Determining the Optimal Control When Numerically Solving Hamilton-Jacobi-Bellman PDEs in Finance

by

Zhe Wang

An essay
presented to the University of Waterloo
in fulfillment of the
research requirement for the degree of
Master of Mathematics
in
Computational Mathematics

Waterloo, Ontario, Canada, 2012

I hereby declare that I am the sole author of this essay. This is a true copy of the essay, including any required final revisions, as accepted by my examiners.

I understand that my essay may be made electronically available to the public.

Abstract

Numerous financial problems can be posed as nonlinear Hamilton-Jacobi-Bellman (HJB) partial differential equations (PDEs). In order to guarantee that the discretized equations converge to the viscosity solution, the required conditions are pointwise consistency, l_∞ stability, and monotonicity. We use the positive coefficient method, choosing central differencing as much as possible, to construct a monotone scheme. Fully implicit timestepping method is used because it is unconditionally stable. However, this method generates nonlinear algebraic equations. To solve the nonlinear algebraic equations, we must find the optimal control that maximizes the objective function at each node. In this paper, we will investigate three methods to numerically find the optimal control: the linear search method, Brent's method and the LQ (Linear, Quadratic) method. We also investigate a new discretization method, which results in a continuous local objective function. We illustrate these methods using a pension plan example. Numerical experiments indicate that both Brent's method and the LQ method outperform the linear search method. The LQ method is preferable to Brent's method because Brent's method requires convex functions in order to guarantee convergence.

Acknowledgements

I would like to thank my supervisors, Peter Forsyth and George Labahn, under whose guidance and help I completed this project. I would also like to thank all my fellows in the Computational Mathematics department. Special thanks to my boyfriend, Yi Yang, for his support throughout the year.

Dedication

This is dedicated to my beloved family.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Many financial problems can be treated as optimal control problems, resulting in nonlinear Hamilton-Jacobi-Bellman (HJB) partial differential equations (PDEs). Typical examples of nonlinear HJB PDEs include American options [6], passport options [9] and unequal borrowing/lending costs [3]. In general, there are no analytical solutions for nonlinear PDEs. However, there exist many approaches in which numerical methods can be used to solve such PDEs.

Kushner proposed an approach that is based on a controlled Markov chain approximation [7]. A Markov chain is essentially equivalent to an explicit finite difference method. Due to the timestep limitations of an explicit method, implicit methods have been developed. In this essay, we focus on the implicit discretization method described in [3, 9].

Usually, a nonlinear PDE has many solutions, so it is necessary to ensure that the discretization converges to the financially relevant solution termed the viscosity solution [1]. It can be shown that the viscosity solution is the solution to the problem posed as a dynamic programming problem. The required conditions for discrete schemes to ensure convergence to the viscosity solution are pointwise consistency, l_∞ stability, and monotonicity [9]. The theoretical proof can be found in [3].

To construct a monotone scheme, we use the positive coefficient method. All the coefficients depend on the choice of using central or upstream differencing for the drift term (i.e., the first order derivative term). In terms of the convergence rate, for smooth functions, central differencing results in a second order method, and upstream results in a first order method. To discretize the PDE, we should use central differencing as much as possible in order to achieve a high convergence rate (at least for smooth functions). We use upstream only when the central differencing generates negative coefficients [9].

In order to have an unconditionally stable scheme, a fully implicit timestepping method is commonly used. However, the implicit method gives rise to nonlinear equations at each timestep. Therefore, we need a method to solve nonlinear algebraic equations. A common technique is policy iteration [8], which approximates the solution by solving linear equations at each iteration. To ensure that the policy iteration converges, we must find the optimal control. Since the discretization at each node is a function of the control, finding the optimal control means searching for the control that maximizes the local objective function at each node.

In this essay, we will examine three different methods to numerically determine the optimal control.

The first method is linear search, originally proposed in [9]. Since the coefficients of the local objective function at each node depend on the control, the type of discretization (e.g. central or upstream differencing) may change and this results in a discontinuous objective function. Wang [9] suggests discretizing the control (assuming the admissible control space is compact) and then using linear search.

In order to ensure convergence as the grid is refined, we increase the number of discrete controls at each refinement. As the controls are refined, linear search becomes expensive. The main purpose of this essay is to investigate two optimization methods that can improve efficiency when finding the optimal control.

Since choosing between central or upstream at each node and timestep depends on the control, the local objective function is in general a discontinuous function of the control. However, with the weighted average discretization [6], the local objective function is a continuous (but non-smooth) function of the control. We will use this method [6] in this essay.

The first local optimization method we investigate is called Brent's method [4]. However, this method is not guaranteed to converge unless the local objective function is convex. We will demonstrate that, in general, the local objective function is not convex, even with the weighted average discretization.

The weighted average discretization is piecewise linear or quadratic. Thus, we can determine the local optimal control of each piecewise interval easily and then compare them to determine the local maximum. We will term this method LQ (Linear, Quadratic) in the following chapters. This technique is guaranteed to converge.

As an example, we will use the pension plan model [2] to demonstrate the above methods.

The main results of this essay are the following:

- We outline the discretization of the pension plan model [2].
- We show that the weighted average discretization [6] generates a continuous local objective function.
- We implemented the linear search method as in [9] to validate the discretization method.
- We compare two techniques for computing the maximum of the local objective function at each node: Brent's method and the LQ technique.

Chapter 2

Preliminaries

In this essay, we use the pension plan model in [2] in order to illustrate the numerical techniques. We outline this model in this chapter.

2.1 Background of the Pension Plan Model

The default strategy for most pension providers is deterministic lifestyling. Initially, all contributions are invested into high-risk assets such as equity funds. At a predetermined date, for example, ten years before retirement, the contributions will be transferred gradually to low-risk assets such as cash and bond funds. The purpose of this strategy is to hedge market fluctuations in the long term and provide a positive return to investors at the maturity date [2].

However, the above strategy is not optimal in terms of maximizing the terminal return. Cairns *et al* [2] proposed a new strategy to ensure an optimal terminal utility. The key feature taken into account is the plan member's lifetime salary. Thus, instead of maximizing the total wealth, the purpose is to find the optimal utility, which is a function of wealth and salary. This optimal asset allocation dynamically adjusts the proportion invested between the low-risk and high-risk assets according to the plan members' stochastic salary. The control in this example represents the proportion of the pension wealth invested in risky assets. More details about this stochastic model are described in [2].

2.2 Stochastic Model

Suppose there are two different underlying assets in the pension plan: one is risk-free (e.g. a cash or bond fund); the other is risky (e.g. an equity fund). The risk-free asset $B(t)$ can be priced as follows:

$$B(t) = B(0)\exp(rt), \tag{2.1}$$

where r is the constant risk-free rate of interest and t is the time.

The increment of the risky asset follows the stochastic differential equation (SDE):

$$dS(t) = S[(r + \xi_1\sigma_1)dt + \sigma_1 dZ_1], \tag{2.2}$$

where Z_1 is the standard Brownian motion, σ_1 is the volatility, r is the constant interest rate, and ξ_1 is the market price of risk.

Every year, plan members contribute a fraction of π of their salary Y to the pension plan. We assume that the increment of salary Y at time t follows the SDE:

$$dY(t) = (r + \mu_Y)Y dt + \sigma_{Y_0} Y dZ_0 + \sigma_{Y_1} Y dZ_1, \tag{2.3}$$

where μ_Y, σ_{Y_0}, and σ_{Y_1} are constants, and Z_0 and Z_1 are two independent Brownian motions.

The plan members' pension wealth is denoted by $W(t)$. In this case, the control variable is the proportion q of the wealth invested in the risky assets. It follows that the proportion invested in the risk-free assets is $1 - q$. The wealth process follows the SDE:

$$dW(t) = W[(r + q\xi_1\sigma_1)dt + q\sigma_1 dZ_1] + \pi Y dt. \tag{2.4}$$

As explained in the previous section, our goal is to maximize the terminal utility, $u(W(T), Y(T))$, which depends on the terminal salary and terminal wealth:

$$u(W(T), Y(T)) = \left\{ \begin{array}{ll} \frac{1}{\gamma}(\frac{W(T)}{Y(T)})^\gamma & \text{where } \gamma < 1 \text{ and } \gamma \neq 0 \\ \log(\frac{W(T)}{Y(T)}) & \text{when } \gamma = 0 \end{array} \right\}. \tag{2.5}$$

Define the ratio of the wealth $W(t)$ and the salary $Y(t)$ as $X(t) = W(t)/Y(t)$, and then apply Ito's Lemma to formulate the SDE for $X(t)$:

$$\begin{aligned} dX =& [\pi + X(-\mu_Y + q\sigma_1(\xi_1 - \sigma_{Y1}) + \sigma_{Y1}^2 + \sigma_{Y0}^2)]dt \\ & - \sigma_{Y0} X dZ_0 + X(q\sigma_1 - \sigma_{Y1})dZ_1. \end{aligned} \tag{2.6}$$

Define $J(t, x, q) = E[u(X_q(T))|X(t) = x]$, where $X_q(t)$ is the path of X given the optimal control $q(t)$. We work backwards from the terminal time T and set $\tau = T - t$. It follows that

$$V(x, \tau) = \sup_{q \in \hat{Q}} E[u(X_q(T))|X(T - \tau) = x] = \sup_{q \in \hat{Q}} J(T - \tau, x, q), \tag{2.7}$$

where $\hat{Q}$ is the set of admissible controls.

Equation (2.7) mathematically shows that the goal is to maximize the terminal utility with the optimal control. Following standard techniques, the solution of Equation (2.7) is equivalent to the viscosity solution of the HJB PDE:

$$V_\tau = \sup_{q \in \hat{Q}} \{\mu_x^q V_x + \frac{1}{2}(\sigma_x^q)^2 V_{xx}\}, \tag{2.8}$$

subject to terminal condition

$$V(x, \tau = 0) = \left\{ \begin{array}{ll} \frac{1}{\gamma} x^\gamma & \text{where } \gamma < 1 \text{ and } \gamma \neq 0 \\ \log(x) & \text{when } \gamma = 0 \end{array} \right\}, \tag{2.9}$$

where

$$\mu_x^q = \pi + x(-\mu_Y + q\sigma_1(\xi_1 - \sigma_{Y1}) + \sigma_{Y1}^2 + \sigma_{Y0}^2), \tag{2.10}$$

$$(\sigma_x^q)^2 = x^2(\sigma_{Y0}^2 + (q\sigma_1 - \sigma_{Y1})^2). \tag{2.11}$$

The derivation of equation (2.8) from equation (2.7) is presented in [1].

2.3 Boundary Conditions and Terminal Condition

By substituting $x = 0$ into the PDE in (2.8), we obtain

$$V_\tau(x = 0, \tau) = \pi V_x. \tag{2.12}$$

As x goes to infinity, the boundary condition becomes

$$V(x \to \infty, \tau) = 0. \tag{2.13}$$

when $\gamma < 0$. For $\gamma > 0$ (a case not used in this essay), we can use the terminal condition as a Dirichlet boundary condition as $x \to \infty$.

For the purpose of computation, we restrict x to the domain $[0, x_{\max}]$, where $x_{\max}$ is large enough that it does not affect the solution.

Since the terminal condition in (2.9) is undefined at $x = 0$, we modify the terminal condition by replacing x by $\max(x, \epsilon)$, where ϵ is a very small positive number. Consider the case when $\gamma < 0$, then the modified terminal condition becomes

$$V(x, \tau = 0) = \left\{ \begin{array}{ll} \frac{1}{\gamma} \max(x, \epsilon)^\gamma & \text{where } \gamma < 0 \\ \log(\max(x, \epsilon)) & \text{when } \gamma = 0 \end{array} \right\}. \tag{2.14}$$

Tests show that $\epsilon = 10^{-3}$ results in solutions, which are accurate to at least seven digits [9].

Chapter 3

Discretization

3.1 Implicit Discretization

The general form of this optimal control problem is

$$V_\tau = \sup_{q \in \hat{Q}} \{\mathcal{L}^q V\}, \tag{3.1}$$

where we define

$$\mathcal{L}^q V = a(x,\tau,q)V_{xx} + b(x,\tau,q)V_x - c(x,\tau,q)V, \tag{3.2}$$

and the coefficients $a(x,\tau,q)$, $b(x,\tau,q)$ and $c(x,\tau,q)$ are:

$$a(x,\tau,q) = \frac{1}{2}(\sigma_x^q)^2; \quad b(x,\tau,q) = \mu_x^q; \quad c(x,\tau,q) = 0. \tag{3.3}$$

We apply the fully implicit timestepping method, which is unconditionally stable, to discretize this HJB equation. We discretize the computational domain x_i, $i = 0, ..., m$; $\tau^n = n\Delta\tau$. The discrete solution at each node (x_i, τ^n) is denoted by V_i^n, which is an approximation to $V(x_i, \tau^n)$. In addition, we define $V^n = [V_0^n, ..., V_m^n]'$.

Using fully implicit timestepping, we obtain

$$\frac{V_i^{n+1} - V_i^n}{\Delta\tau} = \sup_{q \in \hat{Q}} \{(\mathcal{L}^q V^{n+1})_i\}. \tag{3.4}$$

After applying finite differencing to discretize the drift term V_x and the diffusion term V_{xx} in Equation (3.4), we obtain

$$\frac{V_i^{n+1} - V_i^n}{\Delta\tau} = \sup_{q \in \hat{Q}} \{\alpha_i^{n+1}(q^{n+1}) V_{i-1}^{n+1} + \beta_i^{n+1}(q^{n+1}) V_{i+1}^{n+1} - (\alpha_i^{n+1}(q^{n+1}) + \beta_i^{n+1}(q^{n+1})) V_i^{n+1}\}. \tag{3.5}$$

Depending on the choice of using central or upstream differencing of the drift term, α and β have different expressions at each node i and timestep n.

Central differencing gives

$$\begin{aligned} \alpha_{i,central} &= \left[\frac{[\sigma_x^q]_i^2}{(x_i - x_{i-1})(x_{i+1} - x_{i-1})} - \frac{[\mu_x^q]_i}{x_{i+1} - x_{i-1}} \right] \\ \beta_{i,central} &= \left[\frac{[\sigma_x^q]_i^2}{(x_{i+1} - x_i)(x_{i+1} - x_{i-1})} + \frac{[\mu_x^q]_i}{x_{i+1} - x_{i-1}} \right]. \end{aligned} \tag{3.6}$$

Upstream differencing (forward/backward differencing) generates

$$\begin{aligned} \alpha_{i,upstream} &= \left[\frac{[\sigma_x^q]_i^2}{(x_i - x_{i-1})(x_{i+1} - x_{i-1})} + \max(0, \frac{-[\mu_x^q]_i}{x_i - x_{i-1}}) \right] \\ \beta_{i,upstream} &= \left[\frac{[\sigma_x^q]_i^2}{(x_{i+1} - x_i)(x_{i+1} - x_{i-1})} + \max(0, \frac{[\mu_x^q]_i}{x_{i+1} - x_i}) \right]. \end{aligned} \tag{3.7}$$

3.2 Boundary Conditions

Recall that at $x = 0$, we have

$$V_\tau(x = 0, \tau) = \pi V_x. \tag{3.8}$$

This PDE can be discretized by forward differencing:

$$\frac{V_i^{n+1} - V_i^n}{\Delta\tau} = \pi \frac{V_{i+1}^{n+1} - V_i^{n+1}}{x_{i+1} - x_i}. \tag{3.9}$$

According to the boundary condition (2.13), $V = 0$ at $x = x_{\max}$.

3.3 Matrix Form

For computational purposes, we define $V^{n+1} = [V_0^{n+1}, ..., V_m^{n+1}]'$ and $Q^{n+1} = [q_0^{n+1}, ..., q_m^{n+1}]'$, where every q_i^{n+1} is a local optimal control. Then we can write the right-hand side of Equation (3.5) in the matrix form as

$$\begin{aligned} &[A^{n+1}(Q^{n+1}, V^{n+1})V^{n+1}]_i \\ &= \alpha_i^{n+1}(q^{n+1})V_{i-1}^{n+1} + \beta_i^{n+1}(q^{n+1})V_{i+1}^{n+1} - (\alpha_i^{n+1}(q^{n+1}) + \beta_i^{n+1}(q^{n+1}))V_i^{n+1}, \end{aligned} \tag{3.10}$$

where $0 < i < m$.

To completely construct the matrix A, we need to modify the first and last row to satisfy the boundary conditions as stated in Section 3.2. Thus, the general form of the matrix A is

$$A = \begin{bmatrix} -\frac{\pi}{(x_1-x_0)} & \frac{\pi}{(x_1-x_0)} & 0 & \cdots & \cdots & \cdots & 0 \\ \alpha_1 & -\alpha_1 - \beta_1 & \beta_1 & 0 & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 & 0 & 0 \\ 0 & 0 & \alpha_i & -\alpha_i - \beta_i & \beta_i & 0 & 0 \\ 0 & 0 & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & \alpha_{m-1} & -\alpha_{m-1} - \beta_{m-1} & \beta_{m-1} \\ 0 & 0 & \cdots & \cdots & \cdots & 0 & 0 \end{bmatrix}.$$

We define

$$F^{n+1}(Q, V^{n+1}) = A^{n+1}(Q, V^{n+1})V^{n+1}. \tag{3.11}$$

Then the approximation of V, $V^n = [V_0^n, ..., V_m^n]'$, at each timestep can be updated in the matrix form as

$$[I - \Delta\tau A^{n+1}(Q^{n+1}, V^{n+1})]V^{n+1} = V^n, \tag{3.12}$$

where

$$Q^{n+1} = [q_0^{n+1}, ..., q_m^{n+1}]', \tag{3.13}$$

and

$$q_i^{n+1} = \arg\sup_{q \in \hat{Q}} \{[F^{n+1}(Q, V^{n+1})]_i\}. \tag{3.14}$$

3.4 Monotone Scheme

To construct a monotone scheme, it is important to ensure that the discretization (3.10) is a positive coefficient discretization [3]:

$$\alpha_i^{n+1}(q^{n+1}) \geq 0, \ \beta_i^{n+1}(q^{n+1}) \geq 0, \quad i = 0, ..., m \ \text{ and } \ \forall q \in \hat{Q}, \tag{3.15}$$

where $m+1$ is the total number of grid points.

In this section, two discretization methods will be discussed: using central differencing as much as possible and weighted average.

3.4.1 Using Central Differencing As Much As Possible

We want to use central differencing as much as possible because this method converges at a higher rate than using only upstream [9]. The algorithm for using central differencing as much as possible is stated in Algorithm 1.

Algorithm 1 Using Central Differencing As Much As Possible

if $\alpha_{i,central} \geq 0$ and $\beta_{i,central} \geq 0$ **then**

$\alpha_i = \alpha_{i,central}$ and $\beta_i = \beta_{i,central}$

else

$\alpha_i = \alpha_{i,upstream}$ and $\beta_i = \beta_{i,upstream}$

end if

3.4.2 Weighted Average

The basic idea of the weighted average discretization is to use a combination of central and upstream differencing. The smallest possible weight is used for upstream weighting, which results in a positive coefficient method.

The weighted average discretization is given in Algorithm 2.

There are three possible algebraic expressions for α_i and β_i which depend on the following cases:

Algorithm 2 Weighted Average

for i **do** = 1,2,3,...

  w = 1;

  if $\alpha_{i,central} < 0$ **then**

    $w = \frac{\alpha_{i,upstream}}{\alpha_{i,upstream} - \alpha_{i,central}}$

  else

    if $\beta_{i,central} < 0$ **then**

      $w = \frac{\beta_{i,upstream}}{\beta_{i,upstream} - \beta_{i,central}}$

    end if

  end if

  $\alpha_i = w * \alpha_{i,central} + (1 - w) * \alpha_{i,upstream}$

  $\beta_i = w * \beta_{i,central} + (1 - w) * \beta_{i,upstream}$

end for

1. $\alpha_{i,central} \geq 0$ and $\beta_{i,central} \geq 0$;

2. $\alpha_{i,central} < 0$ and $\beta_{i,central} > 0$;

3. $\alpha_{i,central} > 0$ and $\beta_{i,central} < 0$.

It is impossible to have both $\alpha_{i,central} < 0$ and $\beta_{i,central} < 0$ because, when $\alpha_{i,central} < 0$, we have

$$0 < \frac{[\sigma_x^q]_i^2}{(x_i - x_{i-1})(x_{i+1} - x_{i-1})} < \frac{[\mu_x^q]_i}{x_{i+1} - x_{i-1}}, \tag{3.16}$$

and then

$$\beta_{i,central} = \frac{[\sigma_x^q]_i^2}{(x_{i+1} - x_i)(x_{i+1} - x_{i-1})} + \frac{[\mu_x^q]_i}{x_{i+1} - x_{i-1}} > 0. \tag{3.17}$$

Similarly, when $\beta_{i,central} < 0$, then $\alpha_{i,central} > 0$.

If $\alpha_{i,central} \geq 0$ and $\beta_{i,central} \geq 0$, we can use just $\alpha_{i,central}$ and $\beta_{i,central}$ as the coefficients:

$$\begin{aligned} \alpha_i &= \alpha_{i,central} \\ \beta_i &= \beta_{i,central}. \end{aligned} \tag{3.18}$$

When $\alpha_{i,central} < 0$, we have

$$\begin{aligned} \alpha_i &= 0 \\ \beta_i &= \frac{[\mu_x^q]_i}{x_{i+1} - x_i} + \frac{[\sigma_x^q]_i^2}{(x_{i+1} - x_i)(x_{i+1} - x_{i-1})} \\ &\quad + \frac{[\sigma_x^q]_i^2}{(x_i - x_{i-1})(x_{i+1} - x_{i-1})} - \frac{[\sigma_x^q]_i^2}{(x_i - x_{i-1})(x_{i+1} - x_i)} \\ &= \frac{[\mu_x^q]_i}{x_{i+1} - x_i} \\ &= \beta_{i,wt}. \end{aligned} \tag{3.19}$$

Similarly, when $\beta_{i,central} < 0$,

$$
\begin{aligned}
\alpha_i &= \frac{-[\mu_x^q]_i}{x_i - x_{i-1}} + \frac{[\sigma_x^q]_i^2}{(x_{i+1} - x_i)(x_{i+1} - x_{i-1})} \\
&\quad + \frac{[\sigma_x^q]_i^2}{(x_i - x_{i-1})(x_{i+1} - x_{i-1})} - \frac{[\sigma_x^q]_i^2}{(x_i - x_{i-1})(x_{i+1} - x_i)} \\
&= \frac{-[\mu_x^q]_i}{x_i - x_{i-1}} \\
&= \alpha_{i,wt} \\
\beta_i &= 0.
\end{aligned} \tag{3.20}
$$

Chapter 4

Iterative Solution Method

Equation (3.12) is a nonlinear algebraic equation resulted from the implicit timestepping method. A popular approach for solving nonlinear algebraic equations at each iteration is policy iteration.

4.1 Policy Iteration

The algorithm for policy iteration is shown in Algorithm 3. To avoid excessive computation when the value is too small, the term *scale* in Algorithm 3 is one for plans priced in dollars. We also assume the weighted average discretization is used, in which case $A = A(Q)$, as opposed to using central differencing as much as possible [5].

The detailed proof of the convergence of this iteration is described in [8]. The following is a brief sketch of the proof. First, the iteration step in Algorithm 3 can be rearranged as

$$\begin{aligned} &[I - \Delta\tau A^{n+1}(Q^k)](\hat{V}^{k+1} - \hat{V}^k) \\ &= \Delta\tau[A^{n+1}(Q^k)\hat{V}^k - A^{n+1}(Q^{k-1})\hat{V}^k]. \end{aligned} \tag{4.1}$$

We have

$$q_i^k = \arg\sup_{q \in \hat{Q}} \{[F^{n+1}(Q, \hat{V}^k)]_i\}, \tag{4.2}$$

Algorithm 3 Policy Iteration

$(V^{n+1})^0 = V^n$;

$\hat{V}^k = (V^{n+1})^k$;

for k = 0,1,2,... until converge **do**

$[I - \Delta\tau A^{n+1}(Q^k)]\hat{V}^{k+1} = V^n$;

where $Q^k = [q_0^k, ..., q_m^k]'$;

$q_i^k = \arg\sup_{q \in \hat{Q}} \{[F^{n+1}(Q, \hat{V}^k)]_i\}$;

if $(k > 0)$ and $(\max_i \frac{|\hat{V}_i^{k+1} - \hat{V}_i^k|}{\max(scale, |\hat{V}_i^{k+1}|)} < tolerance)$ **then**

quit

end if

end for

where $F(Q, V^n)$ is defined in Equation (3.11), and we denote $[F^{n+1}(Q, \hat{V}^k)]_i$ as the local objective function at each node.

Equation (4.2) indicates that q_i^k is the value maximizing $[F^{n+1}(Q, \hat{V}^k)]_i$, so any other control cannot maximize the objective function, e.g. q_i^{k-1}. Hence, the right-hand side of Equation (4.1) is non-negative, i.e.,

$$\Delta\tau[A^{n+1}(Q^k)\hat{V}^k - A^{n+1}(Q^{k-1})\hat{V}^k]_i \geq 0. \tag{4.3}$$

In addition, it is easy to verify that $[I - \Delta\tau A^{n+1}(Q^k)]$ is an M-matrix because it has positive diagonals, non-positive off-diagonals and is diagonally dominant. By the property of an M-matrix, the inverse matrix of an M-matrix is non-negative, i.e., $[I - \Delta\tau A^{n+1}(Q^k)]^{-1} \geq 0$. Knowing these two results, we can conclude that $\hat{V}^{k+1} - \hat{V}^k \geq 0$, which means we have a non-decreasing sequence. From [8], we know that $\hat{V}^k$ is also a bounded sequence. Hence, we prove that this iteration converges to the solution.

Note that the key step in this proof is to verify that the control q at each iteration maximizes the objective function $[F^{n+1}(Q, \hat{V}^k)]_i$. Consequently, in next chapter, we will focus on three different numerical methods to find the optimal control.

Chapter 5

Optimal Control

In this chapter, we discuss and compare three methods of computing the optimal control. The first method is linear search, which is introduced briefly in Section 5.2. In the case that the weighted average discretization is used, we have more flexibility regarding the choice of optimization method. We investigate two alternative local optimization methods to find the optimal control in the following two sections: (i) Brent's method; and (ii) the LQ method.

5.1 Continuity of Weighted Average

The objective function, which needs to be maximized at each node, is

$$[F^{n+1}(Q, \hat{V}^k)]_i = \alpha_i^{n+1}(q)V_{i-1}^k + \beta_i^{n+1}(q)V_{i+1}^k - (\alpha_i^{n+1}(q) + \beta_i^{n+1}(q))V_i^k, \tag{5.1}$$

where α_i and β_i are functions of the control q.

If we follow Algorithm 1, using central differencing as much as possible, we obtain a possibly discontinuous objective function of the control q at each node since the type of discretization (e.g. central or upstream differencing) may change. Figure 5.1 shows an example of a discontinuous objective function using Algorithm 1, where the subinterval before and after the discontinuity is formed by upstream and central differencing, respectively.

However, a continuous function can be achieved by using the weighted average discretization specified in Algorithm 2. Figure 5.2 demonstrates an example of a continuous

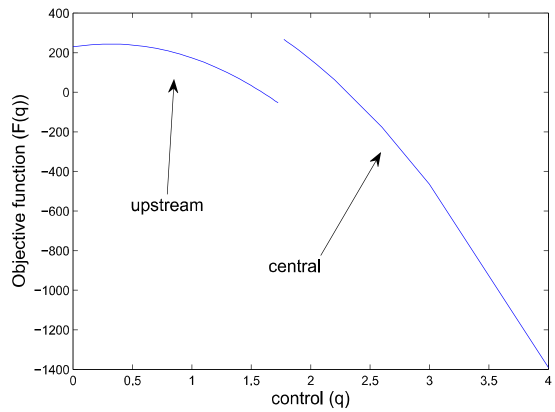

Figure 5.1: Discontinuous objective function of the control (using central as much as possible)

(but non-smooth) function of the control, in which the subinterval before the change point is shifted to weighted averaging, but after the change point, it uses central differencing.

According to the three cases discussed in Section 3.4.2, the weighted average discretization manipulates the objective function to the following form:

$$\begin{cases} [F_{cent}]_i = \alpha_{i,central}(V_{i-1} - V_i) + \beta_{i,central}(V_{i+1} - V_i) & \text{if } (\alpha_{i,central} \geq 0) \text{ and } (\beta_{i,central} \geq 0) \\ [F_{wt}]_i = \beta_{i,wt}(V_{i+1} - V_i) & \text{if } (\alpha_{i,central} < 0) \text{ and } (\beta_{i,central} > 0) \\ [F_{wt}]_i = \alpha_{i,wt}(V_{i-1} - V_i) & \text{if } (\alpha_{i,central} > 0) \text{ and } (\beta_{i,central} < 0) \end{cases}$$

where $\alpha_{i,wt}$ and $\beta_{i,wt}$ are defined in Equation (3.19, 3.20).

We can mathematically prove that the weighted average algorithm generates a continuous function. The potential discontinuities occur when $\alpha_{i,central} = 0$ or $\beta_{i,central} = 0$. Let us first consider the case when $\alpha_{i,central} = 0$. In this case, we have (from Equation (3.6))

$$\frac{[\sigma_x^q]_i^2}{(x_i - x_{i-1})(x_{i+1} - x_{i-1})} = \frac{[\mu_x^q]_i}{x_{i+1} - x_{i-1}}, \tag{5.2}$$

$$\Rightarrow \frac{[\sigma_x^q]_i^2}{(x_i - x_{i-1})} = [\mu_x^q]_i. \tag{5.3}$$

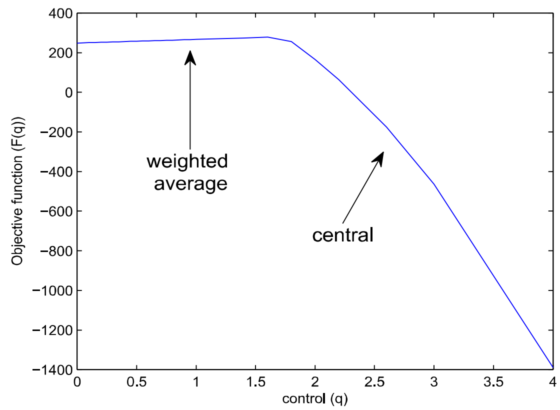

Figure 5.2: Continuous objective function of the control (using weighted average)

Substituting Equation (5.3) into the expression of $\beta_{i,wt}$ in Equation (3.19) gives

$$\begin{aligned}\beta_{i,wt} &= \frac{[\mu_x^q]_i}{x_{i+1}-x_{i-1}} + \frac{[\sigma_x^q]_i^2}{(x_{i+1}-x_i)(x_{i+1}-x_{i-1})} \\ &+ \frac{1}{(x_{i+1}-x_i)}[[\mu_x^q]_i - \frac{[\sigma_x^q]_i^2}{(x_i-x_{i-1})}] \\ &= \frac{[\mu_x^q]_i}{x_{i+1}-x_{i-1}} + \frac{[\sigma_x^q]_i^2}{(x_{i+1}-x_i)(x_{i+1}-x_{i-1})} \\ &= \beta_{i,central}. \end{aligned} \quad (5.4)$$

Since $\beta_{i,central} = \beta_{i,wt}$ at $\alpha_{i,central} = 0$, $[F_{central}]_i = [F_{wt}]_i$ at the potential discontinuity; therefore, the objective function is continuous when $\alpha_{i,central} = 0$.

Analogously, when $\beta_{i,central} = 0$, it follows that

$$\frac{[\sigma_x^q]_i^2}{(x_{i+1}-x_i)(x_{i+1}-x_{i-1})} = \frac{-[\mu_x^q]_i}{x_{i+1}-x_{i-1}}. \quad (5.5)$$

By manipulating the expression of $\alpha_{i,wt}$ in (3.20), we can show easily $\alpha_{i,central} = \alpha_{i,wt}$ at $\beta_{i,central} = 0$. Therefore, we conclude that the weighted average algorithm generates continuous (but non-smooth) objective functions of the control.

In the following, we use the weighted average discretization.

5.2 Linear Search

In this model, the control q represents the fraction of the wealth invested in risky asset. Thus, a reasonable range of the control can be set as

$$\hat{Q} = [0, q_{\max}]. \tag{5.6}$$

q cannot become infinitely large in practice, so we choose the $q_{\max}$ to be large enough such that it does not affect the results [9].

The idea of the linear search method is to non-uniformly discretize the control in the above range $[0, q_{\max}]$. We want to choose more points in the domain $[0, 2]$ because information changes quickly in this range. The optimal control, the one that maximizes the objective function, can be found by performing a linear search.

Originally, this method was proposed in [9]. We implemented and verified the claimed results. The linear search method is very easy to implement, yet costly to run. In terms of performance, computing the objective function is the most time-consuming part of the process. The bottleneck of the linear search method is that as the number of controls increases, the time complexity increases proportionally. This fact will be demonstrated in Section 6.2.

5.3 Brent's Method

Brent's method is a hybrid numerical method of finding extremes (maximum/minimum) by combining golden section search and inverse quadratic interpolation, and retains the advantages of both and achieves a stable and high convergence rate [4]. The details of Brent's method are shown in Appendix A.

We implemented Brent's method to find the optimal control of each local objective function and the results are demonstrated in Chapter 6. Nevertheless, Brent's method is not guaranteed to converge unless the objective function is convex/concave. Generally speaking, the objective function is not always convex/concave even with a weighted average discretization.

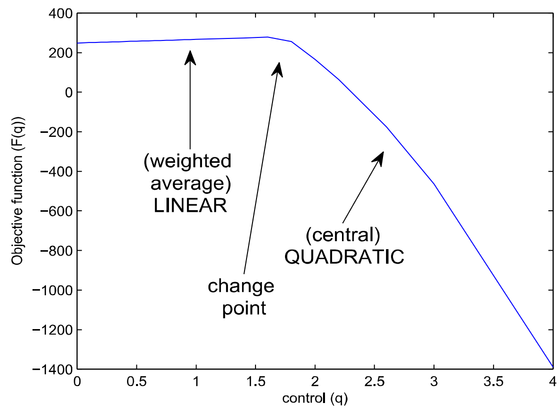

Figure 5.3: Example of weighted average

5.4 the LQ Method

Recall that after applying the weighted average discretization, the objective function becomes

$$\begin{cases} [F_{cent}]_i = \alpha_{i,central}(V_{i-1} - V_i) + \beta_{i,central}(V_{i+1} - V_i) & \text{if } (\alpha_{i,central} \geq 0) \text{ and } (\beta_{i,central} \geq 0) \\ [F_{wt}]_i = \beta_{i,wt}(V_{i+1} - V_i) & \text{if } (\alpha_{i,central} < 0) \text{ and } (\beta_{i,central} > 0) \\ [F_{wt}]_i = \alpha_{i,wt}(V_{i-1} - V_i) & \text{if } (\alpha_{i,central} > 0) \text{ and } (\beta_{i,central} < 0). \end{cases}$$

Since the second order term q^2 appears in both $\alpha_{i,central}$ and $\beta_{i,central}$, $[F_{cent}]_i$ is a quadratic function of the control q. Similarly, $[F_{wt}]_i$ a is a linear function of q because $\beta_{i,wt}$ and $\alpha_{i,wt}$ contain only the first order term q. As illustrated in Figure 5.3, the weighted average discretization is piecewise linear or quadratic: central differencing gives a quadratic function of the control, and weighted average generates a linear one. The curve is continuous but not smooth.

In order to find the maximum, the first step is to determine the change points. The change points occur when $\alpha_{i,central} = 0$ or $\beta_{i,central} = 0$. Recall the equation for $\alpha_{i,central}$

$$\alpha_{i,central} = \frac{[\sigma_x^q]_i^2}{(x_i - x_{i-1})(x_{i+1} - x_{i-1})} - \frac{[\mu_x^q]_i}{x_{i+1} - x_{i-1}}. \tag{5.7}$$

After expanding Equation (5.7), we can see that $\alpha_{i,central}$ is a quadratic function of the control q:

$$\alpha_{i,central} = a_\alpha q^2 + b_\alpha q + c_\alpha, \tag{5.8}$$

where

$$\begin{aligned}
a_\alpha &= \frac{x_i^2 \sigma_1^2}{(x_i - x_{i-1})(x_{i+1} - x_{i-1})} \\
b_\alpha &= \frac{-2x_i^2 \sigma_1 \sigma_{Y1} - x_i \sigma_1 (\xi_1 - \sigma_{Y1})(x_i - x_{i-1})}{(x_i - x_{i-1})(x_{i+1} - x_{i-1})} \\
c_\alpha &= \frac{x_i^2 \sigma_{Y0}^2 + x_i^2 \sigma_{Y1}^2}{(x_i - x_{i-1})(x_{i+1} - x_{i-1})} - \frac{\pi + x_i(-\mu_Y + \sigma_{Y0}^2 + \sigma_{Y1}^2)}{(x_{i+1} - x_{i-1})}.
\end{aligned} \tag{5.9}$$

Similarly, $\beta_{i,central}$ is also a quadratic function of the control:

$$\beta_{i,central} = a_\beta q^2 + b_\beta q + c_\beta, \tag{5.10}$$

where

$$\begin{aligned}
a_\beta &= \frac{x_i^2 \sigma_1^2}{(x_{i+1} - x_i)(x_{i+1} - x_{i-1})} \\
b_\beta &= \frac{-2x_i^2 \sigma_1 \sigma_{Y1} + x_i \sigma_1 (\xi_1 - \sigma_{Y1})(x_{i+1} - x_i)}{(x_{i+1} - x_i)(x_{i+1} - x_{i-1})} \\
c_\beta &= \frac{x_i^2 \sigma_{Y0}^2 + x_i^2 \sigma_{Y1}^2}{(x_{i+1} - x_i)(x_{i+1} - x_{i-1})} + \frac{\pi + x_i(-\mu_Y + \sigma_{Y0}^2 + \sigma_{Y1}^2)}{(x_{i+1} - x_{i-1})}.
\end{aligned} \tag{5.11}$$

Consequently, determining the change points is equivalent to finding the roots of the above equations (5.8, 5.10). Since both $\alpha_{i,central}$ and $\beta_{i,central}$ are quadratic functions of the control q, each has, at most, two real roots. Thus, four real roots are in the interval $[0, q_{\max}]$ at most according to the property of quadratic equations.

The next step is to determine the local optimal control of each piecewise interval. If the subinterval is linear, the local maximum must be at either end point, so we substitute the two ends into the corresponding objective function and choose the end point with the higher function value. If the subinterval is quadratic, the local optimal control, i.e., extreme point, can be determined explicitly from the quadratic objective function. Note that the extreme point may not lie within the range of $[a, c]$. When this happens, the maximum is at one of the end points of the interval.

Now that we have the local optimal controls of all piecewise intervals, the final optimal control q will be the one that yields the largest function value.

Chapter 6

Numerical Results

The parameters used in the pension plan model are shown in Table 6.1. All the experiments in this chapter are based on the weighted average discretization.

Table 6.1: Parameters used in the pension plan example. The time is in years.

tolerance	10^{-7}
ϵ	10^{-3}
$q_{\max}$	200
$x_{\max}$	80
μ_Y	0
σ_1	0.2
σ_{Y0}	0.05
T	20
ξ_1	0.2
σ_{Y1}	0.05
π	0.1
γ	-5

In Table 6.1, *tolerance* is used to determine the terminal condition in Algorithm 3; $x_{\max}$ is the largest value of x in the domain; $q_{\max}$ is the maximized value of q used in the linear search method.

6.1 Verification of the Linear Search Method

In order to verify the validity of the linear search method in [9] and use it as a basis for the comparison of those two improved methods, we implemented the linear search method (with the weighted average discretization) and present the results in Table 6.2.

Table 6.2: Results for the linear search method

$x - Node$	$q - Node$	$Timesteps(N)$	$Utility$
	x = 0		
189	109	640	-0.003653053
377	217	2560	-0.003580836
753	433	10240	-0.003563587
	x = 1		
189	109	640	-0.000431663
377	217	2560	-0.000426846
753	433	10240	-0.000425619

The numerical results obtained in Table 6.2 are extremely close to the results reported in [9]. In this experiment, we refine the grid by inserting a grid node between every two grid nodes and reduce the timestep size by four.

6.2 Numerical Results for Brent's Method and the LQ Method

After verifying the basic method (linear search), we examine the other two methods, Brent's method and the LQ method. Table 6.3 demonstrates the experimental results. In this experiment, we insert a grid node between every two grid nodes and reduce the timestep size by two at each level of refinement.

The first column in Table 6.3 represents the number of grid nodes in x, the second column the number of timesteps (i.e., $N = T/\Delta\tau$), and the third column the utility. As the number of grid nodes and timesteps increases, the utility values computed by all three methods converge. These methods yield consistent results, agreeing with each other up to seven digits. Hence, we conclude that these two improved methods converge to the viscosity solution, as expected.

Table 6.3: Numerical results for three methods: linear search, Brent's and LQ

$x - Node$	$Timesteps(N)$	$Utility$
	x = 0	
	Linear Search	
95	320	-0.003859615
189	640	-0.003653053
377	1280	-0.003606185
753	2560	-0.003582497
	Brent's Method	
95	320	-0.003827772
189	640	-0.003651308
377	1280	-0.003605979
753	2560	-0.003582445
	LQ	
95	320	-0.003827772
189	640	-0.003651308
377	1280	-0.003605979
753	2560	-0.003582445
	x = 1	
	Linear Search	
95	320	-0.000459486
189	640	-0.000431663
377	1280	-0.000428345
753	2560	-0.000426739
	Brent's Method	
95	320	-0.000456354
189	640	-0.000431528
377	1280	-0.000428313
753	2560	-0.000426732
	LQ	
95	320	-0.000456354
189	640	-0.000431528
377	1280	-0.000428313
753	2560	-0.000426732

In terms of performance, computing the objective function costs the most. Therefore, the improvement made by Brent's method and the LQ method focuses mainly on decreasing the number of times the objective function is evaluated. The number of evaluated functions at each node and each timestep is shown in Table 6.4.

Table 6.4: The number of evaluated functions at each node and timestep

Refinement	Linear Search	Brent's Method	LQ
I = 1	28	30	5
I = 2	55	22	4
I = 3	109	17	4
I = 4	217	14	3
I = 5	433	13	3

The first column in Table 6.4 is the level of refinement. For instance, $I = 1$ means the number of grids in $x - Node$ is 48 and the timestep is 160. As the level of refinement increases, a new grid node is inserted in between every two grid nodes, and the timestep size is reduced by two. The rest of the three columns contain the number of evaluated functions at each node and timestep for all three methods, respectively.

As indicated in the second column (in Table 6.4), since the linear search method evaluates the objective function for each possible control, the number of evaluated objective functions increases as the grid is refined. Again, computing the objective function involves the most time, and the linear search method has expensive computational cost.

The third column in Table 6.4 indicates that the number of evaluated functions for Brent's method decreases as the grid is refined, thereby resulting in reduced computational cost. As the grid is refined, central differencing is used more often, making the objective function smoother. Mathematically speaking, using central differencing requires $\alpha_{i,central} \geq 0$, which means

$$\frac{[\sigma_x^q]_i^2}{(x_i - x_{i-1})(x_{i+1} - x_{i-1})} \geq \frac{[\mu_x^q]_i}{x_{i+1} - x_{i-1}}, \tag{6.1}$$

which is equivalent to

$$\frac{[\sigma_x^q]_i^2}{(x_i - x_{i-1})} \geq [\mu_x^q]_i. \tag{6.2}$$

As the grid is refined, the denominator $(x_i - x_{i-1})$ in Equation (6.2) becomes smaller, making the left-hand side in Equation (6.2) larger. Additionally, as noted in [2], because the amount invested in the high-risk assets (i.e., $q(t)x$) converges to zero as x goes to zero, the coefficients (σ_x^q and μ_x^q) in Equation (6.2) are bounded. Consequently, it is easier for Equation (6.2) to hold.

The smoother the objective function is, the more often inverse quadratic interpolation is used compared to golden section search. With inverse quadratic interpolation being more efficient, fewer evaluations are needed to find the optimal control.

The last column in Table 6.4 indicates that the number of evaluated functions for the LQ method also decreases as the grid is refined. The change points in the LQ method occur when $\alpha_{i,central} = 0$ or $\beta_{i,central} = 0$. Because of the choice of the parameters shown in Table 6.1, $\beta_{i,central}$ is always positive in this experiment (according to the expression of $\beta_{i,central}$ in Equation (3.6)). Thus, $\alpha_{i,central} = 0$ is the only case where the change points can possibly occur. In the worst case, there are six evaluations at each node and timestep: two at each change point; plus two at the end points, 0 and q_{max}; plus two at each quadratic interval. As the grid is refined, central differencing is used more often (as discussed above), and we will have even fewer change points than in the worst case scenario.

In conclusion, Brent's method and the LQ method outperform the linear search method. Since the LQ method does not require a convex objective function in order to guarantee convergence, this method is preferable to Brent's method.

Chapter 7

Conclusion

Numerous financial problems can be formulated in nonlinear Hamilton-Jacobi-Bellman (HJB) partial differential equations (PDEs). There is no analytical solution for a nonlinear HJB PDE. Therefore, we need to find numerical solutions by discretizing the PDE. Usually, we do not have unique numerical solutions. Hence, some techniques are needed to ensure that the discretization converges to the viscosity solution. The required conditions for convergence are pointwise consistency, l_∞ stability, and monotonicity.

A monotone scheme can be constructed by a positive coefficient method. The equation coefficients depend on the choice of using central or upstream differencing. In practice, we use central as much as possible because of the high convergence rate. A fully implicit timestepping method is used because it generates an unconditionally stable scheme. Nevertheless, the implicit method produces nonlinear algebraic equations at each iteration. Thus, policy iteration is used to solve the nonlinear discretized equations. To make sure that the policy iteration converges, finding the optimal control (which maximizes the local objective function) is required.

We first verified the linear search method as originally proposed in [9]. The linear search method is easy to implement but has expensive computational cost. Since the linear search method evaluates the objective function for each possible control, the average number of evaluated objective functions at each node and timestep increases as the grid is refined. Most of the computing effort is spent on evaluating the objective functions, so the linear search method is expensive.

To achieve better performance than the linear search method, we propose two new methods: Brent's method and the LQ method. With the numerical results converging to the viscosity solution, it can be concluded that the two improved methods correctly find the optimal control. Furthermore, we find that the number of evaluated functions at each node and timestep for both Brent's method and the LQ method decrease as the grid is refined, thereby improving the computing speed. This is because, as the grid is refined, central differencing is used more often, smoothing the objective function.

In summary, Brent's method and the LQ method outperform the linear search method in terms of computational efficiency. The LQ method is preferable, since it does not require a convex objective function.

When we use the LQ method in this model, the objective function becomes either a quadratic or linear function of the control. If the objective function is more complicated, this problem will be hard to solve. The basic idea of using the LQ method is still to separate the intervals and treat them individually. Solving complicated functions of the control for each piecewise interval is considered future work.

APPENDICES

Appendix A

Brent's Method

Brent's method is a hybrid numerical method for finding extremes (maximum/minimum) by combining golden section search and inverse quadratic interpolation.

Inverse quadratic interpolation requires three known points, say $(a, f(a))$, $(b, f(b))$ and $(c, f(c))$. Then we can apply inverse quadratic interpolation through the above points to estimate the extreme point, which in this case is the minimum. This estimation is formulated as follows:

$$x = b - \frac{(b-a)^2[f(b)-f(c)] - (b-c)^2[f(b)-f(a)]}{2(b-a)[f(b)-f(c)] - (b-c)[f(b)-f(a)]}. \tag{A.1}$$

Inverse quadratic interpolation converges fast but becomes unstable easily. Therefore, we need another method, golden section search, to supplement it.

In golden section search, we need to have three points $a < b < c$ such that $f(a) > f(b)$ and $f(c) > f(b)$ and

$$\frac{c-b}{b-a} = \varphi \simeq 1.6. \tag{A.2}$$

This initial set-up brackets the searching interval in $[a, c]$. The next point d can be evaluated by

$$\frac{c-d}{d-b} = \varphi. \tag{A.3}$$

The updated search interval depends on $f(d)$. More specifically, if $f(d)$ is greater than $f(b)$, the subinterval should be $[a, d]$; otherwise, it should be $[b, c]$. After updating the subinterval, we repeat the computation as illustrated in Equation (A.3).

Golden section search is more stable than inverse quadratic interpolation, but much slower. Thus, Brent's method takes the advantages of both and achieves a stable convergence of high rate. If the new estimated point evaluated by inverse quadratic interpolation is not "reasonable" as specified in [4], it switches to golden section search; otherwise, it keeps using inverse quadratic interpolation to maintain a high convergence rate.

References

[1] T. Björk. *Arbitrage Theory in Continuous Time.* Oxford University Press, USA, 3rd edition, 2009.

[2] A. Cairns, D. Blake, and K. Dowd. Stochastic lifestyling: Optimal dynamic asset allocation for defined contribution pension plans. *Journal of Economic Dynamics and Control*, 30(5):843–877, 2006.

[3] P. A. Forsyth and G. Labahn. Numerical methods for controlled Hamilton-Jacobi-Bellman PDEs in finance. *Journal of Computational Finance*, 11:1–43, 2007.

[4] G. Forsythe, M. Malcolm, and C. Moler. *Computer methods for mathematical computations.* Prentice-hall, 1977.

[5] Y. Huang, P. Forsyth, and G. Labahn. Combined fixed point and policy iteration for Hamilton-Jacobi-Bellman equations in finance. *To appear in SIAM Journal of Numerical Analysis.*

[6] Y. Huang, P. A. Forsyth, and G. Labahn. Methods for pricing American options under regime switching. *SIAM Journal on Scientific Computing*, 33(5):2144–2168, September 2011.

[7] H. J. Kushner and P. G. Dupuis. *Numerical Methods for Stochastic Control Problems in Continuous Time.* Springer-Verlag, New York, 1991.

[8] D. M. Pooley, P. A. Forsyth, and K. R. Vetzal. Numerical convergence properties of option pricing PDEs with uncertain volatility. *IMA Journal of Numerical Analysis*, 23:241–267, 2003.

[9] J. Wang and P. A. Forsyth. Maximal use of central differencing for Hamilton-Jacobi-Bellman PDEs in finance. *SIAM Journal of Numerical Analysis*, 46(3):1580–1601, 2008.