

Optimal Investment With Discretionary Stopping And Trading Constraints

by Xiyu Yang

Submission date: 10-Sep-2019 02:41PM (UTC+0100)

Submission ID: 110634177

File name: Yang_Xiyu_01362660.pdf (555.98K)

Word count: 16527

Character count: 72242

**OPTIMAL INVESTMENT WITH
DISCRETIONARY STOPPING AND
TRADING CONSTRAINTS**

by

Yang Xiyu (CID: 01362660)

Department of Mathematics
Imperial College London
London SW7 2AZ
United Kingdom

Thesis submitted as part of the requirements for the award of the
MSc in Mathematics and Finance, Imperial College London, 2017-2018

IMPERIAL COLLEGE LONDON

THESIS DECLARATION FORM

(for candidates whose degree will be awarded by Imperial College)


Declaration: I hereby confirm that the work contained in this thesis is my own work unless other wises stated.

Name.....Yang...Xiyu..... **CID**01362660.....

Title of Thesis.. Optimal Investment With Discretionary Stopping And Trading Constraints....

Month and year of submission for examination.....September...2019.....

Date.....10th...September...2019.....

Signature of Candidate..........

Contents

1	Introduction	4
2	Optimal Investment Problem and HJB formula	4
2.1	General Optimal Control Problem	4
2.2	HJB formula and Value Functions	5
3	Methods for Solving HJB formula	6
3.1	Close form	6
3.1.1	Using Candidate Solution	6
3.1.2	Using Dual Function and Turnpike Property	8
3.2	Numerical Program	8
3.2.1	Problem Formulation	8
3.2.2	Boundary Conditions	9
3.2.3	Discretization of Equation	10
4	Convergence to the viscosity solution	12
5	Examples by Using Numerical Method	14
5.1	Merton portfolio problem with boundary control	14
5.2	Turnpike problem 1 with boundary control	17
5.3	Turnpike problem 2 with boundary control	20
5.4	Uncertain Volatility	22
5.5	Unequal Borrowing/Lending Rates	26
5.6	Stock Borrowing Fees	31
5.7	Mean-Variance Asset Allocation	34
5.8	Mean-Variance Asset Allocation (Heston model)	40
5.8.1	Problem Formulation	40
5.8.2	Convergence to the Viscosity Solution	44
5.8.3	Numerical Results	45
A	Appendix	47
B	Appendix	47
	Conclusion	48

1 Introduction

In the real finance market, many models are formulated as nonlinear partial differential equations (PDEs) with optimal stochastic controls. These special PDEs are called Hamilton-Jacobi-Bellman (HJB) PDEs and the controls are from realistic constraint conditions. It will be the best if analytical solutions can be found for HJB PDEs. However, for now, there are only a few cases getting close-form solutions and most cases are usually solved by using a numerical method. According to some research papers[1, 2, 3], we get this powerful numerical theory which can guarantee to convergence to the viscosity solutions [4] of HJB PDEs.

In this report, we will apply this numerical approach, which can be found in 3.2, and C++ to solve optimal control problems. By comparing numerical results with analytic ones in the first three examples in section 5, we find that this numerical method can generate quite accurate answers. To prove that it is effective for different kinds of optimal control problems, we then choose several other examples which are without close-form solutions and solve them with different numerical schemes. The final data from these schemes are really close, which ensure that the numerical method 3.2 can be widely used.

The examples in this report involves various kinds of optimal control problems, not only one-dimensional situation, but also multi-dimensional situation. The example, Stock Borrowing Fees, in section 5.6 contains three controls. Example 5.3 is with infinite investment period. Moreover, example 5.7 allows bankruptcy as well as unbounded control. And the last example, Mean-Variance Asset Allocation (Heston model), in section 5.8, talks about the situation which is with two stochastic variables.

Section 2 introduces optimal control problems and HJB PDEs. The detailed numerical method and its theoretical convergence proof are given in section 3 and section 4 respectively. Section 5 shows specific examples and their numerical results.

2 Optimal Investment Problem and HJB formula

We are going to give a specific case to show the concepts of optimal control problems and HJB formulas[6, 7].

2.1 General Optimal Control Problem

We assume that there is a no-arbitrage and self-financing financial market which consists of one bank account (or bond) and one stock. And this market allows continuously trading on these two assets. One investor want to do an investment in this market during a finite period T . The price

process of the riskless asset, the bank account B , is as below

$$\frac{dB_t}{B_t} = rdt, \quad 0 \leq t \leq T \quad (2.1)$$

where the riskless interest rate r is a positive constant. We do not consider receiving dividend from stock S and then have the price process $(S_t)_{t \leq 0}$ like below

$$\frac{dS_t}{S_t} = \mu(t)dt + \sigma(t)dW_t, \quad 0 \leq t \leq T, \quad S_0 = s \quad (2.2)$$

where $\mu(t)$ and $\sigma(t)$ are continuous functions of time, which representing stock returns and volatilities. $(W_t)_{t \leq 0}$ is a standard Brownian motion, which is on a complete probability space $(\Omega, \mathcal{F}, \mathbb{P})$, endowed with a natural filtration \mathcal{F} , augmented by all \mathbb{P} -null sets.

Let π_t represent the proportion of wealth invested in the risky asset S and $1 - \pi_t$ is the proportion to invest into the riskless asset B . The the total wealth process $(X_t)_{t \leq 0}$ satisfies the following stochastic differential equation (SDE)

$$\frac{dX_t}{X_t} = (\pi_t(\mu(t) - r) + r)dt + \pi_t\sigma(t)dW_t + c_t d_t, \quad 0 \leq t \leq T, \quad X_0 = x \quad (2.3)$$

where c_t is the consuming or additional contribution rate at time t . To be more precise, $c_t < 0$ for consumption and $c_t > 0$ for contribution. Meanwhile, π_t is the control for this problem and it satisfies $\pi_t \in \mathcal{K}$ a.s. for $t \in [0, T]$ a.e. \mathcal{K} comes from real constrains and is a closed convex cone in \mathbb{R} . Usually, there are two choices for \mathcal{K} . One is called bounded control which means that $\mathcal{K} = [\pi_{min}, \pi_{max}] \subseteq \mathbb{R}$. The other is unbounded control meaning that $\mathcal{K} = \mathbb{R}$.

Note that, in this report, we write deterministic functions as the form of $b(t)$ and $\sigma(t)$ and stochastic processes as S_t and π_t .

To assess the investment over the period T , we need define a utility function. As the purpose of the investor is maximizing or minimizing the utility function, then assuming that it is maximization case, we have

$$\sup_{\pi \in \mathcal{K}} E[U(X_T)] \quad (2.4)$$

where U is a utility function. For the minimization case, it just needs to change *sup* into *inf*. Actually, by adding a minus at the beginning of the formula, we can then transfer a *inf* case into a *sup* case. What said above is a specific optimal control problem. Apart from this, there are many other optimal control problems which can be derived by the similar process, you can find some of them in section 5 and [7].

2.2 HJB formula and Value Functions

The assessment of investment can be divided into two parts. One is the assessment of the consumption during the investment period. The other is about the utility of the rest money at maturity time. We let $V(t, x) : T \times \mathbb{R} \rightarrow \mathbb{R}$ be a $C^{1,2}$ function, which represent the optimal value of (2.4). To

express (2.4) specifically, we divide it into two situations, based on if this problem consider about time value.

- When it contains time value, we can apply Feynman-Kac Formula [7] to the value function and have

$$V(t, x) = \sup_{\pi \in \mathcal{K}} E \left[\int_t^T e^{-\int_t^s r(u, X_u^{t,x}) du} f(s, X_s^{t,x}) ds + e^{-\int_t^T r(u, X_u^{t,x}) du} g(X_T^{t,x}) \right] \quad (2.5)$$

where $r(t, x)$ reflects the instantaneous interest rate. $f(s, X_s^{t,x})$ represents the utility of consumption during the investing period and $g(X_T^{t,x})$ shows the utility of final wealth.

The corresponding nonlinear HJB equation is as following

$$\begin{aligned} \frac{\partial V}{\partial t} + \sup_{\pi \in \mathcal{K}} [\mathcal{A}^\pi V - rV + f] &= 0 \\ V(T, x) &= U(x) \end{aligned} \quad (2.6)$$

where $\mathcal{A}^\pi V \equiv a(t, x, \pi)V_{XX}(t, x) + b(t, x, \pi)V_X(t, x)$ is the partial differential operator.

- But when time value is not considered, the value function[7] are as

$$V(t, x) = \sup_{\pi \in \mathcal{K}} E \left[\int_t^T F(s, X_s^{t,x}) ds + \Phi(X_T^{t,x}) \right] \quad (2.7)$$

where $F(s, X_s^{t,x})$ shows the utility of consumption and $\Phi(X_T)$ is the rest utility of the wealth at maturity T .

The corresponding nonlinear HJB equation is like

$$\begin{aligned} \frac{\partial V}{\partial t} + \sup_{\pi \in \mathcal{K}} [F + \mathcal{A}^\pi V] &= 0 \\ V(T, x) &= U(x) \end{aligned} \quad (2.8)$$

3 Methods for Solving HJB formula

There are many ways to get results of HJB PDEs. The best way is to obtain the close-form expression of the solution. Meanwhile, we can also use the numerical approach to solve HJB PDEs. In this report, we introduce three methods. Two methods are for getting analytic expression of solution. The rest one is the numerical method.

3.1 Close form

3.1.1 Using Candidate Solution

For some special utility functions, we can get the close form solution by guessing the form of $V(t, x)$, which is called the candidate solution. For example, Merton portfolio allocation problem in finite horizon can be solved by this approach[7]. We are going to show the details below.

This Merton portfolio allocation example is based on the optimal control problem in section 2.1. For convenience, we will introduce its basic information here in brief. There is an investment over a finite horizon T in the market which consists of one riskless asset, bond B , and one risky asset, stock S . We invest the stock with a proportion π_t at price S_t and the bond with $1 - \pi_t$ at price B_t . The processes of B and S are shown in (2.1) and (2.2). Assuming that there is no consumption and additional contribution during the whole period, then the wealth process of X_t is (2.3) with $c_t = 0$. In this Merton case, the utility functions is with a power form, which is as

$$U(x) = \frac{x^p}{p}, \quad x > 0, \quad 0 < p < 1 \quad (3.1)$$

Since our purpose is maximizing the expectation of utility at maturity T , then the value function is defined by

$$V(t, x) = \sup_{\pi \in \mathcal{K}} E\left[\frac{x^p}{p}\right] \quad (3.2)$$

According to the content in 2.2, because this Merton case does not involve time value, we can get its HJB equation which is like

$$\begin{aligned} \frac{\partial V}{\partial t} + \sup_{\pi \in \mathcal{K}} [x(\pi(\mu - r) + r)V_x + \frac{1}{2}x^2\pi^2\sigma^2V_{xx}] &= 0 \\ V(T, x) &= U(x) \end{aligned} \quad (3.3)$$

where $V(t, x)$ represents the value of the utility with the initial wealth x at time t . r is the risk-free interest rate and σ is the volatility of the risky asset S .

We set the candidate solution of (3.3) is in the following form

$$V(t, x) = \phi(t)U(x) \quad (3.4)$$

where $\phi(t)$ is a positive function. By substituting (3.4) into (3.3), we deduce the following ordinary differential equation (ODE)

$$\begin{aligned} \phi'(t) + \rho\phi(t) &= 0, \quad \phi(T) = 1 \\ \rho &= p \sup_{\pi \in \mathcal{K}} [\pi(\mu - r) + r + \frac{1}{2}\pi^2\sigma^2(p - 1)] \end{aligned} \quad (3.5)$$

By dealing with the equation 3.5, we can get

$$\phi(t) = \exp(\rho(T - t))U(x), \quad (t, x) \in [0, T] \times \mathbb{R}_+, \quad (3.6)$$

To get $\phi(t)$, we should obtain ρ , which means solving the \sup part in (3.5). By solving concave function $\pi(\mu - r) + r + \frac{1}{2}\pi^2\sigma^2(p - 1)$, we can attain its maximum value and the final solution of (3.3). When $\mathcal{K} = \mathbb{R}$, we have

$$\hat{\pi} = \frac{\mu - r}{(1 - p)\sigma^2} \quad (3.7)$$

$$\rho = p\left(\frac{(\mu - r)^2}{2\sigma^2(1 - p)} + r\right) \quad (3.8)$$

$$V(t, x) = \exp(\rho(T - t))\frac{x^p}{p} \quad (3.9)$$

However, for real cases, we usually set $\mathcal{K} = [0, 1]$. Under this situation, we just need to adjust $\hat{\pi}$ above to let it in $[0, 1]$, and then get the corresponding ρ and $V(t, x)$. There are many other examples which can be found in book [7].

3.1.2 Using Dual Function and Turnpike Property

Apart from the method above, Dual Function and Turnpike Property can be used to get the close form solution of HJB. We are going to illustrate its general idea here.

When the trading constraint set of an optimal problem is a closed convex cone and its utility function can be strictly concave, continuously differentiable as well as satisfying some growth conditions, we can then transfer this problem and its HJB equation into a dual control problem and dual HJB equation. By dealing with this dual HJB, we can get the optimal control and finally obtain the solution of the original HJB PDE[5, 12].

However, in some optimal problems, their utility functions do not satisfy the conditions of being differentiable or strictly concave, then Turnpike Property can be used to help relax these conditions. The detail can be found in paper [6]. Example 5.2 and 5.3 are two examples which can be solved with this Dual Function and Turnpike Property method.

3.2 Numerical Program

Compared with close form solutions, numerical results are much easier to obtain. In this section, we illustrate the process of solving HJB PDEs numerically.

3.2.1 Problem Formulation

As said in 2.2, the value functions of optimal problems are (2.7) when time value is not involved and (2.5) when it is considered. And the HJB equations are (2.8) and (2.6) respectively. We can generate a more general HJB expression which can contain the both situations. The general expression is

$$\begin{aligned} \frac{\partial V}{\partial t}(t, x) + \sup_{\pi \in \mathcal{K}} [\mathcal{L}^\pi V(t, x) + d(t, x, \pi)] &= 0 \\ V(T, x) &= U(x) \end{aligned} \quad (3.10)$$

where

$$\mathcal{L}^\pi V(t, x) \equiv \mathcal{A}^\pi V(t, x) - c(t, x, \pi)V(t, x)$$

Remark 3.1. When the optimal problem is without time value, we let $c(t, x, \pi) = 0$. Besides, the meaning of $c(t, x, \pi)$ here is totally different with that of c_t in (2.3).

Once the utility function is defined, (3.10) equation can be solved by computing backwards in time from the terminal time $t = T$, to the present time $t = 0$. To make the process simpler and

clearer, we let $\tau = T - t$, then the HJB equation can be dealt with from $\tau = 0$ to $\tau = T$ and (3.10) turns into

$$\begin{aligned} \frac{\partial V}{\partial \tau}(\tau, x) &= \sup_{\pi \in \mathcal{K}} [\mathcal{L}^\pi V(\tau, x) + d(\tau, x, \pi)] \\ V(0, x) &= U(x) \end{aligned} \quad (3.11)$$

where

$$\begin{aligned} \mathcal{L}^\pi V(\tau, x) &\equiv \mathcal{A}^\pi V(\tau, x) - c(\tau, x, \pi)V(\tau, x) \\ &\equiv a(\tau, x, \pi)V_{XX}(\tau, x) + b(\tau, x, \pi)V_X(\tau, x) - c(\tau, x, \pi)V(\tau, x) \end{aligned} \quad (3.12)$$

3.2.2 Boundary Conditions

When using numerical method, it is significant to set the boundary conditions [13, 16, 17] for space range X . Normally, it contains two parts. One is $X \rightarrow \pm\infty$, and the other is $X = 0$. For the second part, it is usually easy for us to get the value of V after imposing $X = 0$ into the HJB. We will give the boundary details at $X = 0$ in every example in the section 5. For the boundary condition at $X \rightarrow \pm\infty$, there are several solutions that can be chose.

- First of all, Dirichlet Boundary Condition, as the simplest method for specifying the boundary behaviour, is a widely used method to settle value V down at $X \rightarrow \pm\infty$. In this report, you can find it is much used in section 5. However, this condition is based on that we already have some additional knowledge about the behaviour at this boundary points. It means that Dirichlet Boundary Condition is not suitable for all cases.
- Besides, the Linear Boundary Condition is also used a lot, which is based on that the value of V is asymptotically linear of X when $X \rightarrow \pm\infty$. It has two way to settle value V down. One is assuming that $V(t, X) = a(t)X + b(t)$ and implementing it back to HJB PDE. Then, we can get two ODEs for $a(t)$ and $b(t)$. The V can be formulated by determining these two ODEs. The other way is letting $V_{xx} = 0$, and applying finite difference schemes to the HJB PDE at the boundary points. You can find this method in section 5.5, 5.6 and 5.7.

Remark 3.2. The Linear Boundary Condition is usually judged by the power number of the utility equation. When the power number is smaller or equal than one, we usually can use this method above to get the boundary value. However, when it is larger than one, we sometimes could still use it after changing the assuming asymptotic form of V . By following the same process, then the corresponding V formula will be gotten [9, 11]. There are examples for this expanded method in section 5.7 and 5.8.

- However, in some cases, we have to use one side differential schemes at the boundary points as we have no idea about the behavior of at this points. This method is called PDE boundary condition and we can find its details in [17].

Remark 3.3. According to [2] and [13], we could decrease the error made in the approximation of this boundary condition by extending the computational domain of the space.

3.2.3 Discretization of Equation

Given a discrete grid of $[0, T]$ ($\tau_i, i = 0, 1, 2, \dots, m$) and a discrete grid of $[X_{min}, X_{max}]$ ($X_j, j = 0, 1, 2, \dots, n$), we represent the value of the utility, at time level τ_i and space point X_j , by $V_j^i = V(\tau_i, X_j)$, and then generate vector V^i which contains all the elements V_j^i at time τ_i , $V^i \in \mathbb{R}^n$. To be more specific, $V^i = [V_0^i, V_1^i, V_2^i, \dots, V_n^i]'$ and its corresponding control processes is $\pi = [\pi_0, \pi_1, \pi_2, \dots, \pi_n]'$. The operator $\mathcal{L}^\pi V$ in (3.12) can be discretized using forward, backward or central differencing in domain X . Then we can get

$$(\mathcal{L}^\pi V^{i+1})_j = \alpha_j^{i+1}(\pi)V_{j-1}^{i+1} + \beta_j^{i+1}(\pi)V_{j+1}^{i+1} - (\alpha_j^{i+1}(\pi) + \beta_j^{i+1}(\pi) + c_j^{i+1}(\pi))V_j^{i+1} \quad (3.13)$$

where $\alpha_j^{i+1}, \beta_j^{i+1}$ can be found in Appendix A and B which is defined in [18]. To let this discretization method converge to right solution, which is discussed in section 4, with good accuracy, we have to make sure $\alpha_j^{i+1}, \beta_j^{i+1} \geq 0$, as well as using central differencing method as much as possible. Specifically, for every j , we first calculate $\alpha_{j,central}^{i+1}$ and $\beta_{j,central}^{i+1}$. If either of them is smaller than zero, we will choose forward or backward method which can let $\alpha_{j,forward/backward}^{i+1}, \beta_{j,forward/backward}^{i+1} \geq 0$. The details are discussed in [18].

For the left hand side of (3.11), it can be discretized by using implicit time-stepping scheme, explicit time-stepping scheme or Crank-Nicolson time-stepping scheme. When using fully implicit scheme, there comes the final discretization as following

$$\frac{V_j^{i+1} - V_j^i}{\Delta\tau} = \sup_{\pi^{i+1} \in \mathcal{K}} \{(\mathcal{L}^{\pi^{i+1}} V^{i+1})_j + d_j^{i+1}\} \quad (3.14)$$

An explicit method would evaluate with the right hand side terms in (3.14) at the old time level i instead of $i+1$. A Crank-Nicolson scheme by adding the fully implicit scheme and an explicit scheme together with an equal weight average. According to [8], implicit scheme is more recommended, so we will mainly use this scheme to do numerical computation in this report. Meanwhile, we also give examples of using Crank-Nicolson scheme in example 5.5 and 5.6. However, as an explicit method is likely leading to the unstable problem, especially when it is unbounded control, we will not consider this method here [9].

To make calculation easier, we could adjust equation (3.14) into a matrix form, and the discretized representation is as

$$\begin{aligned} [I - (1 - \theta)\Delta\tau A^{i+1}(\hat{\pi}^{i+1})]V^{i+1} &= [I + \theta\Delta\tau A^i(\hat{\pi}^i)]V^i \\ &+ (F^{i+1} - F^i) + \theta\Delta\tau D^i(\hat{\pi}^i) \\ &+ (1 - \theta)\Delta\tau D^{i+1}(\pi^{i+1}) \quad (0 \leq i < m) \\ \hat{\pi}^{i+1} &= \arg \sup_{\pi \in \mathcal{K}} [A^{i+1}(\pi)V^{i+1} + D^{i+1}(\pi)] \end{aligned} \quad (3.15)$$

where $\theta = 1$ is fully explicit, $\theta = \frac{1}{2}$ is Crank-Nicolson and $\theta = 0$ is fully implicit time-weighting [8]. $\hat{\pi}^i$ is the optimal control progress at time τ^i . $\Delta\tau^{i+1} = \tau^{i+1} - \tau^i$. F^i and F^{i+1} are vectors in \mathbb{R}^n which encodes boundary conditions when using Dirichlet Boundary Condition. Matrix $A \in \mathcal{M}_n(\mathbb{R})$. For the Linear Boundary Condition, the boundary value are calculating by using the top and bottom rows of matrix A . The rest of A , reflecting interior points, is modified according to (3.13). There will be several clear example in the following section 5 to show what A and F look like. $D^{i+1}(\pi)$ is also a vector designed as

$$(D^{i+1}(\pi))_j = \begin{cases} d_j^{i+1} & \text{for } j \text{ is not a boundary point} \\ 0 & \text{for } j \text{ is a boundary point} \end{cases}$$

However, in real cases, the programming cannot be easily achieved by (3.15) because of the difficulty for calculating optimal control $\hat{\pi}^{i+1}$. Then, we need more achievable algorithm for equation (3.15), which is shown in (3.16) [8].

Policy Iteration 1

Let $(V^{i+1})^0 = V^i$
Let $\hat{V}^k = (V^{i+1})^k$
For $k = 0, 1, 2, \dots$ until convergence
Solve

$$\begin{aligned} [I - (1 - \theta)\Delta\tau^{i+1}A^{i+1}(\pi^k)]\hat{V}^{k+1} &= [I + \theta\Delta\tau^{i+1}A^i(\hat{\pi}^i)]V^i \\ &+ (1 - \theta)\Delta\tau^{i+1}D^{i+1}(\pi^k) \\ &+ \theta\Delta\tau^{i+1}D^i(\hat{\pi}^i) \\ &+ (F^{i+1} - F^i) \end{aligned} \quad (3.16)$$

$$\pi_j^k \in \arg \sup_{\pi \in \mathcal{K}} [A^{i+1}(\pi)\hat{V}^k + D^{i+1}(\pi)]_j$$

If $k > 0$ and

$$\left(\max_j \frac{|\hat{V}_j^{k+1} - \hat{V}_j^k|}{\max(\text{scale}, |\hat{V}_j^{k+1}|)} < \text{tolerance} \right)$$

then quit
EndFor

The term *scale* in Policy Iteration 1 (3.16) is used to settle accuracy levels. Typically, $\text{scale} = 1$ for values expressed in dollars and *tolerance* can be set 0.000001[8].

Theorem 3.4. (Convergence of the Policy Iteration) *If the discretization (3.15) satisfies the Positive Coefficient Condition (4.5), then the Policy Iteration 1 (3.16) converges to the unique solution*

V^{i+1} from V^i [18].

Moreover, there is another approach, Policy Iteration 2 [8], which can deal with the problem of calculating optimal control $\hat{\pi}^{i+1}$. This algorithm use the idea of piecewise constant policy approximation. To be more precise, this algorithm divide control domain \mathcal{K} into finite z parts, $\pi_1 < \pi_2 < \pi_3 < \dots < \pi_z$, from which we can choose the optimal control $\hat{\pi}$ by comparing their corresponding values of V . The detailed algorithm is in (3.17) and its stability proof can be found in [8].

Policy Iteration 2

Let $V^0 = U(s)$, $\hat{\pi}_j^0 = \arg \sup_{\pi \in \mathcal{K}} [A^{i+1}(\pi)V^0 + D^0(\pi)]_j$

For $i = 0, 1, 2, \dots, m - 1$ (Timesteps)

For $z = 1, 2, 3, \dots, z_{max}$

Solve

$$\begin{aligned}
 [I - (1 - \theta)\Delta\tau^{i+1}A^{i+1}(\pi^z)]\mathcal{V}_z^{i+1} &= [I + \theta\Delta\tau^{i+1}A^i(\hat{\pi}^i)]V^i \\
 &+ (1 - \theta)\Delta\tau^{i+1}D^{i+1}(\pi^z) \\
 &+ \theta\Delta\tau^{i+1}D^i(\hat{\pi}^i) \\
 &+ (F^{i+1} - F^i)
 \end{aligned} \tag{3.17}$$

EndFor

$V_j^{i+1} = \max_z (V_z^{i+1})_j$ (the maximum point is on z_{opt})

$\hat{\pi}_j^{i+1} = \pi_{z_{opt}}$

($V_n^{i+1} = F_n^{i+1}$ and $V_0^{i+1} = F_0^{i+1}$ based on the boundary conditions)

EndFor

4 Convergence to the viscosity solution

It is important to ensure that the discretization method we used can guarantee to converge to the viscosity solution [1, 4]. This section are mainly analyzing the convergence of the examples in section 5 except the last one. The last example which contains two-dimensional space will be analyzed in section 5.8. To prove that the numerical discretization scheme converge to the viscosity solution, we have the following theorem.

Theorem 4.1. *From [20, 1], we know that a numerical scheme can converge to the unique viscosity solution as long as the strong comparison property [1, 10, 19] holds in equation (4.1) and the*

numerical method is stable (in the $\|\cdot\|_\infty$ norm), consistent and monotone.

Strong Comparison Property According to the conclusions in [19, 21], equation (4.1) satisfies the strong comparison property on the computational domain with bounded control π , which means that $-\infty < X_{min}, X_{max} < +\infty$ and \mathcal{K} is bounded. In most of the cases in this report, we use bounded controls, such as $\mathcal{K} = [0, 1]$, and set X_{min} and X_{max} a specific number for computational purpose, so the strong comparison holds in this cases .

While, for the unbounded control case, such as 5.7, we could violate one assumption in the proof of the strong comparison property in [21] and make sure πx bounded. Then, we could still guarantee that the discretization is strong comparison [21, 9].

Stability, Consistency and Monotonicity To analyzing stable (in the $\|\cdot\|_\infty$ norm), consistent and monotone, we will firstly define the following terms.

$$\begin{aligned} (\Delta\tau)_{min} &= \min_i(\tau_{i+1} - \tau_i) \\ (\Delta\tau)_{max} &= \max_i(\tau_{i+1} - \tau_i) \\ (\Delta X)_{min} &= \min_j(X_{j+1} - X_j) \\ (\Delta X)_{max} &= \max_j(X_{j+1} - X_j) \end{aligned}$$

where we assume that there are discretization mesh size parameters h_{min}, h_{max} such that

$$\begin{aligned} (\Delta\tau)_{min} &= C_1 h_{min} \\ (\Delta\tau)_{max} &= C_2 h_{max} \\ (\Delta X)_{min} &= C_3 h_{min} \\ (\Delta X)_{max} &= C_4 h_{max} \end{aligned}$$

with C_1, C_2, C_3, C_4 positive and independent of h . Then we can write the discrete equations (3.15) into a new form at each node. This new form is as

$$\begin{aligned} &G_j^{i+1}(h_{max}, V_{j+1}^{i+1}, V_j^{i+1}, V_{j-1}^{i+1}, V_{j+1}^i, V_j^i, V_{j-1}^i) \\ &\equiv \frac{V_j^{i+1} - V_j^i}{\Delta\tau} - (1 - \theta) \sup_{\pi^{i+1} \in \mathcal{K}} (A^{i+1}(\pi^{i+1})V^{i+1} + D^{i+1}(\pi^{i+1}))_j \\ &\quad - \frac{F_j^{i+1} - F_j^i}{\Delta\tau} - \theta \sup_{\pi^i \in \mathcal{K}} (A^i(\pi^i)V^i + D^i(\pi^i))_j \end{aligned} \quad (4.1)$$

Definition 4.2. (Stability) Discretization (4.1) is stable if

$$\|V^{i+1}\|_\infty \leq C, \quad 0 \leq i < m \quad (4.2)$$

for $(\Delta\tau)_{min} \rightarrow 0, (\Delta x)_{min} \rightarrow 0$, where C is independent of $(\Delta\tau)_{min}, (\Delta x)_{min}$. m is the time steps number which can be found at the beginning of section 3.2.3

Definition 4.3. (Consistency) Scheme (4.1) is consistent if, for any smooth function ϕ , with $\phi_j^i = \phi(\tau_i, X_j)$, we have

$$\lim_{h_{max} \rightarrow 0} |(\phi_\tau - \sup_{\pi \in \mathcal{K}} \{\mathcal{L}^\pi \phi + d\}_j^{i+1}) - G_j^{i+1}(h_{max}, \phi_{j+1}^{i+1}, \phi_j^{i+1}, \phi_{j-1}^{i+1}, \phi_{j+1}^i, \phi_j^i, \phi_{j-1}^i)| = 0 \quad (4.3)$$

When the operator degenerates, the definition of consistency becomes into a complicated form which can be found in [1]. And the degeneration always happens at the boundary points. For examples in section 5, the degeneracy occurs all at the boundary point $S = 0$ or $X = 0$. And boundary condition at that point is simply the limit of (3.14) as $S \rightarrow 0$ or $X \rightarrow 0$. Therefore, this degeneracy problem does not arise.

Definition 4.4. (Monotonicity) The discrete scheme (4.1) is monotone if for all $\epsilon_s^i \geq 0$,

$$\begin{aligned} G_j^{i+1}(h_{max}, V_{j+1}^{i+1} + \epsilon_{j+1}^{i+1}, V_j^{i+1}, V_{j-1}^{i+1} + \epsilon_{j-1}^{i+1}, V_{j+1}^i + \epsilon_{j+1}^i, V_j^i + \epsilon_j^i, V_{j-1}^i + \epsilon_{j-1}^i) \\ \leq G_j^{i+1}(h_{max}, V_{j+1}^{i+1}, V_j^{i+1}, V_{j-1}^{i+1}, V_{j+1}^i, V_j^i, V_{j-1}^i) \end{aligned} \quad (4.4)$$

To prove that the numerical scheme are stable, consistent and monotone, we need to introduce the following condition[9].

Condition 4.5. (Positive Coefficient Condition)

$$\alpha_j^{i+1} \geq 0, \beta_j^{i+1} \geq 0, c_j^{i+1} \geq 0, j = 0, 1, 2, \dots, n \quad (4.5)$$

where $\alpha_j^{i+1} \geq 0, \beta_j^{i+1} \geq 0$ is from (3.13). According to [18], this method ensures the Positive Coefficient Condition above. In this report, we also apply this method to get the final numerical results.

From [8] and [9], we can find that if discretization equation (4.1) satisfies Positive Coefficient Condition, and the boundary condition is using Dirichlet Boundary Condition or Linear Boundary Condition, then the numerical discretization method is stable, consistent and monotone. Besides, [8, 22] give simple methods to show that scheme (4.1) satisfies these three properties.

5 Examples by Using Numerical Method

In this section, we introduce several specific optimal control examples and solve them by using the numerical method in section 3.2.

5.1 Merton portfolio problem with boundary control

As in section 3.1.1, we know its utility function 3.1, the value function 3.2 and the corresponding HJB equation 3.3. To make it convenient, we rewrite HJB equation here,

$$\begin{aligned} \frac{\partial V}{\partial \tau} &= \sup_{\pi \in \mathcal{K}} [x(\pi(\mu - r) + r)V_x + \frac{1}{2}x^2\pi^2\sigma^2V_{xx}] \\ V(\tau = 0, x) &= \frac{x^p}{p}, \quad x > 0, \quad 0 < p < 1 \end{aligned} \quad (5.1)$$

Let $p = 0.5$, set $\mathcal{K} = [0, 1]$ and use fully implicit scheme to get its numerical result.

For the boundary points $X = 0$, we can impose $X = 0$ in Merton problem HJB (5.1) and then get $V_\tau(\tau, 0) = 0$, which means that the value of V will not change with time when $X = 0$. Then, $V(\tau, 0) = V(\tau = T, 0) = 0$. Meanwhile, it does not matter how to choose the value of control π . We set the optimal control $\hat{\pi} = 0$ at this point.

When $X \rightarrow +\infty$, we can use Dirichlet Boundary Condition. As this example does not concern about time value, we assume that $V(t, X \rightarrow +\infty) = U(X)$, and still set $\hat{\pi} = 0$ at this point.

For computational purposes, we must truncate this infinite domain to $[0, X_{max}]$. By setting X_{max} much larger than X_{target} , the behavior at X_{max} would be similar to that at $X \rightarrow +\infty$.

For the rest non-boundary points, we mainly use central differencing scheme as well as ensure Positive Coefficient Condition (4.5) hold. The detailed process can be found in section 3.2.3.

As we said earlier, we will use fully implicit time-weighting to deal with this example, then let $\theta = 0$ in equation (3.15) and the matrix form of this discretization is as below

$$\begin{aligned} [I - \Delta\tau A^{i+1}(\hat{\pi}^{i+1})]V^{i+1} &= V^i + (F^{i+1} - F^i) \\ \hat{\pi}^{i+1} &= \arg \sup_{\pi \in \mathcal{K}} [A^{i+1}(\pi)V^{i+1}] \quad 0 \leq i < m \end{aligned} \quad (5.2)$$

The matrix A^{i+1} in equation above is as

$$\begin{pmatrix} 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ \alpha_1^{i+1} & -(\alpha_1^{i+1} + \beta_1^{i+1} + c_1^{i+1}) & \beta_1^{i+1} & 0 & \dots & 0 & 0 \\ 0 & \alpha_2^{i+1} & -(\alpha_2^{i+1} + \beta_2^{i+1} + c_2^{i+1}) & \beta_2^{i+1} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 \end{pmatrix}$$

Vector F^{i+1} and F^i are as below

$$F^{i+1} = F^i = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ U(X_{max}) \end{pmatrix}$$

We finally choose Policy Iteration algorithm 1 (3.16) to get the numerical results at $X = X_{target}$ and $t = 0$.

Table 1 shows the parameters used in the Merton portfolio problem. Table 2 is the corresponding numerical results after convergence studies. By comparing these results with those from analytic solution (3.9) in 3.1.1, we find that the errors are really small, even in a low accuracy level. Figure 5.1 reflects these errors between the numerical method and the analytic solution.

Parameter	Value
r	0.04
T	0.5
$\mu(t)$	0.05
$\sigma(t)$	0.30
x_{target}	100.0
x_{min}	0.0
x_{max}	500.0
$scale$	1.0
Convergence Tolerance (Policy Iteration)	10^{-6}

Table 1: Merton Example Parameters

Level	Nodes	Timesteps	Nonlinear iterations	Option value	Change	Ratio
0	95	100	200	20.206630		
1	189	200	398	20.206622	0.000008	
2	377	400	595	20.206618	0.000004	2.0
3	753	800	1177	20.206617	0.000001	4.0

Table 2: Merton Example. Convergence with fully implicit time-stepping method and Policy Iteration 1 method (3.16). The value at the column of Nodes is the number of spaced nodes. The Nonlinear iterations shows the iteration times after difference schemes being settled down at each nodes. The close-form utility value at $X = 100, t = 0$ is 20.206616

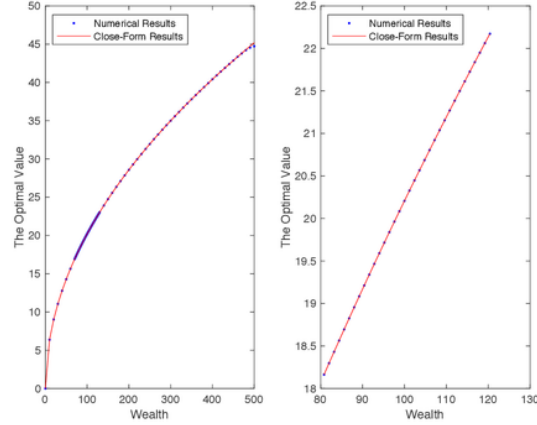


Figure 1: Merton Example. The numerical results are at the accuracy level 0 and the close-form results are from (3.9). As we set the spaced step near X_{target} smaller than that in other place, the points near X_{target} are dense. We show a more detailed figure for these dense points on the right.

5.2 Turnpike problem 1 with boundary control

This example is still based on the information from section 2.1 and its utility function are as following

$$U(x) = H \wedge x \quad H \text{ is a positive constant.} \quad (5.3)$$

As this example does not involve the influence of time value, its HJB equation is (2.8). Let $\tau = T - t$, we have this HJB PDE as

$$\begin{aligned} \frac{\partial V}{\partial \tau} &= \sup_{\pi \in \mathcal{K}} [x(\pi(\mu - r) + r)V_x + \frac{1}{2}x^2\pi^2\sigma^2V_{xx}] \\ V(\tau = 0, x) &= U(x) \end{aligned} \quad (5.4)$$

where $\mathcal{K} = [0, 1]$. According to [6] we can get its close-form solution which is shown below by using Dual Function and Turnpike Property method in section 3.1.2.

$$V(t, x) = \begin{cases} H\Phi(\Phi^{-1}(\frac{x}{H}e^{r(T-t)}) + \theta\sqrt{T-t}) & , 0 \leq x < He^{-r(T-t)} \\ H & , x \geq He^{-r(T-t)} \end{cases} \quad (5.5)$$

where $\theta = \frac{\mu-r}{\sigma}$ and Φ is the cumulative distribution function of a standard normal variable.

When computing this example numerically, we need firstly analyzing its boundary conditions. For the point $X = 0$, the HJB (5.4) becomes to $V_\tau(\tau, 0) = 0$ which can be seen as an ODE since $X \equiv 0$. As we have already know the initial condition $V(\tau = 0, x = 0) = U(0) = 0$, then we can get $V(\tau, 0) = 0$ by dealing with this ODE. It is obvious that there is no relationship with control π when $X = 0$ and then we simply set $\hat{\pi} = 0$ at this boundary point $X = 0$. Besides, we use Dirichlet Boundary conditions to determine the approximate solution of the HJB PDE (5.4) as $X \rightarrow +\infty$. And the specific formula for this solution is below:

$$V(\tau, X \rightarrow +\infty) = U(X) = H$$

We still let $\hat{\pi} = 0$ at this point. After truncating this infinite domain to a finite interval $[X_{min}, X_{max}]$, $X_{min} < X_{target} \ll X_{max} < +\infty$, we then get the computational space domain. The behavior at X_{max} would be similar to that at $X \rightarrow +\infty$. The rest interior points will be dealt with by using the method in section 3.2.3. The matrix A^{i+1} in (3.15) is

$$\begin{pmatrix} 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ \alpha_1^{i+1} & -(\alpha_1^{i+1} + \beta_1^{i+1} + c_1^{i+1}) & \beta_1^{i+1} & 0 & \dots & 0 & 0 \\ 0 & \alpha_2^{i+1} & -(\alpha_2^{i+1} + \beta_2^{i+1} + c_2^{i+1}) & \beta_2^{i+1} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 \end{pmatrix}$$

and vector F^i and F^{i+1} is

$$F^i = F^{i+1} = \begin{pmatrix} U(X_{min}) \\ 0 \\ 0 \\ \vdots \\ 0 \\ U(X_{max}) \end{pmatrix}$$

Table 3 shows the parameters used in this example and the numerical results as well as close-form results are in Table 4. It is really interesting that the numerical results convergence to close-form results quickly when $X_{target} = 100$. The reason is that $X_{target} = 100$ is a special point as $X_{target} = H = 100$ and the value of $V_\tau = 0$ when X is larger or equal than H . Then we choose another $X_{target} = 97.6$ to compute and the corresponding results are in Table 5. By comparing the data in 5, we find that the errors reduce stably. Figure 2 shows the results from the numerical method and the analytic expression of the solution, from which we can find that the numerical results are really close to close-form solutions.

Parameter	Value
r	0.04
T	0.5
$\mu(t)$	0.05
$\sigma(t)$	0.30
H	100.0
x_{target}	100.0
x_{min}	0.0
x_{max}	500.0
$scale$	1.0
<i>Convergence Tolerance (Policy Iteration)</i>	10^{-6}

Table 3: Turnpike Problem 1 Parameters

Level	Nodes	Timesteps	Nonlinear iterations	Option value	Change
0	95	100	100	100	
1	189	200	200	100	0
2	377	400	400	100	0
3	753	800	800	100	0

Table 4: *Turnpike Problem 1. Convergence with fully implicit time-stepping method and Policy Iteration 1 method (3.16). The value at the column of Nodes is the number of spaced nodes. The Nonlinear iterations shows the iteration times after difference schemes being settled down at each nodes. The close-form utility value at $X = 100, t = 0$ is 100.*

Level	Nodes	Timesteps	Nonlinear iterations	Option value	Change	Ratio
0	95	100	100	99.146689		
1	189	200	200	99.310565	0.163879	
2	377	400	400	99.426619	0.116054	1.41
3	753	800	800	99.503256	0.076637	1.51

Table 5: *Turnpike Problem 1. Convergence with fully implicit time-stepping method and Policy Iteration 1 method (3.16). The value at the column of Nodes is the number of spaced nodes. The Nonlinear iterations shows the iteration times after difference schemes being settled down at each nodes. The close-form utility value at $X = 97.6, t = 0$ is 99.600435.*

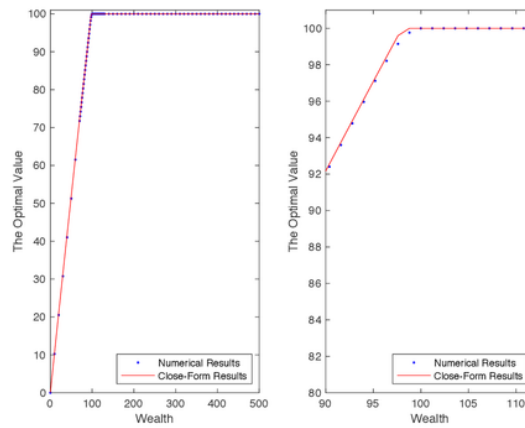


Figure 2: *Turnpike Problem 1. The numerical results are at the accuracy level 0 and the close-form results are from (5.5). As we set the spaced step near X_{target} smaller than that in other place, the points near X_{target} are dense. We show a more detailed figure for these dense points on the right.*

5.3 Turnpike problem 2 with boundary control

In this example, we still choose π as the control which represents the proportion invested in risky assets S . However, the investment period T is set to turn to be infinity which is different from the background information in section 2.1. The utility function of this example is as following

$$U(x) = \frac{1}{3}H^{-3}(x) + H^{-1}(x) + xH(x) \quad , \quad x > 0 \quad (5.6)$$

where

$$H(x) = \left(\frac{2}{-1 + \sqrt{1 + 4x}} \right)^{\frac{1}{2}}$$

We use the HJB (2.8) to compute the final results and then compare them with that from the close-form solution. Let $\tau = T - t$, we have the HJB equation as

$$\begin{aligned} \frac{\partial V}{\partial \tau} &= \sup_{\pi \in \mathcal{K}} [x(\pi(\mu - r) + r)V_x + \frac{1}{2}x^2\pi^2\sigma^2V_{xx}] \\ V(\tau = 0, x) &= U(x) \end{aligned} \quad (5.7)$$

where $\mathcal{K} = [0, 1]$.

The close-form solution of (5.7) is defined as following according to [6].

$$V(t, x) = \frac{2}{3}(y^{-1}e^{(r+\theta^2)t} + 2xy) \quad (5.8)$$

where

$$y^2 = \frac{1}{2x}(e^{(r+\theta^2)t} + \sqrt{e^{(r+\theta^2)t} + 4xe^{3(r+2\theta^2)t}})$$

We analyze boundary conditions of HJB (5.7) as the same way with Turnpike problem 1 in section 5.2. At the points $X = 0$ and $X \rightarrow +\infty$, the values of V can be determined both with Dirichlet Boundary conditions. The specification formula for $X \rightarrow +\infty$ is as below. We set the optimal control $\hat{\pi} = 0$ at both points for convenience.

$$V(\tau, X \rightarrow +\infty) = U(X)$$

After truncating this infinite domain to $[X_{min}, X_{max}]$ and deciding differencing schemes as well as the time-stepping approach, we can then get the numerical results for this example by using equation (3.15). The matrix A^{i+1} and vectors F^i, F^{i+1} in (3.15) have the same forms as those in Turnpike problem 1.

Table 6 shows the parameters used for the Turnpike problem 2. Table 7 shows the numerical results as well as close-form results. Figure 3 shows the difference of the results from numerical method and analytic solution more clearly. By comparing these data in Table 7 and Figure 3, we find that the errors are not small enough, even in a high accuracy level and the numerical results converge really slow to the close-form value. The reason is that we give a small value, 0.5, to the parameter T which violates the original condition, $T \rightarrow +\infty$, of the analytic solution. However, if

we increase the value of T , we need to give a considerably large value to Timesteps to maintain accuracy which leads to a really time-consuming computation. But from the results in Table 7, we still guess that the numerical results will converge slowly to the close-form results with the accuracy increase. To prove it, we change the value of T , and get Table 8. The results in Table 8 shows that the errors is going to be smaller with the decrease of time step even the T is not large enough. From this phenomenon, we could guess that some optimal control problems with infinite T can be solved numerically by reducing T .

Parameter	Value
r	0.04
T	0.5
$\mu(t)$	0.05
$\sigma(t)$	0.30
x_{target}	100.0
x_{min}	0.0
x_{max}	500.0
$scale$	1.0
<i>Convergence Tolerance (Policy Iteration)</i>	10^{-6}

Table 6: Turnpike Problem 2 Parameters

Level	Nodes	Timesteps	Nonlinear iterations	Option value	Change	Ratio
0	95	100	215	45.97496		
1	189	200	413	45.974933	0.000027	
2	377	400	789	45.974919	0.000014	1.93
3	753	800	1476	45.974913	0.000006	2.33

Table 7: *Turnpike Problem 1. Convergence with fully implicit time-stepping method and Policy Iteration 1 method (3.16). The value at the column of Nodes is the number of spaced nodes. The Nonlinear iterations shows the iteration times after difference schemes being settled down at each nodes. The close-form utility value at $X = 100, t = 0$ is 45.286787.*

T	Option value	Change	Ratio
10.0	61.34021		
1.0	46.673983	14.666227	
0.1	45.423557	1.250433	11.73
0.01	45.300444	0.123113	10.16
0.001	45.288152	0.012292	10.02

Table 8: *Turnpike Problem 1. Convergence with fully implicit time-stepping method and Policy Iteration 1 method (3.16). The value of Nodes is 95 and Timesteps is 100. The close-form utility value at $X = 100$, $t = 0$ is 45.286787.*

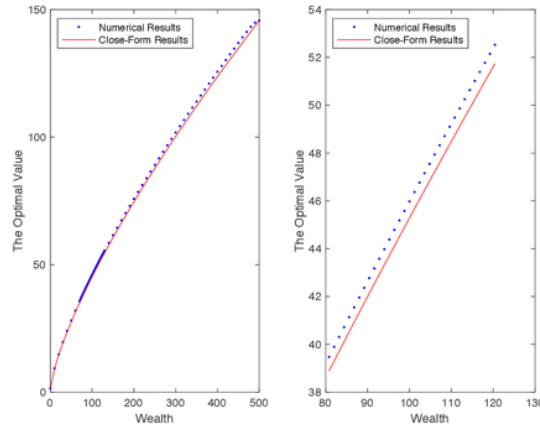


Figure 3: *Turnpike Problem 2. The numerical results are at the accuracy level 0 and the close-form results are from (5.8). As we set the spaced step near X_{target} smaller than that in other place, the points near X_{target} are dense. We show a more detailed figure for these dense points on the right.*

5.4 Uncertain Volatility

In this section, we will consider the case of pricing a butterfly spread [15] on the standpoints of both purchasers and financial institutions who issue this derivative. This butterfly spread can be seen as a portfolio of plain vanilla European Call Options on the same underlying asset. It involves buying (long) a low strike K_1 option, selling (short) two middle strike K_2 options, and buying (long) a high strike K_3 option, all with identical maturities T . The volatility of the price for this butterfly spread contract is uncertain, thus, we need to build an optimal control problem to price this derivative, which is with the uncertain volatility as the control term.

We assume there are three financial products in the market, one riskless bond B with a process

(2.1), one risky stock S with the stochastic process (2.2) and this butterfly spread V . Let $V(t, s)$ be the value of the contract written on the stock asset S with the strike price K_1, K_2, K_3 . Moreover, the volatility $\sigma(t)$ is uncertain, which is the control process of this example.

In this example, if we stand at the spread issuers side, we want to maximize the price of the butterfly spread. Hence, the utility function is the price function, which is defined as

$$U(S) = \max\{0, S - K_1\} - 2\max\{0, S - K_2\} + \max\{0, S - K_3\} \quad (5.9)$$

Consequently, the value function is

$$V(t, s) = \sup_{\sigma \in \Sigma} E[U(S)] \quad (5.10)$$

where $\Sigma = [\sigma_{min}, \sigma_{max}]$. As this optimal control problem considers time value, then according to section 2.2, we can get the corresponding HJB equation (2.6). Let $\tau = T - t$, then HJB is as following

$$V_\tau = \sup_{\sigma \in \Sigma} \left\{ \frac{\sigma^2 S^2}{2} V_{SS} + \mu S V_S - rV \right\} \quad (5.11)$$

$$V(\tau = 0, s) = U(S)$$

Of course, from the perspective of purchasers, this represents the worst case. To get the best price for purchasers, it is just needed to replace the *sup* with an *inf* in the HJB equation above. We will show the results of both cases in this section.

For the boundary points $S = 0$, we can impose it in HJB equation (5.11) and get $V_\tau(\tau, 0) = -rV(\tau, 0)$. By using time-stepping approach, it will then be discretized. As it is not relevant to control process σ , we easily set $\hat{\sigma} = \sigma_{max}$.

When $S \rightarrow +\infty$, we can use Dirichlet Boundary conditions to get the value. We let $\hat{\sigma} = \sigma_{max}$ and determine the following formula:

$$V(S \rightarrow +\infty, \tau) = U(S)e^{-r(T-\tau)}$$

For computational purposes, we must truncate this infinite domain to $[0, S_{max}]$, by setting S_{max} much larger than S_{target} . Then, the behavior at S_{max} would be similar to that at $S \rightarrow +\infty$.

To increase accuracy, we still prefer to use central differencing scheme at the rest non-boundary points. Then, together with the analysis about boundary conditions and fully implicit approach in computation, we can get the matrix A^{i+1}

$$A^{i+1} = \begin{pmatrix} -r & 0 & 0 & 0 & \dots & 0 & 0 \\ \alpha_1^{i+1} & -(\alpha_1^{i+1} + \beta_1^{i+1} + c_1^{i+1}) & \beta_1^{i+1} & 0 & \dots & 0 & 0 \\ 0 & \alpha_2^{i+1} & -(\alpha_2^{i+1} + \beta_2^{i+1} + c_2^{i+1}) & \beta_2^{i+1} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 \end{pmatrix}$$

and vector F^{i+1} is

$$F^{i+1} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ U(S)e^{-r(T-\tau)} \end{pmatrix}$$

All the input parameters are provided in Table 9. Table 10 and Table 11 show the numerical results of the best and the worst cases by doing convergence studies for this problem, from which we can find that the numerical results converge stably with the increase of Nodes and Timesteps. Figure 4 shows these two results in a more perspicuous way.

Parameter	Value
r	0.04
T	0.5
K_1	95.0
K_2	100
K_3	105.0
$\mu(t)$	0.04
σ_{min}	0.30
σ_{max}	0.45
s_{target}	100.0
s_{min}	0.0
s_{max}	500.0
$scale$	1.0
<i>Convergence Tolerance (Policy Iteration)</i>	10^{-6}

Table 9: Uncertain Volatility Example Parameters

Level	Nodes	Timesteps	Nonlinear iterations	Option value	Change	Ratio
0	95	100	226	0.79674718		
1	189	200	443	0.79830842	0.00156124	
2	377	400	852	0.79940966	0.00110124	1.41
3	753	800	1675	0.80036542	0.00095576	1.15

Table 10: *Best Cases of Uncertain Volatility Example at $S = 100$, $t = 0$. Convergence by using fully implicit control approach and the Policy Iteration 1 method (3.16). The value at the column of Nodes is the number of spaced nodes. The Nonlinear iterations shows the iteration times after difference schemes being settled down at each nodes.*

Level	Nodes	Timesteps	Nonlinear iterations	Option value	Change	Ratio
0	95	100	235	0.13148352		
1	189	200	446	0.12907525	0.00240827	
2	377	400	857	0.12746878	0.00160647	1.5
3	753	800	1677	0.12666878	0.0008	2.0

Table 11: *Worst Results of Uncertain Volatility Example at $S = 100$, $t = 0$. Convergence by using fully implicit control approach and the Policy Iteration 1 method (3.16). The value at the column of Nodes is the number of spaced nodes. The Nonlinear iterations shows the iteration times after difference schemes being settled down at each nodes.*

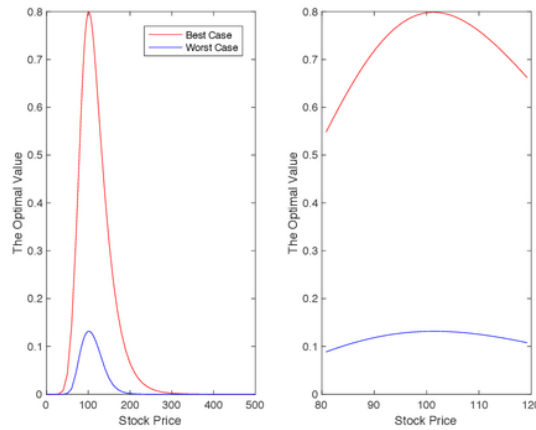


Figure 4: *Uncertain Volatility Example. The numerical results are at the accuracy level 0. As we set the spaced step near S_{target} smaller than that in other place, the points near S_{target} are dense. We show a more detailed figure for these dense points on the right.*

5.5 Unequal Borrowing/Lending Rates

This section shows the optimal control problem of the case where the borrowing rate r_b and the lending rate r_l are unequal and $r_b \geq r_l$ [8]. In other words, if we need money to invest in stocks or other assets, the bank will lend to us at an interest rate r_b . While we can also lend (deposit) money to the bank at a lower interest rate r_l . Next, we will price an option V under this condition.

Assuming the financial market consists of a bank account, one stock and one option. It does not matter what this option here is, and we choose European Straddle Option with strike price K for this section. Therefore the utility function is

$$U(s) = \max\{s - K, 0\} + \max\{K - s, 0\} \quad (5.12)$$

If we stand from the option issuer's view, the pricing process begin with shorting this option and we want the price to be as high as possible. Conversely, we have to assume to long this option from the perspective of investors and we want to pay as little money as possible. The price of the option V is then given by the nonlinear HJB PDE

$$\begin{aligned} \text{Short Position : } & \frac{\partial V}{\partial t} + \frac{\sigma^2 S^2}{2} V_{SS} + \rho(V - SV_S)(SV_S - V) = 0 \\ \text{Long Position : } & \frac{\partial V}{\partial t} + \frac{\sigma^2 S^2}{2} V_{SS} + \rho(SV_S - V)(SV_S - V) = 0 \end{aligned} \quad (5.13)$$

where the control process is

$$\rho(x) = \begin{cases} r_l & \text{for } x \geq 0 \\ r_b & \text{for } x < 0 \end{cases}$$

The detail of the proof can be seen in [8]. Let $\tau = T - t$. Considering time value, the HJB for Short Position is as following:

$$\frac{\partial V}{\partial \tau} = \sup_{q \in \mathbb{Q}} \left\{ \frac{\sigma^2 S^2}{2} V_{SS} + q(SV_S - V) \right\} \quad (5.14)$$

where $\mathbb{Q} = \{r_b, r_l\}$. When analysing Long Position case, we just need to replace *sup* in above PDE (5.14) to *inf*.

For the boundary points $S = 0$, we can impose it in its HJB (5.14) and get $V_\tau(\tau, 0) = \sup_{q \in \mathbb{Q}} \{-qV(\tau, 0)\}$, which just needs to be discretized in time. The optimal control value at this point is by dealing with $\sup_{q \in \mathbb{Q}} \{-qV(\tau, 0)\}$. To be more specific, $\hat{q} = r_b$ when $-V(\tau, 0) > 0$. Otherwise, $\hat{q} = r_l$.

When $S \rightarrow +\infty$, we determine the asymptotic form of the solution by Linear Boundary Condition which can be found in section 3.2.2. We can assume that $V_{SS} = 0$ when $S \rightarrow +\infty$ because the degree of S in utility function (5.12) is one. Then implement V in a linear form which is

$$V(\tau, S \rightarrow +\infty) = a_0(\tau)S + b_0(\tau)$$

By substituting the form above into HJB (5.14), we will obtain ODEs for both $a(t)$ and $b(t)$.

$$\begin{aligned} a'_0(\tau) &= 0 \\ b'_0(\tau) &= -qb_0(\tau) \end{aligned}$$

Together with the condition $V(\tau = 0, S \rightarrow +\infty) = U(S \rightarrow +\infty) = S - K$, then we can determined the final $a_0(\tau) = 1$ and $b_0(\tau) = -Ke^{-q\tau}$. As we have to truncate this infinite domain of S to $[S_{min}, S_{max}]$ for computational purposes, then the value of V at S_{max} will be as following as long as S_{max} is sufficiently large.

$$V(\tau, S_{max}) = a_0(\tau)S_{max} + b_0(\tau) = S_{max} - Ke^{-q\tau}$$

Consequently, the corresponding control q can be determined by maximizing the payoff above.

Central differencing scheme is still the first choice for discretization at the rest non-boundary points. And for this example, we try both fully implicit time-stepping and Crank-Nicolson time-stepping to get the numerical value. The matrix A^{i+1} and vector F^{i+1} are built as following

$$A^{i+1} = \begin{pmatrix} -q & 0 & 0 & 0 & \dots & 0 & 0 \\ \alpha_1^{i+1} & -(\alpha_1^{i+1} + \beta_1^{i+1} + c_1^{i+1}) & \beta_1^{i+1} & 0 & \dots & 0 & 0 \\ 0 & \alpha_2^{i+1} & -(\alpha_2^{i+1} + \beta_2^{i+1} + c_2^{i+1}) & \beta_2^{i+1} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 \end{pmatrix}$$

$$F^{i+1} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ V(\tau^{i+1}, X_{max}) \end{pmatrix}$$

Besides, we use both Policy Iteration 1 (3.16) and algorithm (3.17) to gain numerical results. The results of both Short and Long Position will be shown in the following content. Table 12 shows all the parameters necessary to use in computation. Table 13 shows the data of using Policy Iteration 1 (3.16) with fully implicit time-stepping approach and Crank-Nicolson time-stepping approach. Table 14 shows the results by using Policy Iteration 2 (3.17) with fully implicit time-stepping approach and Crank-Nicolson time-stepping approach. Figure 5 and Figure 6 show the results of Short and Long Position cases respectively. It is clearly that the numerical results are close to each other when we using different time-stepping approaches with different policy iteration algorithms.

Parameter	Value
r_b	0.05
r_l	0.03
T	1.0
K	100.0
$\sigma(t)$	0.30
s_{target}	100.0
s_{min}	0.0
s_{max}	500.0
$scale$	1.0
Convergence Tolerance (Policy Iteration)	10^{-6}

Table 12: Unequal Borrowing/Lending Rates Example

Level	Nodes	Timesteps	Nonlinear iterations	Option value	Change	Ratio
Fully Implicit: Short						
0	101	100	200	23.909523		
1	201	200	400	24.021877	0.112354	
2	401	400	800	24.054311	0.032434	3.5
3	801	800	1600	24.064406	0.010095	3.2
Crank-Nicolson: Short						
0	101	100	202	23.940077		
1	201	200	408	24.037514	0.097437	
2	401	400	825	24.061907	0.024393	3.9
3	801	800	1654	24.068353	0.006446	3.8
Fully Implicit: Long						
0	101	100	201	22.940066		
1	201	200	401	23.058689	0.118623	
2	401	400	800	23.092736	0.034047	3.5
3	801	800	1600	23.103214	0.010478	3.2
Crank-Nicolson: Long						
0	101	100	202	22.970174		
1	201	200	408	23.074024	0.10385	
2	401	400	847	23.100513	0.026489	3.9
3	801	800	1655	23.107051	0.006538	4.0

Table 13: Results of Unequal Borrowing/Lending Rates Example at $S = 100$, $t = 0$ with Policy Iteration 1 (3.16). Convergence for fully implicit time-stepping and Crank-Nicolson time-stepping. The value at the column of Nodes is the number of spaced nodes. The Nonlinear iterations shows the iteration times after difference schemes being settled down at each nodes.

Level	Nodes	Timesteps	Option value	Change	Ratio
Fully Implicit: Short					
0	101	100	23.900759		
1	201	200	24.017376	0.116617	
2	401	400	24.052026	0.03465	3.3
3	801	800	24.063253	0.011227	3.0
Crank-Nicolson: Short					
0	101	100	23.937905		
1	201	200	24.036375	0.09847	
2	401	400	24.061015	0.02464	3.9
3	801	800	24.068052	0.007037	3.5
Fully Implicit: Long					
0	101	100	22.948961		
1	201	200	23.063254	0.114293	
2	401	400	23.095051	0.031797	3.6
3	801	800	23.104382	0.009331	3.4
Crank-Nicolson: Long					
0	101	100	22.972377		
1	201	200	23.07518	0.102803	
2	401	400	23.101017	0.025837	3.9
3	801	800	23.107358	0.006341	4.0

Table 14: *Results of Unequal Borrowing/Lending Rates Example at $S = 100$, $t = 0$ with Policy Iteration 2 (3.17). Convergence for fully implicit time-stepping and Crank-Nicolson time-stepping. The value at the column of Nodes is the number of spaced nodes. The Nonlinear iterations shows the iteration times after difference schemes being settled down at each nodes.*

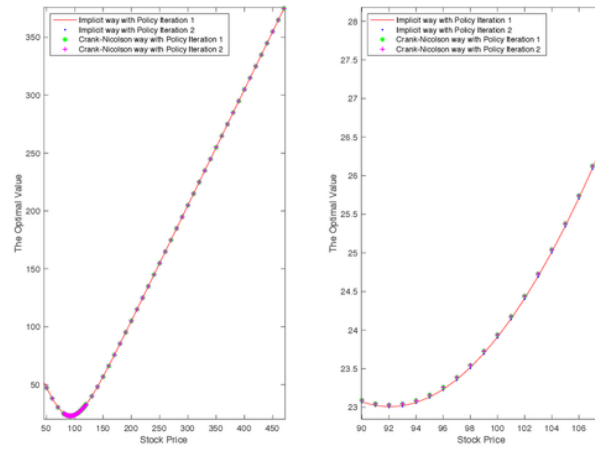


Figure 5: *Short Position case in Unequal Borrowing/Lending Rates Example.* The numerical results are at the accuracy level 0. As we set the spaced step near S_{target} smaller than that in other place, the points near S_{target} are dense. We show a more detailed figure for these dense points on the right.

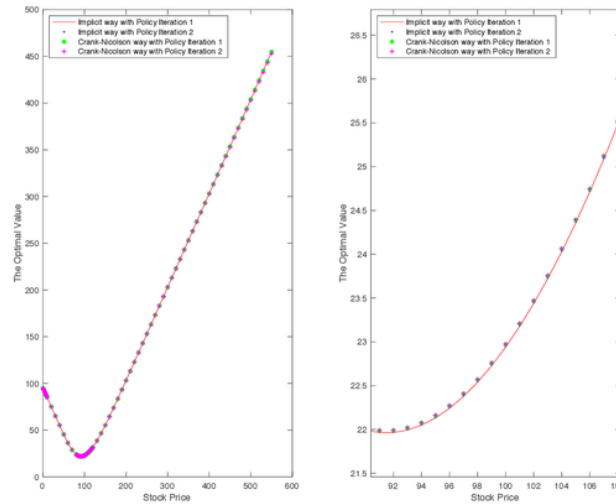


Figure 6: *Long Position case in Unequal Borrowing/Lending Rates Example.* The numerical results are at the accuracy level 0. As we set the spaced step near S_{target} smaller than that in other place, the points near S_{target} are dense. We show a more detailed figure for these dense points on the right.

5.6 Stock Borrowing Fees

The borrowing/lending model in Example 5.5 can be extended to include stock borrowing fees [8]. Such fees are effectively paid to stock lenders when an investor borrows a stock. Specifically, the stock lenders can receive fees from borrowers at an interest rate r_f . Therefore, when investors sell the borrowing stocks and reinvest the money at a rate r_l , they totally gain in rate $r_l - r_f$ and $r_f \neq r_l$. For the borrowing part and the utility function, it is the same as that in example Unequal Borrowing/Lending Rates model 5.5. We also divide this example into two part, Short Position and Long Position, based on issuers' view and purchasers' view respectively. The nonlinear pricing PDE in these two cases are :

$$\begin{aligned}
 \text{Short Position : } & \frac{\partial V}{\partial t} + \frac{\sigma^2 S^2}{2} V_{SS} + H(V_s)[\rho(V - SV_S)(SV_S - V)] \\
 & + H(-V_s)[(r_l - r_f)SV_S - r_l V] = 0 \\
 \text{Long Position : } & \frac{\partial V}{\partial t} + \frac{\sigma^2 S^2}{2} V_{SS} + H(-V_s)[\rho(SV_S - V)(SV_S - V)] \\
 & + H(V_s)[(r_l - r_f)SV_S - r_b V] = 0
 \end{aligned} \tag{5.15}$$

where

$$\rho(x) = \begin{cases} r_l & \text{for } x \geq 0 \\ r_b & \text{for } x < 0 \end{cases} \quad H(y) = \begin{cases} 1 & \text{for } y \geq 0 \\ 0 & \text{for } y < 0 \end{cases}$$

The derivation process of the PDEs (5.15) can be found in [8]. By modifying the Short Position above and setting $\tau = T - t$, we can then get a general HJB PDE for this optimal control problem as below

$$\frac{\partial V}{\partial \tau} = \sup_{Q \in \mathbb{Q}} \left\{ \frac{\sigma^2 S^2}{2} V_{SS} + q_3 q_1 (SV_S - V) + (1 - q_3)[(r_l - r_f)SV_S - q_2 V] \right\} \tag{5.16}$$

where $Q = \{q_1, q_2, q_3\}$ and $\mathbb{Q} = (\{r_l, r_b\}, \{r_l, r_b\}, \{0, 1\})$. The pricing HJB equation for the Long Position is evolved by replacing *sup* by *inf*.

Remark 5.1. When it is Short Position, we have $q_2 = r_l$ in (5.16). Meanwhile, there is $q_2 = r_b$ in Long Position. Therefore, the control process is two-dimensional.

For the boundary condition analysis, differential scheme choice and algorithms choosing, we also use the same analysis method as discussed in Unequal Borrowing/Lending Rates 5.5.

Table 15 shows all the parameters necessary to use in computation. Table 16 shows the data by using Policy Iteration 1 (3.16) with fully implicit time-stepping and Crank-Nicolson time-stepping. Table17 shows the results by using Policy Iteration 2 (3.17) with fully implicit time-stepping and Crank-Nicolson time-stepping. Figure 7 shows the numerical results of Short Position case. Figure 8 shows the numerical results of Long Position case. These two figures reflect a large difference between implicit time-stepping scheme and Crank-Nicolson time-stepping scheme when S is much

large than S_{target} with the same Policy Iteration 1. The reason is that the space step away from S_{target} is not small enough. And the results near S_{target} are close to each other.

Parameter	Value
r_b	0.05
r_l	0.03
r_f	0.004
T	1.0
K	100.0
$\sigma(t)$	0.30
s_{target}	100.0
s_{min}	0.0
s_{max}	500.0
$scale$	1.0
<i>Convergence Tolerance (Policy Iteration)</i>	10^{-6}

Table 15: Stock Borrowing Fees Example

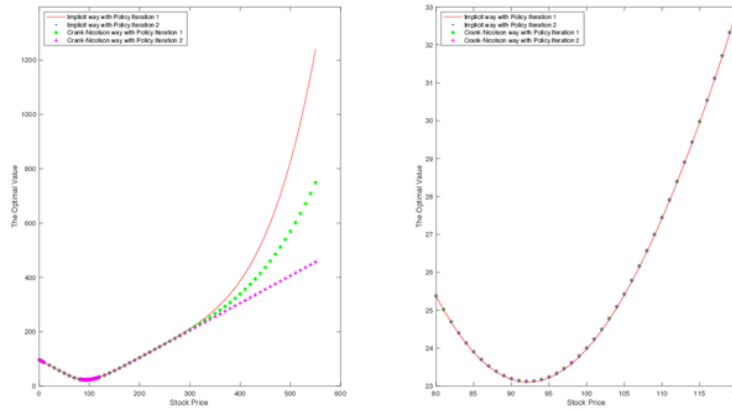


Figure 7: *Short Position case in Stock Borrowing Fees Example. The numerical results are at the accuracy level 0. As we set the spaced step near S_{target} smaller than that in other place, the points near S_{target} are dense. We show a more detailed figure for these dense points on the right.*

Level	Nodes	Timesteps	Nonlinear iterations	Option value	Change	Ratio
Fully Implicit: Short						
0	101	100	201	23.974735		
1	201	200	401	24.086269	0.111534	
2	401	400	801	24.118499	0.03223	3.4
3	801	800	1600	24.128554	0.010055	3.2
Crank-Nicolson: Short						
0	101	100	202	24.005438		
1	201	200	415	24.101989	0.096551	
2	401	400	828	24.126415	0.024426	3.9
3	801	800	1655	24.132525	0.00611	3.9
Fully Implicit: Long						
0	101	100	201	22.514086		
1	201	200	400	22.633284	0.119198	
2	401	400	800	22.667582	0.034298	3.4
3	801	800	1600	22.678194	0.010612	3.2
Crank-Nicolson: Long						
0	101	100	203	22.54526		
1	201	200	417	22.649137	0.103877	
2	401	400	830	22.675525	0.026388	3.9
3	801	800	1646	22.682162	0.006637	3.9

Table 16: Results of Stock Borrowing Fees Example at $S = 100$, $t = 0$ with Policy Iteration 1 (3.16). Convergence for fully implicit time-stepping and Crank-Nicolson time-stepping. The value at the column of Nodes is the number of spaced nodes. The Nonlinear iterations shows the iteration times after difference schemes being settled down at each nodes.

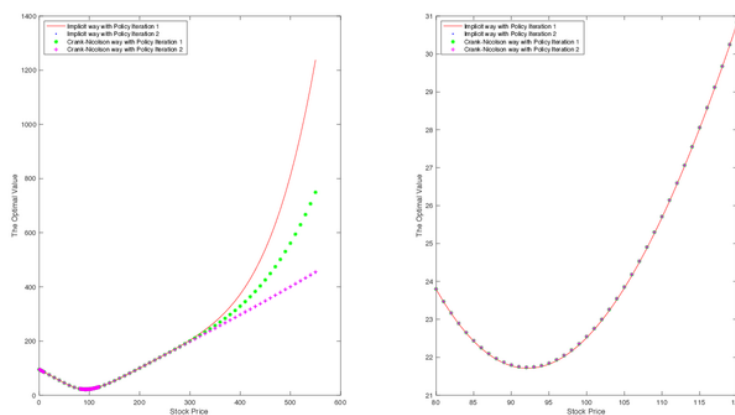


Figure 8: Long Position case in Stock Borrowing Fees Example. The numerical results are at the accuracy level 0. As we set the spaced step near S_{target} smaller than that in other place, the points near S_{target} are dense. We show a more detailed figure for these dense points on the right.

Level	Nodes	Timesteps	Option value	Change	Ratio
Fully Implicit: Short					
0	101	100	23.964579		
1	201	200	24.081047	0.116468	
2	401	400	24.115848	0.034801	3.4
3	801	800	24.127215	0.011367	3.0
Crank-Nicolson: Short					
0	101	100	24.002923		
1	201	200	24.100667	0.097747	
2	401	400	24.125732	0.025065	3.9
3	801	800	24.132175	0.006443	3.9
Fully Implicit: Long					
0	101	100	22.522665		
1	201	200	22.637735	0.11507	
2	401	400	22.669849	0.032114	3.5
3	801	800	22.67934	0.009491	3.3
Crank-Nicolson: Long					
0	101	100	22.547378		
1	201	200	22.650273	0.102895	
2	401	400	22.676115	0.025842	4.0
3	801	800	22.682465	0.00635	4.0

Table 17: *Results of Stock Borrowing Fees at $S = 100$, $t = 0$ with Policy Iteration 2 (3.17). Convergence for fully implicit time-stepping and Crank-Nicolson time-stepping. The value at the column of Nodes is the number of spaced nodes. The Nonlinear iterations shows the iteration times after difference schemes being settled down at each nodes.*

5.7 Mean-Variance Asset Allocation

Assuming that the financial market have two assets, one is a bond and the other is a stock. The dynamics process of the bond B is (2.1), and, S follows the process below

$$\frac{dS_t}{S_t} = (r + \xi\sigma)dt + \sigma dZ_1 \quad (5.17)$$

where r is the risk-free interest rate, σ is the volatility, dZ_1 is the increments of Wiener processes. The market price of volatility risk is ξ , which generates a risk premium price $\xi\sigma$ [9].

Assuming one investor want to do investment in this market, which allows bankruptcy and unbounded borrowing, over a period of T with an initial wealth of X_{target} , his purposes are gaining as much profit as possible with as little as possible risk. During this time, he also continuously

pays constant contribution C into the total investment wealth for every unit time.

Still let π denote the fraction of this wealth invested in the risky asset S_t , consequently $(1 - \pi)$ denotes the proportion of wealth invested in the bond B_t . Then, the investor's wealth process follows

$$\begin{aligned} dX_t &= [(r + \pi\xi\sigma)X_t + C]dt + \pi\sigma X_t dZ_1 \\ X(0) &= X_{target} \end{aligned} \quad (5.18)$$

According to the purposes which are maximizing the terminal expectation of the wealth and minimizing its variance, we then set the following formula as the assessment of this investment.

$$\max_{\pi \in \mathcal{K}} \{E^{t=0}[X_T | X(0) = X_{target}] - \lambda \text{Var}^{t=0}[X_T | X(0) = X_{target}]\}, \lambda \geq 0 \quad (5.19)$$

where λ is a given Lagrange multiplier, which can be interpreted as a coefficient of risk aversion, and the control problem is then to determine the optimal control π over the period T .

Remark 5.2. Note that the expectation and variance in equation (5.19) are both seen at $t = 0$. Then the optimal results of $E^{t=0}[X_T | X(0) = X_{target}]$ and $\text{Var}^{t=0}[X_T | X(0) = X_{target}]$ are $E_{\hat{\pi}}^{t=0}[X_T | X(0) = X_{target}]$ and $\text{Var}_{\hat{\pi}}^{t=0}[X_T | X(0) = X_{target}]$ respectively. For convenience, we will omit $t = 0$ sometime in the following content.

For a fixed $\gamma = \frac{1}{\lambda} + 2E_{\hat{\pi}}^{t=0}[X_T | X(0) = X_{target}]$, with $\lambda > 0$, equation (5.19) is equivalent to

$$\min_{\pi \in \mathcal{K}} E^{t=0}[(X_T - \frac{\gamma}{2})^2 | X(0) = X_{target}] \quad (5.20)$$

The proof can be seen in [9, 24, 23].

Let $J(t, x, \pi) = E[(X_T - \frac{\gamma}{2})^2 | X_t = x]$ and we define the value function of this example as below

$$V(\tau, x) = \inf_{\pi \in \mathcal{K}} J(T - \tau, x, \pi) = \inf_{\pi \in \mathcal{K}} E[(X_T - \frac{\gamma}{2})^2 | X(T - \tau) = x] \quad (5.21)$$

where $\tau = T - t$. As the process does not consider time value, then according to the (2.8) in section 2, HJB equation for this problem is

$$\begin{aligned} \frac{\partial V}{\partial \tau} &= \inf_{\pi \in \mathcal{K}} \{[(r + \pi\xi\sigma)x + C]V_x + \frac{1}{2}(\pi\sigma x)^2 V_{xx}\} \\ V(\tau = 0, x) &= (x - \frac{\gamma}{2})^2 \end{aligned} \quad (5.22)$$

After we pick an arbitrary value of γ and solve problem (5.22), the optimal control $\hat{\pi}$ is determined. Then, we can also determine the corresponding $E_{\hat{\pi}}^{t=0}[X_T | X(0) = x_{target}]$.

Let $U(\tau, x) = E[X_T | X(t = T - \tau) = x, \pi(t = T - \tau, x) = \hat{\pi}(t = T - \tau, x)]$. Then U is given from the solution to

$$\begin{aligned} \frac{\partial U}{\partial \tau} &= \{(r + \pi\xi\sigma)x + C\}U_x + \frac{1}{2}(\pi\sigma x)^2 U_{xx} \Big|_{\pi(T-\tau, x) = \hat{\pi}(T-\tau, x)} \\ U(\tau = 0, x) &= x \end{aligned} \quad (5.23)$$

Since $\hat{\pi}$ is known, then solution of equation (5.23) is very straightforward and inexpensive.

Because bankruptcy is allowed, X can be negative. Then the boundary conditions are at $x \rightarrow \pm\infty$. We still use Linear Boundary Condition to determine the asymptotic form of the solution at boundary points. By according to the power number of $V(\tau = 0, x)$ and $U(\tau = 0, x)$, we assume that

$$\begin{aligned} V(\tau, x \rightarrow \pm\infty) &= a_1(\tau)x^2 + b_1(\tau)x + c_1(\tau) \\ U(\tau, x \rightarrow \pm\infty) &= a_2(\tau)x + b_2(\tau) \end{aligned} \quad (5.24)$$

Then, taking into account the initial conditions (5.22) and (5.23), we have

$$V(\tau, x \rightarrow \pm\infty) \simeq e^{(2k_1+k_2)\tau} x^2 \quad (5.25)$$

$$U(\tau, x \rightarrow \pm\infty) \simeq e^{k_1\tau} x \quad (5.26)$$

where $k_1 = r + \pi\xi\sigma$ and $k_2 = (\pi\sigma)^2$.

Remark 5.3. Actually, we omit the terms of x whose power is smaller than the largest power number. The reason is that the value of low power terms will be much smaller than that from (5.25) and (5.26) when $x \rightarrow \pm\infty$. Therefore, we omit these parts for easy.

Then, the optimal control at $x \rightarrow \pm\infty$ can be determined by maximizing (5.25) and it is

$$\hat{\pi}(t, x \rightarrow \pm\infty) = -\frac{\xi}{\sigma} \quad (5.27)$$

Besides, [9] also gives the same control value at these boundary points. For computational purposes, we truncate the space domain $[-\infty, +\infty]$ to $[X_{min}, X_{max}]$. Then, at $X = X_{min}$ or X_{max} , we use the Linear Boundary Condition (5.25), (5.26) and (5.27) above to give the optimal value V and U .

As what previous examples did, we use fully implicit time-stepping scheme in the discretization of HJB (5.22) and Policy Iteration 1 method (3.16) to get the final optimal value V .

Similarly to V , the equation (5.23) of U can also be written into discretization form which is as below

$$\begin{aligned} \frac{U_j^{i+1} - U_j^i}{\Delta\tau} &= \{(\mathcal{L}^{\pi^{i+1}} U^{i+1})_j\}_{\pi^{i+1} = \hat{\pi}^{i+1}} \\ (\mathcal{L}^{\pi^{i+1}} U^{i+1})_j &= \alpha_j^{i+1}(\pi) U_{j-1}^{i+1} + \beta_j^{i+1}(\pi) U_{j+1}^{i+1} - (\alpha_j^{i+1}(\pi) + \beta_j^{i+1}(\pi)) U_j^{i+1} \end{aligned} \quad (5.28)$$

By applying $\hat{\pi}$ into this discretization equation, we can then get the value $U(\tau = T, X_{target})$ by using fully implicit time-stepping scheme from $\tau = 0$ to $\tau = T$. To be more specific, we use the following matrix form of the discretization (5.28) to calculate the value U at every next time τ^{i+1} step by step.

$$[I - \Delta\tau A^{i+1}(\hat{\pi}^{i+1})]U^{i+1} = U^i + (F^{i+1} - F^i) \quad (0 \leq i < m) \quad (5.29)$$

where matrixes A^{i+1} and vectors F^{i+1} , F^i are determined by both the discretization (5.28) and the boundary condition (5.26).

Remark 5.4. For the wealth case with allowing bankruptcy, we have $x < 0$. In this case, our X grid are likely contains

$$\begin{aligned} & [x_{min}, \dots, x_{-1}, x_0 = 0, x_1, \dots, x_{max}] \\ & x_{min} < \dots < x_{-1} < x_0 = 0 < x_1 < \dots < x_{max} \end{aligned} \quad (5.30)$$

with large $|x_{min}|$ and x_{max} . However, our X grid cannot contain the node $x_0 = 0$, because if $x = 0$ is in the grid, no information can be passed between the negative value nodes and positive value nodes [9]. When the space grid (5.30) above happens, we need to modify it by delete point $x_0 = 0$ and add two more points between $|x_{-1}|$ and x_1 , which are $x_{-1}^{new} = \frac{x_{-1}}{2}$ and $x_1^{new} = \frac{x_1}{2}$. After that, the grid will become

$$\begin{aligned} & [x_{min}, \dots, x_{-1}, x_{-1}^{new}, x_1^{new}, x_1, \dots, x_{max}] \\ & x_{min} < \dots < x_{-1} < x_{-1}^{new} < 0 < x_1^{new} < x_1 < \dots < x_{max} \end{aligned} \quad (5.31)$$

Finally, by combining these equations below

$$\begin{aligned} Var_{\hat{\pi}}^{t=0}[X_T|X(t=0) = X_{target}] &= V(\tau = T, X_{target}) - \frac{\gamma^2}{4} - U^2(\tau = T, X_{target}) \\ &\quad + \gamma U(\tau = T, X_{target}) \\ V(\tau = T, X_{target}) &= E_{\hat{\pi}}^{t=0}[X_T^2|X(t=0) = X_{target}] \\ &\quad - \gamma E_{\hat{\pi}}^{t=0}[X_T|X(t=0) = X_{target}] + \frac{\gamma^2}{4} \\ U(\tau = T, X_{target}) &= E_{\hat{\pi}}^{t=0}[X_T|X(t=0) = X_{target}] \end{aligned} \quad (5.32)$$

we can finally get the efficient frontier pair[9] ($Var_{\hat{\pi}}^{t=0}[X_T|X(t=0) = X_{target}]$, $E_{\hat{\pi}}^{t=0}[X_T|X(t=0) = X_{target}]$).

The algorithm for calculating efficient frontier is as below [9]

Policy Iteration for Efficient Frontier

For $\gamma = \gamma_{min}, \gamma_1, \dots, \gamma_{max}$

 For $i = 0, 1, 2, \dots, m$ (Timesteps)

 Solve equation (5.22) by using Policy Iteration 1 method (3.16)

 Solve equation (5.23) by using (5.28) where $\hat{\pi}$ is from solving V

 EndFor

 Given the initial x_{target} (5.33)

 If $(\frac{\gamma}{2} - E_{\hat{\pi}}^{t=0}[X_T|X(0) = x_{target}]_{\gamma} > 0)$ (for making sure $\lambda > 0$)

 Use (5.32) to get

$(Var_{\hat{\pi}}^{t=0}[X_T|X(0) = x_{target}], E_{\hat{\pi}}^{t=0}[X_T|X(0) = x_{target}])$

 Then get

$(Std_{\hat{\pi}}^{t=0}[X_T|X(0) = x_{target}], E_{\hat{\pi}}^{t=0}[X_T|X(0) = x_{target}])$

 EndIf

 EndFor

 Construct the upper convex hull of the points

$(Std_{\hat{\pi}}^{t=0}[X_T|X(0) = x_{target}], E_{\hat{\pi}}^{t=0}[X_T|X(0) = x_{target}])$ where $\gamma \in [\gamma_{min}, \gamma_{max}]$

In the algorithm above, we trace out the efficient frontier by varying $\gamma \in [\gamma_{min}, \gamma_{max}]$. Hence, we have to detect invalid interval for γ by using the condition that $\lambda > 0$.

As said before, $\gamma = \frac{1}{\lambda} + 2E_{\hat{\pi}}^{t=0}[X_T|X(0) = x_{target}]$ for every X_{target} and $\lambda > 0$. Then γ_{min} can be determined by letting $\lambda \rightarrow +\infty$ which means that the investor cannot bear any risk. At this situation, to minimize risk, the investor would invest all her wealth in the risk-free bond for the all time. Then, her wealth at maturity T would be $x_{target}e^{rT} + C\frac{e^{rT}-1}{r}$ with zero standard deviation. Therefore,

$$\gamma_{min} = 2(x_{target}e^{rT} + C\frac{e^{rT}-1}{r}) \quad (5.34)$$

It is obvious that there is no upper bound for γ under the condition $\lambda > 0$. For numerical purpose, we set $\gamma_{max} = 50$ [9] which is large enough to plot the efficient frontier over a reasonable range of interest.

By using the parameters in Table 18, we can get the results $V(T, x_{target})$ in Table 19 and the results $U(T, x_{target})$ in Table 20. Figure 9 shows the final efficient frontiers $(Std_{\hat{\pi}}^{t=0}[X_T|X(0) = x_{target}], E_{\hat{\pi}}^{t=0}[X_T|X(0) = x_{target}])$. It is clear that the expectation of the total wealth rises with the increase of its variance and the less risk aversion λ is, the larger the expectation and variance are.

Parameter	Value
r	0.03
T	1.0
C	0.1
$\sigma(t)$	0.15
x_{target}	1.0
x_{min}	-5925.0
x_{max}	5925
ξ	0.33
γ_{min}	2.2639
γ_{max}	50.0
$scale$	1.0
<i>Convergence Tolerance (Policy Iteration)</i>	10^{-6}

Table 18: Continuous Time Mean-Variance Asset Allocation Example

Level	Nodes	Timesteps	Nonlinear iterations	$V(T, x_{target})$	Change	Ratio
0	732	50	100	0.10199058		
1	1462	100	200	0.058185082	0.043805498	
2	2921	200	400	0.030634507	0.027550575	1.6

Table 19: Results $V(T, x_{target} = 1.0)$ of Continuous Time Mean-Variance Asset Allocation at $t = 0$, $\gamma = \gamma_{min} = 2.2639$ with Policy Iteration 1 (3.16). Convergence for fully implicit time-stepping. The value at the column of Nodes is the number of spaced nodes. The Nonlinear iterations shows the iteration times after difference schemes being settled down at each nodes

Level	Nodes	Timesteps	$Var(T, x_{target})$	$U(T, x_{target})$
0	732	50	0.31554535	1.0827391
1	1462	100	0.23991713	1.1069529
2	2921	200	0.17457723	1.1194082

Table 20: Results $U(T, x_{target} = 1.0)$ and $Var(T, x_{target} = 1.0)$ of Continuous Time Mean-Variance Asset Allocation at $t = 0$, $\gamma = \gamma_{min} = 2.2639$ with Policy Iteration 1 (3.16). Convergence for fully implicit time-stepping.

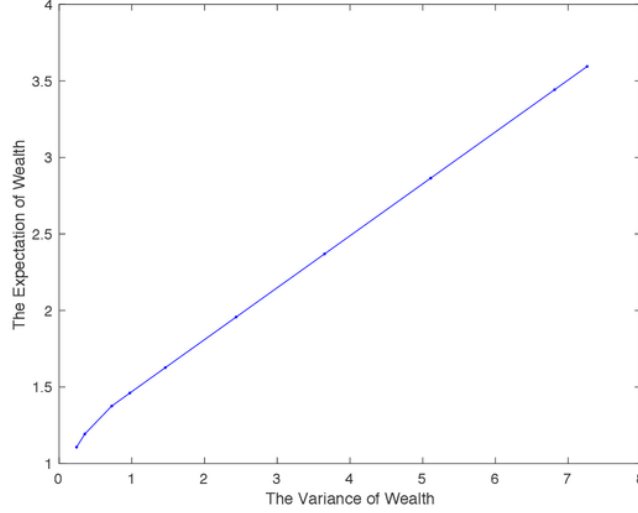


Figure 9: The final efficient frontiers ($Std_{\tilde{\pi}}^{t=0}[X_T|X(0) = 1.0]$, $E_{\tilde{\pi}}^{t=0}[X_T|X(0) = 1.0]$) at $t = 0$ with Policy Iteration 1 (3.16). Convergence for fully implicit time-stepping. The numerical results are at the accuracy level 1.

5.8 Mean-Variance Asset Allocation (Heston model)

5.8.1 Problem Formulation

Apart from optimal control problems with one-dimensional space factor, what we are going to talk about is a multi-dimensional problem which uses Heston Model [25] to process a stock price volatility [11].

The rest background is the same with section 5.7 except two points. One is that bankruptcy is prohibited in the market and there is a leverage constraint for borrowing. The other is that the process of S which follows Heston's model under the real probability measure

$$\frac{dS_t}{S_t} = (r + \xi v(t))dt + \sqrt{v(t)}dZ_1 \quad (5.35)$$

where the variance, $v(t)$, follows a mean-reverting square-root process [25]:

$$dv(t) = \kappa(\theta - v(t))dt + \sigma\sqrt{v(t)}dZ_2 \quad (5.36)$$

with dZ_1 , dZ_2 being standard Brownian motion. The correlation between Z_1 and Z_2 is $dZ_1dZ_2 = \rho dt$. The price for volatility risk is $\xi v(t)$, which generates a risk premium proportional to $v(t)$.

Still assuming one investor want to do investment in this market over a period of T with an initial wealth of W_{target} , his purposes is the same as that in last section 5.7.

π is the control process denoting the proportion invested in $S(t)$. Then, the investor's wealth process follows

$$\begin{aligned} dW(t) &= (r + \pi\xi v(t))W(t)dt + \pi\sqrt{v(t)}W(t)dZ_1 \\ W(0) &= W_{target} \end{aligned} \quad (5.37)$$

According to the purposes, we can the following assessment of this investment.

$$\min_{\pi \in \mathcal{K}} \{E^{t=0}[W_T|W(0) = W_{target}] - \lambda Var^{t=0}[W_T|W(0) = W_{target}]\}, \lambda \geq 0 \quad (5.38)$$

which is equivalent to

$$\min_{\pi \in \mathcal{K}} E^{t=0}[(W_T - \frac{\gamma}{2})^2|W(0) = W_{target}] \quad (5.39)$$

where $\gamma = \frac{1}{\lambda} + 2E_{\hat{\pi}}^{t=0}[W_T|W(0) = W_{target}]$ for every W_{target} .

Let $V(\tau, w)$ be the value function where $\tau = T - t$, then we have

$$V(\tau, w, v) = \inf_{\pi \in \mathcal{K}} E[(W(T) - \frac{\gamma}{2})^2|W(T - \tau) = w] \quad (5.40)$$

As this example does not consider time value, then the corresponding HJB equation for this problem is

$$\begin{aligned} \frac{\partial V}{\partial \tau} &= \inf_{\pi \in \mathcal{K}} \{(r + \pi\xi v)wV_w + \kappa(\theta - v)V_v + \frac{1}{2}(\pi\sqrt{vw})^2 + \pi\rho\sigma\sqrt{vw}V_{vw} + \frac{1}{2}\sigma^2vV_{vv}\} \\ V(0, w, v) &= (w - \frac{\gamma}{2})^2 \end{aligned} \quad (5.41)$$

Still let $U(\tau, w, v) = E_{\hat{\pi}}^{t=0}[W(T)|W(T - \tau) = w]$ and it is give by the following PDE

$$\begin{aligned} \frac{\partial U}{\partial \tau} &= (r + \hat{\pi}\xi v)wU_w + \kappa(\theta - v)U_v + \frac{1}{2}(\hat{\pi}\sqrt{vw})^2 + \hat{\pi}\rho\sigma\sqrt{vw}U_{vw} + \frac{1}{2}\sigma^2vU_{vv} \\ U(0, w, v) &= w \end{aligned} \quad (5.42)$$

where $\hat{\pi}$ is obtained from the solution of the HJB equation (5.41).

The boundary conditions for this problem is a little complicate.

When $v = 0$, the variance and the risk premium vanishes, then equation (5.41) is

$$\frac{\partial V}{\partial \tau} = rwV_w + \kappa\theta V_v \quad (5.43)$$

which is not related to control π and we can simply define $\hat{\pi}(\tau, w, v = 0) = 0$ as risky asset and risk-free asset are same at this situation. On the upper boundary $v \rightarrow +\infty$, the value of V will not change with the changing of v , meaning that $V_v = 0$. Then we can get

$$\frac{\partial V}{\partial \tau} = \inf_{\pi \in \mathcal{K}} \{(r + \pi\xi v)wV_w + \frac{1}{2}(\pi\sqrt{vw})^2V_{ww}\} \quad (5.44)$$

and the optimal $\hat{\pi}$ is by solving (5.44).

On $w = 0$, the equation (5.41) reduces to without π

$$\frac{\partial V}{\partial \tau} = \kappa(\theta - v)V_v + \frac{1}{2}\sigma^2vV_{vv} \quad (5.45)$$

we still can simply define $\hat{\pi}(\tau, w = 0, v) = 0$. When $w \rightarrow +\infty$, the investor prefers the risk free asset because his wealth is very large. Then, we make the assumption that $\hat{\pi} = 0$ at this boundary point because investors do not want any risk at this situation. As the initial condition in (5.41) is independent of v with the power number two, we can apply Linear Boundary Conditions to assume that asymptotic form of the solution at this boundary point is

$$V(\tau, w \rightarrow +\infty, v) = \bar{V}(w) = a_3(\tau)w^2 + b_3(\tau)w + c_3(\tau) \quad (5.46)$$

For these boundary conditions (5.43), (5.44), (5.45) and (5.46), their discretizations are shown in Appendix B [11].

For the interior points, we have to consider how to deal with the cross derivative term V_{wv} in (5.41). Since a classic finite difference method can not produce such a monotone scheme, we will adopt the wide stencil method developed in paper [3, 11] to deal with this second derivative terms. Suppose we discretize equation (5.41) at grid node (w_{j_1}, v_{j_2}) for a fixed control π , then we can get a virtual rotation of the local coordinate system clockwise by the angle η_{j_1, j_2}

$$\eta_{j_1, j_2} = \frac{1}{2} \tan^{-1} \left(\frac{2\rho\pi\sigma w_{j_1} v_{j_2}}{(\pi\sqrt{v_{j_2}} w_{j_1})^2 - (\sigma\sqrt{v_{j_2}})^2} \right) \quad (5.47)$$

And the new transformed coordinate system (y_1, y_2) is obtained by using the following matrix

$$\begin{pmatrix} w \\ v \end{pmatrix} = \begin{pmatrix} \cos(\eta_{j_1, j_2}) & -\sin(\eta_{j_1, j_2}) \\ \sin(\eta_{j_1, j_2}) & \cos(\eta_{j_1, j_2}) \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \quad (5.48)$$

We denote the rotation matrix in (5.48) as \mathbf{R}_{j_1, j_2} . Under this rotation, the second order terms in equation (5.41) are below, which is without the cross derivative term under the new transformed coordinate system (y_1, y_2) ,

$$\begin{aligned} & a_{j_1, j_2} \frac{\partial^2 \mathcal{W}}{\partial y_1^2} + b_{j_1, j_2} \frac{\partial^2 \mathcal{W}}{\partial y_2^2} \\ a_{j_1, j_2} &= \left(\frac{1}{2} (\pi\sqrt{v_{j_2}} w_{j_1})^2 \cos(\eta_{j_1, j_2})^2 + \pi\rho\sigma w_{j_1} v_{j_2} \sin(\eta_{j_1, j_2}) \cos(\eta_{j_1, j_2}) + \frac{1}{2} (\sigma\sqrt{v_{j_2}})^2 \sin(\eta_{j_1, j_2})^2 \right) \\ b_{j_1, j_2} &= \left(\frac{1}{2} (\pi\sqrt{v_{j_2}} w_{j_1})^2 \sin(\eta_{j_1, j_2})^2 - \pi\rho\sigma w_{j_1} v_{j_2} \sin(\eta_{j_1, j_2}) \cos(\eta_{j_1, j_2}) + \frac{1}{2} (\sigma\sqrt{v_{j_2}})^2 \cos(\eta_{j_1, j_2})^2 \right) \end{aligned} \quad (5.49)$$

where $\mathcal{W}(\tau, j_1, j_2)$ is the value function in the transformed coordinate system. The rotation angle η_{j_1, j_2} depends on the grid node and the control, therefore it is impossible to rotate the global coordinate system by a constant angle and build a grid over the entire space (y_1, y_2) . The local coordinate system rotation is only used to construct a virtual grid which overlays the original mesh.

Let us rewrite the HJB equation (5.41) as

$$\begin{aligned} & \sup_{\pi \in \mathcal{K}} \left\{ \frac{DV}{D\tau}(\pi) - \mathcal{L}^\pi V \right\} \\ \frac{DV}{D\tau}(\pi) &= V_\tau - (r + \pi\xi v)wV_w \\ \mathcal{L}^\pi V &= \kappa(\theta - v)V_v + \frac{1}{2}(\pi\sqrt{v}w)^2 + \pi\rho\sigma\sqrt{v}wV_{wv} + \frac{1}{2}\sigma^2 vV_{vv} \end{aligned} \quad (5.50)$$

the V_v is discretized by a standard backward or forward finite differencing discretization, depending on the sign of $\kappa(\theta - v)$ [26].

For the first term of the equation (5.50), it can be seen as the change of V with respect of τ after eliminating the influence of the change of W by time. So, for each V_{j_1, j_2}^{i+1} , we give a new off-grid point $(w_{j_1}^*, v_{j_2}) = (w_{j_1} e^{(r+\pi\xi v_{j_2})\Delta\tau^i}, v_{j_2})$ to approximate the $\frac{DV}{D\tau}(\pi)$, by using following formula

$$\frac{DV}{D\tau}(\pi) = \frac{V_{j_1, j_2}^{i+1} - V_{j_1^*, j_2}^i}{\Delta\tau^i} \quad (5.51)$$

And the second term, $\mathcal{L}^\pi V$, can be discretized into the following form which is denoted by L_h^π

$$\begin{aligned} L_h^\pi V_{j_1, j_2}^{i+1} = & \mathbf{1}_{\{\kappa(\theta - v_{j_2}) \geq 0\}} \frac{\kappa(\theta - v_{j_2})}{h} V_{j_1, j_2+1}^{i+1} - \mathbf{1}_{\{\kappa(\theta - v_{j_2}) < 0\}} \frac{\kappa(\theta - v_{j_2})}{h} V_{j_1, j_2-1}^{i+1} \\ & + \frac{a_{j_1, j_2}}{h} \mathcal{J}_h V^{i+1}(x_{j_1, j_2} + \sqrt{h}(\mathbf{R}_{j_1, j_2})_1) + \frac{a_{j_1, j_2}}{h} \mathcal{J}_h V^{i+1}(x_{j_1, j_2} - \sqrt{h}(\mathbf{R}_{j_1, j_2})_1) \\ & + \frac{b_{j_1, j_2}}{h} \mathcal{J}_h V^{i+1}(x_{j_1, j_2} + \sqrt{h}(\mathbf{R}_{j_1, j_2})_2) + \frac{b_{j_1, j_2}}{h} \mathcal{J}_h V^{i+1}(x_{j_1, j_2} - \sqrt{h}(\mathbf{R}_{j_1, j_2})_2) \\ & - (\mathbf{1}_{\{\kappa(\theta - v_{j_2}) \geq 0\}} \frac{\kappa(\theta - v_{j_2})}{h} - \mathbf{1}_{\{\kappa(\theta - v_{j_2}) < 0\}} \frac{\kappa(\theta - v_{j_2})}{h} + \frac{2a_{j_1, j_2}}{h} + \frac{2b_{j_1, j_2}}{h}) V_{j_1, j_2}^{i+1} \end{aligned} \quad (5.52)$$

where h is the discretization parameter which is defined in above in section 4, $\mathcal{J}_h V$ is an interpolant, $x_{j_1, j_2} = \begin{pmatrix} w_{j_1} \\ v_{j_2} \end{pmatrix}$ and $(\mathbf{R}_{j_1, j_2})_k$ is k -th column of the rotation matrix. For convenience, we defined

$$\begin{aligned} \Psi_{j_1, j_2}^1 &= x_{j_1, j_2} + \sqrt{h}(\mathbf{R}_{j_1, j_2})_1 \\ \Psi_{j_1, j_2}^2 &= x_{j_1, j_2} - \sqrt{h}(\mathbf{R}_{j_1, j_2})_1 \\ \Psi_{j_1, j_2}^3 &= x_{j_1, j_2} + \sqrt{h}(\mathbf{R}_{j_1, j_2})_2 \\ \Psi_{j_1, j_2}^4 &= x_{j_1, j_2} - \sqrt{h}(\mathbf{R}_{j_1, j_2})_2 \end{aligned}$$

Then, equation (5.50) will be solved backwards in time from $\tau = 0$ to $\tau = T$ by combining (5.51) and (5.52).

It is convenient to use a matrix form to represent the discretized equations for computational purposes. Let V_{j_1, j_2}^i be the approximate solution of the equation (5.41) at (w_{j_1}, v_{j_2}, i) , $1 \leq j_1 \leq N_1$, $1 \leq j_2 \leq N_2$ and $0 \leq i \leq M$. We set $V_l^i = V_{j_1, j_2}^i$ where $l = i + (j_1 - 1)N_1$. The final matrix form of the discretized equations is then

$$\begin{aligned} [I - \Delta\tau^i \mathbf{L}^{i+1}(\hat{\pi})] V^{i+1} &= \Phi^{i+1}(\pi) V^i + \Delta\tau^i \mathbf{G}^{i+1}(\pi) + F^{i+1} - F^i \\ \hat{\pi}_l &\in \arg \min_{\pi \in \mathcal{K}} [\Phi^{i+1}(\pi) V^i + \Delta\tau^i (\mathbf{L}^{i+1}(\pi) V^{i+1} + \mathbf{G}^{i+1}(\pi))]_l \end{aligned} \quad (5.53)$$

For the boundary points, $[\mathbf{L}^{i+1}(\hat{\pi})]_l$ is developed from $D_h^\pi V_{j_1, j_2}^{i+1}$, while $\mathbf{L}^{i+1} V^{i+1}$ and \mathbf{G}^{i+1} consist of $L_h^\pi V^{i+1}$ at unboundary points. As said before, the corresponding discretized form of

D_p^h is given in Appendix B. $\mathbf{G}^{i+1}(\pi)$ is used to adjust the situation when the new system points $\Psi_{j_1, j_2}^{1,2,3,4}$ are out of the space domain. And $\mathbf{G}^{i+1}(\pi)$ is defined as

$$\mathbf{G}^{i+1}(\pi)_l = \begin{cases} \mathbf{1}_{\{\Psi_{j_1, j_2}^1 \notin \text{domain}\}} \frac{a_{j_1, j_2}}{h} \bar{V}(\Psi_{j_1, j_2}^1) \\ + \mathbf{1}_{\{\Psi_{j_1, j_2}^2 \notin \text{domain}\}} \frac{a_{j_1, j_2}}{h} \bar{V}(\Psi_{j_1, j_2}^2) \\ + \mathbf{1}_{\{\Psi_{j_1, j_2}^3 \notin \text{domain}\}} \frac{b_{j_1, j_2}}{h} \bar{V}(\Psi_{j_1, j_2}^3) \\ + \mathbf{1}_{\{\Psi_{j_1, j_2}^4 \notin \text{domain}\}} \frac{b_{j_1, j_2}}{h} \bar{V}(\Psi_{j_1, j_2}^4) & \text{for } (w_{j_1}, v_{j_2}) \text{ in domain} \\ 0 & \text{otherwise} \end{cases} \quad (5.54)$$

$\Phi^{i+1}(\pi)V^i$ gives the new value V after adjusting grid point (w_{j_1}, v_{j_2}) to new off-grid point $(w_{j_1^*}, v_{j_2^*})$. Specifically, it is as following

$$[\Phi^{i+1}(\pi)V^i]_l = \begin{cases} \mathcal{J}_h^i V_{j_1^*, j_2^*}^i & \text{for } (w_{j_1^*}, v_{j_2^*}) \text{ in domain} \\ \text{boundary condition(5.46)} & \text{otherwise} \end{cases} \quad (5.55)$$

By using above discretization method, we then can get final $V(T, x_{\text{target}})$ and optimal control process $\hat{\pi}$. Consequently, $U(T, x_{\text{target}}) = E_{\hat{\pi}}^{t=0}[W_T]$ will be got. And we can then get the variance of W_T , $Var(T, x_{\text{target}})$, by using (5.32). After using the same process as in 5.7 of calculating γ_{\min} , we can then get $\gamma_{\min} = 2W_{\text{target}}e^{rT}$. Still using the Policy Iteration for Efficient Frontier (5.32), we can finally obtain the efficient frontier pair of this example.

5.8.2 Convergence to the Viscosity Solution

We still illustrate the convergence of this example from four parts, Strong Comparison Property, Stability, Monotonicity and Consistency by using the theorem 4.1. The detailed proof can be found in [3, 11].

- (Strong Comparison Property) As the control π is bounded in this example, we can get that (5.41) satisfies strong comparison property which has been said in section 4.
- (Stability) From (5.53), it is obvious that $[I - \Delta\tau^i \mathbf{L}^{i+1}(\hat{\pi})]$ has positive diagonals and non-positive off-diagonals. Besides, and the sum of l_{th} row on the matrix is larger than zero. Then, according to [28], this is an M-matrix and diagonally dominant. Finally, from [29], we can get the Stability property by using a straightforward maximum analysis as in .
- (Monotonicity) Since we use a wide stencil to deal with the second derivative term in the L_h^π in (5.52), we can get the proof of monotonicity from [27]. Besides, \mathcal{J}_h^i , the linear interpolant in the semi-Lagrangian time-stepping scheme also ensures monotonicity.
- (Consistency) Since we either use the limitation PDE of equation (5.41) at $w = 0$ and $v = 0$ or use Linear Boundary Conditions at $w \rightarrow +\infty$ and $v \rightarrow +\infty$, we can then use the

same proof in [27] to show consistency at both boundary points and the interior. For the points $(w_{j_1^*}, v_{j_2})$, as this point can only exceed upper boundary w_{max} where we can use the asymptotic solution (5.46) from Linear Boundary Conditions, thus, we do not need the more general definition of consistency [3] to handle the boundary data.

5.8.3 Numerical Results

Table 23 shows all the parameters necessary to use in computation. Table 21 shows the results $V(T, x_{target})$ by using Policy Iteration 1 (3.16) with fully implicit time-stepping. Table 22 shows the results $U(T, x_{target})$ by using Policy Iteration 1 (3.16) with fully implicit time-stepping. Figure 10 shows the final efficient frontiers ($Std_{\tilde{\pi}}^{t=0}[W_T|W(0) = w_{target}], E_{\tilde{\pi}}^{t=0}[W_T|W(0) = w_{target}]$). It is obvious that the expectation of wealth increases with the rise of variance when w_{target} and v_{target} are fixed. When the number of the W Nodes, V Nodes and Timesteps increase together, the results will come up after a thousands-dimensional vector converging for hundreds times, which would take a few days to finish on my computer. Therefore, we only give the following results.

Level	W Nodes	V Nodes	Timesteps	$V(T, x_{target})$
0	13	11	100	6.580916
1	22	21	200	2.6709329

Table 21: Results $V(T, x_{target})$ of Mean-Variance Asset Allocation with Heston Model at $W = 98.0$, $t = 0$, $\gamma = \gamma_{min} = 201.969$ with Policy Iteration 1 (3.16). Convergence for fully implicit time-stepping.

Level	W Nodes	V Nodes	Timesteps	$Var(T, x_{target})$	$U(T, x_{target})$
0	13	11	100	6.5807664	100.99677
1	22	21	200	2.6679456	101.0392

Table 22: Results $U(T, x_{target})$ of Mean-Variance Asset Allocation with Heston Model at $W = 98.0$, $t = 0$, $\gamma = \gamma_{min} = 201.969$ with Policy Iteration 1 (3.16). Convergence for fully implicit time-stepping.

Parameter	Value
r	0.03
T	1.0
κ	5.07
θ	0.0457
σ	0.48
ρ	-0.767
ξ	1.605
γ_{min}	201.969
γ_{max}	1000.0
p_{min}	0.0
p_{max}	2.0
w_{min}	0.0
w_{max}	400
v_{min}	0.0
v_{max}	0.047
w_{target}	98.0
v_{target}	0.0456
$scale$	1.0
<i>Convergence Tolerance (Policy Iteration)</i>	10^{-6}

Table 23: Mean-Variance Asset Allocation Example (Heston Model) Parameters

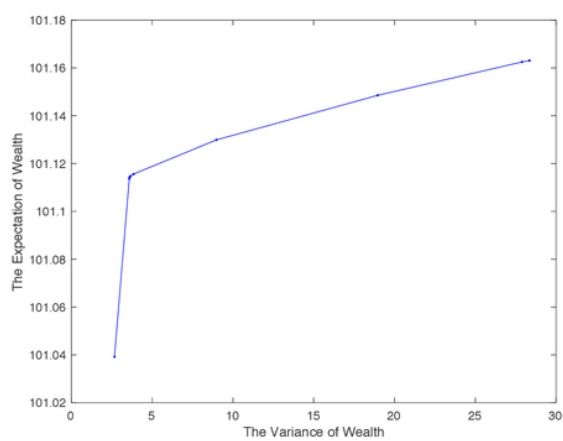


Figure 10: The final efficient frontiers ($Std_{\pi}^{t=0}[W_T|W(0) = 98.0]$, $E_{\pi}^{t=0}[W_T|W(0) = 98.0]$) at $t = 0$ with Policy Iteration 1 (3.16). Convergence for fully implicit time-stepping. The numerical results are at the accuracy level 1.

A Appendix

Let π_i^n denote the optimal control $\hat{\pi}$ at node i , time level n and set

$$a_i^{n+1} = a(z_i, \pi_i^n), \quad b_i^{n+1} = b(z_i, \pi_i^n), \quad c_i^{n+1} = c(z_i, \pi_i^n),$$

Then, we can use central, forward or backward differencing at any node. Central Differencing:

$$\alpha_{i,central}^n = \left[\frac{2a_i^n}{(z_i - z_{i-1})(z_{i+1} - z_{i-1})} - \frac{b_i^n}{z_{i+1} - z_{i-1}} \right]$$

$$\beta_{i,central}^n = \left[\frac{2a_i^n}{(z_{i+1} - z_i)(z_{i+1} - z_{i-1})} + \frac{b_i^n}{z_{i+1} - z_{i-1}} \right]$$

Forward/backward Differencing: ($b_i^n > 0 / b_i^n < 0$)

$$\alpha_{i,forward/backward}^n = \left[\frac{2a_i^n}{(z_i - z_{i-1})(z_{i+1} - z_{i-1})} + \max\left(0, \frac{b_i^n}{z_i - z_{i-1}}\right) \right]$$

$$\beta_{i,forward/backward}^n = \left[\frac{2a_i^n}{(z_{i+1} - z_i)(z_{i+1} - z_{i-1})} + \max\left(0, \frac{b_i^n}{z_{i+1} - z_i}\right) \right]$$

B Appendix

At the boundary condition, cross-derivative term vanishes. Then we can use a standard finite difference method. The discrete forms of (5.43), (5.44) and (5.45) are like

$$D_h^\pi V_{i,j}^n = \alpha_{i,j}^w V_{i-1,j}^n + \beta_{i,j}^w V_{i+1,j}^n + \alpha_{i,j}^v V_{i,j-1}^n + \beta_{i,j}^v V_{i,j+1}^n - (\alpha_{i,j}^w + \beta_{i,j}^w + \alpha_{i,j}^v + \beta_{i,j}^v) V_{i,j}^n$$

where $\alpha_{i,j}^w$, $\beta_{i,j}^w$, $\alpha_{i,j}^v$ and $\beta_{i,j}^v$ are defined as follows

$$\alpha_{i,j}^w = \frac{(\sqrt{v}\pi w_i)^2}{(w_i - w_{i-1})(w_{i+1} - w_{i-1})}$$

$$\beta_{i,j}^w = \frac{(\sqrt{v}\pi w_i)^2}{(w_{i+1} - w_i)(w_{i+1} - w_{i-1})}$$

$$\alpha_{i,j}^v = \frac{(\sqrt{v}\sigma)^2}{(v_i - v_{i-1})(v_{i+1} - v_{i-1})} + \max\left(0, -\frac{\kappa(\theta - v_j)}{v_j - v_{j-1}}\right)$$

$$\beta_{i,j}^v = \frac{(\sqrt{v}\sigma)^2}{(v_{i+1} - v_i)(v_{i+1} - v_{i-1})} + \max\left(0, \frac{\kappa(\theta - v_j)}{v_{j+1} - v_j}\right)$$

The coefficients $\alpha_{i,j}^w$, $\beta_{i,j}^w$, $\alpha_{i,j}^v$ and $\beta_{i,j}^v$ are all non-negative, and is compatible with a monotone scheme. On the upper boundary $v = v_{max}$, the coefficients α_{i,N_2}^v , β_{i,N_2}^v degenerate to zero, and On the lower boundary $w = 0$, $\alpha_{1,j}^w$, $\beta_{1,j}^w$ are set to 0. On the lower boundary $v = 0$, $\alpha_{i,1}^v = 0$, $\beta_{i,1}^v = 0$, $\alpha_{i,1}^v$ and $\beta_{i,1}^v = \frac{\kappa\theta}{v_{j+1}-v_j}$ and $j = 1$.

Conclusion

This report talks about a numerical approach used to solve optimal control problems with nonlinear HJB PDEs. Although there have been some methods to get analytic solutions of HJBs, we still need to explore the numerical approach because it demands less constraint conditions.

We introduce the optimal control problems and HJB PDEs in the section 2 by giving a specific situation. The detailed numerical method which we use in the whole section 5 is shown in section 3.2. Section 5 gives eight specific optimal control problems.

The first example, Merton portfolio with boundary control, is the basic optimal control problem. The numerical results in it are really close to the close-form solution. Besides, Turnpike problem 1 5.2 also shows that this numerical method can deal with HJB accurately. By extending the finite investment period to infinity and applying this condition in the third example, Turnpike Problem 2 5.3, we find that the numerical results converge really slow to the close-form value with the raise of Timesteps. The reason is that parameter T does not satisfy the condition $T \rightarrow +\infty$. However, when we increase the accuracy of time steps by decreasing T , we get a quick convergence. Then, we may guess that other optimal control problems with infinity investment period could be solved numerically by using the same way.

The rest five examples are without close-form expressions of solutions. Example 5.4, 5.5 and 5.6 gives both *sup* and *inf* cases. In particular, example 5.6 shows the situation when there are more than one control factor. The results at the target point in this example still converge well. Example 5.7 shows the case when the control is unbounded and the financial market allows bankruptcy. The conclusion deduced from the data are consistent with the existing knowledge. The last example has two stochastic variables and the numerical results from it also conform the existed knowledge which is that the more risk investors can burden, the larger the expectation and variance are.

Overall, the numerical method in section 3.2 can be widely used to deal with various optimal control problems, as long as it satisfies strong comparison property, stability, consistency and monotonicity.

References

- [1] G. Barles. Convergence of numerical schemes for degenerate parabolic equations arising in finance. In L. C. G. Rogers and D. Talay, editors, *Numerical Methods in Finance*, pages 1-21. Cambridge University Press, Cambridge, 1997.
- [2] G. Barles, CH. Daher, and M. Romano. Convergence of numerical schemes for parabolic equations arising in finance theory. *Mathematical Models and Methods in Applied Sciences*, 5:125-143, 1995.
- [3] G. Barles and P.E. Souganidis. Convergence of approximation schemes for fully nonlinear equations. *Asymptotic Analysis*, 4:271-283, 1991.
- [4] M. G. Crandall, H. Ishii, and P. L. Lions. User's guide to viscosity solutions of second order partial differential equations. *Bulletin of the American Mathematical Society*, 27:1-67, 1992.
- [5] Karatzas I., Shreve, S.E.. *Methods of Mathematical Finance*, Springer, 1998.
- [6] Baojun Bian and Harry Zheng. Turnpike Property and Convergence Rate for an Investment Model with General Utility Functions. 2014.
- [7] Huyn Pham, *Continuous-time Stochastic Control and Optimaization with Financial Application*, Springer, 2009.
- [8] P.A. Forsyth, G. Labahn. *Numerical Methods for Controlled Hamilton-Jacobi-Bellman PDEs in Finance*, 2007.
- [9] Wang J., Forsyth P.A., Numerical solution of the Hamilton-Jacobi-Bellman formulation for continuous time mean variance asset allocation. Forthcoming[J]. *Journal of Economic Dynamics and Control*, 2008.
- [10] Barles, G. and J. Burdeau. The Dirichlet problem for semilinear second-order degenerate elliptic equations and applications to stochastic exit time control problems. *Communications in Partial Differential Equations*20, 1995.
- [11] Forsyth P.A., Ma K.. Numerical solution of the Hamilton-Jacobi-Bellman formulation for continuous-time mean-variance asset allocation under stochastic volatility[J]. *Social Science Electronic Publishing*, 2016.
- [12] Bian B , Miao S , Zheng H . Smooth Value Functions for a Class of Nonsmooth Utility Maximization Problems[J]. *SIAM Journal on Financial Mathematics*, 2010.
- [13] Windcliff H., Forsyth P.A. and K.R. Vetzal. Analysis Of The Stability Of The Linear Boundary Condition For The Black-Scholes Equation[J].*Journal of Computational Finance*, 2003.

- [14] M. Avellaneda, A. Levy and A. Paras. Pricing and hedging derivative securities in markets with uncertain volatilities, *Applied Mathematical Finance*, 2006.
- [15] Forsyth P.A., Vetzal K.R. Numerical Methods for Nonlinear PDEs in Finance. *Handbook of Computational Finance*. Springer Berlin Heidelberg, 2012.
- [16] P. Wilmott, Derivatives, *John Wiley and Sons*, West Sussex, England, 1998.
- [17] D. Tavella and C. Randall, Pricing Financial Instruments: The Finite Difference Method, *John Wiley and Sons, Inc.*, 2000.
- [18] Wang, J. and P. Forsyth. Maximal use of central differencing for Hamilton-Jacobi-Bellman PDEs in finance. *SIAM Journal on Numerical Analysis*, 2007.
- [19] Chaumont, S. A strong comparison result for viscosity solutions to Hamilton-Jacobi-Bellman equations with Dirichlet conditions on a non-smooth boundary. Working paper, Institute Elie Cartan, Universite Nancy I, 2007.
- [20] Barles, G. and P. Souganidis. Convergence of approximation schemes for fully nonlinear equations. *Asymptotic Analysis* , 1991.
- [21] Barles, G. and E. Rouy. A strong comparison result for the Bellman equation arising in stochastic exit time control problems and applications. *Communications in Partial Differential Equations* 23, 1998.
- [22] Barles, G. and E. Jakobsen. Error bounds for monotone approximation schemes for parabolic Hamilton-Jacobi-Bellman equations. Working Paper, Norwegian University of Science and Technology, 2005.
- [23] Li, D. and W.L. Ng. Optimal dynamic portfolio selection: Multiperiod mean variance formulation. *Mathematical Finance*, 2000.
- [24] Zhou, X. and D. Li. Continuous time mean variance portfolio selection: A stochastic LQ framework. *Applied Mathematics and Optimization*, 2000.
- [25] Cox, J. C., J. Ingersoll, E. Jonathan, and S. A. Ross . A theory of the term structure of interest rates. *Econometrica: Journal of the Econometric Society* 53(2),1985.
- [26] Feller, W.. Two singular diffusion problems. *Annals of mathematics* 54 (1), 1951.
- [27] Ma, K. and P.A. Forsyth. An unconditionally monotone numerical scheme for the two factor uncertain volatility model. Working paper, Cheriton School of Computer Science, University of Waterloo, 2014.
- [28] Varga, R. S.. Matrix Iterative Analysis, Volume 27. Springer-Verlag, Berlin, 2009.

-
- [29] d'Halluin, Y., P. A. Forsyth, and G. Labahn. A penalty method for American options with jump diffusion processes. *Numerische Mathematik* 97 (2), 2004.

Optimal Investment With Discretionary Stopping And Trading Constraints

GRADEMARK REPORT

FINAL GRADE

/0

GENERAL COMMENTS

Instructor

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

PAGE 35

PAGE 36

PAGE 37

PAGE 38

PAGE 39

PAGE 40

PAGE 41

PAGE 42

PAGE 43

PAGE 44

PAGE 45

PAGE 46

PAGE 47

PAGE 48

PAGE 49

PAGE 50

PAGE 51
