Multigrid Methods for Hamilton-Jacobi-Bellman and Hamilton-Jacobi-Bellman-Isaacs Equations

by

Dong Han

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2011

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

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

Abstract

We propose multigrid methods for solving Hamilton-Jacobi-Bellman (HJB) and Hamilton-Jacobi-Bellman-Isaacs (HJBI) equations. The methods are based on the full approximation scheme. We propose a damped-relaxation method as smoother for multigrid. In contrast with policy iteration, the relaxation scheme is convergent for both HJB and HJBI equations. We show by local Fourier analysis that the damped-relaxation smoother effectively reduces high frequency error. For problems where the control has jumps, restriction and interpolation methods are devised to capture the jump on the coarse grid as well as during coarse grid correction. We will demonstrate the effectiveness of the proposed multigrid methods for solving HJB and HJBI equations arising from option pricing as well as problems where policy iteration does not converge or converges slowly.

Acknowledgments

First and foremost, I must thank my supervisor, Justin Wan, for his invaluable tutoring, advice and perspective. He guided me through everything from research ideas to theoretic fundamentals and coding techniques. His interest and enthusiasm in my work made completing this thesis an interesting and rewarding process.

I would like to thank my readers, Peter Forsyth and Yuying Li. Their insight and thoughtfulness provided important contributions to the process and product. I would also like to thank all my colleagues in the Scicom Lab and all my friends for making my life in Waterloo a very enjoyable experience.

Finally I would like to thank my family for their encouragement and understanding.

Dedication

This is dedicated to my parents, Lianhui Han and Li Lv, for their love, care and support.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Many real life problems such as financial problems [33, 19] and stochastic games [27, 32] can be modeled as optimal control problems and formulated as nonlinear Hamilton-Jacobi-Bellman (HJB) and Hamilton-Jacobi-Bellman-Isaacs (HJBI) equations. HJB equation is a partial differential equation derived from the dynamic programming principle [7], see derivation and proofs in [22, 18, 9]. The corresponding discrete-time equation is usually referred to as the Bellman equation. In continuous time, it can be considered as an extension of Hamilton-Jacobi equation. HJBI equation is a variation of HJB equation with an additional control set. In many cases, these nonlinear PDEs do not have classical solutions. Existing methods for solving the HJB type problems includes Markov Chain [23, 14], PDE [19, 28, 34], binomial lattice [16] and simulation based methods [6]. These methods suffer from poor accuracy or timestep limitations due to stability considerations.

A more recent approach is based on numerical PDE methods. Since these PDEs are nonlinear, they might have more than one solution. Hence one must ensure that the numerical methods used will converge to the relevant solution, which, in this case, is the viscosity solution [15]. Unconditionally monotone implicit methods, which will ensure the convergence to the viscosity solution, are described in [5]. In these methods, a nonlinear set of discretized equation must be solved at each time step. The common practice in the PDE literature is to apply relaxation-type [5] and Newton-type [19] iterative methods in each time step for the solution of the nonlinear equations. However, these two methods

could be slow for large scale problems or even divergent for HJBI problems.

1.1 Motivation

This thesis is motivated by two finance problems arising from nonlinear asset allocation and option pricing problems in financial modeling. They lead to an HJB and an HJBI equation respectively. We note that there are other methods for solving the similar nonlinear asset allocation and option pricing problems. Since those methods are outside the scope of this thesis, we will refer the interested readers to [23, 36, 10].

1.1.1 HJB Case: Pension Plan Asset Allocation Problem

Suppose there are two assets in the market, one is risk free and the other is risky. The risky asset S follows the stochastic process

$$dS = (r + \xi\sigma) Sdt + \sigma SdZ,$$

where dZ is the increment of a Wiener process, σ is volatility, r is the interest rate, and ξ is the market price of risk. Investor pays into the pension plan at a constant rate π in the unit time. Let $W(t)$ denote the wealth in the pension plan at time t, a proportion q of this wealth is invested in the risky asset and the rest is invested in the risk free asset. Then

$$dW = [(r + q\xi\sigma) W + \pi] dt + q\sigma W dZ. \tag{1.1}$$

Let $W_T = W(T)$ where T is the expiration time of the pension plan. The aim of the investor is to maximize her expected terminal wealth on a given risk level, i.e. $\max_q \left\{E^{t=0}[W_T]\right\}$, such that $Var^{t=0}[W_T]$ =constant, where $E[\cdot]$ is the expectation operator and $Var[\cdot]$ is the variance operator. The superscript $t = 0$ indicates that the expectation and variance are computed at time $t = 0$. Using a Lagrange multiplier $\lambda > 0$, the problem is to determine the control q such that $E^{t=0}[W_T] - \lambda Var^{t=0}[W_T]$ is maximized, subject to (1.1).

For the convenience of computation, we will follow the common practice in the litera-

ture to introduce a parameter $\tau = T - t$. Let w be in a set of all admissible wealth $W(t)$ for $0 \leq t \leq T$. Following the steps in [33], we define an intermediate variable

$$V(w,\tau) = \inf_{q \in \hat{Q}} \left\{ E\left[(W_T - \frac{\gamma}{2})^2 | W(T-\tau) = w \right] \right\},$$

that has terminal condition

$$V(w,0) = \left(w - \frac{\gamma}{2} \right)^2,$$

where γ is a predetermined constant. The pension problem can be simplified to two steps [33]: first solve for $V(w,T)$ which satisfies an HJB equation

$$V_\tau = \inf_{q \in \hat{Q}} \left\{ \frac{1}{2} (q\sigma w)^2 V_{ww} + [\pi + w(r + q\sigma\xi) V_w] \right\}, \tag{1.2}$$

and then compute the optimal expected wealth by solving a Black-Scholes like equation. Equation (1.2) will be used as an example to illustrate our method for the HJB case.

1.1.2 HJBI Case: American Options and Stock Borrowing Fees

Consider the case where the cash borrowing rate and lending rate are not equal. Let the cash borrowing rate be r_b, the lending rate be r_l and $r_l > r_b$. Then the price of a long position option V is given by the nonlinear PDE

$$V_t + \frac{\sigma^2 S^2}{2} V_{SS} + \rho(SV_S - V)(SV_S - V) = 0, \tag{1.3}$$

where

$$\rho(x) = \begin{cases} 1 & if\, x \geq 0 \\ 0 & if\, x < 0 \end{cases}.$$

Notice that (1.3) can be rewritten as

$$V_t + \inf_{Q \in \hat{Q}} \left\{ \frac{\sigma^2 S^2}{2} V_{SS} + q_1 (SV_S - V) \right\} = 0,$$

where $Q = (q_1)$ and $\hat{Q} = (\{r_l, r_b\})$. This model can be extended to include stock borrowing

fees r_f, which are paid to stock lenders when a hedger shorts a stock. In this scenario, the holder of a short position will receive $r_l - r_f$, instead of r_l, on the proceeds of the short sale. Assuming the retail customers do not receive any interest on the proceeds of a short sale, i.e. $r_f = r_l$, the pricing equation for a long position is

$$\begin{aligned} V_t + \frac{\sigma^2 S^2}{2} V_{SS} + H\left(-V_S\right)\left[\rho\left(SV_S - V\right)\left(SV_S - V\right)\right] & \\ + H\left(V_S\right)\left[\left(r_l - r_f\right) SV_S - \rho\left(-V\right) V\right] &= 0, \end{aligned} \tag{1.4}$$

where

$$H\left(y\right) = \begin{cases} 1 & if\, y \geq 0 \\ 0 & if\, y < 0 \end{cases}.$$

And (1.4) can be rewritten as

$$V_t + \inf_{Q \in \hat{Q}} \left\{ \frac{\sigma^2 S^2}{2} V_{SS} + q_3 q_1 \left(SV_S - V\right) + \left(1 - q_3\right)\left[\left(r_l - r_f\right) SV_S - q_{2V}\right] \right\} = 0,$$

where $Q = \left(q_1, q_2, q_3\right)$ and $\hat{Q} = \left(\left\{r_l, r_b\right\}, \left\{r_l, r_b\right\}, \left\{0, 1\right\}\right)$.

For an American option with payoff V^*, its pricing equation is

$$\min\left(-V_t - \left\{\frac{\sigma^2 S^2}{2} V_{SS} + rSV_S - rV\right\}, V - V^*\right) = 0,$$

which can be written in penalty form as

$$V_t + \sup_{\mu \in \{0,1\}} \left\{ \frac{\sigma^2 S^2}{2} V_{SS} + rSV_S - rV + \mu \frac{V^* - V}{\eta} \right\} = 0,$$

where σ is volatility, r is the interest rate and $\eta \ll 1$ is a small positive number. Now consider the case when the stock borrowing model (1.4) is combined with American option early exercise. This gives rise to the equation

$$\begin{aligned} \min(-V_t - \inf_{Q \in \hat{Q}} \left\{ \tfrac{\sigma^2 S^2}{2} V_{SS} + q_3 q_1 \left(SV_S - V\right)\right. & \\ \left.\left(1 - q_3\right)\left[\left(r_l - r_f\right) SV_S - q_2 V\right]\right\}, V - V^*) &= 0, \end{aligned} \tag{1.5}$$

Rewriting it in the penalty form, we obtain

$$V_\tau = \sup_{\mu\in\hat{P}}\inf_{Q\in\hat{Q}} \left\{ \frac{\sigma^2 S^2}{2} V_{SS} + q_3 q_1 \left(SV_S - V\right) + \left(1-q_3\right)\left[\left(r_l - r_f\right) SV_S - q_2 V\right] + \mu \frac{V^* - V}{\eta} \right\}, \tag{1.6}$$

where $\hat{P} = \{0, 1\}$. Equation (1.6) is an HJBI equation and we will use it as an example to illustrate our method for the HJBI case.

We note that the selection of the penalty term η can affect the convergence of (1.6) and may render divergence when the theoretical conditions are satisfied [35]. Therefore solving (1.5) directly instead of (1.6) may yield better convergence.

1.1.3 Stochastic Games

There are also many problems in stochastic games that can be simplified to HJB or HJBI equations. Take a two-player zero-sum game for example. Consider a dynamic system, having a finite state space $S = \{1, 2, ..., N\}$, which is observed at times $t = 0, 1, ...$ The dynamic system is influenced by two players, P_1 and P_2, having opposite aims. For each $x \in S$, there exist one finite nonempty sets of actions for each player, denoted by K_x for P_1 and L_x for P_2. Assume the system is in state x at time t, then P_1 and P_2 will select an action from K_x and L_x respectively, moving the system to a new state y with probability $p\left(y|x,k,l\right)$, where $k \in K_x$, $l \in L_x$ and $\sum_{y\in S} p\left(y|x,k,l\right) = 1$. At the same time, P_1 will receive a possibly negative amount from P_2 denoted by $r\left(x,k,l\right)$. To solve for the total expected discounted reward for P_1 and P_2, we will need to solve an HJBI problem formulated in [32].

1.1.4 General Form of HJB and HJBI Equations

The example HJB problem (1.2) can be written in the general form as

$$V_\tau = \inf_{Q\in\hat{Q}} \left\{ a(S,\tau,Q)V_{SS} + b(S,\tau,Q)V_S - c(S,\tau,Q)V + d\left(S,\tau,Q\right) \right\}, \tag{1.7}$$

where

$$a\left(S,\tau,Q\right)=\frac{1}{2}q^{2}\sigma^{2}S^{2},\ b\left(S,\tau,Q\right)=\pi+S\left(r+q\sigma\xi\right),\ c\left(S,\tau,q\right)=0,\ d\left(S,\tau,q\right)=0.$$

For the case of HJBI problems, an additional set of controls $P\in\hat{P}$ is considered. The example HJBI problem can be written as

$$V_{\tau}=\sup_{P\in\hat{P}}\inf_{Q\in\hat{Q}}\left\{a(S,\tau,Q,P)V_{SS}+b(S,\tau,Q,P)V_{S}-c(S,\tau,Q,P)V+d\left(S,\tau,Q,P\right)\right\},\tag{1.8}$$

where

$$a\left(S,\tau,Q,P\right)=\frac{1}{2}\sigma^{2}S^{2},\ b\left(S,\tau,Q,P\right)=S\left(q_{3}q_{1}+\left(1-q_{3}\right)\left(r_{l}-r_{f}\right)\right),$$

$$c\left(S,\tau,Q,P\right)=q_{3}q_{1}+\left(1-q_{3}\right)q_{2}+\frac{\mu}{\eta},\ d\left(S,\tau,Q,P\right)=\frac{\mu}{\eta}V^{*}.$$

The initial and boundary conditions for both problems are described in details in [19, 33].

1.2 Discretization

Since solving the optimal control problem analytically is difficult for HJB and HJBI equations, we will consider the discrete optimal control problem and investigate numerical schemes instead. We will briefly discuss the discretization for the PDE in the general form of HJB and HJBI equations in this section. A positive coefficient discretization scheme, which will ensure the convergence to the viscosity solution, is applied for both HJB and HJBI case. It is shown that near quadratic convergence can be achieved as the grid size is reduced. The details for the positive coefficient discretization can be found in [19].

Define a grid $\{S_0, S_1, \ldots, S_M\}$ with $S_M = S_{max}$. Let V_i^n be a discrete approximation to $V\left(S_i,\tau^n\right)$ and $V^n=[V_0^n,\ldots,V_M^n]^T$. The objective function in (1.7) at $\left(S_i,\tau^{n+1}\right)$ is discretized using a combination of forward, backward or central differencing methods, giving

$$\begin{aligned}\left(a(S,\tau,Q)V_{SS}+b(S,\tau,Q)V_{S}-c(S,\tau,Q)V\right)_{i} &= \alpha_{i}^{n+1}\left(Q\right)V_{i-1}^{n+1}+\beta_{i}^{n+1}\left(Q\right)V_{i+1}^{n+1} \\ &- \left(\alpha_{i}^{n+1}\left(Q\right)+\beta_{i}^{n+1}\left(Q\right)+c_{i}^{n+1}\left(Q\right)\right)V_{i}^{n+1},\end{aligned}\tag{1.9}$$

where α_i and β_i are defined as

$$\alpha_{i,central}^{n+1} = \frac{2a_i^n}{(S_i - S_{i-1})(S_{i+1} - S_{i-1})} - \frac{b_i^n}{S_{i+1} - S_{i-1}},$$

$$\beta_{i,central}^{n+1} = \frac{2a_i^n}{(S_{i+1} - S_i)(S_{i+1} - S_{i-1})} + \frac{b_i^n}{S_{i+1} - S_{i-1}},$$

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

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

such that a positive coefficient scheme is resulted. Since forward and backward differencing guarantee a positive coefficient method but with only first order accuracy, central differencing is used as much as possible. We will consider fully implicit timestepping and (1.9) to discretize (1.7) and obtain

$$\frac{V_i^{n+1} - V_i^n}{\Delta\tau} = \inf_{Q \in \hat{Q}} \left\{ \left[A(Q) V^{n+1}\right]_i + \left[D(Q)\right]_i^{n+1} \right\}, \quad i < M \tag{1.10}$$

where $\left[A(Q) V^{n+1}\right]_i$ is the matrix form of the operator defined in (1.9) and $\left[D(Q)\right]_i^{n+1}$ is the vector form of $d_i^{n+1}(Q)$. The first and last row of matrix A and vector D are modified accordingly to handle the boundary conditions. Also we note that higher order timestepping such as Crank-Nicolson timestepping can be used [19, 33]. The HJBI equation can be discretized in a similar way as

$$\frac{V_i^{n+1} - V_i^n}{\Delta\tau} = \sup_{P \in \hat{P}} \inf_{Q \in \hat{Q}} \left\{ \left[A(Q,P) V^{n+1}\right]_i + \left[D(Q,P)\right]_i^{n+1} \right\}. \tag{1.11}$$

Since the grid may not be uniform and it is possible that forward or backward differencing is used at some grid points, the discretization method is formally first order accurate in $\max_i (S_{i+1} - S_i)$. However, in practice, forward and backward differencing are only required at a small number of grid points and grid size is usually changed smoothly.

Algorithm 1.1 Policy Iteration for HJB Equations

1: Let $\hat{V}^0 \equiv (V^{n+1})^0 = V^n$
2: **for** $k = 0, 1, 2...$ until convergence **do**
3: Compute $Q^k \in \arg\inf_{Q \in \hat{Q}} \left\{ A^{n+1}(Q)\, \hat{V}^k + D^{n+1}(Q) \right\}$
4: Solve $\left[I - \Delta\tau A^{n+1}(Q^k) \right] \hat{V}^{k+1} = V^n + \Delta\tau D^{n+1}(Q^k)$

1.3 Numerical Solution of Discretized Equations

The solution of the discretized equation (1.10) and (1.11) are not obvious. There are two commonly used numerical algorithms for these problems: policy iteration and relaxation scheme.

1.3.1 Policy Iteration

The policy iteration, also referred as Howard's algorithm [24, 21, 8] is a Newton-like method [26] for nonlinear problems. It is common in literature to apply policy iteration to solve HJB equations [21, 20]. Policy iteration consists of an iterative algorithm on the control and the value functions and generates an improving sequence of controls to the nonlinear problem. It first computes for the optimal control based on an approximate solution, linearizes the problem with the resulting control, and then solves the linear system to obtain a better approximate solution. More specifically, let $\hat{V}^k$ be an approximate solution. The idea of policy iteration is to compute the optimal control Q^k from $\hat{V}^k$. Then an improved approximation $\hat{V}^{k+1}$ is obtained from Q^k. The procedure is repeated until convergence. The policy iteration scheme for HJB equation at time step $n+1$ is described in Algorithm 1.1.

Policy iteration is globally convergent for HJB equations. However, in general, the number of policy iterations for a discrete HJB problem cannot be bounded by a constant that is independent of the number of the grid points [27]. Moreover, the extension of policy iteration for HJBI equations is unclear. One straightforward Newton-like extension is shown in Algorithm 1.2.

However, the sup-inf operator is neither convex nor concave, this extension of policy

Algorithm 1.2 Newton-like Policy Iteration for HJBI Equations

Let $\hat{V}^0 \equiv (V^{n+1})^0 = V^n$
for $k = 0, 1, 2...$ until convergence **do**
  Compute $P^k, Q^k \in \arg\sup_{P\in\hat{P}}\inf_{Q\in\hat{Q}} \left\{ A^{n+1}(Q,P)\,\hat{V}^k + D^{n+1}(Q,P) \right\}$
  Solve $\left[I - \Delta\tau A^{n+1}(Q^k, P^k)\right]\hat{V}^{k+1} = V^n + \Delta\tau D^{n+1}(Q^k, P^k)$

Algorithm 1.3 Another Extension of Policy Iteration for HJBI Equations

Let $\hat{V}^0 \equiv (V^{n+1})^0 = V^n$
Let $P^0 \in \hat{P}$
for $j = 0, 1, 2...$ until HJBI converges **do**
  $\hat{U}^0 \leftarrow \hat{V}^j$
  for $k = 0, 1, 2...$ until HJB converges **do**
    Compute $Q^k \in \arg\inf_{Q\in\hat{Q}} \left\{ A^{n+1}\left(Q, P^j\right)\hat{V}^k + D^{n+1}\left(Q, P^j\right) \right\}$
    Solve $\left[I - \Delta\tau A^{n+1}(Q^k, P^j)\right]\hat{V}^{k+1} = V^n + \Delta\tau D^{n+1}(Q^k, P^j)$
  $\hat{V}^{j+1} \leftarrow \hat{U}^{k+1}$
  Compute $P^{j+1} \in \arg\sup_{P\in\hat{P}}\inf_{Q\in\hat{Q}} \left\{ A^{n+1}(Q,P)\,\hat{V}^{k+1} + D^{n+1}(Q,P) \right\}$

iteration for HJBI equations does not guarantee global convergence [32, 11]. Another extension of policy iteration named Ho-4 is proposed in [11]. It is not a Newton-like method. Unlike the algorithm above, Ho-4 separates the two sets of controls. It first fixes control $P \in \hat{P}$ to reduce the HJBI problem to an HJB equation, solves the HJB equation using the standard policy iteration to obtain an improved approximate solution, and then use this solution to update P. The algorithm can be implemented by nested loops described in Algorithm 1.3.

Ho-4 is globally convergent under certain assumptions. However the nested policy iterations increase the computation complexity. Other variations of policy iterations are also investigated in [29, 13, 30].

1.3.2 Relaxation Scheme

Another method for solving HJB and HJBI equations is a relaxation scheme [5], also known as the value iteration method [23]. As one of the general methods for solving dynamic programs, relaxation scheme solve for the optimal control by computing the

optimal value function. Consider the HJB equation, the discrete equation (1.10) can be written as

$$\begin{aligned} V_i^{n+1} &= \Delta\tau \inf_{Q\in\hat{Q}} \left\{ \alpha_i^{n+1}(Q) V_{i-1}^{n+1} + \beta_i^{n+1}(Q) V_{i+1}^{n+1} \right. \\ &\quad \left. - \left(\alpha_i^{n+1}(Q) + \beta_i^{n+1}(Q) + c_i^{n+1}(Q)\right) V_i^{n+1} + d_i^{n+1}(Q) \right\} + V_i^n. \end{aligned} \tag{1.12}$$

Since V_i^{n+1}does not depend on the control Q , V_i^n and $\Delta\tau$ are constants. Rearranging (1.12), we obtain

$$\begin{aligned} 0 &= \inf_{Q\in\hat{Q}} \left\{ \Delta\tau \left(\alpha_i^{n+1}(Q) V_{i-1}^{n+1} + \beta_i^{n+1}(Q) V_{i+1}^{n+1} + d_i^{n+1}(Q) \right) + V_i^n \right. \\ &\quad \left. - \left[1 + \Delta\tau \left(\alpha_i^{n+1}(Q) + \beta_i^{n+1}(Q) + c_i^{n+1}(Q) \right) \right] V_i^{n+1} \right\}, \end{aligned}$$

which can be written as

$$\begin{aligned} 0 &= \inf_{Q\in\hat{Q}} \left\{ \left[1 + \Delta\tau \left(\alpha_i^{n+1}(Q) + \beta_i^{n+1}(Q) + c_i^{n+1}(Q) \right) \right] \cdot \left[-V_i^{n+1} + \right. \right. \\ &\quad \left. \left. \frac{\Delta\tau \left(\alpha_i^{n+1}(Q) V_{i-1}^{n+1} + \beta_i^{n+1}(Q) V_{i+1}^{n+1} + d_i^{n+1}(Q) \right) + V_i^n}{1 + \Delta\tau \left(\alpha_i^{n+1}(Q) + \beta_i^{n+1}(Q) + c_i^{n+1}(Q) \right)} \right] \right\}. \end{aligned} \tag{1.13}$$

Note that α_i^{n+1}, β_i^{n+1} and c_i^{n+1} are all non-negative. Then from (1.13) we obtain

$$V_i^{n+1} = \inf_{Q\in\hat{Q}} \left\{ \frac{\Delta\tau \left(\alpha_i^{n+1}(Q) V_{i-1}^{n+1} + \beta_i^{n+1}(Q) V_{i+1}^{n+1} + d_i^{n+1}(Q) \right) + V_i^n}{1 + \Delta\tau \left(\alpha_i^{n+1}(Q) + \beta_i^{n+1}(Q) + c_i^{n+1}(Q) \right)} \right\}. \tag{1.14}$$

Let $\hat{V}^k$ be the k^{th} estimate for V^{n+1}, a relaxation scheme can be derived from (1.14)

$$\hat{V}_i^{k+1} = \inf_{Q\in\hat{Q}} \left\{ \frac{\Delta\tau \left(\alpha_i^{n+1}(Q) \hat{V}_{i-1}^k + \beta_i^{n+1}(Q) \hat{V}_{i+1}^k + d_i^{n+1}(Q) \right) + V_i^n}{1 + \Delta\tau \left(\alpha_i^{n+1}(Q) + \beta_i^{n+1}(Q) + c_i^{n+1}(Q) \right)} \right\}. \tag{1.15}$$

Similarly, the relaxation scheme for HJBI problems is

$$\hat{V}_i^{k+1} = \sup_{P\in\hat{P}} \inf_{Q\in\hat{Q}} \left\{ \frac{\Delta\tau \left(\alpha_i^{n+1}(Q,P) \hat{V}_{i-1}^k + \beta_i^{n+1}(Q,P) \hat{V}_{i+1}^k + d_i^{n+1}(Q,P) \right) + V_i^n}{1 + \Delta\tau \left(\alpha_i^{n+1}(Q,P) + \beta_i^{n+1}(Q,P) + c_i^{n+1}(Q,P) \right)} \right\}. \tag{1.16}$$

Theorem 1. *Suppose that the discretization (1.10) satisfies the positive coefficient condition [19]. Then a unique solution of the nonlinear equation (1.12) exists, and the iteration scheme (1.15) is globally convergent for any initial estimate. Furthermore,*

$$\left\|\hat{V}^{k+1}-\hat{V}^{k}\right\|_{\infty} \leq \gamma \left\|\hat{V}^{k}-\hat{V}^{k-1}\right\|_{\infty}$$

where

$$\gamma = \max_{i} \sup_{Q\in\hat{Q}} \left\{ \frac{\alpha_i^{n+1}(Q)+\beta_i^{n+1}(Q)}{1+\Delta\tau\left[\alpha_i^{n+1}(Q)+\beta_i^{n+1}(Q)+c_i^{n+1}(Q)\right]} \right\}. \tag{1.17}$$

Since $\alpha_i^{n+1}(Q)$, $\beta_i^{n+1}(Q)$ and $c_i^{n+1}(Q)$ are non-negative for all $Q \in \hat{Q}$, we have $\gamma < 1$. Hence the scheme (1.15) converges to the unique solution of the discretized equation. The above argument still hold for HJBI case (1.16).

The convergence of the relaxation scheme, however, can be very slow for both HJB and HJBI equations. Suppose Q is constant, then it can be shown that $\gamma \simeq \frac{1}{1+O(h)}$ where h is the grid size [19]. When the grid size is small, γ is close to 1, rendering a slow convergence on the fine grid.

1.4 Contributions and the Chapter Plan

As shown in Section 1.3, the commonly used approaches are complicated or expensive. On the other hand, multigrid methods are considered as efficient numerical methods for solving a wide variety of PDEs [31]. The rate of convergence is often independent of the mesh size. In this thesis, convergent and efficient numerical methods for HJB and HJBI equations using multigrid methods are designed. We propose a multigrid method based on Full Approximation Scheme (FAS) and a relaxation scheme smoother for both HJB and HJBI equations. We also design restriction and interpolation methods for HJB and HJBI equations where the control can change dramatically between neighbouring grid points. Optimal control near the "jump" is handled explicitly to achieve better convergence.

In Chapter 2, we will give an introduction to multigrid methods. Then we will discuss different multigrid methods for HJB and HJBI equations based on policy iteration, relaxation scheme and FAS in Chapter 3. We will demonstrate our algorithms by mainly focusing on two example finance problems. A smoothing analysis will be presented in Chapter 4, followed by numerical results on a variety of examples illustrating the convergence of different methods in Chapter 5.

Chapter 2

Multigrid Methods

Multigrid methods are efficient algorithms that solves differential equations using a hierarchy of discretizations. Multigrid methods are motivated by the properties of stationary iterative methods, which have slow convergence for a large number of linear systems. It is typical that applying the stationary iterations to the elliptic operators will reduce the high frequency error quite quickly, while slow reduction in low frequency error is observed. The idea of multigrid methods is to accelerate the convergence of a relaxation algorithm by removing the low frequency error efficiently. Since the low frequency error is smooth, representing it on a coarser grid will keep most of the information. Also the cost of any global computation is at least proportional to the grid size. Resolving the error on a coarser grid will be effective for smooth error reduction and be less expensive. Multigrid methods use stationary iterations as smoothers to remove the high frequencies and removes the low frequency error by coarse grid correction. In this chapter, we will discuss how these two parts are combined by first focusing on smoothing properties of some standard stationary iterative methods and then discuss the two-grid iteration, multigrid iteration, full multigrid method and full approximation scheme.

2.1 Smoothing

In this section, we examine two stationary iterative methods: Jacobi iteration and Gauss-Seidel iteration. Although these two iterative methods may be efficient for solving

small linear systems, their convergence rates can be unacceptably slow for large scale problems. However, we will see that these two methods are suitable for the smoothing procedures.

2.1.1 Jacobi Iteration

Consider a discrete Poisson equation with Dirichlet boundary conditions

$$\begin{aligned} -\triangle_h u_h(x,y) &= f_h^{\Omega}(x,y), \quad ((x,y) \in \Omega_h), \\ u_h(x,y) &= f_h^{\Gamma}(x,y), \quad ((x,y) \in \Gamma_h = \partial\Omega_h), \end{aligned}$$

in the unit square $\Omega = (0,1)^2 \subset \mathbb{R}^2$ with grid size $h = \frac{1}{n}$, $n \in \mathbb{N}$. Assume $\bar{u}_h(x,y)$ is the exact solution and $\hat{u}_h^k(x,y)$ is the approximation solution after the k^{th} iteration. The iteration formula of the Jacobi iteration is

$$\begin{aligned} z_h^{k+1}(x_i,y_j) &= \frac{1}{4}\Big[h^2 f_h(x_i,y_j) + \hat{u}_h^k(x_i-h,y_j) + \hat{u}_h^k(x_i+h,y_j) \\ &\quad + \hat{u}_h^k(x_i,y_j-h) + \hat{u}_h^k(x_i,y_j+h)\Big], \\ \hat{u}_h^{k+1} &= z_h^{k+1}, \end{aligned}$$

with $(x_i,y_j) \in \Omega_h$. It can be generalized by introducing a damping factor ω and become

$$\hat{u}_h^{k+1} = \hat{u}_h^k + \omega\left(z_h^{k+1} - u_h^{k+1}\right),$$

which is called the damped-Jacobi method. It is obvious that when $\omega = 1$, the damped-Jacobi method becomes Jacobi iteration. The damped-Jacobi iteration can be written as

$$\hat{u}_h^{k+1} = S_h \hat{u}_h^k + \frac{h^2}{4} f_h,$$

where

$$S_h = \frac{\omega}{4}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 4\left(\frac{1}{\omega}-1\right) & 1 \\ 0 & 1 & 0 \end{bmatrix}_h.$$

The eigenfunctions of S_h is

$$\varphi_h^{l_1,l_2}(x,y) = \sin l_1\pi x \sin l_2\pi y, \quad ((x,y)\in\Omega_h;\ (l_1,l_2=1,...,n-1)),$$

with the corresponding eigenvalues

$$\chi_h^{l_1,l_2}(\omega) = 1 - \frac{\omega}{2}(2-\cos l_1\pi h - \cos l_2\pi h). \tag{2.1}$$

Let $\epsilon_h^k = \hat{u}_h^k - \bar{u}_h$ be the error after the k^{th} damped-Jacobi iteration. Expanding ϵ_h^k and ϵ_h^{k+1} into discrete eigenfunction series, we obtain

$$\epsilon_h^k = \sum_{l_1,l_2=1}^{n-1} \alpha_{l_1,l_2}\varphi_h^{l_1,l_2},$$

and

$$\epsilon_h^{k+1} = \sum_{l_1,l_2=1}^{n-1} \chi_h^{l_1,l_2}\alpha_{l_1,l_2}\varphi_h^{l_1,l_2}.$$

Thus the convergence properties is characterized by the spectral radius

$$\rho(S_h) = \max\left\{\left|\chi_h^{l_1,l_2}\right| :\ (l_1,l_2=1,...,n-1)\right\},$$

which is the asymptotic convergence factor of the iteration. Thus, for Jacobi method, we obtain

$$\rho(S_h) = \left|\chi_h^{1,1}\right| = |1-\omega(1-\cos\pi h)| = 1 - O\left(\omega h^2\right) < 1,$$

for $0<\omega\leq 1$ and $\rho(S_h)\geq 1$ otherwise. This shows that the value of ω has to lie in $(0,1]$ to reach convergence, and the convergence rate could be close to 1 as the grid size decreases.

On the other hand, to analyze the smoothing properties of Jacobi method, we will only focus on the high frequency error, i.e. $\frac{n}{2}\leq \max(l_1,l_2)\leq n-1$. Now we define the smoothing factor of $S_h(\omega)$ as

$$\mu(h;\omega) = \max\left\{\left|\chi_h^{l_1,l_2}(\omega)\right| :\ \frac{n}{2}\leq \max(l_1,l_2)\leq n-1\right\}, \tag{2.2}$$

which represent the worst case high frequency error reduction rate. Substituting (2.1) into (2.2), we obtain the smoothing factor for damped-Jacobi method

$$\mu(h;\omega) = \max\left\{\left|1 - \frac{\omega}{2}(2 - \cos l_1\pi h - \cos l_2\pi h)\right| : \frac{n}{2} \le \max(l_1, l_2) \le n-1\right\}.$$

It is easy to show that $\omega = \frac{4}{5}$ will result the minimal smoothing factor which is bounded above by $\frac{3}{5}$. This means that one iteration of damped-Jacobi with $\omega = \frac{4}{5}$ will reduce the high frequency error by at least a factor of $\frac{3}{5}$. This verifies that Jacobi iteration is efficient as a smoother.

2.1.2 Gauss-Seidel Iteration

Gauss-Seidel Iteration is another commonly used smoother in multigrid methods.The iteration formula of Gauss-Seidel with damping factor ω is

$$\begin{aligned} z_h^{k+1}(x_i, y_j) &= \frac{1}{4}\Big[h^2 f_h(x_i, y_j) + \hat{u}_h^{k+1}(x_i - h, y_j) + \hat{u}_h^k(x_i + h, y_j) \\ &\quad + \hat{u}_h^{k+1}(x_i, y_j - h) + \hat{u}_h^k(x_i, y_j + h)\Big], \\ \hat{u}_h^{k+1} &= \hat{u}_h^k + \omega\left(z_h^{k+1} - u_h^{k+1}\right). \end{aligned}$$

The smoothing analysis of Gauss-Seidel iteration requires a different tool, which will be introduced in Chapter 4. For the same discrete Poisson equation in the previous section, the smoothing factor of Gauss-Seidel is bounded above by 0.5 for $\omega = 1$.

2.2 Coarse Grid Correction

Consider a linear system $A_h\bar{u}_h = f_h$ defined on domain Ω_h with grid size h. Let the exact solution of the system be

$$\bar{u}_h = \hat{u}_h^k + \epsilon_h^k,$$

where $\hat{u}_h^k$ is the approximate solution after the k^{th} iteration with error ϵ_h. The residual of $\hat{u}_h^k$ is

$$r_h^k \equiv f_h - A_h\hat{u}_h^k = A_h\left(\bar{u}_h - \hat{u}_h^k\right) = A_h\epsilon_h^k,$$

Algorithm 2.1 Coarse Grid Correction I

1: Obtain an approximate solution $\hat{u}_h^k$
2: Compute residual $r_h^k = f_h - A_h\hat{u}_h^k$
3: Compute ϵ_h^k by solving $A_h\epsilon_h^k = r_h^k$
4: Obtain the exact solution $\bar{u}_h = \hat{u}_h^k + \epsilon_h^k$

Algorithm 2.2 Coarse Grid Correction II

1: Obtain an approximate solution $\hat{u}_h^k$
2: Compute residual $r_h^k = f_h - A_h\hat{u}_h^k$
3: Solve $A_h\epsilon_h^k = r_h^k$ approximately and get $\hat{\epsilon}_h^k$
4: Obtain an improved approximate solution $\hat{u}_h^{k+1} = \hat{u}_h^k + \hat{\epsilon}_h^k$

and ϵ_h^k can be obtained by solving

$$A_h\epsilon_h^k = r_h^k. \tag{2.3}$$

Hence we can obtain $\bar{u}_h$ by following the trivial steps in Algorithm 2.1.

However, if we solve (2.3) approximately in step 2, which is cheaper, we will obtain an improved approximate solution instead of the exact solution. Repeating these steps results an iterative method for solving the linear system with low cost per iteration. The idea of coarse grid correction is to approximately solve (2.3) by solving

$$A_H\hat{\epsilon}_H^k = r_H^k,$$

where A_H is an appropriate approximation of A_h on a coarser grid Ω_H with grid size H. Assume two intergrid transfer operators restriction R and interpolation operator P

$$R : \mathcal{G}(\Omega_h) \to \mathcal{G}(\Omega_H), \quad P : \mathcal{G}(\Omega_H) \to \mathcal{G}(\Omega_h)$$

to be given. r_h^k is restricted to Ω_H by

$$r_H^k = R \cdot r_h^k,$$

and correction $\hat{\epsilon}_H^k$ is interpolated to Ω_h by

$$\hat{\epsilon}_h^k = P \cdot \hat{\epsilon}_H^k.$$

Algorithm 2.3 Coarse Grid Correction III

1: Obtain an approximate solution $\hat{u}_h^k$
2: Compute residual $r_h^k = f_h - A_h\hat{u}_h^k$
3: Restrict the residual $r_H^k = R \cdot r_h^k$
4: Solve $A_H\hat{\epsilon}_H^k = r_H^k$ on Ω_H
5: Interpolate the correction $\hat{\epsilon}_h^k = P \cdot \hat{\epsilon}_H^k$
6: Update the approximation $\hat{u}_h^{k+1} = \hat{u}_h^k + \hat{\epsilon}_h^k$

Algorithm 2.4 $\hat{u}_h^{k+1}$= two-grid($\hat{u}_h^k$,A_h,f_h)

1: **(1) Presmoothing**
2: Compute $\tilde{u}_h^k$ by applying ν_1 iterations of smoothing procedure to $\hat{u}_h^k$:
$\tilde{u}_h^k = smoothing(\hat{u}_h^k, A_h, f_h)$
3: **(2) Coarse Grid Correction (CGC)**
4: Compute the residual: $\tilde{r}_h^k = f_h - A_h\tilde{u}_h^k$
5: Restrict the residual: $\tilde{r}_H^k = R \cdot \tilde{r}_h^k$
6: Solve on coarse grid (Ω_H): $A_H\tilde{\epsilon}_H^k = \tilde{r}_H^k$
7: Interpolate the correction: $\tilde{\epsilon}_h^k = P \cdot \tilde{\epsilon}_H^k$
8: Correct the approximation: $\hat{u}_h^{k,CGC} = \tilde{u}_h^k + \tilde{\epsilon}_h^k$
9: **(3) Postsmoothing**
10: Compute $\tilde{u}_h^{k+1}$ by applying ν_2 iterations of smoothing procedure to $\hat{u}_h^{k,CGC}$:
$\hat{u}_h^{k+1} = smoothing(\hat{u}_h^{k,CGC}, A_h, f_h)$

Thus the coarse grid correction procedure can be written as Algorithm 2.3.

However, the coarse grid correction itself is not convergent due to the high frequency error [31].

2.3 Two-Grid Cycle

The previous two sections show that the smoothing processes remove high frequency error effectively but not the low frequency error, while the coarse grid correction is opposite. Therefore it is natural to combine the two processes to achieve better convergence. Each iteration of a two-grid algorithm consists of three parts: presmoothing, coarse grid correction and postsmoothing, which can be described as Algorithm 2.4.

In this two-grid algorithm, there are some components that haven't been specified, including: the smoothing procedure, the number of smoothing iterations ($\nu_1, \nu_2 > 0$), the coarse grid Ω_H, the restriction (R) and interpolation (P) operators and the coarse grid operator A_H. The choice of these components may have strong effects on the convergence

of the algorithm.

2.4 Multigrid Components

In this section, we will introduce some of the commonly used components of the multigrid methods.

2.4.1 Coarse Grids

Standard coarsening is the simplest and the most commonly used coarse grid Ω_H. It doubles the grid size h in every direction. All the results in this thesis are based on this choice of coarse grid. There are also semicoarsening, which doubles grid size in one direction only for 2D problems and different variants for 3D problems. More about other coarsening such as 4h-coarsening and red-black coarsening can be found in [31].

2.4.2 Coarse Grid Operator

One common coarse grid operator A_H is to use the direct discretization of the operator on coarse grid Ω_H. A different choice called Galerkin coarse grid operator is defined by

$$A_H = R \cdot A_h \cdot P,$$

where R and P are appropriate intergrid transfer operators. We will refer interested readers to [31] for more information on the choice of coarse grid operators.

2.4.3 Restriction and Interpolation Operators

The choice of restriction and interpolation operators should be related to the choice of coarse grid. We will focus on transfer operators for standard coarsening, i.e. $H = 2h$. A restriction operator maps functions on Ω_h to Ω_H. One possibility for a restriction operator is the "injection" operator, in which for a grid function $r_h(x, y)$, its corresponding $r_H(x, y)$

will have

$$r_H(x,y) = R^{injection} \cdot r_h(x,y) = r_h(x,y) \ \ for \ (x,y) \in \Omega_H \subset \Omega_h.$$

Full weighting operator is commonly used and it is applied to all the results in this thesis. Applying the full weighting operator at a coarse grid point $(x,y) \in \Omega_H$ will result in a nine-point weighted average of r_h

$$\begin{aligned} r_H(x,y) &= \frac{1}{16}\left[4r_h(x,y) + 2r_h(x+h,y) + 2r_h(x-h,y) + r_h(x,y+h)\right. \\ &\quad +2r_h(x,y-h) + r_h(x+h,y+h) + r_h(x+h,y-h) \\ &\quad \left.+r_h(x-h,y+h) + r_h(x-h,y-h)\right], \end{aligned}$$

and the operator can also be written in stencil notation as

$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}_h^H .$$

Another possibility is half weighting operator:

$$\frac{1}{8}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix}_h^H .$$

An interpolation operator maps function on Ω_H to Ω_h. One frequently used interpolation is bilinear interpolation, which can be written in stencil notation as

$$\frac{1}{4}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}_H^h .$$

All the results in this thesis is based on bilinear interpolation.

Algorithm 2.5 Multigrid Cycle (Partial)

1: **if** Ω_H is the coarsest grid **then**
2: Solve $A_H \tilde{\epsilon}_H^k = \tilde{r}_H^k$
3: **else**
4: Solve $A_H \tilde{\epsilon}_H^k = \tilde{r}_H^k$ approximately by applying λ times of multigrid cycles using 0 as the initial approximation on the coarser grid

2.5 Multigrid Cycle

The two-grid cycle is not very practical since the coarse grid problem is still complicated. It can be shown that it is not necessary to solve (2.3) exactly on the coarse grid. Therefore, to reduce the computational complexity, we can apply the same idea of two-grid cycle again and solve (2.3) on a coarser grid than Ω_H. Sometimes, the two-grid cycle is applied more than once, say λ times. In fact, it is quite common to consider cases of $\lambda = 1$ and $\lambda = 2$. When $\lambda = 1$, the cycle traverse downward to the coarsest grid and then return directly upward to the finest grid and is referred as V-cycle. When $\lambda = 2$, it is called W-cycle. The cycle takes the form of a downward traverse, followed by a single step up, then down, then up two levels, and so on.

The algorithm of multigrid cycle is only different from two-cycle at one place: Solve on coarse grid (Ω_H): $A_H \tilde{\epsilon}_H^k = \tilde{r}_H^k$ in step 2. If we replace this operation by Algorithm 2.5, we can obtain a multigrid cycle.

2.6 Full Multigrid

The idea of full multigrid is to obtain a better initial approximation by nested iteration. It first solves the equation on the coarsest discretization, and then interpolate the results to the next finer grid. On this finer level, one or more multigrid cycles are performed based on the interpolated results. Then the results are again interpolated to the next finer grid and repeat the whole process until the finest grid is reached. Full multigrid often results a good initial approximation of the solution and therefore a faster convergence.

2.7 Full Approximation Scheme (FAS)

Full Approximation Scheme (FAS) is a nonlinear multigrid algorithm proposed by Brandt. It solves the complete original problem on every level of the grid throughout the multigrid cycle instead of the residual equation (2.3). Consider a nonlinear problem $N_h(\bar{u}_h) = f_h$ on Ω_h with the exact solution $\bar{u}_h$ and an approximate solution $\hat{u}_h^k$. Let the error be $\hat{\epsilon}_h^k = \bar{u}_h - \hat{u}_h^k$. As the operator $N_h(\cdot)$ is nonlinear, we cannot assume that $N_h(\bar{u}_h) - N_h\left(\hat{u}_h^k\right) = N_h\left(\hat{\epsilon}_h^k\right)$ and therefore it is not obvious how to compute $\hat{\epsilon}_h^k$. However, taking the usual form of the residual $\hat{r}_h^k = f_h - N_h\left(\hat{u}_h^k\right)$ and substituting into the original system gives

$$N_h(\bar{u}_h) = \hat{r}_h^k + N_h\left(\hat{u}_h^k\right). \tag{2.4}$$

FAS solves for $\hat{u}_H$ from

$$N_H(\hat{u}_H) = R \cdot \hat{r}_h + N_H(R \cdot \hat{u}_h),$$

which is effectively a modified version of the original system, instead of the residual equation on the coarse grid. It produces an improved approximation to the solution, instead of a correction directly. Note that when applied to a linear system, the FAS scheme is equivalent to the linear multigrid method in the previous section. A multigrid FAS algorithm is described in Algorithm 2.6.

We can observe that nonlinear smoothing processes is needed while the intergrid transfer can be linear. No global linearization is required except for the coarsest grid.

Algorithm 2.6 $\hat{u}_h^{k+1}$= FAS($\hat{u}_h^k$,A_h,f_h)

(1) Presmoothing

Compute $\tilde{u}_h^k$ by applying ν_1 iterations of smoothing procedure to $\hat{u}_h^k$:

$\tilde{u}_h^k = smoothing(\hat{u}_h^k, N_h, f_h)$

(2) Coarse Grid Correction (CGC)

Compute the residual: $\tilde{r}_h^k = f_h - N_h\left(\tilde{u}_h^k\right)$

Restrict the residual: $\tilde{r}_H^k = R \cdot \tilde{r}_h^k$

Restrict $\tilde{u}_h^k$: $\tilde{u}_H^k = R \cdot \tilde{u}_h^k$

Compute the right-hand-side: $f_H = \tilde{r}_H^k + N_H\left(\tilde{u}_H^k\right)$

if Ω_H is the coarsest grid **then**

Solve $N_H\left(\breve{u}_H^k\right) = f_H$ for $\breve{u}_H^k$

else

Solve $N_H\left(\breve{u}_H^k\right) = f_H$ approximately by applying λ times of FAS cycles using $\tilde{u}_H^k$ as the initial approximation on the coarser grid

Compute correction: $\tilde{\epsilon}_H^k = \breve{u}_H^k - \tilde{u}_H^k$

Interpolate the correction: $\tilde{\epsilon}_h^k = P \cdot \tilde{\epsilon}_H^k$

Correct the approximation: $\hat{u}_h^{k,CGC} = \tilde{u}_h^k + \tilde{\epsilon}_h^k$

(3) Postsmoothing

Compute $\tilde{u}_h^{k+1}$ by applying ν_2 iterations of smoothing procedure to $\hat{u}_h^{k,CGC}$:

$\hat{u}_h^{k+1} = smoothing(\hat{u}_h^{k,CGC}, N_h, f_h)$

Chapter 3

Multigrid method for HJB and HJBI Equations

Literature on multigrid methods for HJB and HJBI equations is scarce. In [2, 1, 4, 3], portfolio selection problems are modeled as HJB equations and solved by the multigrid-Howard and the full multigrid-Howard (FMGH) algorithm. In each policy iteration, linear multigrid method or full multigrid method is applied to solve the linearized problem. Hoppe [20] proposed two multigrid schemes for HJB equations, MGS I and MGS II, in which multigrid methods are applied directly to the nonlinear HJB equation. MGS I is based on an iterative numerical scheme which requires the solution of an unilateral variational inequality in each iteration [24] and MGS II is based on policy iteration. MGS's are similar to but different from a Full Approximation Scheme (FAS) [12] and the main difference lies in the coarse grid problem construction. Bloss and Hoppe [25] later proposed another multigrid method, MGHJB, for HJB equations. MGHJB is an updated version of MGS II: MGHJB applies nonlinear Gauss-Seidel iteration as the smoother while MGS II applies linear relaxation to linearized discrete HJB equation. Multigrid methods based on a variant of policy iteration were also applied to HJBI equations [17], in which multigrid is used to solve the linearized HJBI problem.

Algorithm 3.1 Policy Iteration with Multigrid

1: Let $\hat{V}^0 \equiv (V^{n+1})^0 = V^n$
2: **for** $k = 0, 1, 2...$ until convergence **do**
3: Compute $Q_i^k \in \arg\inf_{Q \in \hat{Q}} \left\{ \left[A^{n+1}(Q) \hat{V}^k + D^{n+1}(Q) \right]_i \right\}$
4: $M\left(Q^k\right) \leftarrow I - \Delta\tau A^{n+1}(Q^k)$
5: $f\left(Q^k\right) \leftarrow V^n + \Delta\tau D^{n+1}(Q^k)$
6: Solve $\hat{V}^{k+1} = multigrid\left(\hat{V}^k, M\left(Q^k\right), f\left(Q^k\right)\right)$

3.1 Policy Iteration with Multigrid

It is common in literature to solve the HJB equations with policy iteration. However, in each policy iteration, it is necessary to solve a linear system with M equations and M unknowns. This particular step becomes expensive as the grid size is reduced. Using multigrid method to solve this linear system is proposed by [2], where standard V-cycle full multigrid method, Gauss-Siedel smoother and linear intergrid transfer are applied. Policy iteration with multigrid is easy to implement and its convergence rate is rapid for some problems. The procedures are shown in Algorithm 3.1.

In other words, in each iteration k, an optimal control vector Q^k is computed. Then, a linearized system M is constructed based on Q^k. The linear system is then solved by a standard multigrid method.

3.2 FAS for HJB Equations

In this thesis, we propose to solve the nonlinear HJB and HJBI problems with FAS. For the convenience of applying FAS, the HJB problem is redefined as

$$N_h^Q\left(V^{n+1}\right) = B_h, \tag{3.1}$$

where

$$B_h \equiv V^n, \quad N_h^Q\left(V^{n+1}\right) \equiv V^{n+1} - \Delta\tau \inf_{Q\in\hat{Q}} \left\{\mathcal{L}^Q V^{n+1}\right\}, \tag{3.2}$$

and

$$\mathcal{L}^Q V^{n+1} = A^{n+1}(Q)\, V^{n+1} + D^{n+1}(Q). \tag{3.3}$$

The problem on the coarsest grid can either be solved by relaxation scheme (1.15) or policy iteration. Linear restriction and interpolation are applied for intergrid transfer.

We use the relaxation scheme as a smoother for FAS. Define the optimal control

$$Q_i^k \in \arg\inf_{Q\in\hat{Q}} \left\{\left(\mathcal{F}^Q \hat{V}^k\right)_i\right\},$$

where

$$\left(\mathcal{F}^Q \hat{V}^k\right)_i \equiv \frac{\Delta\tau\left(\alpha_i^{n+1}(Q)\,\hat{V}_{i-1}^k + \beta_i^{n+1}(Q)\,\hat{V}_{i+1}^k + d_i^{n+1}(Q)\right) + V_i^n}{1 + \Delta\tau\left(\alpha_i^{n+1}(Q) + \beta_i^{n+1}(Q) + c_i^{n+1}(Q)\right)}.$$

Then (1.15) can be written as

$$\hat{V}_i^{k+1} = \frac{\Delta\tau\left(\alpha_i^{n+1}\left(Q_i^k\right)\hat{V}_{i-1}^k + \beta_i^{n+1}\left(Q_i^k\right)\hat{V}_{i+1}^k + d_i^{n+1}\left(Q_i^k\right)\right) + V_i^n}{1 + \Delta\tau\left(\alpha_i^{n+1}\left(Q_i^k\right) + \beta_i^{n+1}\left(Q_i^k\right) + c_i^{n+1}\left(Q_i^k\right)\right)}. \tag{3.4}$$

It can be shown that applying one iteration of (3.4) is equivalent to applying one iteration of Jacobi method to the linear problem

$$J^k \cdot \hat{V}^{k+1} = l^k, \tag{3.5}$$

where J^k is a tri-diagonal matrix with elements

$$\begin{aligned}
J_{i,i}^k &= 1 + \Delta\tau\left(\alpha_i^{n+1}\left(Q_i^k\right) + \beta_i^{n+1}\left(Q_i^k\right) + c_i^{n+1}\left(Q_i^k\right)\right), \\
J_{i,i-1}^k &= -\Delta\tau\alpha_i^{n+1}\left(Q_i^k\right), \\
J_{i,i+1}^k &= -\Delta\tau\beta_i^{n+1}\left(Q_i^k\right),
\end{aligned}$$

Algorithm 3.2 Smoothing Iterations

1: Let $\hat{V}^0 = V^n$
2: **for** $k = 0, 1, ..., \mu - 1$ until convergence **do**
3: Determine $Q^k \in \arg\inf_{Q \in \hat{Q}} \left\{ \left(\mathcal{F}^Q \hat{V}^k \right)_i \right\}$
4: Construct J^k and l^k
5: $D \leftarrow diagonal(J^k)$
6: $\hat{V}^{k+1} = \hat{V}^k + \omega \cdot D^{-1} \left(l^k - J^k \hat{V}^k \right)$

for all i. The vector l^k is defined as

$$l^k = V^n + \Delta\tau d^{n+1} \left(Q_i^k \right).$$

To achieve a better smoothing effect, we introduce a damping factor ω to the relaxation scheme, as it is used for the damped-Jacobi method. This leads us to the iterative smoothing scheme Algorithm 3.2.

Then (3.4) becomes

$$\hat{V}_i^{k+1} = (1-\omega) V_i^k + \omega \frac{\Delta\tau \left(\alpha_i^{n+1} \left(Q_i^k \right) \hat{V}_{i-1}^k + \beta_i^{n+1} \left(Q_i^k \right) \hat{V}_{i+1}^k + d_i^{n+1} \left(Q_i^k \right) \right) + V_i^n}{1 + \Delta\tau \left(\alpha_i^{n+1} \left(Q_i^k \right) + \beta_i^{n+1} \left(Q_i^k \right) + c_i^{n+1} \left(Q_i^k \right) \right)},$$

With a carefully chosen damping factor, a smoothing factor close to 0.5 can be achieved. See more detailed smoothing analysis Chapter 4.

3.3 Difference Between Our FAS Method and MGS in [20]

The multigrid scheme MGS described in [20] is similar to our FAS scheme. It first applies smoothing, and then solve a nonlinear problem on the coarser grid, update the current approximation using coarse grid solution and apply the smoothing again. However, MGS and FAS are different in several ways. First, MGS uses W-cycles while our FAS uses V-cycles. Second, different smoothing procedures are applied. MGS first linearizes the HJB problem by finding the optimal control based on the current approximate solution, and then applies common smoothers in standard multigrid for a few iterations to the

linearized problem. The output of the smoother approximates the linearized system. However, FAS applies the non-linear damped-relaxation smoother in Section 3.2 to the problem directly, the output of the smoother approximates the nonlinear HJB problem. Third, the construction of coarse grid problem are different. Suppose the discrete HJB equation on the fine grids is

$$\inf_{Q\in\hat{Q}}\left\{A_h^Q\left(V_h\right)-f_h^Q\right\}=B_h$$

with $B_h\equiv 0$. Then the coarse grid problem is defined as $\inf_{Q\in\hat{Q}}\left\{A_H^Q\left(V_H\right)-f_H^Q\right\}=B_H$ where operator $A_H^Q\left(\cdot\right)$ is obtained from direct discretization. In MGS, $A_h^Q\left(V_h\right)$ and f_H^Q are considered separately. f_H^Q is obtained from

$$f_H^Q=A_H^Q\left(R\cdot V_h\right)+R\cdot\left(f_h^Q-A_h^Q\left(V_h\right)\right),$$

for all $Q\in\hat{Q}$, where R is the restriction operator. B_H is simply set to 0. In FAS, $\inf_{Q\in\hat{Q}}\left\{A_h^Q\left(V_h\right)-f_h^Q\right\}$ is considered as one nonlinear term. Let

$$N_h^Q\left(V_h\right)\equiv\inf_{Q\in\hat{Q}}\left\{A_h^Q\left(V_h\right)-f_h^Q\right\}$$

and $N_H^Q\left(\cdot\right)$ is obtained from direct discretization, therefore as part of the coarse grid operator, f_H^Q is obtained from direct discretization too. B_H is defined as

$$B_H=N_H^Q\left(R\cdot V_h\right)+R\cdot\left(B_h-N_h^Q\left(V_h\right)\right).$$

These differences in the algorithm design result in different convergence results, which will be shown in Chapter 5.

3.4 Multigrid Method for HJBI Equations

We do not apply policy iteration to the HJBI equations due to its uncertainty in convergence. Thus, as for the HJB equations, we will apply the FAS scheme we proposed

in Section 3.2 to the HJBI problem. Applying FAS to HJBI equations is very similar to FAS for HJB. The original nonlinear problem (1.11) is rewritten as

$$N_h^{Q,P}\left(V^{n+1}\right) = B_h, \tag{3.6}$$

where

$$B_h \equiv V^n,$$

$$N_h^{Q,P}\left(V^{n+1}\right) \equiv V^{n+1} - \Delta\tau \sup_{P\in\hat{P}} \inf_{Q\in\hat{Q}} \left\{\mathcal{L}^{Q,P}V^{n+1}\right\},$$

with

$$\mathcal{L}^{Q,P}V^{n+1} = A^{n+1}\left(Q,P\right)V^{n+1} + D^{n+1}\left(Q,P\right).$$

The smoothing iteration is similar to the HJB case except that both Q^k and P^k need to be determined. More precisely, define

$$\left(\mathcal{F}^{Q,P}\hat{V}^k\right)_i \equiv \frac{\Delta\tau\left(\alpha_i^{n+1}\left(Q,P\right)\hat{V}_{i-1}^k + \beta_i^{n+1}\left(Q,P\right)\hat{V}_{i+1}^k + d_i^{n+1}\left(Q,P\right)\right) + V_i^n}{1+\Delta\tau\left(\alpha_i^{n+1}\left(Q,P\right) + \beta_i^{n+1}\left(Q,P\right) + c_i^{n+1}\left(Q,P\right)\right)}.$$

To find the optimal control values Q and P at grid point i in the k^{th} iteration, we compute the value of $\left(\mathcal{F}^{Q,P}\hat{V}^k\right)_i$ for every Q with a fixed $P\in\hat{P}$ to obtain an infimum

$$\mathcal{I}\left(P\right) = \left(\mathcal{F}^{Q_P^*,P}\hat{V}^k\right)_i,$$

where

$$Q_P^* \in \arg\inf_{Q\in\hat{Q}}\left\{\left(\mathcal{F}^{Q,P}\hat{V}^k\right)_i\right\}.$$

Then compute the infimum $\mathcal{I}\left(P\right)$ for every $P\in\hat{P}$ to obtain the supremum of all $\mathcal{I}\left(P\right)$'s and its corresponding optimal P_i^*. The corresponding optimal Q_i^* is given by $Q_{P^*}^*$.

This process can be easily implemented by two nested loops described in Algorithm 3.3. Linear restriction and interpolation are used for intergrid transfer. On the coarsest

Algorithm 3.3 Optimal Control at Grid Point i for HJBI Smoother

Let $Sup = -\infty$
for all $P \in \hat{P}$ **do**
  Let $Inf = +\infty$
  for all $Q \in \hat{Q}$ **do**
    if $\left(\mathcal{F}^{Q,P}\hat{V}^k\right)_i < Inf$ **then**
      $Inf = \left(\mathcal{F}^{Q,P}\hat{V}^k\right)_i$, $Q_P^* = Q$
  if $Inf > Sup$ **then**
    $Sup = Inf$, $Q_i^* = Q_P^*$, $P_i^* = P$

level, the nonlinear problem is solved by applying the relaxation scheme.

3.5 Jumps in Control

The optimal control values P_i^* and Q_i^* can vary significantly from one grid point to another. Take the American options with stock borrowing fees example in Section 1.1.2 for example. In (1.6), there are two sets of controls, $\hat{Q}$ is composed by different combinations of r_l, r_b, 0 and 1, whose values do not change significantly, and so no special care is required. The other control μ has two possible values, 0 and 1. Note that μ is used with the penalty term: $\mu\frac{V-V^*}{\delta}$. Thus one can think of the control $\hat{P}$ is effectively $\{0, 10^8\}$, which will create a large jump when the optimal control P changes from grid point to grid point. This kind of problems can also appear when the control is not bounded. Ignoring such "jumps" in optimal control values could slow down the convergence or even render a diverging result.

As a result, special care for intergrid transfer is need. A modified version of FAS scheme for "jumps" in optimal control is developed. The smoothing iterations and the way to solve the coarsest grid problem are the same as FAS in previous sections, while the restriction and interpolation procedure are different. We use a two-grid FAS for an HJB problem (3.1) with $|\hat{Q}| = 2$ to illustrate the FAS scheme for "jumps" in control. The procedure of coarser grid problem construction and coarse grid correction we proposed are presented in this section.

3.5.1 Coarser Grid Problem Construction

The optimal control on the coarse grid might not be consistent with the optimal control on the fine grid. Suppose the coarse grid function $N_H^Q(\cdot)$ is a direct discretization of $N_h^Q(\cdot)$ on the coarse grid. The only information passed from the fine grid to the coarse grid is the approximate coarse grid solution V_H, which is the restricted approximate fine grid solution. In the standard approach, the coarse grid vector B_H is

$$B_H = R \cdot r_h + N_H^Q(V_H).$$

While fine grid residual r_h depends on the optimal control on the fine grid Q_h^*, $N_H^Q(V_H)$ depends on the optimal control on the coarse grid Q_H^*, which is obtained from (3.2). When $Q_H^* \neq Q_h^*$ at a coarse grid point, the two components of vector B_H are inconsistent with each other. Such discrepancy is introduced by the restriction process and mostly visible near the jump. Consider the plot for the optimal controls of an HJB problem in Figure 3.1 as an illustrative example. The x-axis stands for the grid point indices while the y-axis stands for the optimal control value. The first plot shows the fine grid optimal control Q_h^* on each grid point: the first five fine grid points have optimal control 10^8 while the others have optimal control 0. There is a "jump" in Q_h^* between grid point 5 and 6. The desired Q_H^* which would be consistent with the Q_h^* in the top plot is shown in the second plot, where $Q_H^* = Q_h^*$ at all coarse grid points. Due to the restriction process, Q_H^* could be either the third plot or the last plot in which one grid point is off near the "jump" position. Without taking special care to the coarse grid optimal control with jump, $N_H^Q(V_H)$ might have the Q_H^* shown in the third or the last plot while r_h has Q_h^* shown in the first plot, rendering an inconsistent B_H. When the "jump" size is large, the inconsistency will be very significant and the convergence of the FAS scheme will be slowed down.

To avoid such situation, the new FAS scheme forces Q_H^* to match with Q_h^* by altering V_H. Suppose Q_H^* and Q_h^* are different at a particular grid point, i.e. $Q_h^* = q_1$ and $Q_H^* = q_2$ and $q_1 \neq q_2$. From (3.2), we know that on the coarse grid, $\mathcal{L}^{q_2} V_H$ has to be the smallest among $\mathcal{L}^{Q_H} V_H$ since q_2 is the optimal control. Thus $\mathcal{L}^{q_1} V_H > \mathcal{L}^{q_2} V_H$. However, we need

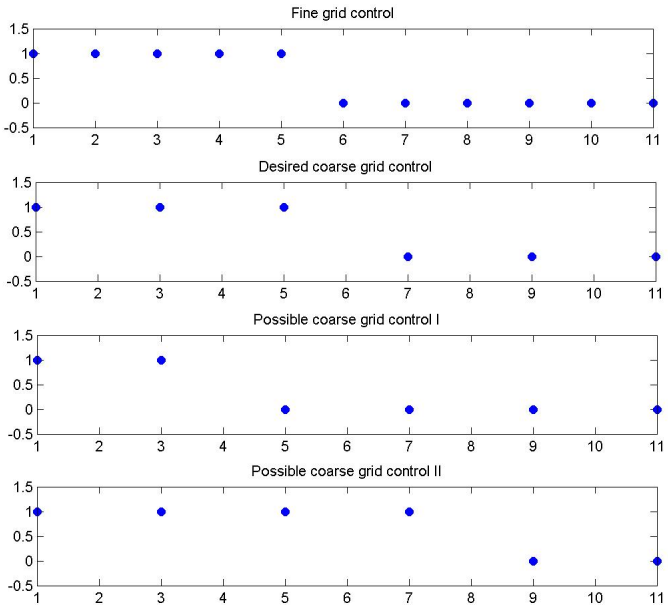

Figure 3.1: Fine Grid and Coarse Grid Controls

Algorithm 3.4 Coarse grid problem construction

1: **for** each coarse grid point **do**
2: i_H =coarse grid index
3: i_h =corresponding fine grid index
4: **if** $(Q_H^*)_{i_H} \neq (Q_h^*)_{i_h}$ **then**
5: $q_1 = (Q_h^*)_{i_h}$, $q_2 = (Q_H^*)_{i_H}$
6: $\delta \leftarrow$ a small positive number
7: Solve $(\mathcal{L}^{q_1} V_H)_{i_H} + \delta = (\mathcal{L}^{q_2} V_H)_{i_H}$ for $(V_H)_{i_H}$

$Q_H^* = q_1$, i.e. $\mathcal{L}^{q_1} V_H < \mathcal{L}^{q_2} V_H$ to make sure that the coarse grid control is consistent with the fine grid control. To force $Q_H^* = q_1$, solve for V_H at the grid point where the controls do not match by solving

$$\mathcal{L}^{q_1} V_H + \delta = \mathcal{L}^{q_2} V_H, \tag{3.7}$$

where δ is a very small positive number, e.g. 10^{-10}. Equation (3.7) is linear since q_1 and q_2 are fixed. Also it is defined on one grid point, hence it is a small linear problem which is easy to solve. This step ensures that $\mathcal{L}^{q_1} V_H < \mathcal{L}^{q_2} V_H$, yielding $Q_H^* = q_1$ and therefore $Q_H^* = Q_h^*$. After handling all the grid points that their controls are different, the coarse grid problem becomes consistent. The construction of coarser grid problem can be implemented as in Algorithm 3.4.

For problems with control set that has more than two values, i.e. $\hat{Q} = \{q_1, q_2, \ldots, q_n\}$,

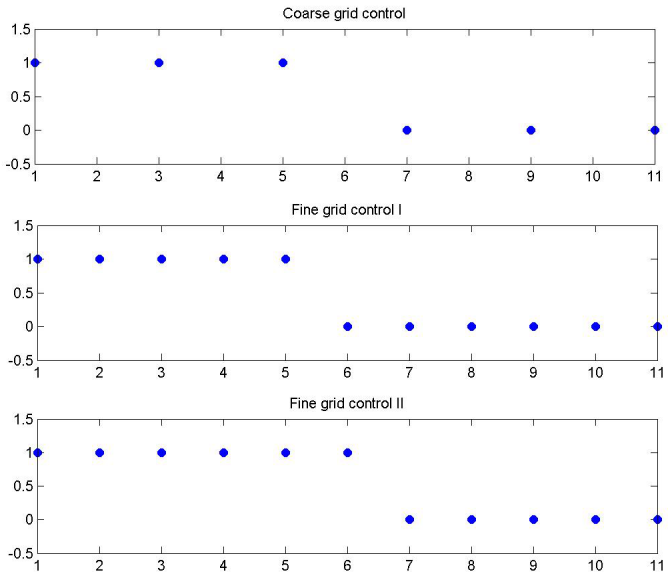

Figure 3.2: Optimal Control Interpolation

change (3.7) to

$$\mathcal{L}^{q_1} V_H + \delta = \min_{Q \in \{q_2, \ldots, q_n\}} \left\{ \mathcal{L}^Q V_H \right\},$$

and keep other steps the same.

3.5.2 Coarse Grid Correction

Unlike the function V, the control does not always have a continuous control set. As such, it is not clear how to interpolate or more precisely, how to define the control on the fine grid from the control on the coarse grid. Consider Figure 3.2. The plots show the optimal control for an HJB problem with a control set $\hat{Q} = \{q_1, q_2\}$, where the x-axis represents the grid points and the y-axis represents the value of optimal control on each grid point in 10^8 scale. From the coarse grid solution shown in the first plot, the optimal control is determined for every other fine grid point (grids with odd indices in this example). When there is no "jump" in the coarse grid control, the fine grid control is obtained from linear interpolation of coarse grid control, thus grids with even indices will have the same control as their neighboring grids have in the example. However, for

grid point 6, its neighboring grid points have different controls due to the "jump". Thus there are two possible scenarios for the optimal control on the fine grid: Fine grid control I and II. In other words, the optimal control at grid point 6 can either be the same as its left neighbor, or its right neighbor. It is not clear which one of the two possible controls we should use. The linear interpolation is not applicable at this point since the "jump" size is large and there is no intermediate control between the two in this example. If the control we choose to use is different from the one it should be, the convergence rate can be significantly slowed down or even yield divergence.

To address this issue, let i denote the fine grid index where the optimal control is different on its left and its right grid point ($i = 6$ in Figure 3.2), and $(Q_h^*)_i$ denote the optimal fine grid control on grid i. Since there are two possible $(Q_h^*)_i$'s, we will consider them separately.

Case A. Assume the correct fine grid optimal control is taken as Fine grid control I in Figure 3.2. Let $Q'_j = (Q_h^*)_j$ for all $j \neq i$ and $Q'_i = (Q_h^*)_{i+1}$. Let V'_h be the improved solution after standard coarse grid correction, i.e. $V'_h = V_h + P \cdot (V_H - R \cdot V_h)$ where V_h is the approximate fine grid solution after presmoothing, V_H is the coarse grid solution and P and R are the interpolation and restriction operator. Due to the jump in the control, the error of V'_h near grid point i could be large. Note that the control on the fine grid is now fixed, $N_h^{Q'}(\cdot)$ becomes a linear operator. As both $N_h^{Q'}(\cdot)$ and B_h are now deterministic, we can compute an improved fine grid solution $\tilde{V}'_h$ corresponding to Q' by solving the linear system $N_h^{Q'}\left(\tilde{V}'_h\right) = B_h$. Since the jump in control mainly affect neighboring grid points, a small local problem is considered instead to simplify the computation. With a pre-determined small positive integer m, which usually lies between 2 and 5, the local linear problem is defined as

$$\left[N_h^{Q'}\left(\tilde{V}'_h\right)\right]_j = (B_h)_j\,,\ j = i-m, ..., i+m, \tag{3.8}$$

which is centered at grid point i with size $2m+1$. Note that when we substitute $\left(\tilde{V}_h\right)_j$ back into (**??**), the resulting optimal control might not be the same as Q'_j. Hence we force the control to be the same as Q'_j by applying the same technique used in Section 3.5.1 to

Algorithm 3.5 Coarse Grid Correction

Let $V_h' = V_h'' = V_h + P \cdot (V_H - R \cdot V_h)$
for $i = 2, 3, ..., M-1$ **do**
  if $(Q_h^*)_{i-1} \neq (Q_h^*)_{i+1}$ **then**
    Let $(Q_h^*)_i = (Q_h^*)_{i-1}$
    Solve $\left[N_h^{Q_h^*}\left(\tilde{V}_h\right)\right]_j = (B_h)_j$, $j = i-m, ..., i+m$
    Obtain $\left(\tilde{Q}_h^*\right)_j \in \arg\inf_{Q \in \hat{Q}} \left\{\mathcal{L}^Q\left(\tilde{V}_h\right)_j\right\}$
    for all $\left(\tilde{Q}_h^*\right)_j \neq (Q_h^*)_j$ **do**
      Solve $\left(\mathcal{L}^{Q_h^*}\tilde{V}_h\right)_j + \delta = \left(\mathcal{L}^{\tilde{Q}_h^*}\tilde{V}_h\right)_j$
    $(V_h')_j = \left(\tilde{V}_h\right)_j$, $j = i-m+1, ..., i+m-1$
    $r_h' = N_h^Q(V_h') - B_h$

    Let $(Q_h^*)_i = (Q_h^*)_{i+1}$
    Repeat the above procedure and obtain V_h''
    $r_h'' = N_h^Q(V_h'') - B_h$

    if $\|r_h'\| < \|r_h''\|$ **then**
      $V_h = V_h'$
    else
      $V_h = V_h''$

the local problem, and then update V_h' by setting $(V_h')_j = \left(\tilde{V}_h\right)_j$.

Case B. Assume the correct fine grid optimal control is taken as Fine grid control II in Figure 3.2. Let $Q_j'' = (Q_h^*)_j$ for all $j \neq i$ and $Q_i'' = (Q_h^*)_{i-1}$. Let $V_h' = V_h + P \cdot (V_H - R \cdot V_h)$. Then repeat the process of Case A and obtain the updated solution $(V_h'')_j$.

V_h' and V_h'' are two possible fine grid updated solutions. In general, it is difficult to tell which one is the desired solution. Assuming the correct fine grid control will yield a solution with smaller residual, we choose the one with smaller residual norm. The scheme is shown in Algorithm 3.5.

Chapter 4

Smoothing Analysis

4.1 Local Fourier Analysis

We investigate the smoothing property of the damped-relaxation smoother by applying Local Fourier Analysis (LFA) [31]. LFA evaluates the quantitative convergence behavior and efficiency of an operator. The idea is to linearize a general discrete operator locally and replace it by one with constant coefficients. Define grid functions as

$$\varphi(\theta, x) = e^{\mathbf{i}\theta x/h},$$

where x varies in a given infinite grid Ω_h, θ characterizes the frequency of the grid function and is continuous. Since

$$\varphi(\theta, x) \equiv \varphi(\theta + 2\pi j, x), \ j = 1, 2, ...,$$

it is sufficient to consider $\varphi(\theta, x)$ with $\theta \in [-\pi, \pi)$. For $\theta \in [-\pi, \pi)$, all grid functions $\varphi(\theta, x)$ are eigenfunctions of any discrete operator. Consider a discrete operator L_h corresponding to a difference stencil. The relation

$$L_h \varphi(\theta, x) = \widetilde{L}_h(\theta) \varphi(\theta, x)$$

holds, where $\widetilde{L}_h(\theta) = \sum_{\kappa} s_\kappa e^{\mathbf{i}\theta\cdot\kappa}$ is the symbol of L_h with constant coefficients $s_\kappa \in \mathbb{R}$.

Also, for the smoothing analysis, we have to distinguish high and low frequency components for standard coarsening, which are defined as

φ low frequency component $\Longleftrightarrow \theta \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right)$,

φ high frequency component $\Longleftrightarrow \theta \in [-\pi, \pi) \setminus \left[-\frac{\pi}{2}, \frac{\pi}{2}\right)$.

For a discretized PDE on Ω_h, let $\hat{u}_h^k$ be the old approximation of the exact solution $\bar{u}_h$ and $\hat{u}_h^{k+1}$ be the new approximation after 1 iteration of smoothing. Let $\epsilon_h^k = \bar{u}_h - \hat{u}_h^k$ and $\epsilon_h^{k+1} = \bar{u}_h - \hat{u}_h^{k+1}$ be the old and new error respectively. If $\epsilon_h^{k+1} = S_h \epsilon_h^k$ where S_h is a discrete operator, we can compute the amplification factor $\left|\widetilde{S}_h(\theta)\right|$. Since we will focus on the smoothing effect, which is the error reduction on high frequency components, the smoothing factor $\mu(S_h)$ is defined as

$$\mu(S_h) \equiv \sup\left\{\left|\widetilde{S}_h(\theta)\right| : \theta \in [-\pi, \pi) \setminus \left[-\frac{\pi}{2}, \frac{\pi}{2}\right)\right\}.$$

The smaller its smoothing factor is, the more desirable a smoother is.

4.2 Smoothing Analysis for HJB Equations

Assume the exact solution for time step $n+1$ is $\bar{V}$ and the approximate solution after the k^{th} smoothing iteration is $\hat{V}^k = \bar{V} + \epsilon^k$, where ϵ^k is the error after the k^{th} iteration. By (1.15), we get

$$\bar{V}_i + \epsilon_i^{k+1} = \inf_{Q\in\hat{Q}}\left\{\frac{\Delta\tau\left(\alpha_i^{n+1}(Q)\left(\bar{V}_{i-1}+\epsilon_{i-1}^k\right) + \beta_i^{n+1}(Q)\left(\bar{V}_{i+1}+\epsilon_{i+1}^k\right) + d_i^{n+1}(Q)\right) + V_i^n}{1+\Delta\tau\left(\alpha_i^{n+1}(Q)+\beta_i^{n+1}(Q)+c_i^{n+1}(Q)\right)}\right\}. \tag{4.1}$$

After obtaining the optimal control and rearranging terms, (4.1) can be written in vector form as

$$\epsilon_i^{k+1} = \begin{bmatrix} \frac{\Delta\tau\cdot\alpha_i^*}{1+\Delta\tau\left(\alpha_i^*+\beta_i^*+c_i^*\right)} & 0 & \frac{\Delta\tau\cdot\beta_i^*}{1+\Delta\tau\left(\alpha_i^*+\beta_i^*+c_i^*\right)} \end{bmatrix} \cdot \begin{bmatrix} \epsilon_{i-1}^k \\ \epsilon_i^k \\ \epsilon_{i+1}^k \end{bmatrix} + C\left(Q_i^k\right), \tag{4.2}$$

where

$$\alpha_i^* = \alpha_i^{n+1}\left(Q_i^k\right),\ \beta_i^* = \beta_i^{n+1}\left(Q_i^k\right),$$

$$c_i^* = c_i^{n+1}\left(Q_i^k\right),\ d_i^* = d_i^{n+1}\left(Q_i^k\right),$$

with

$$Q_i^k \in \underset{Q\in\hat{Q}}{\arg\inf}\left\{\left(\mathcal{F}^Q\left(\bar{V}+\epsilon^k\right)\right)_i\right\},$$

and

$$C\left(Q_i^k\right) = \begin{bmatrix} \frac{\Delta\tau\cdot\alpha_i^*}{1+\Delta\tau\left(\alpha_i^*+\beta_i^*+c_i^*\right)} & -1 & \frac{\Delta\tau\cdot\beta_i^*}{1+\Delta\tau\left(\alpha_i^*+\beta_i^*+c_i^*\right)} \end{bmatrix}\cdot\begin{bmatrix} \bar{V}_{i-1} \\ \bar{V}_i \\ \bar{V}_{i+1} \end{bmatrix}$$
$$+\frac{V_i^n+\Delta\tau\cdot d_i^*}{1+\Delta\tau\left(\alpha_i^*+\beta_i^*+c_i^*\right)}.$$

Let $\bar{Q}_i$ be the optimal control corresponds to the exact solution $\bar{V}$. Suppose $Q_i^k = \bar{Q}_i$ for all i. From (1.12), we deduce that $C\left(Q^k\right)$ is a zero vector. Thus (4.2) can be written as

$$\epsilon^{k+1} = S^k\cdot\epsilon^k,$$

where

$$\begin{aligned} S_{i,i}^k &= 0,\ S_{i,i-1}^k = \frac{\Delta\tau\cdot\alpha_i^*}{1+\Delta\tau\left(\alpha_i^*+\beta_i^*+c_i^*\right)}, \\ S_{i,i+1}^k &= \frac{\Delta\tau\cdot\beta_i^*}{1+\Delta\tau\left(\alpha_i^*+\beta_i^*+c_i^*\right)},\ i=1,2,...,M. \end{aligned}$$

Applying S^k to the eigenfunctions $\varphi(\theta,x)$, we obtain

$$S^k\varphi(\theta,\,x) = \widetilde{S^k}(\theta)\,\varphi(\theta,x),\ -\pi\le\theta<\pi,$$

and the symbol of S^k is

$$\widetilde{S_i^k}(\theta) = \frac{\Delta\tau\cdot\alpha_i^*\cdot e^{-\mathbf{i}\theta}+\Delta\tau\cdot\beta_i^*\cdot e^{\mathbf{i}\theta}}{1+\Delta\tau\left(\alpha_i^*+\beta_i^*+c_i^*\right)}. \tag{4.3}$$

As we use the damped-relaxation scheme (1.15) for smoothing, $\widetilde{S_i^k}(\theta)$ of the HJB smoother can be obtained by introducing damping factor ω to (4.3)

$$\widetilde{S_i^k}(\theta,\omega)=\frac{\Delta\tau\cdot\alpha_i^*\cdot e^{-\mathbf{i}\theta}-(1-\frac{1}{\omega})\left[1+\Delta\tau\left(\alpha_i^*+\beta_i^*+c_i^*\right)\right]+\Delta\tau\cdot\beta_i^*\cdot e^{\mathbf{i}\theta}}{\frac{1}{\omega}\cdot\left[1+\Delta\tau\left(\alpha_i^*+\beta_i^*+c_i^*\right)\right]}. \tag{4.4}$$

Simplify (4.4), we obtain

$$\widetilde{S_i^k}(\theta,\omega)=\frac{\omega\Delta\tau\left(\alpha_i^*+\beta_i^*\right)\cos\theta+(1-\omega)\left[1+\Delta\tau\left(\alpha_i^*+\beta_i^*+c_i^*\right)\right]+\mathbf{i}\omega\Delta\tau\left(\beta_i^*-\alpha_i^*\right)\sin\theta}{1+\Delta\tau\left(\alpha_i^*+\beta_i^*+c_i^*\right)}. \tag{4.5}$$

4.2.1 Smoothing Factors of the Example HJB Equation

Since generating a useful analytical expression for $\widetilde{S_i^k}(\theta,\omega)$ is very complicated, we will consider $\widetilde{S_i^k}(\theta,\omega)$ for specific values of θ, which represent the high, medium and low frequency components respectively. Also, though we assume the optimal control is fixed for each grid point from iteration to iteration, we will examine many possible values of optimal controls for each grid point to make sure that the smoother is efficient even with the worst case optimal control for all grid points.

For simplicity of the analysis, we transform the equation to the log scale, which is a common practice in option pricing literature. Let $X=\log W$. Then $W=e^X$. Substituting $W=e^X$ into (1.2), the HJB example problem on log grid can be written as

$$V_\tau=\inf_{Q\in\hat{Q}}\left\{\frac{1}{2}q^2\sigma^2V_{XX}+\left(r+q\sigma\xi-\frac{1}{2}q^2\sigma^2+\frac{\pi}{e^X}\right)V_X\right\}.$$

For simplicity, we will assume $\pi=0$. Then the coefficients for the example HJB problem on log grid in (1.7) are

$$a(\tau,Q)=\frac{1}{2}q^2\sigma^2,\ b(\tau,Q)=r+Q\sigma\xi-\frac{1}{2}q^2\sigma^2,\ c(\tau,Q)=0,\ d(\tau,Q)=0.$$

We note that the coefficients on the log grid do not depend on X or S, which is a desirable property for LFA. Substitute the above log grid coefficients to (1.10) assuming

central differencing and uniform grid, we obtain

$$\alpha(Q) = \frac{2a(Q)}{2h^2} - \frac{b(Q)}{2h} = \frac{q^2\sigma^2}{2h^2} - \frac{r + q\sigma\xi - \frac{1}{2}q^2\sigma^2}{2h},$$

$$\beta(Q) = \frac{2a(Q)}{2h^2} + \frac{b(Q)}{2h} = \frac{q^2\sigma^2}{2h^2} + \frac{r + q\sigma\xi - \frac{1}{2}q^2\sigma^2}{2h},$$

where h is the grid size. For the pension plan asset allocation problem, the typical values for the parameters are $\sigma = 0.15$, $\xi = 0.33$, $r = 0.03$, $\Delta\tau = 0.01$ and $q \in [0,\ 1.5]$.

4.2.1.1 Low Frequency Components:

Let $\theta \approx 0$. In this case, $\sin\theta \approx 0$ and $\cos\theta \approx 1$,

$$\begin{aligned}
\widetilde{S_i^k}(\theta,\omega) &\approx \frac{\omega\Delta\tau(\alpha_i^* + \beta_i^*) + (1-\omega)[1 + \Delta\tau(\alpha_i^* + \beta_i^*)]}{1 + \Delta\tau(\alpha_i^* + \beta_i^*)} \\
&= 1 - \frac{\omega}{1 + \Delta\tau(\alpha_i^* + \beta_i^*)} \\
&= 1 - \frac{\omega}{1 + \Delta\tau \cdot \frac{q^2\sigma^2}{h^2}} \\
&= 1 - \frac{\omega}{1 + 0.01 \cdot \frac{q^2 0.15^2}{h^2}}.
\end{aligned}$$

Since $q \in [0,\ 1.5]$, $\widetilde{S_i^k}(\theta,\omega) \in [1-\omega,\ 1 - \frac{\omega}{1+0.01\cdot\frac{1.5^2 0.15^2}{h^2}}]$. The modulo of $\widetilde{S_i^k}(0,\omega)$ has to be smaller or equal to 1 to ensure that the smoother does not diverge, therefore we can obtain $\omega \in [0,2]$. To demonstrate the relationship between the parameters and the amplification factor $\left|\widetilde{S_i^k}(\theta,\omega)\right|$, we fix h to be 2^{-6}, which is a common value for grid size, and plot $\left|\widetilde{S_i^k}(\theta,\omega)\right|$ with different values of ω for different control values in Figure 4.1. We can see that for all values of $\omega \in [0,2]$, $\left|\widetilde{S_i^k}(\theta,\omega)\right| < 1$, which is consistent with the analysis. Since the amplification factor for low frequency components is not important in smoother design as long as it is no greater than 1, the possible range of ω is narrowed down to $\omega \in [0,2]$. On the other hand, fixing ω to be $\frac{2}{3}$ and varying h would yield a plot

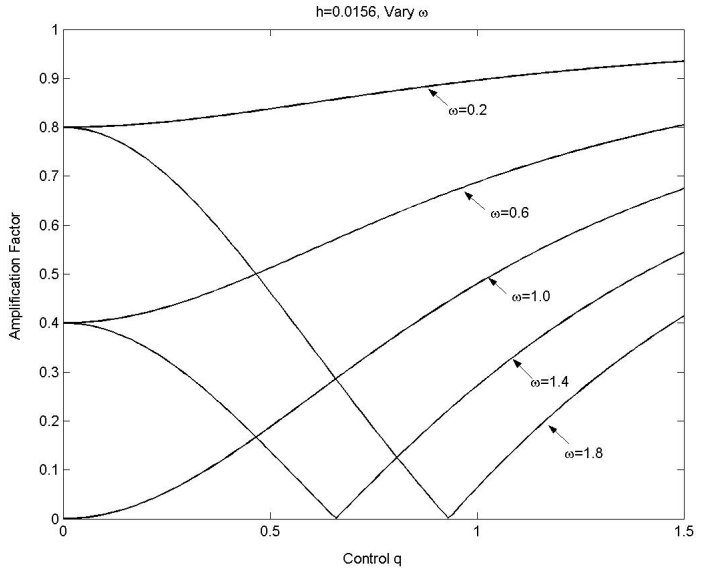

Figure 4.1: Amplification factor for the example HJB equation with $\theta \approx 0$, $h = 2^{-6}$ and different values of ω

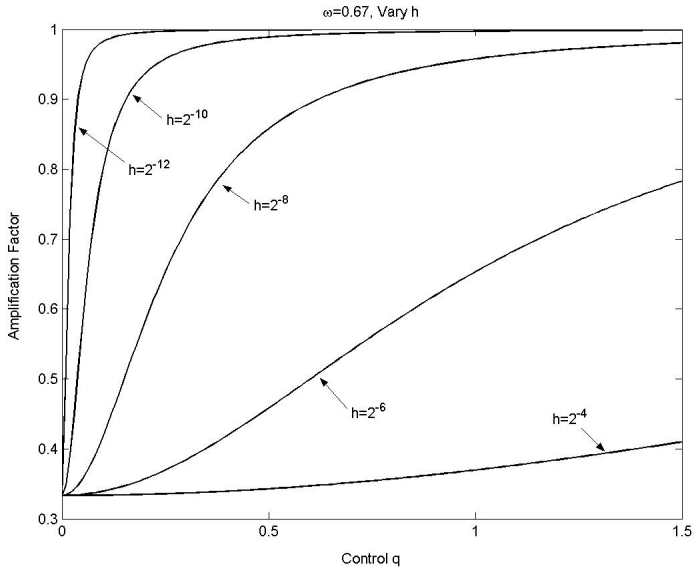

Figure 4.2: Amplification factor for the example HJB equation with $\theta \approx 0$, $\omega = \frac{2}{3}$ and different values of h

shown in Figure 4.2. It is clear that for different values of h and control q, $\left|\widetilde{S_i^k}\left(\theta, \frac{2}{3}\right)\right| \leq 1$.

4.2.1.2 Medium Frequency Components:

Let $\theta \approx \frac{\pi}{2}$. In this case, $\sin\theta \approx 1$ and $\cos\theta \approx 0$. $\widetilde{S_i^k}(\theta, \omega)$ becomes

$$\widetilde{S_i^k}(\theta, \omega) \approx \frac{\mathbf{i}\omega\Delta\tau\left(\beta_i^* - \alpha_i^*\right) + (1-\omega)\left[1 + \Delta\tau\left(\alpha_i^* + \beta_i^*\right)\right]}{1 + \Delta\tau\left(\alpha_i^* + \beta_i^*\right)},$$

and the amplification factor is

$$\left|\widetilde{S_i^k}(\theta, \omega)\right| \approx \frac{\sqrt{\left[\omega\Delta\tau\left(\beta_i^* - \alpha_i^*\right)\right]^2 + \left[(1-\omega)\left(1 + \Delta\tau\left(\alpha_i^* + \beta_i^*\right)\right)\right]^2}}{\left|1 + \Delta\tau\left(\alpha_i^* + \beta_i^*\right)\right|}.$$

It is easy to show that both $\beta_i(Q) - \alpha_i(Q) > 0$ and $\alpha_i(Q) + \beta_i(Q) \geq 0$ for all possible $Q \in \hat{Q}$ with the given constants. Therefore

$$\begin{aligned}\left|\widetilde{S_i^k}(\theta, \omega)\right| &\leq \frac{\omega\Delta\tau\left(\beta_i^* - \alpha_i^*\right) + |1-\omega|\left[1 + \Delta\tau\left(\alpha_i^* + \beta_i^*\right)\right]}{1 + \Delta\tau\left(\alpha_i^* + \beta_i^*\right)} \\ &= |1-\omega| + \frac{\omega h\Delta\tau\left(r + q\sigma\xi - \frac{1}{2}q^2\sigma^2\right)}{h^2 + \Delta\tau q^2\sigma^2}.\end{aligned}$$

Let

$$y(\omega, h) \equiv |1-\omega| + \frac{\omega h\Delta\tau\left(r + q\sigma\xi - \frac{1}{2}q^2\sigma^2\right)}{h^2 + \Delta\tau q^2\sigma^2}. \tag{4.6}$$

With h fixed to 2^{-6} and ω varies between 0 and 2, Figure 4.3 presents the plot of $y(\omega, h)$, which is smaller than 1 for different values of ω and q. When ω is fixed to $\frac{2}{3}$, $y(\omega, h)$ with different values of h is plotted in Figure 4.4. It shows that when the grid size is very small, $\left|\widetilde{S_i^k}(\theta, \omega)\right|$ increases as the control q approaches 0. Substituting $q = 0$ into (4.6), we obtain

$$y(\omega, h) = |1-\omega| + \frac{\omega r\Delta\tau}{h} = |1-\omega| + \frac{3 \times 10^{-4}\omega}{h}.$$

Hence as long as $h > 3 \times 10^{-4}$, $y(\omega, h) < 1$. The analysis and the plots indicate that with a practical grid size, all $\omega \in (0, 2)$ will yield an amplification factor smaller than 1 and

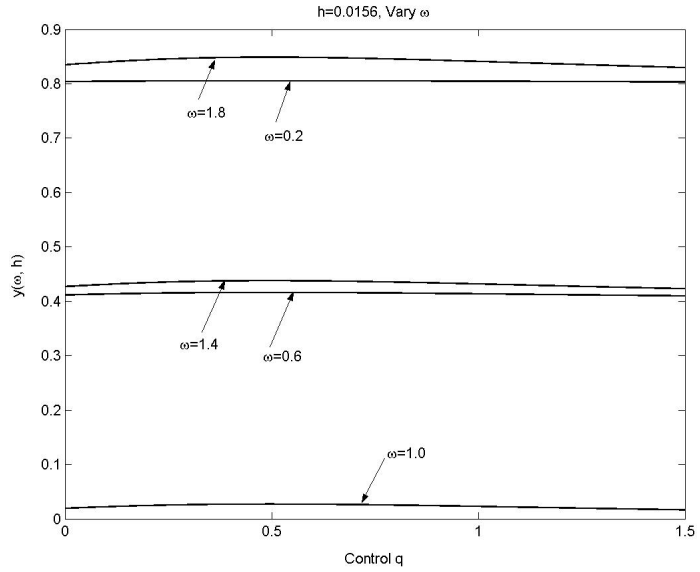

Figure 4.3: $y(\omega, h)$ for the example HJB equation with $\theta \approx \frac{\pi}{2}$, $h = 2^{-6}$ and different values of ω

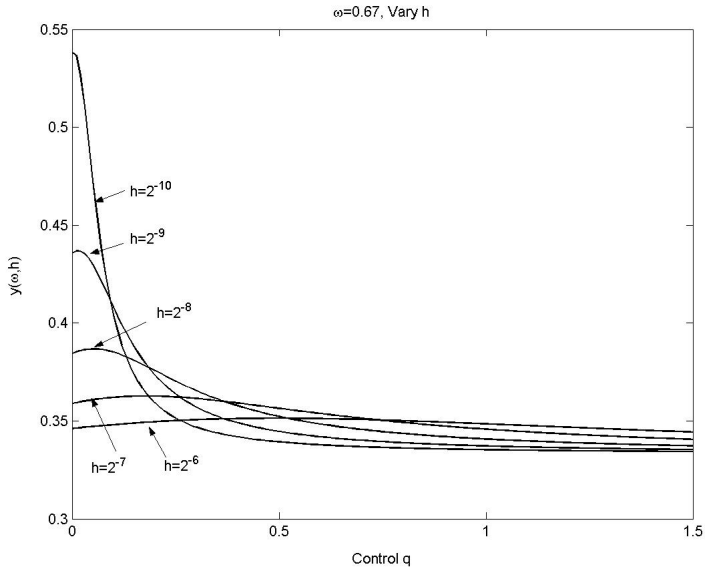

Figure 4.4: $y(\omega, h)$ for the example HJB equation with $\theta \approx \frac{\pi}{2}$, $\omega = \frac{2}{3}$ and different values of h

eligible for smoothers.

4.2.1.3 High Frequency Components:

Let $\theta \approx -\pi$. In this case, $\sin\theta \approx 0$ and $\cos\theta \approx -1$,

$$\begin{aligned}
\widetilde{S_i^k}(\theta,\omega) &\approx \frac{-\omega\Delta\tau(\alpha_i^* + \beta_i^*) + (1-\omega)[1 + \Delta\tau(\alpha_i^* + \beta_i^*)]}{1 + \Delta\tau(\alpha_i^* + \beta_i^*)} \\
&= 1 - 2\omega + \frac{\omega}{1 + \Delta\tau(\alpha_i^* + \beta_i^*)} \\
&= 1 - 2\omega + \frac{\omega}{1 + \Delta\tau \cdot \frac{q^2\sigma^2}{h^2}} \\
&= 1 - 2\omega + \frac{\omega}{1 + 0.01 \cdot \frac{q^2 0.15^2}{h^2}}.
\end{aligned} \tag{4.7}$$

Since $q \in [0,\ 1.5]$, $\widetilde{S_i^k}(\theta,\omega) \in [1 - 2\omega + \frac{\omega}{1+0.01\cdot\frac{1.5^2 0.15^2}{h^2}},\ 1-\omega]$. For $h \to 0$, $\widetilde{S_i^k}(\theta,\omega) \in [1-2\omega,\ 1-\omega]$ and $\left|\widetilde{S_i^k}(\theta,\omega)\right| \le \max(|1-2\omega|, |1-\omega|)$ for all $\omega \in [0,\ 2]$. It is easy to show that the upper bound is less than 1 when $\omega \in (0,\ 1)$ and the upper bound is minimized when $\omega^* = \frac{2}{3}$ and $\left|\widetilde{S_i^k}(\theta,\omega^*)\right| \le \frac{1}{3}$. Plots for $\left|\widetilde{S_i^k}(\theta,\omega)\right|$ with fixed h, different ω and fixed ω, different h are shown in Figure 4.5 and Figure 4.6 respectively, which verifies the above analysis.

The analysis of the three different scenarios shows that $\omega = \frac{2}{3}$ is an eligible damping factor and it yields an efficient smoothing effect among different values of ω.

4.2.1.4 The Actual Smoothing Factor for the Example HJB Equation:

Since we were only able to compute the theoretical value of amplification factor on some particular frequencies, it is desirable to evaluate its actual value by making plots of $\left|\widetilde{S}(\theta,\omega)\right|$ in (4.4) for different frequencies, with different values of ω, grid size and optimal control.

Previous analysis shows that ω has to lie between 0 and 1 to reach convergence and $\omega = \frac{2}{3}$ appears to produce the optimal smoothing effect. To confirm this argument, we will first find out whether the optimal ω is greater than, equal to or less than 1.

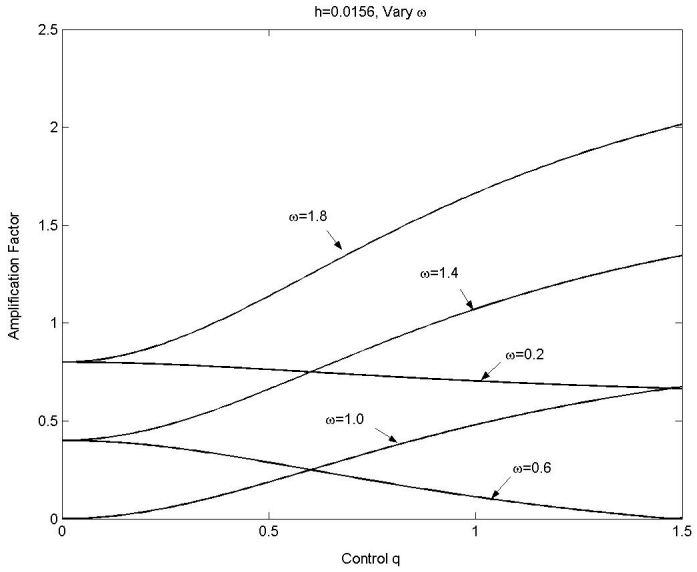

Figure 4.5: Amplification factor for the example HJB equation with $\theta \approx -\pi$, $h = 2^{-6}$ and different values of ω

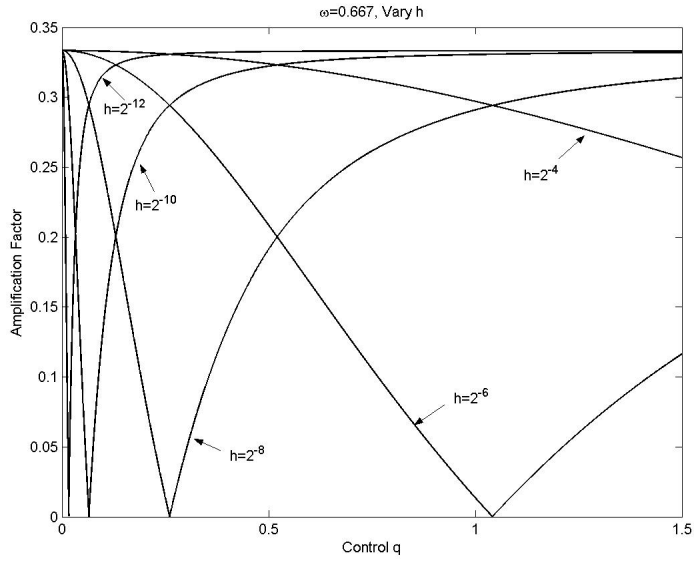

Figure 4.6: Amplification factor for the example HJB equation with $\theta \approx -\pi$, $\omega = \frac{2}{3}$ and different values of h

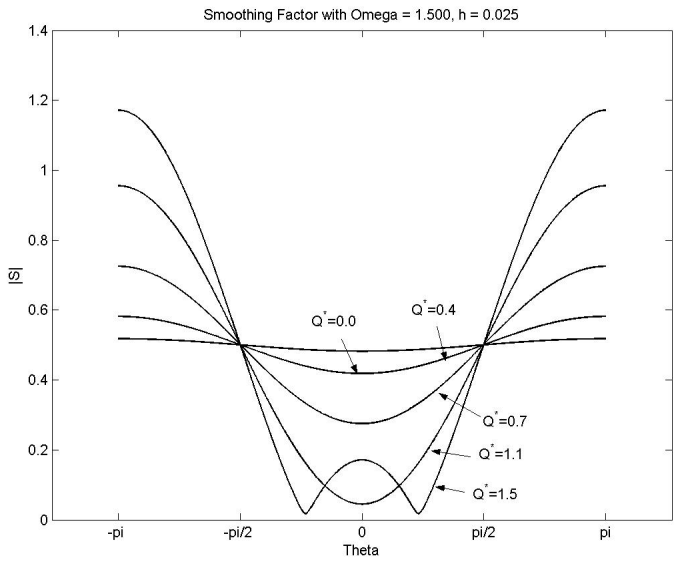

Figure 4.7: Amplification factor for the example HJB equation with $\omega = 1.5$, $h = 0.025$ and different values of optimal control

Let $h = 0.025$, which is a practical value for grid size. Plotting $\left|\widetilde{S}(\theta, \omega)\right|$ against θ which varies from $-\pi$ to π, we obtain Figure 4.7 with $\omega = 1.5$, Figure 4.8 with $\omega = 1$ and Figure 4.9 with $\omega = \frac{2}{3}$. Each plot has five curves corresponding to different values of optimal control Q^* that varies from 0 to 1.5. It is obvious that when $\omega = 1.5$, the smoothing factor can be greater than 1 for high frequency components, rendering a diverging iterative method. Since for all $\omega > 1$, the plots for $\left|\widetilde{S}(\theta, \omega)\right|$ will be similar to Figure 4.7, $\omega > 1$ will not be considered for smoothers. When $\omega = 1$, all the curves are below 1. The iterative scheme is convergent, however $\left|\widetilde{S}(\theta, 1)\right|$ reaches its maximal when θ is 0 or $\pm\pi$. While an efficient smoother will have its smoothing factor minimized for high frequency components, $\omega = 1$ may not be the optimal choice. On the other hand, $\left|\widetilde{S}\left(\theta, \frac{2}{3}\right)\right|$ has the properties of an effective smoother: low for high frequency components and bounded by 1 for low frequency components. It reaches its minimal when $\theta \approx \pm\pi$ and it is bounded above by $\frac{1}{3}$ when $|\theta| \geq \frac{\pi}{2}$. As for all $\omega \in (0, 1)$, their smoothing properties are similar to each other, we now know that the optimal range of damping factor ω is between 0 and 1.

To narrow down the range for ω, we will plot the smoothing factor with different $\omega \in (0, 1)$. As shown in Figure 4.9, all the curves are bounded above by $Q^* = 1.5$ and $Q^* = 0$, hence making plots for these two control values will be sufficient. Figure 4.10

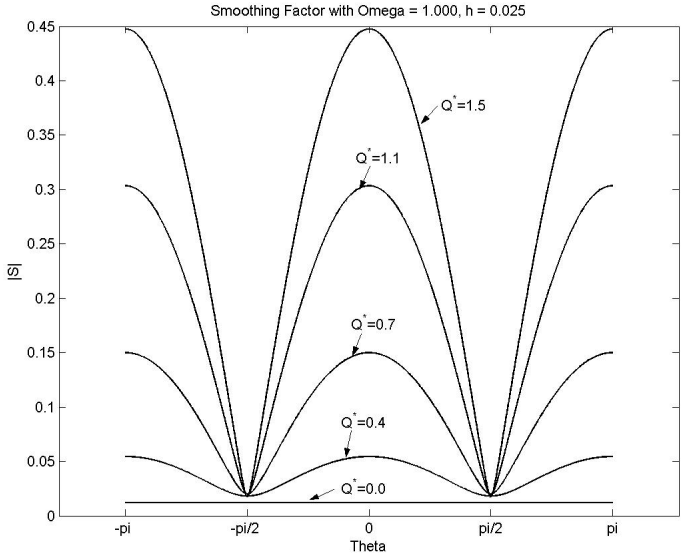

Figure 4.8: Amplification factor for the example HJB equation with $\omega = 1$, $h = 0.025$ and different values of optimal control

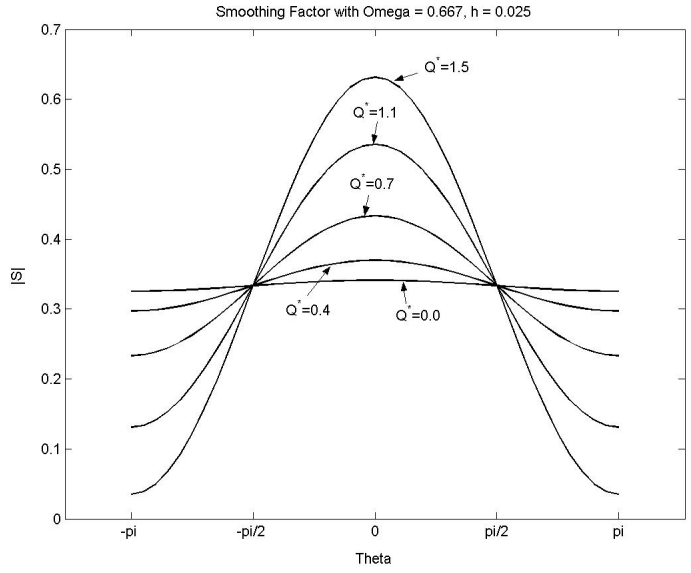

Figure 4.9: Amplification factor for the example HJB equation with $\omega = \frac{2}{3}$,$h = 0.025$ and different values of optimal control

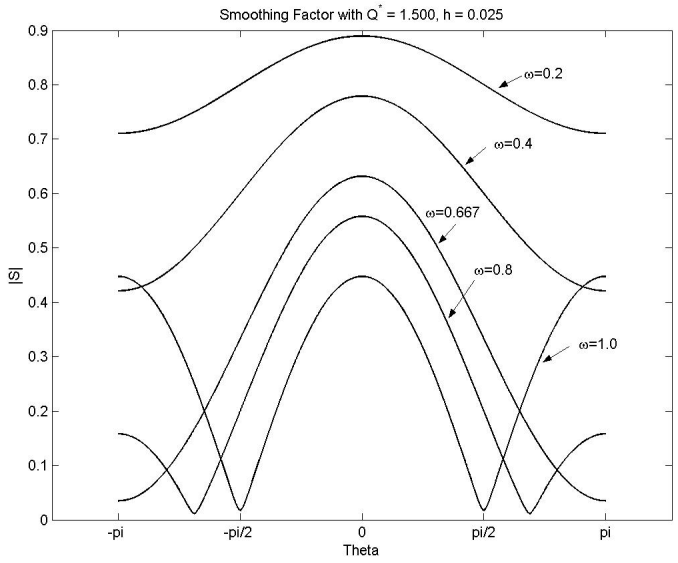

Figure 4.10: Amplification factor for the example HJB equation with $h = 0.025$, $Q^* = 1.5$ and different values of ω

shows the plot for $Q^* = 1.5$. Clearly, $\omega = \frac{2}{3}$ and $\omega = 0.8$ are giving more preferable results than others. Although $\omega = 0.8$ has smaller amplification factor on all low frequency components and part of high frequency components, $\omega = \frac{2}{3}$ is more desirable for the highest frequency components. Considering the purpose of smoothers, we would choose $\omega = \frac{2}{3}$. However in Figure 4.11, the amplification factor gets smaller as ω approaches 1. Hence we cannot determine whether $\omega = \frac{2}{3}$ or $\omega = 0.8$ is better.

To further investigate the relationship between $\left|\widetilde{S}(\theta, \omega)\right|$ and the parameters, we will alter the grid size h this time. When $Q^* = 0$, as shown in Figure 4.12 and Figure 4.13, both $\omega = \frac{2}{3}$ and $\omega = 0.8$ have satisfactory performance and $\omega = 0.8$ has smaller amplification factor for both high and low frequency components. However in Figure 4.14 and Figure 4.15, we can see that as h approaches 0, the "turning point" of the amplification factor gets closer to $\theta = \pm\frac{\pi}{2}$ for both $\omega = \frac{2}{3}$ and $\omega = 0.8$, rendering larger $\left|\widetilde{S}(\theta, \omega)\right|$ for higher frequency components. It is clear from the plots that when h is small, the performance for $\omega = 0.8$ is worse than $\omega = \frac{2}{3}$. Moreover, from (4.7) we can obtain $\widetilde{S}(-\pi, \omega) \to 1 - 2\omega$ as $h \to 0$. Therefore the limit of $\left|\widetilde{S}(-\pi, 0.8)\right|$ is 0.6, which is greater than $\frac{1}{3}$, the limit of $\left|\widetilde{S}\left(-\pi, \frac{2}{3}\right)\right|$. Considering the grid size we are using, $\omega = \frac{2}{3}$ appears to be preferable than

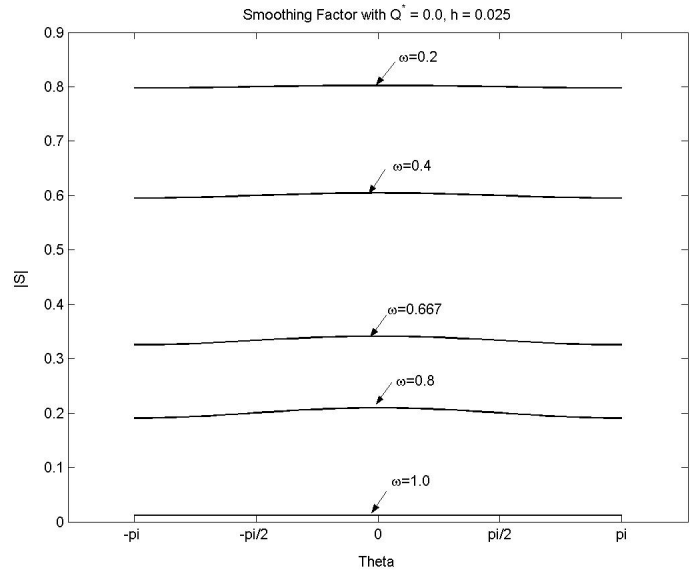

Figure 4.11: Amplification factor for the example HJB equation with $h = 0.025$, $Q^* = 0$ and different values of ω

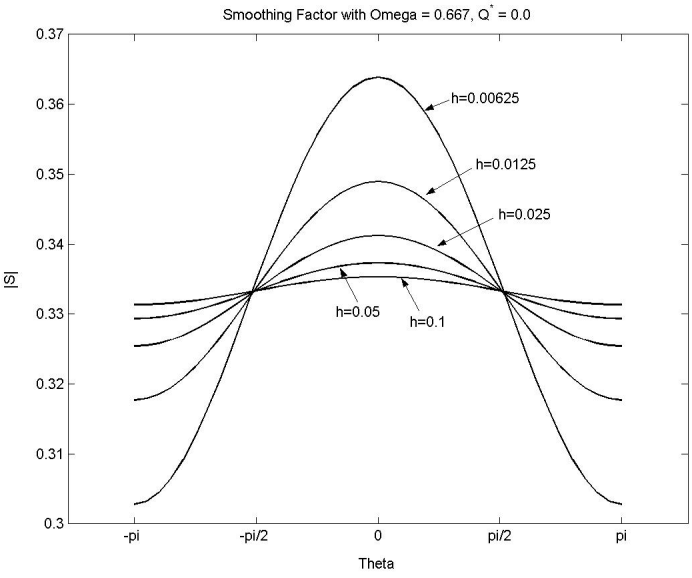

Figure 4.12: Amplification factor for the example HJB equation with $\omega = \frac{2}{3}$, $Q^* = 0$ and different values of h

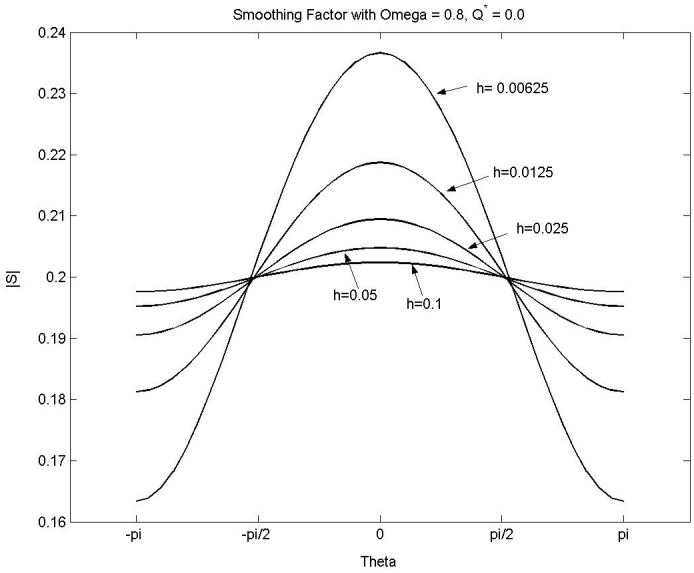

Figure 4.13: Amplification factor for the example HJB equation with $\omega = 0.8$, $Q^* = 0$ and different values of h

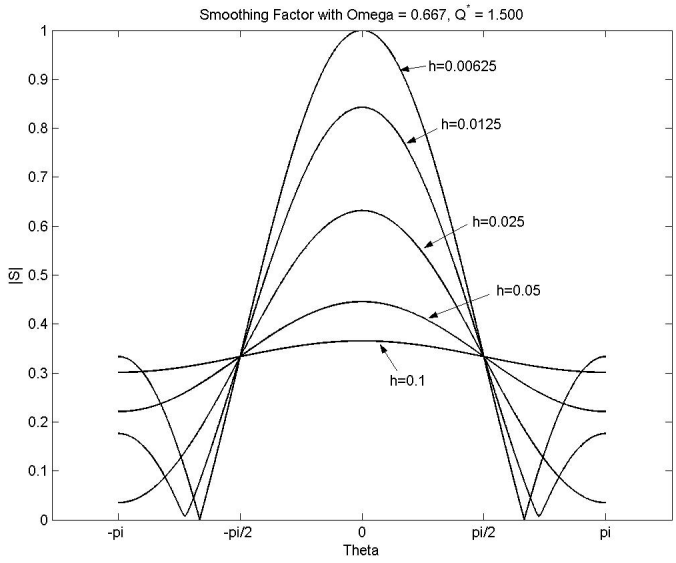

Figure 4.14: Amplification factor for the example HJB equation with $\omega = \frac{2}{3}$, $Q^* = 1.5$ and different values of h

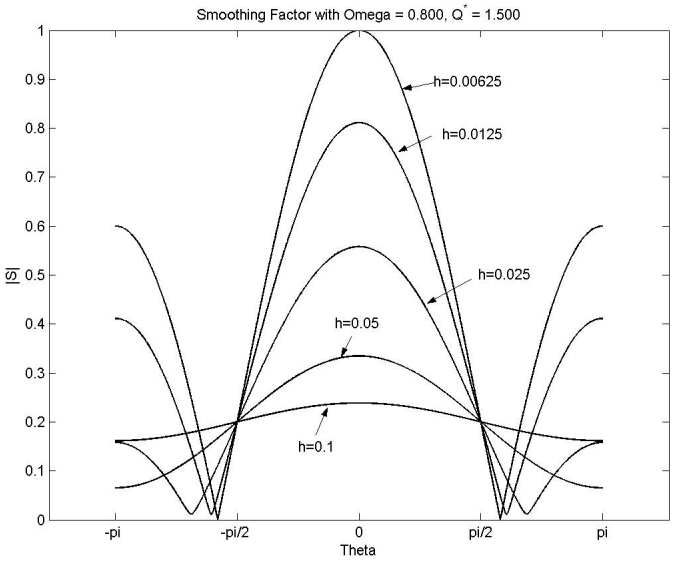

Figure 4.15: Amplification factor for the example HJB equation with $\omega = 0.8$, $Q^* = 1.5$ and different values of h

$\omega = 0.8$.

4.2.1.5 Smoothing Effect on High Frequency Artificial Error

To further illustrate the smoothing effect of the damped-relaxation smoother, we manually fix the initial error before smoothing to be a high frequency one, and compare the error after applying 1 and 2 iterations of smoothing to the approximate solution. Figure 4.16 and Figure 4.17 show the plots of errors before and after smoothing for different values of ω, x-axis represents the grid points and y-axis represents the value of the error. Initial errors for all cases are the same, grid size is set to 0.0125.

Comparing the magnitude of the error after the first and second smoothing iteration, it is obvious that $\omega = \frac{2}{3}$ gives the best smoothing effect among the four. $\omega = \frac{2}{3}$ reduces the high frequency error approximately in a factor of 3 per iteration, which is very efficient.

4.3 Smoothing Analysis for HJBI Equations

Similar to the HJB case, we assume the exact solution for HJBI problem at time step $n+1$ is $\bar{V}$ and the approximation solution after the k^{th} iteration is $\hat{V}^k = \bar{V} + \epsilon^k$, where ϵ^k

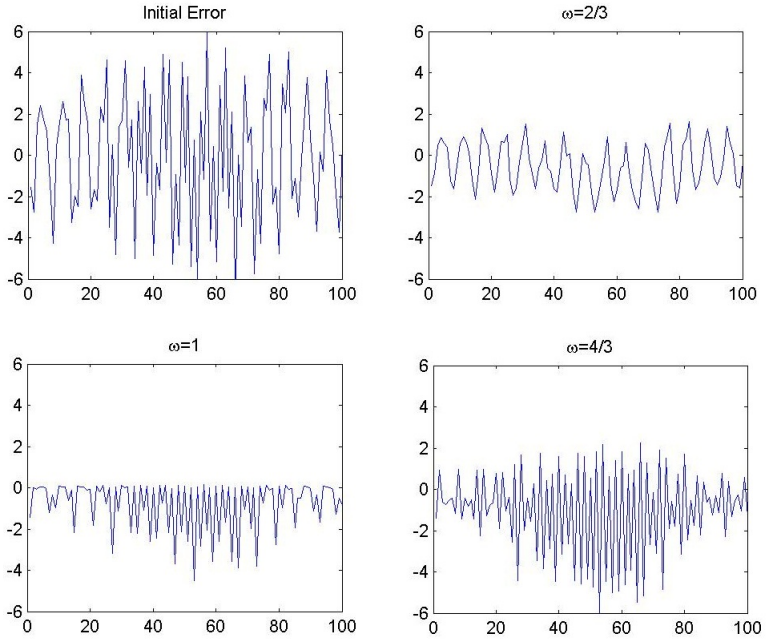

Figure 4.16: Initial error and the error after one smoothing iteration for the example HJB equation with different values of ω

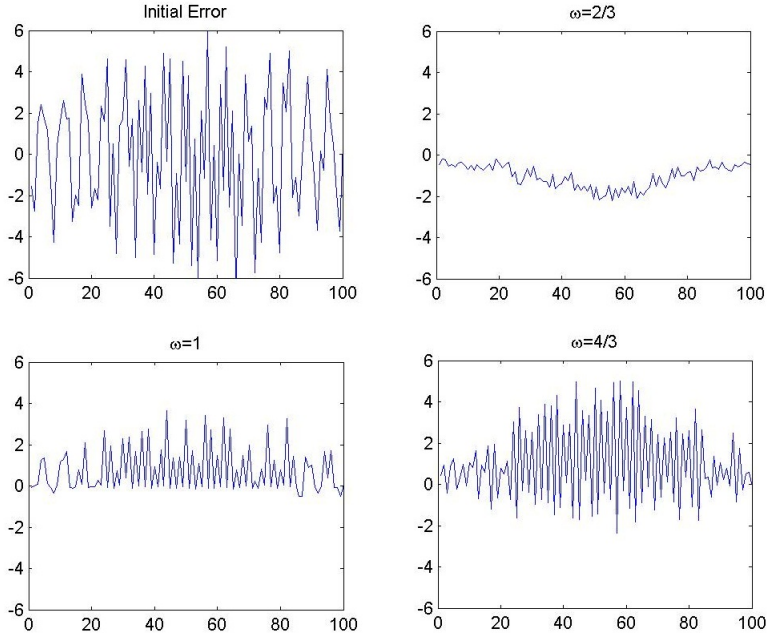

Figure 4.17: Initial error and the error after two smoothing iterations for the example HJB equation with different values of ω

is the error after the k^{th} smoothing iteration. Assume the optimal control for every grid point will not change from iteration to iteration. Applying a similar deduction procedure in Section 4.2 to HJBI case, we obtain

$$\widetilde{S_i^k}(\theta,\omega)=\frac{\omega\Delta\tau\left(\alpha_i^*+\beta_i^*\right)\cos\theta+(1-\omega)\left[1+\Delta\tau\left(\alpha_i^*+\beta_i^*+c_i^*\right)\right]+\mathbf{i}\omega\Delta\tau\left(\beta_i^*-\alpha_i^*\right)\sin\theta}{1+\Delta\tau\left(\alpha_i^*+\beta_i^*+c_i^*\right)}, \tag{4.8}$$

where

$$\alpha_i^*=\alpha_i^{n+1}\left(Q_i^k,\,P_i^k\right),\ \beta_i^*=\beta_i^{n+1}\left(Q_i^k,\,P_i^k\right),$$
$$c_i^*=c_i^{n+1}\left(Q_i^k,\,P_i^k\right),\ d_i^*=d_i^{n+1}\left(Q_i^k,\,P_i^k\right),$$

with

$$Q_i^k,\,P_i^k\in\arg\sup_{P\in\hat{P}}\inf_{Q\in\hat{Q}}\left\{\left(\mathcal{F}^{Q,P}\left(\bar{V}+\epsilon^k\right)\right)_i\right\}.$$

4.3.1 Smoothing Factors of the Example HJBI Equation

As in the HJB case, let $X=\log S$, example HJBI problem on log grid can be expressed as

$$\begin{aligned}V_\tau &= \sup_{P\in\hat{P}}\inf_{Q\in\hat{Q}}\left\{\frac{\sigma^2}{2}V_{XX}+\left[q_3q_1+(1-q_3)(r_l-r_f)-\frac{\sigma^2}{2}\right]V_X\right.\\ &\quad\left.-\left[q_3q_1+q_2(1-q_3)+\frac{\mu}{\eta}\right]V+\frac{\mu}{\eta}V^*\right\}.\end{aligned}$$

The coefficients for example HJBI problem on the log grid (1.8) are

$$\begin{aligned}a(\tau,Q,P) &= \frac{1}{2}\sigma^2,\ b(\tau,Q,P)=q_3q_1+(1-q_3)(r_l-r_f)-\frac{\sigma^2}{2},\\ c(\tau,Q,P) &= q_3q_1+(1-q_3)q_2+\frac{\mu}{\eta},\ d(\tau,Q,P)=\frac{\mu}{\eta}V^*.\end{aligned}$$

Substituting the above coefficients into (1.11), assuming central differencing and uniform

grid, we obtain

$$\alpha(Q,P) = \frac{2a}{2h^2} - \frac{b}{2h} = \frac{\sigma^2}{2h^2} - \frac{q_3 q_1 + (1-q_3)(r_l - r_f) - \frac{\sigma^2}{2}}{2h},$$

$$\beta(Q,P) = \frac{2a}{2h^2} + \frac{b}{2h} = \frac{\sigma^2}{2h^2} + \frac{q_3 q_1 + (1-q_3)(r_l - r_f) - \frac{\sigma^2}{2}}{2h},$$

where h is the grid size. The typical values for the parameters are $r_b = 0.05$, $r_l = 0.03$, $r_f = 0.004$, $\sigma^2 = 0.09$, $\Delta\tau = 0.01$ and $\eta = 10^{-6}\Delta\tau$. For the same reason as in HJB case, we will examine specific values of θ on all possible values of optimal controls.

4.3.1.1 Low Frequency Components:

Let $\theta \approx 0$. In this case, $\sin\theta \approx 0$ and $\cos\theta \approx 1$, (4.8) becomes

$$\begin{aligned}
\widetilde{S_i^k}(\theta,\omega) &\approx \frac{\omega\Delta\tau(\alpha_i^* + \beta_i^*) + (1-\omega)[1 + \Delta\tau(\alpha_i^* + \beta_i^* + c_i^*)]}{1 + \Delta\tau(\alpha_i^* + \beta_i^* + c_i^*)} \\
&\approx 1 - \omega + \frac{\omega\Delta\tau(\alpha_i^* + \beta_i^*)}{1 + \Delta\tau(\alpha_i^* + \beta_i^* + c_i^*)} \\
&\approx 1 - \omega + \frac{\omega\Delta\tau\frac{\sigma^2}{h^2}}{1 + \Delta\tau\left(\frac{\sigma^2}{h^2} + \frac{\mu}{\eta}\right) + \Delta\tau(q_3 q_1 + (1-q_3)q_2)}.
\end{aligned}$$

To narrow down the possible range of ω, we first consider the simplest expression of $\widetilde{S_i^k}(\theta,\omega)$. When $\mu = 1$, $\widetilde{S_i^k}(0,\omega)$ will be close to $1-\omega$ as $h \to 0$ due to the penalty term. Therefore to ensure convergence, it is required that $\omega \in (0,2)$ and ω close to 1 is preferred. When $\mu = 0$,

$$\begin{aligned}
\widetilde{S_i^k}(\theta,\omega) &\approx 1 - \omega + \frac{\omega\Delta\tau\frac{\sigma^2}{h^2}}{1 + \Delta\tau\frac{\sigma^2}{h^2} + \Delta\tau\cdot(q_3 q_1 + (1-q_3)q_2)} \\
&\to 1 \quad as\, h \to 0.
\end{aligned}$$

Figure 4.18 shows the relationship between $\left|\widetilde{S_i^k}(0,\omega)\right|$, ω and h when $\mu = 0$. We can see from the plot and the analysis that all $\omega \in (0,2)$ are acceptable since they all have

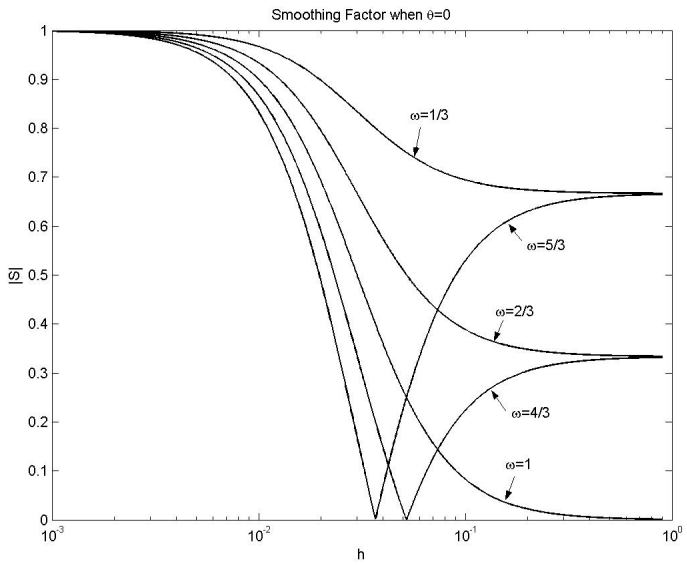

Figure 4.18: Amplification factor for the example HJBI equation with $\theta \approx 0$, $\mu = 0$ and different values of ω

$\left|\widetilde{S_i^k}(0,\omega)\right| < 1.$

4.3.1.2 Medium Frequency Components:

Let $\theta \approx \frac{\pi}{2}$. In this case, $\sin\theta \approx 1$ and $\cos\theta \approx 0$ and (4.8) becomes

$$\widetilde{S_i^k}(\theta,\omega) \approx \frac{\pm \mathbf{i}\omega\Delta\tau\left(\beta_i^* - \alpha_i^*\right) + (1-\omega)\left[1 + \Delta\tau\left(\alpha_i^* + \beta_i^* + c_i^*\right)\right]}{1 + \Delta\tau\left(\alpha_i^* + \beta_i^* + c_i^*\right)}.$$

As $\alpha_i(Q,P) + \beta_i(Q,P) + c_i(Q,P) > 0$ for all possible $Q \in \hat{Q}$ and $P \in \hat{P}$ and the given constants, it is easy to show that

$$\begin{aligned}\left|\widetilde{S_i^k}(\theta,\omega)\right| &\leq \frac{\omega\Delta\tau\left|\beta_i^* - \alpha_i^*\right| + |1-\omega|\left[1 + \Delta\tau\left(\alpha_i^* + \beta_i^* + c_i^*\right)\right]}{1 + \Delta\tau\left(\alpha_i^* + \beta_i^* + c_i^*\right)} \\ &= |1-\omega| + \omega\frac{\frac{\Delta\tau}{h}\left|q_3 q_1 + (1-q_3)(r_l - r_f) - \frac{\sigma^2}{2}\right|}{1 + \Delta\tau\left(\frac{\sigma^2}{h^2} + \frac{\mu}{\eta}\right) + \Delta\tau\left(q_3 q_1 + (1-q_3)q_2\right)}. \qquad (4.9)\end{aligned}$$

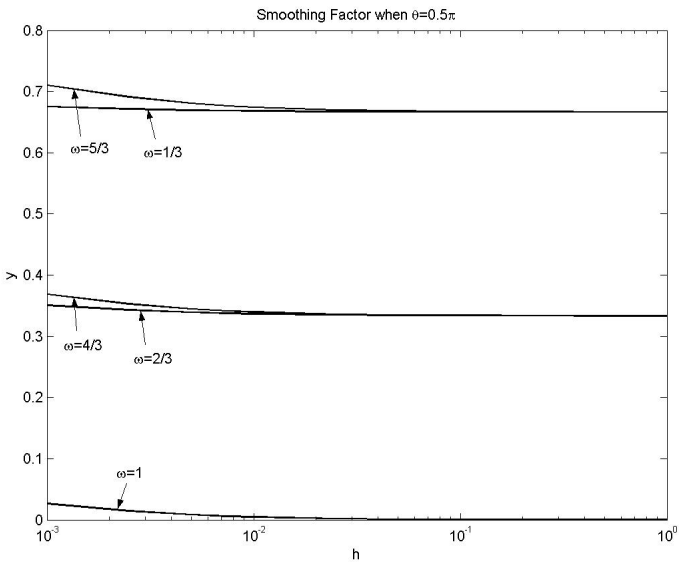

Figure 4.19: y^* for the example HJBI equation with $\mu = 0$, $\theta \approx \frac{\pi}{2}$ and different values of ω

Again, when $\mu = 1$, $\left|\widetilde{S_i^k}(\theta, \omega)\right| \to |1 - \omega|$, therefore $\omega \in (0, 2)$. When $\mu = 0$, let

$$y(Q, \mu = 0, \omega, h) = |1 - \omega| + \omega \frac{\frac{\Delta\tau}{h}\left|q_3 q_1 + (1 - q_3)(r_l - r_f) - \frac{\sigma^2}{2}\right|}{1 + \Delta\tau\left(\frac{\sigma^2}{h^2} + 0\right) + \Delta\tau(q_3 q_1 + (1 - q_3) q_2)}.$$

Substituting in all possible values of Q into $y(Q, \mu = 0, \omega, h)$, we obtain

$$\begin{aligned} y^* &= \max_{Q \in \hat{Q}} \{y(Q, \mu = 0, \omega, h)\} \\ &= |1 - \omega| + \frac{\frac{\Delta\tau}{h} \cdot 0.005}{1 + \Delta\tau\left(\frac{\sigma^2}{h^2} + 0.05\right)} \\ &\to |1 - \omega| \quad as\, h \to 0, \end{aligned}$$

which is an upper bound for $\left|\widetilde{S_i^k}(\theta, \omega)\right|$. Figure 4.19 shows the plot for y^* with different grid size and ω. Unlike the low frequency case, the grid size does not have much impact on the smoothing effect for medium frequency components. All $\omega \in (0, 2)$ have $\left|\widetilde{S_i^k}(\theta, \omega)\right| < 1$,

therefore all $\omega \in (0,2)$ are applicable for smoothers.

4.3.1.3 High Frequency Components:

Let $\theta \approx -\pi$. In this case, $\sin\theta \approx 0$ and $\cos\theta \approx -1$, (4.8) becomes

$$\begin{aligned}
\widetilde{S_i^k}(\theta,\omega) &\approx \frac{-\omega\Delta\tau\left(\alpha_i^* + \beta_i^*\right) + (1-\omega)\left[1 + \Delta\tau\left(\alpha_i^* + \beta_i^* + c_i^*\right)\right]}{1 + \Delta\tau\left(\alpha_i^* + \beta_i^* + c_i^*\right)} \\
&\approx 1 - \omega - \frac{\omega\Delta\tau\left(\alpha_i^* + \beta_i^*\right)}{1 + \Delta\tau\left(\alpha_i^* + \beta_i^* + c_i^*\right)} \\
&\approx 1 - \omega - \frac{\omega\Delta\tau\frac{\sigma^2}{h^2}}{1 + \Delta\tau\left(\frac{\sigma^2}{h^2} + \frac{\mu}{\eta}\right) + \Delta\tau\left(q_3 q_1 + (1-q_3)\,q_2\right)}.
\end{aligned}$$

Similarly, when $\mu = 1$, $\widetilde{S_i^k}(\theta,\omega) \to 1-\omega$. When $\mu = 0$,

$$\begin{aligned}
\widetilde{S_i^k}(\theta,\omega) &\approx 1 - \omega - \frac{\omega\Delta\tau\frac{\sigma^2}{h^2}}{1 + \Delta\tau\frac{\sigma^2}{h^2} + \Delta\tau\left(q_3 q_1 + (1-q_3)\,q_2\right)} \\
&\to 1 - 2\omega
\end{aligned}$$

Therefore ω has to be smaller than 1 to ensure the convergence of the smoother. Furthermore, $\left|\widetilde{S_i^k}(\theta,\omega)\right|$ is bounded by $\max\left(|1-\omega|, |1-2\omega|\right)$, which reaches its minimal $\frac{1}{3}$ when $\omega = \frac{2}{3}$. Also, Figure 4.20 shows that when $\omega > 1$, $\widetilde{S_i^k}(\theta,\omega)$ can be greater than 1, rendering a diverging iteration. When $\omega = 1$, $\widetilde{S_i^k}(\theta,\omega)$ is close to 1 for fine grids, which means the high frequency error will reduce in a very slow rate, and when $\omega < 1$, the smoothing factor is smaller than 1. $\omega = \frac{2}{3}$ has good performance for high frequency components with $\widetilde{S_i^k}\left(-\pi, \frac{2}{3}\right) \le \frac{1}{3}$ for all grids, which is consistent with the analysis.

Summarizing the smoothing effect of low, medium and high frequency components, we observed that ω has to be smaller than 1 for the smoother to converge. When $\mu = 0$, $\omega = \frac{2}{3}$ is optimal for high frequency components with $\left|\widetilde{S_i^k}\left(\theta, \frac{2}{3}\right)\right|$ bounded by $\frac{1}{3}$. $\omega = \frac{2}{3}$ is also eligible for medium and low frequency components. When $\mu = 1$, $\left|\widetilde{S_i^k}(\theta,\omega)\right|$ is close to $1-\omega$, so $\left|\widetilde{S_i^k}\left(\theta, \frac{2}{3}\right)\right|$ is close to $\frac{1}{3}$ for all three cases. From the analysis on specific

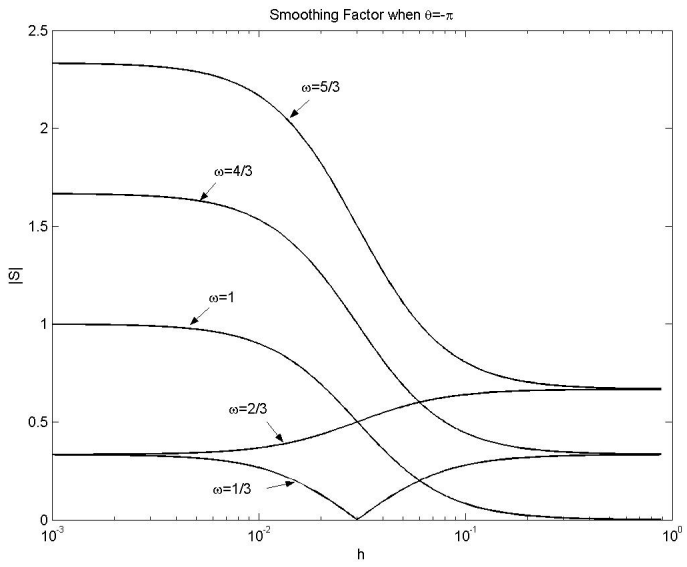

Figure 4.20: Amplification factor for the example HJBI equation with $\theta \approx -\pi$, $\mu = 0$ and different values of ω

frequencies, $\omega = \frac{2}{3}$ appears to be optimal for smoothers.

4.3.1.4 The Actual Smoothing Factor for the Example HJBI Equation:

To justify the conclusion about the optimal value for ω and the smoothing factor, we will plot $\left|\widetilde{S^k}(\theta, \omega)\right|$ for (4.8) on different frequencies, with different ω, grid size h and optimal controls.

First, we fix $h = 0.025$, which is a practical value for grid size. Figure 4.21 shows the plot of $\left|\widetilde{S^k}(\theta, \omega)\right|$ against θ for different frequencies with $\mu = 0$. We can see that $\omega = 1$ is optimal for low frequency components, $\omega = \frac{2}{3}$ is optimal for highest frequency components while $\omega = 0.8$ out performs $\omega = \frac{2}{3}$ except for the highest frequencies. Figure 4.22 shows $\left|\widetilde{S^k}(\theta, \omega)\right|$ with $\mu = 1$. When $\mu = 1$, $\omega \to 1$ is optimal, so $\omega = 0.8$, which gives a smoothing factor close to 0.2, is preferred to $\omega = \frac{2}{3}$ in this case. Both the plots agree with the previous conclusion, however, it is not clear whether $\omega = \frac{2}{3}$ or $\omega = 0.8$ is optimal.

Figure 4.23 and Figure 4.24 shows the relationship between $\left|\widetilde{S^k}(\theta, \omega)\right|$ and the grid size for $\omega = \frac{2}{3}$ and $\omega = 0.8$ respectively. Since when $\mu = 1$, $\left|\widetilde{S^k}(\theta, \omega)\right|$ is not affected by h much, we only show the $\mu = 0$ case. As the grid size decreases, $\left|\widetilde{S^k}(\theta, \omega)\right|$ of low

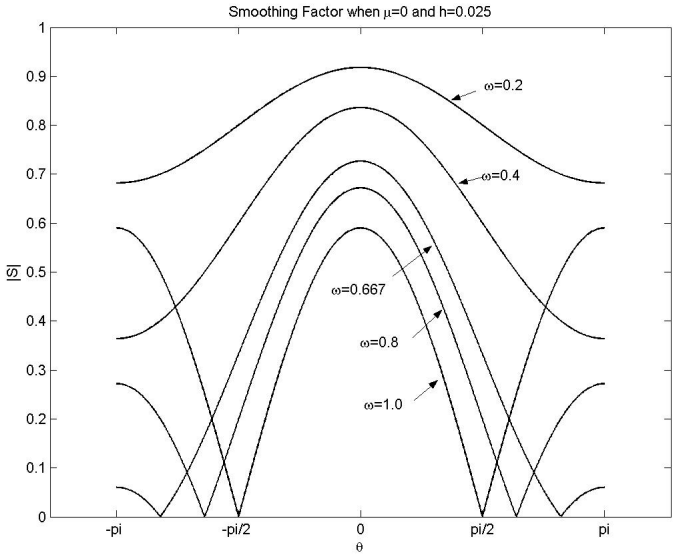

Figure 4.21: Amplification factor for the example HJBI equation with $\mu = 0$, $h = 0.025$ and different values of ω

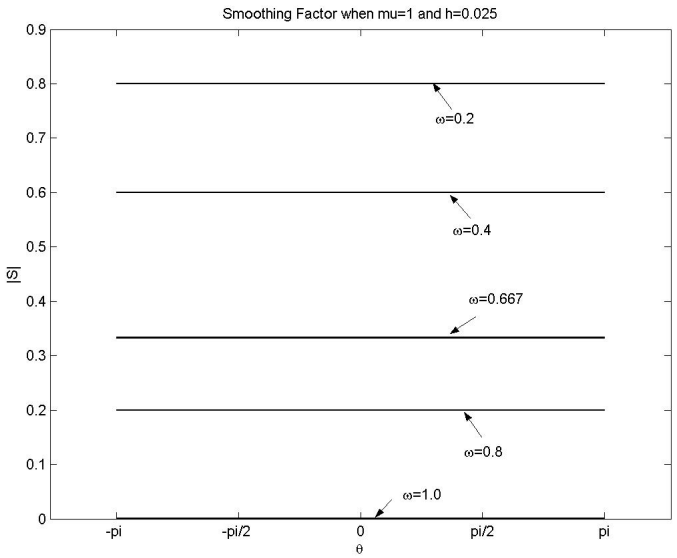

Figure 4.22: Amplification factor for the example HJBI equation with $\mu = 1$, $h = 0.025$ and different values of ω

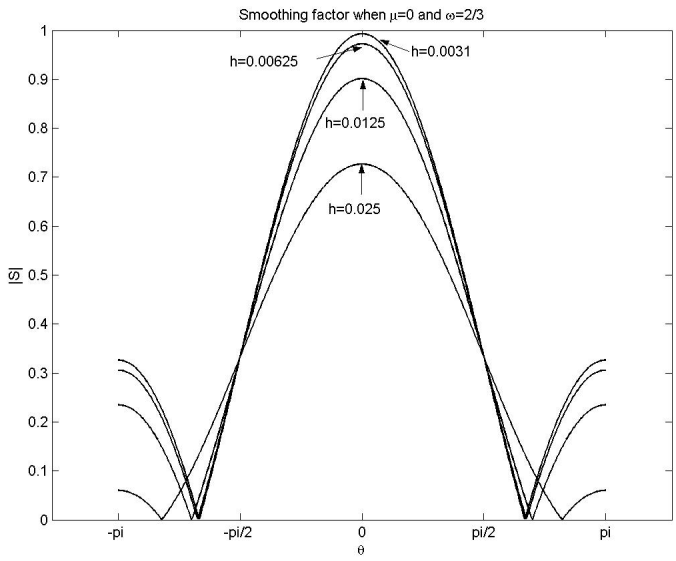

Figure 4.23: Amplification factor for the example HJBI equation with $\mu = 0$, $\omega = \frac{2}{3}$ and different values of h

frequency components moves up to 1 for both cases. However, for very high frequency components on fine grids, $\left|\widetilde{S^k}(\theta, 0.8)\right| \to 0.6$ while $\left|\widetilde{S^k}(\theta, \frac{2}{3})\right| \to \frac{1}{3}$. Considering the grids we will be using, $\omega = \frac{2}{3}$ is more suitable for smoothers.

4.3.1.5 Smoothing Effect on High Frequency Artificial Error

Similar to the HJB case, we manually fix the initial error before smoothing to a high frequency one, and compare the error after applying smoothing iterations to the approximate solution. Figure 4.25 and Figure 4.26 show the plots of errors before and after smoothing for different values of ω. Initial errors for all cases are the same, grid size is set to 0.0125.

Comparing the magnitude of the error after the first and second smoothing iteration, it is obvious that $\omega = \frac{2}{3}$ gives the best smoothing effect among the four and it is very efficient.

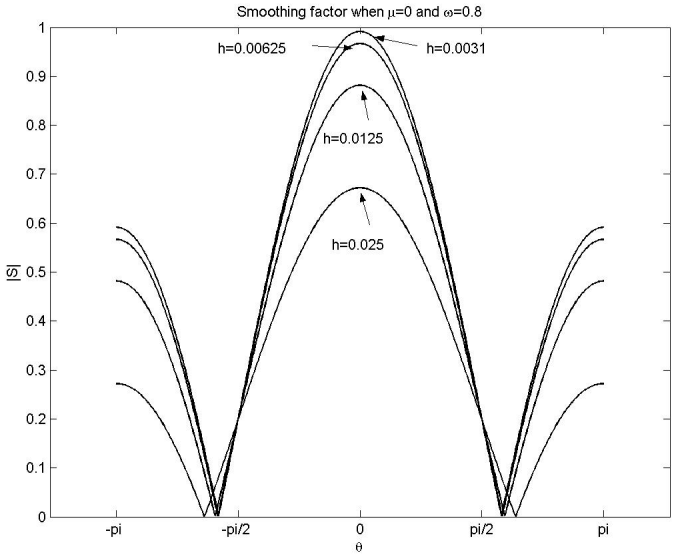

Figure 4.24: Amplification factor for the example HJBI equation with $\mu = 0$, $\omega = 0.8$ and different values of h

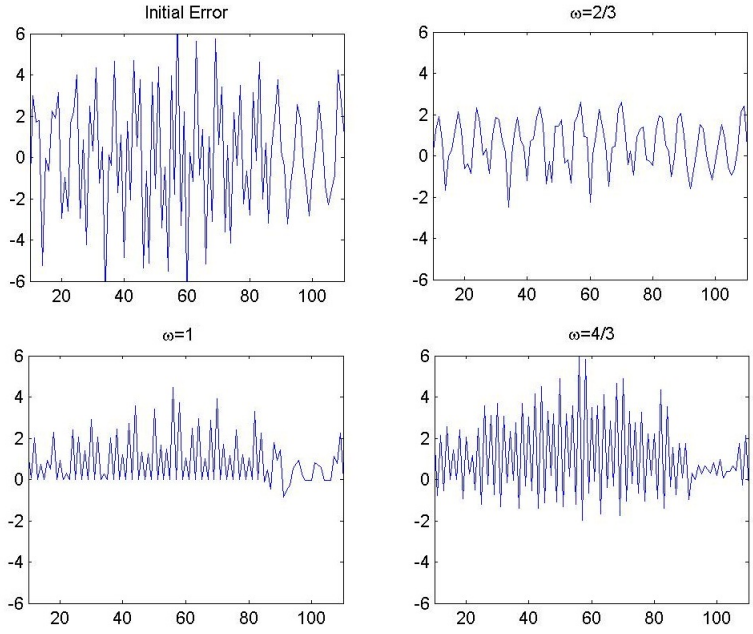

Figure 4.25: Initial error and the error after one smoothing iteration for the example HJBI equation with different values of ω

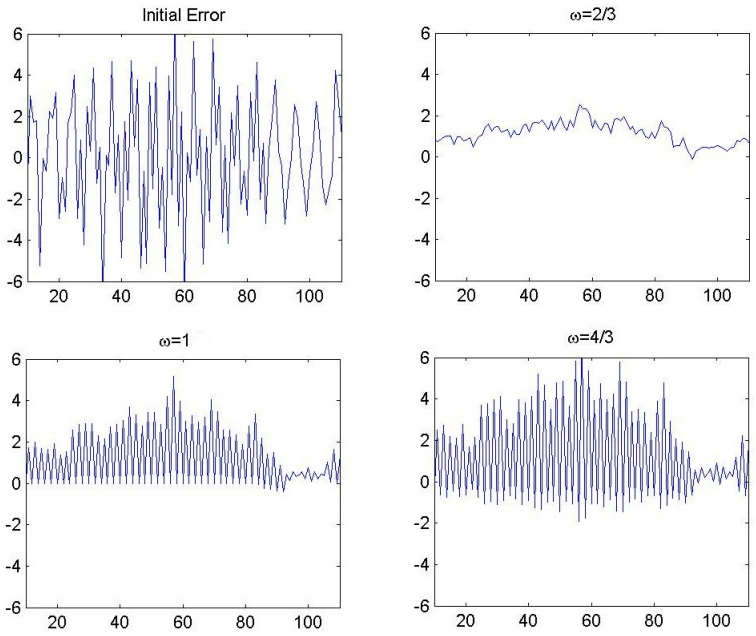

Figure 4.26: Initial error and the error after two smoothing iterations for the example HJBI equation with different values of ω

Chapter 5

Numerical Results

In this section, we will compare the convergence rates of different methods that we discussed in Chapter 1.3. Convergence study will be presented for both of the model problems arising in option pricing and some other representative nonlinear problems. For all the results in this chapter, the FAS scheme used is described in Section 3.2. Two pre- and post-soothing, V-cycle and linear intergrid transfer are applied. The stopping criteria is that the residual norm of the nonlinear problem is smaller than 10^{-6}.

5.1 Policy Iteration for HJB Equation: A Slow Example

An example in [27] shows that in general, the number of policy iteration steps cannot be bounded by a constant that is independent of the number of grid points. Consider an Markovian dynamic programming (MDP) problem which can be simplified to solving

$$V_i = \max\left\{V_{i-1} + f_i^1,\, V_{i+1} + f_i^2\right\},\ i = 0, \cdots, M,$$

where

$$\begin{aligned} f_0^1 = f_0^2 \quad &= \quad f_M^1 = f_M^2 = 0 \\ f_{M-1}^1 = -1 \quad &, \quad f_{M-1}^2 = 2M, \\ f_i^1 = -1 \quad &, \quad f_i^2 = -2, \end{aligned}$$

Grid points (M)	FAS iter.
128	2
256	2
512	2
1024	3

Table 5.1: Convergence of FAS scheme for MDP problem with different number of grid points

with $i = 1, \ldots, M-1$. With an initial guess $V_0 = 0$, the Newton-like policy iteration will correct the optimal control one by one, from grid $M-1$ to grid 1, rendering an $M-1$ iterations convergence. Table 5.1 shows the convergence of our FAS scheme for this MDP problem. On the coarsest level, we solve the nonlinear HJBI problem by applying the Newton-like policy iteration. It is clear that the number of FAS iterations is insensitive to the number of grid points.

5.2 HJB Example in [25]

We will consider an HJB example in [25]. In this example, the problem is defined as

$$\begin{cases} \max\limits_{1\leq v\leq 2} \left[A^v u\left(x,y\right) - f^v\left(x,y\right)\right] = 0, & x,y \in (0,1)\,, \\ u\left(x,y\right) = 0, & x,y \in \{0,1\}\,, \end{cases}$$

where

$$A^1 = -\frac{\partial^2}{\partial x^2} - 0.5\frac{\partial^2}{\partial x \partial y} - \frac{\partial^2}{\partial y^2},\ A^2 = -0.5\frac{\partial^2}{\partial x^2} - 0.1\frac{\partial^2}{\partial x \partial y} - \frac{\partial^2}{\partial y^2}$$

and

$$f^1 = f^2 = \max\left(A^1\bar{u},\ A^2\bar{u}\right),\ \bar{u} = x\left(1-x\right)y\left(1-y\right).$$

$\bar{u}$ turns out to be the exact solution of the corresponding HJB equation. We apply a uniform discretization of the second order derivatives

$$\begin{aligned} \frac{\partial^2}{\partial x^2} &\approx h^{-2} D^+_{h,x} D^-_{h,x},\ \frac{\partial^2}{\partial y^2} \approx h^{-2} D^+_{h,y} D^-_{h,y}, \\ \frac{\partial^2}{\partial x \partial y} &\approx \frac{1}{2} h^{-2}\left[D^+_{h,x} D^+_{h,y} + D^-_{h,x} D^-_{h,y}\right], \end{aligned}$$

Grid size (h)	Level	FAS iter.	Rate of conv.
$\frac{1}{8}$	2	5	0.02236
$\frac{1}{16}$	3	6	0.03914
$\frac{1}{32}$	4	7	0.04366
$\frac{1}{64}$	5	7	0.05039

Table 5.2: Convergence FAS scheme on the HJB example problem in [25]

where $D^{\pm}_{h,x}$ and $D^{\pm}_{h,y}$ denote the forward and backward difference in x and y respectively. We solve this HJB problem using our FAS scheme in Section 3.2. We note that the discretization method we used here may not result in an M matrix, thus the relaxation scheme is not guaranteed to converge. However, since the relaxation scheme converges for this particular example, we will use it for simplicity.

Table 5.2 shows the number of levels, the number of FAS iterations in one time step and the rate of convergence for different grid sizes. On the coarsest level, we solve the nonlinear HJB problem by applying policy iteration. We can observe that the convergence of FAS scheme is insensitive to the grid size. The rate of convergence presented in the last column is computed by averaging $\frac{\|r^{k+1}\|}{\|r^k\|}$ over all the iterations, where r^k is the residual of the HJB equation after the k^{th} FAS iteration. Figure 1 and Figure 2 in [25] show that the convergence rate of MGHJB is approximately 0.7 for this example problem. However as shown in the Table 5.2, all the convergence rates for different grid sizes and number of levels are smaller than 0.1 when applying our FAS scheme.

5.3 Pension Plan Asset Allocation Problem

Parameters in Table 5.3 are used for the HJB numerical tests. The grid size of the control set $\hat{Q}$ is 0.1 and the time step $\Delta\tau = 0.01$. We will show the numerical results for three methods we have discussed: policy iteration with multigrid, relaxation scheme and FAS scheme. Since the convergence is similar in each time step, we will only show the data for the first time step.

5.3.1 Policy Iterations with Multigrid

Table 5.4 presents the convergence result for policy iterations with multigrid, which

r	0.03
σ	0.15
T	20 years
ξ	0.33
π	0.1
γ	9.125
$tolerance$	10^{-6}
$\Delta\tau$	0.01
$\hat{Q}$	[0, 1.5]

Table 5.3: Parameters used in the HJB example

Grid size (h)	Nonlinear iterations	Multigrid iterations per nonlinear iteration		
		1^{st} policy	2^{nd} policy	3^{rd} policy
0.02	2	2	1	N/A
0.01	2	3	1	N/A
0.005	3	4	3	1
0.0025	3	6	4	1

Table 5.4: Convergence result of policy iterations with multigrid for the example HJB equation

is described in Section 3.1. The multigrid method used here is standard. Gauss-Siedel smoother is applied for two pre- and two post-smoothings. Linear intergrid transfer is applied. On the coarsest grid, the linear problem is solved directly. Column 2 shows the number of nonlinear policy iterations required for one time step with different values of grid size. Column 3-5 show the number of multigrid iterations required for solving the linear problem in each nonlinear policy iteration stated in the second column.

5.3.2 Relaxation Scheme and FAS Scheme

In this section we will compare the convergence of the relaxation scheme and the FAS scheme for the model HJB problem. We will first look at the HJB problem on a uniform grid and then, as the smoothing analysis in Section 4 is based on a uniform log grid, we will compare the two schemes on the log grid too. Since the control is smooth for this HJB problem, the standard FAS scheme is enough to generate satisfactory convergence. On the coarsest grid , the HJB is solved by the relaxation scheme.

Table 5.5 shows the convergence for relaxation scheme and FAS scheme on the uniform grid. The convergence rate of relaxation scheme becomes unacceptably slow as grid size

h	Relax. iter.	FAS iter.
0.02	90	4
0.01	330	5
0.005	1320	6
0.0025	≈ 4000	8

Table 5.5: Convergence of the relaxation scheme and FAS scheme for the example HJB equation on uniform grid with $\omega = \frac{2}{3}$

h	Relax. iter.	FAS iter.
0.05	6	5
0.025	17	5
0.0125	50	6
0.00625	151	7

Table 5.6: Convergence of the relaxation scheme and FAS scheme for the example HJB equation on uniform log grid with $\omega = \frac{2}{3}$

decreases, which verifies the conclusion in Section 1.3.2. On the other hand, the number of FAS iterations per time step is insensitive to the grid size. Compared to Table 5.4, FAS is slightly more efficient than policy iteration with multigrid method in terms of total number of multigrid iterations for this example. However, using either of them would yield satisfactory convergence.

We run the same comparison again on the log grid and obtain Table 5.6. For the log grid case, the difference between the efficiency of the two methods is smaller. However, we can still see the trend: the number of relaxation iterations triples when the grid size is reduced by half while the number of FAS iterations is insensitive to grid size.

5.4 Policy Iteration for HJBI Equation: Counter Example I

A pathological example where Newton-like policy iteration does not converge for HJBI problems is shown in [32]. We will solve the counter example with policy iteration and relaxation scheme respectively and compare their convergence. The counter example is a discounted two-person zero-sum stochastic game, which can be simplified to solving

$$V = \max_{l_1 \in L_1} \min_{l_2 \in L_2} \left\{ R(l_1, l_2) + \frac{3}{4} P(l_1, l_2) \cdot V \right\}, \tag{5.1}$$

Algorithm 5.1 Policy Iteration for Counter Example I

1: Let $\hat{V}^0 = 0$
2: **for** $k = 0, 1, 2...$ until convergence **do**
3: Compute $l_1^k, l_2^k \in \arg\max_{l_1 \in L_1} \min_{l_2 \in L_2} \left\{ R(l_1, l_2) + \frac{3}{4} P(l_1, l_2) \cdot \hat{V}^k \right\}$
4: Solve $\hat{V}^{k+1} = R\left(l_1^k, l_2^k\right) + \frac{3}{4} P\left(l_1^k, l_2^k\right) \cdot \hat{V}^{k+1}$

where $L_1 = \{1, 2\}$, $L_2 = \{1, 2\}$ and $R = \begin{bmatrix} 3 & 6 \\ 2 & 1 \end{bmatrix}$, $P = \begin{bmatrix} 1 & \frac{1}{3} \\ 1 & 1 \end{bmatrix}$. The exact solution for this problem is $V = 8$. Since we are solving (5.1) by iterative methods, we will let the approximate solution after the k^{th} iteration be $\hat{V}^k$ and assume $\hat{V}^0 = 0$.

A policy iteration scheme for solving (5.1) is shown in Algorithm 5.1.

On the other hand, the update rule for the relaxation scheme is

$$\hat{V}^{k+1} = \max_{l_1 \in L_1} \min_{l_2 \in L_2} \left\{ R(l_1, l_2) + \frac{3}{4} P(l_1, l_2) \cdot \hat{V}^k \right\}.$$

The relaxation scheme will converge in 13 iterations for this problem while the solution of policy iteration will bounce between two values and never converges. This example shows that Newton-like policy iteration does not guarantee global convergence for HJBI problems.

5.5 Policy Iteration for HJBI Equation: Counter Example II

Another counter example shows that Newton-like policy iteration method does not converge is presented in [11]. We will solve this counter example by Newton-like policy iteration method, relaxation scheme and our FAS scheme and compare their convergence. Consider the following discrete double-obstacle problem: find $V^* = (U_i)_{1 \le i \le N} \in \mathbb{R}^N$ such

that

$$\begin{cases} \max\left(\min\left(-\frac{U_{i-1}-2U_i+U_{i+1}}{\Delta s^2},\, \gamma\left(U_i - g\left(s_i\right)\right)\right),\, \gamma\left(U_i - h\left(s_i\right)\right)\right) = 0, \quad i = 1, \ldots, N, \\ U_0 = 1,\ U_{N+1} = 0.8, \end{cases}$$

where $\Delta s = \frac{1}{N+1}$, $s_i = i\Delta s$, $g\left(s\right) = \max\left(0,\, 1.2 - \left(\left(s - 0.6\right)/0.1\right)^2\right)$, and $h\left(s\right) = \min\left(2,\, 0.3 + \left(\left(s - 0.2\right)/0.1\right)^2\right)$, with $\gamma = 1000$, $N = 127$ and a starting point V^0 such that $V^0 \notin (g, h)$.The Newton-like policy iteration used here is close to the one in the previous section and it does not converge. An iterative method that converges in 16 iterations for this problem is proposed in [11]. However, it involves solving an $N \times N$ linear system for 95 times in total. The relaxation iteration for this problem is

$$\hat{V}_i^{k+1} = -\max_{P\in\{1,2\}}\min_{Q\in\{1,2\}}\left\{\frac{a_i\left(P,Q\right)\cdot\left(\hat{V}_{i-1}^k + \hat{V}_{i+1}^k\right) + c_i\left(P,Q\right)}{b_i\left(P,Q\right)}\right\},$$

where $a_i = \begin{bmatrix} -\frac{1}{\Delta s^2} & 0 \\ 0 & 0 \end{bmatrix}$, $b_i = \begin{bmatrix} \frac{2}{\Delta s^2} & \gamma \\ \gamma & \gamma \end{bmatrix}$ and $c_i = \begin{bmatrix} 0 & -\gamma g\left(s_i\right) \\ -\gamma h\left(s_i\right) & -\gamma h\left(s_i\right) \end{bmatrix}$. While convergence is guaranteed, it will take 9457 iterations for the relaxation scheme to converge. On the other hand, our FAS scheme with 3 damped-relaxation pre- and post-smoothing will converge in 13 iterations.

5.6 American Option and Stock Borrowing Fees

Parameters in Table 5.7 are used for HJBI numerical tests. The time step is chosen to be $\Delta\tau = 0.01$. We will show the numerical results for relaxation scheme and FAS scheme.

Similar to Section 5.3.2, we will first look at the HJBI problem on a uniform grid and then on a uniform log grid. The FAS introduced in Section 3.5 has to be applied to get convergence due to the penalty term η. On the coarsest grid, the HJBI equation is solved by the relaxation scheme. As shown in Table 5.8, the relaxation scheme converges very slowly on fine grid and the number of FAS iterations per time step is insensitive to the grid size. In Column 4, the smallest value for m is presented. Parameter m is defined

r_b	0.05
r_l	0.03
r_f	0.004
μ	$\{0, 1\}$
σ	0.3
T	1.0 year
K	100
$\Delta\tau$	0.01
$tolerance$	10^{-6}
penalty term δ	$10^{-6}\Delta\tau$

Table 5.7: Parameters used in the HJBI example

h	Relax. iter.	FAS iter	m
$10^{-2}K$	319	6	1
$5 \times 10^{-3}K$	610	7	3
$2.5 \times 10^{-3}K$	2500	7	3
$1.25 \times 10^{-3}K$	≈ 10000	8	3

Table 5.8: Convergence of the relaxation scheme and FAS scheme for the example HJBI equation on uniform grid with $\omega = \frac{2}{3}$

in Section 3.5.2. It is a predetermined small positive integer and $2m + 1$ is the size of the local problem we solve to find the correct fine grid control at jump. Here we only list out the smallest possible value for m that will result a convergent scheme. m that is larger than the values presented in the table will also guarantee convergence, but the convergence rate will not improve much.

For HJBI problem on the uniform log grid, Table 5.9 provides the comparison between relaxation scheme and FAS scheme. When the grid size is halved, the number of relaxation iterations per time step becomes four times larger while the number of FAS iterations only increases 1. It is obvious that FAS scheme is more efficient than relaxation scheme for the

h	Relax. iter.	FAS iter.	m
0.025	26	5	2
0.0125	86	6	2
0.00625	340	7	3
0.00313	1350	8	3

Table 5.9: Convergence of the relaxation scheme and FAS scheme for the example HJBI equation on the uniform log grid with $\omega = \frac{2}{3}$

example HJBI problem on the log grid.

5.7 HJBI Example: Pursuit Game

The pursuit game example [17] can be simplified to solving the stationary HJBI equation

$$-\rho + \epsilon \Delta V + \max_{\alpha \in A} (\alpha \cdot \nabla V) + \min_{\beta \in B} (\beta \cdot \nabla V) + \|x\|_2^2 = 0$$

on $(-0.5, 0.5)^2$ with $\epsilon = 0.5$, Neumann boundary conditions and $A = \{(a_1, a_2) \,|a_i = \pm 1, 0\}$, $B = \{(0,0), (1,2), (2,1)\}$. Having ρ set to constant 0.194, we solve the problem using FAS scheme with two iterations of damped-relaxation pre- and two post-smoothing. Linear intergrid transfer is applied, and on the coarsest grid of the V-cycle, we solve the HJBI problem by applying relaxation scheme. For fine grid size of 2^{-4}, 2^{-5}, 2^{-6} and 2^{-7}, the FAS scheme will converge in 3 iterations for all four cases.

Chapter 6

Conclusions

In this thesis, we propose to solve the discretized HJB and HJBI equations by applying FAS with damped-relaxation smoother. We review some variations of policy iteration and their convergence and show that unlike policy iteration, the relaxation scheme is convergent for both HJB and HJBI equations. A damped-relaxation smoother is proposed. When the damping factor is appropriately chosen, the high frequency error will be damped away. Smoothing analysis based on two financial problems shows the efficiency of the smoother. Special restriction and interpolation techniques have been developed to handle the case when there are jumps in the optimal control. Our variation of FAS is applied to eight different example problems including real life problems and examples in which policy iteration will not converge or converges slowly. Numerical results show that our FAS converges in small numbers of iterations for those examples.

Bibliography

[1] M. Akian, P. Sequier, and A. Sulem. A finite horizon multidimensional portfolio selection problem with singular transactions. In *Decision and Control, 1995., Proceedings of the 34th IEEE Conference on*, volume 3, pages 2193 –2198 vol.3, December 1995. 24

[2] Marianne Akian, J.L. Menaldi, and Agnes Sulem. On the investment-consumption model with transaction costs. Research Report RR-1926, INRIA, 1993. Projet META2. 24, 25

[3] Marianne Akian, Jose Luis Menaldi, and Agnes Sulem. Multi-asset portfolio selection problem with transaction costs. *Mathematics and Computers in Simulation*, 38(1-3):163 – 172, 1995. 24

[4] Marianne Akian, Agnes Sulem, and Michael I. Taksar. Dynamic optimization of long-term growth rate for a portfolio with transaction costs and logarithmic utility. *Mathematical Finance*, 11(2):153–188, 2001. 24

[5] Guy Barles and Espen R. Jakobsen. Error bounds for monotone approximation schemes for Hamilton-Jacobi-Bellman equations. *SIAM J. Numer. Anal.*, 43:540–558, February 2005. 1, 9

[6] Christophe Barrera-Esteve, Florent Bergeret, Charles Dossal, Emmanuel Gobet, Asma Meziou, Remi Munos, and Damien Reboul-Salze. Numerical Methods for the Pricing of Swing Options: A Stochastic Control Approach. *Methodology and Computing in Applied Probability*, 8:517–540, 2006. 10.1007/s11009-006-0427-8. 1

[7] R. Bellman. On the Theory of Dynamic Programming. *Proceedings of the National Academy of Science*, 38:716–719, August 1952. 1

[8] R.E. Bellman. *Introduction to the Mathematical Theory of Control Processes: Nonlinear processes*. Mathematics in science and engineering. Academic Press, 1971. 8

[9] A. Bensoussan. *Stochastic control by functional analysis methods*. Studies in mathematics and its applications. North-Holland Pub. Co., 1982. 1

[10] Tomasz R. Bielecki, Hanqing Jin, Stanley R. Pliska, and Xun Yu Zhou. Continuous-time mean-variance portfolio selection with bankruptcy prohibition. *Mathematical Finance*, 15(2):213–244, 2005. 2

[11] Olivier Bokanowski, Stefania Maroso, and Hasnaa Zidani. Some convergence results for Howard's algorithm. *SIAM Journal on Numerical Analysis*, 47(4):3001–3026, 2009. 9, 68, 69

[12] Achi Brandt. Multi-level adaptive technique (MLAT) for fast numerical solution to boundary value problems. In Henri Cabannes and Roger Temam, editors, *Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics*, volume 18 of *Lecture Notes in Physics*, pages 82–89. Springer Berlin / Heidelberg, 1973. 24

[13] Michele Breton and Pierre L'Ecuyer. Approximate solutions to continuous stochastic games. In Raimo Hamalainen and Harri Ehtamo, editors, *Differential Games - Developments in Modelling and Computation*, volume 156 of *Lecture Notes in Control and Information Sciences*, pages 257–264. Springer Berlin / Heidelberg, 1991. 9

[14] Amarjit Budhiraja and Kevin Ross. Convergent Numerical Scheme for Singular Stochastic Control with State Constraints in a Portfolio Selection Problem. *SIAM J. Control Optim.*, 45:2169–2206, January 2007. 1

[15] Michael G. Crandall, Hitoshi Ishii, and Pierre-Louis Lions. User's Guide to Viscosity Solutions of Second Order Partial Differential Equations. 1992. 1

[16] Shi-Jie Deng and Shmuel S. Oren. Incorporating operational characteristics and start-up costs in option-based valuation of power generation capacity. *Probab. Eng. Inf. Sci.*, 17:155–181, April 2003. 1

[17] Sylvie Detournay and Marianne Akian. Presentation at Copper Mountain Multigrid Cconference. March 28-April 1, 2011, Denver, Colorado. 24, 71

[18] Wendell H. Fleming. *Deterministic and Stochastics Optimal Control*. Springer-Verlag, 1975. 1

[19] P. A. Forsyth and G. Labahn. Numerical methods for controlled Hamilton-Jacobi-Bellman PDEs in finance. Technical report, 2006. 1, 6, 7, 11

[20] Ronald H. W. Hoppe. Multi-grid methods for Hamilton-Jacobi-Bellman equations. *Numerische Mathematik*, 49:239–254, 1986. vii, 8, 24, 27

[21] R.A. Howard. Dynamic programming and Markov process. Technical report, Cambridge, MA, USA, 1960. 8

[22] N.V. Krylov and A.B. Aries. *Controlled Diffusion Processes*. Applications of Mathematics. Springer, 2008. 1

[23] H.J. Kushner and P. Dupuis. *Numerical methods for stochastic control problems in continuous time*. Applications of mathematics. Springer, 2001. 1, 2, 9

[24] B. Lions, P.-L.and Mercier. Approximation numerique des equations Hamilton-Jacobi-Bellman. *RAIRO - Analyse numerique*, 14(4):369–393, 1980. 8, 24

[25] Ronald H. W.Hoppe Martina Bloss. Numerical computation of the value function of optimally controlled stochastic switching processes by multi-grid techniques. *Numerical Functional Analysis and Optimization*, 10(3):275–304, 1989. viii, ix, 24, 64, 65

[26] Martin L. Puterman and Shelby L. Brumelle. On the convergence of policy iteration in stationary dynamic programming. *Mathmatics of Operations Research*, 4(1):60–69, 1979. 8

[27] Manuel S. Santos and John Rust. Convergence properties of policy iteration. *SIAM Journal on Control and Optimization*, 42(6):2094–2115, 2004. 1, 8, 63

[28] Matt Thompson, Matt Davison, and Henning Rasmussen. Natural gas storage valuation and optimization: A real options application. *Naval Research Logistics (NRL)*, 56(3):226–238, 2009. 1

[29] Boleslaw Tolwinski. Newton-type methods for stochastic games. In Tamer Basar and Pierre Bernhard, editors, *Differential Games and Applications*, volume 119 of *Lecture Notes in Control and Information Sciences*, pages 128–144. Springer Berlin / Heidelberg, 1989. 9

[30] Boleslaw Tolwinski. Solving dynamic games via Markov game approximations. In Raimo Hamalainen and Harri Ehtamo, editors, *Differential Games - Developments in Modelling and Computation*, volume 156 of *Lecture Notes in Control and Information Sciences*, pages 265–274. Springer Berlin / Heidelberg, 1991. 9

[31] U. Trottenberg, C.W. Oosterlee, and A. Schuller. *Multigrid*. Academic Press, 2001. 11, 18, 19, 36

[32] J. Wal. Discounted Markov games: Generalized policy iteration method. *Journal of Optimization Theory and Applications*, 25:125–138, 1978. 1, 5, 9, 67

[33] J. Wang and P.A. Forsyth. Numerical solution of the Hamilton-Jacobi-Bellman formulation for continuous time mean variance asset allocation. *Journal of Economic Dynamics and Control*, 34(2):207 – 230, 2010. 1, 3, 6, 7

[34] H. Windcliff, J. Wang, P.A. Forsyth, and K.R. Vetzal. Hedging with a correlated asset: Solution of a nonlinear pricing pde. *Journal of Computational and Applied Mathematics*, 200(1):86 – 115, 2007. 1

[35] G.Labahn Y.Huang, P.A. Forsyth. Inexact Arithmetic Considerations for Direct Control and Penalty Methods: American Options under Jump Diffusion. 2011. 5

[36] X. Y. Zhou and D. Li. Continuous-time mean-variance portfolio selection: A stochastic LQ framework. *Applied Mathematics and Optimization*, 42:19–33, 2000. 2