Numerical Methods for Pricing a Guaranteed Minimum Withdrawal Benefit (GMWB) as a Singular Control Problem

by

Yiqing Huang

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
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

Guaranteed Minimum Withdrawal Benefits (GMWB) have become popular riders on variable annuities. The pricing of a GMWB contract was originally formulated as a singular stochastic control problem which results in a Hamilton Jacobi Bellman (HJB) Variational Inequality (VI). A penalty method method can then be used to solve the HJB VI. We present a rigorous proof of convergence of the penalty method to the viscosity solution of the HJB VI assuming the underlying asset follows a Geometric Brownian Motion. A direct control method is an alternative formulation for the HJB VI. We also extend the HJB VI to the case of where the underlying asset follows a Poisson jump diffusion.

The HJB VI is normally solved numerically by an implicit method, which gives rise to highly nonlinear discretized algebraic equations. The classic policy iteration approach works well for the Geometric Brownian Motion case. However it is not efficient in some circumstances such as when the underlying asset follows a Poisson jump diffusion process. We develop a combined fixed point policy iteration scheme which significantly increases the efficiency of solving the discretized equations. Sufficient conditions to ensure the convergence of the combined fixed point policy iteration scheme are derived both for the penalty method and direct control method.

The GMWB formulated as a singular control problem has a special structure which results in a block matrix fixed point policy iteration converging about one order of magnitude faster than a full matrix fixed point policy iteration. Sufficient conditions for convergence of the block matrix fixed point policy iteration are derived. Estimates for bounds on the penalty parameter (penalty method) and scaling parameter (direct control method) are obtained so that convergence of the iteration can be expected in the presence of round-off error.

Acknowledgments

I would like to extend my sincere gratitude to Professor Peter A. Forsyth at the David Cheriton School of Computer Science, University of Waterloo. I have benefited immensely from his patient guidance and creative ideas, without which this thesis will not be possible. Professor Forsyth's passion with computational finance research and way of thinking influence me as well as many other students. I am deeply grateful for his trust in my desire and potential of completing this thesis part-time. Thank you Professor Forsyth for the continuous encouragement, support and numerous research discussions to help me walk along this journey.

I would like to thank my thesis committee members Professors Min Dai, Yuying Li, Ken R. Vetzal, and Justin W.L. Wan for taking time to read my thesis and provide constructive and valuable feedback.

I would also like to thank Professor George Labahn for various helpful discussions in matrix computation. He reviewed our joint work meticulously and always passes encouraging words every time we meet. I am also grateful to Professor Ken R. Vetzal, who has always been willing to provide valuable guide in Finance research area. A special thank goes to Professors Yuying Li and Justin W.L. Wan, who gave me valuable suggestions and comments on the thesis proposal.

I gratefully acknowledge my employer, the Department of Electrical and Computer Engineering in the Faculty of Engineering, University of Waterloo for supporting my part-time Ph.D. study. Working in the ECE department undergraduate lab is very pleasant. I owe a big thank you to all the faculty, staff and students in ECE for creating an inspirational learning and working environment, where I work and study with a delightful heart.

Finally, my thanks go out to my family, especially my parents who have provided strong and loving support continuously. Thank my mother for taking such a good care of my child so that I can work and study without worries at the back of my head. Thank my father for sharing the load of various house chores to free me up for studying. Thank my sister for various conversations where we shared laughters and tears. Thank my husband for being understanding and supportive, taking the kid out for fun during many weekends and holidays while I was writing. My thanks also go out to my parents in law for their generous help. Last but not least, thank to my child, who has been such a bright star in my life and has kept unveiling more and more truth of life.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this chapter we give an overview of the scope of this thesis followed by our major contributions. Then we outline the organization of the thesis.

1.1 Overview

The main purpose of this thesis is to study numerical methods for pricing a Guaranteed Minimum Withdrawal Benefit (GMWB), a popular rider on variable annuities, as a singular control problem. In this section, we first introduce background information about GMWB contracts. Then we discuss the previous research on pricing GMWBs with a focus on the singular control formulation of the pricing problem. This results in a Hamilton Jacobi Bellman (HJB) Variational Inequality (VI). The HJB VI under investigation is a non-linear partial differential equation (PDE) for the case where the underlying asset follows a standard Geometric Brownian Motion and a non-linear partial integro-differential equation (PIDE) for the case where the underlying asset follows a Poisson jump diffusion process [39, 15]. Finally we discuss iterative methods for solving the resulting non-linear system of discretized algebraic equations.

1.1.1 The Guaranteed Minimum Withdrawal Benefit (GMWB)

It is conventional wisdom that the long term investor is better off investing in equities as opposed to risk free bonds, hence the advice to retirees to invest a significant portion of their savings in equities. However, as discussed in [40], investing in equities can be very risky, once retirees begin to draw down their savings. This is because the order of random returns in this case becomes significant. Losses during the early years of retirement, coupled with withdrawals, will have a very different end result compared with losses which occur during the later years of retirement.

In order to mitigate this risk, insurance companies have developed guaranteed minimum withdrawal benefit (GMWB) guarantees. This contract consists of a lump sum payment to an insurance company. This initial sum is invested in risky assets. The holder can withdraw a specified amount each year of the contract, regardless of the performance of the risky asset. The holder can also withdraw more than the contract amount, subject to a penalty. At expiry of the contract, the holder is entitled to the value of the investment amount remaining. This contract allows the holder to participate in market gains, while providing a certain minimum cash flow. In return for providing this guarantee, the insurance company receives a proportional fee. Pricing and hedging these contracts is a problem of much practical interest. For example, the total assets under management for variable annuity accounts at the end of 2009 reached $1.35 trillion in the U.S. market alone [32]. In the GMWB survey by Towers Watson, companies responded that an average of 67% of new variable annuity policies (by premium) contained a GMWB rider [25] The total sales of variable annuities with a guaranteed retirement income rose 8% to $102.8 billion through September 30 2010 from a year earlier in the U.S. market [48].

1.1.2 Pricing GMWB as a Singular Control Problems

There has been limited academic research in the pricing of GMWBs. Milevsky and Salisbury were among the first to create two frameworks for pricing GMWBs [40]. The first framework is a static approach where the policy holders statically withdraw the contract amount each year. The annuity with a GMWB can then be decomposed into a Quanto Asian Put plus a generic term-certain annuity. A similar decomposition approach is discussed in [36]. The second framework is a dynamic approach where policy holders are fully rational and lapse the product when it is to their economic advantage. The pricing of a GMWB guarantee is then formally formulated as a singular stochastic control problem in [18]. In [11], a method is developed to solve an impulse control formulation of this problem. Methods for cases where withdrawals are only allowed at discrete times are given in [6] and [13].

Historically, it has been argued that the dynamic approach assumes optimal behavior of consumers, which is unlikely in practice. The authors in [40] claimed that the true value of the GMWB lies somewhere between the prices obtained by static and dynamic approaches. However, it is now considered prudent to price these contracts assuming optimal behavior, so that a worst case hedge can be constructed [16]. For an extension of these models to cases involving sub-optimal consumer behavior, see [13].

In this thesis, we focus on the singular control formulation of the GMWB pricing problem, which leads to an Hamilton Jacobi Bellman (HJB) Variational Inequality (VI). Stochastic control problems arise in many financial applications [42]. When the set of possible admissible controls becomes unbounded, the control problem is said to be singular. A classical singular control problem in finance concerns optimal investment, where an infinite control corresponds to an instantaneous reallocation between a risky and risk-free asset [52]. In the context of GMWB pricing problem, the infinite control corresponds to an instantaneous withdrawal of a finite amount.

In general, the solutions of singular stochastic control problems in finance are not smooth [42]. Hence, we seek the viscosity solution of such problems [17, 3, 5], well-known to represent the financially relevant solution (the dynamic programming formulation). A survey of numerical methods for stochastic control is given in [35] and [43]. Recently, a penalty method has been suggested in [18] for solution of the HJB VI for pricing a GMWB formulated as a singular control problem, assuming standard Geometric Brownian Motion. This method is a generalization of the penalty method used for American options [27]. The penalty method has also been applied to a singular stochastic control formulation of the continuous time portfolio selection problem [19]. In [18, 19], numerical examples were given by the authors to show the convergence of the proposed penalty method. However no formal proof of convergence was given.

The penalty method is extremely simple to implement, and hence merits thorough analysis. We conduct a rigorous proof of the convergence of the penalty method discretization to the viscosity solution of the HJB VI. For a discussion of the advantages of the penalty method compared with other numerical methods for singular control problems, we refer the reader to [18] and [19].

As an alternative, a direct control method was suggested for solving American option type problems in [31, 9]. We also apply this idea to the singular control formulation of pricing GMWB problem. It is straight forward to extend the proof of convergence to the viscosity solution for the direct control formulation.

We further extend both the penalty method and the direct control method to the case where the underlying asset follows a Poisson jump diffusion process [39, 15], which is a more realistic model of the real world risky asset stochastic process [24], compared with the Geometric Brownian Motion.

1.1.3 Iterative Methods for Solving HJB PIDEs

Both the penalty method and direct control method for pricing the GMWB formulated as a singular control problem lead to nonlinear Partial Integro-Differential Equations (PIDEs). This is often the case for problems which arise in the context of optimal stochastic control [35, 42, 43], in which case the nonlinear PDEs and PIDEs are typically Hamilton Jacobi Bellman (HJB) equations. Other examples include natural gas storage [10, 12, 51], asset allocation [52, 19, 56], and optimal trade execution [1, 37].

Solutions to such equations are not necessarily unique and one must take care to provide numerical procedures which ensure convergence to the viscosity solution. In order to ensure both numerical stability and convergence, implicit methods are typically chosen over explicit methods. Unfortunately implicit methods result in a nonlinear system of algebraic equations at each timestep. Solving these nonlinear equations is often the computational bottleneck.

One popular approach for solving the nonlinear equations resulting from a fully implicit discretization of HJB equations is based on the idea of policy iteration [35, 26, 9]. Policy iteration proceeds by solving a linear system at every step and then finding the control which gives the best local solution. The control which gives the optimal value is then used for the next linear system and the iteration is repeated. Policy iteration is particularly effective when the linear system is sparse or well structured and hence easy to solve.

When the underlying asset follows a Geometric Brownian Motion, the resulting iteration matrix is sparse but highly structured. Policy iteration works efficiently in this case. However, when we want to consider more sophisticated models for the underlying assets that are more consistent with market data, such as jump diffusion process [39, 15], the policy iteration matrix would be dense [21]. Hence the use of a direct solution of each linear system is prohibitive in terms of cost. Difficulties also arise when the underlying stochastic process is modeled using regime switching, another model that better fits the market data [29, 12]. In this case the associated linear system at each iteration is sparse but the sparsity pattern has lost its structure. Using a direct solution method (even with a good ordering technique) turns out to be no longer efficient.

We develop an efficient iteration scheme which we call a *fixed point policy iteration* scheme for solving the nonlinear discretized equations which arise from fully implicit discretization of HJB equations. We show that our approach converges and that the method is considerably more efficient than making use of a full policy iteration in the case that the underlying risky asset follows a jump diffusion process. We show how this fixed point policy iteration can be used to solve the discretized equations resulting from both penalty and direct control methods in the case where the the underlying risky asset follows a jump diffusion process [39, 15]. We refer the reader to [31] for another example of the use of fixed point policy iteration method, in the case of an American option written on an asset which follows a regime switching process [34].

The singular control formulation of the GMWB problem has a special structure that makes a *block matrix fixed point policy iteration* about one order of magnitude faster than a full matrix fixed point policy iteration. We derive sufficient conditions of the convergence of the block matrix fixed point policy iteration and verify that both the penalty method and the direct control method discretized equations can be solved by using the block matrix fixed point policy iteration.

In some cases, we observe that the fixed point policy iteration method does not converge even if theoretical conditions are satisfied. This can be explained by an analysis of the effects of inexact floating point arithmetic. We derive bounds on the penalty parameter (penalty method) and the scaling parameter (direct control method) so that convergence is expected in the presence of inexact arithmetic.

1.2 Contributions

The main contributions of this thesis are

- We review the formulation of the pricing GMWB as a singular control problem and extend it to the case where the underlying asset follows a Poisson jump diffusion process [39, 15]. This results in a HJB variational inequality, which is normally solved numerically after discretizing the original equation.

- We formulate the discretized HJB variational inequality resulting from pricing a GMWB as a singular control problem using both a penalty method [30] and a direct control method [31]. We use the method described in [55], where central differencing is used as much as possible, yet still results in a monotone scheme. This results in noticeably faster convergence (as the mesh is refined) compared to the use of pure upwinding schemes.

- We carry out a rigorous analysis of the penalty method in the context of the GMWB HJB variational inequality when the underlying asset follows a standard Geometric Brownian Motion. Assuming that the GMWB problem satisfies a strong comparison principle, we verify that the penalty method is consistent, stable and monotone. Hence from the results in [5, 3] we deduce convergence to the viscosity solution of the GMWB HJB variational inequality. The analysis can be easily extended to both the penalty method and the direct control method when the underlying asset follows a Poisson jump diffusion process.

- We develop a fixed point policy iteration scheme that is more efficient than the classical policy iteration in order to solve the algebraic equations resulting from a class of implicitly discretized HJB PDEs arising in finance. The singular control formulation of the GMWB problem has a special structure that makes a block matrix fixed point policy iteration about one order of magnitude more efficient than a full matrix fixed point policy iteration. We derive sufficient conditions which ensure convergence of the fixed point policy iteration and verify the conditions required for convergence of both the full matrix and the block matrix fixed point policy iterations.

- Both the penalty method and the direct control method require specification of a parameter, which may affect solution accuracy and convergence of the iteration. We carry out an analysis of this parameter for both formulations. We estimate bounds for the size of this parameter so that convergence can be expected taking into account floating point errors. Numerical tests show that the solution is insensitive to the value of the parameter over several orders of magnitude within the estimated bounds.

- We discuss the advantages and disadvantages, from a computational point of view, of the singular control formulation compared to the impulse control formulation of this problem.

- Although in this thesis we specifically consider the GMWB pricing problem, the methods we analyze here can be easily applied to many other singular stochastic control problems in finance.

- It appears that the direct control formulation has some advantages compared to the penalty formulation. We recommend the use of the direct control formulation.

1.3 Outline

The rest of this thesis is organized as follows. In Chapter 2, the GMWB pricing problem is formulated as a singular control problem with the assumption that the underlying asset follows a standard Geometric Brownian Motion. This leads to an HJB PDE/VI. In Chapter 3 a penalty method is introduced to solve the resulting PDE. The discretization of the PDE by standard finite differences with maximal use of central differencing is discussed. The proof of convergence of the discretization is then described. Chapter 4 introduces a direct control method to solve the PDE and the discretization. In Chapter 5 we describe how to extend the HJB PDE to the case where the underlying asset follows a Poisson jump diffusion process for both the penalty method and the direct control method. The resulting equation is an HJB PIDE/VI. Chapter 6 focuses on the iterative methods for solving the discretized algebraic equations. We review various techniques and present the *fixed point policy iteration* scheme. Sufficient conditions are derived to ensure the convergence of the fixed point policy iteration. A full matrix and a block matrix fixed point policy iterations are introduced for both the penalty method and the direct control method. Chapter 7 presents numerical results of fixed point policy iteration scheme. In Chapter 8, we discuss floating point roundoff error considerations in the context of the fixed point policy iteration, both for the penalty method and the direct control method. Numerical results are presented to demonstrate the effect of the floating point error on the iteration. We draw conclusions in Chapter 9.

Chapter 2

Singular Control GMWB Pricing Problem

In this chapter we formulate the pricing of a GMWB guarantee as a singular control problem assuming that the underlying asset follows a standard Geometric Brownian Motion. In Section 2.1, the GMWB pricing problem is posed as an HJB PDE/VI. In Section 2.2 boundary conditions are discussed. In Section 2.3 the formal definition of the GMWB as a singular control problem is given and viscosity solutions are briefly discussed. We summarize the main results of this chapter in Section 2.4.

2.1 Formulation of HJB VI

This section briefly reviews the singular control formulation of pricing a GMWB guarantee in [18] and introduces the notation to be used in the rest of this thesis. We use $W \equiv W(t)$ to denote the amount in the variable annuity account and $A \equiv A(t)$ to represent the guarantee account balance. We assume that the risky asset S which underlies the variable annuity account (before the deduction of any proportional fees) follows a standard Geometric Brownian Motion under the risk neutral measure. To be more precise, S satisfies the following stochastic differential equation

$$dS = rSdt + \sigma SdZ, \tag{2.1}$$

with r the risk free rate, dZ an increment of a standard Gauss-Wiener process, and σ the volatility associated with dZ.

The major feature of the GMWB is the guarantee on the return of the entire premium via withdrawal. The insurance company charges the policy holder a proportional annual insurance fee η, in return for providing this guarantee. Consequently, we have the following stochastic differential equation for W:

$$dW = \begin{cases} (r-\eta)Wdt + \sigma WdZ + dA & \text{if } W > 0, \\ 0 & \text{if } W = 0. \end{cases} \tag{2.2}$$

Let $\gamma \equiv \gamma(t)$ denote the withdrawal rate at time t and assume $\gamma \in [0, \infty)$. An infinite withdrawal rate corresponds to an instantaneous withdrawal of a finite amount. The policy guarantees that the sum of withdrawals throughout the policy's life is equal to the premium paid up front, which is denoted by ω_0. As a result, we have $A(0) = \omega_0$, and

$$A(t) = \omega_0 - \int_0^t \gamma(u)du, \quad A(t) \geq 0 \ . \tag{2.3}$$

In addition, almost all policies with a GMWB have a cap on the maximum allowed withdrawal rate without penalty. Let G be such a contractual withdrawal rate, and $\kappa < 1$ be the proportional penalty charge applied on the portion of the withdrawal exceeding G.

The net withdrawal rate $f(\gamma)$ received by the policy holder is then

$$f(\gamma) = \begin{cases} \gamma & 0 \leq \gamma \leq G, \\ G + (1-\kappa)(\gamma - G) & \gamma > G. \end{cases} \tag{2.4}$$

Let $V(W, A, t)$ be the value of the variable annuity with a GMWB. The no-arbitrage value $V(W, A, t)$ of the variable annuity with a GMWB is therefore given by [18]

$$V(W, A, t) = \max_{\gamma \in [0,\infty)} E_t \left[e^{-r(T-t)} \max((1-\kappa)A(T), W(T)) + \int_t^T e^{-r(u-t)} f(\gamma(u)) du \right] , \tag{2.5}$$

where T is the policy maturity time and the expectation E_t is taken under the risk neutral measure. The withdrawal rate γ is the control variable chosen to maximize the value of $V(W, A, t)$. Equation (2.5) represents the expected, discounted risk neutral cash flows from the guarantee, as discussed in [18].

With an abuse of notation, we now (and in the rest of this thesis) let $V = V(W, A, \tau = T - t)$. It is shown in [18] that the variable annuity value $V(W, A, \tau)$ is given by the following Hamilton-Jacobi-Bellman (HJB) Variational Inequality (VI)

$$\min \left[V_\tau - \mathcal{L}_\mathcal{G} V - G \max(\mathcal{F}V, 0), \ \kappa - \mathcal{F}V \right] = 0 \ . \tag{2.6}$$

where the operators $\mathcal{L}_\mathcal{G}$ and $\mathcal{F}$ are defined as

$$\begin{aligned} \mathcal{L}_\mathcal{G} V &= \frac{\sigma^2}{2} W^2 V_{WW} + (r - \eta) W V_W - rV \ , \\ \mathcal{F}V &= 1 - V_W - V_A \ . \end{aligned} \tag{2.7}$$

Equation (2.6) or the equivalent form (2.5) are commonly used by insurance firms to determine the no-arbitrage value of the GMWB contract. The solution is also used to determine a hedging strategy for the contract [40, 6, 13, 28, 25].

2.2 Boundary Conditions

2.2.1 Localization

The original GMWB problem is posed on the domain Ω^{∞}

$$(W, A, \tau) \in [0, \infty) \times [0, \omega_0] \times [0, T] \ . \tag{2.8}$$

For computational purposes, we define the GMWB problem on a finite computational domain, as in [18],

$$\Omega^L = [0, W_{\max}] \times [0, \omega_0] \times [0, T] \ . \tag{2.9}$$

We will analyze the convergence of the numerical scheme to the problem defined on Ω^L. Later, we will show that by solving the GMWB problem on successively larger domains, we converge to a unique limiting solution as $W_{\max} \to \infty$. We will also confirm this from some numerical experiments.

2.2.2 The Terminal and Boundary Conditions

Define the following sets of points $(W, A, \tau) \in \Omega^L$

$$
\begin{aligned}
\Omega_{\tau^0} &= [0, W_{\max}] \times [0, \omega_0] \times \{0\} \ , \\
\Omega_{W_0} &= \{0\} \times (0, \omega_0] \times (0, T] \\
\Omega_{W_{\max}} &= \{W_{\max}\} \times [0, \omega_0] \times (0, T] \\
\Omega_{A_0} &= [0, W_{\max}) \times \{0\} \times (0, T] \\
\Omega_{in} &= \Omega^L \backslash \Omega_{\tau^0} \backslash \Omega_{W_0} \backslash \Omega_{W_{\max}} \backslash \Omega_{A_0} \\
\partial \Omega_{in} &= \Omega_{\tau^0} \cup \Omega_{W_0} \cup \Omega_{W_{\max}} \cup \Omega_{A_0} \ . \qquad (2.10)
\end{aligned}
$$

For $(W, A, \tau) \in \Omega_{in}$, we solve

$$
\min\left[V_\tau - \mathcal{L}_\mathcal{G} V - G \max(\mathcal{F}V, 0),\ \kappa - \mathcal{F}V\right] = 0 \qquad (W, A, \tau) \in \Omega_{in} \ . \qquad (2.11)
$$

As discussed in [18], at maturity, the policy holder takes the remaining guarantee withdrawal net of penalty charge or the remaining balance of the personal account, whichever is greater. Therefore at $\tau = 0$, the terminal condition is

$$
V(W, A, \tau = 0) = \max\left[W, (1-\kappa)A\right] \qquad (W, A, \tau) \in \Omega_{\tau^0} \ . \qquad (2.12)
$$

As $W \to 0$, $V_W \to 0$ [18] (since W must be nonnegative). Thus, at $W = 0$, equation (2.6) becomes

$$\min\left[V_\tau - rV - G\max(1 - V_A, 0), \kappa - (1 - V_A)\right] = 0$$
$$(W, A, \tau) \in \Omega_{W_0} \ . \qquad (2.13)$$

As $W \to \infty$, according to [18], the withdrawal guarantee becomes insignificant for W sufficiently large. More precisely, a straightforward financial argument shows that the exact boundary condition at $W_{\max}$ is

$$V(W_{\max}, A, \tau) = e^{-\eta\tau} W_{\max}\left(1 + O\left(\frac{1}{W_{\max}}\right)\right) \ ; \quad W_{\max} \to \infty \ . \qquad (2.14)$$

Therefore as in [18, 23], we impose the following condition at $W_{\max}$

$$V(W_{\max}, A, \tau) = e^{-\eta\tau} W_{\max} \ ,$$
$$(W, A, \tau) \in \Omega_{W_{\max}} \ . \qquad (2.15)$$

As $A \to 0$, no withdrawal is possible, so the variational inequality becomes the following linear PDE [11]

$$V_\tau = \mathcal{L}_\mathcal{G} V$$
$$(W, A, \tau) \in \Omega_{A_0} \ . \qquad (2.16)$$

Note that as discussed in [18], no boundary condition is required at $A = \omega_0$ due to hyperbolic nature of the variable A. Since equations (2.13), (2.16) can be solved without any knowledge of the solution in the interior of Ω^L, they are essentially Dirichlet conditions.

2.3 Compact Representation

We now write the GMWB problem in a compact form, which includes the terminal and boundary conditions as a single equation. Define vector $\mathbf{x} = (W, A, \tau)$, and let $DV(\mathbf{x}) = (V_W, V_A, V_\tau)$ and $D^2V(\mathbf{x}) = V_{WW}$, and the equation

$$F_{\Omega^L} V \equiv F(D^2V(\mathbf{x}), DV(\mathbf{x}), V(\mathbf{x}), \mathbf{x}) = 0, \mathbf{x} \in \Omega^L, \tag{2.17}$$

where operator $F_{\Omega^L} V$ is defined by

$$F_{\Omega^L} V = \begin{cases} F_{\text{in}} V \equiv F_{\text{in}}(D^2V(\mathbf{x}), DV(\mathbf{x}), V(\mathbf{x}), \mathbf{x}), & \mathbf{x} \in \Omega_{in}, \\ F_{A_0} V \equiv F_{A_0}(D^2V(\mathbf{x}), DV(\mathbf{x}), V(\mathbf{x}), \mathbf{x}), & \mathbf{x} \in \Omega_{A_0}, \\ F_{W_0} V \equiv F_{W_0}(DV(\mathbf{x}), V(\mathbf{x}), \mathbf{x}), & \mathbf{x} \in \Omega_{W_0}, \\ F_{W_{\max}} V \equiv F_{W_{\max}}(V(\mathbf{x}), \mathbf{x}), & \mathbf{x} \in \Omega_{W_{\max}}, \\ F_{\tau_0} V \equiv F_{\tau_0}(V(\mathbf{x}), \mathbf{x}), & \mathbf{x} \in \Omega_{\tau_0}, \end{cases} \tag{2.18}$$

with operators

$$F_{in} V = \min\left[V_\tau - \mathcal{L}_{\mathcal{G}} V - G \max(\mathcal{F}V, 0),\ \kappa - \mathcal{F}V\right], \tag{2.19}$$

$$F_{A_0} V = V_\tau - \mathcal{L}_{\mathcal{G}} V, \tag{2.20}$$

$$F_{W_0} V = \min\left[V_\tau + rV - G \max(1 - V_A, 0),\ \kappa - 1 + V_A\right], \tag{2.21}$$

$$F_{W_{\max}} V = V - e^{-\eta\tau} W, \tag{2.22}$$

$$F_{\tau^0} V = V - \max\left[W,\ (1-\kappa)A\right]. \tag{2.23}$$

Definition 2.3.1 (Singular Control GMWB Pricing Problem)**.** *The pricing problem for* *the GMWB guarantee using a singular control formulation is defined as*

$$F_{\Omega^L}(D^2V(\mathbf{x}), DV(\mathbf{x}), V(\mathbf{x}), \mathbf{x}) = 0 \ . \tag{2.24}$$

Clearly, F_{Ω^L} satisfies the ellipticity condition

$$F_{\Omega^L}(D^2V(\mathbf{x}) + \delta, DV(\mathbf{x}), V(\mathbf{x}), \mathbf{x}) \leq F_{\Omega^L}(D^2V(\mathbf{x}), DV(\mathbf{x}), V(\mathbf{x}), \mathbf{x}) \ \ ; \ \ \forall \delta \geq 0 \tag{2.25}$$

since the coefficient of $D^2V(\mathbf{x})$ in F_{Ω^L} is non-positive. Note that F_{Ω^L} is discontinuous [5, 3], since we include the boundary equations in F_{Ω^L}, which are in general not the limit of the equations from the interior.

In the following, let u^* (u_*) denote the upper (lower) semi-continuous envelope of the function $u : X \to \mathbb{R}$, where X is a closed subset of $\mathbb{R}^N$, such that

$$u^*(\hat{x}) = \limsup_{\substack{x \to \hat{x} \\ \hat{x}, x \in X}} u(x), \qquad\qquad u_*(\hat{x}) = \liminf_{\substack{x \to \hat{x} \\ \hat{x}, x \in X}} u(x). \tag{2.26}$$

In general, the solution to a singular stochastic control problem is non-smooth, and we seek the viscosity solution.

Definition 2.3.2 (Viscosity Solution)**.** *A locally bounded function $V : \Omega^L \to \mathbb{R}$ is a viscosity subsolution (respectively supersolution) of* (2.24) *if and only if for all smooth test functions $\phi(\mathbf{x}) \in C^2$, and for all maximum (respectively minimum) points $\mathbf{x}$ of $V^* - \phi$ (respectively $V_* - \phi$), one has*

$$\begin{array}{ll} & (F_{\Omega^L})_*(D^2\phi(\mathbf{x}), D\phi(\mathbf{x}), V^*(\mathbf{x}), \mathbf{x}) \leq 0 \\ \Big(\textit{respectively} & (F_{\Omega^L})^*(D^2\phi(\mathbf{x}), D\phi(\mathbf{x}), V_*(\mathbf{x}), \mathbf{x}) \geq 0 \Big) \ . \end{array} \tag{2.27}$$

A locally bounded function V is a viscosity solution if it is both a viscosity subsolution and a viscosity supersolution.

In [49], it is shown that an impulse control formulation of the GMWB pricing problem (under a jump diffusion) satisfies a strong comparison principle. However, there does not seem to be a proof of this result for the singular control formulation of this problem. [18] states but does not prove the comparison principle for equation (2.24). Let $\Gamma \subset \partial\Omega_{in}$. We make the following assumption.

Assumption 2.3.1 (Strong Comparison)**.** *The GMWB singular control problem as given in Definition 2.3.1 satisfies a strong comparison result in $\Omega_{in} \cup \Gamma$, $\Gamma \subset \partial\Omega_{in}$ hence a unique continuous viscosity solution exists in $\Omega_{in} \cup \Gamma$.*

Remark 2.3.1. *We cannot in general hope for a continuous solution over the whole of Ω^L. It is possible that loss of boundary data can occur over parts of $\partial\Omega_{in}$. For example, for points near $\Omega_{W_{\max}}$, if it is optimal to withdraw a finite amount instantaneously, then the HJB equation degenerates to a first order equation, with outgoing characteristics. Hence the boundary condition at some points in $\Omega_{W_{\max}}$ may be irrelevant, in the sense that the boundary condition at these points does not influence the interior solution.*

[42] discusses another case where singular control problems cannot be continuous over the entire closed solution domain. It may be the case that the terminal condition at Ω_{τ^0} is not compatible with the control problem in the sense that it may be optimal to immediately make a transaction the instant after $\tau = 0$. This would result in a discontinuity in the solution as $\tau \to 0$, from points in $\Omega^L \backslash \Omega_{\tau^0}$. However, this does not occur in our case, since it is never optimal to make an instantaneous withdrawal at $\tau = 0^+$, with the particular initial condition (2.12).

All these issues need to be addressed in proving a strong comparison property, in order to define precisely those regions in Γ we can expect a continuous, unique viscosity solution.

However, the location of Γ *has little impact on the computational algorithm. The boundary data is either used or irrelevant. In all cases we can consider the computed solution as the limiting value approaching* $\partial\Omega_{in}$ *from the interior.*

Remark 2.3.2. *Note that in the case that an asymptotic form of the solution as* $W_{\max} \to \infty$ *is not available, it is possible to impose an arbitrary boundary condition (satisfying certain growth conditions) and take the limit as* $W_{\max} \to \infty$*. This will converge to the viscosity solution in the unbounded domain, as shown in [4].*

2.4 Summary

The main results of this chapter are as follows:

- We formulate the pricing of GMWB as a singular control problem which results in an HJB PDE/VI with the assumption that the underlying asset follows a standard Geometric Brownian Motion.
- We discuss boundary conditions of the resulting PDE.
- We formally define the GMWB as a singular control problem in a compact representation and discuss viscosity solutions.

Chapter 3

Penalty Method

This chapter discusses the numerical scheme for solving equations (2.6) by using a penalty method. Section 3.1 informally derives the equation (2.6) to give some intuition for penalty method numerical scheme. Section 3.2 discusses the finite difference discretization and how to use central differences as much as possible for the penalty method. In Section 3.3 we use a matrix form to represent the discrete penalized equations. In Section 3.4 a rigorous proof is given to show that the discrete penalized equations converge to the viscosity solution of the problem in Definition 2.3.2. Section 3.5 summarizes the main results of this chapter.

3.1 Informal Derivation of HJB VI and the Penalized Form

We repeat here the informal derivation of equation (2.6) given in [18] to give some intuition for the formulation of the penalized HJB PDE/VI. Suppose that we restrict the maximum withdrawal range to be in $\gamma \in [0, \vartheta]$ with $\vartheta > G$ finite. Let $\vartheta = 1/\varepsilon$. Then it is shown in [18] that the variable annuity value parameterized by ε, denoted by $V^{\varepsilon}(W, A, \tau)$ is given from the solution to the following HJB equation

$$V_\tau^\varepsilon = \mathcal{L}_\mathcal{G} V^\varepsilon + \max_{\gamma \in [0,\vartheta]} h(\gamma), \tag{3.1}$$

where $\mathcal{L}_\mathcal{G}$ is given in (2.7) and $h(\gamma)$ is given by

$$\begin{aligned} h(\gamma) &= f(\gamma) - \gamma V_W^\varepsilon - \gamma V_A^\varepsilon \\ &= \begin{cases} (1 - V_W^\varepsilon - V_A^\varepsilon)\gamma & \text{if } 0 \le \gamma \le G, \\ (1 - V_W^\varepsilon - V_A^\varepsilon - \kappa)\gamma + \kappa G & \text{if } \gamma > G. \end{cases} \end{aligned} \tag{3.2}$$

An informal derivation of equation (3.1) using a hedging argument is given in Appendix A. The function $h(\gamma)$ is piecewise linear, so its maximum value is achieved when γ is 0, G, or ϑ. Assuming $\vartheta > G$, we then have

$$\max_{\gamma \in [0,\vartheta]} h(\gamma) = \begin{cases} 0 & \text{if } \mathcal{F}V^\varepsilon \le 0, \\ G\mathcal{F}V^\varepsilon & \text{if } 0 < \mathcal{F}V^\varepsilon < \kappa, \\ \vartheta(\mathcal{F}V^\varepsilon - \kappa) + \kappa G & \text{if } \mathcal{F}V^\varepsilon \ge \kappa. \end{cases} \tag{3.3}$$

The first two cases for $\max_{\gamma \in [0,\vartheta]} h(\gamma)$ in (3.3) are identical to $G \max(0, \mathcal{F}V^\varepsilon)$. Substituting (3.3) into (3.1), we obtain (with $\vartheta = 1/\varepsilon$)

$$- V_\tau^\varepsilon + \mathcal{L}_\mathcal{G} V^\varepsilon + \max \left[G \max(0, \mathcal{F}V^\varepsilon), \frac{(\mathcal{F}V^\varepsilon - \kappa)}{\varepsilon} + \kappa G \right] = 0 \ . \tag{3.4}$$

The value function $V^\varepsilon(W, A, \tau)$ is then the solution of

$$\min \left[V_\tau^\varepsilon - \mathcal{L}_\mathcal{G} V^\varepsilon - G \max(0, \mathcal{F}V^\varepsilon), V_\tau^\varepsilon - \mathcal{L}_\mathcal{G} V^\varepsilon - \kappa G + \frac{(\kappa - \mathcal{F}V^\varepsilon)}{\varepsilon} \right] = 0 \ . \tag{3.5}$$

We can rewrite (3.5) (since $\varepsilon > 0$) equivalently

$$\min\left[V_\tau^\varepsilon - \mathcal{L}_\mathcal{G} V^\varepsilon - G\max(0, \mathcal{F}V^\varepsilon), \kappa - \mathcal{F}V^\varepsilon + \varepsilon\left(V_\tau^\varepsilon - \mathcal{L}_\mathcal{G} V^\varepsilon - \kappa G\right)\right] = 0 \ . \tag{3.6}$$

Taking the limit $\varepsilon \to 0$ (which corresponds to an instantaneous withdrawal of a finite amount) gives the following HJB variational inequality

$$\min\left[V_\tau - \mathcal{L}_\mathcal{G} V - G\max(\mathcal{F}V, 0),\ \kappa - \mathcal{F}V\right] = 0 \ . \tag{3.7}$$

Consequently, we can see, at least intuitively, that

$$\lim_{\varepsilon\to 0}\left\{V_\tau^\varepsilon - \mathcal{L}_\mathcal{G} V^\varepsilon - \max\left[G\max(0, \mathcal{F}V^\varepsilon), \frac{(\mathcal{F}V^\varepsilon - \kappa)}{\varepsilon} + \kappa G\right]\right\} = 0 \tag{3.8}$$

is equivalent to equation (2.6). Keeping ε finite, we can rewrite equation (3.8) in *control form*

$$V_\tau^\varepsilon \ = \ \mathcal{L}_\mathcal{G} V^\varepsilon + \max_{\substack{\varphi\in\{0,1\},\psi\in\{0,1\} \\ \varphi\psi=0}} \left[\varphi G \mathcal{F}V^\varepsilon + \psi\left(\frac{(\mathcal{F}V^\varepsilon - \kappa)}{\varepsilon} + \kappa G\right)\right] \ . \tag{3.9}$$

The basic idea of the penalty method is to discretize equation (3.9), and let $\varepsilon \to 0$ as the mesh and timestep size tend to zero. In Section 3.4, we will give a rigorous proof that this algorithm converges to the viscosity solution of equation (2.6), provided that equation (2.6) satisfies a strong comparison principle.

3.2 Discretized Equation

3.2.1 Penalty Form

We will discretize the penalty form of the equations (3.9) and show that the discrete equations converge to the viscosity solution of the problem in Definition 2.3.2. Using the notation $\mathcal{D}_{WW}V = V_{WW}$, $\mathcal{D}_W V = V_W$ and $\mathcal{D}_A V = V_A$, in $(W, A, \tau) \in \Omega_{in} \cup \Omega_{A_0}$ we will discretize

$$\begin{aligned} V^\varepsilon_\tau &= \mathcal{L}_\mathcal{G} V^\varepsilon + \max_{\substack{\varphi\in\{0,1\},\psi\in\{0,1\}\\ \varphi\psi=0}} \left[\varphi G \mathcal{F} V^\varepsilon + \psi\left(\frac{(\mathcal{F}V^\varepsilon - \kappa)}{\varepsilon} + \kappa G\right)\right] \\ &\quad (W, A, \tau) \in \Omega_{in} \cup \Omega_{A_0} \ . \end{aligned} \tag{3.10}$$

where

$$\mathcal{L}_\mathcal{G} V^\varepsilon = \frac{\sigma^2}{2} W^2 \mathcal{D}_{WW} V^\varepsilon + (r-\eta) W \mathcal{D}_W V^\varepsilon - r V^\varepsilon \ , \tag{3.11}$$

$$\mathcal{F} V^\varepsilon = 1 - \mathcal{D}_W V^\varepsilon - \mathcal{D}_A V^\varepsilon \ . \tag{3.12}$$

and we understand that $\phi = \psi = 0$ in Ω_{A_0}. At $W = 0$, we discretize

$$\begin{aligned} V^\varepsilon_\tau &= -rV^\varepsilon + \max_{\substack{\varphi\in\{0,1\},\psi\in\{0,1\}\\ \varphi\psi=0}} \left[\varphi G (1 - \mathcal{D}_A V^\varepsilon) + \psi\left(\frac{(1 - \mathcal{D}_A V^\varepsilon - \kappa)}{\varepsilon} + \kappa G\right)\right] \\ &\quad (W, A, \tau) \in \Omega_{W_0} \ . \end{aligned} \tag{3.13}$$

3.2.2 Discretization of the Penalized Equations

We will discretize equation (3.10) and equation (3.13) in the domain $\Omega_{\text{in}} \cup \Omega_{A_0} \cup \Omega_{W_0}$. We use an unequally spaced grid in the W direction, given by $\{W_1, \ldots, W_i, \ldots, W_{i_{\max}}\}$.

The nodes in the A direction are denoted by $\{A_1, \ldots, A_j, \ldots, A_{j_{\max}}\}$, where $W_1 = A_1 = 0$, $W_{i_{\max}} = W_{\max}$ and $A_{j_{\max}} = \omega_0$. We denote the n^{th} time-step by $\tau^n = n\Delta\tau$, with $N = T/\Delta\tau$. We will always assume that $W_{i_{\max}} \gg A_{j_{\max}}$.

Denote the approximate solution at (W_i, A_j, τ^n) by $V_{i,j}^n$. We use a standard three point finite difference method to approximate the $\mathcal{D}_{WW}V$ derivative. This approximation is second order for smoothly varying grid spacing. The $\mathcal{D}_A V$ derivative is approximated by a first order backward differencing method. The $\mathcal{D}_W V$ derivative is approximated by a second order central differencing or a first order forward/backward differencing. Let $\mathcal{D}_A^h$, $\mathcal{D}_W^h$ and $\mathcal{D}_{WW}^h$ (defined in Appendix B.1) denote the discretized first and second order partial differential operators. The discretized $\mathcal{L}_\mathcal{G}$ and $\mathcal{F}$ operators can then be written as

$$\mathcal{L}_\mathcal{G}^h V_{i,j}^n = \begin{cases} \frac{\sigma^2}{2} W_i^2 \mathcal{D}_{WW}^h V_{i,j}^n + (r-\eta) W_i \mathcal{D}_W^h V_{i,j}^n - r V_{i,j}^n, & (W_i, A_j, \tau^n) \in \Omega_{in} \cup \Omega_{A_0} \\ -r V_{i,j}^n, & (W_i, A_j, \tau^n) \in \Omega_{W_0} \end{cases}, \quad (3.14)$$

$$\mathcal{F}^h V_{i,j}^n = \begin{cases} 1 - \mathcal{D}_W^h V_{i,j}^n - \mathcal{D}_A^h V_{i,j}^n, & (W_i, A_j, \tau^n) \in \Omega_{in} \\ 1 - \mathcal{D}_A^h V_{i,j}^n, & (W_i, A_j, \tau^n) \in \Omega_{W_0} \\ 0, & (W_i, A_j, \tau^n) \in \Omega_{A_0} \end{cases}. \quad (3.15)$$

Using fully implicit time-stepping, equation (3.10) has the following discretized form

$$\frac{V_{i,j}^{n+1} - V_{i,j}^n}{\Delta\tau} = \mathcal{L}_\mathcal{G}^h V_{i,j}^{n+1} + \max_{\substack{\varphi\in\{0,1\},\psi\in\{0,1\} \\ \varphi\psi=0}} \left[\varphi G \mathcal{F}^h V_{i,j}^{n+1} + \psi \left(\frac{(\mathcal{F}^h V_{i,j}^{n+1} - \kappa)}{\varepsilon} + \kappa G \right) \right]$$
$$i = 1, 2, \ldots, i_{\max} - 1, \; j = 1, 2, \ldots, j_{\max}, \; n = 0, 1, \ldots, N-1 \;, \quad (3.16)$$

or equivalently

$$\frac{V_{i,j}^{n+1} - V_{i,j}^{n}}{\Delta \tau} = \max_{\substack{\varphi \in \{0,1\}, \psi \in \{0,1\} \\ \varphi\psi = 0}} \left[\mathcal{L}_{\mathcal{G}}^{h} V_{i,j}^{n+1} + \varphi G \mathcal{F}^{h} V_{i,j}^{n+1} + \psi \left(\frac{(\mathcal{F}^{h} V_{i,j}^{n+1} - \kappa)}{\varepsilon} + \kappa G \right) \right]$$
$$i = 1, 2, \ldots, i_{\max} - 1,\ j = 1, 2, \ldots, j_{\max},\ n = 0, 1, \ldots, N-1\ , \quad (3.17)$$

and finally (by expanding the $\mathcal{L}_{\mathcal{G}}^{h}$, $\mathcal{F}^{h}$ and $\mathcal{D}_{A}^{h}$ operators)

$$\frac{V_{i,j}^{n+1} - V_{i,j}^{n}}{\Delta \tau} = \max_{\substack{\varphi \in \{0,1\}, \psi \in \{0,1\} \\ \varphi\psi = 0}} \left[\mathcal{A}_{\varphi,\psi}^{h} V_{i,j}^{n+1} + p_{i,j}^{n+1}(\varphi,\psi) V_{i,j-1}^{n+1} + q_{i,j}^{n+1}(\varphi,\psi) \right] ,$$
$$i = 2, 3, \ldots, i_{\max} - 1,\ j = 2, 3, \ldots, j_{\max},\ n = 0, 1, \ldots, N-1\ , \qquad (3.18)$$

where operator $\mathcal{A}_{\varphi,\psi}^{h}$ is defined as

$$\mathcal{A}_{\varphi,\psi}^{h} V_{i,j}^{n} \equiv a_{i,j}^{n}(\varphi,\psi) \mathcal{D}_{WW}^{h} V_{i,j}^{n} + b_{i,j}^{n}(\varphi,\psi) \mathcal{D}_{W}^{h} V_{i,j}^{n} - c_{i,j}^{n}(\varphi,\psi) V_{i,j}^{n} \qquad (3.19)$$

and

$$\begin{array}{rclrcl} a_{i,j}^{n}(\varphi,\psi) & = & \frac{\sigma^2}{2} W_i^2\ , & p_{i,j}^{n}(\varphi,\psi) & = & \frac{(\varphi G + \frac{\psi}{\varepsilon})}{\Delta A_j^{-}}\ , \\ b_{i,j}^{n}(\varphi,\psi) & = & (r-\eta) W_i - (\varphi G + \frac{\psi}{\varepsilon})\ , & q_{i,j}^{n}(\varphi,\psi) & = & \varphi G + \psi(\frac{1-\kappa}{\varepsilon} + \kappa G)\ , \\ c_{i,j}^{n}(\varphi,\psi) & = & r + \frac{(\varphi G + \frac{\psi}{\varepsilon})}{\Delta A_j^{-}}\ , & \Delta A_j^{-} & = & A_j - A_{j-1}\ . \end{array} \qquad (3.20)$$

Remark 3.2.1. *We have written the coefficient $a_{i,j} = a_{i,j}(\varphi,\psi)$ although there is no explicit dependence on (φ,ψ) in this case in order to keep the result more general.*

Let

$$\left(\varphi_{i,j}^{n}, \psi_{i,j}^{n}\right) = \mathop{\arg\max}_{\substack{\varphi \in \{0,1\}, \psi \in \{0,1\} \\ \varphi\psi = 0}} \left[\mathcal{A}_{\varphi,\psi}^{h} V_{i,j}^{n} + p_{i,j}^{n}(\varphi,\psi) V_{i,j-1}^{n+1} + q_{i,j}^{n}(\varphi,\psi) \right]_i . \qquad (3.21)$$

Equation (3.18) becomes

$$\frac{V_{i,j}^{n+1} - V_{i,j}^{n}}{\Delta\tau} = \mathcal{A}^h_{\varphi_{i,j}^{n+1},\psi_{i,j}^{n+1}} V_{i,j}^{n+1} + p_{i,j}^{n+1}(\varphi_{i,j}^{n+1}, \psi_{i,j}^{n+1}) V_{i,j-1}^{n+1} + q_{i,j}^{n+1}(\varphi_{i,j}^{n+1}, \psi_{i,j}^{n+1}) \ ,$$
$$i = 2, 3, \ldots, i_{\max} - 1, \ j = 2, 3, \ldots, j_{\max}, \ n = 0, 1, \ldots, N-1 \ . \quad (3.22)$$

The discretized $\mathcal{D}_W^h V_{i,j}^n$ term in $\mathcal{A}^h_{\varphi,\psi} V_{i,j}^n$ can be obtained by applying central, forward, or backward differencing to the $\mathcal{D}_W V^\varepsilon$ term. A few steps of algebra show that the $\mathcal{A}^h_{\varphi,\psi}$ operator can also be written equivalently as

$$\mathcal{A}^h_{\varphi,\psi} V_{i,j}^n = \alpha_{i,j}^n(\varphi,\psi) V_{i-1,j}^n - \big(\alpha_{i,j}^n(\varphi,\psi) + \beta_{i,j}^n(\varphi,\psi) + c_{i,j}^n(\varphi,\psi)\big) V_{i,j}^n + \beta_{i,j}^n(\varphi,\psi) V_{i+1,j}^n,$$
$$i = 2, 3, \ldots, i_{\max} - 1, \ j = 2, 3, \ldots, j_{\max}, \ n = 1, 2, \ldots, N-1 \ . \quad (3.23)$$

The $\alpha_{i,j}^n(\varphi,\psi)$ and $\beta_{i,j}^n(\varphi,\psi)$ in (3.23) are determined by the differencing method used in W direction, $\alpha_{i,j}^n \in \{\alpha_{i,j,cent}^n, \alpha_{i,j,for/back}^n\}$, $\beta_{i,j}^n \in \{\beta_{i,j,cent}^n, \beta_{i,j,for/back}^n\}$, which are defined in Appendix B.2. We use central differencing as much as possible in the W direction to ensure that the positive coefficient condition is satisfied (see [44])

$$\alpha_{i,j}^n \geq 0 \ \ ; \ \ \beta_{i,j}^n \geq 0 \ . \quad (3.24)$$

Because $c_{i,j} \geq 0$ always holds, condition (3.24) is a sufficient condition to ensure a positive coefficient discretization scheme. Note that different nodes may use different differencing schemes.

By applying forward or backward differencing to $\mathcal{D}_W V^\varepsilon$ in the equation (3.10), the positive coefficient condition is guaranteed. In [18], central differencing is used on $\mathcal{D}_W V^\varepsilon$ term in $\mathcal{L}_\mathcal{G} V^\varepsilon$ and backward differencing is used on $\mathcal{D}_W V^\varepsilon$ term in $\mathcal{F} V^\varepsilon$. This requires a grid spacing condition in order to satisfy the positive coefficient condition. Because backward differencing in $\mathcal{F}V^\varepsilon$ gives a first order truncation error in the W direction, whereas central differencing is second order correct (for smooth functions), we would like to use central differencing as much as possible on the $\mathcal{D}_W V^\varepsilon$ term both in $\mathcal{L}_\mathcal{G} V^\varepsilon$ and $\mathcal{F}V^\varepsilon$. However, we must ensure that the positive coefficient condition (3.24) is satisfied. To use central differencing on the $\mathcal{D}_W V^\varepsilon$ term and maintain a positive coefficient condition at the same time, we require

$$\frac{1}{W_i - W_{i-1}} \geq \frac{(r-\eta) - \frac{(\varphi_{i,j}^{n+1} G + \frac{\psi_{i,j}^{n+1}}{\varepsilon})}{W_i}}{\sigma^2 W_i}\ ; \tag{3.25}$$

$$\frac{1}{W_{i+1} - W_i} \geq -\frac{(r-\eta) - \frac{(\varphi_{i,j}^{n+1} G + \frac{\psi_{i,j}^{n+1}}{\varepsilon})}{W_i}}{\sigma^2 W_i}\ . \tag{3.26}$$

In [55], the authors discussed maximal use of central differencing for HJB PDEs. Note that the differencing method to be used at a given node depends on the value of control parameters. At a given node, for a given control parameter value, we first try to discretize the $\mathcal{D}_W V^\varepsilon$ term by using central differencing. If this gives positive coefficients as described in (3.24), central differencing will be used for the node for this given control parameter value. Otherwise, either forward or backward differencing will be used for the node given this control parameter value. In our case, since we have three possible control parameter values, at each node we determine the differencing method for each one of the three control parameter values. The local optimization criterion in (3.21) subsequently determines which control parameter value is the optimal value. The differencing method corresponding to this optimal control parameter value is then chosen to discretize the equation for the given node. Note that it is shown in Appendix B.2 that at least one of central, forward or backward differencing must result in a positive coefficient scheme.

Equation (3.22) holds for $(W_i, A_j, \tau^{n+1}) \in \Omega_{in}$. The discrete forms of equations (2.12), (2.15), (2.16) and (3.13) are as follows. For $(W_i, A_j, \tau^{n+1}) \in \Omega_{\tau^0}$, $(\tau^n = 0)$ we have simply

$$V_{i,j}^0 = \max\big[W_i, (1-\kappa)A_j\big] \quad . \tag{3.27}$$

In the region $(W_i, A_j, \tau^{n+1}) \in \Omega_{W_0}$ condition (3.13) is imposed by using equation (3.22) with

$$\alpha_{1,j}^{n+1} = \beta_{1,j}^{n+1} = 0, j = 2, 3, \ldots, j_{\max}. \tag{3.28}$$

For $(W_i, A_j, \tau^{n+1}) \in \Omega_{A_0}$, condition (2.16) is imposed by using equation (3.22) with

$$\begin{aligned} \varphi_{i,1}^{n+1} &= \psi_{i,1}^{n+1} = 0 \;\; ; \;\; i = 1, 2, \ldots, i_{\max} - 1. \\ \alpha_{i,1}^{n+1} &= \beta_{i,1}^{n+1} = 0 \;\; ; \;\; i = 1 \quad . \end{aligned} \tag{3.29}$$

At $W = W_{i_{\max}}$, or $(W_i, A_j, \tau^{n+1}) \in \Omega_{W_{\max}}$, we have (from equation (2.15))

$$V_{i_{\max},j}^{n+1} e^{\eta \Delta\tau} = V_{i_{\max},j}^n, \tag{3.30}$$

assuming $V_{i_{\max},j}^0 = W_{\max}$. By setting

$$\begin{aligned} &c_{i_{\max},j}^{n+1} = \eta \;\; ; \;\; \alpha_{i_{\max},j}^{n+1} = \beta_{i_{\max},j}^{n+1} = \varphi_{i_{\max},j}^{n+1} = \psi_{i_{\max},j}^{n+1} = 0; \\ &j = 1, 2, \ldots, j_{\max}, \end{aligned} \tag{3.31}$$

in equation (3.22) we obtain

$$(\frac{1}{\Delta\tau} + \eta) V_{i_{\max},j}^{n+1} = \frac{1}{\Delta\tau} V_{i_{\max},j}^n \tag{3.32}$$

which is a locally second order approximation to equation (3.30). Consequently, at all points $(W_i, A_j, \tau^{n+1}) \in \Omega^L \backslash \Omega_{\tau^0}$, an equation of the form (3.22) holds, if we define $V_{0,j}^{n+1} =$

$V^{n+1}_{i_{\max}+1,j} = V^{n+1}_{i,0} = 0.$

3.3 Matrix Form of the Discretized Equations

It is convenient to use a matrix form to represent the discretized equations. In this section we define a number of matrices and vectors to represent the discretized PDE in (3.22). Define vectors

$$\begin{aligned} v^n_{*,j} &= \left(V^n_{1,j}, V^n_{2,j}, \ldots, V^n_{i_{\max},j}\right)' \\ \mathbf{v}^n &= \left((v^n_{*,1})', (v^n_{*,2})', \ldots, (v^n_{*,j_{\max}})'\right)' \quad . \end{aligned} \tag{3.33}$$

Define an $i_{\max} \times i_{\max}$ tridiagonal matrix $\mathbf{A}^n_{\mathbf{j}}$ so that the entry on the i^{th} row and k^{th} column is defined as

$$[\mathbf{A}^n_{\mathbf{j}}]_{i,k} = \begin{cases} -\alpha^n_{i,j} & \text{if } k = i-1, i = 2, \ldots, i_{\max} \\ -\beta^n_{i,j} & \text{if } k = i+1, i = 1, \ldots, i_{\max}-1 \\ \frac{1}{\Delta\tau} + \alpha^n_{i,j} + \beta^n_{i,j} + c^n_{i,j} & \text{if } k = i, i = 1, \ldots, i_{\max} \\ 0 & \text{otherwise .} \end{cases} \tag{3.34}$$

Define an $i_{\max} \times i_{\max}$ diagonal matrix $\mathbf{P}^n_{\mathbf{j}}$ so that entries on the diagonal are defined as

$$[\mathbf{P}^n_{\mathbf{j}}]_{i,i} = \begin{cases} p^n_{i,j} & \text{if } i \le i_{\max} - 1, \\ 0 & \text{if } i = i_{\max}. \end{cases} \tag{3.35}$$

Let vectors $q^n_{*,j}$ and $\mathbf{q}^n$ be defined by

$$\begin{aligned} q^n_{*,j} &= \left(q^n_{1,j}, q^n_{2,j}, \ldots, q^n_{i_{\max}-1,j}, 0\right)' \ ; \\ \mathbf{q}^n &= \left((q^n_{*,1})', (q^n_{*,2})', \ldots, (q^n_{*,j_{\max}})'\right)' \ . \end{aligned} \tag{3.36}$$

We can write equation (3.22) as

$$\mathbf{A_j}^{n+1} v^{n+1}_{*,j} - \mathbf{P_j}^{n+1} v^{n+1}_{*,j-1} = \tfrac{1}{\Delta\tau} v^n_{*,j} + q^{n+1}_{*,j} \ , \tag{3.37}$$

where

$$\{\varphi^{n+1}_{i,j}, \psi^{n+1}_{i,j}\} = \underset{\substack{\varphi\in\{0,1\},\psi\in\{0,1\} \\ \varphi\psi=0}}{\arg\max} \left[\left[\tfrac{1}{\Delta\tau}\mathbf{I} - \mathbf{A_j}^{n+1}(\varphi,\psi)\right] v^{n+1}_{*,j} + \mathbf{P_j}^{n+1}(\varphi,\psi) v^{n+1}_{*,j-1} + q^{n+1}_{*,j}(\varphi,\psi)\right]_i \ . \tag{3.38}$$

For notational completeness, we adopt the convention that $v^{n+1}_{*,0} = \mathbf{0}$. Note that $\mathbf{A_j}^{n+1} = \mathbf{A_j}^{n+1}(\varphi,\psi)$, $\mathbf{P_j}^{n+1} = \mathbf{P_j}^{n+1}(\varphi,\psi)$, $q^{n+1}_{*,j} = q^{n+1}_{*,j}(\varphi,\psi)$, through the local optimization problem (3.38). An exception occurs at $j = 1$, where $\mathbf{P_1}^{n+1}$ is a zero matrix and $q^{n+1}_{*,1}$ is a zero vector. $\mathbf{A_1}$ no longer depends on the value of the control variables $\{\varphi,\psi\}$ or time $n\Delta\tau$ due to the boundary condition at $A = 0$. The matrix form of the degenerate equations becomes

$$\mathbf{A_1} v^{n+1}_{*,1} = \frac{1}{\Delta\tau} v^n_{*,1} \tag{3.39}$$

on the boundary $j = 1$ (i.e. $A = 0$).

Define matrix $\mathbf{Z}^n$ such that

$$\mathbf{Z}^n = \begin{pmatrix} \mathbf{A}^n_{\mathbf{1}} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ -\mathbf{P}^n_{\mathbf{2}} & \mathbf{A}^n_{\mathbf{2}} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{P}^n_{\mathbf{3}} & \mathbf{A}^n_{\mathbf{3}} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ddots & \mathbf{A}^n_{\mathbf{j}_{\max}-\mathbf{1}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & -\mathbf{P}^n_{\mathbf{j}_{\max}} & \mathbf{A}^n_{\mathbf{j}_{\max}} \end{pmatrix} \tag{3.40}$$

We can write equation (3.22) as

$$\mathbf{Z}^{n+1}\mathbf{v}^{n+1} = \frac{1}{\Delta\tau}\mathbf{v}^n + \mathbf{q}^{n+1} \ , \tag{3.41}$$

where

$$\{\varphi^{n+1}_{i,j}, \psi^{n+1}_{i,j}\} = \underset{\substack{\varphi\in\{0,1\},\psi\in\{0,1\} \\ \varphi\psi=0}}{\arg\max} \left[\left[\tfrac{1}{\Delta\tau}\mathbf{I} - \mathbf{Z}^{n+1}(\varphi,\psi)\right]\mathbf{v}^{n+1} + \mathbf{q}^{n+1}(\varphi,\psi) \right]_i . \tag{3.42}$$

Remark 3.3.1. *The matrix* $\mathbf{Z}^n$ *is an* M *matrix, for it is strictly diagonally dominant with non-positive off-diagonal entries [53].*

Remark 3.3.2. *We remind the reader that a matrix* $\mathbf{A}$ *is an* M *matrix if the offdiagonals are nonpositive,* $\mathbf{A}$ *is nonsingular, and* $\mathbf{A}^{-1} \geq 0$*. A sufficient condition for a matrix to be an* M *matrix is that the offdiagonals are nonpositive, and each row sum is strictly positive (i.e. strictly diagonally dominant) [53].*

3.4 Convergence of the Penalized PDE Discretization

From [5, 3] we find that any scheme which is monotone, consistent (in the viscosity sense) and l_∞ stable converges to the viscosity solution. In the following sections, we will verify each of these properties in turn for the penalty scheme.

It will be convenient at this point to introduce the following definitions

$$\Delta W_{\max} = \max_i (W_{i+1} - W_i) \qquad \Delta W_{\min} = \min_i (W_{i+1} - W_i)$$
$$\Delta A_{\max} = \max_j (A_{j+1} - A_j) \qquad \Delta A_{\min} = \min_j (A_{j+1} - A_j).$$

3.4.1 Stability

The stability of scheme (3.22), (3.27)-(3.32), is a direct result of the following Lemma:

Lemma 3.4.1 (Stability)**.** *If the discretized equation* (3.22) *satisfies the positive coefficient condition* (3.24)*, then scheme* (3.22), (3.27)-(3.32)*, satisfies*

$$e^{-\eta\tau^n} W_i \le V_{i,j}^n \le W_i + A_j \tag{3.43}$$

for $0 \le n \le N$ *as* $\Delta\tau \to 0$, $\Delta W_{\min} \to 0$, $\Delta A_{\min} \to 0$.

Proof. Define a discrete bounding function $B_{i,j}^n$ such that

$$B_{i,j}^n = W_i + A_j \ . \tag{3.44}$$

Define vectors

$$b_{*,j}^n = \left(B_{1,j}^n, B_{2,j}^n, \ldots, B_{i_{\max},j}\right) \quad ; \quad \mathbf{b}^n = \left(b_{*,1}^n, b_{*,2}^n, \ldots, b_{*,j_{\max}}^n\right)' \tag{3.45}$$

Then, some straightforward (but lengthy) algebra shows that

$$\mathbf{Z}^{n+1}(\mathbf{b}^{n+1} - \mathbf{v}^{n+1}) \;=\; \frac{1}{\Delta\tau}[\mathbf{b}^n - \mathbf{v}^n] + \mathbf{h}^{n+1}(\mathbf{v}^{n+1}) \ , \tag{3.46}$$

where

$$[\mathbf{h}^{n+1}]_{i,j} \;=\; \begin{cases} \eta W_i + rA_j + (\varphi_{i,j}^{n+1} G + \frac{\psi_{i,j}^{n+1}}{\varepsilon})(1-\delta_{i,0}) + \psi_{i,j}^{n+1}\kappa(1/\varepsilon - G) & i < i_{\max}, j > 1 \ , \\ \eta(W_i + A_j) & \text{otherwise} \ , \end{cases} \tag{3.47}$$

where $\delta_{i,j}$ is the Kronecker delta. Since $1/\varepsilon > G$, then $\mathbf{h}^{n+1} \geq 0$. Assume $\mathbf{b}^n - \mathbf{v}^n \geq 0$, then, since $\mathbf{Z}^{n+1}$ is an M matrix, $\mathbf{b}^{n+1} - \mathbf{v}^{n+1} \geq 0$. Note from the initial condition (3.27), we have $\mathbf{b}^0 - \mathbf{v}^0 \geq 0$. Hence

$$V_{i,j}^n \;\leq\; W_i + A_j \quad , \quad \forall n \ . \tag{3.48}$$

For the lower bound, define the lower bounding grid function

$$L_{i,j}^n = \frac{W_i}{(1+\eta\Delta\tau)^n} \ . \tag{3.49}$$

Following a similar approach as used for the upper bound, we find that

$$V_{i,j}^n \geq \frac{W_i}{(1+\eta\Delta\tau)^n} > e^{-\eta\tau^n} W_i \ . \tag{3.50}$$

□

Remark 3.4.1. *For a given finite domain Ω^L, bound (3.43) clearly implies that $\|V^n\|_\infty$ is bounded. However, note that for fixed (W, A, τ), bound (3.43) is independent of $W_{\max}$,* *which is an important property if we solve the problem in Definition (2.3.1) on a sequence of larger domains.*

3.4.2 Consistency

This section shows that the discretization scheme (3.22), (3.27)-(3.30) is consistent with the singular control GMWB pricing problem as defined in Definition 2.3.2.

Consider the discretized equation (3.22), and the associated discretized boundary conditions (3.27)-(3.32). We make the following assumption regarding the mesh/time-step size.

Assumption 3.4.1. *There exists a mesh/time-step size parameter h such that*

$$h = \frac{\Delta W_{\max}}{C_1} = \frac{\Delta A_{\max}}{C_2} = \frac{\Delta \tau}{C_3} = \frac{\varepsilon}{C_4}, \tag{3.51}$$

where C_i ($i = 1, 2, 3, 4$) are positive constants independent of h.

Equation (3.22) is equivalent to equation (3.16), which can be re-written as

$$\frac{V_{i,j}^{n+1} - V_{i,j}^{n}}{\Delta \tau} - \mathcal{L}_{\mathcal{G}}^{h} V_{i,j}^{n+1} - \max\left(G \max(\mathcal{F}^h V_{i,j}^{n+1}, 0), \frac{(\mathcal{F}^h V_{i,j}^{n+1} - \kappa)}{\varepsilon} + \kappa G \right) = 0 \ , \tag{3.52}$$

or equivalently

$$\min \Bigg[\frac{V_{i,j}^{n+1} - V_{i,j}^{n}}{\Delta \tau} - \mathcal{L}_{\mathcal{G}}^{h} V_{i,j}^{n+1} - \kappa G - \frac{1}{\varepsilon}(\mathcal{F}^h V_{i,j}^{n+1} - \kappa), \\ \frac{V_{i,j}^{n+1} - V_{i,j}^{n}}{\Delta \tau} - \mathcal{L}_{\mathcal{G}}^{h} V_{i,j}^{n+1} - G \max(\mathcal{F}^h V_{i,j}^{n+1}, 0) \Bigg] = 0 \ . \tag{3.53}$$

Equation (3.53) implies that one of the following holds with equality:

$$\frac{V_{i,j}^{n+1} - V_{i,j}^{n}}{\Delta\tau} - \mathcal{L}_{\mathcal{G}}^{h} V_{i,j}^{n+1} - \kappa G - \frac{1}{\varepsilon}(\mathcal{F}^{h} V_{i,j}^{n+1} - \kappa) \geq 0, \tag{3.54}$$

$$\frac{V_{i,j}^{n+1} - V_{i,j}^{n}}{\Delta\tau} - \mathcal{L}_{\mathcal{G}}^{h} V_{i,j}^{n+1} - G \max(\mathcal{F}^{h} V_{i,j}^{n+1}, 0) \geq 0. \tag{3.55}$$

Since $\varepsilon > 0$, equation (3.54) is equivalent to

$$\varepsilon\left(\frac{V_{i,j}^{n+1} - V_{i,j}^{n}}{\Delta\tau} - \mathcal{L}_{\mathcal{G}}^{h} V_{i,j}^{n+1} - \kappa G\right) + (\kappa - \mathcal{F}^{h} V_{i,j}^{n+1}) \geq 0. \tag{3.56}$$

As a result, equations (3.55) and (3.56) can be combined to give

$$\begin{aligned}
\mathcal{H}_{i,j}^{n+1} \equiv & \mathcal{H}_{i,j}^{n+1}\left(h, V_{i,j}^{n+1}, \left\{V_{a,b}^{n+1}\right\}_{\substack{a\neq i \\ \text{or } b\neq j}}, V_{i,j}^{n}\right) \\
= & \min\Bigg[\frac{V_{i,j}^{n+1} - V_{i,j}^{n}}{\Delta\tau} - \mathcal{L}_{\mathcal{G}}^{h} V_{i,j}^{n+1} - G \max(\mathcal{F}^{h} V_{i,j}^{n+1}, 0), \\
& \quad \varepsilon\left(\frac{V_{i,j}^{n+1} - V_{i,j}^{n}}{\Delta\tau} - \mathcal{L}_{\mathcal{G}}^{h} V_{i,j}^{n+1} - \kappa G\right) + (\kappa - \mathcal{F}^{h} V_{i,j}^{n+1})\Bigg] = 0,
\end{aligned} \tag{3.57}$$

where $\left\{V_{a,b}^{n+1}\right\}_{\substack{a\neq i \\ \text{or } b\neq j}}$ is the set of values $V_{a,b}^{n+1}$, $a = 1, 2, \ldots, i_{\max}$ and $b = 1, 2, \ldots, j_{\max}$, $(a, b) \neq (i, j)$. We can re-formulate the discretization scheme (3.22), (3.27)-(3.32) at node (W_i, A_j, τ^{n+1}) into one equation:

$$\begin{aligned}
& \mathcal{G}_{i,j}^{n+1}\left(h, V_{i,j}^{n+1}, \left\{V_{a,b}^{n+1}\right\}_{\substack{a\neq i \\ \text{or } b\neq j}}, V_{i,j}^{n}\right) \\
= & \begin{cases}
\mathcal{H}_{i,j}^{n+1}, & (W_i, A_j, \tau^{n+1}) \in \Omega_{\text{in}} \cup \Omega_{W_0} \cup \Omega_{A_0}, \\
V_{i,j}^{n+1}(1 + \eta\Delta\tau) - V_{i,j}^{n}, & (W_i, A_j, \tau^{n+1}) \in \Omega_{W_{\max}} \\
V_{i,j}^{n+1} - \max\left[W_i, (1-\kappa)A_j\right], & (W_i, A_j, \tau^{n+1}) \in \Omega_{\tau^0}.
\end{cases} \\
= & 0.
\end{aligned} \tag{3.58}$$

We follow here the definition of consistency in the viscosity sense [3]. For an excellent overview of this topic, we refer the reader to [33].

Definition 3.4.1 (Consistency)**.** *For any smooth test function $\phi(W, A, \tau)$ with $\phi_{i,j}^{n+1} = \phi(W_i, A_j, \tau^{n+1})$, having bounded derivatives of all orders with respect to W, A, and τ, assuming the mesh/time-step size parameter h satisfies Assumption 3.4.1, the numerical scheme $\mathcal{G}_{i,j}^{n+1}\left(h, \phi_{i,j}^{n+1}, \left\{\phi_{a,b}^{n+1}\right\}_{\substack{a\neq i \\ or\ b\neq j}}, \left\{\phi_{i,j}^{n}\right\}\right)$ is consistent if $\forall \hat{\mathbf{x}} = (\hat{W}, \hat{A}, \hat{\tau}) \in \Omega^L$, $\forall \mathbf{x}_{i,j}^{n+1} = (W_i, A_j, \tau^{n+1}) \in \Omega^L$, the following two inequalities hold.*

$$\limsup_{\substack{\mathbf{x}_{i,j}^{n+1} \to \hat{\mathbf{x}} \\ h \to 0 \\ \hbar \to 0}} \mathcal{G}_{i,j}^{n+1}\left(h, \phi_{i,j}^{n+1} + \hbar, \left\{\phi_{a,b}^{n+1} + \hbar\right\}_{\substack{a\neq i \\ or\ b\neq j}}, \left\{\phi_{i,j}^{n} + \hbar\right\}\right) \leq (F_{\Omega^L})^*(\phi(\hat{\mathbf{x}})), \tag{3.59}$$

$$\liminf_{\substack{\mathbf{x}_{i,j}^{n+1} \to \hat{\mathbf{x}} \\ h \to 0 \\ \hbar \to 0}} \mathcal{G}_{i,j}^{n+1}\left(h, \phi_{i,j}^{n+1} + \hbar, \left\{\phi_{a,b}^{n+1} + \hbar\right\}_{\substack{a\neq i \\ or\ b\neq j}}, \left\{\phi_{i,j}^{n} + \hbar\right\}\right) \geq (F_{\Omega^L})_*(\phi(\hat{\mathbf{x}})). \tag{3.60}$$

where $(F_{\Omega^L})^*$ and $(F_{\Omega^L})_*$ are the upper and lower semicontinuous envelopes of F_{Ω^L}. Before proving consistency, we shall need an intermediate result, which is given in the following Lemma.

Lemma 3.4.2 (Local consistency)**.** *Suppose the mesh size and the time-step parameter satisfy Assumption 3.4.1, then for any smooth function $\phi(W, A, \tau)$ having bounded derivatives of all orders in $(W, A, \tau) \in \Omega^L$, with $\phi_{i,j}^{n+1} = \phi(W_i, A_j, \tau^{n+1})$, and for $h, \hbar$ sufficiently* *small, we have that*

$$\mathcal{G}_{i,j}^{n+1}\left(h, \phi_{i,j}^{n+1}+\hbar, \left\{\phi_{a,b}^{n+1}+\hbar\right\}_{\substack{a\neq i \\ or\ b\neq j}}, \left\{\phi_{i,j}^{n}+\hbar\right\}\right)$$
$$= \begin{cases} F_{in}\phi_{i,j}^{n+1} + O(h) + O(\hbar) \ , & (W_i, A_j, \tau^{n+1}) \in \Omega_{in} \ , \\ F_{W_0}\phi_{i,j}^{n+1} + O(h) + O(\hbar) \ , & (W_i, A_j, \tau^{n+1}) \in \Omega_{W_0} \ , \\ F_{A_0}\phi_{i,j}^{n+1} + O(h) + O(\hbar) \ , & (W_i, A_j, \tau^{n+1}) \in \Omega_{A_0} \ , \\ F_{W_{\max}}\phi_{i,j}^{n+1} + O(h) + O(\hbar) \ , & (W_i, A_j, \tau^{n+1}) \in \Omega_{W_{\max}} \ , \\ F_{\tau^0}\phi_{i,j}^{n+1} + O(\hbar) \ , & (W_i, A_j, \tau^{n+1}) \in \Omega_{\tau^0} \ , \end{cases} \tag{3.61}$$

where $\hbar$ is a constant independent of $\mathbf{x}_{i,j}^{n+1}$.

Proof. Before proving the Lemma, we first define the following notations for the operators applied to test functions, evaluated at node (W_i, A_j, τ^{n+1}).

$$\mathcal{L}_{\mathcal{G}}\phi_{i,j}^{n+1} \equiv \mathcal{L}_{\mathcal{G}}\phi(W_i, A_j, \tau^{n+1}) \ , \qquad \mathcal{F}\phi_{i,j}^{n+1} \equiv \mathcal{F}\phi(W_i, A_j, \tau^{n+1}) \ ,$$
$$(\phi_W)_{i,j}^{n+1} \equiv \phi_W(W_i, A_j, \tau^{n+1}) \ , \qquad (\phi_A)_{i,j}^{n+1} \equiv \phi_A(W_i, A_j, \tau^{n+1}) \ ,$$
$$(\phi_\tau)_{i,j}^{n+1} \equiv \phi_\tau(W_i, A_j, \tau^{n+1}) \ .$$

By definitions of discrete operators $\mathcal{L}_{\mathcal{G}}^h$ and $\mathcal{F}^h$ in (3.15), it can be easily verified that

$$\mathcal{L}_{\mathcal{G}}^h(\phi_{i,j}^{n+1}+\hbar) = \mathcal{L}_{\mathcal{G}}^h\phi_{i,j}^{n+1} - r\hbar \ , \tag{3.62}$$
$$\mathcal{F}^h(\phi_{i,j}^{n+1}+\hbar) = \mathcal{F}^h\phi_{i,j}^{n+1} \ . \tag{3.63}$$

From Taylor series expansions and the last two equations above, we have that

$$\mathcal{L}_{\mathcal{G}}^{h}(\phi_{i,j}^{n+1}+\hbar)=\mathcal{L}_{\mathcal{G}}\phi_{i,j}^{n+1}-r\hbar+O(\Delta W_{\max}), \tag{3.64}$$

$$\mathcal{F}^{h}(\phi_{i,j}^{n+1}+\hbar)=\mathcal{F}\phi_{i,j}^{n+1}+O(\Delta W_{\max})+O(\Delta A_{\max}), \tag{3.65}$$

$$\frac{\phi_{i,j}^{n+1}-\phi_{i,j}^{n}}{\Delta\tau}=(\phi_{\tau})_{i,j}^{n+1}+O(\Delta\tau). \tag{3.66}$$

By using equation (3.57) together with the discretization error estimation in the last three equations above, and the inequality $|\min(x,y)-\min(a,b)|\leq\max(|x-a|,|y-b|)$, we can see for nodes $(W_i, A_j, \tau^{n+1})\in\Omega_{\text{in}}$:

$$\begin{aligned}
&\left|\mathcal{G}_{i,j}^{n+1}\left(h,\phi_{i,j}^{n+1}+\hbar,\left\{\phi_{a,b}^{n+1}+\hbar\right\}_{\substack{a\neq i\\ \text{or } b\neq j}},\left\{\phi_{i,j}^{n}+\hbar\right\}\right)-F_{\text{in}}\phi_{i,j}^{n+1}\right|\\
&\leq\max\Bigg[\bigg|\frac{\phi_{i,j}^{n+1}-\phi_{i,j}^{n}}{\Delta\tau}-\mathcal{L}_{\mathcal{G}}^{h}(\phi_{i,j}^{n+1}+\hbar)-G\max\left[\mathcal{F}^{h}(\phi_{i,j}^{n+1}+\hbar),0\right]\\
&\qquad-\left((\phi_{\tau})_{i,j}^{n+1}-\mathcal{L}_{\mathcal{G}}\phi_{i,j}^{n+1}-G\max\left[\mathcal{F}\phi_{i,j}^{n+1},0\right]\right)\bigg|,\\
&\qquad\bigg|\varepsilon\left(\frac{\phi_{i,j}^{n+1}-\phi_{i,j}^{n}}{\Delta\tau}-\mathcal{L}_{\mathcal{G}}^{h}(\phi_{i,j}^{n+1}+\hbar)-\kappa G\right)\\
&\qquad+\left(\mathcal{F}\phi_{i,j}^{n+1}-\mathcal{F}^{h}(\phi_{i,j}^{n+1}+\hbar)\right)\bigg|\Bigg]\\
&\leq\max\Bigg[\bigg|O(\Delta\tau)+O(\Delta W_{\max})+r\hbar+G\left|\mathcal{F}^{h}(\phi_{i,j}^{n+1}+\hbar)-\mathcal{F}\phi_{i,j}^{n+1}\right|\bigg|,\\
&\qquad\bigg|O(\Delta W_{\max})+O(\Delta A_{\max})\\
&\qquad+\varepsilon\left(\frac{\phi_{i,j}^{n+1}-\phi_{i,j}^{n}}{\Delta\tau}-\mathcal{L}_{\mathcal{G}}^{h}(\phi_{i,j}^{n+1}+\hbar)-\kappa G\right)\bigg|\Bigg]
\end{aligned}$$

$$
\begin{aligned}
= \max\Bigg[& \left| O(\Delta\tau) + O(\Delta W_{\max}) + O(\Delta W_{\max} + \Delta A_{\max}) + r\hbar \right|, \\
& \Bigg| O(\Delta W_{\max}) + O(\Delta A_{\max}) \\
& + \varepsilon \left((\phi_\tau)_{i,j}^{n+1} - \mathcal{L}_{\mathcal{G}} \phi_{i,j}^{n+1} + r\hbar - \kappa G + O(\Delta\tau) \right) + O(\Delta W_{\max}) \Big) \Bigg| \Bigg].
\end{aligned} \tag{3.67}
$$

By Assumption 3.4.1 and the inequality (3.67), we obtain

$$
\mathcal{G}_{i,j}^{n+1}\left(h, \phi_{i,j}^{n+1} + \hbar, \left\{ \phi_{a,b}^{n+1} + \hbar \right\}_{\substack{a \neq i \\ \text{or } b \neq j}}, \left\{ \phi_{i,j}^{n} + \hbar \right\} \right) = F_{\text{in}} \phi_{i,j}^{n+1} + O(h) + O(\hbar). \tag{3.68}
$$

This proves the first equation in (3.61). The rest of the equations in (3.61) are proved by following similar arguments. □

Lemma 3.4.3 (Consistency)**.** *Assume that all conditions in Lemma 3.4.2 are satisfied, then scheme* (3.58) *is consistent according to Definition 3.4.1.*

Remark 3.4.2 (Consistency in the viscosity sense)**.** *Given the local consistency result in Lemma 3.4.2, it is straightforward to show that scheme* (3.58) *is consistent in the sense of Definition 3.4.1. We will include these steps here for the convenience of the reader, although this is mainly an exercise in notational manipulation. In general, however, we may not be able to get local consistency everywhere. As an example, in [11], there are nodes in strips near the domain boundaries where local consistency is not achieved. In this case, the more relaxed definition of consistency in the viscosity sense is particularly useful, and the final steps required to prove consistency are non-trivial.*

Proof. First we prove that the inequality (3.59) holds. From the definition of lim sup, there exists sequences $i_k, j_k, n_k, \hbar_k$ and h_k such that

$$\text{as} \quad k \to \infty, \; \mathbf{x}_{i_k,j_k}^{n_k+1} \to \hat{\mathbf{x}}, \; \hbar_k \to 0, \; h_k \to 0, \tag{3.69}$$

and

$$\begin{aligned} &\limsup_{k\to\infty} \mathcal{G}_{i,j}^{n+1}\left(h_k, \phi_{i_k,j_k}^{n_k+1} + \hbar_k, \left\{\phi_{a_k,b_k}^{n_k+1} + \hbar_k\right\}_{\substack{a_k \neq i_k \\ \text{or } b_k \neq j_k}}, \left\{\phi_{i_k,j_k}^{n_k} + \hbar_k\right\}\right) \\ =&\limsup_{\substack{\mathbf{x}_{i,j}^{n+1} \to \hat{\mathbf{x}} \\ h\to 0 \\ \hbar \to 0}} \mathcal{G}_{i,j}^{n+1}\left(h, \phi_{i,j}^{n+1} + \hbar, \left\{\phi_{a,b}^{n+1} + \hbar\right\}_{\substack{a\neq i \\ \text{or } b\neq j}}, \left\{\phi_{i,j}^{n} + \hbar\right\}\right) \end{aligned} \tag{3.70}$$

From Lemma 3.4.2, we have for k sufficiently large, there exist positive constants C_1, C_2 independent of k such that

$$\begin{aligned} &\left|\mathcal{G}_{i,j}^{n+1}\left(h_k, \phi_{i_k,j_k}^{n_k+1} + \hbar_k, \left\{\phi_{a_k,b_k}^{n_k+1} + \hbar_k\right\}_{\substack{a_k \neq i_k \\ \text{or } b_k \neq j_k}}, \left\{\phi_{i_k,j_k}^{n_k} + \hbar_k\right\}\right) - F_{\Omega^L}\phi_{i_k,j_k}^{n_k+1}\right| \\ &\leq C_1 h_k + C_2 \hbar_k \quad ; \quad (W_{i_k}, A_{j_k}, \tau^{n_k+1}) \in \Omega^L \; . \end{aligned} \tag{3.71}$$

Remark 3.4.3. *Suppose, for example, that* $\hat{\mathbf{x}} \in \Omega_{W_0}$*. Note that for* k *sufficiently large,* $\mathbf{x}_{i_k,j_k}^{n_k+1}$ *can be in either* Ω_{W_0} *or* Ω_{in}*. However, in each case, from Lemma 3.4.2, we have that inequality (3.71) holds. This is a consequence of the definition of* F_{Ω^L}*.*

From equations (3.70) and (3.71), we obtain

$$\begin{aligned} &\limsup_{\substack{\mathbf{x}_{i,j}^{n+1} \to \hat{\mathbf{x}} \\ h\to 0 \\ \hbar \to 0}} \mathcal{G}_{i,j}^{n+1}\left(h, \phi_{i,j}^{n+1} + \hbar, \left\{\phi_{a,b}^{n+1} + \hbar\right\}_{\substack{a\neq i \\ \text{or } b\neq j}}, \left\{\phi_{i,j}^{n} + \hbar\right\}\right) \\ \leq &\limsup_{k\to\infty} F_{\Omega^L}\phi_{i_k,j_k}^{n_k+1} + \limsup_{k\to\infty} \left[C_1 h_k + C_2 \hbar_k\right] \\ \leq &(F_{\Omega^L})^*(\phi(\hat{\mathbf{x}})) \; . \end{aligned} \tag{3.72}$$

Similarly,

$$
\begin{aligned}
&\liminf_{\substack{\mathbf{x}_{i,j}^{n+1}\to\hat{\mathbf{x}}\\ h\to 0\\ \hbar\to 0}} \mathcal{G}_{i,j}^{n+1}\left(h, \phi_{i,j}^{n+1}+\hbar, \left\{\phi_{a,b}^{n+1}+\hbar\right\}_{\substack{a\neq i\\ \text{or } b\neq j}}, \left\{\phi_{i,j}^{n}+\hbar\right\}\right)\\
\geq& \liminf_{k\to\infty} F_{\Omega^L}\phi_{i_k,j_k}^{n_k+1} + \liminf_{k\to\infty}\left[-C_1 h_k - C_2\hbar_k\right]\\
\geq& (F_{\Omega^L})_*(\phi(\hat{\mathbf{x}})).
\end{aligned}
\tag{3.73}
$$

□

3.4.3 Monotonicity

Definition 3.4.2 (Monotonicity)**.** *The numerical scheme* $\mathcal{G}_{i,j}^{n+1}\left(h, V_{i,j}^{n+1}, \left\{V_{a,b}^{n+1}\right\}_{\substack{a\neq i\\ or\ b\neq j}}, V_{i,j}^{n}\right)$ *in* (3.58) *is monotone if for all* $Y_{i,j}^n \geq X_{i,j}^n, \forall i, j, n$

$$
\mathcal{G}_{i,j}^{n+1}\left(h, V_{i,j}^{n+1}, \left\{Y_{a,b}^{n+1}\right\}_{\substack{a\neq i\\ or\ b\neq j}}, Y_{i,j}^{n}\right) \leq \mathcal{G}_{i,j}^{n+1}\left(h, V_{i,j}^{n+1}, \left\{X_{a,b}^{n+1}\right\}_{\substack{a\neq i\\ or\ b\neq j}}, X_{i,j}^{n}\right). \tag{3.74}
$$

Lemma 3.4.4 (Monotonicity)**.** *If scheme (3.58) satisfies the positive coefficient condition (3.24) then it is monotone according to Definition 3.4.2.*

Proof. This is easily done using the same steps as in [26]. □

3.4.4 Convergence in Ω^L

Theorem 3.4.1 (Convergence to the viscosity solution)**.** *Assume that scheme (3.58) satisfies all the conditions required for Lemmas 3.4.1, 3.4.3, and 3.4.4, and that Assumption 2.3.1 holds, then the scheme (3.58) converges to the unique, continuous viscosity solution* *of the GMWB problem given in Definition 2.3.2, at any point in $\Omega_{in} \cup \Gamma$ (see Definition of Γ in Assumption 2.3.1).*

Proof. Since the scheme is monotone, consistent and pointwise stable, this follows from the results in [5]. □

Remark 3.4.4. *Note that since we have assumed that strong comparison holds only in $\Omega_{in} \cup \Gamma$, then we can guarantee uniqueness and continuity only in $\Omega_{in} \cup \Gamma$.*

3.4.5 Convergence in Ω^∞

The asymptotic form of the solution for $W \to \infty$ is given in [18], which we impose at finite $W_{\max}$ through boundary condition (2.15). This, of course, causes an error due to finite $W_{\max}$ (see equation (2.14)).

Consider a sequence of converged viscosity solutions $(V(W, A, \tau))^k$, which satisfy Definition 2.3.2 on the sequence of grids $(\Omega^L)^k$, $k \to \infty$, with $W_{\max}^k > W_{\max}^{k-1}$. In [4], the limiting problem of convergence to the viscosity solution on unbounded domains with quadratic growth in the solution is discussed. It is possible to appeal to the results in [4] to show convergence as $(\Omega^L)^k \to \Omega^\infty$. However we can use a simpler approach for problem at hand.

For simplicity, and to avoid notational complexity, we consider only points in $(\Omega_{in})^k$ in the following, since from Theorem 3.4.1 we are ensured of convergence at least to points in $(\Omega_{in})^k$.

We will use the following elementary Lemmas.

Lemma 3.4.5 (Bounds on solution on $(\Omega_{in})^k$)**.** *The converged viscosity solution on each domain $(\Omega_{in})^k$ has the bounds*

$$e^{-\eta\tau} W \le (V(W, A, \tau))^k \le W + A \quad . \tag{3.75}$$

Proof. Since the discrete solution satisfies the bounds in Lemma 3.4.1, independent of h, $W_{\max}$, we take the limit as $h \to 0$, and hence the viscosity solution satisfies these same bounds. □

Lemma 3.4.6. *The following bound holds*

$$(V(W, A, \tau))^{k+1} \;\; \geq \;\; (V(W, A, \tau))^k \;\; ; \;\; (W, A, \tau) \in (\Omega_{in})^k \;\; . \tag{3.76}$$

Proof. We can regard $(V(W, A, \tau))^{k+1}$ on domain $(\Omega^L)^k$, as the solution to the GMWB pricing problem on $(\Omega^L)^k$, but with a known boundary condition at $W = W^k_{\max}$, which in general is not the same boundary condition as used for $V(W, A, \tau))^k$. From Lemma 3.4.5, we have that

$$(V(W^k_{\max}, A, \tau))^{k+1} \geq e^{-\eta\tau} W^k_{\max} = (V(W^k_{\max}, A, \tau))^k \; . \tag{3.77}$$

Hence $(V(W^k_{\max}, A, \tau))^{k+1}$ and $(V(W^k_{\max}, A, \tau))^k$ are solutions to the same PDE and boundary conditions, with the exception of the boundary condition at $W = W^k_{\max}$, which satisfies equation (3.77). Consider two discrete solutions $(V(W, A, \tau))^k_h, (V(W, A, \tau))^{k+1}_h$, defined on the same set of nodes in $(\Omega^L)^k$, and assume that the discretization satisfies all the conditions required for Theorem 3.4.1. Then, from Theorem 5.2 in [26], we have that $(V(W, A, \tau))^{k+1}_h \geq (V(W, A, \tau))^k_h$ at all the nodes. Take the limit as $h \to 0$, and noting that $(V(W, A, \tau))^{k+1}_h \to (V(W, A, \tau))^{k+1}$ and $(V(W, A, \tau))^k_h \to V(W, A, \tau))^k$, we obtain result (3.76).

□

Theorem 3.4.2 (Convergence in Ω^∞)**.** *Consider the sequence of grids* $(\Omega^L)^k$, *with* $W^{k+1}_{\max} >$

$W_{\max}^k$ and

$$\lim_{k\to\infty} (\Omega^L)^k \;=\; \Omega^\infty \;. \tag{3.78}$$

For any fixed point $(W, A, \tau) \in (\Omega_{in})^\infty$ we have that the sequence $(V(W, A, \tau))^k$ converges to a unique value $(V(W, A, \tau))^\infty$ as $k \to \infty$.

Proof. Given a fixed point (W, A, τ), from Lemma 3.4.6 we have that the solution is a non-decreasing function of the domain index k. But from Lemma 3.4.5, the solution is locally upper bounded independent of the domain index k. Hence the sequence $(V(W, A, \tau))^k, k \to \infty$ is bounded and non-decreasing, and thus converges to a limit $(V(W, A, \tau))^\infty$. Consider another set of increasing domains $(\hat{\Omega}^L)^k$. Suppose this set of domains converges to a value

$$(\hat{V}(W, A, \tau))^\infty > (V(W, A, \tau))^\infty \quad . \tag{3.79}$$

But, applying Lemma 3.4.6 to subsequences of $(\Omega^L)^k$ and $(\hat{\Omega}^L)^k$ leads to a contradiction, hence the limit $(V(W, A, \tau))^\infty$ is unique. □

Remark 3.4.5. *We apply scheme (3.58) to a sequence of problems with smaller h, for fixed $W_{\max}$. We then increase $W_{\max}$ and repeat the process. Since we use unequally spaced grids, it is computationally inexpensive to choose a large $W_{\max}$, hence the process of determining the limit $W_{\max} \to \infty$ is rapidly convergent, in practice.*

3.5 Summary

This chapter focuses on using a penalty method to discretize the HJB PDEs resulting from pricing GMWB as a singular control problem. The main results are

- We informally derive the HJB PDE of (2.6), which gives some intuition for the penalty method numerical scheme.

- The derivative terms are discretized by using standard three point finite difference with maximal use of central differencing on the first derivative term while maintaining a positive coefficient condition [44].

- We give a rigorous proof that on a finite domain, the discretization is monotone, consistent and stable, hence assuming that a strong comparison property holds, we can guarantee the convergence to the viscosity solution as the mesh size parameter $h \to 0$.

- We show that as $W_{\max} \to \infty$, at any fixed point (W, A, τ), the discretization converges to a unique limit. This proof makes use of the tight stability bounds in Lemma 3.4.1.

Chapter 4

Direct Control Method

This chapter discusses a direct control method for solving the HJB PDE in (2.6). Section 4.1 introduces the scaled direct control form of the GMWB pricing problem. Section 4.2 presents the discretized scaled direct control form of the equation. Section 4.3 presents the matrix form of the discrete equation. In Section 4.4, we discuss the convergence of the numerical scheme and prove the stability of the direct control method. We finally summarize the main results of this chapter in Section 4.5.

4.1 The Scaled Direct Control Form

A direct control technique was previously suggested for solving American option type problems in [9, 31]. Similarly, for the GMWB problem, one can simply discretize the control form of equation (2.6)

$$\min_{\substack{\varphi\in\{0,1\},\psi\in\{0,1\}\\ \varphi\psi=0}} \left[\psi(\kappa - \mathcal{F}V) + (1-\psi)(V_\tau - \mathcal{L}_\mathcal{G}V - \varphi G\mathcal{F}V) \right] = 0 \ . \tag{4.1}$$

Observe that the term $\kappa - \mathcal{F}V$ is dimensionless whereas $V_\tau - \mathcal{L}_\mathcal{G}V - G\max(\mathcal{F}V, 0)$ has dimensions of *currency/time*. Hence equation (4.1) compares quantities having different units. Of course, in exact arithmetic, this is not an issue of importance. However, an iterative procedure for solution of the discretized equations will involve a test comparing two (in general) non-zero quantities. Hence scaling becomes important. Consequently, we introduce a scaling factor $\Pi > 0$ into equation (4.1)

$$\min_{\substack{\varphi\in\{0,1\},\psi\in\{0,1\}\\ \varphi\psi=0}} \bigg[\Pi\psi(\kappa - \mathcal{F}V) + (1-\psi)(V_\tau - \mathcal{L}_\mathcal{G}V - \varphi G\mathcal{F}V) \bigg] = 0 \ . \qquad (4.2)$$

Remark 4.1.1 (Scaling Factor)**.** *By introducing a scaling factor with dimension of* currency/time*, we ensure the comparison is conducted on two items with the same units. Of course, this still leaves the size of the scaling factor as arbitrary. We will exploit this fact to ensure the convergence of an iterative method in Chapter 6 with a suitable choice for* Π*.*

4.2 Discretized Equation

We will discretize equation (4.2) in the domain $\Omega_{\text{in}} \cup \Omega_{A_0} \cup \Omega_{W_0}$. We use the same unequally spaced mesh and notations as defined in Section 3.2.2 for the penalty method discretization. If we define $V_{0,j}^{n+1} = V_{i_{\max}+1,j}^{n+1} = V_{i,0}^{n+1} = 0$, then equation (4.2) can be written in the following discrete form

$$\begin{aligned}
&(1-\psi_{i,j}^{n+1})\bigg(\frac{1}{\Delta\tau}V_{i,j}^{n+1} - \mathcal{L}_\mathcal{G}^h V_{i,j}^{n+1} + \varphi_{i,j}^{n+1} G(\mathcal{D}_W^h V_{i,j}^{n+1} + \mathcal{D}_A^h V_{i,j}^{n+1})\bigg)\\
&+\Pi\ \psi_{i,j}^{n+1}(\mathcal{D}_W^h V_{i,j}^{n+1} + \mathcal{D}_A^h V_{i,j}^{n+1})\\
&= (1-\psi_{i,j}^{n+1})\frac{1}{\Delta\tau}V_{i,j}^{n} + \Pi\ \psi_{i,j}^{n+1}(1-\kappa) + (1-\psi_{i,j}^{n+1})\varphi_{i,j}^{n+1} G\\
&\qquad\qquad\qquad\qquad (W_i, A_j, \tau^{n+1}) \in \Omega_{in} \cup \Omega_{A_0} \cup \Omega_{W_0}\ , \qquad (4.3)
\end{aligned}$$

where

$$
\begin{aligned}
(\varphi_{i,j}^{n+1}, \psi_{i,j}^{n+1}) &= \underset{\substack{\varphi\in\{0,1\},\psi\in\{0,1\}\\ \varphi\psi=0}}{\arg\max} \Bigg[-\Pi\ \psi(\kappa - \mathcal{F}^h V_{i,j}^{n+1}) - (1-\psi)\bigg(\frac{V_{i,j}^{n+1} - V_{i,j}^{n}}{\Delta\tau} \\
&\qquad - \big(\mathcal{L}_{\mathcal{G}}^h V_{i,j}^{n+1} + \varphi G \mathcal{F}^h V_{i,j}^{n+1}\big)\bigg)\Bigg]_i \ , \quad (W_i, A_j, \tau^{n+1}) \in \Omega_{in} \ , \qquad (4.4) \\
(\varphi_{i,j}^{n+1}, \psi_{i,j}^{n+1}) &= (0,0) \ , \qquad (W_i, A_j, \tau^{n+1}) \in \Omega_{A_0} \ , \qquad (4.5) \\
\mathcal{D}_W V_{i,j}^{n+1} &= 0 \ , \qquad (W_i, A_j, \tau^{n+1}) \in \Omega_{W_0} \ , \qquad (4.6)
\end{aligned}
$$

and we understand that $\varphi_{i,j}^{n+1} = \psi_{i,j}^{n+1} = 0$ in Ω_{A_0} because no withdrawal is possible when $A = 0$. At $W = 0$, we have $\mathcal{D}_W V = V_W = 0$.

When $\psi_{i,j}^{n+1} = 1$, only first derivative terms appear in the equation. We use backward differencing on both $\mathcal{D}_W^h$ and $\mathcal{D}_A^h$ operators and solve the following equation:

$$
\begin{aligned}
\psi_{i,j}^{n+1}\Pi\Bigg[-\frac{1}{\Delta W_i^-} V_{i-1,j}^{n+1} + \Big(\frac{1}{\Delta W_i^-} + \frac{1}{\Delta A_j^-}\Big) V_{i,j}^{n+1} - \frac{1}{\Delta A_j^-} V_{i,j-1}^{n+1} \Bigg] &= \psi_{i,j}^{n+1}\Pi(1-\kappa) \ , \\
\psi_{i,j}^{n+1} = 1 \ , \ i = 2, \ldots, i_{\max} - 1, \ j = 2, \ldots, j_{\max}, \qquad (4.7)
\end{aligned}
$$

When $\psi_{i,j}^{n+1} = 0$, the $\mathcal{D}_A^h$ operator is still discretized by using backward difference. However since the diffusion term appears in the equation, this allows us to use a central difference for the $\mathcal{D}_W^h$ operator in the equation as long as we maintain the positive coefficient condition in (3.24). Using a similar approach as in Section 3.2.2 for the penalty method, we can write equation (4.3) for $\psi_{i,j}^{n+1} = 0$ case (by expanding $\mathcal{L}_{\mathcal{G}}^h$, $\mathcal{F}^h$ operators and backward differencing the $\mathcal{D}_A^h$ operator) in the following form

$$
\begin{aligned}
& (1-\psi_{i,j}^{n+1})\frac{(V_{i,j}^{n+1}-V_{i,j}^{n})}{\Delta\tau} \\
= \quad & (1-\psi_{i,j}^{n+1})\left[\mathcal{B}_{\varphi_{i,j}^{n+1}}^{h}V_{i,j}^{n+1}+\varphi_{i,j}^{n+1}\frac{G}{\Delta A_j^-}V_{i,j-1}^{n+1}+\varphi_{i,j}^{n+1}G\right] \ , \\
& \psi_{i,j}^{n+1}=0 \ , \ i=2,\dots,i_{\max}-1, \ j=2,\dots,j_{\max}, \ n=0,\dots,N-1 \ , \qquad (4.8)
\end{aligned}
$$

and operator $\mathcal{B}_\varphi^h$ has a similar form as the operator $\mathcal{A}_{\varphi,\psi}^h$ as in equation (3.19)

$$
\begin{aligned}
\mathcal{B}_\varphi^h V_{i,j}^n &\equiv a_{i,j}^n(\varphi)\mathcal{D}_{WW}^h V_{i,j}^n + b_{i,j}^n(\varphi)\mathcal{D}_W^h V_{i,j}^n - c_{i,j}^n(\varphi)V_{i,j}^n \\
&= \alpha_{i,j}^n(\varphi)V_{i,j}^n - \big(\alpha_{i,j}^n(\varphi)+\beta_{i,j}^n(\varphi)+c_{i,j}^n(\varphi)\big)V_{i,j}^n + \beta_{i,j}^n(\varphi)V_{i+1,j}^n
\end{aligned}
$$

but with different coefficients as follows

$$
a_{i,j}^n(\varphi)=\frac{\sigma^2}{2}W_i^2 \ , \quad b_{i,j}^n(\varphi)=(r-\eta)W_i-\varphi G \ , \quad c_{i,j}^n(\varphi)=(r+\varphi G\frac{1}{\Delta A_j^-}) \ , \qquad (4.9)
$$

and $\alpha_{i,j}^n$ and $\beta_{i,j}^n$ are computed by using the coefficient defined in (4.9). For maximal use of central differencing, we discretize the term $\mathcal{D}_W^h V_{i,j}^n$ in $\mathcal{B}_\varphi^h V_{i,j}^n$ to ensure the positive coefficient condition in (3.24). A more detailed description is given in Appendix C.

Equation (4.7) and (4.8) hold for $(W_i, A_j, \tau^{n+1}) \in \Omega_{in}$. For $(W_i, A_j, \tau^{n+1}) \in \partial\Omega_{in}$, the results are the same as those in penalty method. We refer readers to equations (3.28) - (3.32) in Section 3.2.2 for a detailed derivation and only present results here.

- For $(W_i, A_j, \tau^{n+1}) \in \Omega_{\tau^0}$,

$$
V_{i,j}^0 = \max[W_i, \ (1-\kappa)A_j] \ . \qquad (4.10)
$$

- For $(W_i, A_j, \tau^{n+1}) \in \Omega_{W_0}$

$$\alpha_{1,j}^{n+1} = \beta_{1,j}^{n+1} = 0, \; j = 2, 3, \ldots, j_{\max} \; . \tag{4.11}$$

- For $(W_i, A_j, \tau^{n+1}) \in \Omega_{A_0}$

$$\varphi_{i,1}^{n+1} = \psi_{i,1}^{n+1} = 0, \qquad i = 1, 2, \ldots, i_{\max -1} \; . \tag{4.12}$$

$$\alpha_{i,1}^{n+1} = \beta_{i,1}^{n+1} = 0, \qquad i = 1 \; . \tag{4.13}$$

- For $(W_i, A_j, \tau^{n+1}) \in \Omega_{W_{\max}}$

$$c_{i_{\max},j}^{n+1} = \eta \;\; ; \;\; \alpha_{i_{\max},j}^{n+1} = \beta_{i_{\max},j}^{n+1} = \varphi_{i_{\max},j}^{n+1} = \psi_{i_{\max},j}^{n+1} = 0; \tag{4.14}$$
$$j = 1, 2, \ldots, j_{\max} \; .$$

Consequently, for all nodes $(W_i, A_j, \tau^{n+1}) \in \Omega^L \backslash \Omega_{\tau^0}$, an equation of the forms (4.7) and (4.8) holds and we remind the reader that for notational completeness we define $V_{0,j}^{n+1} = V_{i_{\max}+1,j}^{n+1} = V_{i,0}^{n+1} = 0$.

4.3 Matrix Form of the Discretized Equations

We use a matrix form to represent the discretized equations in (4.3), which consists of equations (4.7) and (4.8), and the associated boundary conditions in (4.11)-(4.14). Define an $i_{\max} \times i_{\max}$ tridiagonal matrix $\mathbf{D_j^n}$ so that the entry on the i^{th} row and k^{th} column is defined as

$$[\mathbf{D_j^n}]_{i,k} = \begin{cases} -\psi_{i,j}^n \Pi \frac{1}{\Delta W_i^-} - (1-\psi_{i,j}^n)\alpha_{i,j}^n & \text{if } k = i-1, i = 2,\dots,i_{\max} \\ -(1-\psi_{i,j}^n)\beta_{i,j}^n & \text{if } k = i+1, i = 0,\dots,i_{\max}-1 \\ \psi_{i,j}^n \Pi (\frac{1}{\Delta W_i^-} + \frac{1}{\Delta A_j^-}) + (1-\psi_{i,j}^n)(\frac{1}{\Delta \tau} + \alpha_{i,j}^n + \beta_{i,j}^n + c_{i,j}^n) & \text{if } k = i, i = 1,\dots,i_{\max} \\ 0 & \text{otherwise .} \end{cases} \quad (4.15)$$

Define an $i_{\max} \times i_{max}$ diagonal matrix $\mathbf{L_j^n}$ so that entries on the diagonal are defined as

$$[\mathbf{L_j^n}]_{i,i} = \begin{cases} \psi_{i,j}^n \Pi \frac{1}{\Delta A_j^-} + (1-\psi_{i,j}^n)\varphi_{i,j}^n \frac{G}{\Delta A_j^-} & \text{if } i \le i_{max}-1, \\ 0 & \text{if } i = i_{max}. \end{cases} \quad (4.16)$$

Define an $i_{\max}$ length column vector $h_{*,j}^n$ such that the entry on the i^{th} row is defined as

$$[h_{*,j}^n]_i \quad = \quad \psi_{i,j}^n \Pi (1-\kappa) + (1-\psi_{i,j}^n)(\frac{1}{\Delta \tau} V_{i,j}^{n-1} + \varphi_{i,j}^n G) \ . \quad (4.17)$$

We can then write equation (4.3) as

$$\mathbf{D_j}^{n+1} v_{*,j}^{n+1} - \mathbf{L_j}^{n+1} v_{*,j-1}^{n+1} = \quad h_{*,j}^{n+1} \ , \quad (4.18)$$

where

$$\{\varphi_{i,j}^{n+1}, \psi_{i,j}^{n+1}\} = \underset{\substack{\varphi \in \{0,1\}, \psi \in \{0,1\} \\ \varphi\psi = 0}}{\arg\max} \left[-\mathbf{D_j}^{n+1}(\varphi,\psi) v_{*,j}^{n+1} + \mathbf{L_j}^{n+1}(\varphi,\psi) v_{*,j-1}^{n+1} + h_{*,j}^{n+1}(\varphi,\psi) \right]_i \ . \quad (4.19)$$

Similarly as for the matrix form of the discretized equations for the penalty method, for notational completeness, we adopt the convention that $v_{*,0}^{n+1} = \mathbf{0}$. Note that $\mathbf{D_j}^{n+1} = \mathbf{D_j}^{n+1}(\varphi, \psi)$, $\mathbf{L_j}^{n+1} = \mathbf{L_j}^{n+1}(\varphi, \psi)$, $h_{*,j}^{n+1} = h_{*,j}^{n+1}(\varphi, \psi)$, through the local optimization problem (4.19). An exception occurs at $j = 1$, where $\mathbf{L_1}^{n+1}$ is a zero matrix and $h_{*,1}^{n+1} = \frac{1}{\Delta\tau} v_{*,j}^n$. The matrix $\mathbf{D_1}$ no longer depends on the value of the control variables $\{\varphi, \psi\}$ or time $n\Delta\tau$ due to the boundary condition at $A = 0$. The matrix form of the degenerate equations becomes

$$\mathbf{D_1} v_{*,1}^{n+1} = \frac{1}{\Delta\tau} v_{*,1}^n \tag{4.20}$$

on the boundary $j = 1$ (i.e. $A = 0$).

Let

$$\mathbf{D}^n = \begin{pmatrix} \mathbf{D_1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{D_2^n} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{D_{j_{max}}^n} \end{pmatrix} , \quad \mathbf{L}^n = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ -\mathbf{L_2^n} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{L_3^n} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & -\mathbf{L_{j_{max}}^n} & \mathbf{0} \end{pmatrix} \tag{4.21}$$

and

$$\mathbf{Z}^n = \mathbf{D}^n + \mathbf{L}^n \ . \tag{4.22}$$

Let vector $\mathbf{h}^n$ be

$$\mathbf{h}^n = \big((h_{*,1}^n)', (h_{*,2}^n)', \ldots, (h_{*,j_{\max}}^n)'\big)' \ . \tag{4.23}$$

Using the the $\mathbf{v}^n$ notation as defined in 3.3, we can write equation (4.3) as

$$\mathbf{Z}^{n+1} \mathbf{v}^{n+1} = \mathbf{h}^{n+1} \ , \tag{4.24}$$

where

$$\{\varphi_{i,j}^{n+1}, \psi_{i,j}^{n+1}\} = \underset{\substack{\varphi\in\{0,1\},\psi\in\{0,1\}\\ \varphi\psi=0}}{\arg\max} \left[-\mathbf{Z}^{n+1}(\varphi,\psi)\mathbf{v}^{n+1} + \mathbf{h}^{n+1}(\varphi,\psi)\right]_i . \tag{4.25}$$

Remark 4.3.1. *It is easy to verify that matrix* $\mathbf{Z}^n$ *has non-positive off-diagonal entries. However, because when* $\psi_{i,j} = 1$*, the corresponding row sum of* $\left[\mathbf{Z}^n\right]_{i,j}$ *equals zero. This makes matrix* $\mathbf{Z}^n$ *no longer diagonally dominant. Hence it is not obvious that* $\mathbf{Z}^n$ *is an* M *matrix. In order to see that* $\mathbf{Z}^n$ *is an* M *matrix, we split the* $\mathbf{Z}^n$ *into the sum of a block diagonal matrix* $\mathbf{D}^n$ *and a lower triangular matrix* $\mathbf{L}^n$ *as defined in* (4.21)*. The row sums of* $\mathbf{D}^n$ *are strictly positive and the off -diagonals are non-positive due to a positive coefficient discretization. Hence* $\mathbf{D}^n$ *consists of diagonal blocks, each of which is a strictly diagonally dominant* M *matrix. Since* $-\mathbf{L}^n$ *is non-positive, a straightforward computation shows that* $\mathbf{Z}^n$ *is non-singular and that* $(\mathbf{Z}^n)^{-1} \geq 0$*. The matrix* $\mathbf{Z}^n$ *is therefore an* M *matrix. Appendix D uses a* 3×3 *block* $\mathbf{Z}^n$ *as an example to show* $(\mathbf{Z}^n)^{-1}$ *exists and is non-negative. Continuing in this way, it can be shown that* $(\mathbf{Z}^n)^{-1}$ *exists and is non-negative in general.*

4.4 Convergence

For the proof of the convergence of the discretization scheme in (4.7), (4.8) and associated terminal boundary conditions (4.10) - (4.14), consistency and monotonicity are relatively straightforward. The stability proof is more involved. By taking a similar approach as used to prove stability of the penalty method as in Section 3.4.1, we obtain the following results.

Lemma 4.4.1 (Stability: Direct Control Method)**.** *If the discretized equations in* (4.7) *and* (4.8) *satisfy the positive coefficient condition* (3.24)*, then scheme* (4.7), (4.8), (4.10)-(4.14), *satisfies*

$$e^{-\eta\tau^n} W_i \le V_{i,j}^n \le W_i + A_j \tag{4.26}$$

for $0 \le n \le N$ *as* $\Delta\tau \to 0$, $\Delta W_{\min} \to 0$, $\Delta A_{\min} \to 0$.

Proof. We use the same discrete upper bounding functions $B_{i,j}^n = W_i + A_j$ as defined in (3.44) and the vector $\mathbf{b}^n$ as defined in (3.45). Then, some straightforward (but lengthy) algebra shows that

$$\left[\mathbf{Z}^{n+1}(\mathbf{b}^{n+1} - \mathbf{v}^{n+1})\right]_{i,j} \;=\; (1-\psi_{i,j})\frac{1}{\Delta\tau}[\mathbf{b}^n - \mathbf{v}^n]_{i,j} + [\mathbf{g}^{n+1}]_{i,j} \ , \tag{4.27}$$

where

$$[\mathbf{g}^{n+1}]_{i,j} \;=\; \begin{cases} (1-\psi_{i,j}^{n+1})\big(\eta W_i + rA_j + (\varphi_{i,j}^{n+1} G(1-\delta_{i,0})\big) + \psi_{i,j}^{n+1}\Pi\kappa & i < i_{\max}, j > 1 \ , \\ \eta(W_i + A_j) & \text{otherwise} \ , \end{cases} \tag{4.28}$$

where $\delta_{i,j}$ is the Kronecker delta. Assume $\mathbf{b}^n - \mathbf{v}^n \ge 0$, then, $(1-\psi_{i,j}^{n+1})[\mathbf{b}^n - \mathbf{v}^n]_{i,j} \ge 0$. Since $\mathbf{g}^{n+1} \ge 0$, then $\mathbf{Z}^{n+1}(\mathbf{b}^{n+1} - \mathbf{v}^{n+1}) \ge 0$. Because $\mathbf{Z}^{n+1}$ is an M matrix, $\mathbf{b}^{n+1} - \mathbf{v}^{n+1} \ge 0$. Note from the initial condition (3.27), we have $\mathbf{b}^0 - \mathbf{v}^0 \ge 0$. Hence

$$V_{i,j}^n \;\le\; W_i + A_j \ \ , \ \ \forall n \ . \tag{4.29}$$

For the lower bound, following a similar approach as used for the upper bound, by using the lower bounding function $L_{i,j}^n = W_i/(1+\eta\Delta\tau)^n$ as in (3.49), we find that

$$V_{i,j}^n \ge \frac{W_i}{(1+\eta\Delta\tau)^n} > e^{-\eta\tau^n} W_i \ . \tag{4.30}$$

□

4.5 Summary

This chapter focuses on a direct control method to solve the GMWB pricing equation. The main results of this chapter are

- We introduce a scaled direct control method to solve the HJB VI in (2.6).
- We describe the discretization of the scaled direct control form of equations.
- We prove the stability of the discretization. Together with consistency and monotonicity (which can be proven using the same steps as in Chapter 3), and the assumption of the strong comparison principle, the scaled direct control discretization converges to the viscosity solution.

Chapter 5

Jump Diffusion

In this chapter, we extend the GMWB pricing problem by assuming the underlying asset follows a Poisson jump diffusion process. Section 5.1 formulates the GMWB pricing problem for the case of jump diffusion, which results in an HJB PIDE/VI. The resulting PIDE is then written both in a penalized form and scaled direct control form. Boundary conditions are discussed in Section 5.2. Section 5.3 describes the discretization of the resulting PIDE. Section 5.4 briefly sketches the proof of convergence of numerical schemes. Section 5.5 summarizes the main results of this chapter.

5.1 GMWB Pricing Problem with Jump Diffusion

5.1.1 Formulation of HJB PIDE

Increasing empirical evidence shows that the standard Geometric Brownian Motion is not consistent with market data [24]. A Poisson jump diffusion process [39, 15] is one of the popular and more realistic models of the risky asset stochastic process. Assuming that the risky asset S which underlies the variable annuity account (before the deduction of any proportional fees) follows a Poisson jump diffusion process as in [39], the risk neutral paths followed by S then satisfy the following stochastic differential equation

$$\frac{dS}{S} = (r - \lambda\rho)dt + \sigma dZ + (J-1)dY, \tag{5.1}$$

with r the risk free rate, dZ an increment of a standard Gauss-Wiener process, and σ the volatility associated with dZ. In the above, dY is an independent Poisson process and λ is the jump intensity representing the mean arrival rate of the Poisson process:

$$dY = \begin{cases} 0 & \text{with probability } 1 - \lambda dt \\ 1 & \text{with probability } \lambda dt \end{cases}, \tag{5.2}$$

with J a random variable representing the jump size of S. We assume that J follows a log-normal distribution $p(J)$ given by

$$p(J) = \frac{1}{\sqrt{2\pi}\zeta J} \exp\big(-\frac{(\log(J) - \nu)^2}{2\zeta^2}\big) \ , \tag{5.3}$$

with parameters ζ and ν, $\rho = E[J-1]$, where $E[\cdot]$ is the expectation, and $E[J] = \exp(\nu + \zeta^2/2)$ given the distribution function $p(J)$ in (5.3).

Generalizing the formulation in [40, 18, 30] to the case with stochastic process (5.1), the value of the guarantee $V(W, A, \tau)$ is given from the solution to the following Hamilton-Jacobi-Bellman (HJB) Variational Inequality (VI)

$$\min\Big[V_\tau - \mathcal{L}V - \lambda\mathcal{J}V - G\max(\mathcal{F}V, 0), \ \kappa - \mathcal{F}V\Big] = 0 \ . \tag{5.4}$$

where the operators $\mathcal{L}, \mathcal{F}$ and $\mathcal{J}$ are defined as

$$
\begin{aligned}
\mathcal{L}V &= \frac{\sigma^2}{2} W^2 V_{WW} + (r - \eta - \lambda\rho) W V_W - (r + \lambda) V \\
&= \frac{\sigma^2}{2} W^2 \mathcal{D}_{WW} V + (r - \eta - \lambda\rho) W \mathcal{D}_W V - (r + \lambda) V \\
\mathcal{F}V &= 1 - V_W - V_A = 1 - \mathcal{D}_W V - \mathcal{D}_A V \\
\mathcal{J}V &= \int_0^\infty V(JW, A, \tau) p(J) \, dJ
\end{aligned} \tag{5.5}
$$

while $\mathcal{D}_A, \mathcal{D}_W$ and $\mathcal{D}_{WW}$ denote the usual partial derivative operators.

5.1.2 Penalized Form

By extending the idea in Section 3.1, equation (5.4) can be reformulated in penalized form as

$$
V_\tau^\varepsilon = \mathcal{L}V^\varepsilon + \lambda \mathcal{J} V^\varepsilon + \max_{\substack{\varphi \in \{0,1\}, \psi \in \{0,1\} \\ \varphi\psi = 0}} \left[\varphi G \mathcal{F} V^\varepsilon + \psi \left(\frac{(\mathcal{F}V^\varepsilon - \kappa)}{\varepsilon} + \kappa G \right) \right] . \tag{5.6}
$$

5.1.3 Scaled Direct Control Form

By extending the idea in Section 4.1, one can simply discretize the control form of equation (5.4), as in [9],

$$
\min_{\substack{\varphi \in \{0,1\}, \psi \in \{0,1\} \\ \varphi\psi = 0}} \Big[\psi(\kappa - \mathcal{F}V) + (1 - \psi)(V_\tau - \mathcal{L}V - \lambda \mathcal{J} V - \varphi G \mathcal{F} V) \Big] = 0 \; . \tag{5.7}
$$

Equivalently, with the scaling factor $\Pi > 0$ applied to equation (5.7), we obtain the following scaled direct control form of equation

$$\min_{\substack{\varphi \in \{0,1\}, \psi \in \{0,1\} \\ \varphi\psi = 0}} \Big[\Pi\psi(\kappa - \mathcal{F}V) + (1-\psi)(V_\tau - \mathcal{L}V - \lambda\mathcal{J}V - \varphi G\mathcal{F}V) \Big] = 0 \; . \qquad (5.8)$$

5.2 Boundary Conditions

5.2.1 Localization

The original problem (5.4), or equivalently, (5.6) or (5.8) is posed on the domain $\Omega^\infty = [0, \infty] \times [0, \omega_0] \times [0, T]$ as in (2.8). For computational purposes, we localize these equations on the finite computational domain of $\Omega^L = [0, W_{\max}] \times [0, \omega_0] \times [0, T]$ as in (2.9).

5.2.2 Terminal and Boundary Conditions

Define the following sets of points $(W, A, \tau) \in \Omega^L$

$$\begin{aligned}
\Omega_{\tau^0} &= [0, W_{\max}] \times [0, \omega_0] \times \{0\} \; , \\
\Omega_{W_0} &= \{0\} \times (0, \omega_0] \times (0, T] \\
\Omega_{W_{\max}} &= \{W_{\max}\} \times [0, \omega_0] \times (0, T] \\
\Omega_{A_0} &= [0, W_{\max}) \times \{0\} \times (0, T] \\
\partial\Omega_{in} &= \Omega_{\tau^0} \cup \Omega_{W_0} \cup \Omega_{W_{\max}} \cup \Omega_{A_0} \\
\Omega_{\hat{W}_{\max}} &= [\hat{W}_{\max}, W_{\max}) \times [0, \omega_0] \times (0, T] \\
\Omega_{in_a} &= \Omega^L \backslash \partial\Omega_{in} \backslash \Omega_{\hat{W}_{\max}} \\
\Omega_{in_b} &= \Omega^L \backslash \partial\Omega_{in} \backslash \Omega_{in_a} \; . \qquad (5.9)
\end{aligned}$$

For $(W, A, \tau) \in \Omega_{in_a}$, we solve

$$\min\left[V_\tau - \mathcal{L}V - \lambda\mathcal{J}V - G\max(\mathcal{F}V, 0),\ \kappa - \mathcal{F}V\right] = 0 \qquad (W, A, \tau) \in \Omega_{in_a} . \quad (5.10)$$

The integral term in the equations are computed by transforming it to a correlation integral which then can be computed efficiently by FFT. In [20], the author describes how to determine the value of $W_{\max}$ based on $\hat{W}_{\max}$ so that FFT wrap-around effects are less than a user specified tolerance.

At expiry time $\tau = 0$ and when $W \to 0$, the terminal and boundary conditions are the same as those for the standard Geometric Brownian Motion case.

$$V(W, A, \tau = 0) = \max\left[W, (1-\kappa)A\right] \qquad (W, A, \tau) \in \Omega_{\tau^0} ; \quad (5.11)$$

$$\min\left[V_\tau - rV - G\max(1 - V_A, 0), \kappa - (1 - V_A)\right] = 0 \qquad (W, A, \tau) \in \Omega_{W_0} . \quad (5.12)$$

As $W \to \infty$, we make the common assumption that $V_{WW} \simeq 0$ as in [23]. We assume that for $W > \hat{W}_{\max}$, V is linear and $V_{WW} = 0$. Consequently, for $(W, A, \tau) \in \Omega_{in_b}$ the PIDE (5.4) reduces to the Black-Scholes PDE [8, 38, 20]. Effectively, assuming $V_{WW} = 0$, that is equivalent to setting $\lambda = 0$ in equation (5.4) and we solve

$$\min\left[V_\tau - \mathcal{L}_{\mathcal{G}}V - G\max(\mathcal{F}V, 0),\ \kappa - \mathcal{F}V\right] = 0 \qquad (W, A, \tau) \in \Omega_{in_b} . \quad (5.13)$$

According to [18], the withdrawal guarantee becomes insignificant for W sufficiently large, so we use the same boundary condition as for the no jump case discussed in Section 2.2. The exact boundary condition at $W_{\max}$ is

$$V(W_{\max}, A, \tau) = e^{-\eta\tau} W_{\max}\left(1 + O\left(\frac{1}{W_{\max}}\right)\right) \; ; \quad W_{\max} \to \infty \tag{5.14}$$

Therefore as in [18, 23], we impose the following condition at $W_{\max}$

$$\begin{aligned} V(W_{\max}, A, \tau) &= e^{-\eta\tau} W_{\max} \; , \\ &\qquad (W, A, \tau) \in \Omega_{W_{\max}} \; . \end{aligned} \tag{5.15}$$

Note that the integral term in equation (5.10) requires information for $W > W_{\max}$. Based on equations (5.14) and (5.15), we assume that

$$V(W, A, \tau) = e^{-\eta\tau} W \; ; \quad W > W_{\max} \; , \tag{5.16}$$

so that the integral term can be written as as

$$\begin{aligned} \mathcal{J}V(W, A, \tau) &= \int_0^\infty V(JW, A, \tau) p(J) \; dJ \\ &= \int_0^{W_{\max}/W} V(JW, A, \tau) p(J) \; dJ + \int_{W_{\max}/W}^\infty e^{-\eta\tau} W p(J) \; dJ \\ &= \int_0^{W_{\max}/W} V(JW, A, \tau) p(J) \; dJ + e^{-\eta\tau} W \int_{W_{\max}/W}^\infty p(J) \; dJ \end{aligned} \tag{5.17}$$

Note that $\mathcal{J}V$ is non-zero only for $W \leq \hat{W}_{\max}$. As described in [23], we can select $\hat{W}_{\max}$, $W_{\max}$ so that

$$\int_{W_{\max}/\hat{W}_{\max}}^\infty p(J) dJ < \epsilon_1 \; , \tag{5.18}$$

where ϵ_1 is any desired tolerance. In the following, we assume that we have selected ϵ_1

sufficiently small so that the integral term in (5.18) can be ignored, so that

$$\mathcal{J}V(W,A,\tau) \simeq \begin{cases} \int_0^{W_{\max}/W} V(JW,A,\tau)p(J)dJ & \text{if } W \leq \hat{W}_{\max} \ , \\ 0 & \text{if } W > \hat{W}_{\max} \ . \end{cases} \tag{5.19}$$

As $A \to 0$, no withdrawal is possible, so the variational inequality becomes the following PIDE

$$\begin{aligned} V_\tau &= \mathcal{L}V + \lambda \mathcal{J}V \\ & \quad (W,A,\tau) \in \Omega_{A_0} \ . \end{aligned} \tag{5.20}$$

Note that as discussed in [18], no boundary condition is required at $A = \omega_0$ due to hyperbolic nature of the variable A.

5.3 Discretized PIDE

5.3.1 Discretized Integral Term $\mathcal{J}^h V$

Aside from discretizing the PDE part, we need to further discretize the integral term $\mathcal{J}V$ in equations (5.6) and (5.8). We use $\mathcal{J}^h V$ to denote the discretized integral term. The discretization technique is to transform the integral term into a correlation integral combined with a use of the midpoint rule as described in detail in [23, 22, 54]. No information is needed outside the domain (i.e. $W > W_{\max}$) since we assume equation (5.19) holds.

5.3.2 Discretization: Penalty Method

As discussed in Section 3.4, with $\varepsilon = C\Delta\tau$ and The final discretized form of (5.6) becomes

$$\begin{aligned}
&\frac{1}{\Delta\tau}V_{i,j}^{n+1} - \mathcal{L}^h V_{i,j}^{n+1} + \varphi_{i,j}^* G(\mathcal{D}_A^h V_{i,j}^{n+1} + \mathcal{D}_W^h V_{i,j}^{n+1}) + \frac{\psi_{i,j}^*}{\varepsilon}(\mathcal{D}_A^h V_{i,j}^{n+1} + \mathcal{D}_W^h V_{i,j}^{n+1}) \\
&= \lambda[\mathcal{J}^h V^{n+1}]_{i,j} + \varphi_{i,j}^* G + \psi_{i,j}^* (\frac{1-\kappa}{\varepsilon} + \kappa G) + \frac{1}{\Delta\tau} V_{i,j}^n \ , \qquad (5.21)
\end{aligned}$$

where

$$\begin{aligned}
(\varphi_{i,j}^*, \psi_{i,j}^*) \quad = \quad & \underset{\substack{\varphi\in\{0,1\},\psi\in\{0,1\} \\ \varphi\psi=0}}{\arg\max} \Big[\varphi \ G(1 - D_A^h V_{i,j}^{n+1} - \mathcal{D}_W^h V_{i,j}^{n+1}) \\
& + \psi\Big(\frac{1 - \mathcal{D}_A^h V_{i,j}^{n+1} - D_W^h V_{i,j}^{n+1} - \kappa}{\varepsilon} + \kappa G\Big)\Big]_i \ . \qquad (5.22)
\end{aligned}$$

We can also rewrite equation (5.21) in an equivalent form (using a backward difference for $\mathcal{D}_A^h$)

$$\begin{aligned}
\frac{1}{\Delta\tau}V_{i,j}^{n+1} - \mathcal{L}^h V_{i,j}^{n+1} \quad + \quad & \varphi_{i,j}^* G(\frac{V_{i,j}^{n+1}}{\Delta A_j^-} + \mathcal{D}_W^h V_{i,j}^{n+1}) + \frac{\psi_{i,j}^*}{\varepsilon}(\frac{V_{i,j}^{n+1}}{\Delta A_j^-} + \mathcal{D}_W^h V_{i,j}^{n+1}) \\
= \quad & \lambda[\mathcal{J}^h V^{n+1}]_{i,j} + \varphi_{i,j}^* G + \psi_{i,j}^* (\frac{1-\kappa}{\varepsilon} + \kappa G) + \frac{1}{\Delta\tau} V_{i,j}^n \\
& +(\varphi_{i,j}^* G + \frac{\psi_{i,j}^*}{\varepsilon})\frac{1}{\Delta A_j^-} V_{i,j-1}^{n+1} \ . \qquad (5.23)
\end{aligned}$$

5.3.3 Discretization: Direct Control Method

Equation (5.8) can be written in the following discrete form

$$
\begin{aligned}
&(1-\psi^*_{i,j})\left(\frac{1}{\Delta\tau}V^{n+1}_{i,j} - \mathcal{L}^h V^{n+1}_{i,j} + \varphi^*_{i,j} G(\mathcal{D}^h_W V^{n+1}_{i,j} + \mathcal{D}^h_A V^{n+1}_{i,j})\right) \\
&+\Pi\ \psi^*_{i,j}(\mathcal{D}^h_W V^{n+1}_{i,j} + \mathcal{D}^h_A V^{n+1}_{i,j}) \\
&= (1-\psi^*_{i,j})\frac{1}{\Delta\tau}V^n_{i,j} + \Pi\ \psi^*_{i,j}(1-\kappa) + (1-\psi^*_{i,j})\left(\lambda[\mathcal{J}^h V^{n+1}]_{i,j} + \varphi^*_{i,j} G\right)\ ,
\end{aligned}
\tag{5.24}
$$

where

$$
\begin{aligned}
(\varphi^*_{i,j}, \psi^*_{i,j}) \quad = \quad & \underset{\substack{\varphi\in\{0,1\},\psi\in\{0,1\} \\ \varphi\psi=0}}{\arg\max} \Bigg[\Pi\ \psi(\kappa - \mathcal{F}^h V^{n+1}_{i,j}) - (1-\psi)\Bigg(\frac{V^{n+1}_{i,j} - V^n_{i,j}}{\Delta\tau} \\
& -\big(\mathcal{L}^h V^{n+1}_{i,j} + \lambda[\mathcal{J}^h V^{n+1}]_{i,j} + \varphi G \mathcal{F}^h V^{n+1}_{i,j}\big)\Bigg)\Bigg]_i\ .
\end{aligned}
\tag{5.25}
$$

We can also rewrite equation (5.24) in an equivalent form (using a backward difference for $\mathcal{D}^h_A$)

$$
\begin{aligned}
(1-\psi^*_{i,j})\left(\frac{1}{\Delta\tau}V^{n+1}_{i,j}\right. \quad - \quad & \left.\mathcal{L}^h V^{n+1}_{i,j} + \varphi^*_{i,j} G(\frac{V^{n+1}_{i,j}}{\Delta A^-_j} + \mathcal{D}^h_W V^{n+1}_{i,j})\right) + \Pi\ \psi^*_{i,j}(\frac{V^{n+1}_{i,j}}{\Delta A^-_j} + \mathcal{D}^h_W V^{n+1}_{i,j}) \\
= \quad & (1-\psi^*_{i,j})\frac{1}{\Delta\tau}V^n_{i,j} + \Pi\ \psi^*_{i,j}(1-\kappa + \frac{V^{n+1}_{i,j-1}}{\Delta A^-_j}) \\
& +(1-\psi^*_{i,j})\left(\lambda[\mathcal{J}^h V^{n+1}]_{i,j} + \varphi^*_{i,j} G(\frac{V^{n+1}_{i,j-1}}{\Delta A^-_j} + 1)\right)\ .
\end{aligned}
\tag{5.26}
$$

5.4 Convergence: Jump Diffusion Case

Using the methods in Chapter 3, we can show that discretizations (5.23) and (5.26) are monotone, consistent and stable. The details are algebraically tedious and not particularly illuminating given the results in Section 3.4, hence we give only a brief sketch of how this can be done. We assume that $\mathcal{J}^h V$ is discretized as in [23].

Monotonicity

The discretization in [23] uses the midpoint rule, hence the coefficients of $V_{i,j}^{n+1}$ in $\left[\mathcal{J}^h V\right]_{i,j}^{n+1}$ are non-negative and sum to less than or equal to unity [23]. Hence the additional discrete term $\left[\mathcal{J}^h V\right]_{i,j}^{n+1}$ results in a positive coefficient discretization.

Consistency

Since the midpoint rule is $O(h^2)$ for smooth functions, this easily follows.

Stability

We can use the same method as used in Section 3.4.5. The easiest way to do this is to alter the definition of the mean relative jump size ρ slightly. Recall that

$$\rho = E[J-1] = E[J] - 1 \ . \tag{5.27}$$

Instead of using the exact expression for $E[J]$, we evaluate this using the same quadrature rule used to evaluate the $\left[\mathcal{J}^h V\right]$, i.e. $E^h[J]$. This is obviously a consistent approximation. Then when we substitute the bounding function (3.44) into the discretization (5.23) or (5.26), all the discrete jump terms cancel and we can proceed exactly as before. The lower bound proceeds in the same way by substituting the bounding function (3.49) into the discretization.

5.5 Summary

This chapter describes how to extend the GMWB pricing equation to the case where the underlying asset follows a jump diffusion process. The main results are

- The GMWB pricing equation is an HJB VI/PIDE for the jump diffusion case.
- Both the penalty method and the direct control method can be used to solve the resulting PIDE.
- We use a slight modification to the Geometric Brownian Motion case for boundary conditions. The region Ω_{in} is further divided into Ω_{in_a} and Ω_{in_b} with the assumption of linear behavior for nodes in Ω_{in_b}.
- Using the technique of transforming the integral term into a correlation integral combined with a use of the midpoint rule [23, 54] and the standard three point finite difference for derivative terms, we discretize both the penalized form and the scaled direct control form of the resulting PIDE.
- We briefly sketch how to extend the proof of convergence to the viscosity solution for the standard Geometric Brownian Motion case to the jump diffusion case.

Chapter 6

Fixed Point Policy Iteration

In this chapter we discuss a *fixed point policy iteration* scheme for solution of the discretized equations for the singular control formulation of the GMWB pricing problem. Section 6.1 discusses existing iterative methods and presents the fixed point policy iteration scheme. Section 6.2 derives sufficient conditions for the fixed point policy iteration to converge. Section 6.3 discusses a full matrix fixed point policy iteration for solving the discretized HJB PIDEs. Both the penalty method and the direct control method are discussed. Section 6.4 presents a block matrix fixed point policy iteration for solving the same problem with a significant improvement of efficiency. We finally summarize main results of this chapter in Section 6.5.

6.1 Methods for Solving Algebraic Equations

In [26] a number of problems in financial modeling were presented in a general form as nonlinear HJB problems. These problems were then solved by implicitly discretizing the resulting PDE and then solving the resulting discrete algebraic equations. For the applications addressed in [26] an efficient method for solving the associated algebraic systems made use of a (Newton-like) policy iteration scheme. The equations (5.21), (5.23), (5.24) and (5.26) resulting from the GMWB pricing problem can also be written in such a general form. However, when the risky assets follow a Poisson jump diffusion process, the policy iteration has significant efficiency drawbacks. In this section we describe a new procedure, called *fixed point policy iteration* which provides a method for overcoming these computational bottlenecks. This method is a generalization of the method in [23].

6.1.1 Matrix and Vector Notations

It is convenient to use a matrix form to represent the discretized equations. Let $N = i_{\max} \times j_{\max}$ be the size of the $W \times A$ plane grid. We use the following notation for $i_{\max}$ length and N length vectors

$$\begin{aligned} v_{*,j} &= (v_{1,j}, v_{2,j}, \ldots, v_{i_{\max},j})' , \\ \mathbf{v} &= ((v_{*,1})', (v_{*,2})', \ldots, (v_{*,j_{\max}})')'; \end{aligned} \tag{6.1}$$

where $v_{*,j}$ is of length $i_{\max}$ and $\mathbf{v}$ is of length N. Similarly, we can write the controls as vectors

$$\begin{aligned} \chi_{*,j} &= (\chi_{1,j}, \chi_{2,j}, \ldots, \chi_{i_{\max},j})' , \\ \chi &= ((\chi_{*,1})', (\chi_{*,2})', \ldots, (\chi_{*,j_{\max}})')'; \\ & \quad \chi_{i,j} \in X = \big\{(\varphi, \psi) \ \mid \ \varphi \in \{0,1\}, \psi \in \{0,1\}, \varphi\psi = 0\big\} . \end{aligned} \tag{6.2}$$

Let

$$\ell = i + (j-1) i_{\max} \text{ with } 1 \le i \le i_{\max} \text{ and } 1 \le j \le j_{\max}, \tag{6.3}$$

then

$$
\begin{aligned}
[v_{*,j}]_i &= v_{i,j} \\
&= v_\ell \\
&\quad \ell = i + (j-1) i_{\max} \ .
\end{aligned}
\tag{6.4}
$$

As a result, we will sometimes refer to the same entry in the N-length vector $\mathbf{v}$ as v_ℓ or $v_{i,j}$, which will be clear from the context. It is convenient to represent the algebraic equations by matrix notation. In this thesis we use boldface capital letter $\mathbf{T}$ to represent an $N \times N$ matrix with entries $[\mathbf{T}]_{\ell,u} = T_{\ell,u}$. We will also refer to the j^{th} $i_{\max} \times i_{\max}$ subblock of $\mathbf{T}$ using the notation $\mathbf{T_j}$. These subblocks will be defined in Section 6.4.

6.1.2 Discretized Equations in Matrix Form

At each timestep, we solve for the unknowns $V_{i,j}^{n+1}$ in equations (5.21) and (5.24). Letting $[v_{*,j}]_i = V_{i,j}^{n+1}$, the algebraic equations at each timestep can be represented in the following general form as suggested in [26].

$$
\sup_{\chi \in X} \left\{ -\mathbf{T}(\chi)\mathbf{v} + \mathbf{c}(\chi) \right\} = 0 \ , \tag{6.5}
$$

with $\mathbf{T}$ a square matrix, $\mathbf{c}, \chi$ vectors and where X is a set of controls. Equation (6.5) is interpreted as

$$
\begin{aligned}
\mathbf{T}(\chi^*)\mathbf{v} &= \mathbf{c}(\chi^*) \\
\text{with} \quad \chi_\ell^* &= \arg\max_{\chi_\ell \in X} \Big[-\mathbf{T}(\chi)\mathbf{v} + \mathbf{c}(\chi) \Big]_\ell \ .
\end{aligned}
\tag{6.6}
$$

Thus the problem has a potentially different control for each row of the linear system. Note that equation (6.5) is highly nonlinear.

Remark 6.1.1. *It is important to note that due to the nature of the discretized HJB equations [26],* $[\mathbf{T}(\chi)]_{\ell,u}$ *and* $[\mathbf{c}(\chi)]_{\ell}$ *depend only on* χ_ℓ*, and we restrict our attention to matrices and vectors having this property throughout the remainder of this thesis.*

The set of admissible controls X for our problems can be finite or infinite [55, 56], and the local objective function

$$\Big[-\mathbf{T}(\chi)\mathbf{v} + \mathbf{c}(\chi)\Big]_{\ell} \tag{6.7}$$

can be a discontinuous function of the control χ [55]. We make the following assumptions:

Assumption 6.1.1. *Either*

(a) The set of controls X is finite; or

(b) the set of controls X is compact, and the local objective function (6.7) is an upper semi-continuous function of the controls.

Assumption 6.1.1 ensures that there always exists a χ^* such that

$$\begin{aligned} -\mathbf{T}(\chi^*)\mathbf{v} + \mathbf{c}(\chi^*) &= \sup_{\chi_\ell \in X}\Big\{-\mathbf{T}(\chi)\mathbf{v} + \mathbf{c}(\chi)\Big\} \\ &= \max_{\chi_\ell \in X}\Big\{-\mathbf{T}(\chi)\mathbf{v} + \mathbf{c}(\chi)\Big\}\ . \end{aligned} \tag{6.8}$$

Note that this statement holds for each row. We remark that the assumption that the local objective function is upper semi-continuous is not strictly necessary. However removing this assumption results in tedious notational complication [55].

6.1.3 Policy Iteration

Policy iteration is an iterative procedure which constructs a new solution $\mathbf{v}$ from an initial approximation $\mathbf{v}^0$ by first finding a policy (i.e. an admissible control) which maximizes our objective function and then solving a linear system to determine the next candidate. More precisely, let $\mathbf{v}^k$ denote the k^{th} estimate for $\mathbf{v}$ (starting at $\mathbf{v}^0$). Then the policy iteration approach for solution of equation (6.6) is given in Algorithm 6.1.1.

Algorithm 6.1.1 Policy Iteration

$\mathbf{v}^k = (\mathbf{v})^k$ with $\mathbf{v}^0 =$ Initial solution vector of size N
for $k = 0, 1, 2, \ldots$ until converge **do**
  Determine

$$\chi^k = (\chi_1^k, \chi_2^k, \ldots, \chi_N^k)$$
$$\chi_\ell^k \Leftarrow \arg\max_{\chi_\ell \in X} \left[-\mathbf{T}(\chi)\mathbf{v}^k + \mathbf{c}(\chi) \right]_\ell$$

  Solve the linear system

$$\mathbf{T}(\chi^k)\mathbf{v}^{k+1} = \mathbf{c}(\chi^k)$$

  if $k > 0$ and $\max_\ell \dfrac{|v_\ell^{k+1} - v_\ell^k|}{\max\left[scale, v_\ell^{k+1}\right]} < tolerance$ **then**
    break from the iteration
  end if
end for

The term *scale* in Algorithm 6.1.1 is used to ensure that unrealistic levels of accuracy are not required when the value is very small. Typically, $scale = 1$ for options priced in dollars.

There are several possibilities for solving the linear system in the policy iteration method. For example, if $\mathbf{T}$ is sparse, then direct or iterative methods (such as preconditioned GMRES [46]) can be used.

6.1.4 Splitting Methods

Unfortunately it is not always the case that one can easily solve the policy iteration matrix $\mathbf{T}(\chi^k)$ at each iteration. Indeed it is possible to spend a great deal of effort in solving $\mathbf{T}(\chi^k)$ at each iteration. In this thesis we isolate the part of the iteration matrix which prevents efficient linear solution. At the same time there will need to be enough conditions to ensure that any new iteration scheme still converges to the correct solution. Thus our algebraic equations will now be written as

$$\begin{aligned} \big[\mathbf{A}(\chi^*) - \mathbf{B}(\chi^*)\big]\ \mathbf{v} &= \mathbf{c}(\chi^*) \\ \text{with} \quad \chi^*_\ell &= \arg\max_{\chi_\ell \in X} \Big[-\big[\mathbf{A}(\chi) - \mathbf{B}(\chi)\big]\mathbf{v} + \mathbf{c}(\chi) \Big]_\ell \ . \end{aligned} \tag{6.9}$$

We assume that this splitting is such that any linear system having $\mathbf{A}(\chi)$ as its coefficient matrix is easy to solve.

Remark 6.1.2. *We remind the reader that* $[\mathbf{A}(\chi)-\mathbf{B}(\chi)]_{\ell,u}$ *and* $[\mathbf{c}(\chi)]_\ell$ *depend only on* χ_ℓ*, as in Remark 6.1.1. In other words, given* $\mathbf{v}$*, then the optimal control* χ^*_ℓ *can be determined by examining only the* ℓ^{th} *row of* $-\big[\mathbf{A}(\chi) - \mathbf{B}(\chi)\big]\mathbf{v} + \mathbf{c}(\chi)$*.*

6.1.5 Simple Iteration

Using the above notation, then at each step of full policy iteration, we solve

$$\big[\mathbf{A}(\chi^k) - \mathbf{B}(\chi^k)\big]\mathbf{v}^{k+1} = \mathbf{c}(\chi^k) \ . \tag{6.10}$$

However, as discussed above, it may be very costly to solve equation (6.10) using a direct method. An obvious alternative is to use an iterative method. If $(\mathbf{v}^{k+1})^m$ is the m^{th}

estimate for $\mathbf{v}^{k+1}$, then simple iteration for solution of linear system (6.10) is

$$\mathbf{A}(\chi^k)\left(\mathbf{v}^{k+1}\right)^{m+1} \;=\; \mathbf{B}(\chi^k)\left(\mathbf{v}^{k+1}\right)^{m} + \mathbf{c}(\chi^k)\ . \tag{6.11}$$

6.1.6 Fixed Point-Policy Iteration

Instead of solving the linear system to convergence using simple iteration, it is natural to ask whether it suffices to use only a single simple iteration at each nonlinear iteration. In this case we replace Policy Iteration with what we refer to as Fixed Point Policy Iteration.

Algorithm 6.1.2 Fixed Point Policy Iteration

1: $\mathbf{v}^k = (\mathbf{v})^k$ with $\mathbf{v}^0 =$ Initial solution vector of size N
2: **for** $k = 0, 1, 2, \ldots$ until converge **do**
3: Determine
$\chi^k = (\chi_1^k, \chi_2^k, \ldots, \chi_N^k)$
$\chi_\ell^k \Leftarrow \arg\max_{\chi_\ell \in X} \left[-\big[\mathbf{A}(\chi) - \mathbf{B}(\chi)\big]\mathbf{v}^k + \mathbf{c}(\chi) \right]_\ell$
4: Solve the linear system
$\mathbf{A}(\chi^k)\mathbf{v}^{k+1} = \mathbf{B}(\chi^k)\mathbf{v}^k + \mathbf{c}(\chi^k)$
5: **if** $k > 0$ and $\max_\ell \dfrac{|v_\ell^{k+1} - v_\ell^k|}{\max\left[scale, v_\ell^{k+1}\right]} < tolerance$ **then**
6: break from the iteration
7: **end if**
8: **end for**

The above method requires only the solution of the sparse matrix $\mathbf{A}(\chi^k)$ and a matrix-vector multiply $\mathbf{B}(\chi^k)\mathbf{v}^k$ at each nonlinear iteration.

6.2 Convergence of the Fixed Point-Policy Iteration

In [21], the convergence of an iterative scheme for a penalty formulation for American options under a jump diffusion process was proven. This same idea was generalized for other HJB problems in [14]. While it is possible to use this approach to prove convergence of iteration scheme in Algorithm 6.1.2, these proofs are algebraically complex. In the following, we will present a simpler and more general method which proves convergence of the iteration scheme in Algorithm 6.1.2.

In order to ensure convergence of our scheme we need to make some basic assumptions which hold for the applications that are of interest.

Condition 6.2.1. *The matrices $\mathbf{A}(\chi), \mathbf{B}(\chi)$ and vector $\mathbf{c}(\chi)$ satisfy:*

(i) The matrices $\mathbf{A}(\chi)$ and $\mathbf{A}(\chi) - \mathbf{B}(\chi)$ are M matrices.

(ii) The matrices $\mathbf{A}(\chi), \mathbf{B}(\chi)$, the vector $\mathbf{c}(\chi)$, and $\|\mathbf{A}^{-1}(\chi)\|_\infty$ are bounded, independent of χ.

(iii) There is a constant $C_1 < 1$ such that

$$\|\mathbf{A}^{-1}(\chi^k)\mathbf{B}(\chi^{k-1})\|_\infty \leq C_1 \text{ and } \|\mathbf{A}^{-1}(\chi^k)\mathbf{B}(\chi^k)\|_\infty \leq C_1. \tag{6.12}$$

Remark 6.2.1. *We remind the reader that a sufficient condition for a matrix to be an M matrix is that the offdiagonals are nonpositive, and each row sum is strictly positive [53]. We will use this condition in the following. See Remark 3.3.2 for the definition of an M matrix.*

Remark 6.2.2. *In order to ensure convergence, the discretizations of our financial problems as in (6.6) need to be monotone, consistent and ℓ_∞ stable. This requires a positive coefficient discretization resulting in the M matrices of (i) and bounded matrices $\mathbf{A}(\chi)$, $\mathbf{B}(\chi)$ and vector $\mathbf{c}(\chi)$ of (ii). Property (iii) states that the $\mathbf{B}$ component is small (perhaps using scaling) in comparison to $\mathbf{A}$ in order to ensure convergence of the discrete scheme.*

Before proving the main result of this section, it will be helpful to note the following Proposition and Lemmas.

Proposition 6.2.1 (Convergent Sequence)**.** *Given a bounded infinite sequence* (v_n)*, such that*

$$v_{k+1} \geq v_k - \beta^k \alpha \ , \tag{6.13}$$

where α *is a constant independent of* k*, and* $|\beta| < 1$*, then the sequence converges.*

Proof. An immediate consequence of property (6.13) is that, $\forall \epsilon > 0, \exists N_1$ such that $\forall k > k' > N_1$

$$v_k \geq v_{k'} - \frac{\epsilon}{2} \ . \tag{6.14}$$

Let $s = \limsup v_n$. Recall the properties of the lim sup,

$$\forall \epsilon > 0, \exists N_2, \text{ s.t. } \forall k > N_2, v_k < s + \epsilon \tag{6.15}$$

$$\forall \epsilon > 0, \forall N, \exists j > N, \text{ s.t. } v_j > s - \frac{\epsilon}{2} \ . \tag{6.16}$$

Choose N_2 such that equation (6.15) holds. Choose $N_3 = \max(N_1, N_2)$, so that $\forall k > k' > N_3$

$$v_k \geq v_{k'} - \frac{\epsilon}{2} \qquad \text{From (6.14)} \tag{6.17}$$

$$v_k < s + \epsilon \qquad \text{From (6.15)} \tag{6.18}$$

Choose $k' > N_3$, such that (from (6.16))

$$v_{k'} > s - \frac{\epsilon}{2} \ . \tag{6.19}$$

From equations (6.17) and (6.19), we have

$$\forall \epsilon > 0, \forall k > k', v_k > s - \epsilon \ . \tag{6.20}$$

From equations (6.18) and (6.20) we have

$$\forall \epsilon > 0, \exists k' \text{ s.t. } \forall k > k', |v_k - s| < \epsilon \ . \tag{6.21}$$

□

Lemma 6.2.1 (Bounded Iterates)**.** *Let matrices* $\mathbf{A}(\chi), \mathbf{B}(\chi)$ *and the vector* $\mathbf{c}(\chi)$ *satisfy Condition 6.2.1. Then* $\|\mathbf{v}^k\|_\infty$ *is bounded independent of* k*.*

Proof. From Algorithm 6.1.2 we have

$$\begin{aligned} \|\mathbf{v}^{k+1}\|_\infty &\leq \|\mathbf{A}^{-1}(\chi^k)\mathbf{B}(\chi^k)\|_\infty \|\mathbf{v}^k\|_\infty + \|\mathbf{A}^{-1}(\chi^k)\mathbf{c}(\chi^k)\|_\infty \\ &\leq C_1 \|\mathbf{v}^k\|_\infty + C_2 \end{aligned} \tag{6.22}$$

for some constant C_2 independent of k. Iterating equation (6.22) gives

$$\begin{aligned} \|\mathbf{v}^{k+1}\|_\infty &\leq C_1^{k+1} \|\mathbf{v}^0\|_\infty + C_2 \sum_{i=0}^{k} C_1^i \\ &\leq \|\mathbf{v}^0\|_\infty + \frac{C_2}{1 - C_1} \ , \end{aligned} \tag{6.23}$$

which follows since $C_1 < 1$. $\square$

Lemma 6.2.2 (Uniqueness of Solution)**.** *Assume that the set of controls satisfy Assumption 6.1.1 and that* $\mathbf{A}(\chi)$*,* $\mathbf{B}(\chi)$*, and* $\mathbf{c}(\chi)$ *satisfy Condition 6.2.1. If the iterative scheme in Algorithm 6.1.2 converges, then it converges to the unique solution of equation (6.9).*

Proof. Manipulation of the method in Algorithm 6.1.2 results in

$$\begin{aligned} \mathbf{A}(\chi^k)(\mathbf{v}^{k+1} - \mathbf{v}^k) &= -\mathbf{A}(\chi^k)\mathbf{v}^k + \mathbf{B}(\chi^k)\mathbf{v}^k + \mathbf{c}(\chi^k) \\ &= \max_{\chi_\ell \in X} \left\{ -\mathbf{A}(\chi)\mathbf{v}^k + \mathbf{B}(\chi)\mathbf{v}^k + \mathbf{c}(\chi) \right\} . \end{aligned} \tag{6.24}$$

Suppose $\lim_{k\to\infty} \mathbf{v}^k = \mathbf{v}^*$. Then

$$\lim_{k\to\infty} \mathbf{A}(\chi^k)(\mathbf{v}^{k+1} - \mathbf{v}^k) = \mathbf{0} \tag{6.25}$$

since $\mathbf{A}(\chi)$ is bounded. Consequently

$$\begin{aligned} \mathbf{0} &= \lim_{k\to\infty} \max_{\chi_\ell \in X} \left\{ -\mathbf{A}(\chi)\mathbf{v}^k + \mathbf{B}(\chi)\mathbf{v}^k + \mathbf{c}(\chi) \right\} \\ &= \max_{\chi_\ell \in X} \left\{ -\mathbf{A}(\chi)\mathbf{v}^* + \mathbf{B}(\chi)\mathbf{v}^* + \mathbf{c}(\chi) \right\} , \end{aligned} \tag{6.26}$$

since $\max(\cdot)$ is a continuous function of $\mathbf{v}^k$ [1] , and hence $\mathbf{v}^*$ is a solution of equation (6.9).

As for uniqueness, suppose there are two solutions $\mathbf{u}$, $\mathbf{v}$, such that

$$(\mathbf{A}(\chi^u) - \mathbf{B}(\chi^u))\mathbf{u} = \mathbf{c}(\chi^u) \ ; \ \chi^u_\ell = \arg\max_{\chi_\ell \in X} \left[-\mathbf{A}(\chi)\mathbf{u} + \mathbf{B}(\chi)\mathbf{u} + \mathbf{c}(\chi) \right]_\ell \tag{6.27}$$

$$(\mathbf{A}(\chi^v) - \mathbf{B}(\chi^v))\mathbf{v} = \mathbf{c}(\chi^v) \ ; \ \chi^v_\ell = \arg\max_{\chi_\ell \in X} \left[-\mathbf{A}(\chi)\mathbf{v} + \mathbf{B}(\chi)\mathbf{v} + \mathbf{c}(\chi) \right]_\ell . \tag{6.28}$$

[1] The proof of the continuity of $\max(\cdot)$ is in Appendix E

Manipulate equation (6.28) to obtain

$$(\mathbf{A}(\chi^u) - \mathbf{B}(\chi^u))\mathbf{v} = (\mathbf{A}(\chi^u) - \mathbf{B}(\chi^u))\mathbf{v} - (\mathbf{A}(\chi^v) - \mathbf{B}(\chi^v))\mathbf{v} + \mathbf{c}(\chi^v) \ . \quad (6.29)$$

Subtract equation (6.29) from equation (6.27) to obtain

$$\begin{aligned}(\mathbf{A}(\chi^u) - \mathbf{B}(\chi^u))(\mathbf{u} - \mathbf{v}) &= -\mathbf{A}(\chi^u)\mathbf{v} + \mathbf{B}(\chi^u)\mathbf{v} + \mathbf{c}(\chi^u) \\ &\quad - (-\mathbf{A}(\chi^v)\mathbf{v} + \mathbf{B}(\chi^v)\mathbf{v} + \mathbf{c}(\chi^v)) \ . \end{aligned} \quad (6.30)$$

But χ^v maximizes $(-\mathbf{A}(\chi^v)\mathbf{v} + \mathbf{B}(\chi^v)\mathbf{v} + \mathbf{c}(\chi^v))$, hence the rhs of equation (6.30) is non-positive. Since $(\mathbf{A}(\chi^u) - \mathbf{B}(\chi^u))$ is an M matrix, then $(\mathbf{u} - \mathbf{v}) \leq 0$. Interchange $\mathbf{u}$ and $\mathbf{v}$ to obtain $(\mathbf{v} - \mathbf{u}) \leq 0$, hence $\mathbf{u} = \mathbf{v}$. □

Theorem 6.2.1 (Convergence of Scheme)**.** *If the matrices* $\mathbf{A}(\chi), \mathbf{B}(\chi)$ *and vector* $\mathbf{c}(\chi)$ *satisfy Condition 6.2.1, then the iteration scheme in Algorithm 6.1.2 converges to the unique solution of equation (6.9), for any initial iterate* $\mathbf{v}^k$*.*

Proof. Algorithm 6.1.2 can be written as

$$\begin{aligned}\mathbf{A}(\chi^k)(\mathbf{v}^{k+1} - \mathbf{v}^k) &= \mathbf{B}(\chi^{k-1})(\mathbf{v}^k - \mathbf{v}^{k-1}) - \mathbf{A}(\chi^k)\mathbf{v}^k + \mathbf{B}(\chi^k)\mathbf{v}^k + \mathbf{c}(\chi^k) \\ &\quad - \left(-\mathbf{A}(\chi^{k-1})\mathbf{v}^k + \mathbf{B}(\chi^{k-1})\mathbf{v}^k + \mathbf{c}(\chi^{k-1})\right) . \end{aligned} \quad (6.31)$$

Since χ^k maximizes $(-\mathbf{A}(\chi)\mathbf{v}^k + \mathbf{B}(\chi)\mathbf{v}^k + \mathbf{c}(\chi))$ we have

$$-\mathbf{A}(\chi^k)\mathbf{v}^k + \mathbf{B}(\chi^k)\mathbf{v}^k + \mathbf{c}(\chi^k) - \left(-\mathbf{A}(\chi^{k-1})\mathbf{v}^k + \mathbf{B}(\chi^{k-1})\mathbf{v}^k + \mathbf{c}(\chi^{k-1})\right) \geq 0 \ . \quad (6.32)$$

Equations (6.31) and (6.32) combined with the fact that $\mathbf{A}(\chi^k)$ is an M matrix then implies

$$\mathbf{v}^{k+1} - \mathbf{v}^k \geq \left(\mathbf{A}^{-1}(\chi^k)\mathbf{B}(\chi^{k-1})\right)(\mathbf{v}^k - \mathbf{v}^{k-1}) \ . \quad (6.33)$$

From Condition 6.2.1

$$\|\mathbf{A}^{-1}(\chi^k)\mathbf{B}(\chi^{k-1})\|_\infty \leq C_1 < 1 \tag{6.34}$$

and so we have

$$(\mathbf{v}^{k+1} - \mathbf{v}^k) \;\; \geq \;\; -C_1^k \|\mathbf{v}^1 - \mathbf{v}^0\|_\infty \, \mathbf{e} \tag{6.35}$$

where $\mathbf{e} = (1, 1, ..., 1)'$. Let $C_3 = \|\mathbf{v}^1 - \mathbf{v}^0\|_\infty$. Then, in component form we have

$$[\mathbf{v}^{k+1}]_\ell \;\; \geq \;\; [\mathbf{v}^k]_\ell - C_1^k C_3 \; . \tag{6.36}$$

From Lemma 6.2.1, the sequence $[\mathbf{v}^{k+1}]_\ell$ is bounded, hence the iteration converges from Proposition 6.2.1. In the limit, the iteration converges to the unique solution of equation (6.9) from Lemma 6.2.2. □

Remark 6.2.3 (Monotone Convergence)**.** *We can eliminate condition (6.34) if we require that* $(\mathbf{v}^1 - \mathbf{v}^0) \geq 0$*, and* $\mathbf{B}(\chi) \geq 0$*, since then the iteration will generate a monotone non-decreasing sequence from equation (6.33). Often, it is straightforward to enforce* $(\mathbf{v}^1 - \mathbf{v}^0) \geq 0$ *by specifying* $\mathbf{v}^0 = 0$ *[21], and in many cases* $\mathbf{B}(\chi) \geq 0$*. However, a natural choice for* $\mathbf{v}^0$ *in time-dependent problems is the solution from the previous timestep, hence the choice of* $\mathbf{v}^0 = 0$ *is a poor initial estimate. In fact, tests in [21] show that enforcing monotone convergence using a special choice for the first iterate converges more slowly than using the natural choice of the solution from the previous step, as one might expect. In addition, numerical experiments indicate that floating point errors are amplified if condition (6.34) is violated, and hence the sequence* $\mathbf{v}^k$ *may not be non-decreasing (in inexact arithmetic) even if* $(\mathbf{v}^1 - \mathbf{v}^0) \geq 0$*.*

Remark 6.2.4 (Previous Work)**.** *Various forms of modified policy iteration have been* *suggested in the context of infinite horizon Markov chain problems [35]. However, convergence results in [45] require that the initial iterate be selected so as to enforce monotone convergence (as in Remark 6.2.3). Moreover, we do not require that* $\mathbf{A}(\chi), \mathbf{B}(\chi), \mathbf{c}(\chi)$ *be continuous functions of the control* χ *[55].*

Condition 6.2.1 involves bounding a matrix norm of the form

$$
\begin{aligned}
\|\mathbf{A}^{-1}\mathbf{B}\|_\infty &= \max_{\mathbf{y}\neq\mathbf{0}} \frac{\|\mathbf{A}^{-1}\mathbf{B}\mathbf{y}\|_\infty}{\|\mathbf{y}\|_\infty} \\
&= \max_{\mathbf{y}\neq\mathbf{0}} \frac{\|\mathbf{x}\|_\infty}{\|\mathbf{y}\|_\infty} \qquad \text{where } \mathbf{Ax} = \mathbf{By}
\end{aligned}
\tag{6.37}
$$

with $\mathbf{A}$ an M matrix. The following lemma will be useful in this regard.

Lemma 6.2.3. *Suppose* $\mathbf{A}$ *is a strictly diagonally dominant* M *matrix and* $\mathbf{B} \geq \mathbf{0}$*. We have*

$$
\|\mathbf{A}^{-1}\mathbf{B}\|_\infty \leq \max_\ell \frac{Row\_Sum_{\ \ell}\ (\mathbf{B})}{Row\_Sum_{\ \ell}\ (\mathbf{A})} \tag{6.38}
$$

Proof. Suppose $\mathbf{Ax} = \mathbf{By}$ with $\mathbf{A}$ a strictly diagonally dominant M matrix and $\mathbf{B} \geq \mathbf{0}$. Then for any $\underline{\ell}$ such that $|x_{\underline{\ell}}| = \|\mathbf{x}\|_\infty$, by using the same type of maximum analysis as in [21], we have

$$
A_{\underline{\ell},\underline{\ell}} x_{\underline{\ell}} = -\sum_{u\neq\ell} A_{\underline{\ell},u} x_u + \sum_u B_{\underline{\ell},u} y_u . \tag{6.39}
$$

Taking absolute values on both sides and using the fact that $A_{\underline{\ell},u}$ is nonpositive whenever $u \neq \underline{\ell}$ we have that

$$
A_{\underline{\ell},\underline{\ell}} |x_{\underline{\ell}}| \leq -(\sum_{u\neq\ell} A_{\underline{\ell},u}) \|\mathbf{x}\|_\infty + (\sum_u B_{\underline{\ell},u}) \|\mathbf{y}\|_\infty . \tag{6.40}
$$

Since $|x_{\underline{\ell}}| = \|\mathbf{x}\|_\infty$, we obtain

$$(\sum_u A_{\underline{\ell},u})\|\mathbf{x}\|_\infty \;\; \leq \;\; (\sum_u B_{\underline{\ell},u})\|\mathbf{y}\|_\infty. \tag{6.41}$$

Consequently

$$\begin{aligned}
\|\mathbf{A}^{-1}\mathbf{B}\|_\infty &= \max_{\mathbf{y}\neq\mathbf{0}} \frac{\|\mathbf{A}^{-1}\mathbf{B}\mathbf{y}\|_\infty}{\|\mathbf{y}\|_\infty} = \max_{\mathbf{y}\neq 0} \frac{\|\mathbf{x}\|_\infty}{\|\mathbf{y}\|_\infty} \\
&\leq \max_{\underline{\ell}} \frac{\sum_u B_{\underline{\ell},u}}{\sum_u A_{\underline{\ell},u}} = \max_{\underline{\ell}} \frac{\text{Row_Sum}_{\;\underline{\ell}}\;(\mathbf{B})}{\text{Row_Sum}_{\;\underline{\ell}}\;(\mathbf{A})} \\
&\leq \max_{\ell} \frac{\text{Row_Sum}_{\;\ell}\;(\mathbf{B})}{\text{Row_Sum}_{\;\ell}\;(\mathbf{A})}\;.
\end{aligned} \tag{6.42}$$

□

6.3 Full Matrix Fixed Point-Policy Iteration

In this section, we write equations (5.21) and (5.24) in the general matrix form as in (6.9). Then we show the fixed point policy iteration scheme in Algorithm 6.1.2 converges to the unique solution of (6.9) by verifying Condition 6.2.1 for both equations.

6.3.1 Full Matrix Iteration: Penalty Method

We can represent the linear relationships given in equation (5.21) in matrix form as follows. Let vector $\mathbf{u}$ be an $N - length$ vector. Define square $N \times N$ matrices $\mathbf{A}, \mathbf{B}$ and vector $\mathbf{c}$

of size N by

$$\begin{aligned}
[\mathbf{A}(\chi^k)\mathbf{u}]_\ell &= [\mathbf{A}^k\mathbf{u}]_\ell &=& \; u_\ell - \Delta\tau\mathcal{L}^h u_\ell + \varphi_\ell^k G[D_A^h u_\ell + D_W^h u_\ell]\Delta\tau \\
& & & + \frac{\psi_\ell^k}{\varepsilon}[D_A^h u_\ell + D_W^h u_\ell]\Delta\tau \\
\big[\mathbf{B}(\chi^k)\mathbf{u}\big]_\ell &= [\mathbf{B}^k\mathbf{u}]_\ell &=& \; \lambda\Delta\tau[\mathcal{J}^h\mathbf{u}]_\ell \\
\big[\mathbf{c}(\chi^k)\big]_\ell &= \quad c_\ell^k &=& \; \varphi_\ell^k G\Delta\tau + \psi_\ell^k\big[\frac{(1-\kappa)}{\varepsilon} + \kappa G\big]\Delta\tau + V_\ell^n
\end{aligned} \tag{6.43}$$

with controls

$$\chi_\ell^k = (\varphi_\ell^k, \psi_\ell^k) = \underset{\substack{\varphi_\ell\in\{0,1\},\psi_\ell\in\{0,1\} \\ \varphi_\ell\psi_\ell=0}}{\arg\max} \Big[-\mathbf{A}(\varphi_\ell,\psi_\ell)\mathbf{u}^k + \mathbf{B}(\varphi_\ell,\psi_\ell)\mathbf{u}^k + \mathbf{c}(\varphi_\ell,\psi_\ell)\Big]_\ell \ . \tag{6.44}$$

The discretized equations (5.21) become

$$\sup_{\chi_\ell\in X}\Big\{-\mathbf{A}(\chi)\mathbf{v}^{n+1} + \mathbf{B}(\chi)\mathbf{v}^{n+1} + \mathbf{c}(\chi)\Big\} = 0 \ . \tag{6.45}$$

Remark 6.3.1. *We have written the matrix $\mathbf{B} = \mathbf{B}^k$ although there is no explicit dependence on $(\varphi_\ell^k, \psi_\ell^k)$ in this case in order to use the general form of the previous section.*

Remark 6.3.2. *Note that the separation of $\mathbf{A}(\chi)$ and $\mathbf{B}(\chi)$ in this case is carried out by placing the discretization of the jump diffusion term entirely in $\mathbf{B}(\chi)$. The case when $\mathbf{B}(\chi) = \mathbf{0}$ (and $\lambda = 0$) corresponds to the underlying asset following Geometric Brownian Motion. Consequently, $\mathbf{A}(\chi)$ is easily seen to be a sparse, strictly diagonally dominant M-matrix. In this case policy iteration is quite efficient. When there is a jump diffusion term, difficulties arise since the discretization at any node is linked to all the other nodes resulting in a dense system.*

Remark 6.3.3. *The discretization of the jump term $\mathcal{J}V$ (5.5) as in [23] results in a dense matrix $\mathbf{B}$. However the method of discretization used in that paper implies that* *vector product* $\mathbf{B}\mathbf{v}^n$ *can be computed efficiently in* $O(N \log N)$ *operations using an FFT.*

Recall that in order to ensure convergence to the viscosity solution of equation (5.4), the discretization must be monotone, consistent and l_∞ stable [5]. A positive coefficient discretization guarantees monotonicity [26]. The positive coefficient condition and the discretization of the jump term as in [23] give the following result.

Proposition 6.3.1. *Suppose a positive coefficient discretization [26] is used and the jump operator* $\mathcal{J}^h$ *is discretized using the method in [23] with linear behavior assumed for* $i \geq \hat{i}$ *[23, 54]. Then*

(a) $\mathbf{B}(\chi^k) \geq 0$,

(b) Suppose row ℓ *corresponds to grid node* (i, j) *as in (6.3). Then the* ℓ^{th} *row sums for* $\mathbf{A}(\chi^k)$ *and* $\mathbf{B}(\chi^k)$ *are*

$$
\begin{aligned}
\textit{Row\_Sum}_{\ \ell}\ (\ \mathbf{A}(\chi^k)\ ) \quad &= \quad \begin{cases} 1 + (r + \lambda)\Delta\tau & 2 \leq i < \hat{i} \\ 1 + r\Delta\tau & i = \hat{i}, \ldots, i_{\max} - 1 \ \textit{or}\ i = 1 \\ 1 & i = i_{\max} \end{cases} \\
\textit{Row\_Sum}_{\ \ell}\ (\ \mathbf{B}(\chi^k)\ ) \quad &\leq \quad \begin{cases} \lambda\Delta\tau & 2 \leq i < \hat{i} \\ 0 & i = \hat{i}, \ldots, i_{\max} - 1 \ \textit{or}\ i = 1 \end{cases}
\end{aligned}
\tag{6.46}
$$

(c) The matrices $\mathbf{A}(\chi) - \mathbf{B}(\chi)$ *and* $\mathbf{A}(\chi)$ *in equation (6.45) are strictly diagonally dominant* M *matrices.*

Proof. The construction of $\mathbf{B}(\chi^k)$ using the discretization of $\mathcal{J}V$ as detailed in [23] implies that

$$\sum_u [\mathcal{J}^h]_{\ell,u} \leq 1 \text{ and } [\mathcal{J}^h]_{\ell,u} \geq 0. \tag{6.47}$$

This holds since $p(J)$ in (5.5) is a probability density function. When the grid node (i, j) satisfies $i = 1$ or $i > \hat{i}$ then the ℓ^{th} row of $\mathbf{B}(\chi^k)$is identically zero. This gives (a) and the second part of (b).

In order to prove the remaining part of (b) we note that the row sum is the same as $[\mathbf{A}(\chi^k)\mathbf{e}]_\ell$ with $\mathbf{e} = (1, ..., 1)'$. Since $\mathcal{D}^h_{WW}1 = \mathcal{D}^h_W 1 = \mathcal{D}^h_A 1 = 0$ we see that $\mathcal{L}^h 1 = -(r+\lambda)$. Thus $[\mathbf{A}(\chi^k)\mathbf{e}]_\ell = 1 + (r+\lambda)\Delta\tau$ for $2 \leq i < \hat{i}$. A similar argument shows that $[\mathbf{A}(\chi^k)\mathbf{e}]_\ell = 1 + r\Delta\tau$ for $\hat{i} \leq i < i_{\max}$ or $i = 1$. When $i = i_{\max}$ then the corresponding row is just the ℓ^{th} identity row (since it is just a boundary assignment) and hence its row sum is just unity.

(c) follows since the off-diagonals of $\mathbf{A}(\chi) - \mathbf{B}(\chi)$ and $\mathbf{A}(\chi)$ are non-positive (since the discretization is monotone [26]) and from (b), the row sums are strictly positive. □

Lemma 6.3.1. *If the discretization for the GMWB problem satisfies the conditions required for Proposition 6.3.1 then the discretization satisfies Condition 6.2.1.*

Proof. Because $\mathbf{B}(\chi^k)$ is independent of χ^k, we need only show that

$$\|\mathbf{A}^{-1}(\chi^k)\mathbf{B}(\chi^k)\|_\infty \leq C_1 \tag{6.48}$$

for some constant $C_1 < 1$. Lemma 6.2.3 combined with Proposition 6.3.1 implies that

$$\|\mathbf{A}^{-1}(\chi^k)\mathbf{B}(\chi^k)\|_\infty \;\; \leq \;\; \frac{\lambda\Delta\tau}{1 + (r+\lambda)\Delta\tau} < 1\ . \tag{6.49}$$

To prove that $\|\mathbf{A}^{-1}(\chi)\|_\infty$ is bounded independent of χ, we repeat the above argument setting $\mathbf{B}$ to the identity matrix. □

Remark 6.3.4. *The method in [23] uses the following technique to avoid FFT wrap-around effects. Suppose the original computational domain is* $W \in [0, \hat{W}_{\max}]$. *The domain is extended to* $W_{\max} > \hat{W}_{\max}$ *and the solution in* $[\hat{W}_{\max}, W_{\max}]$ *is assumed to be linear in* W. *This is essentially the same philosophy as used in equation* (5.17) *where the integral is evaluated for* $W > \hat{W}_{\max}$ *by assuming linearity. Note that in* $[\hat{W}_{\max}, W_{\max}]$*, the assumption of linearity causes the jump terms to cancel, leading to a standard Black-Scholes equation [8, 38]. This makes this technique simple to implement.*

6.3.2 Full Matrix Iteration: Direct Control Method

We define the full matrices and vectors used in Algorithm 6.1.2 (assuming discretization (5.24)). Let $\mathbf{y} = (y_1, y_2, \ldots, y_N)'$ be a vector of size N, and let

$$
\begin{aligned}
[\mathbf{A}(\chi^k)\mathbf{y}]_\ell &= [\mathbf{A}^k\mathbf{y}]_\ell = (1-\psi_\ell^k)\left(\frac{1}{\Delta\tau}y_\ell - \mathcal{L}^h y_\ell + \varphi_\ell^k G(\mathcal{D}_W^h y_\ell + \mathcal{D}_A^h y_\ell)\right) \\
&\quad + \Pi\ \psi_\ell^k(\mathcal{D}_W^h y_\ell + \mathcal{D}_A^h y_\ell) \\
&= [\mathbf{A}_D^k\mathbf{y}]_\ell + [\mathbf{A}_L^k\mathbf{y}]_\ell \\
&\quad [\mathbf{A}_D^k\mathbf{y}]_\ell = (1-\psi_\ell^k)\left(\frac{1}{\Delta\tau}y_\ell - \mathcal{L}^h y_\ell + \varphi_\ell^k G\left(\mathcal{D}_W^h y_\ell + \frac{y_\ell}{\Delta A_j^-}\right)\right) \\
&\qquad\qquad + \Pi\ \psi_\ell^k\left(\mathcal{D}_W^h y_\ell + \frac{y_\ell}{\Delta A_j^-}\right) \\
&\quad [\mathbf{A}_L^k\mathbf{y}]_\ell = -(1-\psi_\ell^k)\varphi_\ell^k G\left(\frac{y_{\ell-i_{\max}}}{\Delta A_j^-}\right) - \Pi\psi_\ell^k\left(\frac{y_{\ell-i_{\max}}}{\Delta A_j^-}\right) \\
\big[\mathbf{B}(\chi^k)\mathbf{y}\big]_\ell &= \big[\mathbf{B}^k\mathbf{y}\big]_\ell = (1-\psi_\ell^k)\lambda[\mathcal{J}^h\mathbf{y}]_\ell \\
\big[\mathbf{c}(\chi^k)\big]_\ell &= (1-\psi_\ell^k)\frac{1}{\Delta\tau}V_\ell^n + \Pi\ \psi_\ell^k(1-\kappa)\ .
\end{aligned}
\tag{6.50}
$$

Proposition 6.3.2. *Suppose a positive coefficient discretization [26] is used and the jump operator* $\mathcal{J}^h$ *is discretized using the method in [23] with linear behavior assumed for* $i \geq \hat{i}$ *[23, 54]. Then discretization (6.50) satisfies* *(a)* $\mathbf{B}(\chi) \geq 0$,

(b) The ℓ^{th} *row sum for* $\mathbf{B}(\chi^{\mathbf{k}})$ *is*

$$Row\_Sum_{\ \ell}\ (\ \mathbf{B}(\chi)\ )\ \ \leq\ \ \begin{cases} (1-\psi_\ell^k)\lambda & 2 \leq i < \hat{i} \\ 0 & i = \hat{i}, ..., i_{\max}\ \ or\ i = 1 \end{cases} \tag{6.51}$$

(c) The ℓ^{th} *row sum for* $\mathbf{A}_D(\chi^k)$ *is*

$$\begin{aligned} & Row\_Sum_{\ \ell}\ (\ \mathbf{A}_D(\chi^k)) \\ &= \begin{cases} (1-\psi_\ell^k)\big(\frac{1}{\Delta\tau} + (r+\lambda) + \varphi_\ell^k G \frac{1}{\Delta A_j^-}\big) + \psi_\ell^k \Pi \frac{1}{\Delta A_j^-} & i < \hat{i} \\ (1-\psi_\ell^k)\big(\frac{1}{\Delta\tau} + r + \varphi_\ell^k G \frac{1}{\Delta A_j^-}\big) + \psi_\ell^k \Pi \frac{1}{\Delta A_j^-} & i = \hat{i}, \ldots, i_{\max} - 1\ \ or\ i = 1 \\ \frac{1}{\Delta\tau} + \eta & i = i_{\max} \end{cases} \end{aligned} \tag{6.52}$$

(d) The matrices $\mathbf{A}(\chi) - \mathbf{B}(\chi)$ *and* $\mathbf{A}(\chi)$ *in equation (6.50) are* M *matrices.*

Proof. To prove (a) and (b), we follow the same argument as in the proof of Proposition 6.3.1.

In order to prove (c), first observe the following

$$\begin{aligned} & \mathcal{D}_{WW}^h 1 = 0\ ;\ \mathcal{D}_W^h 1 = 0\ ;\ \mathcal{D}_W^h A_j = 0\ ;\ \mathcal{D}_A^h 1 = 0\ ;\ \mathcal{D}_A^h A_j = 1 \\ & \mathcal{L}^h 1 = \begin{cases} -(r+\lambda) & 2 \leq i < \hat{i} \\ -r & \hat{i} \leq i < i_{\max} \text{ or } i = 1 \end{cases} \end{aligned} \tag{6.53}$$

The row sum of $\mathbf{A}_D$ is $[\mathbf{A}_D(\chi^k)\mathbf{e}]_i$ with $\mathbf{e} = (1, ..., 1)'$, and consequently (c) follows using results (6.53), for $i < i_{\max}$. When $i = i_{\max}$ then from the boundary assignment of equation (3.32), the row sum is just $(1/\Delta\tau + \eta)$.

To prove (d), consider that $\mathbf{A}(\chi) = \mathbf{A}_D(\chi) + \mathbf{A}_L(\chi)$. Note that $\mathbf{A}_D(\chi)$ is block diagonal, and $\mathbf{A}_L(\chi)$ is lower triangular. From (c), the row sums of $\mathbf{A}_D(\chi)$ are strictly positive, and off-diagonals are non-positive since a positive coefficient discretization is used. Hence $\mathbf{A}_D(\chi)$ consists of diagonal blocks, each of which is a strictly diagonally dominant M matrix. Since $\mathbf{A}_L(\chi)$ is non-positive, a straightforward computation shows that $\mathbf{A}(\chi)$ is non-singular and that $\mathbf{A}^{-1}(\chi) \geq 0$. Noting that $\mathbf{A} - \mathbf{B} = \mathbf{A}(\mathbf{I} - \mathbf{A}^{-1}\mathbf{B})$, a similar argument shows that $\mathbf{A}(\chi) - \mathbf{B}(\chi)$ is also an M matrix. □

Lemma 6.3.2. *If the discretization for the GMWB penalty method satisfies the conditions required for Proposition 6.3.2, and in addition*

$$\Pi > A_{\max} \lambda \frac{1 + (r + \lambda)\Delta\tau}{1 + r\Delta\tau} \ , \tag{6.54}$$

then the matrices $\mathbf{A}, \mathbf{B}$ *satisfy Condition 6.2.1, and hence from Theorem 6.2.1, Algorithm 6.1.2 converges.*

Proof. We need to show that there is a constant C_1 such that

$$\|\mathbf{A}^{-1}(\chi^k)\mathbf{B}(\chi^p)\|_\infty \leq C_1 \ , \tag{6.55}$$

where $p = k, k-1$. Consider an arbitrary vector $\mathbf{z}$, and a vector $\mathbf{y}$ such that

$$\mathbf{A}(\chi^k)\mathbf{y} \ = \ \mathbf{B}(\chi^p)\mathbf{z} \ , \tag{6.56}$$

then condition (6.55) is equivalent to requiring that

$$\frac{\|\mathbf{y}\|_\infty}{\|\mathbf{z}\|_\infty} \leq C_1 < 1 \ . \tag{6.57}$$

From equation (6.50), we can see that $[\mathbf{Ae}]_\ell = 0; \ i < \hat{i}, \ \psi_\ell^k = 1$, hence we are obliged to use a different method from that in Section 6.3.1 to prove this result.

First, note that

$$\begin{aligned} \left[\mathbf{A}(\chi^k)\mathbf{y}\right]_\ell &= \left[\mathbf{B}(\chi^p)\mathbf{z}\ \right]_\ell \ ;\ \ell = i + (j-1)i_{\max}\ ;\ 2 \leq i < \hat{i} && (6.58)\\ \left[\mathbf{A}(\chi^k)\mathbf{y}\right]_\ell &= 0 \qquad\qquad\ ;\ \ell = i + (j-1)i_{\max}\ ;\ i \geq \hat{i} \text{ or } i = 1 && (6.59) \end{aligned}$$

due to the linear behavior assumed for $i \geq \hat{i}$ (see equation(6.51)) [23]. Define a bounding grid function $\hat{\mathbf{y}}$

$$[\hat{\mathbf{y}}]_\ell = \frac{\|\mathbf{z}\|_\infty \lambda \Delta\tau}{1 + (r+\lambda)\Delta\tau} + A_j \frac{\|\mathbf{z}\|_\infty \lambda}{\Pi} \ ;\ \ \ell = i + (j-1)i_{\max}\ . \tag{6.60}$$

Noting properties (6.53), and substituting equation (6.60) into equation (6.50) gives

$$\begin{aligned} \left[\mathbf{A}(\chi^k)\hat{\mathbf{y}}\right]_\ell &\geq \|\mathbf{z}\|_\infty \lambda\ ;\ 2 \leq i < \hat{i} && (6.61)\\ \left[\mathbf{A}(\chi^k)\hat{\mathbf{y}}\right]_\ell &\geq 0 \qquad\ ;\ i \geq \hat{i} \text{ or } i = 1\ . && (6.62) \end{aligned}$$

Subtracting equation (6.58) from equation (6.61) yields (noting properties (a) and (b) of $\mathbf{B}$ in Proposition 6.3.2)

$$\begin{aligned} \left[\mathbf{A}(\chi^k)\left(\hat{\mathbf{y}} - \mathbf{y}\right)\right]_\ell &\geq \|\mathbf{z}\|_\infty \lambda - [\mathbf{B}(\chi^p)\mathbf{z}]_\ell \\ &\geq \|\mathbf{z}\|_\infty \lambda - \|\mathbf{z}\|_\infty \lambda \\ &= 0 \qquad\qquad ;\ \ 2 \leq i < \hat{i}\ \ . && (6.63) \end{aligned}$$

Similarly, subtract equation (6.59) from equation (6.62) to give

$$\left[\mathbf{A}(\chi^k)\left(\hat{\mathbf{y}} - \mathbf{y}\right)\right]_\ell \geq 0\ \ ;\ \ \ i \geq \hat{i} \text{ or } i = 1\ \ . \tag{6.64}$$

Hence in all cases

$$\mathbf{A}(\chi^k)\,(\hat{\mathbf{y}} - \mathbf{y}) \geq 0 \quad . \tag{6.65}$$

Since $\mathbf{A}(\chi^k)$ is an M matrix, we have that $\mathbf{y} \leq \hat{\mathbf{y}}$. Similar arguments give $\mathbf{y} \geq -\hat{\mathbf{y}}$. Hence

$$\|\mathbf{y}\|_\infty \leq \frac{\|\mathbf{z}\|_\infty \lambda \Delta\tau}{1 + (r+\lambda)\Delta\tau} + A_{\max}\frac{\|\mathbf{z}\|_\infty \lambda}{\Pi} \quad . \tag{6.66}$$

If we require that $\|\mathbf{y}\|_\infty < \|\mathbf{z}\|_\infty$, we obtain the condition on Π

$$\Pi > A_{\max}\lambda \frac{1 + (r+\lambda)\Delta\tau}{1 + r\Delta\tau} \quad . \tag{6.67}$$

□

6.4 Efficient Block Matrix Implementation

Let $v_{*,j} = V_{*,j}^{n+1}$, and let $(v_{*,j})^k$ be the k^{th} iterate for $v_{*,j}$. From the boundary condition (2.16), we can observe that $v_{*,1}$ can be computed without any knowledge of interior nodes in the computational domain. To ensure a positive coefficient discretization, the $\mathcal{D}_A^h$ operator is always backward differenced, hence $v_{*,j}$ depends only on $v_{*,j-1}$ for $j > 1$. This special structure of the system makes the iteration more efficient when we solve $v_{*,j}$ before proceeding to solve $v_{*,j+1}$. We write the full matrix system in (6.50) as an equivalent block matrix linear system as follows

$$\begin{pmatrix} \mathbf{A_1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{A_2}(\chi_{*,2}) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A_{j_{\max}}}(\chi_{*,j_{\max}}) \end{pmatrix} \begin{pmatrix} v_{*,1} \\ v_{*,2} \\ \vdots \\ v_{*,j_{\max}} \end{pmatrix}$$
$$= \begin{pmatrix} \mathbf{B_1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{B_2}(\chi_{*,2}) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{B_{j_{\max}}}(\chi_{*,j_{\max}}) \end{pmatrix} \begin{pmatrix} v_{*,1} \\ v_{*,2} \\ \vdots \\ v_{*,j_{\max}} \end{pmatrix} + \begin{pmatrix} c_{*,1} \\ c_{*,2}(\chi_{*,2}, v_{*,1})) \\ \vdots \\ c_{*,j_{\max}}(\chi_{*,j_{\max}}, v_{*,j_{\max}-1}) \end{pmatrix} , \tag{6.68}$$

with

$$\chi_{i,j} = \underset{\chi_{i,j} \in X}{\arg\max} \Big[-\mathbf{A_j}(\chi_{*,j}) v_{*,j} + \mathbf{B_j}(\chi_{*,j}) v_{*,j} + c_{*,j}(\chi_{*,j}, v_{*,j-1}) \Big]_i . \tag{6.69}$$

Note that $\mathbf{A_1}, \mathbf{B_1}, c_{*,1}$ are independent of χ. Each smaller block matrix system $\mathbf{A_j} v_{*,j} = \mathbf{B_j} v_{*,j} + c_{*,j}$ is then resolved by using a fixed point policy iteration as in Algorithm 6.1.2 with the previous computed $v_{*,j-1}$ appearing only in $c_{*,j}$. The detailed procedure is given in Algorithm 6.4.1.

Algorithm 6.4.1 Block Matrix Fixed Point Policy Iteration

```
1: Solve v_{*,1} from A_1 v_{*,1} = B_1 v_{*,1} + c_{*,1}
2: for j = 2, 3, ..., j_max do
3:     With initial solution (v_{*,j})^0 = V^n_{*,j}, use Algorithm 6.1.2 to solve v_{*,j} from
       A_j(χ_{*,j}) v_{*,j} = B_j(χ_{*,j}) v_{*,j} + c_{*,j}(χ_{*,j}, v_{*,j-1})
4: end for
```

6.4.1 Block Implementation: Penalty Method

Recall in Section 5.3.2, equation (5.21) is written equivalently as equation (5.23) assuming that the first derivative in the A direction is always backward differenced. We can then represent the linear relationships given in discretized equations (5.23) and (3.32) in block matrix form. Let $\mathbf{u} = ((u_{*,1})', (u_{*,2})', \ldots, (u_{*,j_{\max}})')'$ be an N length vector. The $i_{\max} \times i_{\max}$ square matrices $\mathbf{A_j}, \mathbf{B_j}$ and the vector $c_{*,j}$ of size $i_{\max}$ are given by

$$
\begin{aligned}
\left[\mathbf{A_j}(\chi^k_{*,j})u_{*,j}\right]_i \quad = [\mathbf{A_j}{}^k u_{*,j}]_i &= \frac{1}{\Delta\tau}u_{i,j} - \mathcal{L}^h u_{i,j} + \varphi^k_{i,j} G[\frac{1}{\Delta A^-_j}u_{i,j} + \mathcal{D}^h_W u_{i,j}] \\
&\quad + \frac{\psi^k_{i,j}}{\varepsilon}[\frac{1}{\Delta A^-_j}u_{i,j} + \mathcal{D}^h_W u_{i,j}] \\
\left[\mathbf{B_j}(\chi^k_{*,j},)u_{*,j}\right]_i \quad = [\mathbf{B_j}{}^k u_{*,j}]_i &= \lambda[\mathcal{J}^h_j u_{*,j}]_i \\
\left[c_{*,j}(\chi^k_{*,j}, u_{*,j-1})\right]_i = \quad \left[c^k_{*,j}\right]_i &= \varphi^k_{i,j} G + \psi^k_{i,j}\left[\frac{(1-\kappa)}{\varepsilon} + \kappa G\right] + \frac{1}{\Delta\tau}V^n_{i,j} \\
&\quad + \left(\varphi^k_{i,j} G + \frac{\psi^k_{i,j}}{\varepsilon}\right)\frac{1}{\Delta A^-_j}u_{i,j-1}
\end{aligned} \tag{6.70}
$$

with controls

$$
\begin{aligned}
(\chi_{i,j})^k &= \left(\varphi^k_{i,j}, \psi^k_{i,j}\right) = \underset{\substack{\varphi\in\{0,1\},\psi\in\{0,1\} \\ \varphi\psi=0}}{\arg\max}\Big[-\mathbf{A_j}(\varphi_{*,j}, \psi_{*,j})u_{*,j} \\
&\qquad + \mathbf{B_j}(\varphi_{*,j}, \psi_{*,j})u_{*,j} + c_{*,j}(\varphi_{*j}, \psi_{*,j}, u_{*,j-1})\Big]_i ,
\end{aligned} \tag{6.71}
$$

where $\mathcal{J}^h_j$ is the subblock of $\mathcal{J}^h$ which operates on $u_{*,j}$, where $u_{*,-1} = 0$. The discretized equations (5.21) become a set of equations as follows:

$$
\begin{aligned}
-\mathbf{A_j}u_{*,j} + \mathbf{B_j}u_{*,j} + c_{*,j} = 0 \ , \quad j = \ & 1 \\
\sup_{\chi_{i,j}\in X}\left[-\mathbf{A_j}(\chi_{*,j})u_{*,j} + \mathbf{B_j}(\chi_{*,j})u_{*,j} + c_{*,j}(\chi_{*,j}, u_{*,j-1})\right] = 0 \ , \quad j = \ & 2, 3, \ldots, j_{\max}
\end{aligned} \tag{6.72}
$$

Remark 6.4.1. *We have written the matrix* $\mathbf{B_j} = \mathbf{B_j}^k$ *although there is no explicit dependence on* $(\varphi^k_{*,j}, \psi^k_{*,j})$ *in this case in order to use the general form of the previous section.*

Recall that in order to ensure convergence to the viscosity solution of equation (2.6), the discretization must be monotone, consistent and l_∞ stable [5]. A positive coefficient discretization guarantees monotonicity [26]. The positive coefficient condition and the discretization of the jump term as in [23] give the following result.

Proposition 6.4.1. *Suppose a positive coefficient discretization [26] is used and the jump operator* $\mathcal{J}^h_j$ *is discretized using the method in [23] with linear behavior assumed for* $i \geq \hat{i}$ *[23, 54]. Then*

(a) $\mathbf{B_j}(\chi_{*,j}) \geq 0$,

(b) The i^{th} *row sums for* $\mathbf{A_j}(\chi^k_{*,j})$ *and* $\mathbf{B_j}(\chi_{*,j}k)$ *are*

$$
\begin{aligned}
\mathit{Row\_Sum}_{\ i}\ (\ \mathbf{A_j}(\chi^k_{*,j})\ ) \ &= \ \begin{cases} \frac{1}{\Delta\tau} + (r+\lambda) + (\varphi^k_{i,j}G + \frac{\psi^k_{i,j}}{\varepsilon})\frac{1}{\Delta A^-_j} & 2 \leq i < \hat{i} \\ \frac{1}{\Delta\tau} + r + (\varphi^k_{i,j}G + \frac{\psi^k_{i,j}}{\varepsilon})\frac{1}{\Delta A^-_j} & i = \hat{i}, \ldots, i_{\max} - 1 \ \mathit{or}\ i = 1 \\ \frac{1}{\Delta\tau} + \eta & i = i_{\max} \end{cases} \\
\mathit{Row\_Sum}_{\ i}\ (\ \mathbf{B_j}(\chi^k_{*,j})\ ) \ &\leq \ \begin{cases} \lambda & 2 \leq i < \hat{i} \\ 0 & i = \hat{i}, \ldots, i_{\max}\ \mathit{or}\ i = 1 \end{cases}
\end{aligned}
\tag{6.73}
$$

(c) The matrices $\mathbf{A_j}(\chi_{*,j}) - \mathbf{B_j}(\chi_{*,j})$ *and* $\mathbf{A_j}(\chi_{*,j})$ *in equation (6.72) are diagonally dominant M matrices.*

Proof. (a) and the second part of (b) follow in similar fashion as in the proof of Proposition 6.3.1.

In order to prove the remaining part of (b) we note that the row sum is the same as $[\mathbf{A_j}(\chi^k_{*,j})\mathbf{e}]_i$ with $\mathbf{e} = (1, ..., 1)'$. Noting properties (6.53), we see that $[\mathbf{A_j}(\chi^k_{*,j})\mathbf{e}]_i = \frac{1}{\Delta\tau} + (r+\lambda) + (\varphi^k_{i,j}G + \frac{\psi^k_{i,j}}{\varepsilon})\frac{1}{\Delta A^-_j}$ for $2 \le i < \hat{i}$. A similar argument shows that $[\mathbf{A_j}(\chi^k_{*,j})\mathbf{e}]_i = \frac{1}{\Delta\tau} + r + (\varphi^k_{i,j}G + \frac{\psi^k_{i,j}}{\varepsilon})\frac{1}{\Delta A^-_j}$ for $\hat{i} \le i < i_{\max}$ or $i = 1$. When $i = i_{\max}$ then from the boundary assignment of equation (3.32), its row sum is just $(1/\Delta\tau + \eta)$.

To prove (c), note that $\mathbf{A_j}$ and $(\mathbf{A_j} - \mathbf{B_j})$ have non-positive off-diagonals (since a positive coefficient discretization is used [26]). From (b), the row sums of $(\mathbf{A_j} - \mathbf{B_j})$, $\mathbf{A_j}$ are strictly positive. Hence $\mathbf{A_j}$ and $(\mathbf{A_j} - \mathbf{B_j})$ are M matrices [53]. □

We can now state the convergence result for the block matrix method, using the penalty formulation.

Lemma 6.4.1. *If the discretization for the GMWB penalty method satisfies the conditions required for Proposition 6.4.1, then the matrices* $\mathbf{A_j}, \mathbf{B_j}$ *satisfy Condition 6.2.1, and hence from Theorem 6.2.1, Algorithm 6.4.1 converges.*

Proof. Because $\mathbf{B_j}(\chi^k_{*,j})$ is independent of $\chi^k_{*,j}$, we only need to show that

$$\|\mathbf{A_j}^{-1}(\chi^k_{*,j})\mathbf{B_j}(\chi^k_{*,j})\|_\infty \le C_1 \tag{6.74}$$

for some constant $C_1 < 1$. From Lemma 6.2.3 and Proposition 6.4.1, it implies that

$$\|\mathbf{A_j}^{-1}(\chi^k_{*,j})\mathbf{B_j}(\chi^k_{*,j})\|_\infty \le \max_{i,j}\left[\frac{\lambda}{\frac{1}{\Delta\tau} + (r+\lambda) + (\varphi^k_{i,j}G + \frac{\psi^k_{i,j}}{\varepsilon})\frac{1}{\Delta A^-_j}}\right] < 1 \tag{6.75}$$

By setting $\mathbf{B_j} = \mathbf{I}$, we obtain immediately that $\mathbf{A_j}^{-1}(\chi^k_{*,j})$ is bounded independent of χ. □

6.4.2 Block Implementation: Direct Control

To use block matrix form to represent the discretized direct control equation, we use equation (5.26), which is equivalent as equation (5.24) with the first derivative in A direction backward differenced. We represent the discretization (5.26) in terms of matrices $\mathbf{A_j}, \mathbf{B_j}$ and vector $\mathbf{c_j}$, given by

$$
\begin{aligned}
[\mathbf{A_j}(\chi^k_{*,j})u_{*,j}]_i = [\mathbf{A_j}{}^k u_{*,j}]_i &= (1-\psi^k_{i,j})\left(\frac{1}{\Delta\tau}u_{i,j} - \mathcal{L}^h u_{i,j} + \varphi^k_{i,j} G(\frac{u_{i,j}}{\Delta A^-_j} + \mathcal{D}^h_W u_{i,j})\right) \\
&\quad + \psi^k_{i,j}\Pi\ G(\frac{u_{i,j}}{\Delta A^-_j} + \mathcal{D}^h_W u_{i,j}) \\
\left[\mathbf{B_j}(\chi^k_{*,j})u_{*,j}\right]_i = [\mathbf{B_j}{}^k u_{*,j}]_i &= (1-\psi^k_{i,j})\lambda[\mathcal{J}^h_j u_{*,j}]_i \\
\left[c_{*,j}(\chi^k_{*,j}, u_{*,j-1})\right]_i = [c_{*,j}]^k_i &= (1-\psi^k_{i,j})\frac{1}{\Delta\tau}V^n_{i,j} + (1-\psi^k_{i,j})\varphi^k_{i,j}\frac{G}{\Delta A^-_j}u^{n+1}_{i,j-1} \\
&\quad + \psi^k_{i,j}\Pi(1-\kappa) + \psi^k_{i,j}\Pi\frac{1}{\Delta A^-_j}u^{n+1}_{i,j-1} \qquad (6.76)
\end{aligned}
$$

with controls

$$
\begin{aligned}
(\chi_{i,j})^k &= \left(\varphi^k_{i,j}, \psi^k_{i,j}\right) \in \underset{\substack{\varphi\in\{0,1\},\psi\in\{0,1\} \\ \varphi\psi=0}}{\arg\max} \Big[-\mathbf{A_j}(\varphi_{*,j},\psi_{*,j})u_{*,j} \\
&\quad + \mathbf{B_j}(\varphi_{*,j},\psi_{*,j})u_{*,j} + c_{*,j}(\varphi_{*,j},\psi_{*,j},u_{*,j-1})\Big]_i\ . \qquad (6.77)
\end{aligned}
$$

If we write control as in the form of (6.2), then the discretized equations (5.26) become a set of equations as follows:

$$
\begin{aligned}
-\mathbf{A_j}u_{*,j} + \mathbf{B_j}u_{*,j} + c_{*,j} = 0\ , &\qquad j = 1 \\
\sup_{\chi_{i,j}\in X}\left[-\mathbf{A_j}(\chi_{*,j})u_{*,j} + \mathbf{B_j}(\chi_{*,j})u_{*,j} + c_{*,j}(\chi_{*,j},u_{*,j-1})\right]_i = 0\ , &\qquad j = 2,\ldots,j_{\max} \quad (6.78)
\end{aligned}
$$

Proposition 6.4.2. *Suppose a positive coefficient discretization [26] is used and the jump* *operator $\mathcal{J}^h$ is discretized using the method in [23] and linear behavior is assumed for $i > \hat{i}$ [23, 54]. Then*

(a) $\mathbf{B_j}(\chi^k_{*,j}) \geq 0$,

(b) The i^{th} row sums for $\mathbf{A_j}(\chi^k_{*,j})$ and $\mathbf{B_j}(\chi^k_{*,j})$ are

$$
\begin{aligned}
&\textit{Row\_Sum}_{\ i} \ (\ \mathbf{A_j}(\chi^k_{*,j})\ ) \\
&= \begin{cases}
(1-\psi^k_{i,j})\big(\frac{1}{\Delta\tau} + (r+\lambda) + \varphi^k_{i,j} G \frac{1}{\Delta A^-_j}\big) + \psi^k_{i,j}\Pi\frac{1}{\Delta A^-_j} & 2 \leq i < \hat{i} \\
(1-\psi^k_{i,j})\big(\frac{1}{\Delta\tau} + r + \varphi^k_{i,j} G \frac{1}{\Delta A^-_j}\big) + \psi^k_{i,j}\Pi\frac{1}{\Delta A^-_j} & i = \hat{i}, \ldots, i_{\max}-1 \ \textit{or}\ i = 1 \\
\frac{1}{\Delta\tau} + \eta & i = i_{\max}
\end{cases}
\end{aligned}
\tag{6.79}
$$

$$
\begin{aligned}
&\textit{Row\_Sum}_{\ i} \ (\ \mathbf{B_j}(\chi^k_{*,j})\ ) \\
&= \begin{cases}
(1-\psi^k_{i,j})\lambda & i < \hat{i} \\
0 & i = \hat{i}, \ldots, i_{\max}\ ,
\end{cases}
\end{aligned}
$$

(c) The matrices $\mathbf{A_j}(\chi_{*,j}) - \mathbf{B_j}(\chi_{*,j})$ and $\mathbf{A_j}(\chi_{*,j})$ in equation (6.78) are strictly diagonally dominant M matrices.

Proof. The proof follows using similar arguments as in the proof of Proposition 6.4.1. □

Define $\Delta A_{\max} = \max_j [A_j - A_{j-1}]$. The following Lemma gives the conditions under which Algorithm 6.4.1 converges.

Lemma 6.4.2. *If the discretization for the GMWB direct control method satisfies the conditions required for Proposition 6.4.2 and $\Pi > \lambda \Delta A_{\max}$, then the matrices $\mathbf{A_j}, \mathbf{B_j}$ defined in equation* (6.78) *satisfy Condition 6.2.1, hence Algorithm 6.4.1 converges from Theorem 6.2.1.*

Proof. Suppose

$$\arg\max_i \left[\frac{\text{Row\_Sum }_i\ (\mathbf{B_j})}{\text{Row\_Sum }_i\ (\mathbf{A_j^{-1}})} \right] = p\ . \tag{6.80}$$

If $2 \leq p < \hat{i}$, and $\psi_{p,j}^k = 0$, then Lemma 6.2.3 and Proposition 6.4.2 implies

$$\|\mathbf{A_j}^{-1}(\chi_{*,j}^k)\mathbf{B_j}(\chi_{*,j}^k)\|_\infty \leq \frac{\lambda}{\frac{1}{\Delta\tau} + (r+\lambda) + \frac{\varphi_{p,j}^k G}{\Delta A_j^-}} < 1 \tag{6.81}$$

When $p \geq \hat{i}$ or $p = 1$, or $\psi_{p,j}^k = 1$, Row_Sum $_i$ $(\mathbf{B_j}(\chi_{*,j}^k)) = 0$. In either case bound (6.81) holds.

If $2 \leq p < \hat{i}$, $\psi_{i,j}^{k-1} = 0$ and $\psi_{i,j}^k = 1$

$$\|\mathbf{A_j}^{-1}(\chi_{*,j}^k)\mathbf{B_j}(\chi_{*,j}^{k-1})\|_\infty \leq \frac{\lambda}{\Pi \frac{1}{\Delta A_j^-}} < \frac{\Delta A_j^-}{\Delta A_{\max}} \leq 1\ . \tag{6.82}$$

In all other cases, $\|\mathbf{A_j}^{-1}(\chi_{*,j}^k)\mathbf{B_j}(\chi_{*,j}^{k-1})\|_\infty \leq C_1 < 1$ unconditionally. Repeating the above argument setting $\mathbf{B_j}(\chi_{*,j})$ to the identity shows that $\|\mathbf{A_j}^{-1}(\chi_{*,j})\|_\infty$ is bounded independent of χ. □

Remark 6.4.2. *Choosing a scaling factor which satisfies condition* (iii) *in Condition 6.2.1 means that this same scaling factor must be used in the optimization step in line* 3 *of Algorithm 6.1.2. Consequently, choosing different scaling factors will result, in general, in different choices for control at each iteration.*

6.5 Summary

The main results of this chapter are as follows

- The conventional policy iteration suffers from computational inefficiency when solving discretized controlled HJB PIDEs in general. A modified policy iteration, *fixed point policy iteration*, is developed by using a splitting method. This method permits efficient solution of the discretized equations.

- Condition 6.2.1 is a sufficient condition for the fixed point policy iteration to converge. This condition normally is easy to satisfy when a positive coefficient discretization is used.

- Both the penalty discretization and the direct control discretization can be verified to satisfy Condition 6.2.1, hence the discrete equations can be solved by using a fixed point policy iteration each timestep.

- A block matrix fixed point policy iteration is more efficient since it takes advantage of the special structure of the singular control formulation of the GMWB pricing matrix.

Chapter 7

Fixed Point Policy Iteration: Numerical Results

In this chapter, several fixed point policy iteration numerical examples are presented. We price an example GMWB contract used in [13]. The contract parameters are given in Table 7.1. The jump diffusion parameters are given in Table 7.2, which are typical market data [2]. Table 7.3 gives the mesh size and timestep parameters. In the localized computational domain, we set $W_{\max} = 1000\omega_0$. Section 7.1 presents our numerical results. Section 7.2 summarizes the main results of this chapter.

7.1 Results

7.1.1 No-arbitrage fee

Using the efficient block matrix fixed point policy as described in Algorithm 6.4.1, Table 7.4 presents the fair insurance fee η charged by the insurance company computed by solving

Parameter	Value
Expiry time T	10.0 years
Interest rate r	0.05
Maximum no penalty withdrawal rate G	10/year
Withdrawal penalty κ	0.10
Initial lump-sum premium ω_0	100
Initial guarantee account balance $A(0)$	100
Initial personal annuity account balance $W(0)$	100

Table 7.1: A sample GMWB contract parameters used in the numerical experiments.

Parameter	Value
ζ	.45
ν	-.9
λ	.1

Table 7.2: Jump diffusion parameters.

Refine Level	W Nodes	A Nodes	Time steps
1	125	111	120
2	249	221	240
3	497	441	480
4	993	881	960
5	1985	1761	1920

Table 7.3: Grid and timestep data for convergence experiments. At each refinement, new fine grid nodes are introduced between each two coarse grid nodes, and the timesteps are halved.

the following equation [30]

$$V(\eta; W = \omega_0, A = \omega_0, \tau = T) \;\; = \;\; \omega_0 \tag{7.1}$$

with the nonlinear convergence tolerance for the fixed point-policy iteration set to

$$\max_{\ell} \frac{|v_\ell^{k+1} - v_\ell^k|}{\max(scale, |v_\ell^{k+1}|)} \;\; < \;\; 10^{-8} \; . \tag{7.2}$$

Newton iteration is used to solve the equation with the convergence tolerance

$$\frac{|\eta^{m+1} - \eta^k|}{\max(\eta^{m+1}, \eta^m)} < 10^{-8} \; , \tag{7.3}$$

where η^m is the $m'th$ iterate.

The results show that with jump diffusion assumptions, the fair insurance fee is noticeably higher than with the standard Geometric Brownian Motion assumption.

7.1.2 Full Matrix Iteration vs Block Matrix Iteration

Table 7.5 presents the convergence results for the GMWB for the penalty method and the direct control method. Both the full matrix fixed point policy iteration scheme as described in Algorithm 6.1.2 and the block matrix fixed point policy iteration scheme as described in Algorithm 6.4.1 are used.

The penalty parameter was set to $1/\varepsilon = 10^4 \omega_0/\Delta\tau$. Some intuition may be useful at this point to justify the choice of ε. Recall from equation (3.1) that $1/\varepsilon = \vartheta$, which is the maximum rate of withdrawal. Hence $1/\varepsilon$ should have units of $dollars/time$. If

$$\frac{1}{\varepsilon} = \frac{\omega_0}{\Delta\tau} \; , \tag{7.4}$$

Refine	$\sigma = 0.2$		$\sigma = 0.3$	
Level	Fair Fee	Ratio	Fair Fee	Ratio
		Jump Diffusion Case		
1	0.034427	N/A	0.046890	N/A
2	0.032854	N/A	0.045789	N/A
3	0.032439	3.79	0.045536	4.34
4	0.032329	3.78	0.045471	3.91
5	0.032297	3.37	0.045452	3.35
		No Jump Diffusion Case		
1	0.018705	N/A	0.033899	N/A
2	0.015245	N/A	0.031904	N/A
3	0.014245	3.19	0.031431	4.22
4	0.013961	3.38	0.031319	4.22
5	0.013886	3.80	0.031286	3.43

Table 7.4: Convergence study for the fair insurance fee η value with and without jump diffusions. Contract parameters are given in Table 7.1. Jump diffusion parameters are given in Table 7.2. Ratio is the ratio of successive changes in the solution as the mesh is refined.

then the entire initial investment can be withdrawn in a single timestep, which would be (effectively) an infinite rate. However, from equation (3.6), we can see that it is desirable to make ε small at any finite grid size so that the term $\varepsilon(V^{\varepsilon}_{\tau} - \mathcal{L}_{\mathcal{G}} V^{\varepsilon} - \kappa G)$ is small. Hence we choose $1/\varepsilon = 10^4 \omega_0 / \Delta\tau$.

The scaling factor parameter is set to $\Pi = 10^3$ in Algorithm 6.1.2, which satisfies the condition (6.67) and $\Pi = 1$ in Algorithm 6.4.1, which satisfies the condition in Lemma 6.4.2. The nonlinear convergence tolerance for the fixed point-policy iteration is given by

$$\max_{\ell} \frac{|v_{\ell}^{k+1} - v_{\ell}^{k}|}{\max(scale, |v_{\ell}^{k+1}|)} \quad < \quad 10^{-6} \ . \tag{7.5}$$

The results show that both the penalty method and the direct control method converge and the computed results from both methods agree to seven digits. The number of iterations for the full matrix fixed point policy iteration scheme is an order magnitude larger

Refinement	Penalty Method			Direct Control Method		
Level	Value	Itns/step	Ratio	Value	Itns/step	Ratio
Algorithm 6.4.1 Block Matrix Iteration						
0	101.19905	4.28	N/A	101.19906	4.13	N/A
1	100.33789	4.16	N/A	100.33789	4.09	N/A
2	100.08441	4.03	3.40	100.08441	3.98	3.40
3	100.02144	3.89	4.03	100.02145	3.93	4.03
4	100.00498	3.89	3.82	100.00498	3.89	3.82
5	100.00003	3.88	3.33	100.00003	3.87	3.33
Algorithm 6.1.2 Full Matrix Iteration						
0	100.19905	83.90	N/A	101.19906	57.63	N/A
1	100.33789	189.72	N/A	100.33789	100.83	N/A

Table 7.5: Convergence experiments for the GMWB guarantee value at $t = 0$ and $W = A = \omega_0 = 100$ using penalty method ($1/\varepsilon = 10^4\omega_0/\Delta\tau$) and direct control method ($\Pi = 1$). Contract parameters are given in Table 7.1. Volatility $\sigma = 0.3$ and fair insurance fee $\eta = 0.045452043$ are imposed. Itns/step refers to the average number of iterations per timestep for the lines $2-4$ in Algorithm 6.4.1 and lines $2-8$ in Algorithm 6.1.2 respectively. Ratio is the ratio of successive changes in the solution as the mesh/timesteps are refined. Since the no-arbitrage fee is imposed, the numerical solution should converge to $Value = \omega_0 = 100$.

than the number of iterations for the block matrix fixed point policy iteration scheme. This is because the full matrix iteration does not take advantage of the special structure of the matrix by computing $v_{*,j}^{n+1}$ before computing $v_{*,j-1}^{n+1}$. The ratio of computational cost for these two methods is approximately equal to the ratio of iterations per timestep. In the rest of the numerical examples, we will consider only the block matrix scheme.

7.1.3 Fixed Point Policy Iteration vs Full Policy Iteration

Using fully implicit timestepping, Table 7.6 presents the convergence results for the GMWB value with respect to two volatility values, assuming the no-arbitrage insurance fee is imposed. We compared the block matrix fixed point-policy in Algorithm 6.4.1 with the block matrix implementation of full policy iteration in Algorithm 6.1.1 [1]. A simple iteration (6.11) method was used to solve the full policy iteration matrix equations. The nonlinear convergence tolerance for the policy and fixed point-policy iteration is set to 10^{-8} A relative update tolerance of 10^{-8} was also used for the simple iteration in Algorithm 6.11.

These two schemes show no difference in computed values to seven digits. However the fixed point-policy scheme requires less than half the iterations that is required by the full policy iteration. The computational cost for these methods is dominated by the FFTs required to carry out the dense matrix-vector multiply, hence the CPU time is proportional to the number of iterations. The results show that the fixed point-policy iteration scheme requires significantly smaller computational cost compared to the full policy scheme.

[1] Using the block matrix definitions in Section 6.4, it is straight forward to implement the block matrix full policy iteration scheme.

Refine Level	Value	Total Itns/Step		Outer Itns/Step	Ratio
		Fixed Pt Policy	Full Policy	Full Policy	
$\sigma = 0.2, \eta = 0.032297$					
1	100.6090	4.67	10.16	3.88	N/A
2	100.1775	4.57	9.32	3.92	N/A
3	100.0471	4.33	9.08	3.98	3.31
4	100.0108	4.21	8.64	4.02	3.59
5	99.9999	4.08	8.04	4.05	3.32
$\sigma = 0.3, \eta = 0.045452$					
1	100.3375	4.91	10.94	4.18	N/A
2	100.0842	4.84	10.19	4.32	N/A
3	100.0213	4.64	9.89	4.38	4.03
4	100.0049	4.65	9.47	4.45	3.83
5	100.0000	4.44	8.81	4.42	3.34

Table 7.6: Iteration and convergence experiments for the GMWB guarantee value at $t = 0$ and $W = A = \omega_0 = 100$ using the fixed point-policy and full policy schemes. Contract parameters are given in Table 7.1. Jump diffusion parameters are given in Table 7.2. Total Itns/step refers to the average number of iterations per timestep to solve the equation. Outer Itns/Step refers to the average number of outer iterations in the full policy iteration scheme. Ratio is the ratio of successive changes in the solution as the mesh/timesteps are refined. Since the fair insurance fee is imposed, the numerical solution should converge to $Value = \omega_0 = 100$. All methods used the same number of timesteps. Fully implicit timestepping is used.

7.1.4 Effect of Maximal Use of Central Differencing on V_W term

Table 7.7 presents the convergence results for the GMWB value with respect to two volatility values, assuming the no-arbitrage insurance fee is imposed and there is no jump diffusion process (i.e. $\lambda = 0$). In this case, the matrix $\mathbf{B} = \mathbf{0}$, so the fixed point policy iteration degenerates to conventional policy iteration. The penalty method is used in the experiment.

Aside from fully implicit timestepping, we have also carried out some tests using Crank Nicolson timestepping, using an obvious modification of equation (3.22). Note that convergence has only been proven for the fully implicit method since Crank Nicolson timestepping is not monotone in general. The differencing method for the V_W term, which uses central differencing as much as possible, is also compared with forward or backward differencing only for the V_W term.

The Itns/step column in Table 7.7 shows the average number of iterations in each timestep required for lines $2-4$ in Algorithm 6.4.1. About $3-4$ non-linear iterations per timestep are required for the $\sigma = .2$ case, and about $4-5$ iterations per timestep are required in the $\sigma = .3$ case. The convergence ratio in the table is the ratio of successive changes in the solution, as the timestep and mesh size are reduced by a factor of two.

The number of iterations per timestep appears to be fairly insensitive to the grid size in Table 7.7. Note that since the timestep is reduced as the grid spacing is reduced, we have an excellent initial solution estimate at each timestep. This is consistent with the results for time dependent problems as reported in [9]. For steady state problems, [47] and [9] report grid dependent number of iterations for policy iteration.

It can be seen that using central differencing as much as possible for the V_W term leads to more rapid convergence (as the mesh is refined) compared to pure forward or backward differencing for this term. Rather unexpectedly, the convergence ratios for both Crank Nicolson and fully implicit timestepping are similar. Figure 7.1 shows a plot of V_{tt} versus

Refinement	Central Differencing First			For/Backward Differencing Only		
Level	Value	Itns/step	Ratio	Value	Itns/step	Ratio
$\sigma = 0.2, \eta = 0.013886$						Fully Implicit Method
1	101.3114	3.51	N/A	101.6030	3.47	N/A
2	100.4488	3.62	N/A	100.6914	3.55	N/A
3	100.1267	3.70	2.68	100.2816	3.66	2.22
4	100.0270	3.77	3.23	100.1082	3.74	2.36
5	99.9999	3.89	3.69	100.0346	3.88	2.36
$\sigma = 0.2, \eta = 0.013886$						Crank Nicolson Method
1	101.3085	3.39	N/A	101.6017	3.35	N/A
2	100.4474	3.49	N/A	100.6909	3.42	N/A
3	100.1261	3.55	2.68	100.2815	3.52	2.22
4	100.0262	3.55	3.22	100.1082	3.52	2.36
5	99.9995	3.57	3.75	100.0343	3.55	2.35
$\sigma = 0.3, \eta = 0.031286$						Fully Implicit Method
1	100.5946	4.19	N/A	100.8998	4.10	N/A
2	100.1488	4.31	N/A	100.3363	4.26	N/A
3	100.0357	4.33	3.94	100.1173	4.32	2.57
4	100.0081	4.39	4.09	100.0435	4.38	2.97
5	100.0000	4.38	3.40	100.0167	4.37	2.76
$\sigma = 0.3, \eta = 0.031286$						Crank Nicolson Method
1	100.5882	4.01	N/A	100.8949	3.93	N/A
2	100.1448	4.12	N/A	100.3342	4.08	N/A
3	100.0338	4.16	4.00	100.1154	4.14	2.56
4	100.0072	4.18	4.17	100.0426	4.17	3.00
5	99.9996	4.17	3.48	100.0163	4.16	2.78

Table 7.7: Convergence experiments for the GMWB guarantee value at $t = 0$ and $W = A = \omega_0 = 100$ using a fully implicit method and Crank Nicolson method . The penalty method is used. Contract parameters are given in Table 7.1. The column "Central Differencing First" uses central differencing as much as possible for the V_W term in the equation. The column "For/Backward Differencing Only" uses forward or backward differencing for the V_W term in the equation. Itns/step refers to the average number of iterations per timestep for the lines $2-4$ in Algorithm 6.4.1. Ratio is the ratio of successive changes in the solution as the mesh/timesteps are refined. Since the no-arbitrage fee is imposed, the numerical solution should converge to $Value = \omega_0 = 100$.

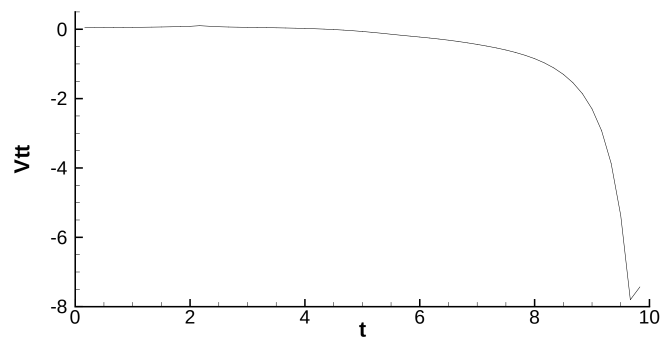


Figure 7.1: V_{tt} versus t for node ($W = 100, A = 100$). $\sigma = 0.3$. Fair insurance fee (i.e. $\eta = 0.031286$) is imposed. Contract parameters are given in Table 7.1.

(forward) time, at the node ($W = 100, A = 100$). At $t = 0$ ($\tau = T$), we can see that $V_{tt} \simeq 0$, which would result in similar time truncation error for both Crank Nicolson and fully implicit timestepping. We also computed the error norms at each level for all the nodes in the W direction. The ratio of successive changes in the error norm then is computed as the mesh/timesteps are refined. Table 7.8 shows the error norm convergence ratio results for different volatilities when central differencing is used as much as possible. The fully implicit method and the Crank Nicolson method appear to have a similar convergence ratio.

Although the first column in Table 7.7 uses central differencing as much as possible, there are large regions in the solution domain where the optimal strategy is to withdraw a finite amount (an infinite rate), as shown in Figure 7.2. In these regions, forward or backward differencing is used in both the W and A directions, which should result in first order errors. However, in the finite withdrawal amount (infinite withdrawal rate) regions, we essentially solve the PDE

$$1 - V_W - V_A = \kappa \; . \tag{7.6}$$

Refinement	l_1 error norm ratio		l_2 error norm ratio		l_∞ error norm ratio	
Level	Fully Imp.	CN	Fully Imp.	CN	Fully Imp.	CN
$\sigma = 0.2, \eta = 0.031286$						
3	3.70	3.62	2.71	2.68	2.00	2.00
4	3.87	3.84	2.78	2.77	2.00	2.00
5	3.95	3.94	2.81	2.81	2.00	2.00
$\sigma = 0.3, \eta = 0.013886$						
3	3.67	3.61	2.69	2.66	2.00	2.00
4	3.84	3.81	2.76	2.75	2.00	2.00
5	3.92	3.48	2.80	2.80	2.00	2.00

Table 7.8: Error norm convergence ratio experiments for the GMWB guarantee value at $t = 0$ and $A = \omega_0 = 100$ using a fully implicit method and Crank Nicolson method. The column "Fully Imp." uses the fully implicit method. The column "CN" uses the Crank Nicolson method. Error norms are computed at each refinement level. The error norm ratio is the ratio of successive changes in the error norm as the mesh/timesteps are refined. The penalty method is used. Contract parameters are given in Table 7.1. Central differencing is used as much as possible for the V_W term in the equation.

Noting that V is linear in A at $W = 0$, and linear in W as $W \to \infty$, then the solution of this PDE in the finite withdrawal region (assuming that this region is connected to $W = 0$ or $W \to \infty$) will be a linear function of (W, A), hence the use of forward or backward differencing is exact.

It is also interesting to see a region labeled *Withdrawal at rate G or no withdrawal.* Recall that in the finite withdrawal region, the solution satisfies

$$V_\tau = \mathcal{L}V + \max_{\gamma \in [0,G]} \left[\gamma(1 - V_W - V_A) \right] . \tag{7.7}$$

The solution in this region appears to converge to a value having $(1 - V_W - V_A) \simeq 0$. This suggests that the optimal control is a finite rate, but not unique, since either a rate of zero or G is optimal. The value function is, however, unique. This is consistent with the results in [11].

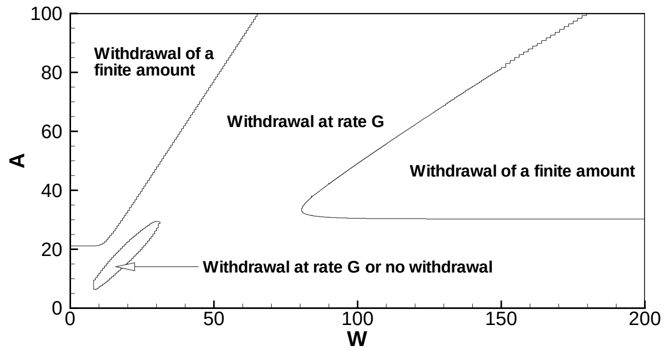


Figure 7.2: The contour plot of optimal withdrawal strategy of the GMWB at $t = \Delta\tau$ in the (W,A)-plane. $\sigma = 0.3$. Fair fee $\eta = .031286$ is imposed. Jump diffusion is removed from the underlying assets. Contract parameters are given in Table 7.1. This plot is similar to the results in [11].

Since it appears (at least for this example) that fully implicit timestepping converges at a similar rate compared to Crank Nicolson, and that convergence can only be proven for fully implicit timestepping, it would appear that fully implicit timestepping is preferable to Crank Nicolson.

7.1.5 Optimal Withdrawal Strategy

A GMWB contract holder is perhaps more interested in the optimal withdrawal strategy. Figure 7.3 shows contour plots of the optimal withdrawal strategy at various times assuming jump diffusion process. The top two plots in Figure 7.3 are generated by both the penalty and direct control methods. It can be observed that these contour plots are very similar. The differences are due to small differences in the computed values, which are amplified by the contouring algorithm.

The other plots in Figure 7.3 are generated by the penalty method. It is interesting to observe that the top left corner infinite withdrawal region is almost time-invariant, except when the contract is close to expiration. The no withdrawal region widens as time moves forward. These results are consistent with the discrete withdrawal computations in [13].

7.1.6 Nodes Around Boundaries

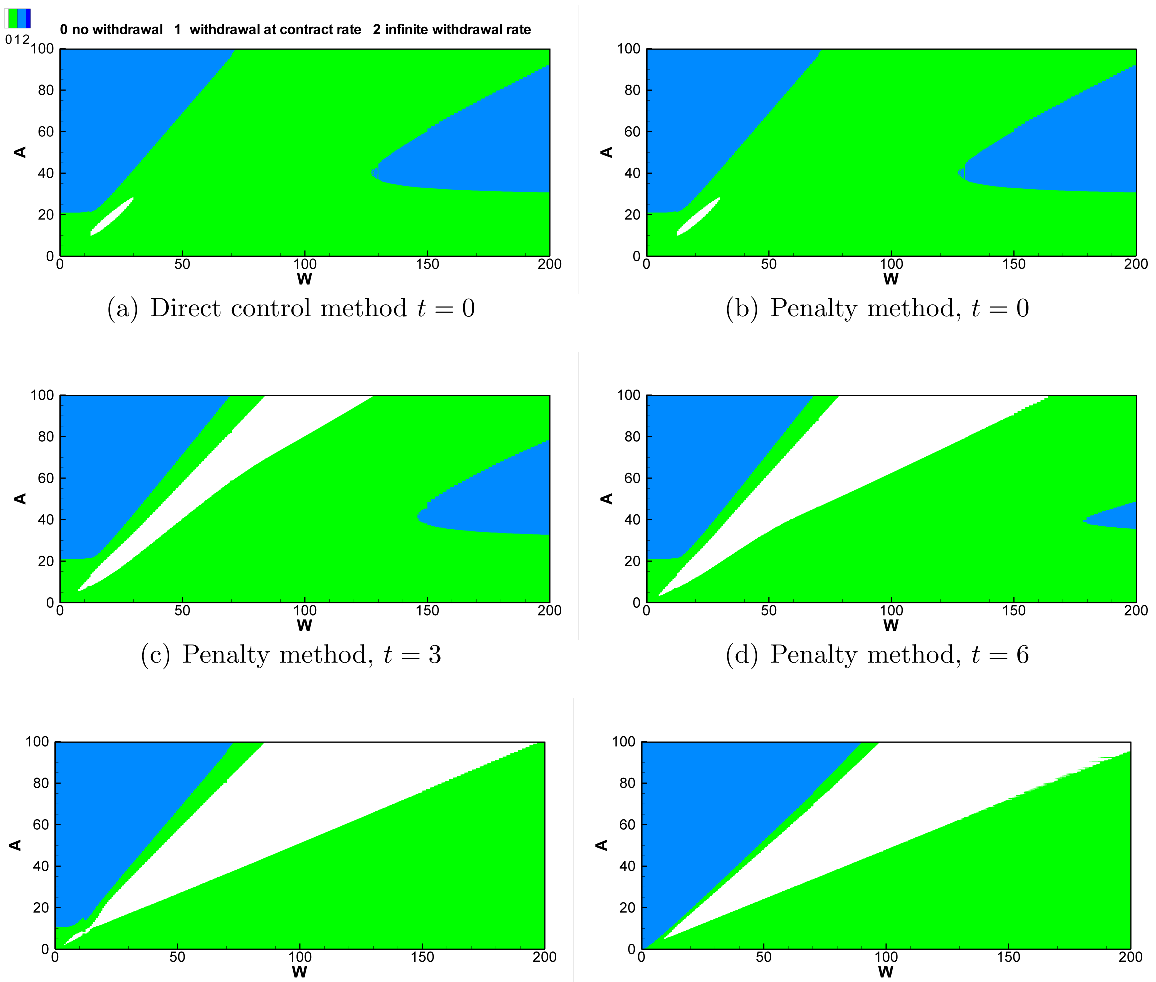
It is also interesting to study the convergence of the singular control formulation for nodes near (or at) the finite withdrawal boundary. Figure 7.4 shows the location of the withdrawal boundaries at $A = 100$ versus t, when no insurance fee ($\eta = 0$) is imposed. Note that the node $(100, 100)$ is very near (or at) the boundary between a finite withdrawal rate and no withdrawal at $t = T$.

Examination of the solution near maturity (which is near the start of the numerical solution since we solve backwards in time) shows that the numerical solution changes between being in the region of withdrawal at rate G to being in a region of zero withdrawal at refinement level 4 and above. This occurs when central differencing is used as much as possible. Table 7.9 gives the convergence results for this case ($\eta = 0$). We have proven that this method is convergent, but clearly convergence can be erratic at some exceptional nodes. Convergence (at this node) is smoother if the V_W term is discretized using a forward or backward differencing only.

7.1.7 Comparison: Singular Control and Impulse Control

As outlined in [57], it is almost always possible to formulate a singular control problem as an impulse control problem, with arbitrarily small error. It is therefore interesting to consider the computational issues for both formulations.

If h is the discretization parameter (as in Assumption 3.4.1), then the computational



(a) Direct control method $t = 0$

(b) Penalty method, $t = 0$

(c) Penalty method, $t = 3$

(d) Penalty method, $t = 6$

(e) Penalty method, $t = 9$

(f) Penalty method, $t = 10 - \Delta\tau$

Figure 7.3: Contour plot of the optimal withdrawal strategy for the GMWB guarantee at different times in the (W, A) -plane. $\sigma = 0.2$. A fair insurance fee of $\eta = 0.032296686$ is imposed. Contract parameters are given in Table 7.1 and jump diffusion parameters are given in Table 7.2. The penalty parameter is set to $1/\varepsilon = 10^4\omega_0/\Delta\tau$ and the scaling factor is set to $\Pi = 1$. The iteration convergence tolerance is set to 10^{-6}.

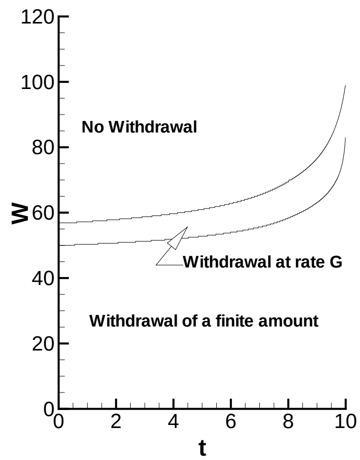


Figure 7.4: The contour plot for the withdrawal boundary versus time t at $A = 100$, $\sigma = 0.3$. No insurance fee (i.e. $\eta = 0$) is imposed. Contract parameters are given in Table 7.1. Maximal use of central differencing on V_W term is applied.

Fully Implicit Method						
Refinement	Central Differencing First			For/Backward Differencing Only		
	Value	Itns/step	Ratio	Value	Itns/step	Ratio
1	116.0354	2.88	N/A	116.2730	2.88	N/A
2	115.9134	2.89	N/A	116.0339	2.91	N/A
3	115.8879	2.97	4.78	115.9477	3.00	2.77
4	115.8845	3.10	7.52	115.9143	3.12	2.59
5	115.8859	3.25	-2.40	115.9008	3.26	2.47
6	115.8876	3.38	0.86	115.8950	3.39	2.33
extrapolated value from [11]				115.8897		

Table 7.9: Convergence experiments for the GMWB guarantee value at $t = 0$ and $W = A = \omega_0 = 100$ by using the fully implicit method. $\sigma = 0.3$. No insurance fee ($\eta = 0$) is imposed. Contract parameters are given in Table 7.1. The column "Central Differencing First" use central differencing as much as possible for the V_W term. The column "For/Backward Differencing Only" uses forward or backward differencing for the V_W term. Itns/step refers to the average number of iterations per timestep for the lines $2 - 4$ in Algorithm 6.4.1. Ratio is the ratio of successive changes in the solution as the refinement is increased.

complexity of the penalty method, singular control formulation is

$$\text{Complexity: Penalty method} \quad = \quad C'h^{-3} \tag{7.8}$$

where C' is the average number of iterations per step. Since it appears that C' is independent of h, then for the block matrix iteration method, the complexity of the singular control method is $O(h^{-3})$. Note that for the full matrix iteration, the number of iterations does appear to increase as $h \to 0$.

In the impulse control formulation, the numerical method described in [11] has a complexity of $O(h^{-4})$. This is due to the linear search required in the local optimization step of the algorithm in [11]. The linear search guarantees location of the global maximum with $O(h)$ error for smooth test functions.

On the basis of complexity, it would appear that the singular control method is a clear winner. However, as noted in [11], it is trivial to handle discrete withdrawal times and complex contract features using an impulse control formulation. These generalizations may be very difficult to handle with a singular control formulation. [57] suggests that an impulse control formulation is preferred in general. In addition, the experimental convergence rate in [11] is smooth as the mesh is refined. This contrasts with the sometimes erratic convergence of the singular control method for nodes near the withdrawal boundaries. As well, the impulse control formulation does not require an estimate of the constant for the penalty parameter (for the penalty method) or the scaling factor (for the direct control method). There also appears to be a limit on the solution accuracy, due to numerical precision problems, with the singular control formulation. However, this limit is probably at a level of accuracy which is far beyond what would be required in practice.

7.2 Summary

- Numerical results show that the block matrix fixed point policy iteration is convergent and efficient. It is superior to the full matrix fixed point policy iteration or the block matrix simple iteration.

- Maximal use of central differencing leads to faster convergence in general, though the nodes around withdrawal boundaries have a smoother convergence ratio if we use forward or backward differencing for the V_W term.

- The singular control formulation is computationally less expensive than the impulse control formulation, but with loss of generality. A limit on the solution accuracy also appears with the singular control formulation, though at a level which is far beyond what would be required in practice.

Chapter 8

Floating Point Considerations

In this chapter we discuss floating point issues in the fixed point policy iteration. The discussion focuses on the block matrix fixed point policy implementation. Section 8.1 presents general results describing how roundoff errors affect convergence of the iteration. Section 8.2 discusses the effect of roundoff error for the penalty method and estimates a bound for the penalty parameter ε. Section 8.3 focuses on the effect of floating point error for the direct control method. Bounds for the scaling parameter Π are estimated. Section 8.4 presents numerical results. Section 8.5 summarizes the main results of this chapter.

8.1 Floating Point Considerations: General Results

During the course of our numerical experiments, we observed that, even if the conditions required by Theorem 6.2.1 were satisfied, the fixed point policy iteration in Algorithm 6.1.2 sometimes failed to converge for certain values of the penalty parameter or the direct control scaling factor. This non-convergence was a result of the oscillatory behavior of the iterates. These oscillations were above the level of the convergence tolerance, hence the scheme did not terminate.

Testing Algorithm 6.4.1 with $\mathbf{B} = \mathbf{0}$ was revealing. In this case, we can see that in exact arithmetic, equations (6.32-6.33) show that the iterates are monotone non-decreasing, i.e. oscillations cannot occur. However, in floating point arithmetic, equation (6.32) is not always true. When there is no jump diffusion, this problem can be ameliorated by forcing the right hand side of equation (6.31) to be always non-negative. However, we cannot use this approach here, when $\mathbf{B} \neq \mathbf{0}$.

Let $fl(x)$ be the floating point representation of a real number x. Define error vector $\mathbf{\Delta e}_\delta^k$ which is generated by the unit roundoff δ.

$$\begin{aligned}\mathbf{\Delta e}_\delta^k &= fl\left(-\mathbf{A}(\chi^k)\mathbf{v}^k + \mathbf{B}(\chi^k)\mathbf{v}^k + \mathbf{c}(\chi^k)\right) \\ &\quad - \left[-\mathbf{A}(\chi^k)\mathbf{v}^k + \mathbf{B}(\chi^k)\mathbf{v}^k + \mathbf{c}(\chi^k)\right] \qquad (8.1)\end{aligned}$$

The floating point error in the fixed point policy iteration is dominated by the computation in equation (8.1), since the computation of these terms involves computing numerical derivatives of $\mathbf{v}^k$. Numerical experiments showed that this source of error far outweighed any other source of floating point error (e.g. the linear equation solve).

Note that in [7, 41], the effect of propagation of errors in policy iteration is discussed. However, the error bound in [7, 41] depends on the *effective discount rate*. In our context, the effective discount rate tends to unity as the mesh is refined, hence the upper bound for the accumulated error in [7, 41] becomes infinite in this limit.

Consequently, we will adopt a somewhat informal, but instructive approach to analyze these errors in the following. Suppose that in exact arithmetic Algorithm 6.1.2 would terminate at step $k+1$. Let $\mathbf{v}^k, \mathbf{v}^{k+1}$ be the iterates computed in exact arithmetic, and let $\mathbf{\Delta v}_\delta^k$ be the floating point error in $\mathbf{v}^{k+1}$ generated by $\mathbf{\Delta e}_\delta^k$ at step k. Then, from equations

(6.31), (8.1),

$$\mathbf{A}(\chi^k)\left[(\mathbf{v}^{k+1}-\mathbf{v}^k)+\Delta\mathbf{v}_\delta^k\right]=\left[-\mathbf{A}(\chi^k)\mathbf{v}^k+\mathbf{B}(\chi^k)\mathbf{v}^k+\mathbf{c}(\chi^k)\right]+\boldsymbol{\Delta}\mathbf{e}_\delta^k\ , \tag{8.2}$$

which gives $\boldsymbol{\Delta}\mathbf{v}_\delta^k=\mathbf{A}^{-1}(\chi^k)\boldsymbol{\Delta}\mathbf{e}_\delta^k$. Clearly, problems will arise if

$$\frac{|(\boldsymbol{\Delta}\mathbf{v}_\delta^k)_\ell|}{\max(scale,|v_\ell^{k+1}|)}>tolerance \tag{8.3}$$

since, even if $\left|\left[(\mathbf{v}^{k+1}-\mathbf{v}^k)\right]_\ell\right|$ is small, the iteration will not converge according to the criteria in Algorithm 6.1.2.

Consequently, we can estimate bounds for parameters that will minimize the effect of floating point errors by requiring that

$$\max_\ell\left[\frac{|(\boldsymbol{\Delta}\mathbf{v}_\delta^k)_\ell|}{\max(|v_\ell^{k+1}|,scale)}\right]=\max_\ell\left[\frac{[\mathbf{A}^{-1}(\chi^k)\boldsymbol{\Delta}\mathbf{e}_\delta^k]_\ell}{\max(|v_\ell^{k+1}|,scale)}\right]<tolerance\ . \tag{8.4}$$

A rigorous bound for condition (8.4) is too pessimistic to be useful. We make the following approximation

$$\max_\ell\left[\frac{[\mathbf{A}^{-1}(\chi^k)\boldsymbol{\Delta}\mathbf{e}_\delta^k]_\ell}{\max(|v_\ell^{k+1}|,scale)}\right]\ \simeq\ \max_\ell\left[\frac{\|\mathbf{A}^{-1}(\chi^k)\|_\infty|\boldsymbol{\Delta}\mathbf{e}_\delta^k|_\ell}{\max(|v_\ell^k|,scale)}\right]\ , \tag{8.5}$$

so that bound (8.4) is estimated as

$$\max_\ell\left[\frac{\|\mathbf{A}^{-1}(\chi^k)\|_\infty|\boldsymbol{\Delta}\mathbf{e}_\delta^k|_\ell}{\max(|v_\ell^k|,scale)}\right]<tolerance\ . \tag{8.6}$$

8.2 Penalty Method Floating Point Considerations

For the penalty method, the floating point error of each iteration is dominated by computation of the following term in equation (8.1)

$$\frac{1}{\varepsilon}\left(1 - \kappa - (\mathcal{D}_W^h v_{i,j}^k + \mathcal{D}_W^h v_{i,j}^k)\right) \ . \tag{8.7}$$

The worst case roundoff error for this term occurs in the area where the grid is fine, where we subtract two nearly equal numbers. This error is then magnified by dividing by the grid spacing and by ε. In Appendix F.3, we obtain the following result (equation (F.20)),

$$\left|\left[\mathbf{\Delta e}_\delta^k\right]_{i,j}\right| \leq \frac{4\delta}{\varepsilon}(\frac{1}{\Delta W_{\min}} + \frac{1}{\Delta A_{\min}}) \max(|v_{i,j}^k|, scale) \ , \tag{8.8}$$

where

$$\Delta A_{\min} = \min_j (A_j - A_{j-1}) \ ; \qquad \Delta W_{\min} = \min_i (W_i - W_{i-1}). \tag{8.9}$$

From Lemma 6.2.3, and Proposition 6.4.1, and setting $\mathbf{B_j} = \mathbf{I}$, we obtain

$$\|\mathbf{A_j}^{-1}(\chi_{*,j}^k)\|_\infty \ \leq \ \max_i \frac{1}{\frac{1}{\Delta\tau} + (r+\lambda) + (\varphi_{i,j}^k G + \frac{\psi_{i,j}^k}{\varepsilon})\frac{1}{\Delta A_j^-}} \leq \Delta\tau \ . \tag{8.10}$$

Consequently, from equations (8.8) and equation (8.10) we obtain

$$\|\mathbf{A_j}^{-1}(\chi_{*,j}^k)\|_\infty |\mathbf{\Delta e}_\delta^k|_{i,j} \ \leq \ \frac{4\delta\Delta\tau}{\varepsilon}(\frac{1}{\Delta W_{\min}} + \frac{1}{\Delta A_{\min}}) \max(|v_{i,j}^k|, scale) \ . \tag{8.11}$$

Now substitute equations (8.11) into equation (8.6) to obtain

$$\frac{4\delta\Delta\tau}{\varepsilon}(\frac{1}{\Delta W_{\min}} + \frac{1}{\Delta A_{\min}}) < tolerance \ . \tag{8.12}$$

In order to ensure that the penalty method is consistent with the original HJB variational inequality, we require that the penalty parameter $\varepsilon = C\Delta\tau$ for any constant $C > 0$ [30]. Intuitively, $1/\varepsilon$ is the maximum withdrawal rate, so that it has dimensions of $dollars/time$.

Define a dimensionless constant C^* such that

$$\frac{1}{\varepsilon} = C^* \frac{\omega_0}{\Delta\tau} \ . \tag{8.13}$$

Substituting equation (8.13) into equation (8.12)

$$C^* < \frac{1}{4}\left(\frac{tolerance}{\delta}\right)\left(\frac{\Delta W_{\min}}{\omega_0}\right)\left(\frac{1}{1 + \frac{\Delta W_{\min}}{\Delta A_{\min}}}\right) \ . \tag{8.14}$$

8.3 Direct Control Method Floating Point Considerations

For the Direct Control approach, the worst case floating point error in equation (8.1) (for Π large) will be generated by the term

$$\Pi\left(1 - \kappa - (\mathcal{D}_W^h v_{i,j}^k + \mathcal{D}_W^h v_{i,j}^k)\right) \ . \tag{8.15}$$

We can estimate the upper bound for Π, using the same approach as in Section 8.2, and we can deduce the bound by setting $\Pi = 1/\varepsilon$ in equation (8.12) to obtain (where we consider $\Pi \to \infty$ when estimating $\|\mathbf{A_j}^{-1}\|_\infty$)

$$\Pi \ < \ \frac{1}{4}(\frac{\text{tolerance}}{\delta})(\frac{\Delta W_{\min}}{\Delta\tau})(\frac{1}{1 + \frac{\Delta A_{\min}}{\Delta W_{\min}}}) \ . \tag{8.16}$$

Conversely, if Π is small, then the worst case floating point error will be generated by the term $\frac{1}{2}\sigma^2 W_i^2 \mathcal{D}_{WW}^h V_{i,j}^{n+1}$ in equation (5.24), since this term involves a numerical second derivative. (See the definition of $\mathcal{L}$ in equation (5.5)). From the result in Appendix F, equation (F.21), we have

$$|(\mathbf{\Delta e}_\delta^k)_{i,j}| \leq 4\delta \frac{\sigma^2 W_i^2}{(\Delta W_{\min})_i^2} \max(scale, |v_{i,j}|) \ , \tag{8.17}$$

where $(\Delta W_{\min}) = \min(W_{i+1} - W_i, W_i - W_{i-1})$. From Lemma 6.2.3 (setting $\mathbf{B_j} = \mathbf{I}$), using Proposition 6.4.2, and considering the case where $\Pi \to 0$, we obtain

$$\|\mathbf{A_j}^{-1}(\chi_{*,j}^k)\|_\infty \ \ \leq \ \ \frac{\Delta A_{\max}}{\Pi} \ , \tag{8.18}$$

where $\Delta A_{\max} = \max_j(A_j - A_{j-1})$. Substituting equations (8.17) and (8.18) into equation (8.6) to obtain

$$\Pi > 4\sigma^2 \Delta A_{\max} \left(\frac{W_{\underline{i}}^2}{(\Delta W_{\min})_{\underline{i}}^2} \right) \left(\frac{\delta}{\text{tolerance}} \right) \ , \tag{8.19}$$

where

$$\max_i \left(\frac{W_{\underline{i}}^2}{(\Delta W_{\min})_i^2} \right) = \frac{W_{\underline{i}}^2}{(\Delta W_{\min})_{\underline{i}}^2} \ . \tag{8.20}$$

Combining equation Lemma 6.4.2 and (8.19) we obtain

$$\Pi \ \ > \ \ \max \left[\lambda \Delta A_{\max}, \ 4\sigma^2 \Delta A_{\max} \left(\frac{W_{\underline{i}}^2}{(\Delta W_{\min})_{\underline{i}}^2} \right) \left(\frac{\delta}{\text{tolerance}} \right) \right] \ . \tag{8.21}$$

Unlike the penalty method, the discretized direct control method does not require the existence of a constant C such that $1/\Pi = C\Delta\tau$ to achieve the consistency. Recall from Remark 4.1.1, that Π has dimensions of *dollars/time*. However, in order to compare the direct control method with the penalty method, we also introduce a dimensionless constant

C^*

$$\Pi = C^* \frac{\omega_0}{\Delta \tau} \ . \tag{8.22}$$

The upper and lower bounds of C^* for the direct control method are then (from equations (8.16), (8.21), (8.22)),

$$\begin{aligned} C^* &> \max\left(\frac{\lambda \Delta A_{\max} \Delta \tau}{\omega_0},\ 4\big(\frac{\sigma^2 \Delta A_{\min} \Delta \tau}{\omega_0}\big)\big(\frac{W_{\underline{i}}^2}{(\Delta W_{\min})_{\underline{i}}^2}\big)\big(\frac{\delta}{\text{tolerance}}\big) \right) , \\ C^* &< \frac{1}{4}\big(\frac{\text{tolerance}}{\delta}\big)\big(\frac{\Delta W_{\min}}{\Delta \omega_0}\big)\big(\frac{1}{1 + \frac{\Delta A_{\min}}{\Delta W_{\min}}}\big) \ . \end{aligned} \tag{8.23}$$

8.4 Numerical Results

In previous sections, we discussed expressing the penalty parameter ε in terms of a dimensionless parameter C^* as in (8.13). We also expressed the scaling factor Π also in terms of C^* as in (8.22) in order to compare the direct control method with the penalty method. Note that the direct control method does not require the existence of a constant C such that $1/\Pi = C\Delta\tau$. Writing $\Pi = C^*\omega_0/\Delta\tau$ is only for the purpose of comparing direct control method with the penalty method.

We refer to the bound on C^* imposed by effect of floating point arithmetic as a Type I bound. The bound on C^* imposed by requiring that $\|\mathbf{A}^{-1}(\chi^k)\mathbf{B}(\chi^{k-1})\|_\infty < 1$, will be refereed to as a Type II bound.

Table 8.1 compares the GMWB value priced by both penalty method and direct control method when $C^* \in [10^{-8}, 10^6]$. The left two columns show the estimated bounds of C^* from equations (8.14) and (8.23). The finest grids are around node $(W = 100, A = 100)$, so we set $W_{\underline{i}} = 100$, in the estimate of the floating point errors in equation (8.23). We take double precision machine epsilon to be $\delta = 1.11 \times 10^{-16}$. The "N/A" entries in the table indicate that the iterative scheme did not satisfy the convergence criteria in Algorithm

6.1.2 after 6000 iterations.

For the entries where the computed values have asterisks, although the convergence criterion in line 3 of Algorithm 6.1.2 was satisfied, we view these results as unreliable. Note that the convergence criterion in Algorithm 6.1.2 is not able to clearly distinguish between very slowly diverging sequences and truly converging sequences. We remind the reader that the Type II bound from Lemma 6.4.2 is a sufficient condition for convergence in exact arithmetic, from Condition 6.2.1.

However, choosing a C^* smaller than the estimated lower bound of C^* from bound (8.23) produces questionable results. It is obvious the values with asterisks deviate somewhat from the other values.

As discussed previously, the direct control method does not require Π to be scaled by $\Delta\tau$, whereas the penalty parameter ε is required to be scaled by $1/\Delta\tau$ for consistency purposes [30]. In order to compare the direct control method with the penalty method, we present Table 8.2 where both ε and Π are not scaled by $\Delta\tau$. The other computational parameters are the same as those used to compute Table 8.1. The values in the column "Bound" are computed according to bounds (8.21) and (8.16).

In the previous numerical examples, the lower bound for the scaling factor Π is dominated by a Type II bound. To see the effect of Type I lower bound in isolation, we remove the jump diffusion from the underlying asset model (e.g. $\lambda = 0$). Consequently, $\|\mathbf{A}^{-1}(\chi^k)\mathbf{B}(\chi^{k-1})\|_\infty < 1$ always holds since $\forall k, \mathbf{B}(\chi^k) = 0$, hence the Type II bound disappears.

Table 8.3 shows the GMWB values priced at refinement level 5 ($\lambda = 0$). Without the presence of Type II bound, we can further decrease the scaling factor by three orders of magnitude. The estimated Type I lower bound for C^* is remarkably close to the experimental result.

	tolerance $= 10^{-6}$		Direct Control		Penalty Method	
Type	Bound	Π or $1/\varepsilon$	Value	Itns/step	Value	Itns/step
I	$0.67 \times 10^{-9}\omega_0/\Delta\tau$	$10^{-9}\omega_0/\Delta\tau$	N/A	N/A		
		$10^{-8}\omega_0/\Delta\tau$	99.999992*	4.03		
		$10^{-7}\omega_0/\Delta\tau$	99.999992*	3.97		
II	$0.33 \times 10^{-6}\omega_0/\Delta\tau$	$10^{-6}\omega_0/\Delta\tau$	100.00003	3.94		
		$10^{-5}\omega_0/\Delta\tau$	100.00003	3.89		
		$10^{-4}\omega_0/\Delta\tau$	100.00003	3.84		
		$10^{-3}\omega_0/\Delta\tau$	100.00003	3.31		
		$10^{-2}\omega_0/\Delta\tau$	100.00003	3.80		
		$10^{-1}\omega_0/\Delta\tau$	100.00003	3.82	99.969172	3.83
		$10^{0}\omega_0/\Delta\tau$	100.00003	3.83	99.996899	3.85
		$10^{1}\omega_0/\Delta\tau$	100.00003	3.87	99.999715	3.87
		$10^{2}\omega_0/\Delta\tau$	100.00003	3.88	99.999998	3.88
		$10^{3}\omega_0/\Delta\tau$	100.00003	3.88	100.00002	3.88
		$10^{4}\omega_0/\Delta\tau$	100.00003	3.88	100.00003	3.88
		$10^{5}\omega_0/\Delta\tau$	100.00003	3.88	100.00003	3.88
I	$0.35 \times 10^{6}\omega_0/\Delta\tau$	$10^{6}\omega_0/\Delta\tau$	100.00003	3.88	100.00002	3.88
		$10^{7}\omega_0/\Delta\tau$	N/A	N/A	N/A	N/A

Table 8.1: The effect of the scaling factor Π and penalty parameter $1/\varepsilon$ in terms of C^* on pricing the GMWB guarantee at refinement level 5. $\sigma = 0.3, W = A = 100$ and $t = 0$. Fair insurance fee (i.e. $\eta = 0.045452043$) is imposed. Fully implicit method is used. Tolerance for iteration is set to 10^{-6}. Contract parameters are given in Table 7.1. Jump diffusion parameters are given in Table 7.2. Itns/step refers to the average number of iterations per timestep for the lines $2-4$ in Algorithm 6.4.1. Type I bounds refer to bounds based on floating point considerations. Type II bounds refer to sufficient conditions for convergence in exact arithmetic, from Condition 6.2.1.

tolerance $= 10^{-6}$			Direct Control		Penalty Method	
Type	Bound	$1/\varepsilon$ or Π	Value	Itns/step	Value	Itns/step
I	0.13×10^{-6}	10^{-5}	N/A	N/A		
		10^{-5}	N/A	N/A		
		10^{-4}	99.999908*	4.13		
		10^{-3}	99.999930*	3.98		
II	0.63×10^{-2}	10^{-2}	100.00003	3.94		
		10^{-1}	100.00003	3.90		
		10^{0}	100.00003	3.87		
		10^{1}	100.00003	3.02	96.209487	3.00
		10^{2}	100.00003	3.78	99.472632	3.83
		10^{3}	100.00003	3.82	99.941407	3.82
		10^{4}	100.00003	3.82	99.994038	3.83
		10^{5}	100.00003	3.86	99.999426	3.86
		10^{6}	100.00003	3.88	99.999969	3.88
		10^{7}	100.00003	3.88	100.00003	3.88
		10^{8}	100.00003	3.88	100.00003	3.88
		10^{9}	100.00003	3.88	100.00003	3.88
I	0.68×10^{10}	10^{10}	100.00003	3.88	100.00003	3.88
		10^{11}	N/A	N/A	N/A	N/A

Table 8.2: The effect of the scaling factor $1/\varepsilon$ and penalty parameter ε on pricing the GMWB guarantee at refinement level 5. $\sigma = 0.3, W = A = 100$ and $t = 0$. Fair insurance fee (i.e. $\eta = 0.045452043$) is imposed. Fully implicit method is used. Tolerance for iteration is set to 10^{-6}. Contract parameters are given in Table 7.1. Jump diffusion parameters are given in Table 7.2. Itns/step refers to the average number of iterations per timestep for the lines $2 - 4$ in Algorithm 6.4.1. Type I bounds refer to bounds based on floating point considerations. Type II bounds refer to sufficient conditions for convergence in exact arithmetic.

Remark 8.4.1 (Range of Values). *These examples clearly show that the range of useful values of the scaling parameter for the direct control method is much larger than the range of useful values for the penalty parameter in the penalty method.*

We also carried out an experiment to determine the order of magnitude of $[C^*_{\min}, C^*_{\max}]$, as a function of convergence tolerance, such that $\forall C^* \in [C^*_{\min}, C^*_{\max}]$ the computed GMWB values agree to $(n+1)^{th}$ digit, where $tolerance = 10^{-n}$. Table 8.4 compares the computed order of magnitude of $C^*_{\max}$ and $C^*_{\min}$ with the estimated C^* upper and lower bounds from equation (8.23).

A similar experiment was also conducted to seek the order of magnitude of the range $[\Pi_{\min}, \Pi_{\max}]$, as a function of iteration tolerance, such that $\forall\, \Pi \in [\Pi_{\min}, \Pi_{\max}]$ the computed GMWB values agree to $(n+1)^{th}$ digit, where $tolerance = 10^{-n}$. Table 8.5 compares the computed order of magnitude of $\Pi_{\max}$ and $\Pi_{\min}$ with the estimated Π upper and lower bounds from equations (8.16) and (8.21).

8.5 Summary

The main results of this chapter are as follows.

- Condition 6.2.1 is a sufficient condition for fixed point policy iteration to converge under exact arithmetic. In practice, a floating point system is used and additional conditions are required so that convergence can be expected in the presence of inexact arithmetic.
- By Condition 6.2.1 together with an estimation of local optimization roundoff error, we estimate bounds for the penalty parameter in the penalty method and the scaling

	$tolerance = 10^{-6}$		Direct Control		Penalty Method	
Type	Bound	Π or $1/\varepsilon$	Value	Itns/step	Value	Itns/step
		$10^{-11}\omega_0/\Delta\tau$	N/A	N/A		
		$10^{-10}\omega_0/\Delta\tau$	115.88596	2.69		
I	$0.67 \times 10^{-9}\omega_0/\Delta\tau$	$10^{-9}\omega_0/\Delta\tau$	115.88596	2.69		
		$10^{-8}\omega_0/\Delta\tau$	115.88596	2.69		
		$10^{-7}\omega_0/\Delta\tau$	115.88596	2.69		
II	$0.33 \times 10^{-6}\omega_0/\Delta\tau$	$10^{-6}\omega_0/\Delta\tau$	115.88596	2.69		
		$10^{-5}\omega_0/\Delta\tau$	115.88596	2.69		
		$10^{-4}\omega_0/\Delta\tau$	115.88596	2.69		
		$10^{-3}\omega_0/\Delta\tau$	115.88596	2.68		
		$10^{-2}\omega_0/\Delta\tau$	115.88596	2.83		
		$10^{-1}\omega_0/\Delta\tau$	115.88596	2.84	115.85508	2.85
		$10^{0}\omega_0/\Delta\tau$	115.88596	2.85	115.88281	2.84
		$10^{1}\omega_0/\Delta\tau$	115.88596	2.85	115.88565	2.85
		$10^{2}\omega_0/\Delta\tau$	115.88596	2.85	115.88593	2.85
		$10^{3}\omega_0/\Delta\tau$	115.88596	2.85	115.88596	2.85
		$10^{4}\omega_0/\Delta\tau$	115.88596	2.85	115.88596	2.85
		$10^{5}\omega_0/\Delta\tau$	115.88596	2.85	115.88596	2.85
I	$0.35 \times 10^{6}\omega_0/\Delta\tau$	$10^{6}\omega_0/\Delta\tau$	115.88596	2.87	115.88596	2.86
		$10^{7}\omega_0/\Delta\tau$	N/A	N/A	N/A	N/A

Table 8.3: The effect of Type I upper and lower bounds on the scale factor Π and penalty parameter $1/\varepsilon$ on pricing the GMWB guarantee at refinement level 5. No jump diffusion presented. $\sigma = 0.3, W = A = 100$ and $t = 0$. No insurance fee (i.e. $\eta = 0$) is imposed. Fully implicit method is used. Contract parameters are given in Table 7.1. Itns/step refers to the average number of iterations per timestep for the lines $2 - 4$ in Algorithm 6.4.1. Type I bounds refer to bounds based on floating point considerations. Type II bounds refer to sufficient conditions for convergence in exact arithmetic. In this case the Type II bound is not required since the jump term is absent.

	Tolerance					
	10^{-6}		10^{-8}		10^{-10}	
Level	C^* *upper bound*	$C^*_{\max}$	C^* *upper bound*	$C^*_{\max}$	C^* *upper bound*	$C^*_{\max}$
0	0.11×10^8	10^8	0.11×10^6	10^7	0.11×10^4	10^6
1	0.56×10^7	10^7	0.56×10^5	10^6	0.56×10^3	10^5
2	0.28×10^7	10^7	0.28×10^5	10^5	0.28×10^3	10^5
3	0.14×10^7	10^7	0.14×10^5	10^5	0.14×10^3	10^4
4	0.70×10^6	10^6	0.70×10^4	10^4	0.70×10^2	10^3
5	0.35×10^6	10^6	0.35×10^4	10^4	0.35×10^2	10^2
Level	C^* *lower bound*	$C^*_{\min}$	C^* *lower bound*	$C^*_{\min}$	C^* *lower bound*	$C^*_{\min}$
0	0.33×10^{-3}	10^{-4}	0.33×10^{-3}	10^{-4}	0.33×10^{-3}	10^{-4}
1	0.83×10^{-4}	10^{-4}	0.83×10^{-4}	10^{-5}	0.83×10^{-4}	10^{-5}
2	0.21×10^{-4}	10^{-5}	0.21×10^{-4}	10^{-6}	0.21×10^{-4}	10^{-5}
3	0.52×10^{-5}	10^{-5}	0.52×10^{-5}	10^{-7}	0.67×10^{-5}	10^{-5}
4	0.13×10^{-5}	10^{-5}	0.13×10^{-5}	10^{-6}	0.67×10^{-5}	10^{-5}
5	0.33×10^{-6}	10^{-6}	0.33×10^{-6}	10^{-7}	0.67×10^{-5}	10^{-5}

Table 8.4: Experimental C^* upper ($C^*_{\max}$)and lower ($C^*_{\min}$) bounds as a function of iteration convergence tolerance. The theoretical bounds C^* *upper bound* and C^* *lower bound* are also shown. Both penalty and direct control method with block matrix implementation as in Algorithm 6.4.1 and produce the same results of $C^*_{\max}$. Direct control method is used for computing $C^*_{\min}$. Contract parameters are in Table 7.1. Jump diffusion parameters are in Table 7.2. $\sigma = 0.3, \eta = 0.045452043$. Finest grids are around node $(W, A) = (100, 100)$, which are used to compute the bounds.

	Tolerance					
	10^{-6}		10^{-8}		10^{-10}	
Level	Π *upper bound*	$\Pi_{\max}$	Π *upper bound*	$\Pi_{\max}$	Π *upper bound*	$\Pi_{\max}$
0	0.68×10^{10}	10^{11}	0.68×10^{8}	10^{9}	0.68×10^{6}	10^{9}
1	0.68×10^{10}	10^{11}	0.68×10^{8}	10^{9}	0.68×10^{6}	10^{8}
2	0.68×10^{10}	10^{10}	0.68×10^{8}	10^{9}	0.68×10^{6}	10^{8}
3	0.68×10^{10}	10^{10}	0.68×10^{8}	10^{9}	0.68×10^{6}	10^{7}
4	0.68×10^{10}	10^{10}	0.68×10^{8}	10^{8}	0.68×10^{6}	10^{7}
5	0.68×10^{10}	10^{10}	0.68×10^{8}	10^{8}	0.68×10^{6}	10^{6}
Level	Π *lower bound*	$\Pi_{\min}$	Π *lower bound*	$\Pi_{\min}$	Π *lower bound*	$\Pi_{\min}$
0	0.20×10^{0}	10^{-2}	0.20×10^{0}	10^{-2}	0.20×10^{0}	10^{-2}
1	0.10×10^{0}	10^{-1}	0.10×10^{0}	10^{-2}	0.10×10^{0}	10^{-2}
2	0.50×10^{-1}	10^{-1}	0.50×10^{-1}	10^{-2}	0.50×10^{-1}	10^{-1}
3	0.25×10^{-1}	10^{-2}	0.25×10^{-1}	10^{-3}	0.32×10^{-1}	10^{-1}
4	0.13×10^{-1}	10^{-1}	0.13×10^{-1}	10^{-2}	0.64×10^{-1}	10^{-1}
5	0.63×10^{-2}	10^{-2}	0.63×10^{-2}	10^{-2}	0.13×10^{0}	10^{0}

Table 8.5: Experimental upper bounds ($\Pi_{\max}$) for Π and $1/\varepsilon$ and lower bound for Π ($\Pi_{\min}$), as a function of iteration convergence tolerance. The theoretical bounds Π *upper bound* and Π *lower bound* also shown. Both penalty and direct control method with block matrix implementation as in Algorithm 6.4.1 are used in upper bound experiment and produce the same results for $\Pi_{\max}$. Direct control method is used to compute $\Pi_{\min}$. Contract parameters are in Table 7.1. Jump diffusion parameters are in Table 7.2. $\sigma = 0.3, \eta = 0.045452043$. Finest grids are around node $(W, A) = (100, 100)$. These nodes are used to compute the theoretical bounds.

parameter in the direct control method so that we expect the fixed point policy iteration will converge in the presence of roundoff error.

- Numerical results show that the estimated bounds are of the correct order of magnitude. The useful numerical range of the scaling parameter for the direct control method is much larger than the penalty parameter for the penalty method.

- A useful rule of thumb is to choose the penalty parameter or the direct control scaling parameter two orders of magnitude less than the upper bound estimate.

Chapter 9

Conclusions and Future Work

9.1 Conclusions

This thesis studies numerical methods for solving the HJB PIDE/VI resulting from pricing a GMWB as a singular control problem [40] with the additional assumption that the underlying asset follows a Poisson jump diffusion process. We extend the penalty method [18] and direct control method [9] to solve the resulting HJB PIDE/VI. Provided the original problem satisfies a strong comparison property, we prove that the penalty method discretization converges to the unique viscosity solution of the HJB VI for the case of standard Geometric Brownian Motion. We discuss the proof of the convergence of the direct control method discretization to the unique viscosity solution of the HJB VI by giving detailed proof of the stability for the case of standard Geometric Brownian Motion. We also briefly sketch the proof of the convergence of both the penalty method and the direct control method discretizations to the unique viscosity solution of the HJB VI for the jump diffusion case. Maximal use of central differencing [55] results in noticeably faster convergence (as the grid/timesteps are refined) compared to forward or backward differencing only discretization.

An efficient fixed point policy iteration scheme is developed to solve a class of discretized controlled HJB PDEs in finance including but not limited to the PDE resulting from pricing a GMWB as a singular problem. This method is particularly useful if the risky asset (in a financial application) follows a jump diffusion or regime switching process. Sufficient conditions are derived to ensure the convergence of the fixed point policy iteration. In the penalty method case, these conditions are typically satisfied if a monotone discretization method is used, which is normally required in order to ensure convergence to the viscosity solution. In case of the direct control method, we applied a scaling factor to the discrete equation. The convergence of the fixed point policy iteration in this case can only be guaranteed if the scaling parameter satisfies certain conditions. However it is always possible to select a scaling parameter which satisfies this condition.

The singular control formulation of a GMWB has a special structure that can significantly improve the efficiency of solving the resulting nonlinear system. A block matrix fixed point policy iteration scheme is developed and the conditions required for convergence are determined. Numerical results show that this method is an order of magnitude better in terms of number of iterations compared to a full matrix formulation.

Both the penalty method and the direct control method require specification of a parameter. This parameter affects both convergence and accuracy. We estimate bounds for these parameters for both methods, so that convergence in floating point arithmetic can be expected. To the best of our knowledge, such analysis has not been carried out previously. Numerical experiments indicate that these estimates are reasonably accurate.

Our experimental results show that the singular control formulation has some limitations in determining the withdrawal boundaries to high accuracy. For nodes near the withdrawal boundaries, convergence is somewhat erratic. However, the singular control formulation is easy to implement and convergence is fast to a level of accuracy probably far beyond what would be required in practice. This method has a lower complexity than the impulse control approach in [11], though at the expense of some loss of generality.

9.2 Recommendations

Based on our analysis and numerical experiment results, we make the following recommendations from a practical perspective regarding pricing a GMWB.

- The singular control formulation for pricing a GMWB is easy to implement and converges rapidly. In a situation where speed over-weighs unnecessary accuracy, a singular control formulation appears to be a good methodology. However, complex contractual features may be difficult to implement with a singular control formulation.
- The block matrix fixed point policy iteration is a recommended efficient implementation to solve the resulting nonlinear system for both the penalty method and the direct control method.
- It is safe to choose the penalty parameter and the direct control scaling factor two orders of magnitude away from the estimated bounds (based on floating point error analysis).
- It would appear that the order of magnitude useful range of the scaling parameter for the direct control method is much larger than the useful range for the penalty parameter in the penalty method. The accuracy and convergence rate for both methods are similar for parameters within the useful range. Consequently, it would appear that the direct control method is superior to the penalty method in this regard.

9.3 Future Work

A few interesting future research directions appear while we are studying the GMWB pricing problem and they are as follows.

- A singular control formulation is an often used methodology for modeling problems in finance. The fixed point policy iteration scheme described in this thesis is based on a very general form of discretized controlled HJB equation, which may be applied to other singular control problems in finance beyond the GMWB pricing problem. It would be worthwhile to explore how the fixed point policy iteration performs with a wider range of singular control problems in finance.

- Both Poisson jump diffusion and regime switching are considered better models for the underlying assets that are more consistent with market data. Studying the GMWB problem as a singular control problem under regime switching is also an interesting research direction.

- The scaling factor in the direct control method appears to be a newly observed parameter and merits further study.

Appendix A

Hedging Argument for (3.1)

In this Appendix, we give an informal hedging argument for deriving equation (3.1). Consider the following scenario. The underlying asset W (a mutual fund) in the investor's account follows the process

$$dW = (\mu - \eta)W dt + W\sigma dZ \ , \tag{A.1}$$

where μ is the drift rate, η is the fee for the guarantee, and dZ is the increment of a Wiener process.

We assume that the mutual fund tracks an index $\hat{W}$ which follows the process

$$d\hat{W} = \mu \hat{W} dt + \hat{W}\sigma dZ \ . \tag{A.2}$$

We assume that it is not possible to short the mutual fund, so that the obvious arbitrage opportunity cannot be exploited. (This is typically a fiduciary requirement.) We further assume that it is possible to track the index $\hat{W}$ without basis risk.

Now, consider the writer of the GMWB contract, with no-arbitrage value $V(W, A, t)$.

The writer sets up the hedging portfolio

$$\Pi(W, \hat{W}, A, t) = -V(W, A, t) + x\hat{W} \ , \tag{A.3}$$

where x is the number of units of the index $\hat{W}$.

Over the time interval $t \to t + dt$, assuming that Ito's Lemma can be used, we obtain

$$d\Pi = -\left[\left(V_t + (\mu - \eta)WV_W + \frac{1}{2}\sigma^2 W^2 V_{WW} + f(\gamma) - \gamma V_W - \gamma V_A\right) dt + \sigma W V_W dZ\right] \\ + x[\mu\hat{W}dt + \sigma\hat{W}dZ] \ , \tag{A.4}$$

where γ is the (finite) rate of withdrawal by the contract holder.

Choose

$$x = \frac{W}{\hat{W}} V_W, \tag{A.5}$$

so that equation (A.4) becomes

$$d\Pi = -\left[\left(V_t - \eta W V_W + \frac{1}{2}\sigma^2 W^2 V_{WW} + f(\gamma) - \gamma V_W - \gamma V_A\right) dt\right] . \tag{A.6}$$

The worst case for the hedger will be when the contract holder chooses an action to minimize the value of the hedging portfolio (this of course corresponds to the contract holder maximizing her no-arbitrage long position), so that

$$d\Pi = \min_{\gamma}\left[-\left(V_t - \eta W V_W + \frac{1}{2}\sigma^2 W^2 V_{WW} + f(\gamma) - \gamma V_W - \gamma V_A\right) dt\right] . \tag{A.7}$$

Let r be the risk free rate, and so setting $d\Pi = r\Pi \ dt$ (since the portfolio is now riskless) gives

$$
\begin{aligned}
r\left(-V+V_W W\right) &= -\max_{\gamma}\left[\left(-V_\tau-\eta W V_W+\frac{1}{2}\sigma^2 W^2 V_{WW}+f(\gamma)-\gamma V_W-\gamma V_A\right)\right] \\
&= V_\tau+\eta W V_W-\frac{1}{2}\sigma^2 W^2 V_{WW}-\max_{\gamma}\left[f(\gamma)-\gamma V_W-\gamma V_A\right], \qquad \text{(A.8)}
\end{aligned}
$$

which is equation (3.1).

Another way to verify this equation is the following. Imagine that the hedger replicates the cash flows associated with the total GMWB contract. In this case, the underlying mutual fund can be regarded as a purely virtual instrument, following process (A.1). The actual hedging instrument on the other hand follows process (A.2). Having eliminated the random term by delta hedging, the hedger then assumes the worst case which occurs when the contact holder maximizes (deterministically) the no-arbitrage value of the contract. In this case, $V = U + W$, where V is the value of the entire contract, and U is the value of the guarantee. We can obtain an equation for the guarantee portion U by substituting $V = U + W$ into equation (A.8).

[13] uses a similar argument to value the guarantee portion of the GMWB using the impulse control formulation.

Of course, the above arguments assume that the rate of withdrawal is finite, and that the solution is sufficiently smooth so that Ito's Lemma can be applied. These assumptions are not in general valid (i.e. we take the limit as the maximum withdrawal rate becomes infinite), and a much more careful analysis is required to derive the singular control problem in rigorous fashion. Delta hedging strategies for GMWB contacts are commonly used in the insurance industry [6, 28], although usually based on the impulse control formulation.

Appendix B

Finite Difference Approximation

B.1 First and Second Derivatives Approximation

In this appendix, we use a standard finite difference method to approximate the first and second partial derivatives in the PDE. The discretized differential operators $\mathcal{D}_A^h$, $\mathcal{D}_W^h$ and $\mathcal{D}_{WW}^h$ are given by

$$\mathcal{D}_A^h V_{i,j}^n = \frac{V_{i,j}^n - V_{i,j-1}^n}{\Delta A_j^-}, \quad \text{backward differencing,} \tag{B.1}$$

$$\mathcal{D}_W^h V_{i,j}^n = \begin{cases} \frac{V_{i,j}^n - V_{i-1,j}^n}{\Delta W_i^-} & \text{backward differencing,} \\ \frac{V_{i+1,j}^n - V_{i,j}}{\Delta W_i^+} & \text{forward differencing,} \\ \frac{V_{i+1,j}^n - V_{i-1,j}^n}{\Delta W_i^\pm} & \text{central differencing,} \end{cases} \tag{B.2}$$

$$\begin{aligned} \mathcal{D}_{WW}^h V_{i,j}^n &= \frac{\frac{V_{i-1,j}^n - V_{i,j}^n}{\Delta W_i^-} + \frac{V_{i+1,j}^n - V_{i,j}^n}{\Delta W_i^+}}{\frac{\Delta W_i^\pm}{2}} \\ &= \frac{\mathcal{D}_W^h V_{i+1,j}^n - \mathcal{D}_W^h V_{i,j}^n}{\frac{\Delta W_i^\pm}{2}} \quad (\mathcal{D}_W^h \text{ is backward differenced).} \end{aligned} \tag{B.3}$$

where

$$\Delta A_j^- = A_j - A_{j-1}, \quad \Delta W_i^- = W_i - W_{i-1}, \quad \Delta W_i^+ = W_{i+1} - W_i, \quad \text{and} \quad \Delta W_i^{\pm} = W_{i+1} - W_{i-1}.$$

B.2 Discrete Equation Coefficients

Let $\{\varphi, \psi\}$ denote the local control parameter value for node (W_i, A_j, τ^n).

$$\begin{aligned}
\mathcal{A}^h_{\varphi,\psi} V^n_{i,j} &= a_{i,j}(\varphi,\psi)\mathcal{D}^h_{WW} V^n_{i,j} + b_{i,j}(\varphi,\psi)\mathcal{D}^h_W V^n_{i,j} - c_{i,j}(\varphi,\psi) V^n_{i,j} \\
&= \alpha_{i,j}(\varphi,\psi) V^n_{i-1,j} - \left[\alpha_{i,j}(\varphi,\psi) + \beta_{i,j}(\varphi,\psi) + c_{i,j}(\varphi,\psi)\right] V^n_{i,j} + \beta_{i,j}(\varphi,\psi) V^n_{i+1,j}.
\end{aligned}$$

If central differencing is used for the $\mathcal{D}^h_W V^n_{i,j}$ term, then

$$\begin{aligned}
\alpha_{i,j,cent} &= \frac{2a_{i,j}(\varphi,\psi)}{\Delta W_i^{\pm} \Delta W_i^-} - \frac{b_{i,j}(\varphi,\psi)}{\Delta W_i^{\pm}}, \\
\beta_{i,j,cent} &= \frac{2a_{i,j}(\varphi,\psi)}{\Delta W_i^{\pm} \Delta W_i^+} + \frac{b_{i,j}(\varphi,\psi)}{\Delta W_i^{\pm}}.
\end{aligned} \tag{B.4}$$

When a forward/backward differencing is used for the $\mathcal{D}^h_W V^n_{i,j}$ term, we obtain

$$\begin{aligned}
\alpha_{i,j,for/back} &= \frac{2a_{i,j}(\varphi,\psi)}{\Delta W_i^{\pm} \Delta W_i^-} + \max\left[0, \frac{-b_{i,j}(\varphi,\psi)}{\Delta W_i^-}\right], \\
\beta^n_{i,j,for/back} &= \frac{2a_{i,j}(\varphi,\psi)}{\Delta W_i^{\pm} \Delta W_i^+} + \max\left[0, \frac{b_{i,j}(\varphi,\psi)}{\Delta W_i^+}\right].
\end{aligned} \tag{B.5}$$

where

$$\Delta W_i^- = W_i - W_{i-1}, \quad \Delta W_i^+ = W_{i+1} - W_i, \quad \text{and} \quad \Delta W_i^{\pm} = W_{i+1} - W_{i-1}.$$

Appendix C

Maximal Use of Central Differencing: Direct Control Method

Similar to the penalty method discretization, the discretized $\mathcal{D}_W^h V_{i,j}^n$ term in $\mathcal{B}_\varphi^h V_{i,j}^n$ in equation (4.8) can be obtained by applying central, forward, or backward differencing to the $\mathcal{D}_W V^\varepsilon$ term. We again write the $\mathcal{B}_\varphi^h$ operator in the following form

$$\begin{aligned} \mathcal{B}_\varphi^h V_{i,j}^n &= \alpha_{i,j}(\varphi) V_{i-1,j}^n - \big(\alpha_{i,j}(\varphi) + \beta_{i,j}(\varphi) + c_{i,j}(\varphi)\big) V_{i,j}^n + \beta_{i,j}(\varphi) V_{i+1,j}^n, \\ & \quad i = 2, 3, \dots, i_{\max} - 1,\ j = 1, 2, \dots, j_{\max},\ n = 1, 2, \dots, N-1 \ . \end{aligned} \tag{C.1}$$

The $\alpha_{i,j}(\varphi)$ and $\beta_{i,j}(\varphi)$ in (C.1) are determined by the differencing method used in W direction, $\alpha_{i,j} \in \{\alpha_{i,j,cent}, \alpha_{i,j,for/back}\}$, $\beta_{i,j} \in \{\beta_{i,j,cent}, \beta_{i,j,for/back}\}$, which are defined in Appendix B.2. Although the $\alpha_{i,j}$ and $\beta_{i,j}$ for the direct control method does not have a dependency on the control variable ψ, the method in Appendix B.2 is written in a general form so that it can handle this case as well. We use the coefficients as in (4.9) to compute $\alpha_{i,j}$ and $\beta_{i,j}$ for the direct control method. The positive coefficient condition (see [44]) requires

$$\alpha_{i,j} \geq 0 \ \ ; \ \ \beta_{i,j} \geq 0 \ . \tag{C.2}$$

Because $c_{i,j} \geq 0$ always holds, condition (3.24) is a sufficient condition to ensure a positive coefficient discretization scheme. To use central differencing on the $\mathcal{D}_W V$ term and maintain a positive coefficient condition at the same time, we require $\psi_{i,j}^{n+1} = 0$ and

$$\frac{1}{W_i - W_{i-1}} \geq \frac{(1-\psi_{i,j}^{n+1})(r_i - \eta) - \frac{(1-\psi_{i,j}^{n+1})\varphi G + \psi \Pi}{W_i}}{(1-\psi_{i,j}^{n+1})\sigma^2 W_i} \ ; \tag{C.3}$$

$$\frac{1}{W_{i+1} - W_i} \geq -\frac{(1-\psi_{i,j}^{n+1})(r_i - \eta) - \frac{(1-\psi_{i,j}^{n+1})\varphi G + \psi \Pi}{W_i}}{(1-\psi_{i,j}^{n+1})\sigma^2 W_i} \ . \tag{C.4}$$

Appendix D

M Matrix Property of $\mathbf{Z}^n$ in the Direct Control Method

Since $\mathbf{Z}^n$ is a block lower triangular matrix, its inverse $(\mathbf{Z}^n)^{-1}$ is also a block lower triangular matrix if $(\mathbf{Z}^n)^{-1}$ exists. When $\mathbf{Z}^n$ is a 2×2 block matrix, it is easy to show that $\mathbf{Z}^n$ is an M matrix. We use a 3×3 block matrix as a non-trivial example to show that $(\mathbf{Z}^n)^{-1}$ is nonsingular and $(\mathbf{Z}^n)^{-1} \geq 0$.

Let

$$\mathbf{Z}^n = \begin{pmatrix} \mathbf{D_1} & \mathbf{0} & \mathbf{0} \\ -\mathbf{L_2^n} & \mathbf{D_2^n} & \mathbf{0} \\ \mathbf{0} & -\mathbf{L_3^n} & \mathbf{D_3^n} \end{pmatrix} \quad ; \quad \mathbf{X} = \begin{pmatrix} \mathbf{X_{11}} & \mathbf{0} & \mathbf{0} \\ \mathbf{X_{21}} & \mathbf{X_{22}} & \mathbf{0} \\ \mathbf{X_{31}} & \mathbf{X_{32}} & \mathbf{X_{33}} \end{pmatrix} , \tag{D.1}$$

where $\mathbf{X_{ij}}$ $(1 \leq j \leq i \leq 1, 2, 3)$ is a block matrix that has the same dimension as $\mathbf{D_1}$ and $\mathbf{D_i^n}$ $(i = 2, 3)$. We remind the reader that $\mathbf{D_1}$ and $\mathbf{D_i^n}$ $(i = 2, 3)$ are block M matrices and $\mathbf{L_i^n}$ $(i = 2, 3)$ is non-negative. If the following set of equations in (D.2) have a unique solution $\mathbf{X}$, then $(\mathbf{Z}^n)^{-1} = \mathbf{X}$. In (D.2), $\mathbf{I_i}$ $(i = 1, 2, 3)$ is the identity matrix with the same dimension as $\mathbf{D_1}$ and $\mathbf{D_i^n}$ $(i, = 2, 3)$.

$$\begin{pmatrix} \mathbf{D_1} & \mathbf{0} & \mathbf{0} \\ -\mathbf{L_2^n} & \mathbf{D_2^n} & \mathbf{0} \\ \mathbf{0} & -\mathbf{L_3^n} & \mathbf{D_3^n} \end{pmatrix} \begin{pmatrix} \mathbf{X_{11}} & \mathbf{0} & \mathbf{0} \\ \mathbf{X_{21}} & \mathbf{X_{22}} & \mathbf{0} \\ \mathbf{X_{31}} & \mathbf{X_{32}} & \mathbf{X_{33}} \end{pmatrix} = \begin{pmatrix} \mathbf{I_1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I_2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I_3} \end{pmatrix} . \tag{D.2}$$

The system of equations in (D.2) are equivalent to three sets of matrix equations as follows.

$$\begin{pmatrix} \mathbf{D_1} & \mathbf{0} & \mathbf{0} \\ -\mathbf{L_2^n} & \mathbf{D_2^n} & \mathbf{0} \\ \mathbf{0} & -\mathbf{L_3^n} & \mathbf{D_3^n} \end{pmatrix} \begin{pmatrix} \mathbf{X_{11}} \\ \mathbf{X_{21}} \\ \mathbf{X_{31}} \end{pmatrix} = \begin{pmatrix} \mathbf{I_1} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix} , \tag{D.3}$$

$$\begin{pmatrix} \mathbf{D_1} & \mathbf{0} & \mathbf{0} \\ -\mathbf{L_2^n} & \mathbf{D_2^n} & \mathbf{0} \\ \mathbf{0} & -\mathbf{L_3^n} & \mathbf{D_3^n} \end{pmatrix} \begin{pmatrix} \mathbf{0} \\ \mathbf{X_{22}} \\ \mathbf{X_{32}} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{I_2} \\ \mathbf{0} \end{pmatrix} , \tag{D.4}$$

$$\begin{pmatrix} \mathbf{D_1} & \mathbf{0} & \mathbf{0} \\ -\mathbf{L_2^n} & \mathbf{D_2^n} & \mathbf{0} \\ \mathbf{0} & -\mathbf{L_3^n} & \mathbf{D_3^n} \end{pmatrix} \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{X_{33}} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{I_3} \end{pmatrix} . \tag{D.5}$$

Since $\mathbf{Z}^n$ is a block lower triangular matrix, one can use forward substitution to solve the equations in (D.3), (D.4) and (D.5). We use (D.3) as an example, which can be written as a system of linear equations as follows.

$$\mathbf{D_1}\mathbf{X_{11}} = \mathbf{I_1} \tag{D.6}$$

$$-\mathbf{L_2^n}\mathbf{X_{11}} + \mathbf{D_2^n}\mathbf{X_{21}} = \mathbf{0} \tag{D.7}$$

$$-\mathbf{L_3^n}\mathbf{X_{21}} + \mathbf{D_2^n}\mathbf{X_{31}} = \mathbf{0} \tag{D.8}$$

The first equation (D.6) only involves $\mathbf{X_{11}}$, thus one can solve for $\mathbf{X_{11}}$ directly. The second equation (D.7) only involves $\mathbf{X_{11}}$ and $\mathbf{X_{21}}$, and thus can be solved once the already solved value for $\mathbf{X_{11}}$ is substituted in. Continuing in this way, the third equation (D.8) only involves $\mathbf{X_{21}}$ and $\mathbf{X_{31}}$, and one can solve for $\mathbf{X_{31}}$ using the previously solved values for $\mathbf{X_{21}}$. The resulting solution is

$$\mathbf{X_{11}} = \mathbf{D_1^{-1}} , \tag{D.9}$$

$$\mathbf{X_{21}} = (\mathbf{D_2^n})^{-1}\mathbf{L_2^n}\mathbf{D_1^{-1}} , \tag{D.10}$$

$$\mathbf{X_{31}} = (\mathbf{D_3^n})^{-1}\mathbf{L_3^n}(\mathbf{D_2^n})^{-1}\mathbf{L_2^n}\mathbf{D_1^{-1}} . \tag{D.11}$$

Using the same forward substitution approach, one can solve equations in (D.4) and (D.5). Finally we obtain the solution of $\mathbf{X}$.

$$\mathbf{X} = \begin{pmatrix} \mathbf{D_1^{-1}} & \mathbf{0} & \mathbf{0} \\ (\mathbf{D_2^n})^{-1}\mathbf{L_2^n}\mathbf{D_1^{-1}} & (\mathbf{D_2^n})^{-1} & \mathbf{0} \\ (\mathbf{D_3^n})^{-1}\mathbf{L_3^n}(\mathbf{D_2^n})^{-1}\mathbf{L_2^n}\mathbf{D_1^{-1}} & (\mathbf{D_3^n})^{-1}\mathbf{L_3^n}(\mathbf{D_2^n})^{-1} & (\mathbf{D_3^n})^{-1} \end{pmatrix} \tag{D.12}$$

The matrix $\mathbf{X}$ is a block lower triangular matrix with $(\mathbf{D_1})^{-1}$ and $(\mathbf{D_i^n})^{-1}$ $(i = 2, 3)$ on the block diagonals. The determinant $\det(\mathbf{X}) = \det(\mathbf{D_1}^{-1})\det((\mathbf{D_2^n})^{-1})\det((\mathbf{D_3^n})^{-1})$. Since $\mathbf{D_1}$ and $\mathbf{D_i^n}$ $(i = 2, 3)$ are block M matrices, $\mathbf{D_1}^{-1}$ and $(\mathbf{D_i^n})^{-1}$ $(i = 2, 3)$ are non-singular. So determinants $\det(\mathbf{D_1}^{-1}) \neq 0$ and $\det((\mathbf{D_i^n})^{-1}) \neq 0$ $(i = 2, 3)$. Therefore $\det(\mathbf{X}) \neq 0$, hence $\mathbf{X}$ is nonsingular. This proves that $(\mathbf{Z}^n)^{-1}$ is nonsingular.

Due to the fact that $\mathbf{D_1}$ and $\mathbf{D_i^n}$ $(i = 2, 3)$are M matrices, we have $(\mathbf{D_1})^{-1} \geq 0$ and $(\mathbf{D_i^n})^{-1} \geq 0$ $(i = 2, 3)$. Together with the fact that $\mathbf{L_i^n} \geq 0$, it follows that $\mathbf{X_{ij}} \geq 0$ $(i, j = 1, 2, 3)$. Because $(\mathbf{Z}^n)^{-1} = \mathbf{X}$ is nonsingular and $(\mathbf{Z}^n)^{-1} \geq 0$, $\mathbf{Z}^n$ is an M matrix.

Continuing the same way of forward substitution when $\mathbf{Z}^n$ is a $j_{\max} \times j_{\max}$ block matrix, one can obtain the solution of the block lower triangular matrix $\mathbf{X}$ shown in (D.13). Similarly, we can prove that $\mathbf{Z}^n$ is an M matrix for $j_{\max} > 3$.

$$\mathbf{X_{ij}} = \begin{cases} (\mathbf{D_1})^{-1}, \quad (i = j = 1) \\ (\mathbf{D_i^n})^{-1}, \quad (1 < i = j \leq j_{\max}) \\ \mathbf{0}, \quad (1 \leq i < j \leq j_{\max}) \\ (\mathbf{D_i^n})^{-1}\mathbf{L_i^n}(\mathbf{D_{i-1}^n})^{-1}\mathbf{L_{i-1}^n} \ldots (\mathbf{D_{j+1}^n})^{-1}\mathbf{L_{j+1}^n}(\mathbf{D_j^n})^{-1}, \quad (1 < j < i \leq j_{\max}) \\ (\mathbf{D_i^n})^{-1}\mathbf{L_i^n}(\mathbf{D_{i-1}^n})^{-1}\mathbf{L_{i-1}^n} \ldots (\mathbf{D_2^n})^{-1}\mathbf{L_2^n}\mathbf{D_1^{-1}}, \quad (1 = j < i \leq j_{\max}) \,. \end{cases} \tag{D.13}$$

Appendix E

Continuity of Local Optimization Objective Function

Define local optimization problem objective function as

$$G(\mathbf{v}) \;=\; \max_{\chi_\ell \in X}\left\{-\mathbf{A}(\chi)\mathbf{v}^k + \mathbf{B}(\chi)\mathbf{v}^k + \mathbf{c}(\chi)\right\} . \tag{E.1}$$

We would like to show that $G(\mathbf{v}^k)$ is a continuous function of $\mathbf{v}^k$. That is

$$\lim_{\mathbf{v}^k \to \mathbf{v}^*} G(\mathbf{v}^k) \;=\; G(\mathbf{v}^*) . \tag{E.2}$$

Proof. Let

$$\chi^k = \arg\max_{\chi_\ell \in X}\left\{-\mathbf{A}(\chi)\mathbf{v}^k + \mathbf{B}(\chi)\mathbf{v}^k + \mathbf{c}(\chi)\right\} , \tag{E.3}$$

then by definition of $G(\mathbf{v})$ in (E.1), we have

$$G(\mathbf{v}^k) = -\mathbf{A}(\chi^k)\mathbf{v}^k + \mathbf{B}(\chi^k)\mathbf{v}^k + \mathbf{c}(\chi^k) \ , \tag{E.4}$$

$$G(\mathbf{v}^k) \geq -\mathbf{A}(\chi)\mathbf{v}^k + \mathbf{B}(\chi)\mathbf{v}^k + \mathbf{c}(\chi) \ , \forall \chi \in X. \tag{E.5}$$

Similarly, let

$$\chi^* = \arg\max_{\chi_\ell \in X} \left\{ -\mathbf{A}(\chi)\mathbf{v}^* + \mathbf{B}(\chi)\mathbf{v}^* + \mathbf{c}(\chi) \right\} \ , \tag{E.6}$$

we have

$$G(\mathbf{v}^*) = -\mathbf{A}(\chi^*)\mathbf{v}^k + \mathbf{B}(\chi^*)\mathbf{v}^k + \mathbf{c}(\chi^*) \ , \tag{E.7}$$

$$G(\mathbf{v}^*) \geq -\mathbf{A}(\chi)\mathbf{v}^k + \mathbf{B}(\chi)\mathbf{v}^k + \mathbf{c}(\chi) \ , \forall \chi \in X \ . \tag{E.8}$$

Since (E.8) holds for all $\chi \in X$, it also holds for χ^k. Substitute χ with χ^k into (E.8) and then subtract the resulting equation from equation (E.4), we obtain

$$\begin{aligned} G(\mathbf{v}^k) - G(\mathbf{v}^*) &\leq -\mathbf{A}(\chi^k)\mathbf{v}^k + \mathbf{B}(\chi^k)\mathbf{v}^k + \mathbf{c}(\chi^k) \\ &\quad -\left[-\mathbf{A}(\chi^k)\mathbf{v}^* + \mathbf{B}(\chi^k)\mathbf{v}^* + \mathbf{c}(\chi^k)\right] \\ &= \left[-\mathbf{A}(\chi^k) + \mathbf{B}(\chi^k)\right](\mathbf{v}^k - \mathbf{v}^*) \ . \end{aligned} \tag{E.9}$$

Similarly, substitute χ with χ^* into equation (E.5) and together with equation (E.7), we obtain

$$\begin{aligned} G(\mathbf{v}^k) - G(\mathbf{v}^*) &\geq -\mathbf{A}(\chi^*)\mathbf{v}^k + \mathbf{B}(\chi^*)\mathbf{v}^k + \mathbf{c}(\chi^*) \\ &\quad -\big[-\mathbf{A}(\chi^*)\mathbf{v}^* + \mathbf{B}(\chi^*)\mathbf{v}^* + \mathbf{c}(\chi^*)\big] \\ &= \big[-\mathbf{A}(\chi^*) + \mathbf{B}(\chi^*)\big](\mathbf{v}^k - \mathbf{v}^*) \ . \end{aligned} \tag{E.10}$$

Consequently, we have

$$\Big[-\mathbf{A}(\chi^*) + \mathbf{B}(\chi^*)\Big](\mathbf{v}^k - \mathbf{v}^*) \leq G(\mathbf{v}^k) - G(\mathbf{v}^*) \leq \Big[-\mathbf{A}(\chi^k) + \mathbf{B}(\chi^k)\Big](\mathbf{v}^k - \mathbf{v}^*) \ . \tag{E.11}$$

Since $\mathbf{A}(\chi)$ and $\mathbf{B}(\chi)$ are bounded independent from χ (by Condition 6.2.1), then we have

$$\lim_{\mathbf{v}^k \to \mathbf{v}^*} G(\mathbf{v}^k) - G(\mathbf{v}^*) = 0 \ . \tag{E.12}$$

□

Appendix F

Floating Point Arithmetic Error Analysis

F.1 Roundoff Error Propagation

Let $\mathbf{x} = (x_1, x_2, \ldots, x_n)'$. To compute a function $y = \phi(\mathbf{x})$ by using floating point arithmetic, an error Δy of $y\delta_0$ has to be expected, where $|\delta_0| < \delta$, the machine epsilon [50]. Furthermore, there exists an input error $\mathbf{\Delta x} = (\Delta x_1, \Delta x_2, \ldots, \Delta x_n)'$ due to the floating point representation of real numbers or previous calculation of $\mathbf{x}$ (we do not consider the measurement input error because it is beyond the control of numerical computation method). The two sources of error are unavoidable no matter how we arrange the floating point operations. The third source of error comes from the intermediate roundoff errors and it depends how we arrange the floating point operations. Based on differential error analysis, the total floating point arithmetic error of computing y denoted by Δy, to the first order approximation, is given by

$$\Delta y = \mathcal{D}\phi(\mathbf{x})\Delta\mathbf{x} + y\delta_0 \ + \sum_{i=1}^{r} \Delta^{(i)} y \tag{F.1}$$

with $\mathcal{D}\phi(\mathbf{x})$ being the Jacobian matrix of $\phi(\mathbf{x})$ and $\Delta^{(i)}y$ being the intermediate roundoff error generated at step i. We assume there are r intermediate steps and each step performs elementary operations such as $+, -, \times, \div$ and $\sqrt{\ }$ [50].

F.2 Derivative Roundoff Error by Finite Difference

Using the standard finite difference method to compute the first derivative involves floating point arithmetic of computing the function with form

$$y = \phi_1(\mathbf{x}) = \frac{x_1 - x_2}{x_3} \ . \tag{F.2}$$

Let the input relative error be denoted by $\delta_{\mathbf{x}} = (\delta_{x_1}, \delta_{x_2}, \delta_{x_3})' = (\Delta x_1/x_1, \Delta x_2/x_2, \Delta x_3/x_3)'$. If we compute $y_1 = x_1 - x_2$ first, then proceed to divide the intermediate result y_1 by x_3, from equation (F.1), we have

$$\Delta y \ = \ \delta_{x_1}\frac{x_1}{x_3} - \delta_{x_2}\frac{x_2}{x_3} - \delta_{x_3} y + \delta_0 y + \delta_1 y$$

where $|\delta_i| < \delta (i = 1, 2)$ and $\delta_1 y$ is the intermediate roundoff error. Further assuming $|\delta_{x_3}| \le \delta$ and $|x_3| \le \Delta h_{\min}$, we obtain the bound of Δy as follows

$$|\Delta y| \ \le \ |\delta_{x_1}|\frac{|x_1|}{\Delta h_{\min}} + |\delta_{x_2}|\frac{|x_2|}{\Delta h_{\min}} + 3\delta|y| \ . \tag{F.3}$$

Let

$$x_2 = (1 + a_1)x_1,\ x_1 = (1 + a_2)x_2, \qquad |x_3| \le \Delta h_{\min}\ . \tag{F.4}$$

The bound of y is given by

$$|y| \le |a_i|\frac{|x_i|}{\Delta h_{\min}}\ , \qquad i = 1, 2\ . \tag{F.5}$$

Suppose input error $\Delta\mathbf{x}$ is due to representing the real number in the floating point system or from a previous calculation whose error is within machine epsilon δ, so we have $\|\delta_{\mathbf{x}}\|_\infty \le \delta$ and consequently

$$|\Delta y| \le \delta(2 + 4|a_i|)\frac{|x_i|}{\Delta h_{\min}} \qquad i = 1, 2\ . \tag{F.6}$$

Applying the result to discretized $\mathcal{D}^h_W V^{n+1}_{i,j}$ and $\mathcal{D}^h_A V^{n+1}_{i,j}$ in equation (B.3), we obtain the absolute roundoff error of computing first derivatives by using backward difference as follows

$$\begin{aligned}
|\Delta\mathcal{D}^h_A V^{n+1}_{i,j}| \le &\quad \delta(2 + 4|a_3|)\tfrac{|V^{n+1}_{i,j}|}{\Delta A^-_j}\ ,\quad V^{n+1}_{i,j-1} = (1 + a_3)V^{n+1}_{i,j} \\
|\Delta\mathcal{D}^h_W V^{n+1}_{i,j}| \le &\quad \delta(2 + 4|a_4|)\tfrac{|V^{n+1}_{i,j}|}{\Delta W^-_i}\ ,\quad V^{n+1}_{i-1,j} = (1 + a_4)V^{n+1}_{i,j} \\
|\Delta\mathcal{D}^h_W V^{n+1}_{i+1,j}| \le &\quad \delta(2 + 4|a_5|)\tfrac{|V^{n+1}_{i,j}|}{\Delta W^+_i}\ ,\quad V^{n+1}_{i+1,j} = (1 + a_5)V^{n+1}_{i,j}\ .
\end{aligned} \tag{F.7}$$

Applying the result in equation (F.5) to $\mathcal{D}^h_A V_{i,j}$, $\mathcal{D}^h_W V_{i,j}$ and $\mathcal{D}^h_W V^{n+1}_{i+1,j}$, we have

$$|\mathcal{D}^h_A V^{n+1}_{i,j}| \le |a_3|\frac{|V^{n+1}_{i,j}|}{\Delta A^-_j}\ ,\ |\mathcal{D}^h_W V^{n+1}_{i,j}| \le |a_4|\frac{|V^{n+1}_{i,j}|}{\Delta(W_{\min})_i}\ ,\ |\mathcal{D}^h_W V^{n+1}_{i+1,j}| \le |a_5|\frac{|V^{n+1}_{i,j}|}{\Delta(W_{\min})_i}\ , \tag{F.8}$$

where $(\Delta W_{\min})_i = \min_i(W_{i+1} - W_i, W_i - W_{i-1})$. Together with the standard 3 point finite difference method to compute the second derivative as in equation (B.3) , we obtain

$$\begin{aligned} |\mathcal{D}^h_{WW} V^{n+1}_{i,j}| &\leq (|\mathcal{D}^h_W V^{n+1}_{i+1,j}| + |\mathcal{D}^h_W V^{n+1}_{i,j}|)\frac{1}{(\Delta W_{\min})_i} \\ &\leq (|a_4| + |a_5|)\frac{|V^{n+1}_{i,j}|}{(\Delta W_{\min})^2_i} \ . \end{aligned} \tag{F.9}$$

To bound the roundoff error of $\mathcal{D}^h_{WW} V^{n+1}_{i,j}$, set

$$\mathbf{x} = (\mathcal{D}^h_W V^{n+1}_{i+1,j}, \mathcal{D}^h_W V^{n+1}_{i,j}, \frac{\Delta W^{\pm}_i}{2})' \ , \qquad \delta_{\mathbf{x}} = (\frac{\Delta \mathcal{D}^h_W V^{n+1}_{i+1,j}}{\mathcal{D}^h_W V^{n+1}_{i+1,j}}, \frac{\Delta \mathcal{D}^h_W V^{n+1}_{i,j}}{\mathcal{D}^h_W V^{n+1}_{i,j}}, \delta_{x_3})' \ . \tag{F.10}$$

Assuming $|\delta_{x_3}| < \delta$, by equations (B.3), (F.3), (F.8) and (F.9) and the fact that $\Delta W^{\pm}_i/2 \geq \Delta W_{\min}$, we obtain the following bound

$$\begin{aligned} |\Delta \mathcal{D}^h_{WW} V^{n+1}_{i,j}| &\leq \frac{|\Delta \mathcal{D}^h_W V^{n+1}_{i,j}| + |\Delta \mathcal{D}^h_W V^{n+1}_{i+1,j}|}{\Delta W_{\min}} + 3|\mathcal{D}^h_{WW} V^{n+1}_{i,j}| \\ &\leq \delta(4 + 5|a_4| + 5|a_5|)\frac{|V^{n+1}_{i,j}|}{(\Delta W_{\min})^2_i} \ . \end{aligned} \tag{F.11}$$

F.3 Roundoff Error Estimation of Local Optimization Problem

During each iteration, we solve a local optimization problem and the objective function involves calculating the following two terms

$$f_1(W_i, A_j, \tau^{n+1}) = \kappa - 1 + \mathcal{D}^h_W V^{n+1}_{i,j} + \mathcal{D}^h_A V^{n+1}_{i,j} \tag{F.12}$$

$$f_2(W_i, A_j, \tau^{n+1}) = \frac{\sigma^2 W_i^2}{2} \mathcal{D}^h_{WW} V^{n+1}_{i,j} + O(W_i) \ . \tag{F.13}$$

Computing f_1 involves calculating a function of the form $g(\mathbf{x}) = (x_1 + x_2) + (x_3 + x_4)$.

From equation (F.1), we obtain

$$
\begin{aligned}
|\Delta g| &\leq \sum_{i=1}^{4} |\Delta x_i| + \delta|g| + \delta|x_1 + x_2| + \delta|x_3 + x_4| && \text{(F.14)} \\
&\leq \sum_{i=1}^{4} |\Delta x_i| + 2\delta|x_1 + x_2| + 2\delta(|x_3| + |x_4|) \ . && \text{(F.15)}
\end{aligned}
$$

Setting $x_1 = \kappa, x_2 = -1, x_3 = \mathcal{D}_W^h V_{i,j}^{n+1}, x_4 = \mathcal{D}_A^h V_{i,j}^{n+1}$ and applying equations (F.7) and (F.8) with the fact that $0 < \kappa < 1$, we obtain the bound of absolute roundoff error of f_1 as follows

$$
\begin{aligned}
|\Delta f_1| &\leq \delta(2 + 6|a_3|)\frac{|V_{i,j}^{n+1}|}{\Delta A_{\min}} + \delta(2 + 6|a_4|)\frac{|V_{i,j}^{n+1}|}{(\Delta W_{\min})_i} + 3(1-\kappa)\delta \\
&\lessapprox 2\delta(\frac{1 + 3|a_3|}{\Delta A_{\min}} + \frac{1 + 3|a_4|}{\Delta W_{\min}})|V_{i,j}^{n+1}| \ , && \text{(F.16)}
\end{aligned}
$$

where we discard the smaller error term of $3\delta(1-\kappa)$, and $\Delta A_{\min} = \min_j(A_j - A_{j-1})$ and $\Delta W_{\min} = \min_i(W_i - W_{i-1})$.

To analyze the roundoff error of f_2, we notice that only multiplication and division operations are involved given $\mathcal{D}_{WW}^h V_{i,j}^{n+1}$ as one of the operands. From equation (F.1), it can be easily seen that the roundoff error of

$$
\begin{aligned}
g_2(\mathbf{x}) = x_1 \times x_2 \qquad & |\Delta g_2| \leq (|\delta_{x_1}| + |\delta_{x_2}| + |\delta_0|)|g_2| && \text{(F.17)} \\
g_3(\mathbf{x}) = x_1 \div x_2 \qquad & |\Delta g_2| \leq (|\delta_{x_1}| + |\delta_{x_2}| + |\delta_0|)|g_3| \ . && \text{(F.18)}
\end{aligned}
$$

So computing of $c_i = \sigma^2 W_i^2/2$ will accumulate $9\delta|c_i|$ roundoff errors assuming the input error of σ and W_i is smaller than δ, the machine epsilon. The final roundoff error of

$f_2 = c_i \mathcal{D}^h_{WW} V^{n+1}_{i,j}$ is then given by

$$\begin{aligned} |\Delta f_2| &\leq (10|\delta| + \frac{|\Delta \mathcal{D}^h_{WW} V^{n+1}_{i,j}|}{|\mathcal{D}^h_{WW} V^{n+1}_{i,j}|})|c_i||\mathcal{D}^h_{WW} V^{n+1}_{i,j}| \\ &\leq \delta(4 + 15|a_4| + 15|a_5|)\frac{\sigma^2 W_i^2}{2(\Delta W_{\min})_i^2}|V^{n+1}_{i,j}| \ . \end{aligned} \quad \text{(F.19)}$$

In the area where the grids are fine, we have $V_{i,j} \approx V_{i\pm 1,j} \approx V_{i,j-1}$. So normally $|a_i| \ll 1$ for $i = 3, 4, 5$. It may be safe to estimate that $|a_i| \leq 0.1, i = 3, 4, 5$. Finally the following estimation of roundoff errors of computing f_1 and f_2 are obtained

$$\begin{aligned} |\Delta(\kappa - 1 - \mathcal{D}^h_W V^{n+1}_{i,j} + \mathcal{D}^h_A V^{n+1}_{i,j})| &\leq 4\delta(\frac{1}{\Delta A_{\min}} + \frac{1}{\Delta W_{\min}})|V^{n+1}_{i,j}| \\ &\leq 4\delta(\frac{1}{\Delta A_{\min}} + \frac{1}{\Delta W_{\min}}) \max(|V^{n+1}_{i,j}|, scale) \ , \end{aligned} \quad \text{(F.20)}$$

$$\begin{aligned} |\Delta(\frac{\sigma^2 W_i^2}{2}\mathcal{D}^h_{WW} V^{n+1}_{i,j})| &\leq 4\delta\frac{\sigma^2 W_i^2}{\Delta W^2_{\min}}|V^{n+1}_{i,j}| \\ &\leq 4\delta\frac{\sigma^2 W_i^2}{\Delta W^2_{\min}} \max(|V^{n+1}_{i,j}|, scale) \ . \end{aligned} \quad \text{(F.21)}$$

Bibliography

[1] R. Almgren and N. Chriss. Optimal execution of portfolio transactions. *Journal of Risk*, 3:5–39, 2000/2001 (Winter). 5

[2] L. Andersen and J. Andreasen. Jump-diffusion processes: Volatility smile fitting and numerical methods for option pricing. *Review of Derivatives Research*, 4:231–262, 2000. 98

[3] G. Barles. Convergence of numerical schemes for degenerate parabolic equations arising in finance. In L. C. G. Rogers and D. Talay, editors, *Numerical Methods in Finance*, pages 1–21. Cambridge University Press, Cambridge, 1997. 4, 7, 17, 32, 36

[4] G. Barles, CH. Daher, and M. Romano. Convergence of numerical schemes for parabolic equations arising in finance theory. *Mathematical Models and Methods in Applied Sciences*, 5:125–143, 1995. 19, 42

[5] G. Barles and P.E. Souganidis. Convergence of approximation schemes for fully nonlinear equations. *Asymptotic Analysis*, 4:271–283, 1991. 4, 7, 17, 32, 42, 83, 92

[6] D. Bauer, A. Kling, and J. Russ. A universal pricing framework for guaranteed minimum benefits in variable annuities. *ASTIN Bulletin*, 38:621–651, 2008. 3, 13, 136

[7] D.P. Bertsekas and J. Tsitsiklis. *Neuro-dynamic Programming*. Athena, 1996. 116

[8] F. Black and M. Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3), 1973. 60, 85

[9] O. Bokanowski, S. Maroso, and H. Zidani. Some convergence results for Howard's algorithm. *SIAM Journal on Numerical Analysis*, 47:3001–3026, 2009. 4, 5, 46, 58, 105, 130

[10] Z. Chen and P.A. Forsyth. A semi-Lagrangian approach for natural gas storage valuation and optimal control. *SIAM Journal on Scientific Computing*, 30:339–368, 2007. 5

[11] Z. Chen and P.A. Forsyth. A numerical scheme for the impulse control formulation for pricing variable annuities with a guaranteed minimum withdrawal benefit (GMWB). *Numerische Mathematik*, 109:535–569, 2008. 3, 15, 39, 108, 109, 112, 113, 132

[12] Z. Chen and P.A. Forsyth. Implications of a regime switching model on natural gas storage valuation and optimal operation. *Quantitative Finance*, 10:159–176, 2009. 5, 6

[13] Z. Chen, K. Vetzal, and P.A. Forsyth. The effect of modelling parameters on the value of GMWB guarantees. *Insurance: Mathematics and Economics*, 43:165–173, 2008. 3, 13, 98, 110, 136

[14] S.S. Clift. *Linear and non-linear monotone methods for valuing financial options under two factor, jump diffusion models.* PhD thesis, School of Computer Science, University of Waterloo, 2007. 74

[15] R. Cont and P. Tankov. *Financial Modelling with Jump Processes.* Chapman and Hall, 2004. 1, 4, 5, 6, 7, 56

[16] E. Cramer, P. Matson, and L. Rubin. Common practices relating to FASB statement 133, Accounting for Derivative Instruments and Hedging Activities as it Relates to Variable Annuities with Guaranteed Benefits. Practice Note, American Academy of Actuaries, available at www.actuary.org, 2007. 3

[17] M. G. Crandall, H. Ishii, and P. L. Lions. User's guide to viscosity solutions of second order partial differential equations. *Bulletin of the American Mathematical Society*, 27:1–67, 1992. 4

[18] M. Dai, Y.K. Kwok, and J. Zong. Guaranteed minimum withdrawal benefit in variable annuities. *Mathematical Finance*, 18:595–611, 2008. 3, 4, 10, 12, 13, 14, 15, 18, 20, 26, 42, 57, 60, 61, 62, 130

[19] M. Dai and Y. Zhong. Penalty methods for continuous time portfolio selection with transaction costs. *Journal of Computational Finance*, 13(3):1–31, 2010. 4, 5

[20] Y. d'Halluin. *Numerical Methods for Real Options in Telecommunications*. PhD thesis, School of Computer Science, University of Waterloo, 2004. 60

[21] Y. d'Halluin, P.A. Forsyth, and G. Labahn. A penalty method for American options with jump diffusion processes. *Numerische Mathematik*, 97:321–352, 2004. 5, 73, 79, 80

[22] Y. d'Halluin, P.A. Forsyth, and G. Labahn. A semi-Lagrangian approach for American Asian options under jump diffusion. *SIAM Journal on Scientific Computing*, 27(1):315–345, 2005. 62

[23] Y. d'Halluin, P.A. Forsyth, and K.R. Vetzal. Robust numerical methods for contingent claims under jump diffusion processes. *IMA Journal of Numerical Analysis*, 25:87–112, 2005. 15, 60, 61, 62, 65, 66, 68, 82, 83, 85, 88, 92, 95

[24] B. Eraker and M. Johannes. The impact of jumps in volatility and returns. *Journal of Finance*, 58:1269–1300, 2003. 4, 56

[25] J. Fenton and D. Czernicki. Pricing GMWB features in a challenging market. Towers Watson Report, available at www.towerswatson.com, 2010. 2, 13

[26] P.A. Forsyth and G. Labahn. Numerical methods for controlled Hamilton-Jacobi-Bellman PDEs in finance. *Journal of Computational Finance*, 11 (Winter):1–44, 2008. 5, 41, 43, 67, 68, 69, 70, 83, 84, 85, 92, 93, 94

[27] P.A. Forsyth and K.R. Vetzal. Quadratic convergence of a penalty method for valuing American options. *SIAM Journal on Scientific Computing*, 23:2095–2122, 2002. 4

[28] C. Gilbert, K. Ravindran, and R. Reitano. Results on the survey on variable annuity hedging programs for life insurance companies. Society of Actuaries Report, available at www.soa.org, 2007. 13, 136

[29] M. Hardy. A regime switching model of long term stock returns. *North American Actuarial Journal*, 5:41–53, 2001. 6

[30] Y. Huang and P.A. Forsyth. Analysis of a penalty method for pricing a guaranteed minimum withdrawal benefit (GMWB). To appear in IMA Journal of Numerical Analysis, 2010. 7, 57, 100, 119, 122

[31] Y. Huang, P.A. Forsyth, and G. Labahn. Combined fixed point and policy iteration for HJB equations in finance. Working paper, University of Waterloo, submitted to SIAM Journal on Numerical Analysis, 2010. 4, 6, 7, 46

[32] Insured Retirement Institute. Variable Annuity Assets Soar in 4Q, February 25, 2010. 2

[33] E. Jakobsen. Monotone schemes. In R. Cont, editor, *Encyclopedia of Quantitative Finance*, pages 1253–1263. Wiley, New York, 2010. 36

[34] A.Q.M. Khaliq and R.H. Liu. New numerical scheme for pricing American options with regime switching. *International Journal of Theoretical and Applied Finance*, 12:319–340, 2009. 6

[35] H. Kushner. *Numerical Methods for Stochastic Control Problems in Continuous Time.* Springer, 2001. 4, 5, 80

[36] Y. Liu. *Pricing and Hedging the Guaranteed Minimum Withdrawal Benefits in Variable Annuities.* PhD thesis, Statistics and Actuarial Science Department , University of Waterloo, 2010. 3

[37] J. Lorenz and R. Almgren. Adaptive arrival price. In Algorithmic Trading III: Precision, Control, Execution, Brian R. Bruce, editor, Institutional Investor Journals, 2007. 5

[38] R.C. Merton. Theory of rational option pricing. *The Bell Journal of Economics and Management Science*, pages 141–183, 1973. 60, 85

[39] R.C. Merton. Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics*, 3:125–144, 1976. 1, 4, 5, 6, 7, 56, 57

[40] M.A. Milevsky and T.S. Salisbury. Financial valuation of guaranteed minimum withdrawal benefits. *Insurance: Mathematics and Economics*, 38:21–38, 2006. 2, 3, 13, 57, 130

[41] R. Munos. Error bounds for approximate policy iteration. *Proceedings of the 20th International Congress on Machine Learning, Washington*, pages 560–567, 2003. 116

[42] H. Pham. On some recent aspects of stochastic control and their applications. *Probability Surveys*, 2:506–549, 2005. 3, 4, 5, 18

[43] H. Pham. *Continuous-time Stochastic Control and Optimization with Financial Applications*. Springer, 2009. 4, 5

[44] D.M. Pooley, P.A. Forsyth, and K.R. Vetzal. Numerical convergence properties of option pricing PDEs with uncertain volatility. *IMA Journal of Numerical Analysis*, 23:241–267, 2003. 26, 45, 139

[45] M. Puterman and M.C. Shin. Modified policy iteration for discounted Markov decision problems. *Management Science*, 24:1127–1137, 1978. 80

[46] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, 2003. 71

[47] M. Santos and J. Rust. Convergence properties of policy iteration. *SIAM Journal on Control and Optimization*, 42:2094–2115, 2004. 105

[48] L. Scism and E. Holm. Variable Annuities' Giants Grow. *The Wall Street Journal*, Novermber 17, 2010. 2

[49] R.C. Seydel. General existence and uniqueness of viscosity solutions for impulse control of jump-diffusions. *Arxiv preprint arXiv:1101.0172*, 2010. 18

[50] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis*. Springer-Verlag, second edition, 1993. 148, 149

[51] M. Thompson, M. Davison, and H. Rasmussen. Natural gas storage valuation and optimization: A real options application. *Naval Research Logistics*, 56:226–238, 2009. 5

[52] A. Tourin and T. Zariphopoulou. Viscosity solutions and numerical schemes for investment/consumption models with transaction costs. In *Numerical Methods in Finance*, editors: L.C.G. Rogers and D. Talay, Cambridge University Press, 1997. 3, 5

[53] R. Varga. *Matrix Iterative Analysis*. Prentice Hall, 1961. 31, 74, 93

[54] I.R. Wang, J.W.I. Wan, and P.A. Forsyth. Robust numerical valuation of European and American options under the CGMY process. *Journal of Computational Finance*, 10:4(Summer):86–115, 2007. 62, 66, 83, 85, 92, 95

[55] J. Wang and P.A. Forsyth. Maximal use of central differencing for Hamilton-Jacobi-Bellman PDEs in finance. *SIAM Journal on Numerical Analysis*, 46:1580–1601, 2008. 7, 27, 70, 80, 130

[56] 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*, 34:207–230, 2010. 5, 70

[57] V.I. Zakamouline. A unified approach to portfolio optimization with linear transaction costs. *Mathematical Methods of Operations Research*, 62:319–343, 2005. 110, 113