Option Pricing and Hedging by Reinforcement Learning

by

Rong Dang

20598430

Supervised by

Tony Wirjanto

Yuying Li

A Research Report for Master of Computational Mathematics

University of Waterloo

December 2020

Abstract

The Black-Scholes Merton model provides a solution for option pricing in a perfect mathematical setup where the instruments have a lognormal distribution of prices and hedge can be continuously taken at no cost. In this paper, the Q-Learning Black-Scholes approach is presented where the option is hedged and priced under a discrete-time version of the classical Black-Scholes-Merton model, and is based on Reinforcement Learning. The option price is measured as an optimal Q-function while the optimal hedge is an argument of it. By learning the Q-function dynamically, the optimal hedge and optimal price are learned directly from data without any reference to any model explicitly. This paper investigates the performance of the Dynamic Programming solution and the Fitted Q Iteration solution for a data-driven Reinforcement Learning model and compares to the classical Black-Scholes-Merton model which is used as a benchmark.

Acknowledgement

I would like to thank Professor Tony Wirjanto and Professor Yuying Li for their supervision throughout the project and for helping me finalize the project, and thank Professor Chengguo Weng for his generous help with proofreading. I would also express my sincere appreciation to all the instructors and fellow students in the Master of Computational Mathematics program for their education and support.

Table of Contents

List of Figures

1 Introduction

1.1 Motivation

The Black-Scholes Merton (BSM) model, published in 1973 by Black and Scholes, and independently also by Merton, is the most well-known option pricing model in quantitative finance with its theoretical foundation that options can be priced in terms of other tradable assets, which is also known as a dynamic option replication. Specifically, an option can be mimicked by a simple hedge portfolio consisting of certain shares of the underlying asset and corresponding amount of cash. The hedge portfolio, under the BSM model however, is continuously rebalanced between stock shares and cash account in a self-financing way with no cash infusions or withdrawals after inception, aiming to mimic the option as closely as possible. This gives the BSM model an attractive property, that the pricing formula for European options is relatively simple and expressible in a closed form. However, the limitation also arises from its simplicity - the assumption under the BSM model that the rebalancing can be done continuously at no cost is only in an ideal setup and does not happen in the real market. Meanwhile, if the continuous-time setting were to happen, it would make the total portfolio completely risk-free since the replication of option would be instantaneously perfect.

In this paper, instead of the paradoxical continuous re-hedging setting, we consider a realistic finite-time hedging setup, where rebalancing between the underlying asset and the cash account in the portfolio can only happen discretely, thus taking mis-hedging risk into consideration. In this case, the overall goal is to minimize the slippage risk arises from the change in price of the underlying assets between any two consecutive re-hedges, where the risk is compensated in the option price relative to the classical BSM price. The process of rebalancing between the underlying asset and the cash account, in other words, determining

a hedging position at each time step throughout the option lifetime, is indeed a sequential decision-making process. Inspired by this setup, option pricing and hedging in discrete time becomes feasible to work on.

Reinforcement Learning (RL) takes the paths of the underlying asset prices as input and uses a Q-learning framework to output the optimal hedges and prices. The framework, coined as a Q-Learning Black-Scholes (QLBS) model, does not depend on any presumed distribution of the underlying assets. Instead it combines the data-driven Q-learning algorithm and the method of dynamic option replication under the BSM model. In this paper, we consider the simplest case to price and hedge European vanilla options with the QLBS approach, although the method can be further extended to other more complicated financial instruments.

The only intake of the QLBS approach would be the paths of the underlying asset price. As just mentioned earlier, since it is distribution-free and purely data-driven, it affords greater flexibility that either historical stock data or simulated data can be used in this approach. In this paper, for convenience, a Monte Carlo simulation is implemented for a stock prices generation. As for a numerical analysis, the Dynamic Programming (DP) solution and the Fitted Q Iteration (FQI) solution of RL are investigated. Moreover, as an extension of the classical BSM model, the performance of both solutions of RL will be compared to the BSM result which is used as a benchmark.

1.2 Organization of the Research Paper

The remaining parts of the paper is organized as follows. In Section 2, we present the foundation of RL in general. In Section 3, option hedging and pricing will be introduced. Starting from Section 4, the problem will be formulated in the RL setup, following which both the DP solution and the FQI solution will

be further discussed in Sections 5 and 6. In Section 7, a comparison of the implementation performance will be displayed with a corresponding sensitivity analysis. Finally, conclusions are made in Section 8.

2 Reinforcement Learning

Reinforcement Learning, as another sub-field of machine learning other than the traditional supervised and unsupervised learning, creates algorithms that can learn to take an optimal action in an environment. The optimal action is defined as an action that maximizes the expected lifetime reward [1]. The difference between supervised learning and RL is that a supervised learning solution would try to teach an agent which action to take given the current state, while a RL solution would instruct the agent to try out different strategies to find out which is the best [2]. Though RL is traditionally used in Atari games and the famous AlphaGo, it is more and more commonly applied to finance recently.

2.1 Markov Decision Processes

As shown in Figure 1, in the RL framework, an agent and an environment interact with each other through actions and rewards. An agent is the decision maker who selects an action a given the current situation in the environment, and the environment further reacts to the action with a reward R and takes the agent to a new state S. This process is called a Markov decision processes (MDP).

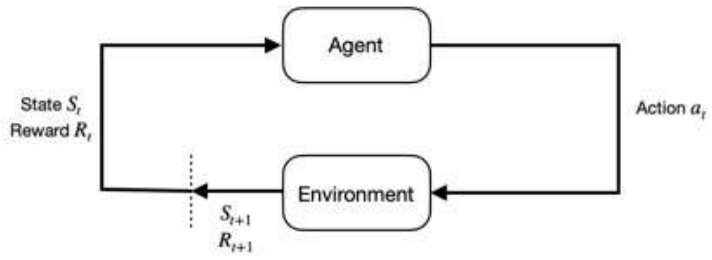

Figure 1: Reinforcement learning framework

This interaction occurs in a sequence following the time steps $t = 0, 1, 2, ...$ and the whole interaction process is given by: $S_0, a_0, R_1, S_1, a_1, R_2, S_2, ...$ until some termination condition is reached. The goal of this exercise is to learn the policy that maximizes the total rewards received over the entire episode. This brings out a dilemma between exploration and exploitation: the optimal action in the current state might result in the maximized immediate reward but not a large positive reward in the long run; whereas the action not being the most optimal for the current state might lead to a higher total reward.

To balance trade-off between the exploration and exploitation, we define the total reward as the accumulated future rewards being discounted to the current state, namely

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... \tag{1}$$

where R_t is an immediate reward received at time t and γ is a discount rate to be further specified in the context of a certain problem. The total rewards at any time t can be further expressed in a recursive form:

$$\begin{aligned} G_t &= R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... \\ &= R_{t+1} + \gamma(R_{t+2} + \gamma R_{t+3} + ...) \\ &= R_{t+1} + \gamma G_{t+1} \end{aligned} \tag{2}$$

The core idea of the RL algorithms is to learn the optimal policy $\pi(a \mid s)$, which maps a certain state s to a probability distribution over all possible actions to take. Given the state s and the policy π, the state-value function V and the action-value function Q are defined as measurements of the future reward expected in s:

$$V_\pi(s) = \mathbb{E}_\pi \left[G_t \mid S_t = s \right] \tag{3}$$

and

$$Q_\pi(s,a) = \mathbb{E}_\pi \left[G_t \mid S_t = s, A_t = a \right] \tag{4}$$

where the expectation is taken at π, ie. following the given policy π.

These two functions can be used interchangeably, where the only difference is that the action-value function Q is more specific as the value of being in state s and taking action a and following a policy π afterwards.

Following the same idea as (2), the value functions could also be written in a recursive form, which are also known as Bellman equations, as:

$$\begin{aligned} V_\pi(s) &= \mathbb{E}_\pi \left[G_t \mid S_t = s \right] \\ &= \mathbb{E}_\pi \left[R_{t+1} + \gamma G_{t+1} \mid S_t = s \right] \\ &= \mathbb{E}_\pi \left[R_{t+1} + \gamma V_\pi(s') \mid S_t = s \right] \end{aligned} \tag{5}$$

and

$$\begin{aligned} Q_\pi(s,a) &= \mathbb{E}_\pi \left[G_t \mid S_t = s, A_t = a \right] \\ &= \mathbb{E}_\pi \left[R_{t+1} + \gamma \max_{a' \in \mathcal{A}} Q_\pi(s', a') \mid S_t = s, A_t = a \right] \end{aligned} \tag{6}$$

where s' is the next state at $t+1$ and a' is the action to take in the next state.

In RL, the MDP is for the agent to learn the optimal policy π^* which maximizes the value function $V_t^\pi(X_t)$, or equivalently, the action-value function $Q_t^\pi(X_t, a_t)$:

$$\pi_t^*(X_t) = arg\max_\pi V_t^\pi(X_t) = arg\max_{a_t \in \mathcal{A}} Q_t^\pi(X_t, a_t) \tag{7}$$

3 Option Pricing and Hedging

3.1 Black-Scholes Merton Model

Option hedging and pricing under the classical BSM is driven by dynamics of a continuous-time Geometric Brownian motion (GBM) with a drift μ and an instantaneoes standard deviation or, for the purpose of this paper, volatility σ:

$$dS_t = \mu S_t d_t + \sigma S_t dW_t \tag{8}$$

where W_t is a standard Brownian motion.

For a call option $C(S,t)$ with the underlying stock price S at time t, use of Itô's Lemma immediately gives us

$$dC(S,t) = \left(\mu S \frac{\partial C}{\partial S} + \frac{\partial C}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 C}{\partial S^2}\right) dt + \sigma S \frac{\partial C}{\partial S} dW_t \tag{9}$$

With the help of a Taylor expansion [3], the Black-Scholes formula can be expressed as

$$rS_t \frac{\partial C_t}{\partial S_t} + \frac{\partial C_t}{\partial t} + \frac{1}{2}\sigma^2 S_t^2 \frac{\partial^2 C_t}{\partial S_t^2} - rC_t = 0 \tag{10}$$

which can be solved with boundary conditions $C(S,T) = (S-K)_+$, $C(0,t) = 0$ for all t and $C(S,t) \to S$ as $S \to \infty$ for a call option. The solution gives us the famous option pricing formula:

$$\begin{aligned} C(S_t,t) &= N(d_1)S_t - N(d_2)Ke^{-r(T-t)} && \text{for call options, and} \\ P(S_t,t) &= C(S_t,t) + Ke^{-r(T-t)} - S_t && \text{for put options} \end{aligned} \tag{11}$$

where

$$d_1 = \frac{ln(\frac{S_t}{K}) + (r + \frac{\sigma^2}{2})(T-t)}{\sigma\sqrt{T-t}}$$
$$d_2 = d_1 - \sigma\sqrt{T-t}$$

and $N(\cdot)$ is the cumulative distribution function (CDF) of the standard normal distriution.

The other quantity of interest in this paper is a delta hedge, which is defined as the sensitivity of the option price to a change in the price of the underlying asset. Under the BSM model, by definition, and from (11), the theoretical delta hedge is given by:

$$\begin{aligned} \delta_{BS} &= \frac{\partial V}{\partial S} = N(d_1) && \text{for call options, and} \\ \delta_{BS} &= \frac{\partial V}{\partial S} = -N(-d_1) = N(d_1) - 1 && \text{for put options} \end{aligned} \tag{12}$$

3.2 Selected literature review for reinforcement learning studies to option hedging

Halperin (2019) [4] introduces the QLBS approach which combines the Q-Learning with the method of dynamic option replication under the Black-Scholes model. It uses the Q-Learning method for option pricing and hedging based on the risk adjusted returns. This method does not take into account transaction costs and assumes the Black-Scholes assumptions hold except for continuous-time rebalancing. Notably, the optimal hedge, as the action to take in the RL setup, is an argument to the value function which gives the option price. This means that the two quantities of interest can be solved jointly. The implementation result is presented separately by Halperin (2018) [5].

Ritter and Kolm (2019) [6] address the issue of the absence of transac-

tion costs. They do not assume linearity of transaction costs but state that their method is compatible with a transaction cost function of any form. They present a method which could be used for any pricing or simulation method as in Halperin's QLBS approach.

Cao et al. (2019) [7] focus on the presence of transaction costs in option hedging as well, but consider a stochastic volatility, which is modelled by a SABR model introduced by Hagan et al. (2002) [8], instead of a constant volatility assumed in the classical BSM model. Their results indicate a statistically significant improvement in reducing hedging costs and a small but non-significant increase in the variance of the agent's wealth.

4 Problem Setup

Consider a European put option with maturity T and terminal payoff $H_T(S_T) = (K - S_T)_+$ from the holder's point of view. It can be hedged by a replicating (hedge) portfolio Π_t made of the stock S_t and a risk-free bank deposit B_t:

$$\Pi_t = u_t S_t + B_t \tag{13}$$

at any time $t \leq T$, where u_t is the position in the stock at time t taken to hedge risk in the option.

At T, the option holder can choose to exercise it or not, in either case the underlying stock should be cleared out with $u_T = 0$ [9] and thus, there is only cash account in the portfolio which should be valued equivalently with the option payoff under the law of one price, namely

$$\Pi_T = B_T = H_T(S_T) \tag{14}$$

Under the self-financing constraint, we cannot have any cash inflows or outflows between any two rebalancing times, so all future changes in the hedge portfolio should be funded from an initially set bank account [10]. Precisely, the portfolio we have at any time t, when it accumulates to the next time step t+1, the accumulated value should be equivalent to the new rebalanced portfolio at t+1, otherwise there would be an arbitrage. Mathematically,

$$u_t S_{t+1} + e^{r\Delta t} B_t = u_{t+1} S_{t+1} + B_{t+1} \tag{15}$$

which implies

$$B_t = e^{-r\Delta t} \left[B_{t+1} + (u_{t+1} - u_t) S_{t+1} \right] \tag{16}$$

and

$$\Pi_t = e^{-r\Delta t}[\Pi_{t+1} - u_t \Delta S_t] \tag{17}$$

where $\Delta S_t = S_{t+1} - e^{r\Delta t} S_t$ for $t = T-1, ..., 0$.

The position of underlying asset and the amount in cash account depend on the stock price, to evaluate the portfolio value Π_t and the corresponding cash B_t, the paths of underlying stock price S_t is needed. Though historical stock prices can be used, for convenience and feasibility reason, we use Monte Carlo (MC) simulation to generate N_{MC} underlying stock price paths $S_1 \to S_2 \to ... \to S_T$ and then evaluates Π_t backwards on each path.

Under BSM, the underlying stock price S_t follows a GBM as shown in (8) and it implies

$$S_{t+1} = S_t e^{(\mu - \frac{\sigma^2}{2})\Delta t + \sqrt{\Delta t} Z} \tag{18}$$

where $Z \sim N(0,1)$ is a standard Brownian motion.

For simplicity, define an adjusted form state variable X_t to convert the non-stationary S_t to a time-homogeneous one X_t:

$$X_t = -(\mu - \frac{\sigma^2}{2})t + log S_t \tag{19}$$

which implies

$$dX_t = -(\mu - \frac{\sigma^2}{2})dt + dlog S_t = \sigma dW_t \tag{20}$$

In addition, notation wise, the hedging position a_t at each rebalancing point is the action to take in the RL setup, $a_t = a_t(X_t) = u_t(S_t)$, and they can be used interchangeably.

The limitation of BSM model comes from its assumption that the re-hedging is happening continuously at no cost, which is counterfactual in practice. Thus, while a discrete-time hedging is considered in this paper, the hedging goal be-

comes to minimize the hedging risk, which in this case is measured by the variance of the hedge portfolio values across all MC paths [11]. This is because the portfolio is set to mimic the option and its value is expected to be as stable as possible.

4.1 Optimal hedging

Since in practice we do not know the future when we compute a hedge at each time step t, it can only based on the available information set in the sigma filtration dated at time t, $\mathcal{F}_t$. Computed by cross-sectional analysis over all MC simulated paths and backward in time, starting from maturity T, the goal is to find the optimal hedging position $u_t(S_t)$ such that the variance of Π_t across all MC paths can be minimized conditional on $\mathcal{F}_t$, i.e.

$$\begin{aligned} u_t^*(S_t) &= arg\min_u Var(\Pi_t \mid \mathcal{F}_t) \\ &= arg\min_u Var(\Pi_{t+1} - u_t \Delta S_t \mid \mathcal{F}_t) \end{aligned} \tag{21}$$

Solve (21) by setting its first derivative $\frac{\partial u_t^*(S_t)}{\partial u}$ to zero and it gives:

$$u_t^*(S_t) = \frac{Cov(\Pi_{t+1}, \Delta S_t \mid \mathcal{F}_t)}{Var(\Delta S_t \mid \mathcal{F}_t)} \tag{22}$$

4.2 Optimal option pricing

Theoretical fair option price under BSM model $\hat{P}_t$ is the expected value of of hedge portfolio Π_t at time t:

$$\hat{P}_t = \mathbb{E}_t[\Pi_t \mid \mathcal{F}_t] \tag{23}$$

However, in discrete-time hedging setting, the hedging risk that the bank account B_t may fail to cover the requirement of the portfolio needs to be com-

pensated by adding this part of risk premium to the consideration of the fair option price. Here one possible specification of a risk premium is indicated by the cumulative expected discounted variance of the hedge portfolio along all time steps $t = 0, ..., T$, with a risk-aversion parameter λ, i.e.

$$P_0^{(ask)}(S, u) = \mathbb{E}_0 \left[\Pi_0 + \lambda \sum_{t=0}^{T} e^{-rt} Var(\Pi_t \mid \mathcal{F}_t) \mid S_0 = S, u_0 = u \right] \tag{24}$$

The object is to minimize the above fair option price, or equivalently, to maximize the objective value function in the Q-Learning setup $V_t = -C_t^{(ask)}$, i.e.

$$V_t(S_t) = \mathbb{E}_t \left[-\Pi_t - \lambda \sum_{t'=t}^{T} e^{-r(t'-t)} Var(\Pi_{t'} \mid \mathcal{F}_{t'}) \mid \mathcal{F}_t \right] \tag{25}$$

4.3 Value functions and Bellman equations

From (14), rewrite S_t into X_t, and the object is equivalent to maximize the value function

$$\begin{aligned} V_t^{\pi}(X_t) &= \mathbb{E}_t \left[-\Pi_t(X_t) - \lambda \sum_{t'=t}^{T} e^{-r(t'-t)} Var\left(\Pi_{t'}(X_{t'}) \mid \mathcal{F}_{t'}\right) \mid \mathcal{F}_t \right] \\ &= \mathbb{E}_t \left[-\Pi_t(X_t) - \lambda Var\left(\Pi_t(X_t)\right) - \lambda \sum_{t'=t+1}^{T} e^{-r(t'-t)} Var\left(\Pi_{t'}(X_{t'}) \mid \mathcal{F}_{t'}\right) \mid \mathcal{F}_t \right] \end{aligned} \tag{26}$$

over policy $\pi(t, X_t)$ that maps the time t and the current state $X_t = x_t$ into an action $a_t \in \mathcal{A}$ (i.e. the hedging position):

$$a_t = \pi(t, x_t) \tag{27}$$

The last term in (26) can be expressed in terms of V_{t+1} using the definition

of the value function with a shifted time argument:

$$-\lambda \mathbb{E}_{t+1}\left[\sum_{t'=t+1}^{T} e^{-r(t'-t)} Var\left[\Pi_{t'}(X_{t'}) \mid \mathcal{F}_{t'}\right] \mid \mathcal{F}_t\right] = \gamma\left(V_{t+1} + \mathbb{E}_{t+1}[\Pi_{t+1}]\right), \gamma \equiv e^{-r\Delta t} \tag{28}$$

Plug (28) back into (26) and get the Bellman equation for the value function:

$$\begin{aligned} V_t^{\pi}(X_t) &= \mathbb{E}_t\left[-\Pi_t - \lambda Var(\Pi_t) + \gamma(V_{t+1} + \mathbb{E}_{t+1}[\Pi_{t+1}]) \mid \mathcal{F}_t\right] \\ &\overset{(17)}{=} \mathbb{E}_t\left[-e^{-r\Delta t}(\Pi_{t+1} - u_t \Delta S_t) - \lambda Var(\Pi_t) + \gamma V_{t+1} + \gamma \mathbb{E}_{t+1}[\Pi_{t+1}] \mid \mathcal{F}_t\right] \\ &= \mathbb{E}_t\left[-\gamma \Pi_{t+1} + \gamma a_t \Delta S_t - \lambda Var(\Pi_t) + \gamma V_{t+1} + \gamma \mathbb{E}_{t+1}[\Pi_{t+1}] \mid \mathcal{F}_t\right] \\ &= \mathbb{E}_t^{\pi}\left[R(X_t, a_t, X_{t+1}) + \gamma V_{t+1}^{\pi}(X_{t+1})\right] \end{aligned} \tag{29}$$

where

$$R_t(X_t, a_t, X_{t+1}) = \gamma a_t \Delta S_t(X_t, X_{t+1}) - \lambda Var(\Pi_t \mid \mathcal{F}_t), t = T-1, ..., 0 \tag{30}$$

is the one-step time-dependent random reward.

The variance term in (30) can be expressed as

$$\begin{aligned} Var(\Pi_t \mid \mathcal{F}_t) &\overset{(17)}{=} Var\left[e^{-r\Delta t}(\Pi_{t+1} - u_t \Delta S_t) \mid \mathcal{F}_t\right] \\ &= \gamma^2 Var\left[\Pi_{t+1} - u_t \Delta S_t \mid \mathcal{F}_t\right] \\ &= \gamma^2 \mathbb{E}_t\left[\Big((\Pi_{t+1} - \mathbb{E}_t[\Pi_{t+1}]) - (a_t \Delta S_t - \mathbb{E}_t[a_t \Delta S_t])\Big)^2\right] \\ &= \gamma^2 \mathbb{E}_t\left[(\hat{\Pi}_{t+1} - a_t \Delta \hat{S}_t)^2\right] \\ &= \gamma^2 \mathbb{E}_t\left[\hat{\Pi}_{t+1}^2 - 2a_t \Delta \hat{S}_t \hat{\Pi}_{t+1} + a_t^2 (\Delta \hat{S}_t)^2\right] \end{aligned} \tag{31}$$

where $\hat{\Pi}_{t+1} \equiv \Pi_{t+1} - \bar{\Pi}_{t+1}$ with $\bar{\Pi}_{t+1}$ is the sample mean of Π_{t+1} over all MC paths, and similarly $\Delta \hat{S}_t = \Delta S_t - \Delta \bar{S}_t$. Thus, plugging (31) back into (30) will

further derive the reward function as:

$$R_t(X_t, a_t, X_{t+1}) = \gamma a_t \Delta S_t - \lambda \gamma^2 \mathbb{E}_t \left[\hat{\Pi}_{t+1}^2 - 2 a_t \Delta \hat{S}_t \hat{\Pi}_{t+1} + a_t^2 (\Delta \hat{S}_t)^2 \right] \quad (32)$$

The expected reward, consequently, is given by

$$\mathbb{E}_t \left[R_t(X_t, a_t, X_{t+1}) \right] = \gamma a_t \mathbb{E}_t[\Delta S_t] - \lambda \gamma^2 \mathbb{E}_t \left[\hat{\Pi}_{t+1}^2 - 2 a_t \Delta \hat{S}_t \hat{\Pi}_{t+1} + a_t^2 (\Delta \hat{S}_t)^2 \right] \quad (33)$$

as a quadratic function of the action variable a_t. And it is known that at maturity $t = T$, we have the terminal condition $\Pi_T = B_T = H_T(S_T)$, thus $R_T = -\lambda Var(\Pi_T)$.

Similarly, the action-value function, known as the Q-function in the RL setting, is defined in the same way as the value function, but conditioned on both the current state $X_t = x$ and the initial action $a_t = a$, while following a policy π afterwards:

$$Q_t^\pi(x, a) = \mathbb{E}_t \left[-\Pi_t(X_t) \mid X_t = x, a_t = a \right] - \lambda \mathbb{E}_t^\pi \left[\sum_{t'=t}^{T} e^{-r(t'-t)} Var[\Pi_{t'}(X_{t'}) \mid \mathcal{F}_{t'}] \mid X_t = x, a_t = a \right] \quad (34)$$

By now, the RL setup of the BSM model is formulated as - finding the optimal policy $\pi_t^*(X_t)$ which minimizes the value function $V_t^\pi(X_t)$, or equivalently, the action-value function $Q_t^\pi(X_t, a_t)$:

$$\pi_t^*(X_t) = arg \max_\pi V_t^\pi(X_t) = arg \max_{a_t \in \mathcal{A}} Q_t^\pi(X_t, a_t) \quad (35)$$

with the corresponding optimal value function satisfies the Bellman optimality equation:

$$V_t^*(X_t) = \mathbb{E}_t^{\pi^*} \left[R(X_t, u_t = \pi_t^*(X_t), X_{t+1}) + \gamma V_{t+1}^*(X_{t+1}) \right] \quad (36)$$

and the Bellman optimality equation for the action-value function:

$$Q_t^*(x,a) = \mathbb{E}_t\left[R_t(X_t,a_t,X_{t+1}) + \gamma \max_{a_{t+1}\in\mathcal{A}} Q_{t+1}^*(X_{t+1},a_{t+1}) \mid X_t = x, a_t = a\right], t = T-1,...,0 \tag{37}$$

where the terminal condition at $t = T$ again, is given by

$$Q_T^*(X_T, a_T = 0) = -\Pi_T(X_T) - \lambda Var[\Pi_T(X_T)] \tag{38}$$

5 Dynamic Programming Solution for QLBS

5.1 Formulation

The Markov decision process in this setup is to solve the Bellman optimality equation for action-value function (37) jointly with the optimal policy (35) together, starting from the terminal condition (38) at $t = T$ and go backward recursively. To solve, substitute (33) into (37):

$$\begin{aligned} Q_t^*(X_t, a_t) &= \mathbb{E}_t\left[\gamma a_t \Delta S_t\right] - \lambda\gamma^2 \mathbb{E}_t\left[\hat{\Pi}_{t+1}^2 - 2a_t\Delta\hat{S}_t\hat{\Pi}_{t+1} + a_t^2(\Delta\hat{S}_t)^2\right] + \gamma\mathbb{E}_t\left[Q_{t+1}^*(X_{t+1}, a_{t+1}^*)\right] \\ &= \gamma\mathbb{E}_t\left[Q_{t+1}^*(X_{t+1}, a_{t+1}^*) + a_t\Delta S_t\right] - \lambda\gamma^2\mathbb{E}_t\left[\hat{\Pi}_{t+1}^2 - 2a_t\Delta\hat{S}_t\hat{\Pi}_{t+1} + a_t^2(\Delta\hat{S}_t)^2\right], t = T-1, ..., 0 \end{aligned} \tag{39}$$

which is quadratic in a_t, since the very first term inside the first expectation $Q_{t+1}^*(X_{t+1}, a_{t+1}^*)$ is independent of a_t, under the assumption that any of our action a_t will not be large enough to impact the market. Thus, the optimal action $a_t^*(X_t)$ that maximizes $Q_t^*(X_t, a_t)$ can be solved analytically by setting its first derivative $\frac{\partial Q_t^*(X_t, a_t)}{\partial a_t}$ to zero, which gives the optimal action to take at time t

$$a_t^*(X_t) = \frac{\mathbb{E}_t\left[\hat{\Pi}_{t+1}\Delta\hat{S}_t + \frac{1}{2\gamma\lambda}\Delta S_t\right]}{\mathbb{E}_t\left[\left(\Delta\hat{S}_t\right)^2\right]} \tag{40}$$

Plug (40) back to (39) to get an explicit recursive formula for the optimal action-value function:

$$Q_t^*(X_t, a_t^*) = \gamma\mathbb{E}_t\left[Q_{t+1}^*(X_{t+1}, a_{t+1}^*) - \lambda\gamma\hat{\Pi}_{t+1}^2 + \lambda\gamma(a_t^*(X_t))^2(\Delta\hat{S}_t)^2\right], t = T-1, ..., 0 \tag{41}$$

Given the terminal condition at T and proceeds analytically by (40) and (41) jointly, all the way backward starting from $t = T - 1$ to the present $t = 0$, the optimal hedging at each time step is given by a_t^*, while the optimal

option price in this setting, referred to as the QLBS option price, is given by $P_t^{(QLBS)}(S_t, ask) = -Q_t^*(S_t, a_t^*)$.

5.2 Implementation

For implementation, we do have access to all the of the N_{MC} paths of the state variable X_t simulated in the Monte Carlo setting, thus all terms involved in (40) and (41) for analytical calculation is available by looking at all the scenarios over N_{MC} paths at time t and $t+1$ simultaneously. Ideally, this is how we solve for the optimal policy a_t^* and the optimal Q-function Q_t^* analytically. However, this is computationally infeasible. Instead, spline basis is used for approximation. We choose a set of basis functions $\Phi_n(x)$ and expand the optimal action (hedge) $a_t^*(X_t)$ and optimal Q-function $Q_t^*(X_t, a_t^*)$ in basis functions, with time-dependent coefficients:

$$a_t^*(X_t) = \sum_n^N \phi_{nt} \Phi_n(X_t) \tag{42}$$

$$\text{and } Q_t^*(X_t, a_t^*) = \sum_n^N \omega_{nt} \Phi_n(X_t) \tag{43}$$

Coefficients ϕ_{nt} and ω_{nt} are computed recursively backward in time for $t = T-1, ..., 0$:

$$\phi_t^* = A_t^{-1} B_t \tag{44}$$

where

$$A_{nm}^t = \sum_{k=1}^{N_{MC}} \Phi_n\left(X_t^k\right) \Phi_m\left(X_t^k\right) \left(\Delta \hat{S}_t^k\right)^2 \tag{45}$$

$$B_n^t = \sum_{k=1}^{N_{MC}} \Phi_n\left(X_t^k\right) \left[\hat{\Pi}_{t+1}^k \Delta \hat{S}_t^k + \frac{1}{2\gamma\lambda} \Delta S_t^k\right] \tag{46}$$

and

$$\omega_t = C_t^{-1} D_t \tag{47}$$

where

$$C_{nm}^t = \sum_{k=1}^{N_{MC}} \Phi_n\left(X_t^k\right) \Phi_m\left(X_t^k\right) \tag{48}$$

$$D_n^t = \sum_{k=1}^{N_{MC}} \Phi_n(X_t^k) \left(R_t(X_t^k, a_t^{k*}, X_{t+1}^k) + \gamma \max_{a_{t+1} \in \mathcal{A}} Q_{t+1}^*(X_{t+1}^k, a_{t+1}) \right) \tag{49}$$

When implementing in Python with the MC simulated paths and selected B-spline basis functions, the following steps are taken:

1) Compute the optimal hedge a_t^* and the corresponding portfolio value Π_t (one optimal set for each path): Starting from the terminal condition at T with $a_T = 0$ and $\Pi_T = H_T(S_T) = payoff(S_T) = (K - S_T)_+$ for a European put option, by a backward recursion from time $t = T - 1$ to $t = 0$, in each iteration, calculate A_t, B_t, ϕ_t, $a_t^*(X_t)$ by (42), and $\Pi_t = \gamma \left[\Pi_{t+1} - a_t^* \Delta S_t\right]$.

2) Compute rewards for all paths: Starting from the terminal condition at T with $R_T = -\lambda Var(\Pi_T)$, by backward recursive from time $t = T - 1$ to $t = 0$, in each iteration, calculate $R_t(X_t, a_t, X_{t+1})$ by equation (32), where the last variance term is calculated over all MC paths.

3) Compute the optimal Q-function thus the QLBS option price: Starting from the terminal condition at T with $Q_T(X_T, a_T = 0) = -\Pi_T(X_T) - \lambda Var(\Pi_T(X_T)))$, by backward recursive from time $t = T - 1$ to $t = 0$, in each iteration, calculate C_t, D_t, ω_t, and $Q_t^*(X_t, a_t^*)$ by equation (43), where the last variance term is calculated over all MC paths. All the way to $t = 0$ and $P_0^{DP}(S_0, ask) = -Q_0^*(S_0, a_0^*)$.

5.3 Implementation Result

The state variables X_t and the corresponding stock prices S_t are simulated $N_{MC} = 50000$ times with the parameters: initial stock price $S_0 = 100$, stock drift $\mu = 0.05$, and volatility $\sigma = 0.15$. An "at-the-money" (ATM) European put option is set up with maturity $M = 1$, strike price $K = 100$ and risk-free interest rate $r = 0.03$. Rehedges are done bi-weekly (i.e. $T = 24$ with time interval $\Delta t = 1/24$). Twelve basis functions are chosen to be cubic B-splines on a range of smallest and largest values of the simulated X_t. Risk aversion parameter λ is picked as 0.001.

With this numerical setup, the option price calculated directly from the BSM model is 4.53, while the resulting QLBS optimal put option price is higher as expected, taking the hedging risk into consideration, around 5.02 from the MC simulation.

Figure 2 shows 10 of the 50000 simulated paths with their optimal action a_t^*, optimal portfolio value Π_t, rewards R_t and optimal DP Q-function Q_t^* at each re-balance time step.

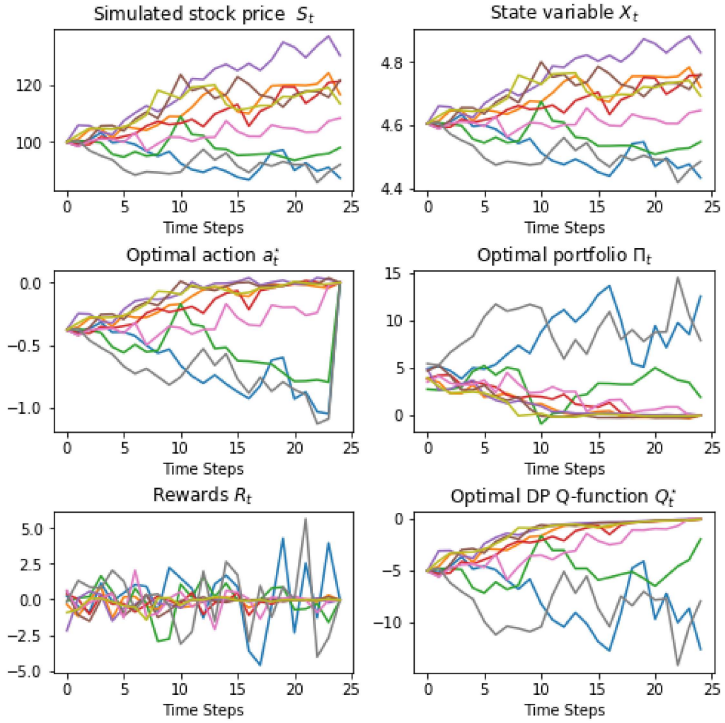

Figure 2: DP solution for the ATM put option on a sub-set of MC paths

6 Fitted Q Iteration Solution for QLBS

6.1 Formulation

Compared to the DP solution where the transition probabilities and reward functions are assumed to be given, and thus optimal hedging position is calculated iteratively, in the Q-Learning setup, optimal policy needs to be found relying on samples. At each time step t, the available information is $\mathcal{F}_t^{(n)} = \{X_t^{(n)}, a_t^{(n)}, R_t^{(n)}, X_{t+1}^{(n)}\}$, namely N_{MC} set of the state variable X_t, the corresponding hedge position a_t, the instantaneous reward R_t and the value of the state variable at next time step X_{t+1}. Instead of making any assumptions on the data-generating process, the information $\mathcal{F}_t^{(n)}$ at each time step t is simply treated as given. Thus the action-value function Q_t^* can be updated according to $\mathcal{F}_t^{(n)}$.

6.2 Implementation

Under the FQI scheme, for computational feasibility, once again we consider the use of a spline basis. Similar to the DP solution, the values to our interest are the optimal action $a_t^*(X_t)$ and the optimal action-value function $Q_t^*(X_t, a_t^*)$, they are represented in a parametric forms. Using the same set of basis functions $\Phi_n(x)$ as in the DP solution, since we have already showed in equation (39) that

$Q_t^*(X_t, a_t)$ is quadratic in a_t, the representation is:

$$Q_t^*(X_t, a_t) = \left(1, a_t, \frac{1}{2}a_t^2\right) \begin{pmatrix} W_{11}(t) & W_{12}(t) & ... & W_{1M}(t) \\ W_{21}(t) & W_{22}(t) & ... & W_{2M}(t) \\ W_{31}(t) & W_{32}(t) & ... & W_{3M}(t) \end{pmatrix} \begin{pmatrix} \Phi_1(X_t) \\ ... \\ \Phi_M(X_t) \end{pmatrix}$$

$$\equiv \mathbf{A}_t^T \mathbf{W}_t \mathbf{\Phi}(X_t) \equiv \mathbf{A}_t^T \mathbf{U}_W(t, X_t) \tag{50}$$

$$= \sum_{i=1}^{3} \sum_{j=1}^{M} \left(\mathbf{W}_t \odot \left(\mathbf{A}_t \otimes \Phi^T(X)\right)\right)_{ij}$$

$$= \vec{\mathbf{W}}_t \cdot vec\left(\mathbf{A}_t \otimes \mathbf{\Phi}^T(X)\right) \equiv \vec{\mathbf{W}}_t \vec{\mathbf{\Psi}}(X_t, a_t) \tag{51}$$

where $\odot$ stands for element-wise product and $\otimes$ stands for tensor product, $\mathbf{W}_t$ is the time-dependent coefficients matrix and concatenating its columns converts it to the vector $\vec{\mathbf{W}}_t$, and similarly $\vec{\mathbf{\Psi}}(X_t, a_t) = vec(\mathbf{A}_t \otimes \mathbf{\Phi}^T(X))$ is also obtained by concatenating columns of the outer product of $\mathbf{A}_t$ and $\mathbf{\Phi}^T(X)$.

Coefficients $\vec{\mathbf{W}}_t$ is computed recursively backward in time from $t = T-1$ to $t = 0$:

$$\vec{\mathbf{W}}_t^* = \mathbf{S}_t^{-1} \mathbf{M}_t \tag{52}$$

where

$$S_{nm}^{(t)} = \sum_{k=1}^{N_{MC}} \Psi_n\left(X_t^k, a_t^k\right) \Psi_m\left(X_t^k, a_t^k\right) \tag{53}$$

$$M_n^{(t)} = \sum_{k=1}^{N_{MC}} \Psi_n\left(X_t^k, a_t^k\right) \left(R_t(X_t^k, a_t^k, X_{t+1}^k) + \gamma \max_{a_{t+1} \in \mathcal{A}} Q_{t+1}^*(X_{t+1}^k, a_{t+1})\right) \tag{54}$$

and

$$Q_{t+1}^*(X_{t+1}, a_{t+1}^*) = \mathbf{U}_W^{(0)}(t+1, X_{t+1}) + a_{t+1}^* \mathbf{U}_W^{(1)}(t+1, X_{t+1}) + \frac{(a_{t+1}^*)^2}{2} \mathbf{U}_W^{(2)}(t+1, X_{t+1}) \tag{55}$$

When implementing in Python with the MC simulated paths and selected B-spline basis functions, the following steps are taken:

1) Get optimal actions and rewards for all paths: In addition to the MC simulated stock price paths, the optimal actions a_t^* and rewards R_t from DP solution are taken as to the FQI solution. For an off-policy algorithm [12], where instead of taking the optimal action directly, some noise are added to the optimal action, so that we can explore more possible options for the actions to take in the future. To add the noise, randomly generate a set of uniformed distributed numbers in the interval $[1-\eta, 1+\eta]$ where $0 < \eta < 1$ and multiplied to the optimal each a_t^*.

2) Construct $\mathbf{A}$ and then compute $\vec{\mathbf{\Psi}}$.

3) Compute the optimal Q-function thus the QLBS option price: Starting from the terminal condition at T with $Q_T(X_T, a_T = 0) = -\Pi_T(X_T) - \lambda Var(\Pi_T(X_T)))$, by backward recursive from time $t = T-1$ to $t = 0$, in each iteration, calculate S_t, M_t, $Q_{t+1}^*(X_{t+1}, a_{t+1}^*)$, $\vec{\mathbf{W}}_t^*$, and $Q_t^*(X_t, a_t)$ by equation (51). All the way to $t = 0$ and $P_0^{FQI}(S_0, ask) = -Q_0^*(S_0, a_0^*)$.

6.3 Implementation result

Using the same parameters as the DP implementation in Section 5.3, with the noise parameter = 0.5. While the option price calculated directly from the BSM model is 4.53 and from the DP solution is 5.02, the resulting solution from FQI method is a bit higher around 5.08 from the MC simulation.

Figure 3 shows 10 of the 50000 simulated paths with their optimal action a_t^*, optimal portfolio value Π_t, rewards R_t and optimal DP Q-function Q_t^* at each re-balance time step.

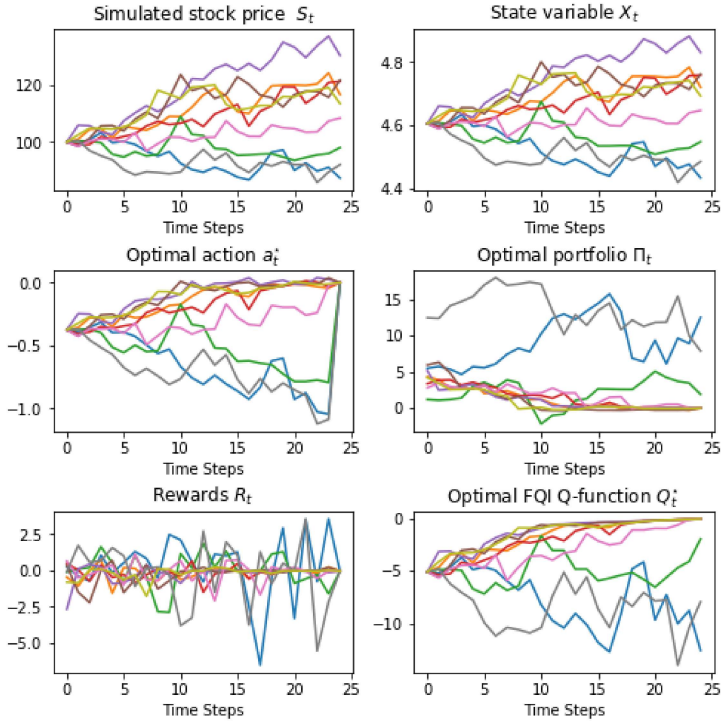

Figure 3: FQI solution for the ATM put option on a sub-set of MC paths

7 Result Discussion and Sensitivity Analysis

7.1 Delta hedging position comparison

In Section 3.1, it is shown that the delta hedge for a European put option under the BSM model is $\delta_{BS} = \frac{\partial V}{\partial S} = -N(-d_1) = N(d_1) - 1$, where $N(d_1)$ is the CDF for a normal distribution whose value is in $[0, 1]$, thus, the delta hedging position for a European put option under BSM is always between -1 (sell 1 share of underlying asset) and 0 (holding no underlying asset).

If we look at a fixed time, the hedging position should depends only on the underlying stock price. With the BSM delta hedge known theoretically, we set it as a benchmark and compare the delta hedge result from DP and FQI solution to it.

Figure 4 shows the comparison among BSM, DP and FQI delta hedge at inception $t = 0$, $t = M/4$, $t = M/2$, $t = 3M/4$, and maturity $t = M$.

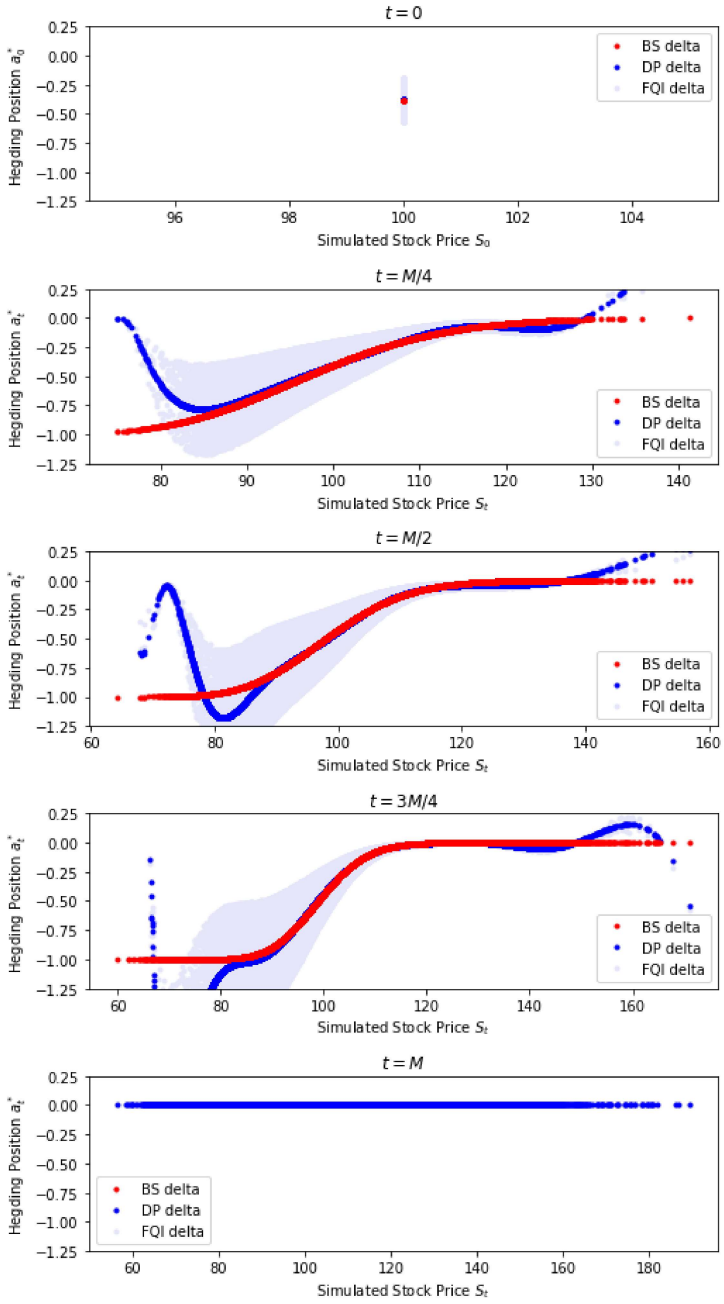

Figure 4: Comparison among DP, FQI and BSM hedging position vs stock price at fixed time steps

It is demonstrated that both RL solutions give a very good approximation of the BSM delta, especially when the stock price is closer to the initial price which is also the strike price in this ATM setup. Additionally, The approximation gets better as the option goes closer to maturity. Moreover, FQI solution yields a wider range of hedging position because an off-policy algorithm is used which adds some noise to the action value.

7.2 Option price comparison

Similarly, if we look at a fixed time, the option price should depends only on the underlying stock price. With the BSM being set as a benchmark, we compare the option pricing result from DP and FQI solution to it.

Figure 5 shows the comparison among BSM, DP and FQI option price at inception $t = 0$, $t = M/4$, $t = M/2$, $t = 3M/4$, and maturity $t = M$.

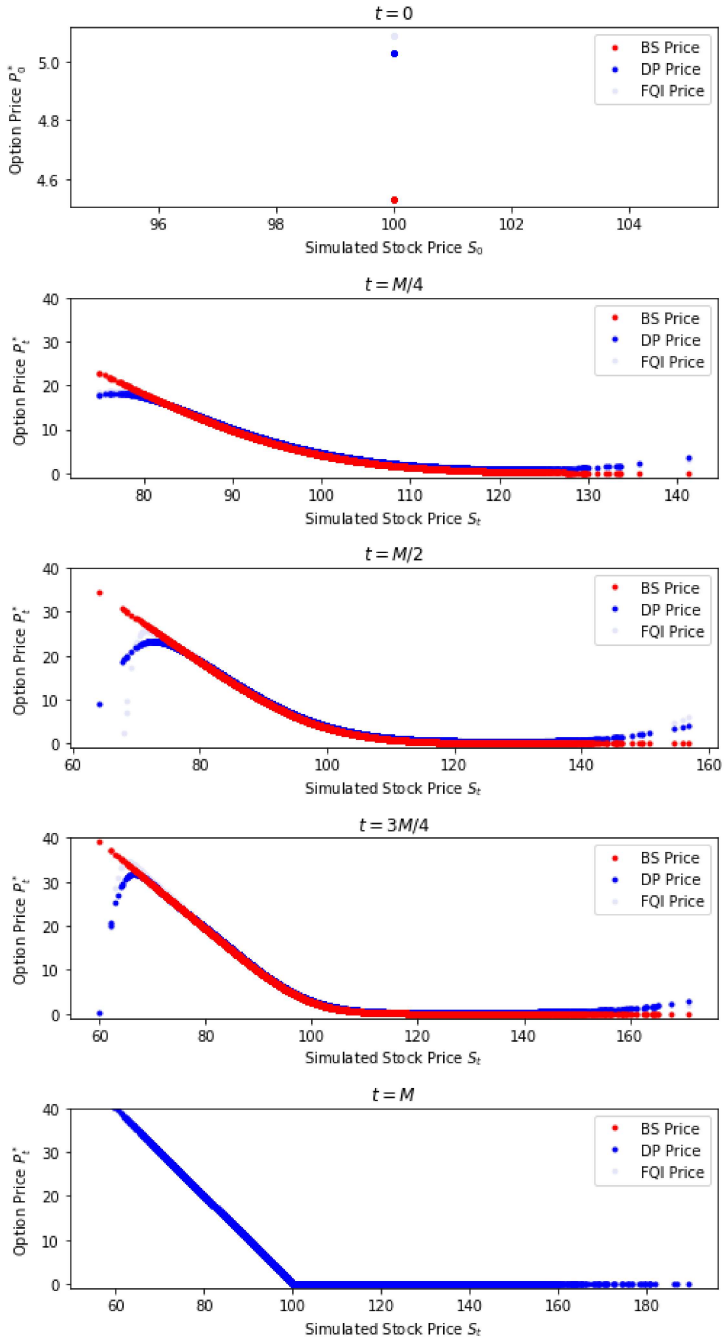

Figure 5: Comparison among DP, FQI and BSM option pricing vs stock price at fixed time steps

Once again, it is demonstrated that both RL solutions give a very good approximation of the BSM price, especially when the stock price is closer to the initial price which is also the strike price in this ATM setup. Furthermore, The approximation gets better as the option goes closer to maturity.

7.3 Risk aversion parameter

As defined in equation (24), the risk aroused from mis-hedging is compensated by adding them to the option price with a risk averse parameter λ. It is obvious from the definition and the equation that a larger λ adds more weights to the cumulative risk, which means the investor is more risk averse, and will lead to a higher option price.

Figure 6 shows the option pricing result from the DP solution against different value of risk averse λ, compared to the BSM price as a benchmark. The result is consistent with the above analysis from the equation.

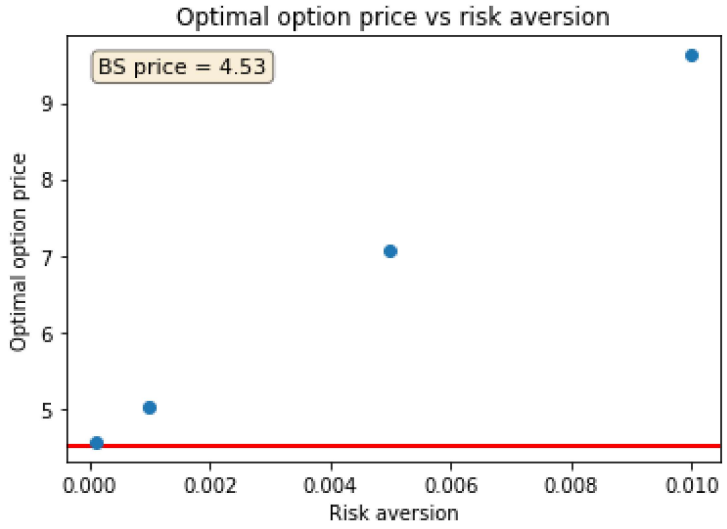

Figure 6: The ATM put option price from DP solution compared to BSM vs risk aversion parameter λ

7.4 Hedging frequency

We have discussed the result from bi-weekly hedging in detail, for analysis purpose, different hedging frequency are performed with all others remain the same and we see how RL works.

Figure 7 shows the option price from DP and FQI solution from a one-time hedging ($T = 2$), monthly hedging ($T = 12$), bi-weekly hedging ($T = 24$) and weekly hedging ($T = 52$), and the results are compared to BSM price as a benchmark.

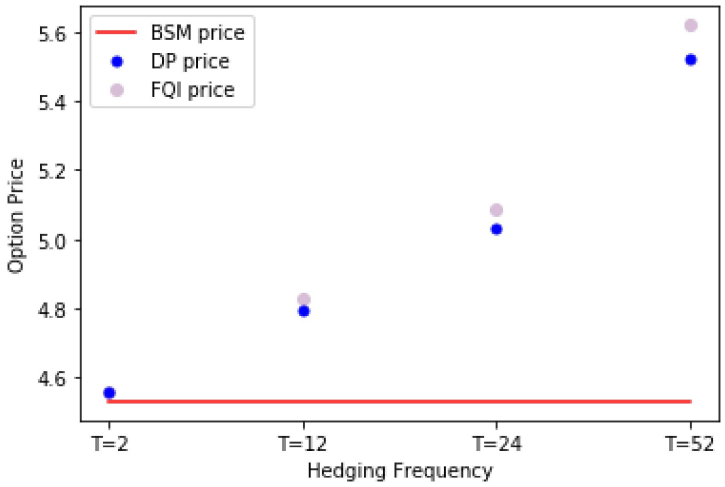

Figure 7: The ATM put option price from BSM, DP and FQI solution vs different hedging frequency

As illustrated, both RL solutions yield a higher price compared to BSM price, this has been explained before that as mis-hedging risk is taken into consideration it needs to be compensated with a higher expected return. Moreover, it is noticed that the FQI price is always slightly higher than the DP price, this again is because instead of the analytically optimal action, in the off-policy FQI algorithm, some noise is added so that we can explore more action to take for

future benefits.

The interesting point is, as the hedging goes more frequently, the RL prices also get higher. This can be explained by the pricing function (13), as the accumulated variance could be larger when the frequency becomes higher. However, on the other hand, if we keep increasing the hedging frequency and as the time interval between consecutive rebalancing time eventually becomes infinitely small, the result should be approaching to BSM price. The implementation here does not show this part, one possible reason might be the frequency performed is still not high enough due to computational feasibility. As the hedging becomes daily, even hourly or secondly, there might be a point where the option price from RL methods starts to decrease and approach to the BSM model, as the variance starts to drop. This is subject to further research.

8 Summary

8.1 Conclusion

In this paper we have investigated the performance of the RL approach when it is applied to option hedging and pricing. In particular, the data used is generated from a Monte Carlo simulation which gets rid of extra assumptions or noises, and the implementation are done for both the dynamic programming solution and the fitted Q iteration solution. The results are then analyzed and contrasted to the the results obtained from the classical Black-Scholes model which is used as a benchmark.

We first presented a general background knowledge of RL and option pricing and hedging. Then we conducted some review of the selected literature. As the next step, we formulated the problem of interest in a mathematical setup and conduct implementation on two solutions within the QLBS approach, namely the dynamic programming solution and the fitted Q iteration solution. The data used is synthetically generated from a Monte Carlo simulation which gets rid of extra assumptions or noises. And the results are analyzed and contrasted to the results obtained from the classical BSM model which is used as a benchmark. Both models are shown to produce reasonably good performances, with the dynamic programming model being more stable while the fitted Q iteration one providing greater potential for further research.

8.2 Future work

Based on the current research, further investigation can be extended in the following areas.

Firstly, due to data availability and computational feasibility, the implementation in this research is only done with the simulated data, and is up to

a weekly frequency. However, this method should also work for market data since there is no certain constraint or any underlying assumptions made about the data. Instead it is completely model-free and entirely dependent on the sample data. Further experiments can be carried out with historical data and the performance could be compared to the real hedge and price in the financial market. Additionally, a higher frequency of hedging can also be conducted.

Secondly, the formulation of the value functions in this paper only provides one possible setup; other forms of elaboration with the potential to better minimize the mis-hedging risk are worthwhile investigating in the future.

Lastly, in this paper, the exploration of the optimal action is conducted by an off-policy algorithm where some noise is added to the greedy action. For potential improvements, more exploration techniques such as an ϵ-greedy method and an upper-confidence-bound action selection method could be implemented in the future.

References

[1] David Silver et al. "Deterministic Policy Gradient Algorithms". In: *Proceedings of the 31st International Conference on Machine Learning*. Ed. by Eric P. Xing and Tony Jebara. Vol. 32. Proceedings of Machine Learning Research 1. Bejing, China: PMLR, 22–24 Jun 2014, pp. 387–395. URL: http://proceedings.mlr.press/v32/silver14.html.

[2] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: A Bradford Book, 2018. ISBN: 0262039249.

[3] Martin Haugh. *The Black-Scholes Model*. 2016. URL: http://www.columbia.edu/~mh2078/FoundationsFE/BlackScholes.pdf.

[4] Igor Halperin. *QLBS: Q-Learner in the Black-Scholes(-Merton) Worlds*. 2019. arXiv: 1712.04609 [q-fin.CP].

[5] Igor Halperin. *The QLBS Q-Learner Goes NuQLear: Fitted Q Iteration, Inverse RL, and Option Portfolios*. 2018. arXiv: 1801.06077 [q-fin.CP].

[6] Gordon Ritter and Petter Kolm. "Dynamic Replication and Hedging: A Reinforcement Learning Approach". In: *SSRN Electronic Journal* (Jan. 2019). DOI: 10.2139/ssrn.3281235.

[7] Jay Cao et al. "Deep Hedging of Derivatives Using Reinforcement Learning". In: (Dec. 2019). DOI: 10.2139/ssrn.3514586.

[8] Patrick Hagan et al. "Managing Smile Risk". In: *Wilmott Magazine* 1 (Jan. 2002), pp. 84–108.

[9] Emanuel Derman, Deniz Ergener, and Iraj Kani. "Static Options Replication". In: *Journal of Derivatives* 2 (Nov. 2000). DOI: 10.3905/jod.1995.407927.

[10] Robert (Bob) Korkie and Harry Turtle. "A Mean-Variance Analysis of Self-Financing Portfolios". In: *Management Science* 48 (Mar. 2002), pp. 427–443. DOI: 10.1287/mnsc.48.3.427.7725.

[11] Martin Schweizer. "Variance-Optimal Hedging in Discrete Time". In: *Mathematics of Operations Research* 20.1 (1995), pp. 1–32. ISSN: 0364765X, 15265471. URL: http://www.jstor.org/stable/3690105.

[12] Matthew F. Dixon, Igor Halperin, and Paul Bilokon. *Machine Learning in Finance From Theory to Practice*. Springer, Cham, 2019. ISBN: 978-3-030-41068-1.