0)))ds + \varepsilon\int_0^t \zeta(X_t(\varepsilon\zeta))dW_t\right]^2 \\
\leq& 2\mathbb{E}^x\left[\Big|\int_0^t b(X_s(\varepsilon\zeta), \pi(X_s(\varepsilon\zeta))) - b(X_s(0), \pi(X_s(0)))ds\Big|^2 + \varepsilon^2\left(\int_0^t \zeta(X_s(\varepsilon\zeta))dW_s\right)^2\right] \\
=& 2\mathbb{E}^x\left[\Big|\int_0^t b(X_s(\varepsilon\zeta), \pi(X_s(\varepsilon\zeta))) - b(X_s(0), \pi(X_s(0)))ds\Big|^2\right] \\
&+ 2\varepsilon^2\mathbb{E}^x\left[\int_0^t \zeta^2(X_s(\varepsilon\zeta))ds\right] \\
\leq& 2\mathbb{E}^x\left[(L(M+1))^2 t\int_0^t \Big|X_s(\varepsilon\zeta) - X_s(0)\Big|^2 ds\right] + 2\varepsilon^2\mathbb{E}^x\left[\int_0^t \zeta^2(X_s(\varepsilon\zeta))ds\right] \\
=& 2(L(M+1))^2 t\int_0^t \mathbb{E}^x\Big|X_s(\varepsilon\zeta) - X_s(0)\Big|^2 ds + 2\varepsilon^2\mathbb{E}^x\left[\int_0^t \zeta^2(X_s(\varepsilon\zeta))ds\right]
\end{aligned}$$

Denote $g_\zeta(t,x) := \mathbb{E}^x\left[\int_0^t \zeta^2(X_s(\varepsilon\zeta))ds\right]$. Apply Grönwall's lemma, we have,

$$\begin{aligned}
&\mathbb{E}^x|X_t(\varepsilon\zeta) - X_t(0)|^2 \\
\leq& 2\varepsilon^2 g_\zeta(t,x) + \int_0^t 2\varepsilon^2 g_\zeta(s,x) 2(L(M+1))^2 t \exp\left(\int_s^t (L(M+1))^2 t dr\right) ds \\
=& 2\varepsilon^2 g_\zeta(t,x) + 2\varepsilon^2 2(L(M+1))^2 t \int_0^t g_\zeta(s,x) \exp\left\{2(L(M+1))^2 t(t-s)\right\} ds
\end{aligned}$$

Since $g_\zeta(t,x)$ is non-negative non-decreasing in t, we have,

$$\begin{aligned}
\sup_{0\leq t\leq T, \pi\in Lip_M} \mathbb{E}^x|X_t(\varepsilon\zeta) - X_t(0)|^2 \leq& 2\varepsilon^2 g_\zeta(T,x) \\
&+ 2\varepsilon^2 2(L(M+1))^2 T \int_0^T g_\zeta(s,x) \exp\left\{2(L(M+1))^2 T(T-s)\right\} ds
\end{aligned}$$

□

5.3 Proof of Theorem 3.2

Proof.

$$\begin{aligned}
|\Upsilon^{k-1,i}_{1,\sigma} - \Upsilon^{k-1}_{1,0}| &= |b(\check{X}_{t_{k-1},i}(\sigma), \pi(\tilde{X}^h_{t_{k-1}}(\sigma))) + \sigma\check{W}_i(t_{k-1}) - b(X_{t_{k-1}}(0)), \pi(X_{t_{k-1}}(0)))| \\
&\leq L|\check{X}_{t_{k-1},i}(\sigma) - X_{t_{k-1}}(0)| + LM|\tilde{X}^h_{t_{k-1}}(\sigma) - X_{t_{k-1}}(0)| + \sigma|\check{W}_i(t_{k-1})|
\end{aligned}$$

$$\begin{aligned}
|\Upsilon^{k-1,i}_{2,\sigma} - \Upsilon^{k-1}_{2,0}| &= |b(\check{X}_{t_{k-1},i}(\sigma) + \Upsilon^{k-1,i}_{1,\sigma}/2, \pi(\tilde{X}^h_{t_{k-1}}(\sigma))) \\
&\quad + \sigma\check{W}_i(t_{k-1} + T/N) - b(X_{t_{k-1}}(0) + \Upsilon^{k-1}_{1,0}/2, \pi(X_{t_{k-1}}(0)))| \\
&\leq L|\check{X}_{t_{k-1},i}(\sigma) + \Upsilon^{k-1,i}_{1,\sigma}/2 - X_{t_{k-1}}(0) - \Upsilon^{k-1}_{1,0}/2| + LM|\tilde{X}^h_{t_{k-1}}(\sigma) - X_{t_{k-1}}(0)| \\
&\quad + \sigma|\check{W}_i(t_{k-1} + T/(2N))| \\
&\leq L|\check{X}_{t_{k-1},i}(\sigma) - X_{t_{k-1}}(0)| + \frac{1}{2}|\Upsilon^{k-1,i}_{1,\sigma} - \Upsilon^{k-1}_{1,0}| + LM|\tilde{X}^h_{t_{k-1}}(\sigma) - X_{t_{k-1}}(0)| \\
&\quad + \sigma|\check{W}_i(t_{k-1} + T/(2N))| \\
&\leq \frac{3}{2}\left(L|\check{X}_{t_{k-1},i}(\sigma) - X_{t_{k-1}}(0)| + LM|\tilde{X}^h_{t_{k-1}}(\sigma) - X_{t_{k-1}}(0)|\right) + \sigma|\check{W}_i(t_{k-1} + T/(2N))|
\end{aligned}$$

$$
\begin{aligned}
|\Upsilon_{3,\sigma}^{k-1,i} - \Upsilon_{3,0}^{k-1}| &= |b(\check{X}_{t_{k-1},i}(\sigma) + \Upsilon_{2,\sigma}^{k-1,i}/2, \pi(\tilde{X}_{t_{k-1}}^h(\sigma))) + \\
&\quad \sigma\check{W}_i(t_{k-1} + T/N) - b(X_{t_{k-1}}(0) + \Upsilon_{2,0}^{k-1}/2, \pi(X_{t_{k-1}}(0)))| \\
&\leq L|\check{X}_{t_{k-1},i}(\sigma) - X_{t_{k-1}}(0)| + \frac{1}{2}|\Upsilon_{2,\sigma}^{k-1,i} - \Upsilon_{2,0}^{k-1}| + LM|\tilde{X}_{t_{k-1}}^h(\sigma) - X_{t_{k-1}}(0)| \\
&\quad + \sigma|\check{W}_i(t_{k-1} + T/(2N))| \\
&\leq \frac{7}{4}\left(L|\check{X}_{t_{k-1},i}(\sigma) - X_{t_{k-1}}(0)| + LM|\tilde{X}_{t_{k-1}}^h(\sigma) - X_{t_{k-1}}(0)|\right) + \frac{3}{2}\sigma|\check{W}_i(t_{k-1} + T/(2N))|
\end{aligned}
$$

$$
\begin{aligned}
|\Upsilon_{4,\sigma}^{k-1,i} - \Upsilon_{4,0}^{k-1}| &= |b(\check{X}_{t_{k-1},i}(\sigma) + \Upsilon_{3,\sigma}^{k-1,i}, \pi(\tilde{X}_{t_{k-1}}^h(\sigma))) + \\
&\quad \sigma\check{W}_i(t_{k-1} + T/N) - b(X_{t_{k-1}}(0) + \Upsilon_{3,0}^{k-1}, \pi(X_{t_{k-1}}(0)))| \\
&\leq L|\check{X}_{t_{k-1},i}(\sigma) - X_{t_{k-1}}(0)| + |\Upsilon_{3,\sigma}^{k-1,i} - \Upsilon_{3,0}^{k-1}| + LM|\tilde{X}_{t_{k-1}}^h(\sigma) - X_{t_{k-1}}(0)| \\
&\quad + \sigma|\check{W}_i(t_{k-1} + T/N)| \\
&\leq \frac{11}{4}\left(L|\check{X}_{t_{k-1},i}(\sigma) - X_{t_{k-1}}(0)| + LM|\tilde{X}_{t_{k-1}}^h(\sigma) - X_{t_{k-1}}(0)|\right) + \frac{3}{2}\sigma|\check{W}_i(t_{k-1} + T/(2N))| \\
&\quad + \sigma|\check{W}_i(t_{k-1} + T/N)|
\end{aligned} \tag{5.1}
$$

$$
\begin{aligned}
\text{Cov}\left(\tilde{X}_{t_{k-1}}^h(\sigma), \Upsilon_{1,\sigma}^{k-1,i}\right) &= \text{Cov}\left(\tilde{X}_{t_{k-1}}^h(\sigma), b(\check{X}_{t_{k-1},i}(\sigma), \pi(\tilde{X}_{t_{k-1}}^h(\sigma))) + \sigma\check{W}_i(t_{k-1})\right) \\
&= \text{Cov}\left(\tilde{X}_{t_{k-1}}^h(\sigma), b(\check{X}_{t_{k-1},i}(\sigma), \pi(\tilde{X}_{t_{k-1}}^h(\sigma)))\right) \\
&= \left[\text{Var}(\tilde{X}_{t_{k-1}}^h(\sigma))\text{Var}(b(\check{X}_{t_{k-1},i}(\sigma), \pi(\tilde{X}_{t_{k-1}}^h(\sigma))))\right]^{\frac{1}{2}} \cdot \\
&\qquad \text{Corr}\left(\tilde{X}_{t_{k-1}}^h(\sigma), b(\check{X}_{t_{k-1},i}(\sigma), \pi(\tilde{X}_{t_{k-1}}^h(\sigma)))\right) \\
&\to 0, \text{ as } i \to \infty
\end{aligned}
$$

And

$$
\begin{aligned}
&\text{Cov}\left(\tilde{X}^h_{t_{k-1}}(\sigma), \Upsilon^{k-1,i}_{2,\sigma}\right) \\
=&\text{Cov}\left(\check{X}_{t_{k-1},i}(\sigma), b(\check{X}_{t_{k-1},i}(\sigma) + \Upsilon^{k-1,i}_{1,\sigma}/2, \pi(X_{t_{k-1}}(0))) + \sigma\check{W}_i(t_{k-1} + T/(2N))\right) \\
=&\text{Cov}\left(\check{X}_{t_{k-1},i}(\sigma), b(\check{X}_{t_{k-1},i}(\sigma) + \Upsilon^{k-1,i}_{1,\sigma}/2, \pi(X_{t_{k-1}}(0)))\right) \\
\to& 0, \text{ as } i \to \infty
\end{aligned}
$$

Similarly, $\text{Cov}\left(\tilde{X}^h_{t_{k-1}}(\sigma), \Upsilon^{k-1,i}_{j,\sigma}\right) \to 0$, for $j = 3, 4$, as $i \to \infty$. Thus,

$$
\begin{aligned}
&\text{Cov}\Bigg(\tilde{X}^h_{t_{k-1}}(\sigma) - X_{t_{k-1}}(0), \\
&\qquad \frac{T}{6mN}\sum_{i=0}^{m-1}\left((\Upsilon^{k-1,i}_{1,\sigma} - \Upsilon^{k-1}_{1,0}) + 2(\Upsilon^{k-1,i}_{2,\sigma} - \Upsilon^{k-1}_{2,0}) + 2(\Upsilon^{k-1,i}_{3,\sigma} - \Upsilon^{k-1}_{3,0}) + (\Upsilon^{k-1,i}_{4,\sigma} - \Upsilon^{k-1}_{4,0})\right)\Bigg) \\
=&\frac{T}{6mN}\sum_{i=0}^{m-1}\text{Cov}\left(\tilde{X}^h_{t_{k-1}}(\sigma), \Upsilon^{k-1,i}_{1,\sigma} + 2\Upsilon^{k-1,i}_{2,\sigma} + 2\Upsilon^{k-1,i}_{3,\sigma} + \Upsilon^{k-1,i}_{4,\sigma}\right) \\
=&o(T/N)
\end{aligned}
$$

$$
\begin{aligned}
&\mathbb{E}\left[\tilde{X}^h_{t_k}(\sigma) - X_{t_k}(0)\right]^2 \\
&= \mathbb{E}\Bigg[\left(\tilde{X}^h_{t_{k-1}}(\sigma) - X_{t_{k-1}}(0)\right) \\
&\quad + \frac{T}{6mN}\sum_{i=0}^{m-1}\Big((\Upsilon^{k-1,i}_{1,\sigma} - \Upsilon^{k-1}_{1,0}) + 2(\Upsilon^{k-1,i}_{2,\sigma} - \Upsilon^{k-1}_{2,0}) + 2(\Upsilon^{k-1,i}_{3,\sigma} - \Upsilon^{k-1}_{3,0}) \\
&\qquad + (\Upsilon^{k-1,i}_{4,\sigma} - \Upsilon^{k-1}_{4,0})\Big)\Bigg]^2 \\
&= \mathbb{E}\left[\tilde{X}^h_{t_{k-1}}(\sigma) - X_{t_{k-1}}(0)\right]^2 + \mathbf{Cov}\Bigg(\tilde{X}^h_{t_{k-1}}(\sigma) - X_{t_{k-1}}(0), \\
&\quad \frac{T}{6mN}\sum_{i=0}^{m-1}\Big((\Upsilon^{k-1,i}_{1,\sigma} - \Upsilon^{k-1}_{1,0}) + 2(\Upsilon^{k-1,i}_{2,\sigma} - \Upsilon^{k-1}_{2,0}) + 2(\Upsilon^{k-1,i}_{3,\sigma} - \Upsilon^{k-1}_{3,0}) \\
&\qquad + (\Upsilon^{k-1,i}_{4,\sigma} - \Upsilon^{k-1}_{4,0})\Big)\Bigg) \\
&\quad + \mathbb{E}\Bigg[\frac{T}{6mN}\sum_{i=0}^{m-1}\Big((\Upsilon^{k-1,i}_{1,\sigma} - \Upsilon^{k-1}_{1,0}) + 2(\Upsilon^{k-1,i}_{2,\sigma} - \Upsilon^{k-1}_{2,0}) + 2(\Upsilon^{k-1,i}_{3,\sigma} - \Upsilon^{k-1}_{3,0}) \\
&\qquad + (\Upsilon^{k-1,i}_{4,\sigma} - \Upsilon^{k-1}_{4,0})\Big)\Bigg]^2 \\
&\le \mathbb{E}\left[\tilde{X}^h_{t_{k-1}}(\sigma) - X_{t_{k-1}}(0)\right]^2 + o(T/N) \\
&\quad + \frac{T^2}{36N^2m^2}\mathbb{E}\Bigg[\sum_{i=0}^{m-1}\Big((\Upsilon^{k-1,i}_{1,\sigma} - \Upsilon^{k-1}_{1,0}) + 2(\Upsilon^{k-1,i}_{2,\sigma} - \Upsilon^{k-1}_{2,0}) + 2(\Upsilon^{k-1,i}_{3,\sigma} - \Upsilon^{k-1}_{3,0}) \\
&\qquad + (\Upsilon^{k-1,i}_{4,\sigma} - \Upsilon^{k-1}_{4,0})\Big)\Bigg]^2 \\
&\le \mathbb{E}\left[\tilde{X}^h_{t_{k-1}}(\sigma) - X_{t_{k-1}}(0)\right]^2 + o(T/N) \\
&\quad + \frac{T^2}{36N^2m^2}\mathbb{E}\Bigg[\sum_{i=0}^{m-1}\Big(|\Upsilon^{k-1,i}_{1,\sigma} - \Upsilon^{k-1}_{1,0}| + 2|\Upsilon^{k-1,i}_{2,\sigma} - \Upsilon^{k-1}_{2,0}| + 2|\Upsilon^{k-1,i}_{3,\sigma} - \Upsilon^{k-1}_{3,0}| \\
&\qquad + |\Upsilon^{k-1,i}_{4,\sigma} - \Upsilon^{k-1}_{4,0}|\Big)\Bigg]^2
\end{aligned}
$$

$$
\begin{aligned}
&\leq \mathbb{E}\left[\tilde{X}^h_{t_{k-1}}(\sigma) - X_{t_{k-1}}(0)\right]^2 + o(T/N) \\
&\quad + \frac{T^2}{36N^2m^2}\mathbb{E}\Bigg[\sum_{i=0}^{m-1}\Bigg(\frac{41}{4}\left(L|\check{X}_{t_{k-1},i}(\sigma) - X_{t_{k-1}}(0)| + LM|\tilde{X}^h_{t_{k-1}}(\sigma) - X_{t_{k-1}}(0)|\right) \\
&\qquad + \sigma|\check{W}_i(t_{k-1})| + \frac{13}{2}\sigma|\check{W}_i(t_{k-1}+T/(2N))| + \sigma|\check{W}_i(t_{k-1}+h)|\Bigg)\Bigg]^2 \\
&\leq \left(1 + \frac{41LMT^2}{72N^2m}\right)\mathbb{E}\left[\tilde{X}^h_{t_{k-1}}(\sigma) - X_{t_{k-1}}(0)\right]^2 + o(T/N) \\
&\quad + \frac{T^2}{36N^2m^2}2\mathbb{E}\left[\sum_{i=0}^{m-1}\frac{41}{4}L|\check{X}_{t_{k-1},i}(\sigma) - X_{t_{k-1}}(0)|\right]^2 + \frac{T^2}{36N^2m^2}O(\sigma^2 m)
\end{aligned}
$$

where the second last inequality comes from (5.1).

$$
\begin{aligned}
&\mathbb{E}\left[\sum_{i=0}^{m-1}|\check{X}_{t_{k-1},i}(\sigma) - X_{t_{k-1}}(0)|\right]^2 \\
\leq&\mathbb{E}\left[m\max_{1\leq i\leq m-1}|\check{X}_{t_{k-1},i}(\sigma) - X_{t_{k-1}}(0)|\right]^2 \\
\leq&m^2\Bigg\{\mathbb{E}\left[\max_{1\leq i\leq m-1}||\check{X}_{t_{k-1},i}(\sigma) - \tilde{X}^h_{t_{k-1}}(\sigma)|\right]^2 \\
&\qquad + \mathbb{E}\left[|\tilde{X}^h_{t_{k-1}}(\sigma) - X_{t_{k-1}}(0)|\right]^2\Bigg\} \\
\leq&m^2\mathbb{E}\left[\max_{1\leq i\leq m-1}||\sum_{j=0}^{i-1}\frac{1}{6}h\left(\Upsilon^{k-1,j}_{1,\sigma} + 2\Upsilon^{k-1,j}_{2,\sigma} + 2\Upsilon^{k-1,j}_{3,\sigma} + \Upsilon^{k-1,j}_{4,\sigma}\right)|\right]^2 \\
&\qquad + m^2\mathbb{E}\left[|\tilde{X}^h_{t_{k-1}}(\sigma) - X_{t_{k-1}}(0)|\right]^2 \\
\leq&\frac{T^2m^2}{6N^2}\mathbb{E}\left[\sum_{j=0}^{k-2}\left|\Upsilon^{k-1,j}_{1,\sigma} + 2\Upsilon^{k-1,j}_{2,\sigma} + 2\Upsilon^{m-1,j}_{3,\sigma} + \Upsilon^{k-1,j}_{4,\sigma}\right|\right]^2 \\
&\qquad + m^2\mathbb{E}\left[|\tilde{X}^h_{t_{k-1}}(\sigma) - X_{t_{k-1}}(0)|\right]^2 \\
\leq&O\left(\frac{T^2m^4}{6N^2}\right) + m^2\mathbb{E}\left[|\tilde{X}^h_{t_{k-1}}(\sigma) - X_{t_{k-1}}(0)|\right]^2
\end{aligned}
$$

Thus,

$$
\begin{aligned}
&\mathbb{E}\left[\tilde{X}^h_{t_k}(\sigma) - X_{t_k}(0)\right]^2 \\
\leq\ &\left(1 + \frac{41LMT^2}{72N^2m}\right)\mathbb{E}\left[\tilde{X}^h_{t_{k-1}}(\sigma) - X_{t_{k-1}}(0)\right]^2 + o(T/N) \\
&\quad + \frac{T^2}{36N^2m^2}\left\{2\mathbb{E}\left[\sum_{i=0}^{m-1}\frac{41}{4}L|\check{X}_{t_{k-1},i}(\sigma) - X_{t_{k-1}}(0)|\right]^2 + O(\sigma^2 m)\right\} \\
\leq\ &\left(1 + \frac{41LMT^2}{72N^2m}\right)\mathbb{E}\left[\tilde{X}^h_{t_{k-1}}(\sigma) - X_{t_{k-1}}(0)\right]^2 + o(T/N) \\
&\quad + \frac{T^2}{36N^2m^2}\left\{O(\frac{T^2m^4}{N^2}) + O(m^2)\mathbb{E}\left[|\tilde{X}^h_{t_{k-1}}(\sigma) - X_{t_{k-1}}(0)|\right]^2 + O(\sigma^2 m)\right\} \\
=\ &(1 + O(T^2/N^2))\mathbb{E}\left[\tilde{X}^h_{t_{k-1}}(\sigma) - X_{t_{k-1}}(0)\right]^2 + o(T/N) \\
&\quad + O((T/N)^4m^2 + \sigma^2(T/N)^2/m)
\end{aligned}
$$

By induction, we have

$$
\begin{aligned}
&\mathbb{E}\left[|\check{X}_{t_k}(\sigma) - X_{t_k}(0))|^2\right] \\
\leq &\left(1 + O(T^2/N^2)\right) \mathbb{E}\left[|\tilde{X}^h_{t_{k-1}}(\sigma) - X_{t_{k-1}}(0)|^2\right] \\
&\quad + o(T/N) + O((T/N)^4 m^2 + \sigma^2 (T/N)^2/m) \\
\leq &\left(1 + O(T^2/N^2)\right) \Bigg\{ \left(1 + O(T^2/N^2)\right) \mathbb{E}\left[|\tilde{X}^h_{t_{k-2}}(\sigma) - X_{t_{n-2}}(0)|^2\right] \\
&\quad + o(T/N) + O((T/N)^4 m^2 + \sigma^2 (T/N)^2/m) \Bigg\} + o(T/N) \\
&\quad + O((T/N)^4 m^2 + \sigma^2 (T/N)^2/m) \\
&\cdots \\
\leq &\left(1 + O(T^2/N^2)\right)^n \mathbb{E}\left[|\tilde{X}^h_0(\sigma) - X_0(0)|^2\right] \\
&\quad + \sum_{i=1}^{k-1} \left(1 + O(T^2/N^2)\right)^i \left[o(T/N) + O((T/N)^4 m^2 + \sigma^2 (T/N)^2/m)\right] \\
= &\frac{1 - (1 + O(T^2/N^2))^{k-1}}{1 - (1 + O(T^2/N^2))} \left[o(T/N) + O((T/N)^4 m^2 + \sigma^2 (T/N)^2/m)\right] \\
= &\frac{1}{O(T^2/N^2)} \left(\left(1 + O(T^2/N^2)\right)^{k-1} - 1\right) \left[o(T/N) + O((T/N)^4 m^2 + \sigma^2 (T/N)^2/m)\right] \\
\sim &\frac{1}{O(T^2/N^2)} \left(e^{O(T^2/N^2)k} - 1\right) \left[o(T/N) + O((T/N)^4 m^2 + \sigma^2 (T/N)^2/m)\right] \\
\sim &\frac{1}{O(T^2/N^2)} O(T^2/N^2) k \left[o(T/N) + O((T/N)^4 m^2 + \sigma^2 (T/N)^2/m)\right] \\
= &O\left((T/N)^3 m^2 + \sigma^2 (T/N)/m)\right), \quad \forall k \leq N \\
\sim &O\left(h^2 (T/N) + \sigma^2 (T/N)^2/h\right), \quad \forall k \leq N
\end{aligned}
$$

where the last approximation is because $h \sim mT/N$. □

5.4 Proof of Theorem 3.3

Proof. For simplicity, let $\tilde{r}(x) = r(x, \pi(x))$. We have,

$$\begin{aligned}
|\tilde{r}(x) - \tilde{r}(y)| =& |r(x, \pi(x)) - r(y, \pi(y))| \\
\leq& L_r \left[|x - y| + |\pi(x) - \pi(y)| \right] \\
\leq& L_r (M + 1) |x - y|
\end{aligned}$$

We know that $\forall \xi > 0$, $\exists T$, s.t. $e^{-\beta T} \leq \frac{\xi}{4M_r}$. We will prove $\forall \pi \in Lip_M$, $\exists \tau$ s.t. $\forall \xi \in (0, \tau)$, $V^{\pi}_{\xi\sigma}(x) - V^{\pi}_0(x) \leq \xi$.

$$\begin{aligned}
& V^{\pi}_{\sigma}(x) - V^{\pi}_0(x) \\
=& \mathbb{E}\left[\int_0^{\infty} e^{-\beta t} [\tilde{r}(X_t(\sigma)) - \tilde{r}(X_t(0))] dt \middle| X_0(\sigma) = x, X_0 = x \right] \\
=& \lim_{T \to \infty} \mathbb{E}\left[\int_0^{T} e^{-\beta t} [\tilde{r}(X_t(\sigma)) - \tilde{r}(X_t(0))] dt \middle| X_0(\sigma) = x, X_0 = x \right] \\
=& \lim_{T \to \infty} \int_0^{T} e^{-\beta t} \mathbb{E}\left[|\tilde{r}(X_t(\sigma)) - \tilde{r}(X_t(0))| \, X_0(\sigma) = x, X_0 = x \right] dt \\
=& \int_0^{T} e^{-\beta t} \mathbb{E}\left[|\tilde{r}(X_t(\sigma)) - \tilde{r}(X_t(0))| \, X_0(\sigma) = x, X_0 = x \right] dt \\
& + \int_T^{\infty} e^{-\beta t} \mathbb{E}\left[|\tilde{r}(X_t(\sigma)) - \tilde{r}(X_t(0))| \, X_0(\sigma) = x, X_0 = x \right] dt \\
\leq& \int_0^{T} e^{-\beta t} \mathbb{E}\left[|\tilde{r}(X_t(\sigma)) - \tilde{r}(X_t(0))| \, X_0(\sigma) = x, X_0 = x \right] dt + 2M_r e^{-\beta T} \\
\leq& \int_0^{T} e^{-\beta t} \mathbb{E}\left[|\tilde{r}(X_t(\sigma)) - \tilde{r}(X_t(0))| \, X_0(\sigma) = x, X_0 = x \right] dt + \frac{\xi}{2}
\end{aligned}$$

The second equation is from dominated convergence theorem and the third equation is from Fubini's theorem.

We know that from Theory 3.1, $\exists M_T^x$, $\exists \tau$, s.t. $\tau^2 M_T^x \leq \frac{\xi^2}{4[L_r(M+1)]^2}$ and,

$$\sup_{0\leq t\leq T} \mathbb{E}\left[\left|X_t(\tau\zeta) - X_t(0)\right|^2 | X_0(\zeta) = x, X_0(0) = x\right] \leq \tau^2 M_T^x$$

Then with cauchy-Schwarz inequality, we have $\forall t \in [0, T]$,

$$\mathbb{E}^2\left[\tilde{r}(X_t(\tau\zeta)) - \tilde{r}(X_t(0))| \, X_0(\zeta) = x, X_0(0) = x\right] \leq \tau^2[L_r(M+1)]^2 M_T^x \leq \frac{\xi^2}{4}$$

Thus, $\forall \varepsilon \in (0, \tau)$,

$$\begin{aligned}
&V_{\varepsilon\zeta}^{\pi}(x) - V_0^{\pi}(x) \\
=&\int_0^T e^{-\beta t}\mathbb{E}\left[\tilde{r}(X_t(\varepsilon\zeta)) - \tilde{r}(X_t(0))| \, X_0(\varepsilon\zeta) = x, X_0 = x\right] dt + \frac{\xi}{2} \\
\leq&\int_0^T e^{-\beta t}\xi[L_r(M+1)]\sqrt{M_T^x}dt + \frac{\xi}{2} \\
\leq&\varepsilon[L_r(M+1)]\sqrt{M_T^x} + \frac{\xi}{2} \\
\leq&\xi
\end{aligned}$$

□

5.5 Proof of Lemma 3.5

Proof. If $\frac{\partial g_\zeta(t,x)}{\partial x}$ exists and is finite $\forall t \in [0, T]$,

$$\frac{d}{dx}\sqrt{2g_\zeta(T,x) + 4L^2T\int_0^T g_\zeta(s,x)\exp(2L^2T(T-s))ds} < \infty$$

$\forall \xi > 0$, $\exists \tau, T$, s.t. $e^{-\beta T} \leq \frac{\xi}{4M_r}$, and

$$\tau \left| L_r(M+1)\frac{d}{dx}\sqrt{\left|2g_\zeta(T,x)+4L^2T\int_0^T g_\zeta(s,x)\exp(2L^2T(T-s))ds\right|}\right| < \frac{\xi}{2}$$

$\forall \varepsilon \in (0,\tau)$, $\forall \pi \in Lip_M$, we have,

$$\begin{aligned}
&\frac{\partial V^\pi_{\varepsilon\zeta}(x)}{\partial x} - \frac{\partial V^\pi_0(x)}{\partial x} = \frac{\partial\left\{V^\pi_{\varepsilon\zeta}(x) - V^\pi_0(x)\right\}}{\partial x} \\
=&\lim_{y\to x}\frac{\left\{V^\pi_{\varepsilon\zeta}(x) - V^\pi_0(x)\right\} - \left\{V^\pi_{\varepsilon\zeta}(y) - V^\pi(y)\right\}}{x-y} \\
=&\lim_{y\to x}\frac{1}{x-y}\left\{\mathbb{E}\left[\int_0^\infty e^{-\beta t}[\tilde r(X_t(\varepsilon\zeta)) - \tilde r(X_t)]dt\middle| X_0(\varepsilon\zeta)=x, X_0=x\right]\right. \\
&\qquad\left. -\mathbb{E}\left[\int_0^\infty e^{-\beta t}[\tilde r(X_t(\varepsilon\zeta)) - \tilde r(X_t)]dt\middle| X_0(\varepsilon\zeta)=y, X_0=y\right]\right\} \\
\le&\lim_{y\to x}\frac{1}{x-y}\left\{\int_0^T e^{-\beta t}\mathbb{E}\left[\tilde r(X_t(\varepsilon\zeta)) - \tilde r(X_t)\middle| X_0(\varepsilon\zeta)=x, X_0=x\right]dt + \frac{\xi}{2}\right. \\
&\qquad\left. -\int_0^T e^{-\beta t}\mathbb{E}\left[\tilde r(X_t(\varepsilon\zeta)) - \tilde r(X_t)\middle| X_0(\varepsilon\zeta)=y, X_0=y\right]dt\right\} \\
=&\int_0^T e^{-\beta t}\lim_{y\to x}\frac{1}{x-y}\left\{\mathbb{E}\left[\tilde r(X_t(\varepsilon\zeta)) - \tilde r(X_t)\middle| X_0(\varepsilon\zeta)=x, X_0=x\right]\right. \\
&\qquad\left. -\mathbb{E}\left[\tilde r(X_t(\varepsilon\zeta)) - \tilde r(X_t)\middle| X_0(\varepsilon\zeta)=y, X_0=y\right]\right\}dt + \frac{\xi}{2} \\
=&\int_0^T e^{-\beta t}\frac{d\mathbb{E}\left[\tilde r(X_t(\varepsilon\zeta)) - \tilde r(X_t)\middle| X_0(\varepsilon\zeta)=x, X_0=x\right]}{dx}dt + \frac{\xi}{2} \\
\le&[L_r(M+1)]\int_0^T e^{-\beta t}\frac{d\mathbb{E}\left[|X_t(\varepsilon\zeta) - X_t|\,\middle| X_0(\varepsilon\zeta)=x, X_0=x\right]}{dx}dt + \frac{\xi}{2} \\
\le&[L_r(M+1)]\varepsilon\frac{d}{dx}\sqrt{\left|2g_\zeta(T,x)+4L^2T\int_0^T g_\zeta(s,x)\exp(2L^2T(T-s))ds\right|}\int_0^T e^{-\beta t}dt + \frac{\xi}{2} \\
\le&\xi
\end{aligned}$$

where the fourth equation is from dominant control theorem and the second last inequality is from Theorem 3.1. So far, We proved $\forall \xi > 0$, $\exists \tau$, s.t. $\forall \varepsilon \in (0,\tau)$,

$\forall \pi \in Lip_M$,

$$\frac{\partial V_{\varepsilon\zeta}^{\pi}(x)}{\partial x} - \frac{\partial V_0^{\pi}(x)}{\partial x} \leq \xi$$

□

5.6 Proof of Lemma 3.8

Proof. Similar to Section 5.5, $\exists T > 0, \forall \xi > 0$, s.t.

$$\begin{aligned}
|\tilde{V}_{\sigma}^{h,\pi}(x) - \tilde{V}_0^{0,\pi}(x)| &= \mathbb{E}\left[\int_0^{\infty} e^{-\beta t}\left[\tilde{r}(\tilde{X}_t^h(\sigma)) - \tilde{r}(\check{X}_t(0))\right] dt \middle| \tilde{X}_0^h(\sigma) = x, \check{X}_0(0) = x\right] \\
&\leq \mathbb{E}\left[\int_0^{T} e^{-\beta t}\left[\tilde{r}(\tilde{X}_t^h(\sigma)) - \tilde{r}(\check{X}_t(0))\right] dt \middle| \tilde{X}_0^h(\sigma) = x, \check{X}_0(0) = x\right] + \frac{\xi}{2} \\
&\leq L_r \mathbb{E}\left[\int_0^{T} e^{-\beta t} |\tilde{X}_t^h(\sigma) - \check{X}_t(0)| dt \middle| \tilde{X}_0^k(\sigma) = x, \tilde{X}_0 = x\right] + \frac{\xi}{2} \\
&\leq \sup_{1\leq k\leq n} \mathbb{E}\left|\tilde{X}_{t_k}^h(\sigma) - \check{X}_{t_k}(0)\right| L_r \int_0^{T} e^{-\beta T} dt + \frac{\xi}{2} \\
&\leq O\left(L_r \sqrt{h^2(T/N) + (\sigma^2/h)(T/N)}\right) + \frac{\xi}{2} \\
&\leq \xi
\end{aligned}$$

where the second last inequality is from Theorem 3.2. □

5.7 Proof of Lemma 3.9

Proof. Similar to Section 5.5, $\exists T > 0, \forall \xi > 0$, s.t.

$$
\begin{aligned}
& V_0^{\pi}(x) - \tilde{V}_0^{0,\pi}(x) \\
=& \left[\int_0^{\infty} e^{-\beta t} \left[\tilde{r}(X_t(0)) - \tilde{r}(\check{X}_t(0)) \right] dt \middle| X_0(0) = x, \check{X}_0(0) = x \right] \\
\leq& \left[\int_0^{T} e^{-\beta t} \left[\tilde{r}(X_t(0)) - \tilde{r}(\check{X}_t(0)) \right] dt \middle| X_0(0) = x, \check{X}_0(0) = x \right] + \frac{\xi}{2} \\
\leq& L_r \left[\int_0^{T} e^{-\beta t} \left| X_t(0) - \check{X}_t(0) \right| dt \middle| X_0(0) = x, \check{X}_0(0) = x \right] + \frac{\xi}{2} \\
\leq& L_r \left[\sum_{k=1}^{n} \int_{t_{k-1}}^{t_k} e^{-\beta t} \left[|X_t(0) - X_{t_k}(0)| + \left| X_{t_k}(0) - \check{X}_{t_k}(0) \right| \right] dt \middle| X_0(0) = x, \check{X}_0(0) = x \right] + \frac{\xi}{2} \\
\leq& L_r \sup_{0 \leq k \leq n,\ 0 \leq t \leq T} |X_t(0) - X_{t_k}(0)| \int_0^{T} e^{\beta t} dt \\
& \quad + L_r \left[\sum_{k=1}^{n} \int_{t_{k-1}}^{t_k} e^{-\beta t} |X_{t_k}(0) - \check{X}_{t_k}(0)| dt \middle| X_0(0) = x, \check{X}_0(0) = x \right] + \frac{\xi}{2} \\
\leq& L_r \sup_{0 \leq k \leq n,\ 0 \leq t \leq T} |X_t(0) - X_{t_k}(0)| + L_r \left[\sum_{k=1}^{n} \int_{t_{k-1}}^{t_k} e^{-\beta t} |O((T/N)^4) dt \middle| X_0(0) = x, \check{X}_0(0) = x \right] + \frac{\xi}{2} \\
\leq& L_r \sup_{0 \leq k \leq n,\ 0 \leq s \leq T} b(X_s(0)) T/N + O((T/N)^4) + \frac{\xi}{2} \\
\leq& \xi
\end{aligned}
$$

where the second last inequality is from mean value theorem and because $\check{X}_{t_k}$ is generated from **4th** order RK method. □

5.8 Proof of Lemma 3.12

Proof. $\forall \xi > 0, \exists K,\ T$, s.t. $e^{-\beta T} \leq \frac{\xi}{8M_r}$ and $[L_r(M+1)]O\left(\frac{h}{k}\right) \leq \frac{\xi}{2}, \forall k > K$.

$\forall \pi \in Lip_M$, we have,

$$
\begin{aligned}
&\frac{\partial \tilde{V}_\sigma^{h,\pi}(x)}{\partial x} - \frac{\partial V_0^\pi(x)}{\partial x} = \frac{\partial \left\{\tilde{V}_\sigma^{h,\pi}(x) - V_0^\pi(x)\right\}}{\partial x} \\
=&\lim_{y\to x} \frac{\left\{\tilde{V}_\sigma^{h,\pi}(x) - V_0^\pi(x)\right\} - \left\{\tilde{V}_\sigma^{h,\pi}(y) - V_0^\pi(y)\right\}}{x-y} \\
=&\lim_{y\to x} \frac{1}{x-y}\left\{\mathbb{E}\left[\int_0^\infty e^{-\beta t}[\tilde{r}(\tilde{X}_t^h(\sigma)) - \tilde{r}(X_t(0))]dt \middle| \tilde{X}_0^h(\sigma) = x, X_0(0) = x\right]\right. \\
&\qquad\left. -\mathbb{E}\left[\int_0^\infty e^{-\beta t}[\tilde{r}(\tilde{X}_t^h(\sigma)) - \tilde{r}(X_t(0))]dt \middle| \tilde{X}_0^h(\sigma) = y, X_0(0) = y\right]\right\} \\
\le&\lim_{y\to x} \frac{1}{x-y}\left\{\int_0^T e^{-\beta t}\mathbb{E}\left[\tilde{r}(\tilde{X}_t^h(\sigma)) - \tilde{r}(X_t(0))\middle| \tilde{X}_0^h(\sigma) = x, X_0(0) = x\right]dt + \frac{\xi}{2}\right. \\
&\qquad\left. -\int_0^T e^{-\beta t}\mathbb{E}\left[\tilde{r}(\tilde{X}_t^h(\sigma)) - \tilde{r}(X_t(0))\middle| \tilde{X}_0^h(\sigma) = y, X_0(0) = y\right]dt\right\} \\
=&\int_0^T e^{-\beta t}\lim_{y\to x}\frac{1}{x-y}\left\{\mathbb{E}\left[\tilde{r}(\tilde{X}_t^h(\sigma)) - \tilde{r}(X_t(0))\middle| \tilde{X}_0^h(\sigma) = x, X_0(0) = x\right]\right. \\
&\qquad\left. -\mathbb{E}\left[\tilde{r}(\tilde{X}_t^h(\sigma)) - \tilde{r}(X_t(0))\middle| \tilde{X}_0^h(\sigma) = y, X_0(0) = y\right]\right\}dt + \frac{\xi}{2} \\
=&\int_0^T e^{-\beta t}\frac{d\mathbb{E}\left[\tilde{r}(\tilde{X}_t^h(\sigma)) - \tilde{r}(X_t(0))\middle| \tilde{X}_0^h(\sigma) = x, X_0(0) = x\right]}{dx}dt + \frac{\xi}{2} \\
\le&[L_r(M+1)]\int_0^T e^{-\beta t}\frac{d\mathbb{E}\left[|\tilde{X}_t^h(\sigma) - X_t(0)|\,\middle|\, \tilde{X}_0^h(\sigma) = x, X_0(0) = x\right]}{dx}dt + \frac{\xi}{2}
\end{aligned}
$$

where the fourth equation is from dominant control theorem. We know,

$$
\begin{aligned}
&\frac{d}{dx}\mathbb{E}\left[|\tilde{X}_t^h(\sigma) - X_t(0)|\,\middle|\, \tilde{X}_0^h(\sigma) = x, X_0(0) = x\right] \\
\le&\frac{d}{dx}\mathbb{E}\left[|\tilde{X}_t^h(\sigma) - \check{X}_t(0)| + |\check{X}_t(0) - X_t(0)|\,\middle|\, \tilde{X}_0^h(\sigma) = x, X_0(0) = x\right] \\
\le&\frac{d}{dx}\left\{O\left(h^2(T/N) + \left(\mathbb{E}\left[\sigma^2(\tilde{X}_t^h)|\tilde{X}_0^h(\sigma) = x\right]/h\right)(T/N)^2\right) + O((T/N)^4)\right\} \\
=&O\left(\frac{T^2}{N^2 h}\frac{d}{dx}\mathbb{E}\left[\sigma^2(\tilde{X}_t^h)|\tilde{X}_0^h(\sigma) = x\right]\right)
\end{aligned}
$$

where the second inequality is because of Theorem 3.2 and RK being 4th order. Thus,

$$\begin{aligned}
&\frac{\partial \tilde{V}_{\sigma}^{h,\pi}(x)}{\partial x} - \frac{\partial \tilde{V}_{0}^{0,\pi}(x)}{\partial x} \\
\leq &[L_r(M+1)] \int_0^T e^{-\beta t} O\left(\frac{T^2}{N^2 h} \frac{d}{dx} \mathbb{E}\left[\sigma^2(\tilde{X}_t^h) | \tilde{X}_0^h(\sigma) = x\right]\right) dt + \frac{\xi}{2} \\
= &[L_r(M+1)] O\left(\frac{T^2}{N^2 h} \frac{d}{dx} \mathbb{E}\left[\sigma^2(\tilde{X}_t^h) | \tilde{X}_0^h(\sigma) = x\right]\right) \int_0^T e^{-\beta t} dt + \frac{\xi}{2} \\
\leq &[L_r(M+1)] O\left(\frac{T^2}{N^2 h}\right) + \frac{\xi}{2} \\
\leq &\xi
\end{aligned}$$

We proved $\forall \xi > 0$, $\exists \tau$, s.t. $\forall \xi \in (0, \tau)$, $\forall \pi \in Lip_M$,

$$\frac{\partial \tilde{V}_{\sigma}^{h,\pi}(x)}{\partial x} - \frac{\partial V_{0}^{\pi}(x)}{\partial x} \leq \xi$$

□

REFERENCES

Bellman, Richard. 1956. Dynamic programming and lagrange multipliers. *Proceedings of the National Academy of Sciences of the United States of America* 42(10): 767.

Bertsekas, Dimitri P, Dimitri P Bertsekas, Dimitri P Bertsekas, and Dimitri P Bertsekas. 1995. *Dynamic programming and optimal control*, vol. 1. Athena scientific Belmont, MA.

Boyan, Justin A. 2002. Technical update: Least-squares temporal difference learning. *Machine learning* 49(2-3):233–246.

Bradtke, Steven J, and Andrew G Barto. 1996. Linear least-squares algorithms for temporal difference learning. *Machine learning* 22(1-3):33–57.

Dann, Christoph, Gerhard Neumann, Jan Peters, et al. 2014. Policy evaluation with temporal differences: A survey and comparison. *Journal of Machine Learning Research* 15:809–883.

Doya, Kenji. 2000. Reinforcement learning in continuous time and space. *Neural computation* 12(1):219–245.

Engel, Yaakov, Shie Mannor, and Ron Meir. 2003. Bayes meets bellman: The gaussian process approach to temporal difference learning. In *Proceedings of the 20th international conference on machine learning (icml-03)*, 154–161.

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. 2001. *The elements of statistical learning*. Springer series in statistics New York.

Howard, R.A. 1960. *Dynamic programming and markov processes*. Technology Press of Massachusetts Institute of Technology.

Kushner, Harold, and Paul G Dupuis. 1992. *Numerical methods for stochastic control problems in continuous time*, vol. 24. Springer Science & Business Media.

Kushner, Harold J. 1999. Consistency issues for numerical methods for variance control, with applications to optimization in finance. *IEEE Transactions on Automatic Control* 44(12):2283–2296.

Kushner, HJ, J Yang, and D Jarvis. 1990. Controlled and optimally con-trolled multiplexing systems: A numerical exploration. questa, 20: 255–291, 1995.| 6| | hj kushner. numerical methods for stochastic control problems in continuous time. *SIAM J. Control and Optimization* 28:999–1048.

Lillicrap, Timothy P., Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. Continuous control with deep reinforcement learning. `1509.02971`.

Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nature* 518(7540): 529–533.

Pham, Huyên. 2009. *Continuous-time stochastic control and optimization with financial applications*, vol. 61. Springer Science & Business Media.

Powell, Warren B, and Jun Ma. 2011. A review of stochastic algorithms with continuous value function approximation and some new approximate policy iteration algorithms for multidimensional continuous applications. *Journal of Control Theory and Applications* 9(3):336–352.

Recht, Benjamin. 2019. A tour of reinforcement learning: The view from continuous control. *Annual Review of Control, Robotics, and Autonomous Systems* 2:253–279.

Silver, David, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. 2017. Mastering the game of go without human knowledge. *Nature* 550(7676):354–359.

Steele, J Michael. 2012. *Stochastic calculus and financial applications*, vol. 45. Springer Science & Business Media.

Sutton, Richard S, Andrew G Barto, et al. 1998. *Introduction to reinforcement learning*, vol. 135. MIT press Cambridge.

Sutton, Richard S, Hamid R. Maei, and Csaba Szepesvári. 2009a. A convergent o(n) temporal-difference algorithm for off-policy learning with linear function approximation. In *Advances in neural information processing systems 21*, ed. D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, 1609–1616. Curran Associates, Inc.

Sutton, Richard S, Hamid Reza Maei, Doina Precup, Shalabh Bhatnagar, David Silver, Csaba Szepesvári, and Eric Wiewiora. 2009b. Fast gradient-descent methods for temporal-difference learning with linear function approximation.

Szepesvári, Csaba. 2010. Algorithms for reinforcement learning. *Synthesis lectures on artificial intelligence and machine learning* 4(1):1–103.

Tallec, Corentin, Léonard Blier, and Yann Ollivier. 2019. Making deep q-learning methods robust to time discretization.