Numerical Methods for Long-Term Impulse Control Problems in Finance

by

Amélie Bélanger

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2008

Abstract

Several of the more complex optimization problems in finance can be characterized as impulse control problems. Impulse control problems can be written as quasi-variational inequalities, which are then solved to determine the optimal control strategy. Since most quasi-variational inequalities do not have analytical solutions, numerical methods are generally used in the solution process.

In this thesis, the impulse control problem framework is applied to value two complex long-term option-type contracts. Both pricing problems considered are cast as impulse control problems and solved using an implicit approach based on either the penalty method or the operator splitting scheme.

The first contract chosen is an exotic employee stock option referred to as an infinite reload option. This contract provides the owner with an infinite number of reload opportunities. Each time a reload occurs, the owner pays the strike price using pre-owned company shares and, in return, receives one share for each option exercised and a portion of a new reload option. Numerical methods based on the classic Black-Scholes equation are developed while taking into account contract features such as vesting periods. In addition, the value of an infinite reload option to it's owner is obtained by using a utility maximization approach.

The second long-term contract considered is a variable annuity with a guaranteed minimum death benefit (GMDB) clause. Numerical methods are developed to determine the cost of the GMDB clause while including features such as partial withdrawals. The pricing model is then used to determine the *fair insurance charge* which minimizes the cost of the contract to the issuer. Due to the long maturity of variable annuities, non-constant market parameters expressed through the use of regime-switching are included in the GMDB pricing model.

Acknowledgements

This thesis is the result of many years of hard work during which time I was encouraged and supported by many people; I would like to extend my thanks to all of them.

First, I would like to thank Professor Peter Forsyth who supervised the bulk of my PhD work. Most of the results obtained here would not have been possible without his continuous guidance and insight. Thanks must also be extended to my co-supervisor Professor George Labahn. His input on technical proofs was invaluable and enabled me to make the last chapter of this thesis readable. I would also like to thank my committee members for taking the time to review my thesis and provide me with valuable comments: Professors Christina Christara, Ken Vetzal, Justin Wan and Sivabal Sivaloganathan.

While my time at Waterloo was motivated by academic pursuit, it was made enjoyable both on and off-campus by many people. Members of the Scicom group both past and present made my daily life much more interesting; special mention must be made of Shannon, Simon, Zhuliang, Iris, Omar and Lei. One of the things that I will miss most about UW is the CS Grad Basketball team; special mention must be made of the more 'senior' members such as Dana, Hao, Richard, Georgia and Tyrel. Even though we didn't always win, we always managed to have fun.

I also made quite a few close friends while at UW, most of which have now moved on to bigger and better things including Howard, Dave, Liz, Shannon, Dana, Stacy and Reinhold. Hopefully, we will find a way to stay in touch even if life has squattered us across the continent.

A special thanks to my family for their continuous support and encouragement: my parents Paul and Suzanne and my little brother Philippe. Finally, I would like to thank my husband Yann for his boundless patience, help and support.

Contents

1 Introduction **1**

1.1 Contributions 4

2 Infinite Reload Options **7**

2.1 Increased Reload Pricing Problem 9

2.1.1 Boundary Conditions (No Similarity Reduction) 13

2.2 Discretization of the Reload Pricing Problem 16

2.2.1 Scaled Grid Construction 17

2.2.2 Discrete Equations 18

2.3 Convergence to the Viscosity Solution 22

2.3.1 Stability 23

2.3.2 Monotonicity 25

2.3.3 Consistency 27

2.4 Solution Algorithm 29

2.5 Numerical Results 34

2.5.1 Convergence Study 35

2.5.2 Implicit vs Explicit Application of the Reload Constraint 42

2.5.3 Adding a Volatility Surface 43

2.6 Summary 47

3 Infinite Reload Options with Vesting **51**

3.1 Option Pricing Model 53

3.1.1 Solution Domain and Boundary Conditions 57

3.1.2 Reformulation of the Continuous Pricing Problem 59

3.2 Derivation of the Discrete Equations 60

3.2.1 Theoretical Issues 67

3.2.2 Comments on the Solution Method 70

3.3 Numerical Results 71

3.4 Summary 78

4 Expected Utility Pricing **81**

4.1 Mathematical Model 83

4.1.1 Step 1: Solving the PDE 85

4.1.2 Semi-Lagrangian Solution 87

4.1.3 Step 2: Monte Carlo Simulations 91

4.1.4 Special case when $q = 0$ 93

4.2 Valuing Classic Employee Stock Options 94

4.2.1 Mathematical Model 94

4.2.2 Numerical Results 97

4.3 Valuing Infinite Reload Options 100

4.3.1 Mathematical Model 101

4.3.2 Underlying Grid and Discrete Equations 106

4.3.3 Numerical Results 107

4.4 Summary 112

5 GMDB with Partial Withdrawals **115**

5.1 GMDB Model with Constant parameters 119

5.1.1 Boundary Conditions 126

5.2 GMDB Guarantee with Regime-Switching 128

5.2.1 Boundary Conditions 132

5.3 Numerical Solution of the GMDB Problem 134

5.3.1 Construction of the Underlying Grid 134

5.3.2 Discrete Equations 136

5.3.3 Optimal Withdrawal 140

5.4 Convergence to the Viscosity Solution 140

5.4.1 Stability 143

5.4.2 Monotonicity 144

5.4.3 Consistency 145

5.5 Results from Numerical Experiments 148

5.5.1 Results for Constant Volatility 148

5.5.2 Numerical Results with Regime Switching 153

5.6 Summary 157

6 Conclusion **159**

6.1 Future Work 161

A Discretization **163**

B Proof of Theorem 2.4 **167**

C Proof of Theorem 2.10 **173**

D Reload Options with Cap **179**

E American Options with Operator Splitting **183**

F Derivation of Analytical Solution **187**

G Derivation of the Boundary Condition as $S \to \infty$ **191**

H Proof of Theorem 5.5 **193**

I Proof of Theorem 5.9 **201**

J Comparison with Previous GMDB Results **209**

Bibliography **213**

List of Tables

2.1 Parameter values used when pricing increased reload option contracts. . . 36

2.2 Value of an increased reload option with $p = 5\%$ at $S = \$100$ for different refinement levels. 37

2.3 Value of an increased reload option with $p = 0\%$ (infinite reload option) at $S = \$100$ for different refinement levels. 38

2.4 Value of an increased reload option with $p = 0\%$ at $S = \$100$ when the solution is obtained on a full $S \times K$ grid and when the solution is obtained using the similarity reduction. 38

2.5 Price of an increased reload option at $S = \$100$ for different p values. . . . 41

2.6 Value of an increased reload option contract with $p = 0\%$ when the reload constraint in (2.3) is applied both implicitly and explicitly. 43

2.7 Value of an increased reload option with $p = 0\%$ at $S = \$100$ when the reload constraint is applied explicitly for different refinement levels. 44

2.8 Value of an increased reload option when $p = 0\%$ at $S = \$100$ for different volatility assumptions. 46

3.1 Parameter values used when pricing infinite reload options with vesting. . 72

3.2 Value of an infinite reload option at $S = \$100$ with a vesting period of 0.5 years for different refinement levels. 73

3.3 Value of an infinite reload option with different vesting period lengths and different volatility values. 74
3.4 Value of an infinite reload option at $S = \$100$ with a 0.5 year vesting period where the reload constraint is applied explicitly. 75
3.5 Value of an infinite reload option with a maturity of 10 years at different S values for vesting periods ranging from 0.5 to 3 years. 77

4.1 Different employee situations or Cases as defined in [50]. Each case assumes one of two employee portfolio allocations (Sn_s, n_o and w) and one of two risk-aversion levels (α). 97
4.2 Expected utility of the employee when given classic stock options with $K = \$30$ for different refinement levels. 98
4.3 Parameters used when pricing classic employee stock options in a utility context. 99
4.4 Cash equivalent value of a classic employee stock option as a function of the employee's risk aversion coefficient and investment profile. 100
4.5 Parameters used when pricing infinite reload options in a utility framework. 108
4.6 Expected utility of an employee when given infinite reload option at $S = \$100$ for different refinement levels. 109
4.7 Cash equivalent value of an infinite reload option at $S = \$100$ as a function of the employee's external wealth level (w). 110
4.8 Cash equivalent value of an infinite reload option as a function of the dividend yield (q). 111
4.9 Error when using Monte Carlo simulations to determine the cash equivalent value of an infinite reload option when $q = 0$. 111

4.10 Cash equivalent value of an infinite reload option as a function of the distribution of the firm-related wealth (estimated at $12 million) between company stock and options. 112

5.1 Parameter values used when pricing the GMDB guarantee in the classic Black-Scholes context. 149

5.2 Cost of the GMDB guarantee when the owner is assumed to be a male of 50 years old at the time of purchase, $\omega = \$80$ and $\rho_{ins} = 0.008$. 151

5.3 Fair insurance fee (ρ_{ins}) for a GMDB guarantee for different grid refinement levels when the owner is assumed to be a male of 50 years old at the time of purchase and $\omega = \$80$. 151

5.4 Fair insurance charge (ρ_{ins}) for contracts containing a GMDB clause with annual ratchet events, as a function of the minimal deposit amount (ω). . 152

5.5 Fair insurance charge (ρ_{ins}) for a GMDB guarantee with different ratchet intervals ranging from 0.5 to 10 years. 153

5.6 Parameter values used when pricing GMDB contracts with regime-switching. Jump sizes and intensities taken from [5]. 154

5.7 Fair insurance fee (ρ_{ins}) for a GMDB guarantee with regime-switching for different grid refinement levels. 155

5.8 Fair insurance charge (ρ_{ins}) for contracts containing a GMDB clause with annual ratchet events as a function of the minimal deposit amount (ω) assuming the economy is in regime e_1. 157

D.1 Characteristics of four different grid constructions. 181

D.2 Value of an increased reload option with $p = 0\%$ at $S = \$100$ with a capped boundary condition for different underlying grids defined in Table D.1. . . 181

E.1 Value of an American put option at $S = \$100$ using the operator splitting method with different refinement levels. 185

E.2 Value of an American put option at $S = \$100$ for different refinement levels when the penalty method is used. 186

J.1 Fair insurance charge ρ_{ins} for a GMDB contract with discrete ratchet events. 211

List of Figures

2.1 Calculation of the reload constraint $\mathcal{A}V$ according to equation (2.16) on a typical $[0, S_{\max}] \times [0, K_{\max}]$ domain. 17

2.2 Example of a scaled grid construction for the two-dimensional $S \times K$ domain. 18

2.3 Diagonal interpolation is used when determining the value of new reload options $V(S_i^j, S_i^j(1+p), \tau^{n+1})$ in the reload constraint in equation (2.16). 22

2.4 Option value (V), delta (V_S) and gamma (V_{SS}) for increased reload options with different p values. 39

2.5 Value of an increased reload option (at $S = \$100$) as a function of the percentage increase p. 41

2.6 Effect of small changes in p on the exercise boundary of the increased reload option on the grant date. 42

2.7 Plot of volatility surface obtained by calibrating to synthetic market prices as outlined in [90]. 46

2.8 Plot of the option value (V), the delta (V_S) and the gamma (V_{SS}) for an increased reload option with $p = 0\%$ under different modelling assumptions. 48

3.1 Example of a three-dimensional $S \times K \times u$ domain where each $S \times K$ plane contains a scaled grid as shown in Figure 2.2. 61

3.2 Value of an infinite reload option as a function of the length of the vesting period (v_p). 77

3.3 Plot of the option value (V), the delta (V_S) and the gamma (V_{SS}) functions when pricing an infinite reload option with various different periods. . . . 79

5.1 Representation of a $[0, S_{\max}] \times [D, B_{\max}]$ plane where each one-dimensional S grid is built using the *scaled grid* technique defined in equation (5.38). 135

5.2 Three dimensional solution domain to price the GMDB guarantee in economic state e_m. Each $S \times B$ plane is constructed as in Figure 5.1. 136

List of Algorithms

2.1 Implicit Application of the Reload Constraint 31
2.2 Non-Linear Iteration 32
2.3 Explicit Application of the Reload Constraint 35
3.1 Update Process in Operator Splitting Method 66
5.1 Calculation of Withdrawal Constraint for GMDB Contracts 141
A.1 Coefficient Discretization 165

Chapter 1

Introduction

Optimal control problems arise in many areas of finance, and several can be characterized as *impulse control problems* [14]. By definition, an impulse control problem involves determining a sequence of intervention times as well as a sequence of corresponding optimal actions that modifies the value of the underlying asset [63]. Actions for example could include discrete dividends and buying or selling assets. Furthermore, impulse control problems can be written as quasi-variational inequalities, which are then solved to determine the optimal control strategy. The impulse control framework is a rather general approach which allows us to handle complex contractual features. For example, this approach has been applied to problems of optimal consumption and portfolio composition with transaction costs [62, 76, 66, 40], cash management and index tracking [17, 7], problems related to currency exchange rates [20, 21] and determining the optimal dividend policy of a company [19, 58].

To clarify further, let us define a continuous value function $F(x)$, where the underlying variable x (which may be a vector) follows a stochastic process. When an impulse occurs, the system state jumps from x to a new state $J(x, \alpha)$ where J is a given function and α

is the control. In this context, the non-local optimal control operator $\mathcal{A}$ is defined as:

$$\mathcal{A}F(x) = \sup_{\alpha} \left[F(J(x,\alpha)) + G(x,\alpha) - c \right], \tag{1.1}$$

where $F(J(x,\alpha))$ is the intervention term, $G(x,\alpha)$ is a cash flow or payoff term and c is a fixed cost. Note that the fixed cost c is included in equation (1.1) to ensure that the impulse control problem is well-posed. Singular control problems [93] can sometimes be viewed as a special case of impulse control problems where, in the limit, the fixed cost $c \to 0^+$. Using the operator defined in equation (1.1), an impulse control problem can be written as a set of quasi-variational inequalities:

$$\mathcal{L}F(x) \geq 0, \tag{1.2}$$

$$(\mathcal{A}F(x) - F(x)) \geq 0, \tag{1.3}$$

$$(\mathcal{A}F(x) - F(x))(\mathcal{L}F(x)) = 0, \tag{1.4}$$

where $\mathcal{L}$ is a differential operator [63].

Since most quasi-variational inequalities do not have analytical solutions, numerical methods are generally used in the solution process. Previous solution methods include an iterative scheme combining stochastic control and optimal stopping problems [24].

In this thesis, the impulse control framework is applied to value two complex long-term option-type contracts. The first contract considered is an exotic employee stock option known as an infinite reload option. The infinite reload option can be exercised an arbitrary number of times prior to maturity [39, 32]. When the option is exercised, the contract owner pays the current strike price with pre-owned company shares and, in return, receives one share for each option exercised and a portion of a new infinite reload option with increased strike price. The impulse control problem in this case determines

the optimal timing of exercise rights.

The second long-term contract valued in this thesis is a variable annuity contract containing a guaranteed minimum death benefit clause (GMDB) [72, 74]. This particular variable annuity contract includes a death benefit that is initially set to the invested capital and can be increased at periodic ratchet dates. Since GMDB contracts often contain a partial withdrawal feature which can be invoked at any time prior to expiry, this problem can be considered as an impulse control problem.

To solve the resulting inequalities in (1.2)–(1.4), we apply an implicit timestepping scheme, based on either a penalty method [44] or an operator splitting scheme [55]. Applied previously to solve simpler variational inequalities [87, 35, 36, 90], the penalty method enables us to solve a complex problem in a straightforward manner by providing us with a robust numerical scheme to value both reload options and variable annuities with a GMDB clause. Assuming that the original problem satisfies a strong comparison result, convergence to the viscosity solution [8] is verified in both cases (see [31, 8] and references therein for more details on viscosity solutions).

While the main goal of this thesis is to show how impulse control problems can be valued with an implicit timestepping method, additional modelling details are also investigated. In Chapter 2, the no-arbitrage pricing model for infinite reload options, based on the classic Black-Scholes approach, is developed. Companies that have issued employee stock options need to include them as an expense in their annual reports [1], and may be looking to reduce the value of these contracts. In particular, infinite reload options are very expensive due to the reload feature. We will show how a specific contract modification can reduce the no-arbitrage price significantly.

In Chapter 3, the original infinite reload pricing model is augmented by including vesting periods. This feature is also shown to reduce the no-arbitrage price of infinite reload options. Since it is well-known that employees place a lower value on stock options

than the actual no-arbitrage price [51, 50, 64, 22], Chapter 4 focuses on valuing infinite reload options from an employee's perspective. A utility-maximizing pricing model is developed and used to estimate the gap between the no-arbitrage price and the value of the contract to its owner.

In Chapter 5, long-term insurance contracts with a guaranteed minimum death benefit (GMDB) clause are considered. Numerical methods are developed to determine the cost of the GMDB clause while including contract features such as partial withdrawals. A realistic fee structure is also included in the pricing model. The GMDB pricing model is then used to determine the *fair insurance charge* which minimizes the contract cost to the issuer. Since variable annuities can be considered as long-term options, the constant volatility assumption associated with the classic Black-Scholes model is insufficient. As such, non-constant market parameters expressed through the concept of regime-switching [52], are used when determining the fair insurance charge for the GMDB clause.

1.1 Contributions

The main contributions of this thesis include:

- In Chapter 2, the increased reload pricing problem is outlined and characterized as an impulse control problem [14], which results in a Hamilton-Jacobi-Bellman variational inequality. In our formulation, the infinite reload pricing problem becomes a special case of the increased reload pricing problem where there is no increase in the reload strike. The discrete problem is formulated using a penalty method.

- In Chapter 2, we show that the discretized Hamilton-Jacobi-Bellman equations satisfy the classic stability, monotonicity and consistency requirements outlined in [8], and hence converge to the viscosity solution [31], assuming a strong comparison result holds. The most challenging requirement turns out to be l_∞-stability. To

demonstrate l_∞-stability, we need to obtain sharper bounds than what is usually obtained with standard maximum analysis.

- In addition, the time discretization of the reload constraint is considered in Chapter 2. When a penalty term is used to impose the reload constraint (implicit method), we demonstrate how applying the reload constraint implicitly, as opposed to explicitly, results in significantly more accurate results. From a financial perspective, we show that making small changes to the reload contract can dramatically lower the no-arbitrage value.

- Vesting periods are added to the pricing model for infinite reload options as outlined in Chapter 3. The addition of vesting periods is shown to reduce the value of infinite reload options and affect the optimal exercise policy.

- In Chapter 4, a multi-step valuation model is developed to determine the value of employee stock options to the contract owner based on a utility maximization approach [64]. The first step involves solving a partial differential equation while the second step makes use of Monte Carlo simulations [85]. Both simple stock options and infinite reload options are valued from an employee's perspective. For infinite reload options, it is shown that the contract value to the employee is about one third the no-arbitrage price and is highly sensitive to assumptions made regarding the employee's portfolio composition.

- A robust pricing model for the GMDB guarantee is presented in Chapter 5. Cast as an impulse control problem, the pricing model includes the partial withdrawal feature and makes use of a regime-switching model [52]. Assuming that the original problem satisfies a strong comparison result, convergence to the viscosity solution is demonstrated by showing the discrete equations satisfy the stability, monotonic-

ity and consistency requirements [8]. The method used to prove l_∞-stability in Chapter 2 is extended to this more complex case.

- In Chapter 5, numerical results for the fair insurance fee charged by the GMDB issuer are presented. We introduce optimal withdrawals and show that typical fees being charged by insurance companies are not enough to fund hedging costs. This contrasts with previous results in the literature [72, 74] which did not take into account partial withdrawals.

Chapter 2

Infinite Reload Options: Pricing and Analysis

Numerous companies have included employee stock options in their executive compensation packages since they are believed to align the executive's interests with those of the share holders. However, in the last few years many large firms have stopped issuing new employee stock options. This change in compensation philosophy may be a direct consequence of the recent changes in accounting requirements regarding employee stock options in the United States. Indeed, the Financial Accounting Standards Board now requires companies issuing stock options to include these contracts as an expense on their balance sheet. Consequently, companies are looking to establish the *fair* or no-arbitrage value of employee stock option contracts currently on their balance sheets using numerical techniques. In addition, companies that have issued more exotic, and often more valuable, stock options may be looking to modify these contracts in order to reduce their no-arbitrage value and thus minimize the expense associated with stock options.

This chapter focuses on valuing a particularly expensive type of employee stock option

referred to as a reload option. These contracts allow the owner to pay the current strike price using a certain amount of pre-owned company stock and, in return, receive new options where the strike price is set to the prevailing stock price. For an infinite reload option, the employee is entitled to take advantage of his reload right as often as he chooses prior to the expiration of the contract. Only limited work has been done regarding the valuation of reload options. In [39], the authors use a binomial model (essentially an explicit finite difference method) to price infinite reload options and outline the optimal exercise policy whereby the owner should exercise his option whenever the stock price exceeds the current strike price. In [61], the author develops analytical pricing formulas for different types of infinite reload options and considers the impact of contract restrictions such as the indivisibility of shares on the contract value. Meanwhile, in [32], the authors outline a binomial pricing model for reload options with both finite and infinite reload opportunities where the reload feature is incorporated using dynamic programming [42].

Companies that have issued infinite reload options may now be looking for ways to reduce their no-arbitrage price [59]. One particular contractual change which has been considered by some companies [59] is to increase the strike price of new options received following a reload event by a certain percentage. We refer to this modified contract as an *increased reload option* and will show how this contract modification can reduce the option expense.

More specifically, we can summarize this chapter's contributions as follows:

- The increased reload pricing problem is outlined and characterized as an impulse control problem [14], which results in a Hamilton-Jacobi-Bellman variational inequality. In our formulation, the infinite reload pricing problem becomes a special case of the increased reload pricing problem where there is no increase in the reload strike.

- In this context, the question of convergence to the viscosity solution must be addressed. Assuming a unique viscosity solution exists, we show that the discretized Hamilton-Jacobi-Bellman equations satisfy the classic stability, monotonicity and consistency requirements outlined in [8].

- Furthermore, the time discretization of the reload constraint is considered. While a penalty term is used to impose the reload constraint (implicit method), we demonstrate how applying the reload constraint implicitly provides more accurate results than applying the constraint explicitly. Note that previous work on reload options involved applying the constraint explicitly [39, 32].

- From a financial perspective, we show that both the option value and the optimal exercise policy are highly sensitive to the percentage increase in the reload strike. Indeed, even a small percentage increase means that it is no longer optimal to exercise whenever the stock price exceeds the strike price.

- Finally, we outline how a local volatility surface can be included in our pricing model for increased reload options.

2.1 Increased Reload Pricing Problem

One of the main goals of this chapter is to investigate how a particular contract modification reduces the no-arbitrage price of infinite reload options. For standard infinite reload options, the following exchange takes place each time a reload occurs: the owner pays the current strike price K using K/S pre-owned company shares (assuming $S > K$) and, in return, receives one unit of company stock and K/S new reload options where the strike price is set to the current stock price $K = S$. We assume that the employee previously owns a sufficient amount of stock that can be used as tender [39]. To clarify,

let us now consider a more concrete example of the reload event. We assume that a given employee owns 10 reload options where the strike price is set to $K = \$100$. When the company stock price reaches $S = \$125$, the employee decides to reload all of his stock options. He will need 8 units of stock to pay the strike price of all 10 reload options ($10 \times K = \$1000$). In exchange, the employee will receive 10 new units of stock (one for each option exercised) and 8 new reload options where the strike price is $K = \$125$.

For increased reload options, the contractual change considered [59] implies that the strike price of new reload options received following a reload event is increased to $K = S \times (1 + p)$, where $p \geq 0$ represents the fraction increase. Note that the classic infinite reload option contract is a special case of the increased reload contract where $p = 0$.

The value of an increased reload option, denoted as $V = V(S, K, t)$, depends on the company stock price S, the option strike price K and time t. We assume that the company stock price S follows geometric Brownian motion [54], namely:

$$\frac{dS}{S} = (\mu - q)dt + \sigma(S, K, t)dZ, \tag{2.1}$$

where μ is the drift rate, $q \geq 0$ is the dividend yield[1], $\sigma(S, K, t)$ is the volatility of the company stock and dZ is the increment of a Wiener process [54]. Notice that the asset volatility in equation (2.1) is written as a function of S, K and t allowing us to model volatility both as a constant and as a function of S, K and t through the use of a local volatility surface [85].

At maturity of the contract ($t = T$), the owner receives one unit of stock for each increased reload option owned, which can then be sold at market value. Hence, the option

[1]While our pricing model assumes a constant dividend yield, discrete dividends could also be included easily.

payoff received by the employee at expiry is:

$$V(S,K,t=T) = \text{Payoff}(S,K) = \max(S-K,0), \tag{2.2}$$

where K is the strike price of the option at expiry and S is the market value of the company stock at expiry.

A reload constraint $\mathcal{A}V = \mathcal{A}V(S,K,t)$ must be imposed to ensure that the current value of the increased reload option is never less than the value obtained by the owner following a reload event. Since the owner of an increased reload option will only consider reloading when $S > K$, the increased reload constraint $\mathcal{A}V$ is defined as:

$$\mathcal{A}V(S,K,t) = \begin{cases} (S-K) + \frac{K}{S}V(S,S(1+p),t) & \text{if } S > K, \\ 0 & \text{otherwise,} \end{cases} \tag{2.3}$$

where $V(S,S(1+p),t)$ is the value of the new reload option obtained with strike $K = S(1+p)$. Note that the infinite reload constraint stated in [32] and [39] is recovered by setting $p = 0$ in equation (2.3).

Furthermore, the constraint in equation (2.3) can be related to the impulse control framework outlined in Chapter 1. In this context, $\mathcal{A}$ is the non-local impulse operator defined in equation (1.1) with only one possible value for the control (denoted as α in equation (1.1)). In equation (2.3), $\frac{K}{S}V(S,S(1+p),t)$ corresponds to the intervention term which modifies the state of the system ($F(J(x,\alpha))$ in equation (1.1)), while $(S-K)$ is the cash flow ($G(x,\alpha)$ in equation (1.1)). No fixed cost is required in equation (2.3) (i.e. $c = 0$ in equation (1.1)) since the operator $\mathcal{A}$ is non-zero only for $S > K$.

Defining the differential operator $\mathcal{L}V$ as:

$$\mathcal{L}V \equiv \frac{\sigma(S,K,\tau)^2 S^2}{2} V_{SS} + (r-q)SV_S - rV, \tag{2.4}$$

where r is the risk-free rate of return, the increased reload pricing problem can be stated as [32, 39]:

$$\min\left(V_\tau - \mathcal{L}V, V - \mathcal{A}V\right) = 0, \tag{2.5}$$

where $\mathcal{A}V$ is defined in equation (2.3) and $\tau = T - t$ is the time to maturity of the contract. The increased reload pricing problem can also be written as a penalized problem:

$$\lim_{\epsilon \to 0}\left(V_\tau - \mathcal{L}V - \frac{1}{\epsilon}\max(\mathcal{A}V - V, 0)\right) = 0, \tag{2.6}$$

which will be the starting point for our numerical methods. The pricing problem in equation (2.6) will be solved numerically using the penalty method outlined in [44]. We will show that as $\epsilon \to 0$, equation (2.6) is consistent with equation (2.5).

When homogeneity properties are satisfied (i.e. when volatility is constant), the increased reload constraint in equation (2.3) can be simplified by using the following property [68].

Definition 2.1 (Homogeneous Option Values). *Option values $V(S, K, t)$ are homogeneous of degree one in both S and K if, for any $\lambda > 0$, the following property holds:*

$$V(\lambda S, \lambda K, t) = \lambda V(S, K, t). \tag{2.7}$$

When equation (2.7) holds, equation (2.3) can be simplified considerably. Setting $K = S(1 + p)$ in equation (2.7), we obtain:

$$V(\lambda S, \lambda S(1 + p), t) = \lambda V(S, S(1 + p), t), \tag{2.8}$$

and now setting $\lambda = \frac{K}{S(1+p)}$, we get:

$$V(S, S(1+p), t) = (1+p)\frac{S}{K}V\left(\frac{K}{1+p}, K, t\right). \tag{2.9}$$

Using equation (2.9), the reload constraint in equation (2.3) becomes:

$$\mathcal{A}V(S, K, t) = \begin{cases} (S-K) + (1+p)V\left(\frac{K}{1+p}, K, t\right) & \text{if } S > K, \\ 0 & \text{otherwise.} \end{cases} \tag{2.10}$$

Therefore, when a similarity reduction can be applied, the constraint in equation (2.10) can be used when solving equation (2.6). We then need only to solve equation (2.6) for a single value of K. Thus, the use of a similarity reduction effectively reduces the solution of a two-dimensional problem in (S, K) to a one-dimensional problem in S [85]. Since this approach can only be applied when volatility is constant, it will be treated as a special case of the general increased reload pricing problem.

2.1.1 Boundary Conditions (No Similarity Reduction)

Theoretically, the increased reload pricing problem in equation (2.6) should be solved on an unbounded two-dimensional domain. In practice, we truncate both the S and K domains and define the solution domain as: $[0, S_{\max}] \times [0, K_{\max}]$, where $S_{\max} \gg K_{\max}$.

To localize the increased reload problem, we specify additional boundary conditions in both the S and K directions. We begin by considering the case where $S \to 0$. When $S = 0$, equation (2.6) simplifies to:

$$V_\tau + rV = 0, \tag{2.11}$$

since $\mathcal{A}V = 0$ at $S = 0$. As $K \to 0$, no additional boundary condition is necessary since

the differential operator $\mathcal{L}$ in equation (2.4) contains no K derivatives.

However, some care must be taken when considering the boundary conditions as $K \to \infty$ and $S \to \infty$. For $S = S_{\max}$, we apply the following boundary condition:

$$V = \max(\text{Payoff}(S, K), \mathcal{A}V). \tag{2.12}$$

Note that the condition in equation (2.12) is an approximation and was chosen since it can easily be shown that $V \geq \text{Payoff}(S, K)$. We expect that the error incurred by applying the approximate boundary condition in (2.12) will be small in the area of interest, assuming $S_{\max}$ is chosen sufficiently large. This will be verified in Section 2.5.

As $K \to K_{\max}$, we could assume that the contract contains a cap, whereby no reload is possible when $K \geq K_{\max}$. In this case, we would solve:

$$V_\tau - \mathcal{L}V = 0 \quad \text{at } K = K_{\max}, \tag{2.13}$$

and the reload constraint in equation (2.3) would become:

$$\mathcal{A}V(S, K, t) = \begin{cases} (S - K) + \frac{K}{S} V(S, S^*(1+p), t) & \text{if } S > K, K < K_{\max}\ , \\ 0 & \text{otherwise,} \end{cases} \tag{2.14}$$

where $S^* = \min\left(S, \frac{K_{\max}}{(1+p)}\right)$. Of course, if equation (2.14) is used, then a similarity reduction is not possible.

Another possibility, and our preferred choice, is to assume that a similarity reduction is valid for $K = K_{\max}$, $S(1+p) > K_{\max}$. In the context of equation (2.4), this implies that $\sigma(S, K, \tau)$ is assumed to be constant as $S \to K_{\max}$. Making this assumption, the solution for $S > K_{\max}/(1+p)$ can be approximated by a similarity solution with little error, provided $K_{\max}$ is sufficiently large.

More precisely, setting $K = K_{\max}$ in equation (2.9), and then using equation (2.3), we obtain the modified reload constraint:

$$\mathcal{A}V(S,K,t) = \begin{cases} (S-K) + \frac{K}{S}V\left(S, S(1+p), t\right) & \text{if } K < S \leq \frac{K_{\max}}{(1+p)}, \\ (S-K) + \frac{K(1+p)}{K_{\max}}V\left(\frac{K_{\max}}{1+p}, K_{\max}, t\right) & \text{if } S > K \text{ and } S(1+p) > K_{\max}, \\ 0 & \text{otherwise.} \end{cases} \tag{2.15}$$

In practice, in order to guarantee the stability of the numerical scheme, we will modify equation (2.15):

$$\mathcal{A}V(S,K,t) = \begin{cases} (S-K) + \frac{K}{S}V\left(S, S(1+p), t\right) & \text{if } S > K \text{ and } S(1+p) \leq K_{\max}, \\ (S-K) + \min\left(1, \frac{K(1+p)}{K_{\max}}\right)V\left(\frac{K_{\max}}{1+p}, K_{\max}, t\right) & \text{if } S > K \text{ and } S(1+p) > K_{\max}, \\ 0 & \text{otherwise.} \end{cases} \tag{2.16}$$

Figure 2.1 depicts how the reload constraint $\mathcal{A}V$ is calculated according to equation (2.16) on the $[0, S_{\max}] \times [0, K_{\max}]$ solution domain.

Remark 2.2. *We remind the reader that equation* (2.15) *for* $S(1+p) > K_{\max}$ *is an artificial boundary condition used to localize the problem and is in general an approximation. Equation* (2.16) *simply replaces one artificial condition by another for* $S(1+p) > K_{\max}$*. Note that since the* K *values of interest satisfy* $K \ll K_{\max}$ *and* $p < 1$*, the difference between equation* (2.16) *and* (2.15) *occurs only for* K *values remote from the areas of interest (i.e.* $K > K_{\max}/2$*). The effect of both approximations should be negligible for* $K_{\max}$ *sufficiently large. This will be verified in some numerical tests in Section 2.5.*

To summarize, we solve the following equation:

$$V_\tau - \mathcal{L}V - \frac{1}{\epsilon}\max(\mathcal{A}V - V, 0) = 0, \tag{2.17}$$

on the domain $[0, S_{\max}] \times [0, K_{\max}]$ with initial conditions:

$$V(S, K, \tau = 0) = \max(S - K, 0), \tag{2.18}$$

and boundary conditions:

$$V_\tau + rV = 0 \quad \text{for } S = 0, \tag{2.19}$$

$$V = \max(\text{Payoff}(S, K), \mathcal{A}V) \quad \text{for } S = S_{\max}, \tag{2.20}$$

where $\mathcal{A}V$ is given by equation (2.16) and $\mathcal{L}$ is defined in equation (2.4). This fully specifies our option pricing problem. Since we truncate the solution domain to finite $[0, S_{\max}] \times [0, K_{\max}]$, we are solving an approximation to the original pricing problem on $[0, \infty] \times [0, \infty]$. We will verify in Section 2.5 through numerical experiments that the localization error due to finite $S_{\max}$, $K_{\max}$ is easily reduced to negligible values.

2.2 Discretization of the Reload Pricing Problem

In this section, we consider the discretization of equation (2.17) on the $[0, S_{\max}] \times [0, K_{\max}]$ domain. The construction of the underlying grid is described in Section 2.2.1, while Section 2.2.2 contains the details of the discrete equations.

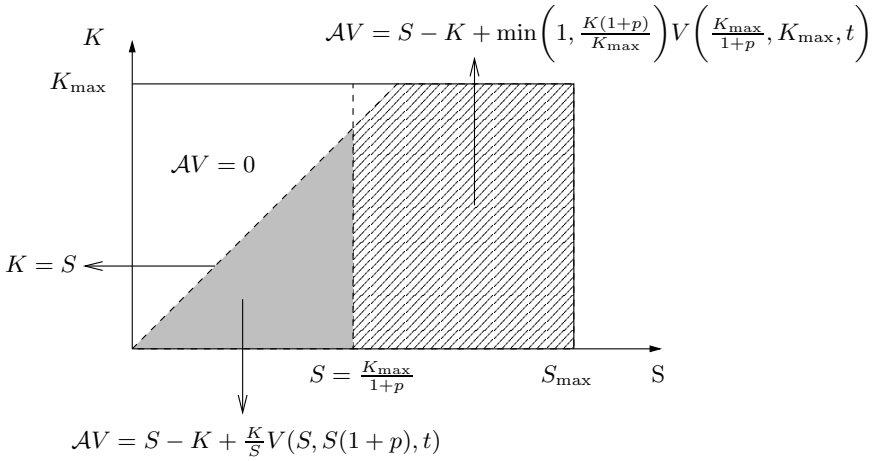


FIGURE 2.1: *Calculation of the reload constraint $\mathcal{A}V$ according to equation* (2.16) *on a typical* $[0, S_{\max}] \times [0, K_{\max}]$ *domain.*

2.2.1 Scaled Grid Construction

In this section, we describe the construction of the underlying grid on the $[0, S_{\max}] \times [0, K_{\max}]$ domain. Since equation (2.4) contains no derivatives with respect to K, we can discretize equation (2.17) using a set of one-dimensional S grids. Let K_{init} be the initial strike price. We build a set of nodes in the K direction $\{K_j\}$, for $j = 0, \ldots, j_{\max}$, such that there exists an index l where $K_l = K_{\text{init}}$ and an index u where $K_u = K_{\text{init}}(1+p)$. Note that the bulk of the K_j nodes are placed around K_{init}.

For each K_j, we construct a set of S grid nodes $\{S_i^j\}$ as follows:

$$\begin{aligned} S_i^j &= \frac{K_j}{K_{\text{init}}} \frac{K_i}{(1+p)} \quad \text{for } i = 0, \ldots, j_{\max} - 1, \\ S_{j_{\max}}^j &= \frac{K_{j_{\max}}}{K_{\text{init}}} \frac{K_{j_{\max}}}{(1+p)}. \end{aligned} \tag{2.21}$$

This guarantees that for any j, the nodes $\left(\frac{K_j}{1+p}, K_j\right)$ and (K_j, K_j) are included in the grid. As shown in Figure 2.2, this *scaled grid* construction concentrates nodes near the line $K = (1+p)S$ so that the constraint in equation (2.16) can be computed accurately.

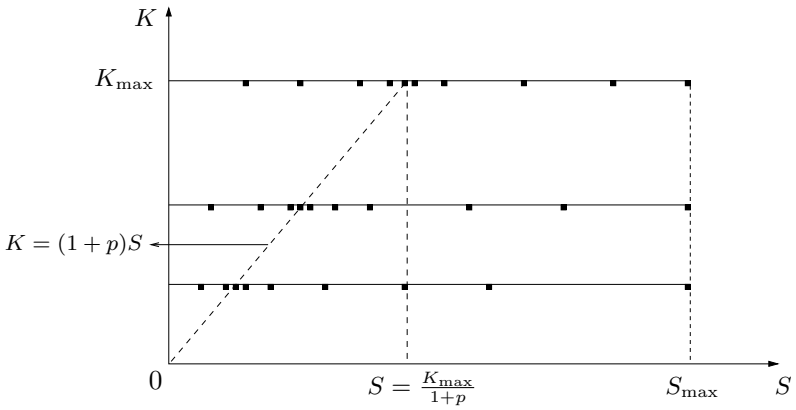


FIGURE 2.2: *Example of a scaled grid construction for the two-dimensional* $[0, S_{\max}] \times [0, K_{\max}]$ *domain. Nodes are concentrated around the diagonal* $K = (1+p)S$ *line.*

We remark that this type of grid generally requires interpolation to estimate $\mathcal{A}V$.

The scaled grid construction contrasts with the simple idea of defining:

$$S_i^j = K_i \quad \text{for } i = 0, \ldots, j_{\max}, \tag{2.22}$$

for a given K_j which results in the so-called *repeated grids* discussed in [90] and [87]. No interpolation is required to estimate $\mathcal{A}V$ (when $p = 0$) on a repeated grid. However, tests in [87] show that a scaled grid, along with diagonal interpolation (to be discussed later) is superior to a repeated grid.

2.2.2 Discrete Equations

In this section, we derive the discrete equations for the increased reload pricing problem. To outline the dependence of the option value on S, K and τ, the following notation is used for the discrete option value: $V_{i,j}^n = V(S_i^j, K_j, \tau^n)$. Similarly, discrete operators will be denoted as $\mathcal{L}^h$ and $\mathcal{A}^h$ where the superscript h represents the space discretization

parameter. The discrete form of equation (2.17) is obtained by using finite difference approximations and introducing a discrete penalty term $P(V_{i,j}^{n+1}, \mathcal{A}^h V_{i,j}^{n+1})$:

$$\frac{V_{i,j}^{n+1} - V_{i,j}^{n}}{\Delta\tau} = (1-\theta)[\mathcal{L}^h V]_{i,j}^{n+1} + \theta[\mathcal{L}^h V]_{i,j}^{n} + P(V_{i,j}^{n+1}, \mathcal{A}^h V_{i,j}^{n+1}), \tag{2.23}$$

where $\Delta\tau = T/N$, $\mathcal{A}^h V_{i,j}^{n+1} = \mathcal{A}^h V(S_i^j, K_j, \tau^{n+1})$ is the discrete reload constraint and $0 \leq \theta \leq 1$ indicates the timestepping method used ($\theta = 0$ implies that a fully implicit method is chosen, while $\theta = 1/2$ implies that Crank-Nicolson timestepping is used [54]). The discrete differential operator $\mathcal{L}^h$ is defined as:

$$[\mathcal{L}^h V]_{i,j}^{n} = \alpha_{i,j}^{n} V_{i-1,j}^{n} + \beta_{i,j}^{n} V_{i+1,j}^{n} - (\alpha_{i,j}^{n} + \beta_{i,j}^{n} + r) V_{i,j}^{n}, \tag{2.24}$$

where $\alpha_{i,j}^{n}$ and $\beta_{i,j}^{n}$ are determined according to the algorithm in Appendix A, and satisfy:

$$\alpha_{i,j}^{n} \geq 0 \; ; \; \beta_{i,j}^{n} \geq 0 \quad \forall i, j, n. \tag{2.25}$$

The discrete penalty term $P(V_{i,j}^{n+1}, \mathcal{A}^h V_{i,j}^{n+1})$ is defined as:

$$P(V_{i,j}^{n+1}, \mathcal{A}^h V_{i,j}^{n+1}) = L_{i,j}^{n+1} \left[\mathcal{A}^h V_{i,j}^{n+1} - V_{i,j}^{n+1} \right], \tag{2.26}$$

where

$$L_{i,j}^{n+1} = L(V_{i,j}^{n+1}, \mathcal{A}^h V_{i,j}^{n+1}) = \begin{cases} \frac{1}{\epsilon} & \text{if } \mathcal{A}^h V_{i,j}^{n+1} > V_{i,j}^{n+1}, \\ 0 & \text{otherwise.} \end{cases} \tag{2.27}$$

Similarly, the boundary condition (2.12) in penalized form for $i = j_{\max}$ is:

$$V_{j_{\max},j}^{n+1} = \text{Payoff}(S_{j_{\max}}^{j}, K_j) + P(V_{j_{\max},j}^{n+1}, \mathcal{A}^h V_{j_{\max},j}^{n+1}), \tag{2.28}$$

where $\text{Payoff}(S_i^j, K_j) = \max(S_i^j - K_j, 0)$ in accordance with equation (2.2).

Writing the penalty term as a control term, equation (2.26) becomes:

$$P(V_{i,j}^{n+1}, \mathcal{A}^h V_{i,j}^{n+1}) = \max_{\gamma \in \{0,1\}} \frac{\gamma}{\epsilon}\left[\mathcal{A}^h V_{i,j}^{n+1} - V_{i,j}^{n+1}\right]. \tag{2.29}$$

The control formulation in equation (2.29), is sometimes useful for carrying out analysis of the discrete equations.

When computing the constraint $\mathcal{A}^h V_{i,j}^{n+1}$ in equation (2.26), we use diagonal interpolation along the $K = S(1+p)$ line to determine $V(S, S(1+p), \tau)$ in equation (2.16). For each (i,j) pair, having determined m such that $K_m \le S_i^j(1+p) \le K_{m+1}$, diagonal interpolation can be written as:

$$\begin{aligned} V(S_i^j, S_i^j(1+p), \tau^{n+1}) =& V\left(\frac{K_m}{1+p}, K_m, \tau^{n+1}\right)\left(1 - \frac{S_i^j(1+p) - K_m}{K_{m+1} - K_m}\right) \\ &+ V\left(\frac{K_{m+1}}{(1+p)}, K_{m+1}, \tau^{n+1}\right)\frac{S_i^j(1+p) - K_m}{K_{m+1} - K_m} \\ &+ \mathcal{O}((K_{m+1} - K_m)^2). \end{aligned} \tag{2.30}$$

Tests in [87] show that diagonal interpolation for shout options is superior to the usual bilinear interpolation. Indeed, since a similarity reduction can be used in the increased reload pricing model, the option value is linear along the line $K = S(1+p)$ implying that diagonal interpolation will be exact when used in the context of equation (2.16) [87]. Defining the interpolation weight $0 \le \omega \le 1$ as:

$$\omega = \frac{S_i^j(1+p) - K_m}{K_{m+1} - K_m}, \tag{2.31}$$

equation (2.30) can be written as:

$$V(S_i^j, S_i^j(1+p), \tau^{n+1}) \simeq V_{l,m}^{n+1}(1-\omega) + V_{l,m+1}^{n+1}\omega, \tag{2.32}$$

where l is an index such that $S_l^m = K_m/(1+p)$, $S_l^{m+1} = K_{m+1}/(1+p)$ (see equation (2.21)) and $V_{l,m}^{n+1} = V(S_l^m, K_m, \tau^{n+1})$. Figure 2.3 shows a graphical representation of diagonal interpolation along the $K = S(1+p)$ line.

Consequently, the discrete penalty term can be written as:

$$P(V_{i,j}^{n+1}, \mathcal{A}^h V_{i,j}^{n+1}) = \begin{cases} \frac{1}{\epsilon}\left[\mathcal{A}^h V_{i,j}^{n+1} - V_{i,j}^{n+1}\right] & \text{when } \mathcal{A}^h V_{i,j}^{n+1} > V_{i,j}^{n+1} \text{ and } S_i^j > K_j, \\ 0 & \text{otherwise,} \end{cases} \tag{2.33}$$

where

$$\mathcal{A}^h V_{i,j}^{n+1} = S_i^j - K_j + \frac{K_j}{S_i^j}\left((1-\omega)V_{l,m}^{n+1} + \omega V_{l,m+1}^{n+1}\right). \tag{2.34}$$

For those nodes where a similarity reduction is applied, such as in equation (2.28) at node $i = j_{\max}$, the discrete penalty term $P(V_{i,j}^{n+1}, \mathcal{A}^h V_{i,j}^{n+1})$ will still be of the general form presented in equation (2.33) but $\mathcal{A}^h V_{i,j}^{n+1}$ will be (from equations (2.10) and (2.16)):

$$\mathcal{A}^h V_{i,j}^{n+1} = S_i^j - K_j + \min\left(1, \frac{K_j(1+p)}{K_{j_{\max}}}\right) V_{l,j_{\max}}^{n+1}. \tag{2.35}$$

No interpolation is required in this case since the node $(S_l^{j_{\max}}, K_{j_{\max}})$ is included in our initial grid (see equation (2.21)).

The numerical scheme in equation (2.23) is a positive coefficient discretization [43] as defined below.

Definition 2.1 (Positive Coefficient Scheme). *The numerical scheme in equation* (2.23)

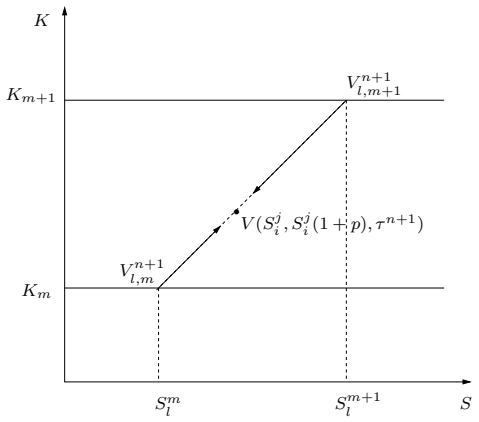


FIGURE 2.3: *Diagonal interpolation is used when determining the value of new reload options* $V(S_i^j, S_i^j(1+p), \tau^{n+1})$ *in the reload constraint in equation* (2.16).

is a positive coefficient discretization when:

$$\alpha_{i,j}^{n+1} \geq 0, \qquad \beta_{i,j}^{n+1} \geq 0, \qquad r \geq 0, \qquad \forall i, j, n. \tag{2.36}$$

Since $\alpha_{i,j}^{n+1}$ and $\beta_{i,j}^{n+1}$ satisfy equation (2.25) (see Appendix A) and $r > 0$ for all problems considered, the numerical scheme in equation (2.23) is a positive coefficient discretization.

2.3 Convergence to the Viscosity Solution

In [81], the authors demonstrate how some reasonable discretization schemes either never converge or converge to a wrong solution. Hence, it is important to verify that our discretization method converges to the unique viscosity solution [31], which corresponds to the financially relevant solution. In this section, we show that the discrete equations

in (2.23) satisfy the stability, monotonicity and consistency requirements which generally ensure convergence of the numerical solution to the unique viscosity solution [11, 9, 8].

A unique, continuous viscosity solution exists if the pricing problem in equation (2.5) satisfies a strong comparison property [8]. Intuitively, the strong comparison principle implies that when the payoff of a first option is larger than the payoff of a second option, the same relation holds for the value of the respective contracts at all times [8]. In a financial context, this must hold to preclude arbitrage opportunities. While such a result exists for many first and second order equations, our increased reload pricing problem differs due to the non-local character of the reload constraint $\mathcal{A}^h V_{i,j}^{n+1}$ defined in equations (2.34) and (2.35). However, the authors of [84], [24], [4] and [57] show that a strong comparison principle holds for similar (but not identical) impulse control problems.

Assumption 2.1. *We assume that a unique, continuous viscosity solution exists [8, 31] for the localized reload pricing problem in equation* (2.17) *with initial conditions outlined in equation* (2.18) *and localization conditions in equations* (2.16), (2.19) *and* (2.20).

2.3.1 Stability

In this section, we demonstrate that the discrete equations in (2.23) satisfy the l_∞-stability requirement which involves showing that the discrete option value $V_{i,j}^{n+1}$ is bounded. We define the vector V_j^{n+1} as:

$$V_j^{n+1} = \begin{bmatrix} V_{0,j}^{n+1} \\ V_{1,j}^{n+1} \\ \vdots \\ V_{j_{\max},j}^{n+1} \end{bmatrix}, \tag{2.37}$$

and further define:

$$\Delta S^j_{\max} = \max_i (S^j_{i+1} - S^j_i),\ \Delta K_{\max} = \max_j (K_{j+1} - K_j).$$

Definition 2.2 (Stability). *For a fixed $S_{\max}$, $K_{\max}$ and T, the discretization in equation* (2.23) *is l_∞-stable if*

$$||V^{n+1}_j||_\infty < C, \tag{2.38}$$

for $0 \leq n \leq N$, as $\Delta\tau \to 0$, $\Delta S^j_{\max} \to 0$, $\Delta K_{\max} \to 0$ and $\epsilon \to 0$, where C is a constant independent of $\Delta\tau$, $\Delta S^j_{\max}$, $\Delta K_{\max}$ and ϵ.

Assumption 2.3. *We assume that $\Delta K_{\max}$, $\Delta S^j_{\max}$, $\Delta\tau$ and ϵ are parametrized as:*

$$\Delta K_{\max} = c_1 h,\ \Delta S^j_{\max} = c_2 h,\ \Delta\tau = c_3 h,\ \epsilon = c_4 h, \tag{2.39}$$

where c_1, c_2, c_3 and c_4 are constants.

The stability of the discrete scheme in (2.23) is a consequence of the following Theorem.

Theorem 2.4 (Bound for $V^n_{i,j}$). *Assuming that the numerical scheme satisfies Definition 2.1, that the boundary conditions are applied as outlined in Section 2.1.1 and that the initial conditions are given by the discrete version of equation* (2.18)*, the value of the increased reload option contract satisfies:*

$$0 \leq V^n_{i,j} \leq S^j_i \quad \forall i, j, n, \tag{2.40}$$

in the case of fully implicit timestepping ($\theta = 0$).

Proof. To prove Theorem 2.4, we write the discrete equations for $V_{i,j}^{n+1}$ in matrix form:

$$\mathcal{Q}^{n+1}(V^{n+1})V^{n+1} = V^n + \text{ other terms}, \tag{2.41}$$

where the coefficient matrix $\mathcal{Q}^{n+1}(V^{n+1})$ is an M-matrix. Having shown that the terms on the right-hand side of equation (2.41) are positive, we use the properties of $\mathcal{Q}^{n+1}(V^{n+1})$ to show that $V_{i,j}^{n+1} \geq 0$. We then apply the coefficient matrix $\mathcal{Q}^{n+1}(V^{n+1})$ to $(S_i^j - V_{i,j}^{n+1})$. Using the same approach (positive right-hand terms and properties of $\mathcal{Q}^{n+1}(V^{n+1})$), we show that $S_i^j - V_{i,j}^{n+1} \geq 0$. The complete proof of Theorem 2.4 is included as Appendix B. □

Since $S_i^j \leq S_{\max}$, the numerical scheme in equation (2.23) satisfies the stability requirement in Definition 2.2, assuming fully implicit timestepping is used. We can extend the above analysis when Crank-Nicolson is used ($\theta = 1/2$ in equation (2.23)) to show that Crank-Nicolson timestepping is l_∞-stable if the following timestepping condition is satisfied:

$$\Delta\tau \leq \frac{2}{\alpha_{i,j}^n + \beta_{i,j}^n + r} \quad \forall i, j, n. \tag{2.42}$$

2.3.2 Monotonicity

In this section, we show that the numerical scheme in equation (2.23) is monotone. We denote the discrete equations in equation (2.23) at interior nodes ($i < j_{\max}$) as:

$$\begin{aligned} g_{i,j}(h, x, V_{i,j}^{n+1}, \{V_{k,j}^{n+1}\}_{k\neq i}, \{V_{i,j}^n\}, V_{l,m}^{n+1}, V_{l,m+1}^{n+1}) & \\ = & V_{i,j}^{n+1} - V_{i,j}^n - (1-\theta)\Delta\tau[\mathcal{L}^h V]_{i,j}^{n+1} - \theta\Delta\tau[\mathcal{L}^h V]_{i,j}^n \\ & - \Delta\tau L(V_{i,j}^{n+1}, \mathcal{A}^h V_{i,j}^{n+1})\left[\mathcal{A}^h V_{i,j}^{n+1} - V_{i,j}^{n+1}\right] \\ = & 0, \end{aligned} \tag{2.43}$$

and at boundary nodes ($i = j_{\max}$) as:

$$\begin{aligned} g_{j_{\max},j}(h, x, V^{n+1}_{j_{\max},j}, \{V^{n+1}_{k,j}\}_{k\neq j_{\max}}, \{V^{n}_{j_{\max},j}\}, V^{n+1}_{l,m}, V^{n+1}_{l,m+1}) \\ = V^{n+1}_{j_{\max},j} - \text{Payoff}(S^{j}_{j_{\max}}, K_j) - L(V^{n+1}_{i,j}, \mathcal{A}^h V^{n+1}_{i,j})\left[\mathcal{A}^h V^{n+1}_{i,j} - V^{n+1}_{i,j}\right] \\ = 0, \end{aligned} \tag{2.44}$$

where $x = (S^j_i, K_j, \tau^{n+1})$, $0 \leq \theta \leq 1$, $[\mathcal{L}^h V]^{n+1}_{i,j}$ is defined in equation (2.24) and $\mathcal{A}^h V^{n+1}_{i,j}$ is defined as in equation (2.34) for $i < j_{\max}$ and as in equation (2.35) for $i = j_{\max}$.

Definition 2.5 (Monotonicity). *The numerical scheme* $g_{i,j}(h, x, V^{n+1}_{i,j}, \{V^{n+1}_{k,j}\}_{k\neq i}, V^{n}_{i,j}, V^{n+1}_{l,m}, V^{n+1}_{l,m+1})$ *in equations* (2.43) *and* (2.44) *is monotone if, for all* $\epsilon^n_{i,j} \geq 0$*:*

$$\begin{aligned} g_{i,j}(h, x, V^{n+1}_{i,j}, \{V^{n+1}_{k,j} + \epsilon^{n+1}_{k,j}\}_{k\neq i}, \{V^{n}_{i,j} + \epsilon^{n}_{i,j}\}, V^{n+1}_{l,m} + \epsilon^{n+1}_{l,m}, V^{n+1}_{l,m+1} + \epsilon^{n+1}_{l,m+1}) \\ - g_{i,j}(h, x, V^{n+1}_{i,j}, \{V^{n+1}_{k,j}\}_{k\neq i}, V^{n}_{i,j}, V^{n+1}_{l,m}, V^{n+1}_{l,m+1}) \leq 0. \end{aligned} \tag{2.45}$$

Note that this definition of monotonicity is equivalent to that presented in [8].

Theorem 2.6 (Monotone Discretization). *Assuming the discretization satisfies Definition 2.1, the numerical scheme* $g_{i,j}(h, x, V^{n+1}_{i,j}, \{V^{n+1}_{k,j}\}_{k\neq i}, \{V^{n}_{i,j}\}, V^{n+1}_{l,m}, V^{n+1}_{l,m+1})$ *defined in equations* (2.43) *and* (2.44)*, is monotone.*

Proof. According to Definition 2.1, the scheme presented in equation (2.23) is a positive coefficient scheme. Also, recall that the discrete increased reload pricing problem can be defined as a control problem (see equation (2.29)). In [43], the authors show that the discretization of an optimal control problem with positive coefficients is also monotone. Using the same technique as in [43], it is straightforward to show that the numerical scheme in equations (2.43) and (2.44) is monotone and satisfies Definition 2.5. □

2.3.3 Consistency

To facilitate the definition of consistency, we outline some notational details. For the increased reload pricing problem, the impulse control problem can be written as:

$$F(V(x)) = 0 \text{ for all } x = (S, K, \tau), \tag{2.46}$$

where

$$F(V(x)) = \begin{cases} F_{in}(V(x)) & \text{if } S < S_{\max}, \\ F_{bound}(V(x)) & \text{if } S = S_{\max}. \end{cases} \tag{2.47}$$

The continuous problem evaluated at discrete interior nodes when $S_i^j < S_{\max}$ can be written as:

$$F_{in}(V)_{i,j}^{n+1} = \left[\min(V_\tau - \mathcal{L}V, V - \mathcal{A}V)\right]_{i,j}^{n+1} = 0, \tag{2.48}$$

where the continuous operator $\mathcal{L}$ is defined in equation (2.4). At the boundary nodes where $S_i^j = S_{\max}$, we have:

$$F_{bound}(V)_{j_{\max},j}^{n+1} = \left[V - \max(\text{Payoff}(S, K), \mathcal{A}V)\right]_{j_{\max},j}^{n+1} = 0. \tag{2.49}$$

Since $\epsilon > 0$, the numerical scheme at interior nodes ($i < j_{\max}$) in equation (2.43) can be rewritten as:

$$\begin{aligned} &\hat{g}_{i,j}(h, x, V_{i,j}^{n+1}, \{V_{k,j}^{n+1}\}_{k\neq i}, \{V_{i,j}^n\}, V_{l,m}^{n+1}, V_{l,m+1}^{n+1}) = \\ &\min\left(\epsilon\left(\frac{V_{i,j}^{n+1} - V_{i,j}^n}{\Delta\tau} - [\mathcal{L}^h V]_{i,j}^{n+1}\right) + V_{i,j}^{n+1} - \mathcal{A}^h V_{i,j}^{n+1}, \frac{V_{i,j}^{n+1} - V_{i,j}^n}{\Delta\tau} - [\mathcal{L}^h V]_{i,j}^{n+1}\right) = 0, \end{aligned} \tag{2.50}$$

while equation (2.44) for $i = j_{\max}$ can be rewritten as:

$$\hat{g}_{j_{\max},j}(h, x, V^{n+1}_{j_{\max},j}, \{V^{n+1}_{k,j}\}_{k\neq j_{\max}}, \{V^{n}_{j_{\max},j}\}, V^{n+1}_{l,m}, V^{n+1}_{l,m+1}) = \tag{2.51}$$
$$\min\left(\epsilon\left(V^{n+1}_{j_{\max},j} - \text{Payoff}(S^j_{j_{\max}}, K_j)\right) + V^{n+1}_{j_{\max},j} - \mathcal{A}^h V^{n+1}_{j_{\max},j}, V^{n+1}_{j_{\max},j} - \text{Payoff}(S^j_{j_{\max}}, K_j)\right) = 0.$$

To formally define the notion of consistency, we require the concept of upper and lower semi-continuous envelope of a function.

Definition 2.7. *Assume we have a function $f : C \to \mathbb{R}$ where C is a topological space. Then the upper semi-continuous and lower semi-continuous envelopes of f are defined as:*

$$f^*(y) = \limsup_{\substack{x\to y\\ y\in C}} f(x) \quad \textit{and} \quad f_*(y) = \liminf_{\substack{x\to y\\ y\in C}} f(x). \tag{2.52}$$

Definition 2.8 (Consistency). *For any smooth test function ϕ with bounded derivatives of all orders with respect to S and τ, the scheme $\hat{g}_{i,j}(h, x, \phi^{n+1}_{i,j}, \{\phi^{n+1}_{k,j}\}_{k\neq i}, \{\phi^{n}_{i,j}\}, \phi^{n+1}_{l,m}, \phi^{n+1}_{l,m+1})$ is consistent if, for all points in the domain $\hat{x} = (\hat{S}, \hat{K}, \hat{\tau})$ with $x = (S^j_i, K_j, \tau^{n+1})$, we have:*

$$\limsup_{\substack{h,\xi\to 0\\ x\to\hat{x}}} \hat{g}_{i,j}(h, x, \phi^{n+1}_{i,j} + \xi, \{\phi^{n+1}_{k,j} + \xi\}_{k\neq i}, \{\phi^{n}_{i,j} + \xi\}, \phi^{n+1}_{l,m} + \xi, \phi^{n+1}_{l,m+1} + \xi) \leq F^*(\phi(\hat{x})) \tag{2.53}$$

$$\liminf_{\substack{h,x,\xi\to 0\\ x\to\hat{x}}} \hat{g}_{i,j}(h, x, \phi^{n+1}_{i,j} + \xi, \{\phi^{n+1}_{k,j} + \xi\}_{k\neq i}, \{\phi^{n}_{i,j} + \xi\}, \phi^{n+1}_{l,m} + \xi, \phi^{n+1}_{l,m+1} + \xi) \geq F_*(\phi(\hat{x})). \tag{2.54}$$

where $\phi^n_{i,j} = \phi(S^j_i, K_j, \tau^n)$ and $\xi \geq 0$.

Remark 2.9 (Continuous Scheme). *When the numerical scheme is continuous over*

the entire domain (both interior nodes and boundary), the conditions in equations (2.53) *and* (2.54) *reduce to:*

$$\lim_{h\to 0}\left|F(\phi)_{i,j}^{n+1} - \hat{g}_{i,j}(h, x, \phi_{i,j}^{n+1}, \{\phi_{k,j}^{n+1}\}_{k\neq i}, \{\phi_{i,j}^{n}\}, \phi_{l,m}^{n+1}, \phi_{l,m+1}^{n+1})\right| = 0. \tag{2.55}$$

Equation (2.55) *is the typical formulation used when verifying consistency of a numerical scheme and applies, for example, to cases where the equation on the boundary is obtained by taking the limit of the equation on the interior nodes. Unfortunately, this is not the case for our increased reload pricing model which is why the consistency requirements are outlined as in equations* (2.53) *and* (2.54)*.*

Theorem 2.10 (Consistent Discretization). *The numerical scheme in equation* (2.23) *is consistent according to Definition 2.8.*

Proof. See Appendix C. □

Having shown that the discrete equations in (2.23) are monotone, stable and consistent, we can state the following theorem.

Theorem 2.11 (Convergence to the Viscosity Solution). *Assuming a unique viscosity solution exists (i.e. Assumption 2.1 is satisfied), the numerical scheme in equation* (2.23)*, with boundary conditions corresponding to the discrete version of equations* (2.19) *and* (2.20) *and initial conditions outlined in the discrete version of equation* (2.18)*, converges to the unique viscosity solution of the localized problem in equation* (2.5)*, with boundary conditions in* (2.16)*,* (2.19) *and* (2.20)*.*

2.4 Solution Algorithm

In this section, we specify algorithmic details about the solution process. Recall that at each timestep, we solve a set of one-dimensional problems each with a different strike

value K_j. More specifically, when the reload constraint is applied implicitly, the following equation is solved at each timestep (assuming fully implicit timestepping is used):

$$\mathcal{B}_j^{n+1} V_j^{n+1} = V_j^n + E_j^{n+1} \mathcal{A}^h V_j^{n+1}, \tag{2.56}$$

where the vectors V_j^n, V_j^{n+1} are defined in equation (2.37) and

$$E_j^{n+1} = \begin{bmatrix} \Delta\tau L_{0,j}^{n+1} \\ \Delta\tau L_{1,j}^{n+1} \\ \vdots \\ \Delta\tau L_{j_{\max}-1,j}^{n+1} \\ L_{j_{\max},j}^{n+1} \end{bmatrix} ; \mathcal{A}^h V_j^{n+1} = \begin{bmatrix} \mathcal{A}^h V_{0,j}^{n+1} \\ \mathcal{A}^h V_{1,j}^{n+1} \\ \vdots \\ \mathcal{A}^h V_{j_{\max}-1,j}^{n+1} \\ \mathcal{A}^h V_{j_{\max},j}^{n+1} \end{bmatrix}, \tag{2.57}$$

with $L_{i,j}^{n+1} = L(V_{i,j}^{n+1}, \mathcal{A}^h V_{i,j}^{n+1})$ defined in equation (2.27) and $\mathcal{A}^h V_{i,j}^{n+1}$ defined in equation (2.34) (or (2.35) in the similarity reduction case). The matrix $\mathcal{B}_j^{n+1}$ is constructed such that:

$$\begin{aligned} [\mathcal{B}_j^{n+1} V_j^{n+1}]_i = V_{i,j}^{n+1}(1 + \Delta\tau(\alpha_{i,j}^{n+1} + \beta_{i,j}^{n+1} + r + L_{i,j}^{n+1})) \\ - \Delta\tau\alpha_{i,j}^{n+1} V_{i-1,j}^{n+1} - \Delta\tau\beta_{i,j}^{n+1} V_{i+1,j}^{n+1}\,, \end{aligned} \tag{2.58}$$

for rows where the diagonal interpolation does not involve the current pricing problem, namely when $S_i^j(1+p) > K_{j+1}$.

For some points S_i^j near K_j, diagonal interpolation defined in equation (2.34) involves data from the current pricing problem. When $K_j \le S_i^j(1+p) \le K_{j+1}$, diagonal interpolation will use $V_{l,j}^{n+1}$ and $V_{l,j+1}^{n+1}$ to determine $V(S_i^j, S_i^j(1+p), \tau^{n+1})$ as in equation (2.32). Hence, the rows in $\mathcal{B}_j^{n+1}$ corresponding to these points are redefined since

for $j = 0, 1, \ldots, j_{\max}$ **do**
  for $i = 0, 1, \ldots, j_{\max}$ **do**
    $V^0_{i,j} = \text{Payoff}(S^j_i, K_j)$
  end for
end for
for $n = 0, 1, \ldots, N$ **do**
  for $j = j_{\max}, j_{\max} - 1, \ldots, 0$ **do**
    Solve: $\mathcal{B}^{n+1}_j V^{n+1}_j = V^n_j + E^{n+1}_j \mathcal{A}^h V^{n+1}_j$
  end for
end for

ALGORITHM 2.1: *Implicit Application of the Reload Constraint*

they now contain an extra entry:

$$[\mathcal{B}^{n+1}_j V^{n+1}_j]_i = V^{n+1}_{i,j}(1 + \Delta\tau(\alpha^{n+1}_{i,j} + \beta^{n+1}_{i,j} + r + L^{n+1}_{i,j})) - \Delta\tau\alpha^{n+1}_{i,j} V^{n+1}_{i-1,j} - \Delta\tau\beta^{n+1}_{i,j} V^{n+1}_{i+1,j} - \Delta\tau L^{n+1}_{i,j} \frac{K_j}{S^j_i}(1-\omega) V^{n+1}_{l,j}. \tag{2.59}$$

Similarly, we redefine $\mathcal{A}^h V^{n+1}_{i,j}$ for these points as:

$$\mathcal{A}^h V^{n+1}_{i,j} = S^j_i - K_j + \frac{K_j}{S^j_i}\omega V^{n+1}_{l,j+1}\,. \tag{2.60}$$

Since interpolation is used when calculating the entries in $\mathcal{A}^h V^{n+1}_j$, their value will generally depend on the solution from pricing problems with higher strike values: V^{n+1}_m where $K_m > K_j$. Hence, we need to solve each of the V^{n+1}_m problems first before proceeding to compute V^{n+1}_j. Consequently, we will solve the pricing problems in a specific order namely with decreasing strike (i.e. from $j = j_{\max}$ to $j = 0$). The detailed solution method is presented as Algorithm 2.1.

Having noted that equation (2.56) is non-linear, non-linear iteration is used to deter-

$\bar{V}_j^0 = V_j^n$
for $k = 0, 1, \ldots,$ until convergence **do**
  Solve: $\bar{\mathcal{B}}_j^k \bar{V}_j^{k+1} = V_j^n + \bar{E}_j^k \mathcal{A}^h \bar{V}_j^k$
  if $\max_i \left[\frac{|\bar{V}_{i,j}^{k+1} - \bar{V}_{i,j}^k|}{\max(1, |V_{i,j}^k|)} \right] < tol$ **then**
    Stop iteration - Exit For loop
  end if
end for
$V_j^{n+1} = \bar{V}_j^{k+1}$

ALGORITHM 2.2: *Non-Linear Iteration*

mine V_j^{n+1} for each j value. We denote the kth estimate for V_j^{n+1} as $(V_j^{n+1})^k = \bar{V}_j^k$. Similarly, we define $(E_j^{n+1})^k = \bar{E}_j^k$ and $(\mathcal{B}_j^{n+1})^k = \bar{\mathcal{B}}_j^k$. Algorithm 2.2 outlines the iteration algorithm to determine V_j^{n+1} for a given j value. Note that the convergence tolerance in Algorithm 2.2, denoted by tol, is chosen sufficiently small, i.e. $tol \ll 1$. Also, we notice that $\bar{\mathcal{B}}_j^k$ is not a tri-diagonal matrix since some rows contain an extra non-zero entry outside the usual tri-diagonal envelope. Nonetheless, $\bar{\mathcal{B}}_j^k \bar{V}_j^{k+1} = V_j^n + \bar{E}_j^{k+1} \mathcal{A}^h \bar{V}_j^k$ can be easily solved using a direct sparse solver.

Theorem 2.1 (Convergence of Non-linear Iteration). *Since the matrix $\bar{\mathcal{B}}_j^k$ satisfies all the properties of an M-matrix, the non-linear iteration process in Algorithm 2.2 converges to the unique solution of equation* (2.56).

Proof. Proof of Theorem 2.1 follows from [43]. Recall that we can write the solution step in Algorithm 2.2 as a control problem. Hence, a straightforward maximum analysis can be used to bound $\bar{V}_j^k$, independent of k. Since $\bar{\mathcal{B}}_j^k$ is an M-matrix, the iterates form a bounded non-decreasing sequence. Thus, convergence to the unique solution of equation (2.56) can be shown using the same technique as in [43]. □

Alternatively, the reload constraint in equation (2.16) can be applied explicitly. In this case, an intermediate solution value $\hat{V}_j^{n+1}$ is determined by solving the equation:

$$\hat{\mathcal{B}}_j^{n+1}\hat{V}_j^{n+1} = V_j^n, \tag{2.61}$$

where all rows in $\hat{\mathcal{B}}_j^{n+1}$ are defined such that:

$$\begin{aligned}[\hat{\mathcal{B}}_j^{n+1}\hat{V}_j^{n+1}]_i = \hat{V}_{i,j}^{n+1}(1 + \Delta\tau(\alpha_{i,j}^{n+1} + \beta_{i,j}^{n+1} + r)) \\ - \Delta\tau\alpha_{i,j}^{n+1}\hat{V}_{i-1,j}^{n+1} - \Delta\tau\beta_{i,j}^{n+1}\hat{V}_{i+1,j}^{n+1}\,.\end{aligned} \tag{2.62}$$

Note that the matrix $\hat{\mathcal{B}}_j^{n+1}$ is a tri-diagonal matrix and the resulting system is now linear. As such, no iteration is required when solving equation (2.61).

Once $\hat{V}_j^{n+1}$ has been determined, the reload constraint is applied explicitly for each i:

$$V_{i,j}^{n+1} = \max(\mathcal{A}^h V_{i,j}^{n+1}, \hat{V}_{i,j}^{n+1})\,, \tag{2.63}$$

where $\mathcal{A}^h V_{i,j}^{n+1}$ is defined as:

$$\mathcal{A}^h V_{i,j}^{n+1} = S_i^j - K_j + \frac{K_j}{S_i^j}\left[(1-\omega)\bar{V}_{l,m}^{n+1} + \omega V_{l,m+1}^{n+1}\right], \tag{2.64}$$

where $\bar{V}_{i,j}^{n+1}$ is computed using the most recent information available (i.e. we solve for j in decreasing order), which implies:

$$\bar{V}_{l,m}^{n+1} = \begin{cases} V_{l,m}^{n+1} & \text{if } m \neq j, \\ \hat{V}_{l,m}^{n+1} & \text{if } m = j. \end{cases} \tag{2.65}$$

For those cases when a similarity reduction is used, $\mathcal{A}^h V_{i,j}^{n+1}$ is defined as:

$$\mathcal{A}^h V_{i,j}^{n+1} = S_i^j - K_j + \min\left(1, \frac{K_j(1+p)}{K_{j_{\max}}}\right) \hat{V}_{l,j_{\max}}^{n+1}, \tag{2.66}$$

and no interpolation is required to determine $\mathcal{A}^h V_{i,j}^{n+1}$.

When no similarity reduction is possible, the calculation of $\mathcal{A}^h V_{i,j}^{n+1}$ still requires interpolation and use of the data from pricing problems with higher strike values. Consequently, the one-dimensional problems are once again solved in decreasing order, namely from $j = j_{\max}$ to $j = 0$. Algorithm 2.3 describes the complete solution process [2].

In Section 2.5, we will show that the implicit application of the reload constraint provides much more accurate option values when compared to the explicit application of the constraint, although both methods only converge at a first order rate (in the worst case). Note that previous work on reload options in both [39] and [32] has utilized an explicit application of the reload constraint. Though simpler to implement and often commonly used in practice for pricing American-type options, this approach is not the best choice in this case since it results in poor convergence.

2.5 Numerical Results

Having examined the analytical properties of the increased reload pricing equations, we now consider numerical results obtained when pricing such contracts. We carry out a convergence study of the increased reload pricing model in Section 2.5.1 and show that companies can reduce their option expense by replacing infinite reload options ($p = 0$) by increased reload options (with $p > 0$). In Section 2.5.2, the implicit application of the reload constraint (see equation (2.16)) is shown to be superior to the explicit application

[2] It is straightforward to show that the explicit constraint (Algorithm 2.3) is unconditionally stable and monotone when fully implicit timestepping is used.

for $j = 0, \ldots, j_{\max}$ **do**
  for $i = 0, \ldots, j_{\max}$ **do**
    $V_{i,j}^0 = \text{Payoff}(S_i^j, K_j)$
  end for
end for

for $n = 0, \ldots, N$ **do**
  for $j = j_{\max}, \ldots, 0$ **do**
    Determine $\hat{V}_j^{n+1}$ by solving $\hat{\mathcal{B}}_j^{n+1}\hat{V}_j^{n+1} = V_j^n$
    for $i = 0, \ldots, j_{\max}$ **do**
      $V_{i,j}^{n+1} = \max(\mathcal{A}^h V_{i,j}^{n+1}, \hat{V}_{i,j}^{n+1})$
    end for
  end for
end for

ALGORITHM 2.3: *Explicit Application of the Reload Constraint*

of this same constraint when pricing increased reload options. Finally, in Section 2.5.3, we introduce a volatility surface [46] and study its effect on the value of increased reload options with $p = 0$.

2.5.1 Convergence Study

In this section, we carry-out a convergence analysis for the general case when $p \neq 0$ and the special case of $p = 0$. To validate our pricing model, the numerical values obtained for infinite reload options ($p = 0$) are compared with analytical values for these contracts presented in [32]. The parameter values chosen for the convergence analysis are presented in Table 2.1 and will be used throughout this section unless otherwise specified. Though not presented here, numerical tests were conducted to ensure that the choice of $K_{\max} = \$2000$, and consequently $S_{\max} = \$40000$, resulted in a minimum of 6 digits of accuracy in the numerical solution. Also, the convergence tolerance in Algorithm 2.2 is

Parameter	Value
σ - Volatility	0.30
r - Risk-free interest rate	0.04
q - Dividend yield	0.0
K - Initial strike price	\$100
S - Initial asset price	\$100
T - Contract maturity	10 years
$K_{\max}$ - Grid parameter	\$2000
$S_{\max}$ - Grid parameter	\$40000

TABLE 2.1: *Parameter values used when pricing increased reload option contracts.*

set to 1×10^{-8} and $\epsilon = \Delta\tau \times 10^{-6}$ in the penalty term (see equation (2.27)).

Table 2.2 holds the results of a convergence study for increased reload options when $p = 5\%$. Let us specify the content of each column of Table 2.2. The first column (*Refinement*) contains the refinement level used when pricing the reload contract. Each refinement level almost doubles the number of grid nodes in both the S and K directions and cuts the initial timestep size in half. Both of these parameters are included as the second (*Nodes*) and third (*Timesteps*) columns of Table 2.2. The fourth column (*Option Value*) presents the option value obtained for each refinement level. The fifth column (*Difference*) presents the difference between the option value obtained for two successive refinement operations, while the last column (*Ratio*) presents the ratio of two successive difference values. The ratio obtained in the last column indicates the convergence of the timestepping method used. For example, a ratio of 2 indicates linear convergence while a value of 4 is associated with quadratic convergence. The convergence ratio obtained in Table 2.2 for each timestepping method is consistent with local truncation error analysis, assuming a smooth solution. Indeed, linear convergence is expected when fully implicit timestepping is used, while quadratic convergence is associated with Crank-Nicolson timestepping. Note that constant timesteps are taken when fully implicit timestepping is used while variable timesteps are taken in the case of Crank-Nicolson.

Increased Reload Options when $p = 5\%$					
Refinement	Nodes	Timesteps	Option Value	Difference	Ratio
Fully Implicit					
0	61	100	54.596846	n.a.	n.a.
1	121	200	54.704649	0.107803	n.a.
2	241	400	54.749564	0.044915	2.40
3	481	800	54.769482	0.019918	2.25
4	961	1600	54.778859	0.009377	2.12
Crank-Nicolson (variable timesteps)					
0	61	101	54.741440	n.a.	n.a.
1	121	211	54.773632	0.032192	n.a.
2	241	448	54.784286	0.010654	3.02
3	481	940	54.786926	0.002640	4.04
4	961	1925	54.787581	0.000655	4.03

TABLE 2.2: *Value of an increased reload option with $p = 5\%$ at $S = \$100$ using both fully implicit and Crank-Nicolson timestepping (with variable timesteps) for different refinement levels. The reload constraint is applied implicitly as specified in Algorithm 2.1 and the initial timestep is $\Delta\tau_0 = 0.1$ years on the coarsest grid. Other parameter values are presented in Table 2.1.*

See [44] for details on the timestep selector used and an explanation of the importance of variable timestepping for American-type constraints.

To further validate our pricing model, we carry out a convergence study in the special case when $p = 0$, the results of which are presented in Table 2.3. In this case, linear convergence is obtained when fully implicit timestepping is used but quadratic convergence is not obtained when Crank-Nicolson timestepping is chosen. Indeed, Crank-Nicolson only provides linear convergence. Additional tests using a second order BDF scheme were also carried out with similar results. As discussed subsequently, we believe that the lack of quadratic convergence is due to the non-smoothness of the solution at $S = \$100$. Nonetheless, the results in Table 2.3 appear to be converging to the analytical value for infinite reload options obtained in [32]: \$64.67 at $S = \$100$. As a side note, both the similarity reduction and the full two-dimensional approach provide identical results, as shown in Table 2.4. Thus, when applicable, the similarity reduction may be considered

Infinite Reload Options					
Refinement	Nodes	Timesteps	Option Value	Difference	Ratio
Fully Implicit					
0	61	100	64.428679	n.a.	n.a.
1	121	200	64.562116	0.133437	n.a.
2	241	400	64.620278	0.058162	2.29
3	481	800	64.647254	0.026976	2.16
4	961	1600	64.660219	0.012965	2.08
Crank-Nicolson (variable timesteps)					
0	61	104	64.548919	n.a.	n.a.
1	121	229	64.619955	0.071036	n.a.
2	241	506	64.648809	0.028854	2.46
3	481	1088	64.661325	0.012516	2.31
4	961	2262	64.667198	0.005873	2.13

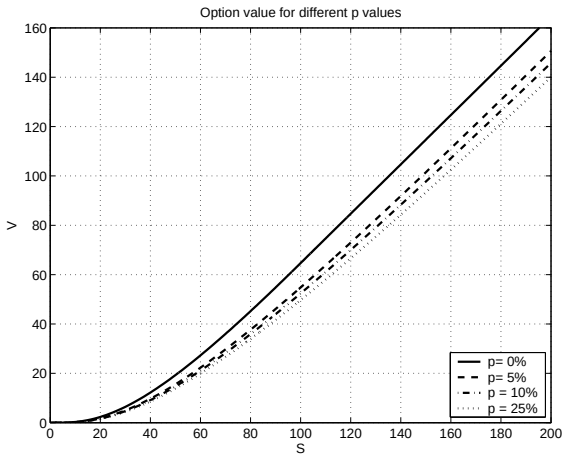
TABLE 2.3: *Value of an increased reload option with $p = 0\%$ (infinite reload option) at $S = \$100$ using both fully implicit and Crank-Nicolson timestepping (with variable timesteps) for different refinement levels. The reload constraint is applied implicitly as specified in Algorithm 2.1 and the initial timestep is $\Delta\tau_0 = 0.1$ years on the coarsest grid. Other parameter values are presented in Table 2.1.*

Nodes	Timesteps	Full 2-D	Sim. Red.
61	100	64.428679	64.428679
121	200	64.562116	64.562116
241	400	64.620278	64.620278
481	800	64.647254	64.647254
961	1600	64.660219	64.660219

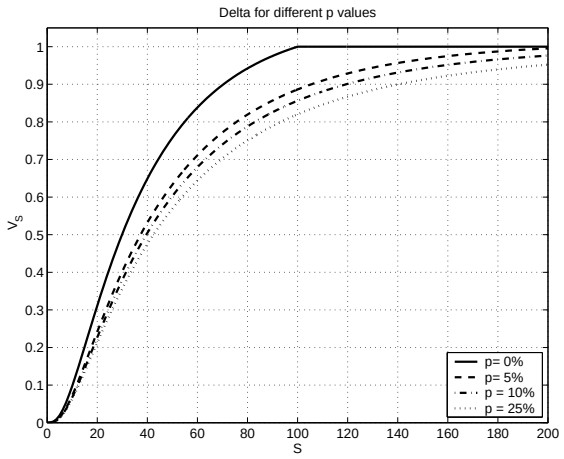
TABLE 2.4: *Value of an increased reload option with $p = 0\%$ at $S = \$100$ when the solution is obtained on a full $S \times K$ grid (*Full 2-D*) and when the solution is obtained on a single S grid using the similarity reduction (*Sim. Red.*).Fully implicit timestepping is used and the reload constraint is applied implicitly. Other parameter values used in these calculations are presented in Table 2.1.*

as an alternate and less computationally expensive solution method.

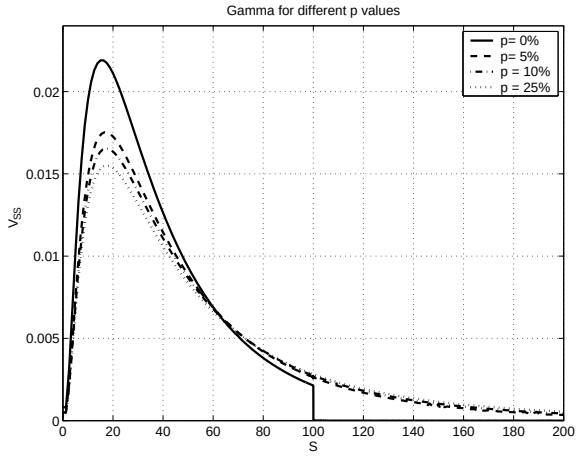
As a complement to the results presented in Table 2.3, Appendix D includes numerical values for infinite reload options when a capped boundary condition is applied at $K_{\max}$ as outlined in equation (2.14). The results in Appendix D show that a larger grid is required to provide an acceptable level of accuracy when using the capped boundary condition. Therefore, it would appear that our choice of applying a similarity reduction at $K_{\max}$, as



(A) Option value for different p values.



(B) Option delta for different p values.



(C) Option gamma for different p values.

FIGURE 2.4: *Option value (V), delta (V_S) and gamma (V_{SS}) for increased reload options with different p values. The parameters used in the pricing process are presented in Table 2.1.*

outlined in equation (2.16), remains the better of the two options considered.

To complete our analysis, we consider the particular features of the option delta (V_S) and gamma (V_{SS}). These quantities are hedging parameters, and are hence of practical

importance. Figure 2.4 presents the option value, the delta and the gamma of increased reload option contracts with different values of p. Note that the delta and gamma curves obtained when $p = 0$ are distinctly different from those obtained when $p > 0$. Indeed, the option delta curve for infinite reload options ($p = 0$) shows a kink in the curve around the strike ($S = \$100$) resulting in a discontinuity in the gamma. On the other hand, the delta and gamma curves obtained when $p > 0$ are all similar to one another and contain no such non-smoothness. The distinct shape of the gamma curve when $p = 0$ reflects the optimal exercise policy for this contract; the kink in the option gamma curve implies that it is optimal to exercise the reload option whenever $S > K$ [39, 32]. However, increased reload option contracts where $p > 0$ do not follow this optimal exercise policy which results in smooth delta and gamma curves. Clearly, the non-smoothness of the solution at $S = \$100$ has a negative effect on the convergence rate.

Since this contract modification is suggested as a tool to reduce the stock option expense for companies issuing infinite reload options, we investigate how the option value is affected by the value of p. Table 2.5 presents the value of increased reload options for five different choices of p. We see that increasing p from 0% to 1% results in an 8% reduction of the contract value. However, setting p to larger values has a less significant impact on the option value. This trend is confirmed by Figure 2.5 which demonstrates that setting p to any value greater than 30% results in the same option value[3]. Beyond that point, the value of the reload contract is essentially identical to the value of a 10 year European option.

Thus, we conclude that transforming infinite reload options ($p = 0$) into increased reload option contracts (with $p > 0$) results in a significant price reduction for small

[3]Note that the effect of p on the option value is a function of both σ and r. For example, when choosing a higher volatility ($\sigma = 0.40$ and $r = 0.04$), the value of a European option is only recovered when $p \geq 40\%$.

Refinement Level	Nodes	Percentage Increase (p) 0%	1%	5%	10%	25%
0	61	64.548919	59.398274	54.741440	52.293088	49.495693
1	121	64.619955	59.431475	54.773632	52.355101	49.631507
2	241	64.648809	59.441275	54.784286	52.370947	49.673673
3	481	64.661325	59.443599	54.786926	52.374873	49.685124
4	961	64.667198	59.444172	54.787581	52.375864	49.688072

TABLE 2.5: *Price of an increased reload option at $S = \$100$ for different p values. Additional parameters used are presented in Table 2.1. Crank-Nicolson timestepping with variable timesteps was used; the initial timestep is set to $\Delta\tau_0 = 0.1$ years.*

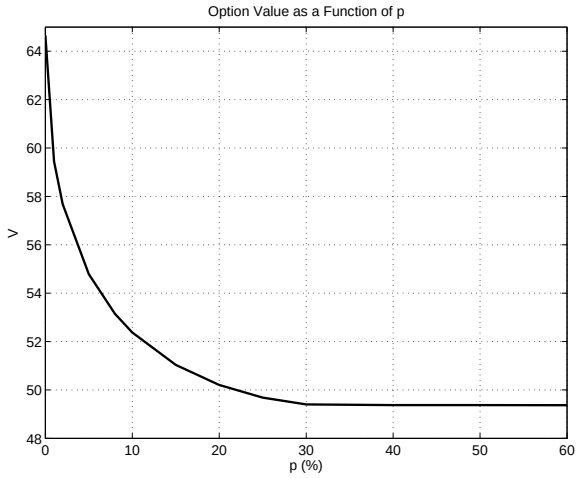


FIGURE 2.5: *Value of an increased reload option (at $S = \$100$) as a function of the percentage increase p. The parameters used are presented in Table 2.1.*

values of p. However, the effect of this contract modification is somewhat limited since the option value tends to the value of a European option for larger choices of p.

In [39], it is shown that for standard infinite reload options ($p = 0\%$), it is always optimal to reload whenever $S > K$. However, Figure 2.4 shows that for small values of p, i.e. $p = 5\%$, it is not optimal to reload for $S < \$200$ (when $K = \$100$). In fact, when $p = 5\%$, it is optimal to reload only for $S \simeq \$215$. This is confirmed by Figure 2.6 which plots the effect of small values of p on the exercise boundary of the option contract. Figure 2.6 only considers $p \leq 15\%$ since it is not optimal to exercise increased reload

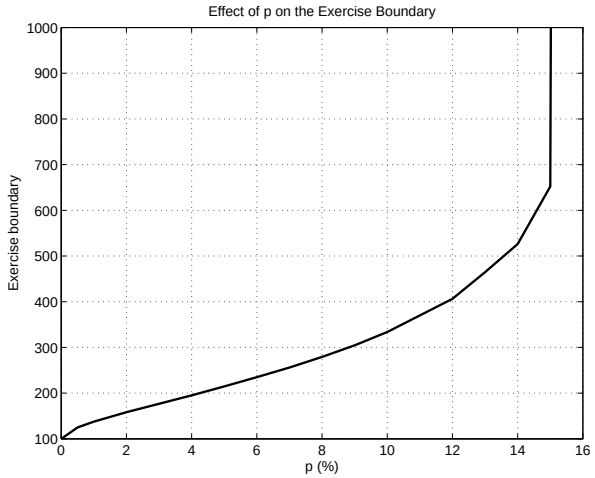


FIGURE 2.6: *Effect of small changes in p on the exercise boundary of the increased reload option on the grant date. The parameters used are presented in Table 2.1.*

contracts on the grant date when $p > 15\%$. Consequently, the optimal reload policy is extremely sensitive to small changes in p. Intuitively, this is logical since, as p increases, employees are expected to reload less often. Thus, as $p \to \infty$, the price of the increased reload option simply converges to the price of a European call option.

2.5.2 Implicit vs Explicit Application of the Reload Constraint

All numerical results presented thus far have been obtained when the reload constraint in equation (2.16) is applied implicitly (as in Algorithm 2.1). An alternative to this choice is to apply the reload constraint explicitly at each timestep as outlined in Algorithm 2.3. If we are using a fully implicit timestepping method, which is only $\mathcal{O}(\Delta\tau)$, then one may argue that we might as well use Algorithm 2.3 as done in both [39] and [32]. Though much simpler to implement, this explicit method results in poor accuracy. Indeed, Table 2.6 contains the value at $S = \$100$ of an increased reload option with $p = 0\%$ when the constraint is both applied implicitly and explicitly. In these examples, we use a similarity reduction so that no interpolation is required to determine $\mathcal{A}^h V_{i,j}^{n+1}$ (see equation (2.66)).

		Explicit Constraint			Implicit Constraint		
Nodes	Timesteps	$S = 90$	$S = 100$	$S = 110$	$S = 90$	$S = 100$	$S = 110$
61	100	52.519348	62.233168	72.233168	54.554268	64.428679	74.428679
121	200	53.208489	62.974357	72.974357	54.683619	64.562116	74.562116
241	400	53.678279	63.479419	73.479419	54.739847	64.620278	74.620278
481	800	54.005953	63.831411	73.831411	54.765882	64.647254	74.647254
961	1600	54.236391	64.078763	74.078763	54.778383	64.660219	74.660219

TABLE 2.6: *Value of an increased reload option contract with $p = 0\%$ when the reload constraint in* (2.3) *is applied both implicitly and explicitly. These results are obtained using a similarity reduction and fully implicit timestepping. The parameter values used in these calculations are presented in Table 2.1.*

As shown in Table 2.6, the numerical results obtained when the constraint is applied explicitly are very far from the analytic values for infinite reload option contracts obtained in [32]: \$54.79 at $S = \$90$, \$64.67 at $S = \$100$ and \$74.67 at $S = \$110$.

These comments are confirmed by the convergence analysis presented in Table 2.7. The results in Table 2.7 show that when the constraint is applied explicitly, the numerical solution remains far from the analytical option value for reasonable refinement levels. Indeed, the option value obtained for refinement level 4 with Crank-Nicolson timestepping and explicit application of the reload constraint is still about \$0.40 below the analytical value of \$64.67 from [32].

2.5.3 Adding a Volatility Surface

It is well known that constant volatility models cannot reproduce observed market option prices. In this section, we determine the value of an increased reload option contract with $p = 0\%$ when a volatility surface is used to replace the constant volatility assumption. A standard approach is to assume that $\sigma = \sigma(S, \tau)$ and to determine $\sigma(S, \tau)$ by calibration [85]. We consider two additional modelling assumptions regarding the properties and use of the local volatility surface. The first modelling assumption is to consider the volatility as a function of the *moneyness* of the option. The second assumption implies

Infinite Reload Options - Explicit Constraint					
Refinement	Nodes	Timesteps	Option Value	Difference	Ratio
Fully Implicit					
0	61	100	62.233168	n.a.	n.a.
1	121	200	62.974357	0.741189	n.a.
2	241	400	63.479419	0.505062	1.47
3	481	800	63.831411	0.351992	1.43
4	961	1600	64.078763	0.247352	1.42
Crank-Nicolson (constant timesteps)					
0	61	100	62.995249	n.a.	n.a.
1	121	200	63.498765	0.503516	n.a.
2	241	400	63.843231	0.344465	1.46
3	481	800	64.085195	0.241964	1.42
4	961	1600	64.256482	0.171287	1.41

TABLE 2.7: *Value of an increased reload option with $p = 0\%$ at $S = \$100$ when the reload constraint is applied explicitly using both fully implicit and Crank-Nicolson (constant timesteps) timestepping for different refinement levels. Analytical contract value is [32]:* \$64.67. *Note that a similarity reduction is used. Other parameter values chosen are presented in Table 2.1.*

that the volatility surface is *rolled forward* periodically. Both of these assumptions are often used by practitioners and will be considered in turn.

Let us outline the implications of the first modelling assumption which is referred to as the *sticky delta* property. Since local volatility surfaces can be a challenge to calibrate for different combinations of strike and asset values, this first assumption implies that the volatility skew will always be centered around the strike of the contract considered.

The *sticky delta* property stems from the following assumption:

$$\sigma(S, K, t) = \sigma(\rho S, \rho K, t). \tag{2.67}$$

Setting $\rho = \frac{K_{\text{init}}}{K}$, we obtain:

$$\sigma(S, K, t) = \sigma\left(\frac{S}{K} K_{\text{init}}, K_{\text{init}}, t\right), \tag{2.68}$$

where $\frac{S}{K}$ represents the moneyness of the option considered. This assumption is particularly relevant for pricing infinite reload options due to the definition of the reload constraint (see equation (2.16)).

The second assumption implies that the volatility surface is *rolled forward* or updated periodically. Since employee stock options have longer maturities, it is particularly pertinent to include this update process in our model. Mathematically, this second assumption implies:

$$\sigma(S, K, t) = \sigma(S, K, t - t_r), \tag{2.69}$$

where $t_r \leq t \leq t_{r+1}$ and $\{t_r\}$ represents the times when the local volatility surface is *rolled forward*.

Table 2.8 presents numerical values for infinite reload options when using a volatility surface and assuming that equation (2.68) and/or equation (2.69) hold. The local volatility surface used to price these contracts was obtained by calibration to synthetic option prices. As outlined in [90], synthetic market prices were generated for both vanilla call and put options using the exact European prices under a Merton jump diffusion model [69]. These synthetic values were generated monthly from $[0, 1.0]$ and yearly from $[1.0, 5.0]$ for 7 different strike values. Note that $\sigma = .2359$ and $\sigma = .3167$ are the constant volatility values which reproduce the jump diffusion model price of an at-the-money call option in the context of a classic Black-Scholes model (without jumps) when $T = 0.25$ years and $T = 5$ years respectively. See Figure 2.7 for a graphical representation of the volatility surface obtained. More details regarding the parameters used to generate this volatility surface are specified in [90].

Table 2.8 holds values obtained when using a volatility surface to price increased reload option contracts with $p = 0\%$. We consider two specific cases namely *Case 1* where the volatility surface described above is used and equation (2.68) holds, and *Case 2* where

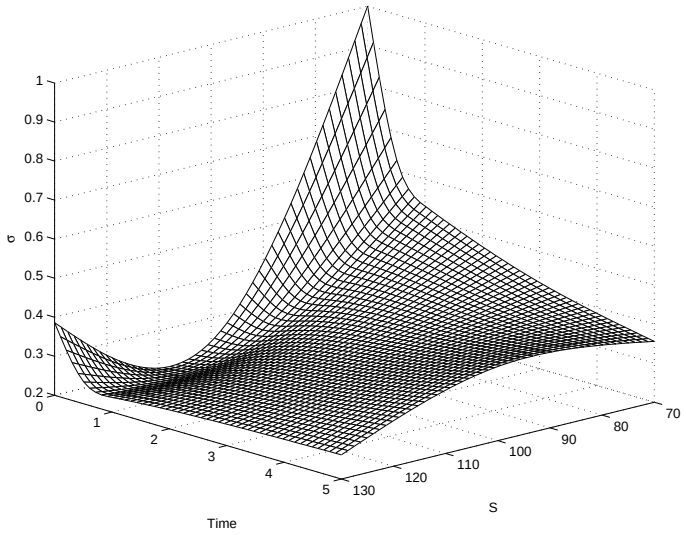


FIGURE 2.7: *Plot of volatility surface obtained by calibrating to synthetic market prices as outlined in [90].*

		Vol. Surf.		Const. Vol.	
Nodes	Timesteps	Case 1	Case 2	$\sigma = .2359$	$\sigma = .3167$
61	100	72.330961	81.703373	57.374825	66.113595
121	200	72.386614	81.554748	57.552235	66.237392
241	400	72.414143	81.481324	57.621998	66.292977
481	800	72.427872	81.444101	57.652163	66.319226
961	1600	72.434732	81.425310	57.666067	66.331968

TABLE 2.8: *Value of an increased reload option when $p = 0\%$ at $S = \$100$ for different volatility assumptions.* Vol. Surf. *indicates the use of a volatility surface while* Case 1 *implies that equation* (2.68) *holds and* Case 2 *implies that both equations* (2.68) *and* (2.69) *hold. For comparison purposes, we include results for two constant volatility values (*Const. Vol.*). Note that fully implicit timestepping is chosen. Additional parameter values are presented in Table 2.1.*

the same volatility surface is used but now both equations (2.68) and (2.69) hold. In *Case 2*, the volatility surface is *rolled forward* every year during the contract lifetime. For comparison purposes, we also present option values obtained under the assumption of constant volatility when $\sigma = .2359$ and $\sigma = .3167$. The data from *Case 1* and *Case 2* in Table 2.8 highlights the impact of updating the volatility surface on the contract

value. Indeed, updating the volatility surface every year results in a significant increase in option value. This seems intuitively correct since the volatility surface flattens out as t grows larger. Furthermore, the use of a volatility surface, compared to the assumption of constant volatility, results in increased option values.

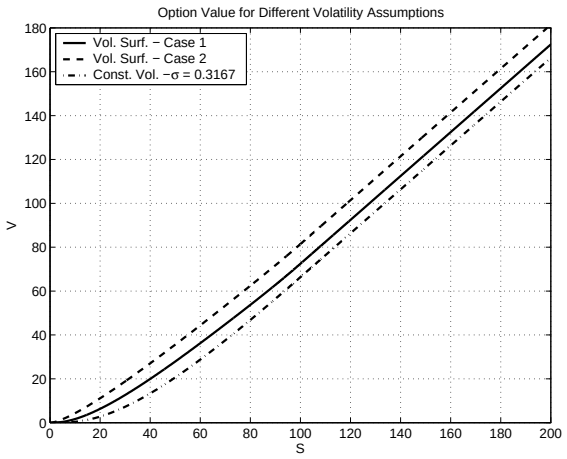
Figure 2.8 shows the option value (V), the delta (V_S) and the gamma (V_{SS}) for the cases considered in Table 2.8. While the shape of the option value curve is relatively unaffected by the use of a local volatility surface, this is not the case for the delta and gamma of the option. Indeed, the delta curve (and consequently the gamma curve) is significantly different when a local volatility surface is used for $S < K$. Nonetheless, the characteristic non-smoothness at the strike in the delta curve is preserved when a local volatility surface is added to the pricing model.

2.6 Summary

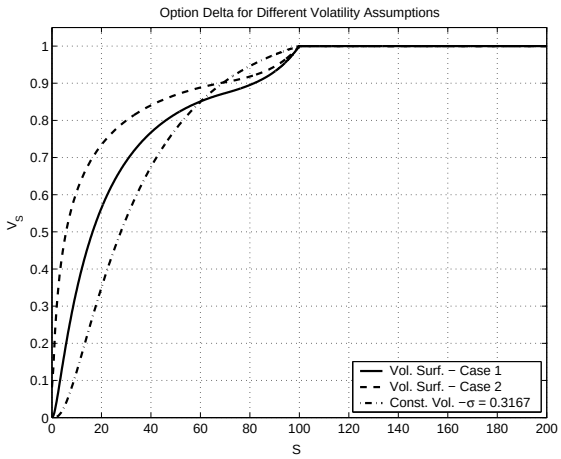
Infinite reload options are considered as some of the more complex and costly employee stock options. As such, issuing companies that need to include these contracts in their balance sheets may be looking for ways to reduce their no-arbitrage value. In this Chapter, we defined the increased reload option which is obtained by modifying the infinite reload option contract. Instead of receiving new options where $K = S$ following a reload event (which is the case for infinite reload options), owners would receive options where $K = S \times (1 + p)$, where p is a percentage increase parameter. Defined in this context, the infinite reload option contract becomes a special case of the increased reload contract where $p = 0$. Our numerical results have shown that the increased reload option contract can be used as a tool to help companies reduce their option expense.

More specifically, the following contributions were made:

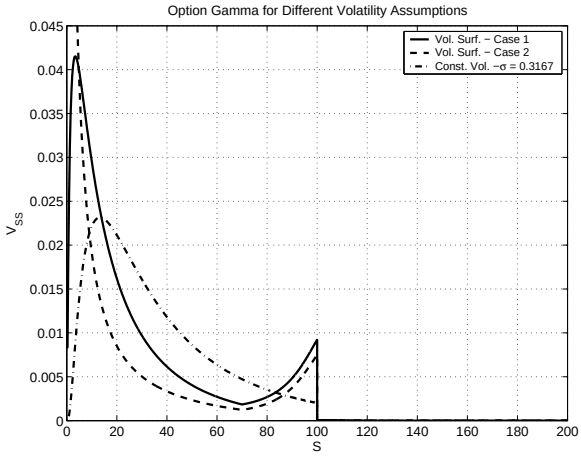
- A detailed pricing model for increased reload options was outlined as an impulse



(A) Option value.



(B) Option delta.



(C) Option gamma.

FIGURE 2.8: *Plot of the option value (V), the delta (V_S) and the gamma (V_{SS}) for an increased reload option with $p = 0\%$ under different modelling assumptions.* Vol. Surf. – Case 1 *implies that a local volatility surface is used and equation* (2.68) *holds.* Vol. Surf. – Case 2 *implies that a local volatility surface is used and equations* (2.68) *and* (2.69) *both hold. The update interval in this case is 1 year.* Const. Vol. *makes the assumption that $\sigma = .3167$ is a constant parameter. Other parameters used in the pricing process are presented in Table 2.1.*

control problem which was formulated as a Hamilton-Jacobi-Bellman equation. A penalization method was also used to apply the reload constraint implicitly.

- To ensure convergence to the viscosity solution, we have shown that the discrete equations obtained from the pricing model are consistent, stable and monotone [8].

- It was also shown that the implicit application of the reload constraint is clearly superior to applying the constraint explicitly.

- Furthermore, the effect of the percentage increase parameter value p on both the option value and the optimal exercise policy was outlined. Indeed, setting p to a small non-zero value results in a significant reduction in contract value. In addition, the exercise policy is very sensitive to small values of p.

Since setting p to a small non-zero value significantly reduces the no-arbitrage price of the stock options considered, this simple contract modification may be easily accepted by stock option owners while providing a non-negligible price reduction for issuing companies. While we considered the impact of a rather simple contract modification on the option value, there are numerous other changes that could be made to infinite reload options to reduce their no-arbitrage price. Consider for example imposing a minimum holding period for either company stock or employee stock options following a reload event. This contract modification would limit the number of possible reloads during the lifetime of the contract, thus reducing the value of infinite reload options significantly. These different contract modifications will surely be considered more seriously by companies that have issued stock options, and that are now seeking to reduce the stock option expense recorded on their balance sheet.

Chapter 3

Infinite Reload Options with Vesting

Companies granting employee stock options often include a vesting clause or vesting period[1] in these contracts. While the typical maturity for employee stock options is ten years, the length of a vesting period generally ranges from six months to three years. Depending on their structure, some employee stock options may be more significantly impacted by the addition of a vesting period. For example, the initial reload option grant, as well as each subsequent option received following a reload event, is subject to the time vesting requirement. In this chapter, we investigate how the vesting period affects the no-arbitrage value of infinite reload options.

In [39], [33] and [61], the authors discuss numerical methods for valuing reload options with a time vesting constraint. The authors of [39] use a trinomial pricing model with two state variables to price infinite reload options with vesting. Results in [39] show

[1]During the vesting period, employee stock options cannot be exercised. For example, owners of reload options cannot trigger a reload event during the vesting period of the contract.

that the infinite reload option value is reduced and the optimal exercise policy[2] is no longer applicable when vesting periods are added. In [61], the author presents analytical pricing formulas for infinite reload options with time vesting constraints while assuming that only the initial option grant is subject to vesting (options received following a reload event vest immediately). In [33], a pricing method applicable to all reload options with a time vesting requirement (including those with finite and infinite reload possibilities) is developed. This pricing method is based on the binomial tree and numerical values for reload options with both finite or infinite reload possibilities are presented. The authors of [33] also state that the value of an infinite reload option is equal to that of a reload option with finite reload rights where the number of remaining reloads is a function of the time to expiry.

In this chapter, we extend the impulse control pricing model for infinite reload options presented in Chapter 2 by including the time vesting requirement. The derived partial differential equation contains an extra path-dependent variable to keep track of the vesting period. Since the resulting pricing equation is convection-dominated, a semi-Lagrangian scheme [37, 80, 41] is used during the discretization process. The infinite reload pricing problem with vesting is solved using an operator splitting method [55, 56] due to the observed poor convergence of the penalty iteration scheme. However, the operator splitting scheme has the disadvantage that it is not monotone. Numerical results are generated including a convergence analysis of the pricing method.

[2]Recall that the optimal exercise policy for infinite reload options states that the holder should reload whenever the option is in-the-money. See Chapter 2 for more details.

3.1 Option Pricing Model

To add vesting periods to the reload pricing model outlined in Chapter 2, we define an additional path-dependent variable which determines whether or not the vesting period has ended. Let u be defined as the time that has elapsed since the most recent reload event:

$$u = t - t_r, \tag{3.1}$$

where t_r is the time at which the last reload event occurred and t is the current time. When the option is granted, u is set to zero and increases at the same rate as time (going forward). When a reload event occurs, the owner receives new options that are subject to the same vesting period and u is reset to zero. Once the vesting period is over, we no longer need to keep track of u since it no longer affects the option value. Consequently, u is defined on the range $[0, v_p]$, where $v_p > 0$ is the vesting period of the initial option contract.

Let $V = V(S, K, u, t)$ denote the value of an infinite reload option with vesting period v_p, where S is the company stock price and K is the option strike price. As in Chapter 2, we assume that S follows geometric Brownian motion (see equation (2.1)).

Once the vesting period has elapsed ($u \geq v_p$), the owner of the contract can choose to reload his option at any time. A reload constraint is applied to ensure that the current value of the reload option is never less than the value obtained following a reload event. The reload constraint, denoted as $\mathcal{A}V = \mathcal{A}V(S, K, u, t)$, is defined as:

$$\mathcal{A}V(S, K, u, t) = \begin{cases} S - K + \frac{K}{S} V(S, S, u = 0, t) & \text{if } S > K \text{ and } u \geq v_p, \\ 0 & \text{otherwise,} \end{cases} \tag{3.2}$$

where $V(S, S, u = 0, t)$ represents the new option received by the owner with strike price

$K = S$ (and subject to a vesting period of v_p). The reload constraint in equation (3.2) can be defined as a non-local impulse operator (see Chapter 1) where $S - K$ is the cash flow and $\frac{K}{S}V(S, S, u = 0, t)$ is the intervention operator.

If the option has vested ($u \geq v_p$) when the contract reaches maturity ($t = T$), the owner receives one unit of stock for each infinite reload option owned. However, if the reload option expires before the end of the vesting period ($u < v_p$), the owner forfeits any payoff. Thus, the payoff of an infinite reload option with vesting can be written as:

$$\text{Payoff}(S, K) = \begin{cases} \max(S - K, 0) & \text{if } u \geq v_p, \\ 0 & \text{otherwise,} \end{cases} \tag{3.3}$$

where K is the strike price of the option at expiry and S is the value of the company stock at expiry.

Due to the presence of an extra path-dependent variable, the value of an infinite reload option with vesting no longer satisfies the classic Black-Scholes equation. The specific pricing equation for infinite reload options with vesting is now derived using the same steps as for the Black-Scholes derivation [85].

We construct a hedging portfolio Π which contains the infinite reload option and a certain amount β of company stock:

$$\Pi = V - \beta S, \tag{3.4}$$

where we choose $\beta = V_S$.

Applying Ito's lemma [85, 54] to $V = V(S, K, u, t)$, we obtain:

$$dV = V_t dt + V_u du + V_S dS + \frac{1}{2}\sigma^2 S^2 V_{SS} dt. \tag{3.5}$$

Since u increases at the same rate as forward time t, we have:

$$du = dt. \tag{3.6}$$

Taking equation (3.6) into consideration, equation (3.5) becomes:

$$dV = V_t dt + V_u dt + V_S dS + \frac{1}{2}\sigma^2 S^2 V_{SS} dt. \tag{3.7}$$

From equation (3.4), the change in portfolio value Π is:

$$\begin{aligned} d\Pi &= dV - [\beta dS + \beta S q dt] \\ &= V_t dt + V_u dt + \frac{1}{2}\sigma^2 S^2 V_{SS} dt - \beta S q dt, \end{aligned} \tag{3.8}$$

where q is the dividend rate.

Since the portfolio Π is risk-free, we have:

$$d\Pi = r\Pi dt, \tag{3.9}$$

where r is the risk-free rate of return.

Combining equations (3.4), (3.8) and (3.9), we obtain:

$$V_t + V_u + (r-q)SV_S + \frac{1}{2}\sigma^2 S^2 V_{SS} - rV = 0. \tag{3.10}$$

Defining $\tau = T - t$ as the time to expiry, equation (3.10) can be written as:

$$V_\tau = V_u + (r-q)SV_S + \frac{1}{2}\sigma^2 S^2 V_{SS} - rV. \tag{3.11}$$

Equation (3.11) can now be combined with the reload constraint in equation (3.2) to

provide a complete pricing framework for infinite reload options with vesting. Defining the differential operator $\mathcal{L}$ as in equation (2.4) in Chapter 2, the no-arbitrage value of the infinite reload option with vesting is obtained by solving the impulse control problem:

$$\min\Big(V_\tau - V_u - \mathcal{L}V, V - \mathcal{A}V\Big) = 0, \tag{3.12}$$

where $\mathcal{A}V = \mathcal{A}V(S, K, u, \tau)$ is defined in equation (3.2). The initial conditions for the pricing problem in (3.12) are set to the payoff defined in equation (3.3):

$$V(S, K, u, \tau = 0) = \begin{cases} \max(S - K, 0) & \text{if } u \geq v_p, \\ 0 & \text{otherwise.} \end{cases} \tag{3.13}$$

The infinite reload pricing problem with vesting can also be written as a penalized problem:

$$\lim_{\epsilon \longrightarrow 0} \left(V_\tau - V_u - \mathcal{L}V - \frac{1}{\epsilon}\max(\mathcal{A}V - V, 0)\right) = 0. \tag{3.14}$$

Even though equation (3.14) suggests that the penalty method in [44] could be applied easily, to solve the option pricing problem, we use the operator splitting method outlined in [55] and [56]. This choice is a consequence of the slow convergence of the penalty method when applied to equation (3.14) and is discussed in more detail in Section 3.2.2.

Assuming homogeneity conditions are met, the reload constraint in equation (3.2) can be simplified. If $V(S, K, u, \tau)$ is homogeneous of degree one in (S, K) (see Definition 2.1), then the constraint in equation (3.2) can be simplified by setting $K = S$ in equation (2.7) and choosing $\lambda = \frac{K}{S}$, to get:

$$V(S, S, u, \tau) = \frac{S}{K} V(K, K, u, \tau). \tag{3.15}$$

Using equation (3.15), the reload constraint in equation (3.2) can be written as:

$$\mathcal{A}V(S,K,u,\tau) = \begin{cases} S - K + V(K,K,u=0,\tau) & \text{if } S > K \text{ and } u \geq v_p \text{ ,} \\ 0 & \text{otherwise.} \end{cases} \tag{3.16}$$

Hence, when a similarity reduction is possible, the constraint in equation (3.16) can be used when solving equation (3.12) in lieu of the usual reload constraint in equation (3.2). Note that the use of a similarity reduction also reduces the dimensionality of the option pricing problem. However, since the similarity reduction is only applicable when homogeneity conditions are met, it will be treated as a special case.

3.1.1 Solution Domain and Boundary Conditions

In this section, we determine the discrete solution domain and associated boundary conditions for the infinite reload pricing problem in equation (3.12). Since the option value no longer depends on u once the vesting period is over, we have:

$$V(S,K,u^*,\tau) = V(S,K,v_p,\tau), \tag{3.17}$$

for any $u^* > v_p$. Restricting the domain in the u direction to $[0, v_p]$, we define the discrete solution domain as: $[0, S_{\max}] \times [0, K_{\max}] \times [0, u_{\max}]$, where $S_{\max} \gg K_{\max}$ and $u_{\max} = v_p$.

To localize the reload option pricing problem, we specify additional boundary conditions in each of the S, K and u directions. Let us first consider the boundary condition as $S \to 0$. When $S = 0$, equation (3.12) simplifies to:

$$V_\tau - V_u + rV = 0. \tag{3.18}$$

As $K \to 0$, no additional boundary condition is necessary since the differential operator

$\mathcal{L}$ in equation (2.4) contains no K derivatives.

The boundary conditions as $K \to K_{\max}$ and $S \to S_{\max}$ should be handled with some care. For $S = S_{\max}$ (with $S_{\max} \gg K_{\max}$), we make the assumption that:

$$V \sim H(u,\tau)S\,; \quad S \to S_{\max}, \tag{3.19}$$

which implies:

$$\min(V_\tau - V_u + qV, V - \mathcal{A}V) = 0\,; \quad S \to S_{\max}. \tag{3.20}$$

As in Chapter 2, we assume that a similarity reduction is applicable for $K = K_{\max}$ and/or $S > K_{\max}$. This is equivalent to assuming that the volatility is approximately constant for those cases when $K = K_{\max}$ and/or $S > K_{\max}$. The reload constraint in equation (3.2) can now be rewritten to reflect this assumption:

$$\mathcal{A}V(S,K,u,\tau) = \begin{cases} S - K + \frac{K}{S}V(S,S,u=0,\tau) & \text{if } K < S \le K_{\max} \text{ and } u = v_p, \\ S - K + V(K,K,u=0,\tau) & \text{if } S > K_{\max} \text{ and } u = v_p, \\ 0 & \text{otherwise.} \end{cases} \tag{3.21}$$

As $u \to 0$, recall that the reload constraint is no longer applied. As such, the equation solved at $u = 0$ is:

$$V_\tau - V_u - \mathcal{L}V = 0, \tag{3.22}$$

where $\mathcal{L}$ is defined in equation (2.4). Note that this does not cause any problems since there is an outward characteristic at $u = 0$. As $u \to \infty$, the pricing problem no longer depends on u once $u \ge v_p$ which implies $V_u = 0$ at $u = v_p$. Thus, we simply solve equation (3.12) at $u = v_p$ with $V_u = 0$.

In summary, pricing infinite reload options with vesting involves solving:

$$\min(V_\tau - V_u - \mathcal{L}V, V - \mathcal{A}V) = 0, \tag{3.23}$$

on the discrete domain $[0, S_{\max}] \times [0, K_{\max}] \times [0, u_{\max}]$ with initial conditions:

$$V(S, K, u, \tau = 0) = \begin{cases} \max(S - K, 0) & \text{if } u = u_{\max} = v_p, \\ 0 & \text{otherwise,} \end{cases} \tag{3.24}$$

and boundary conditions:

$$V_\tau - V_u + rV = 0 \qquad \text{for } S = 0, \tag{3.25}$$

$$\min(V_\tau - V_u + qV, V - \mathcal{A}V) = 0 \qquad \text{for } S = S_{\max}, \tag{3.26}$$

$$V_\tau - V_u - \mathcal{L}V = 0 \qquad \text{for } u = 0, \tag{3.27}$$

$$\min(V_\tau - \mathcal{L}V, V - \mathcal{A}V) = 0 \qquad \text{for } u = u_{\max},\ S < S_{\max}, \tag{3.28}$$

where the reload constraint $\mathcal{A}V$ is calculated as in equation (3.21).

3.1.2 Reformulation of the Continuous Pricing Problem

To avoid algebraic complexity, the infinite reload pricing problem with vesting has thus far been defined as an impulse control problem in equation (3.12) or as a penalized problem in equation (3.14). However, since a discrete penalty method [44] cannot be used due to the slow convergence of the iteration (as discussed in Section 3.2.2), an operator splitting approach is used in the solution process. In this section, we reformulate the continuous pricing problem as an equivalent set of quasi-variational inequalities to facilitate the description of the discrete solution process.

We rewrite the pricing equation (3.12) as a set of quasi-variational inequalities:

$$\begin{aligned} V_\tau - V_u - \mathcal{L}V &\geq 0 \\ V - \mathcal{A}V &\geq 0 \\ (V_\tau - V_u - \mathcal{L}V)(V - \mathcal{A}V) &= 0 \end{aligned} \tag{3.29}$$

where the last equation indicates that either $(V - \mathcal{A}V) = 0$ or $(V_\tau - V_u - \mathcal{L}V) = 0$ at any given point in the domain.

Based on the formulation of the linear complementarity problem in [55] and [56], we introduce an auxiliary variable ξ and rewrite equation (3.29) as:

$$V_\tau - V_u = \mathcal{L}V + \xi \tag{3.30}$$

$$V - \mathcal{A}V \geq 0 \; ; \quad \xi \geq 0 \tag{3.31}$$

$$(V - \mathcal{A}V)\xi = 0. \tag{3.32}$$

Note that the initial and boundary conditions outlined previously are still needed to fully specify the reformulated option pricing problem defined above.

3.2 Derivation of the Discrete Equations

In this section, we outline how equation (3.12) is discretized over the solution domain in the context of the operator splitting method [55]. To simplify our derivation, we describe in detail the construction of the underlying grid.

Since the option value depends on three path-dependent variables as well as time, we construct a set of nodes which spans the $[0, S_{\max}] \times [0, K_{\max}] \times [0, v_p]$ domain. A set of equally spaced nodes in the u direction $\{u_k\}$, for $k = 0, \ldots, k_{\max}$ is constructed where

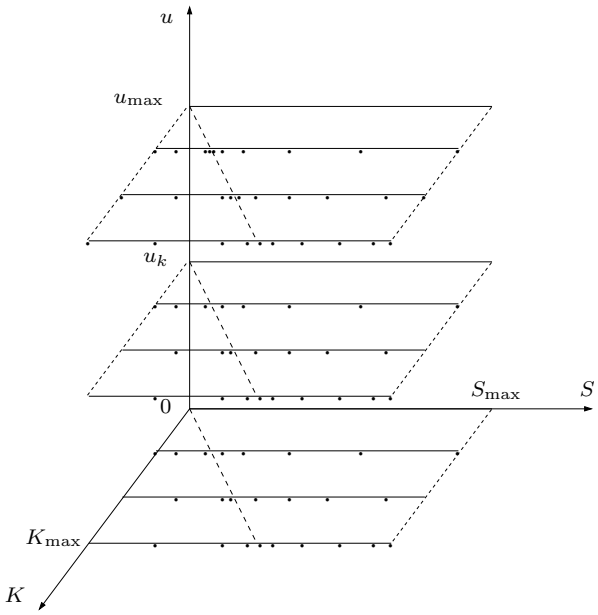


FIGURE 3.1: *Example of a three-dimensional $S \times K \times u$ domain where each $S \times K$ plane contains a scaled grid as shown in Figure 2.2.*

$u_0 = 0$ and $u_{k_{\max}} = v_p$. For each u_k, we build a two-dimensional *scaled grid* which covers the $[0, S_{\max}] \times [0, K_{\max}]$ domain. As described in Section 2.2.1, the characteristic of the *scaled grid* construction is that it concentrates nodes near the line $S = K$. Thus, we obtain $k_{\max} + 1$ rectangular grids spanning the $S \times K$ domain, each associated with a u_k value. An example of the final three-dimensional grid is included as Figure 3.1.

Having defined the underlying grid, we discretize equation (3.30) in both space and time, setting aside for now the additional constraints presented as equations (3.31) and (3.32) and the auxiliary variable ξ. Due to the absence of a diffusion term in the u direction, equation (3.30) is considered to be convection-dominated. While the differential term on the right-hand side of equation (3.30) is discretized using standard finite difference methods, the terms on the left-hand side of equation (3.30) are discretized using a semi-Lagrangian approach as done in [36]. Often used in areas such as weather prediction,

oceanography and fluid dynamics, semi-Lagrangian schemes are designed to deal with numerical problems arising from convection-dominated equations. The semi-Lagrangian approach enables us to trace the trajectory of flow particles back in time while knowing their current position. In [36], semi-Lagrangian timestepping is used to solve a partial integral differential equation when pricing Asian options with jumps in the underlying. We apply a similar technique to our infinite reload pricing equation.

Let us consider a trajectory in the u direction denoted by $\hat{u} = \hat{u}(\tau; S, K)$, where S and K are constant and fixed values. Using equation (3.6) and recalling that $dt = -d\tau$, the $\hat{u}$ trajectory satisfies:

$$\frac{d\hat{u}}{d\tau} = -1. \tag{3.33}$$

Defining the Lagrangian derivative of V as:

$$\frac{DV}{D\tau} = V_\tau + V_u \frac{d\hat{u}}{d\tau} \tag{3.34}$$

and using equation (3.33), we can write equation (3.30) (ignoring for now the auxiliary variable ξ) as:

$$\frac{DV}{D\tau} = \mathcal{L}V, \tag{3.35}$$

where $\mathcal{L}$ is defined in equation (2.4).

We assume the trajectory of $\hat{u}$ values arrives at the discrete node (S_i^j, K_j, u_k) at time $\tau = \tau^{n+1}$, with both S_i^j and K_j being constant. The departure point of this trajectory at time $\tau = \tau^n$ is denoted as $\hat{u}(\tau^n) = \hat{u}(\tau^n; S_i^j, K_j, u_k, \tau^{n+1})$. Keep in mind that the departure point of the trajectory may not coincide with an existing grid node since $\hat{u}(\tau^n)$

is determined by solving:

$$\frac{d\hat{u}}{d\tau} = -1 \qquad (3.36)$$
$$S = S_i^j \ ; \ K = K_j \ ; \ \hat{u}(\tau) = u_k \text{ at } \tau = \tau^{n+1},$$

from $\tau = \tau^{n+1}$ to $\tau = \tau^n$. Therefore, $\hat{u}(\tau^n)$ will satisfy:

$$\begin{aligned} \hat{u}(\tau^n) &= u_k - \int_{\tau^{n+1}}^{\tau^n} d\tau \\ &= u_k + (\tau^{n+1} - \tau^n) \end{aligned} \qquad (3.37)$$

where $T \geq \tau^{n+1} > \tau^n$.

Let $V_{i,j,k}^n = V(S_i^j, K_j, u_k, \tau^n)$ denote the discrete reload option value at node (S_i^j, K_j, u_k) and time τ^n, and $V_{i,j,k'}^n = V(S_i^j, K_j, \hat{u}(\tau^n), \tau^n)$ denote the option value at the departure point of the trajectory $\hat{u}(\tau^n)$. Limited quadratic interpolation along the u-axis can be used when calculating $V_{i,j,k'}^n$ [41]. However, linear interpolation is required to ensure monotonicity if desired. Assuming the index p such that $u_p \leq \hat{u}(\tau^n) \leq u_{p+1}$ is known, linear interpolation along the u-axis implies:

$$V_{i,j,k'}^n = (1-\gamma)V_{i,j,p}^n + \gamma V_{i,j,p+1}^n \qquad (3.38)$$

where both $V_{i,j,p}^n$ and $V_{i,j,p+1}^n$ correspond to pre-existing grid nodes (see Section 3.2) and the interpolation weight $0 \leq \gamma \leq 1$ is defined as:

$$\gamma = \frac{\hat{u}(\tau^n) - u_p}{u_{p+1} - u_p}. \qquad (3.39)$$

Thus, the time discretization of equation (3.35) along the characteristic curve implies:

$$\frac{V^{n+1}_{i,j,k} - V^{n}_{i,j,k'}}{\Delta\tau} = \theta[\mathcal{L}^h V]^n_{i,j,k'} + (1-\theta)[\mathcal{L}^h V]^{n+1}_{i,j,k} \tag{3.40}$$

where $0 \leq \theta \leq 1$ determines which timestepping method is used in the solution process and $[\mathcal{L}^h V]^n_{i,j,k}$ is defined as:

$$[\mathcal{L}^h V]^n_{i,j,k} = \alpha_{i,j} V^n_{i-1,j,k} + \beta_{i,j} V^n_{i+1,j,k} - (\alpha_{i,j} + \beta_{i,j} + r) V^n_{i,j,k} \tag{3.41}$$

where $\alpha_{i,j} \geq 0$ and $\beta_{i,j} \geq 0$ are determined according to the algorithm in Appendix A.

In the context of the operator splitting method [55], the solution process involves two fractional steps. First, equation (3.40), augmented by the auxiliary variable ξ, is solved providing us with an intermediate solution denoted as $\bar{V}^n_{i,j,k} = \bar{V}(S^j_i, K_j, u_k, \tau^n)$. Second, the reload constraint in equation (3.21) is applied and the auxiliary variable ξ is updated accordingly. The discrete value of the auxiliary variable ξ is denoted as $\xi^n_{i,j,k} = \xi(S^j_i, K_j, u_k, \tau^n)$.

To facilitate the description of this process, we use the following vector notation:

$$V^n_{j,k} = \begin{bmatrix} V^n_{0,j,k} \\ V^n_{1,j,k} \\ \vdots \\ V^n_{j_{\max},j,k} \end{bmatrix}; \quad \bar{V}^n_{j,k} = \begin{bmatrix} \bar{V}^n_{0,j,k} \\ \bar{V}^n_{1,j,k} \\ \vdots \\ \bar{V}^n_{j_{\max},j,k} \end{bmatrix}; \quad \xi^n_{j,k} = \begin{bmatrix} \xi^n_{0,j,k} \\ \xi^n_{1,j,k} \\ \vdots \\ \xi^n_{j_{\max},j,k} \end{bmatrix}, \tag{3.42}$$

and define the tri-diagonal coefficient matrix $\mathcal{D}$ as:

$$\left[\mathcal{D} V^n_{j,k}\right]_i = V^n_{i,j,k}(\alpha_{i,j} + \beta_{i,j} + r)\Delta\tau - \Delta\tau\beta_{i,j} V^n_{i+1,j,k} - \Delta\tau\alpha_{i,j} V^n_{i-1,j,k}. \tag{3.43}$$

We denote the discrete value of the reload constraint as $\mathcal{A}^h V^n_{i,j,k} = \mathcal{A}^h V(S^j_i, K_j, u_k, \tau^n)$

and use the vector notation defined in (3.42) to obtain:

$$\mathcal{A}^h V^n_{j,k} = \begin{bmatrix} \mathcal{A}^h V^n_{0,j,k} \\ \mathcal{A}^h V^n_{1,j,k} \\ \vdots \\ \mathcal{A}^h V^n_{j_{\max},j,k} \end{bmatrix}. \tag{3.44}$$

Similarly, the interpolated semi-Lagrangian terms are denoted by:

$$\mathcal{I}_{i,j,k'} V^n = \begin{bmatrix} V^n_{0,j,k'} \\ V^n_{1,j,k'} \\ \vdots \\ V^n_{j_{\max},j,k'} \end{bmatrix} = aV^n_{j,p} + bV^n_{j,p+1} \tag{3.45}$$

where $\mathcal{I}_{i,j,k'}$ represents the interpolation operator, $a = [(1-\gamma)\ (1-\gamma)\ \ldots\ (1-\gamma)]$ and $b = [\gamma\ \gamma\ \ldots\ \gamma]$ in accordance with equation (3.38).

The first step of the solution method involves solving:

$$[I + (1-\theta)\mathcal{D}]\bar{V}^{n+1}_{j,k} = \mathcal{I}_{i,j,k'} V^n - \theta \mathcal{I}_{i,j,k'} \mathcal{D} V^n + \Delta\tau \xi^n_{j,k}, \tag{3.46}$$

to determine the intermediate solution vector $\bar{V}^{n+1}_{j,k}$. Note that equation (3.46) corresponds to equation (3.40) in matrix form with the addition of the discrete value of the auxiliary variable. Consistent with [55], we initially set the auxiliary variable to $\xi^0_{i,j,k} = 0$ for all i, j, k.

The second fractional step involves updating the option value and the auxiliary variable component-wise to enforce the linear complementarity conditions. In [55], the au-

for $i = 0, \ldots, j_{\max}$ **do**
  $V^{n+1}_{i,j,k} = \max\left(\bar{V}^{n+1}_{i,j,k}, \mathcal{A}^h V^{n+1}_{i,j,k}\right)$
  if $\bar{V}^{n+1}_{i,j,k} > \mathcal{A}^h V^{n+1}_{i,j,k}$ **then**
    $\xi^{n+1}_{i,j,k} = 0$
  else
    $\xi^{n+1}_{i,j,k} = \xi^{n}_{i,j,k} + \frac{1}{\Delta\tau}\left(\bar{V}^{n+1}_{i,j,k} - \mathcal{A}^h V^{n+1}_{i,j,k}\right)$
  end if
end for

ALGORITHM 3.1: *Update Process in Operator Splitting Method*

thors outline the update process as:

$$\bar{V}^{n+1}_{i,j,k} - V^{n+1}_{i,j,k} - \Delta\tau(\xi^{n+1}_{i,j,k} - \xi^{n}_{i,j,k}) = 0, \tag{3.47}$$

$$\xi^{n+1}_{i,j,k}(V^{n+1}_{i,j,k} - \mathcal{A}^h V^{n+1}_{i,j,k}) = 0, \tag{3.48}$$

$$V^{n+1}_{i,j,k} \geq \mathcal{A}^h V^{n+1}_{i,j,k} \ ; \quad \xi^{n+1}_{i,j,k} \geq 0, \tag{3.49}$$

when fully implicit timestepping ($\theta = 0$) or Crank-Nicolson timestepping ($\theta = 1/2$) is used to solve equation (3.46). Equations (3.47)–(3.49) are equivalent to the update process in Algorithm 3.1. In [56], it is shown that the discrete scheme in equations (3.46)–(3.49) approximates the reformulation of the linear complementarity problem in (3.30)–(3.32) to second order in time for sufficiently smooth solutions.

When computing the discrete reload constraint defined in equation (3.21), diagonal interpolation is used along the $K = S$ line to determine $V(S, S, u = 0, \tau)$ as described in equation (2.30) (with $p = 0$). The discrete version of the reload constraint in (3.21) can

be written as:

$$\mathcal{A}^h V(S_i^j, K_j, u_k, \tau^{n+1}) = \begin{cases} S_i^j - K_j + \frac{K_j}{S_i^j}\left((1-\omega)\, V_{l,m,0}^{n+1} + \omega V_{l,m+1,0}^{n+1}\right) & \\ \qquad\qquad \text{if } S_i^j > K_j,\ S_i^j \leq K_{j_{\max}},\ u_k = v_p, & \\ S_i^j - K_j + V_{l,j,0}^{n+1} \qquad \text{if } S_i^j > K_j,\ S_i^j > K_{j_{\max}},\ u_k = v_p, & \\ 0 \qquad\qquad \text{otherwise,} & \end{cases} \tag{3.50}$$

where the interpolation weight $0 \leq \omega \leq 1$ is defined in equation (2.31) (with $p = 0$) and the index l is such that $S_l^j = K_j$ and $V_{l,j,0}^{n+1} = V(S_l^j, K_j, u_0, \tau^{n+1})$. When a similarity reduction is applicable, the calculation of the reload constraint requires no interpolation since the node (S_l^j, K_j) is included in the grid constructed for each u_k (see Section 2.2.1).

3.2.1 Theoretical Issues

In this section, we verify the theoretical properties of the discrete equations to ensure that the solution of our numerical scheme converges to the unique viscosity solution [31]. Based on [8, 11], verifying that the numerical scheme is stable, monotone and consistent ensures convergence to the viscosity solution, assuming that the strong comparison principle applies [8]. However, if one of these three properties is not satisfied by the numerical scheme, then convergence to the viscosity solution cannot be guaranteed.

Consistency

In [56], the authors show that the discrete operator splitting scheme is consistent. Therefore, the discrete solution method presented previously satisfies the consistency requirement.

Stability

While the authors of [56] also claim that the numerical scheme is stable, it is not clear that this can be proved rigorously.

Monotonicity

We show that the discrete equations do not satisfy the monotonicity property by providing a counter-example. We denote the discrete equations obtained from the operator splitting scheme as $g_{i,j,k} = g_{i,j,k}(V^{n+1}_{i,j,k}, \mathcal{I}_{i,j,k'}V^n, \xi^n_{j,k}) = 0$, where $\mathcal{I}_{i,j,k'}V^n = \mathcal{I}_{i,j,k'}V^n(V^n_{j,p}, V^n_{j,p+1})$ and $\xi^n_{j,k} = \xi^n_{j,k}(\xi^{n-1}_{j,k}, \mathcal{I}_{i,j,k'}V^{n-1}(V^{n-1}_{j,p}, V^{n-1}_{j,p+1})) = \xi^n_{j,k}(V^{n-1}_{j,k}, V^{n-2}_{j,k}, \ldots)$ are defined in equations (3.45) and (3.42) respectively. For the special case when $V^{n+1}_{i,j,k} = \bar{V}^{n+1}_{i,j,k}$, we can write the solution process as a single step (from equation (3.46)):

$$g_{i,j,k}\left(V^{n+1}_{i,j,k}, \mathcal{I}_{i,j,k'}V^n, \xi^n_{j,k}\right) = \left[V^{n+1}_{j,k}\right]_i - \left[\mathcal{M}^{-1}\mathcal{I}_{i,j,k'}V^n + \Delta\tau\mathcal{M}^{-1}\xi^n_{j,k}\right]_i = 0, \qquad (3.51)$$

assuming fully implicit timestepping is used, where the matrix $\mathcal{M} = I + \mathcal{D}$ and the matrix $\mathcal{D}$ is defined in equation (3.43). Since $\mathcal{M}$ is an M-matrix, $g_{i,j,k}$ is a decreasing function of $\xi^n_{j,k}$. Furthermore, $\mathcal{I}_{i,j,k'}V^n$ is a monotone function of $V^n_{j,k}$ if linear interpolation is used (see equation (3.45)), which we assume to be the case. In this context, we can define the monotonicity requirement as follows.

Definition 3.1 (Monotonicity). *The numerical scheme* $g_{i,j,k}(V^{n+1}_{i,j,k}, \mathcal{I}_{i,j,k'}V^n, \xi^n_{j,k})$ *in equation* (3.51) *is monotone if for all* $\epsilon^n_{i,j,k} \geq 0$*:*

$$\begin{aligned} &g_{i,j,k}\left(V^{n+1}_{i,j,k}, \mathcal{I}_{i,j,k'}V^n(V^n_{j,p} + \epsilon^n_{j,p}, V^n_{j,p+1} + \epsilon^n_{j,p+1}), \xi^n_{j,k}\big(V^{n-1}_{j,k} + \epsilon^{n-1}_{j,k}, V^{n-2}_{j,k} + \epsilon^{n-2}_{j,k}, \ldots\big)\right) \\ &\quad - g_{i,j,k}\left(V^{n+1}_{i,j,k}, \mathcal{I}_{i,j,k'}V^n(V^n_{j,p}, V^n_{j,p+1}), \xi^n_{j,k}\big(V^{n-1}_{j,k}, V^{n-2}_{j,k}, \ldots\big)\right) \leq 0 \end{aligned} \qquad (3.52)$$

where $\epsilon^n_{j,k} = [\epsilon^n_{0,j,k}, \epsilon^n_{1,j,k}, \ldots, \epsilon^n_{j_{\max},j,k}]'$.

We will show that equation (3.52) does not hold when $V^{n+1}_{i,j,k} = \bar{V}^{n+1}_{i,j,k}$.

Lemma 3.2 (Non-Monotone Discretization). *If*

$$\bar{V}^{n+1}_{i,j,k} > \mathcal{A}^h V^{n+1}_{i,j,k} \,, \tag{3.53}$$

$$\xi^n_{i,j,k} > 0 \;\; \textit{and} \;\; \xi^{n-1}_{i,j,k} = 0 \tag{3.54}$$

for all i, the numerical scheme $g_{i,j,k}(V^{n+1}_{i,j,k}, \mathcal{I}_{i,j,k'}V^n, \xi^n_{j,k})$ in equation (3.51) *is not monotone even if linear interpolation (presented in equation* (3.38)*) is used to obtain the semi-Lagrangian term. In this case, the requirement outlined in equation* (3.52) *from Definition 3.1 is not satisfied.*

Proof. A necessary condition for monotonicity is that the auxiliary variable $\xi^n_{i,j,k}$ be an increasing function of $V^{n-1}_{i,j,k'}$. When equations (3.53) and (3.54) are satisfied, $V^{n+1}_{i,j,k} = \bar{V}^{n+1}_{i,j,k}$ and the pricing equation for $V^{n+1}_{i,j,k}$ can be written as (from equation (3.46)):

$$[V^{n+1}_{j,k}]_i = \left[\mathcal{M}^{-1}(\mathcal{I}_{i,j,k'}V^n + \Delta\tau\xi^n_{j,k})\right]_i \,, \tag{3.55}$$

where $\mathcal{I}_{i,j,k'}V^n$ is defined in equation (3.45). Since $\mathcal{M}$ is an M-matrix, the entries in $\mathcal{M}^{-1}$ are non-negative with positive diagonals.

Since $\xi^n_{i,j,k} > 0$, Algorithm 3.1 implies that $\bar{V}^n_{i,j,k} \leq \mathcal{A}^h V^n_{i,j,k}$ and:

$$\xi^n_{i,j,k} = \xi^{n-1}_{i,j,k} + \frac{1}{\Delta\tau}\left(\bar{V}^n_{i,j,k} - \mathcal{A}^h V^n_{i,j,k}\right), \tag{3.56}$$

where $\bar{V}^n_{i,j,k}$ is obtained as in (3.46). Since $\xi^{n-1}_{i,j,k} = 0$ and $\xi^n_{i,j,k} > 0$ by assumption, Algorithm 3.1 implies that $V^n_{i,j,k} = \mathcal{A}^h V^n_{i,j,k}$.

For all i, we choose $\epsilon_{i,j,p}^{n-1} > 0$ sufficiently large such that:

$$\begin{aligned}
[\bar{V}_{i,j,k}^{n}]_{\text{perturbed}} &= \Big[\mathcal{M}^{-1}\big(a(V_{j,p}^{n-1} + \epsilon_{j,p}^{n-1}) + b(V_{j,p+1}^{n-1})\big) + \Delta\tau\xi_{j,k}^{n-1}\Big)\Big]_i \\
&= \Big[\mathcal{M}^{-1}\big(a(V_{j,p}^{n-1} + \epsilon_{j,p}^{n-1}) + b(V_{j,p+1}^{n-1})\big)\Big]_i \\
&> \mathcal{A}^h V_{i,j,k}^{n} \qquad (3.57)
\end{aligned}$$

where $\mathcal{I}_{i,j,k'}V^{n-1} = aV_{j,p}^{n-1} + bV_{j,p+1}^{n-1}$ from (3.45). This implies that $[\xi_{i,j,k}^{n}]_{\text{perturbed}} = 0$.

Consequently, we have shown that, for $\epsilon_{i,j,p}^{n-1}$ sufficiently large:

$$\xi_{i,j,k}^{n}\big(\xi_{j,k}^{n-1}, \mathcal{I}_{i,j,k'}V^{n-1}(V_{j,p}^{n-1} + \epsilon_{j,p}^{n-1}, V_{j,p+1}^{n-1})\big) - \xi_{i,j,k}^{n}\big(\xi_{j,k}^{n-1}, \mathcal{I}_{i,j,k'}V^{n-1}(V_{j,p}^{n-1}, V_{j,p+1}^{n-1})\big) < 0,$$

and we get:

$$\begin{aligned}
&g_{i,j,k}\left(V_{i,j,k}^{n+1}, \mathcal{I}_{j,k'}V^{n}\big(V_{j,p}^{n}, V_{j,p+1}^{n}\big), \xi_{j,k}^{n}\big(V_{j,k}^{n-1} + \epsilon_{j,k}^{n-1}, V_{j,k}^{n-2}, \ldots\big)\right) \\
&\quad - g_{i,j,k}\left(V_{i,j,k}^{n+1}, \mathcal{I}_{i,j,k'}V^{n}\big(V_{j,p}^{n}, V_{j,p+1}^{n}\big), \xi_{j,k}^{n}\big(V_{j,k}^{n-1}, V_{j,k}^{n-2}, \ldots\big)\right) > 0, \qquad (3.58)
\end{aligned}$$

which contradicts equation (3.52) in Definition 3.1. Hence, $g_{i,j,k}(V_{i,j,k}^{n+1}, \mathcal{I}_{i,j,k'}V^{n}, \xi_{j,k}^{n})$ is not monotone. □

This counter-example shows that the operator splitting scheme is not monotone in general, and hence, convergence to the viscosity solution is not guaranteed.

3.2.2 Comments on the Solution Method

In this section, we discuss some of the numerical issues that were encountered when pricing infinite reload options with vesting. When first attempting to price these contracts, we chose to determine their numerical value using the penalty method outlined in [44]. Recall that this solution method was used successfully in Chapter 2 when pricing classic

infinite reload option contracts [13]. However, a serious setback was encountered with this iterative scheme. It was observed that a very large number of iterations were necessary to obtain convergence of the solution at each timestep. This led us to consider alternate solution methods for handling the reload constraint.

Consequently, the operator splitting method presented in [55] and [56] was chosen to solve the infinite reload pricing problem with vesting. While this method provided a satisfactory alternative to the iterative penalty scheme, some drawbacks were also observed. In addition to the issues outlined in Section 3.2.1, less than ideal convergence ratios were obtained. While the authors of [56] claim that quadratic convergence is possible when the method uses Crank-Nicolson timestepping, we were not able to generate a consistent quadratic convergence rate when pricing a simple American put option with the operator splitting method. As shown in Appendix E, only sub-quadratic convergence is obtained when pricing an American put option (without vesting) using Crank-Nicolson. Note that this is consistent with the results in [56] where the authors obtain rather erratic convergence ratios when pricing American options with stochastic volatility using the operator splitting method. The authors claim that the solution is not smooth enough with respect to time to provide quadratic convergence when Crank-Nicolson timestepping is used.

Since quadratic convergence was not obtained when pricing a simpler contract such as the American put option, it is unlikely that second order convergence will be observed when pricing infinite reload options using the operator splitting solution method and Crank-Nicolson timestepping.

3.3 Numerical Results

In this section, numerical results for infinite reload options with different vesting periods are presented. All results were computed using the operator splitting method. We begin

Parameter	Value
σ - Volatility	0.3
r - Risk-free interest rate	0.05
q - Dividend yield	0.0
K - Initial strike price	\$100
S - Initial asset price	\$100
T - Contract maturity	10 years
v_p - Vesting Period	0.5 years
$K_{\max}$ - Grid parameter	\$2000
$S_{\max}$ - Grid parameter	\$40000

TABLE 3.1: *Parameter values used when pricing infinite reload options with vesting.*

by carrying out a convergence analysis and then move on to consider the effect of the length of the vesting period on the option value. Also, we demonstrate the robustness of our pricing model by comparing the numerical results obtained here with those presented in [39] and [33]. The parameter values chosen when generating these results are presented in Table 3.1 and will be used throughout this section unless otherwise noted.

Table 3.2 holds the results of the convergence analysis for an infinite reload option contract with a vesting period of 0.5 years. The underlying grid is built as described in Section 3.2 and a similarity reduction is used in the option value calculation. Furthermore, quadratic interpolation is chosen for the semi-Lagrangian term calculation and diagonal interpolation is used when calculating the reload constraint. Constant timesteps are used for both fully implicit and Crank-Nicolson timestepping in accordance with the results in Appendix E.

From Table 3.2, we see that linear convergence is obtained when fully implicit timestepping is used. This result is consistent with the theoretical convergence of this timestepping method. However, sub-quadratic convergence is obtained when Crank-Nicolson timestepping is chosen even though diagonal interpolation is used to calculate the reload constraint. As noted previously, we attribute this less than theoretical convergence to the

Infinite Reload Option with a $v_p = 0.5$ years					
	Nodes				
Refinement	K Grid	u Grid	Option Value	Difference	Ratio
Fully Implicit (implicit constraint)					
0	61	51	63.070009	n.a.	n.a.
1	121	101	63.187814	0.117805	n.a.
2	241	201	63.236508	0.048694	2.42
3	481	401	63.258609	0.022101	2.20
4	961	801	63.269171	0.010562	2.09
Crank-Nicolson (implicit constraint)					
0	61	51	63.199299	n.a.	n.a.
1	121	101	63.254138	0.054840	n.a.
2	241	201	63.270394	0.016255	3.37
3	481	401	63.275846	0.005452	2.98
4	961	801	63.277916	0.002071	2.63

TABLE 3.2: *Value of an infinite reload option at* $S = \$100$ *with a vesting period of* 0.5 *years using fully implicit and Crank-Nicolson timestepping for different refinement levels. Constant timesteps are taken in both cases and the initial timestep is set to* 0.05 *years on the coarsest grid. Other parameter values are presented in Table 3.1. Note that a similarity reduction was used when computing these results.*

use of the operator splitting method as well as the complex nature of the reload constraint applied at each timestep.

The results presented in Table 3.2 are consistent with those presented in [33] and [39]. Using the parameters in Table 3.1, the authors of [39] use extrapolation to establish the contract value as \$63.26, while the authors of [33] determine the option value as being \$63.28. To further compare our results with those obtained in [33] and [39], we consider alternate volatility and vesting period values. Table 3.3 presents the value of an infinite reload option contract with a vesting period of either 0.5 or 1 year and a volatility of either 0.3 or 0.4. The results obtained in each case are consistent with the values presented in [33] and [39]. Note that fully implicit timestepping is used since both timestepping methods were found to provide first-order convergence.

Since the implicit application of the reload constraint within the operator splitting

	$v_p = 0.5$ years		$v_p = 1$ year	
Refinement	$\sigma = 0.3$	$\sigma = 0.4$	$\sigma = 0.3$	$\sigma = 0.4$
0	63.070009	71.601479	61.604386	70.102681
1	63.187814	71.689259	61.695624	70.167970
2	63.236508	71.732622	61.731127	70.198941
3	63.258609	71.754397	61.746449	70.214171
4	63.269171	71.765399	61.753563	70.221788
Values from [33]	63.28	71.78	61.76	70.23
Values from [39]	63.26	71.76	61.78	70.21

TABLE 3.3: *Value of an infinite reload option with a maturity of* 10 *years for different vesting period lengths* (v_p) *and different volatility values* (σ). *Fully implicit timestepping is used with an initial timestep of* 0.05 *years on the coarsest grid. Other parameter values chosen are presented in Table 3.1. For comparison, the last two rows present numerical results obtained in Dai and Kwok [33] and Dybvig and Loewenstein [39].*

method does not provide quadratic convergence, the explicit application of the constraint is considered as a possible alternative. Applying the reload constraint explicitly implies first solving:

$$[I + (1-\theta)\mathcal{D}]\bar{V}_{j,k}^{n+1} = V_{j,k'}^{n} - \theta\mathcal{I}_{i,j,k'}\mathcal{D}V^{n} \tag{3.59}$$

to determine the intermediate solution, where $\mathcal{D}$ is defined in equation (3.43), $0 \leq \theta \leq 1$ is determined by the timestepping method and $\mathcal{I}_{i,j,k'}V^n$ is calculated as in (3.45). The reload constraint is then applied component-wise:

$$V_{i,j,k}^{n+1} = \max(\bar{V}_{i,j,k}^{n+1}, \mathcal{A}^h V_{i,j,k}^{n+1}). \tag{3.60}$$

An advantage of this explicit constraint scheme is that it is easy to show that this method (in the fully implicit case) is l_∞-stable and monotone (assuming linear interpolation is used when computing the semi-Lagrangian term). Consistency can be shown using an argument similar to that in Chapter 2 for increased reload options. Hence, we are guaranteed that this scheme converges to the unique viscosity solution.

Table 3.4 presents values obtained when pricing an infinite reload option with $v_p = 0.5$

Infinite Reload Option with a $v_p = 0.5$ years					
	Nodes				
Refinement	K Grid	u Grid	Option Value	Difference	Ratio
Fully Implicit (explicit constraint)					
0	61	51	63.039664	n.a.	n.a.
1	121	101	63.172646	0.132982	n.a.
2	241	201	63.228925	0.056280	2.36
3	481	401	63.254818	0.025893	2.17
4	961	801	63.267276	0.012458	2.08
Crank-Nicolson (explicit constraint)					
0	61	51	63.187866	n.a.	n.a.
1	121	101	63.248805	0.060939	n.a.
2	241	201	63.267922	0.019117	3.19
3	481	401	63.274711	0.006789	2.82
4	961	801	63.277392	0.002681	2.53

TABLE 3.4: *Value of an infinite reload option at* $S = \$100$ *with a vesting period of* 0.5 *years where the reload constraint is applied explicitly. Constant timesteps are used for both timestepping methods and the initial timestep is set to* 0.05 *years on the coarsest grid. Other parameter values are presented in Table 3.1. Note that the similarity reduction property was used when obtaining these results.*

years and applying the reload constraint explicitly. Quadratic interpolation is used to calculate the semi-Lagrangian term and diagonal interpolation is used to compute the reload constraint. As expected, both the fully implicit and Crank-Nicolson timestepping schemes provide first-order convergence. When comparing the results in Table 3.4 with those in Table 3.2, we see that the accuracy of the option values are comparable. This is especially true for higher refinement levels (i.e. refinement of 3 or 4). As such, it would appear that the explicit application of the reload constraint is an acceptable alternative for those looking for a solution method that is simpler to implement and also has guaranteed convergence properties (if linear interpolation is used). We also increased both $K_{\max}$ and $S_{\max}$ and found that when $K_{\max} = \$2000$ the numerical results are accurate for 8 digits. In the remainder of this chapter, the numerical results are generated using the operator splitting scheme.

We now consider the effect of the length of the vesting period on the infinite reload option value. Table 3.5 presents values of infinite reload options with a vesting period of 0.5, 1 and 3 years. For comparison purposes, numerical values for the standard infinite reload option contract (no vesting period) and the classic employee stock option (similar to a simple European call option) are also included in Table 3.5. Clearly, the addition of a vesting period to infinite reload option contracts results in a significant reduction of its no-arbitrage price.

More specifically, adding a vesting period reduces the premium associated with the infinite reload right. According to the results in Table 3.5, the difference between the classic employee stock option and the infinite reload option value is about \$14. This amount corresponds to the premium paid for the right to reload at any time prior to maturity. Adding a vesting period of 1 year reduces the option value by about \$5 which corresponds to a reduction of the reload premium by about 35%. However, this contract modification does not appear to be as effective in reducing the option value as increasing the reload strike [13]. As shown in Chapter 2, increasing the strike of new reload options by only 1% reduces the option value to about \$61.70 resulting in a similar impact as the addition of a vesting period of 1 year. From the point of view of the employee, increasing the strike price of new reload options by 1% is probably more appealing than including a vesting period of 1 year to the reload contract and it may be easier to negotiate its acceptance with employees.

More information on the effect of the vesting period on the value of an infinite reload option can be found in Figure 3.2 which plots the option value as a function of the vesting period length. The plot in Figure 3.2 reveals that adding a vesting period of more than 5 years will reduce the option value to that of a classic employee stock option with identical maturity (see numerical value in Table 3.5). Indeed, when $v_p > 5$ years, the reload option essentially becomes a classic stock option since the owner can never reload his option as

S	Vesting Period v_p (in years) 0.0	0.5	1.0	3.0	Classic Employee Option
90	56.85	53.65	52.27	48.87	44.29
100	66.76	63.28	61.75	57.92	52.57
110	76.76	73.06	71.41	67.19	61.10

TABLE 3.5: *Value of an infinite reload option with a maturity of* 10 *years at different* S *values for vesting periods* (v_p) *ranging from* 0.5 *to* 3 *years. For comparison purposes, the value of a classic employee stock option is also included. Note that the numerical results shown are precise to within* \$0.01. *The parameter values used in the calculations are presented in Table 3.1.*

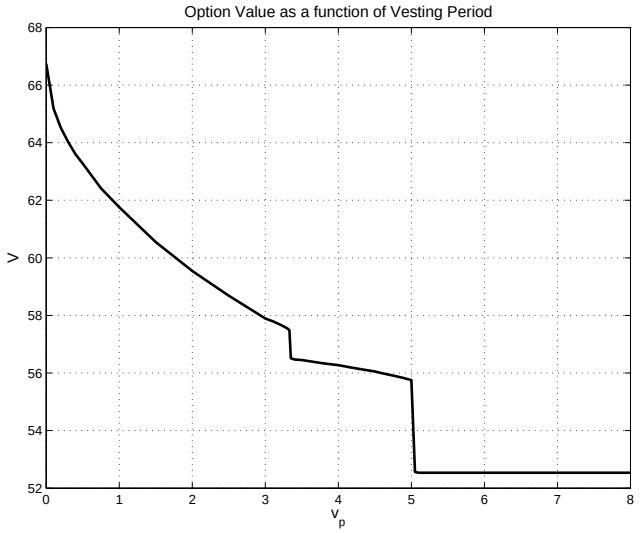


FIGURE 3.2: *Value of an infinite reload option as a function of the length of the vesting period* (v_p). *Contract parameters used during the pricing process are presented in Table 3.1.*

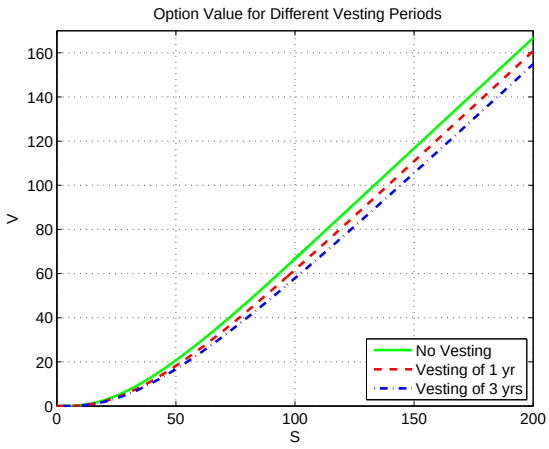
it would then expire prior to the end of the vesting period.

Also, consider the two price drops that occur at approximately 3.33 years and 5 years respectively. These sharp price reductions occur when the number of possible reloads during the lifetime of the contract is reduced. The first drop occurs at $v_p = 3.33$ years which corresponds to the time when the number of possible reloads drops from 2 to 1. Similarly, the price of the contract drops sharply at $v_p = 5$ years since the owner loses all reload possibilities when $v_p > 5$.

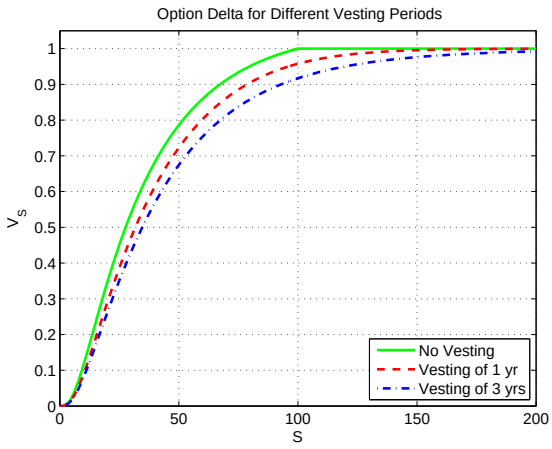
Lastly, we study the effect of the vesting period length on the option value (V), the delta (V_S) and the gamma (V_{SS}). Figure 3.3 presents these three quantities for infinite reload option contracts with a vesting period of 1 and 3 years. For comparison purposes, the curves obtained for classic infinite reload options (*No vesting*) are also included. While the shape of the option value curve is not significantly affected by the vesting period (Figure 3.3(a)), the shape of the delta curve changes when a vesting period is introduced, as shown in Figure 3.3(b). The plot of the delta for the classic infinite reload option contains a kink at the strike which disappears with the addition of a vesting period. Consequently, the discontinuity in the gamma observed for classic infinite reload options disappears when a vesting period is added (see Figure 3.3(c)). For infinite reload options without vesting, recall that the discontinuity in the gamma represents the optimal exercise policy whereby the owner should reload whenever $S > K$ [39]. Clearly, this optimal exercise policy no longer applies when vesting periods are introduced; this is reflected in the smooth gamma curves observed when $v_p = 1$ year and $v_p = 3$ years. This result is consistent with the comments in [39] and [33] regarding the optimal exercise policy. The authors of [39] and [33] both state that the optimal exercise policy is affected by the introduction of a vesting period, even resulting in periods during the lifetime of the contract where it is not optimal to exercise at all.

3.4 Summary

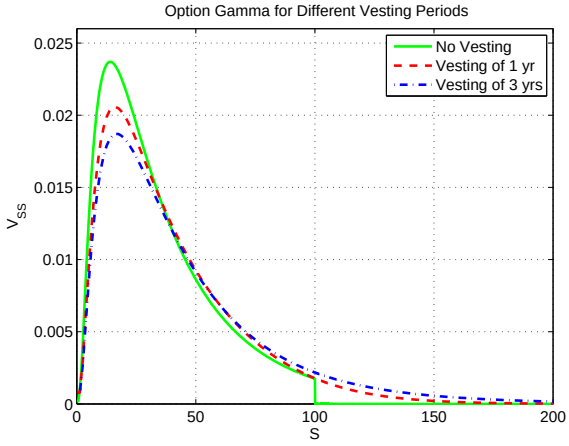
In this chapter, we have extended the no-arbitrage pricing model for infinite reload options presented in Chapter 2 by taking into consideration possible vesting periods. Since the modelling of this feature results in the addition of a new path-dependent variable, the appropriate pricing equation for infinite reload options with vesting was derived. Due to the convection-dominated nature of the pricing equation, a semi-Lagrangian scheme was



(A) Option value for different v_p values.



(B) Option delta for different v_p values.



(C) Option gamma for different v_p values.

FIGURE 3.3: *Plot of the option value (V), the delta (V_S) and the gamma (V_{SS}) functions when pricing an infinite reload option with different vesting periods. The contract parameters used during the pricing process are presented in Table 3.1.*

used during the discretization process. While the impulse control problem considered could have been solved using the penalty method [44], an operator splitting method was chosen due to the large number of iterations required when solving the pricing problem

with the penalty method. It was shown that the resulting numerical scheme is not monotone and, therefore, convergence of the solution to the unique viscosity solution cannot be guaranteed. Our numerical results where nonetheless consistent with those in [39] and [33]. Numerical results obtained using the operator splitting method with an implicit application of the reload constraint were also compared to those recovered when the constraint is applied explicitly. The values obtained from the explicit scheme were found to be comparable in terms of accuracy to those from the operator splitting scheme.

While the introduction of a vesting period reduces the no-arbitrage price of an infinite reload option, the price drop was not as significant as one would have expected. This is especially true when this contract modification is compared to increasing the strike of new reload options (Chapter 2). Indeed, increasing the reload strike by only 1% has the same impact as adding a vesting period of 1 year and is probably more appealing to employees. However, the price difference remains non-negligible which implies that the vesting period of an infinite reload option should not be ignored during the pricing process, if such a feature is included in the option contract.

Chapter 4

Expected Utility Pricing of Infinite Reload Options

One of the main issues when using stock options as a compensation tool is the well-established notion that employees place a lower value on stock options than the no-arbitrage price of the contract, as discussed in [51]. Having been demonstrated repeatedly in previous literature (see [70], [50], [64], [22]), this appears to be a consequence of the trading restrictions imposed on the employee regarding his or her company stock. While a no-arbitrage pricing model for infinite reload options was developed in Chapter 2 and extended in Chapter 3 by adding vesting periods, this chapter focuses on determining the value of employee stock options, and infinite reload options in particular, to the holder of the contract.

A substantial amount of literature is available regarding the pricing of classic vanilla-type stock options from the holder's point of view. Using different pricing techniques, both [60] and [70] show that the value of a classic stock option to the holder is significantly less than the no-arbitrage price computed using the Black-Scholes approach. Both [83] and

[50] determine the value of classic vanilla-type stock options using a certainty equivalence approach. In [83], the authors determine the incentive effects of classic stock options, while [50] tries to shed light on various related issues such as option repricing and exercise price policies. In [23], the author compares the value obtained for classic stock options using a certainty equivalence approach with those obtained using an extension of the American pricing model and finds that they are quite similar. Finally, in [22] the authors attempt to reduce the gap between the objective and subjective option value by suggesting a particular hedging strategy.

The main contributions of this chapter can be summarized as follows:

- A general utility-based pricing framework is developed to determine the value of a stock option to the holder. Our pricing model is a multi-step process where two expected utility values are computed. The first step involves solving a convection-dominated partial differential equation which is discretized using a semi-Lagrangian scheme. The second step determines the cash equivalent option value by using Monte Carlo simulations or an analytical solution when applicable.

- To highlight the versatility of our pricing model, both classic stock options and infinite reload options are valued from the holder's perspective.

- While we noted in Chapter 2 that the reload feature significantly increases the no-arbitrage price of an infinite reload option, we now demonstrate that the infinite reload opportunities are significantly undervalued by the contract owner. Furthermore, the contract value to the employee is shown to be highly sensitive to assumptions made regarding the portfolio content of the contract owner.

4.1 Mathematical Model

In this section, we outline details of the mathematical model used to determine the value of employee stock options to contract holders. The pricing model adopts a utility maximizing or certainty equivalence approach [64, 50]. We assume that the employee's total wealth or portfolio, denoted by $\mathcal{W}$, is composed of n_s units of company stock, n_o stock options and some amount of cash w held in a risk-free bank account. The implied assumption is that the employee does not invest in any other market instruments. Keep in mind that executives are generally restricted in trading their employer's stock. Hence, the employee's total wealth at time t, denoted as $\mathcal{W}(S, K, w, t)$, can be written as:

$$\mathcal{W}(S, K, w, t) = n_s S + n_o V(S, K, t) + w, \tag{4.1}$$

where S is the current price of the company stock, K is the option strike price, t is forward time and $V(S, K, t)$ represents the value of a single stock option.

We assume that the employee has constant relative risk aversion (as in [64]) which results in a power utility function denoted as:

$$\mathcal{U}(\mathcal{W}) = \frac{\mathcal{W}^{1-\alpha}}{1-\alpha}, \tag{4.2}$$

where $\alpha > 1$ is the constant relative risk aversion coefficient of the contract holder.

To determine the option value to the contract holder, we apply a certainty equivalence approach [64, 50]. This method determines the upfront cash payment that, when given to the executive as part of his or her compensation package instead of employee stock options, provides the equivalent expected utility level. This cash payment provides an estimate of the intrinsic value of the stock options to the holder on the grant date.

More specifically, we are looking for the cash amount, denoted as c, that provides the

same expected utility as when the employee is given stock options. Mathematically, we have:

$$E^P[\mathcal{U}(\mathcal{W}(S,K,w,t=T))] = E^P[\mathcal{U}(\mathcal{W}_c(t=T))], \tag{4.3}$$

where $E^P[\cdot]$ denotes the expected value under the P measure [77], $\mathcal{W}_c(t) = n_s S(t) + w(t) + e^{rt}c$ is the portfolio value of the employee assuming cash is received instead of options and the utility function $\mathcal{U}(\mathcal{W})$ is defined in (4.2). The cash amount c will represent the private value of n_o stock options to the holder of the contract[1].

Solving equation (4.3) numerically is a multi-step process since it requires the computation of two expected utility values. We evaluate the left-hand side of equation (4.3) by solving a partial differential equation (PDE) to obtain an approximation of the employee's expected utility when given employee stock options. We then evaluate the right-hand side of equation (4.3) using Monte Carlo simulations or an analytical expression when applicable to estimate the employee's utility when the stock options are replaced by cash. Alternatively, computing the right-hand side of equation (4.3) could be done with PDEs. However, this alternate approach would require the repeated solution of a PDE with at least two time-dependent variables which can be computationally expensive. As such, we prefer to use an analytical solution or Monte Carlo simulations to evaluate the right-hand side of equation (4.3). Having determined an initial estimate for the expected utility, Newton's method is then used to obtain a more precise approximation of c. Each step is described in more detail in the following sections.

[1]Note that there exists a unique solution c to equation (4.3) for a given stock price $S(T)$. In that case, the expected value on the right-hand side of equation (4.3) becomes a constant function which crosses the curve of $E^P[\mathcal{U}(\mathcal{W}(S,K,w,t=T))]$ from the left-hand side of equation (4.3) at a single point.

4.1.1 Step 1: Solving the PDE

We assume that the company stock price S follows the standard Brownian motion process:

$$dS = S\mu dt + S\sigma dZ, \tag{4.4}$$

where μ is the drift rate, σ is the volatility and Z is a Wiener process. Taking into consideration the Capital Asset Pricing Model [85], the drift rate is set to:

$$\mu = r + \beta(r_m - r) - q, \tag{4.5}$$

where r is the risk-free interest rate, r_m is the expected market return, β is the firm's systematic risk measure and q is the continuous dividend yield.

We assume that any dividends received by the owner are deposited in the risk-free bank account containing the employee's external wealth w. As such, the process for w is:

$$dw = (wr + qn_sS)dt. \tag{4.6}$$

While we assume that dividends are paid continuously, our pricing model could be extended to consider discrete dividend payments as well.

Applying Ito's lemma [85, 54], we find that the employee's expected utility, denoted by $U = U(S, K, w, t)$, satisfies the drift-dominated PDE (see [75]):

$$U_t + \mu SU_S + \frac{1}{2}\sigma^2 S^2 U_{SS} + (wr + qn_sS)U_w = 0. \tag{4.7}$$

Defining τ as the time to expiry ($\tau = T - t$), we can rewrite equation (4.7) as :

$$U_\tau = \mu SU_S + \frac{1}{2}\sigma^2 S^2 U_{SS} + (wr + qn_sS)U_w. \tag{4.8}$$

While similar to the classic Black-Scholes equation, equation (4.8) does not include a discounting term since we are determining the employee's expected utility at maturity (when $t = T$). Also, notice that equation (4.8) is convection-dominated since there is no diffusion term in the w direction.

Although equation (4.8) is enough to determine the employee's expected utility when given simple stock options, we may need to impose additional American-type constraints when valuing more complex contracts such as reload options. In such cases, the pricing problem becomes an impulse control problem resulting in a set of quasi-variational inequalities. As in Chapters 2 and 3, we let $\mathcal{A}U = \mathcal{A}U(S, K, w, t)$ denote the American-type constraint applied when the employee owns more exotic stock options. In the case of reload options for example, $\mathcal{A}U$ corresponds to the employee's expected utility following a reload event.

Defining the differential operator $\mathcal{L}$ as:

$$\mathcal{L}U \equiv \mu S U_S + \frac{1}{2}\sigma^2 S^2 U_{SS}, \tag{4.9}$$

we can write the utility maximizing problem as an impulse control problem:

$$\min\left(U_\tau - (wr + qn_s S)U_w - \mathcal{L}U, U - \mathcal{A}U\right) = 0. \tag{4.10}$$

Equation (4.10) can also be written as a penalized problem:

$$\lim_{\epsilon \to 0}\left(U_\tau - (wr + qn_s S)U_w - \mathcal{L}U - \frac{1}{\epsilon}\max(\mathcal{A}U - U, 0)\right) = 0. \tag{4.11}$$

To determine the employee's expected utility value, equation (4.11) is solved numerically working backward in time using a penalty method [44].

To localize the utility maximizing problem, we need to specify boundary and initial

conditions for equation (4.8) (or (4.10) as the case may be). The initial utility values at $\tau = 0$ are calculated using the payoff of the stock option:

$$\begin{aligned} U(S, K, w, \tau = 0) &= U(\mathcal{W}(S, K, w, \tau = 0)) \\ &= U(n_s S + n_o \text{ Payoff}(S, K) + w), \end{aligned} \tag{4.12}$$

where:

$$\text{Payoff}(S, K) = \max(0, S - K), \tag{4.13}$$

for classic employee stock options[2].

Depending on the stock option considered, the employee's expected utility is a function of two or more path-dependent variables, as well as time. The notation used thus far assumes a rather complex case where the expected utility is a function of three variables and time, namely S, K and w: $U = U(S, K, w, \tau)$. For simple contracts, such as classic employee stock options, the dimensionality of the problem may be reduced. Therefore, the boundary conditions necessary to fully define the pricing problem will be determined on a case-by-case basis depending on the dimensionality of the pricing problem considered.

4.1.2 Semi-Lagrangian Solution

Recognizing that equation (4.8) is convection-dominated, we use a semi-Lagrangian discretization [36] when solving the utility maximizing problem in equation (4.10) numerically[3].

The solution to equation (4.10) can be approximated by a penalty method [44] which

[2]Classic stock options are similar to vanilla European call options.

[3]See Section 3.2 for more information on semi-Lagrangian schemes.

results in a non-linear PDE:

$$U_\tau - (wr + qn_sS)U_w = \mathcal{L}U + P(U, \mathcal{A}U), \tag{4.14}$$

where $\mathcal{L}U$ is defined in equation (4.9) and $P(U, \mathcal{A}U)$ represents the penalty term which imposes the American-type constraint $\mathcal{A}U$.

To discretize equation (4.14) adequately, we assume that an unequally spaced grid has been constructed in each of the S, K and w directions. Along the S direction, the grid is built over the range $[0, S_{\max}]$ and the i^{th} node is denoted as S_i. Similar grids are built along the K and w directions. Let $U^n_{i,j,k} = U(S_i, K_j, w_k, \tau^n)$ denote the discrete value of the employee utility when the current strike price is K_j and the non-firm related wealth at time τ^n is w_k.

We now outline details of the semi-Lagrangian discretization. Let us consider a trajectory in the w direction denoted as $\hat{w} = \hat{w}(\tau; S, K)$ where S and K are fixed. From equation (4.6), the values along the trajectory $\hat{w}$ satisfy:

$$\frac{d\hat{w}}{d\tau} = -(wr + qn_sS). \tag{4.15}$$

Defining the Lagrangian derivative as:

$$\frac{DU}{D\tau} = \frac{\partial U}{\partial \tau} + \frac{\partial U}{\partial w}\frac{d\hat{w}}{d\tau}, \tag{4.16}$$

we can write equation (4.14) as:

$$\frac{DU}{D\tau} = \mathcal{L}U + P(U, \mathcal{A}U). \tag{4.17}$$

We assume that the trajectory of $\hat{w}$ values arrives at the discrete node (S_i, K_j, w_k) at

time $\tau = \tau^{n+1}$. We are looking to determine the departure point of the trajectory at time $\tau = \tau^n$ which is denoted as $\hat{w}(\tau^n) = \hat{w}(\tau^n; S_i, K_j, w_k, \tau^{n+1})$. The departure point $\hat{w}(\tau^n)$ does not necessarily coincide with a grid node w_k since $\hat{w}(\tau^n)$ is determined by solving:

$$\begin{aligned} &\frac{d\hat{w}}{d\tau} = -wr - qn_sS_i, \\ &\quad S = S_i \ , \ K = K_j \ , \ \hat{w}(\tau) = w_k \text{ at } \tau = \tau^{n+1}, \end{aligned} \tag{4.18}$$

from $\tau = \tau^{n+1}$ to $\tau = \tau^n$. Therefore, we can estimate the initial wealth position $\hat{w}(\tau^n)$ by approximating the following integral along the characteristic curve numerically:

$$\hat{w}(\tau^n) = w_k - \int_{\tau^{n+1}}^{\tau^n} (wr + qn_sS)d\tau, \tag{4.19}$$

where $T \geq \tau^{n+1} > \tau^n$.

In this particular case, we can find the exact value for $\hat{w}(\tau^n)$ by solving equation (4.18). Considering S as a constant term, we find that:

$$\hat{w}(\tau) = \frac{\mathcal{C}e^{-r\tau} - qn_sS}{r}, \tag{4.20}$$

where $\mathcal{C}$ is a constant independent of w but a function of S and r. Using this result, we can find an expression for w_k at time $\tau = \tau^{n+1}$ and use it to obtain:

$$\text{At time } \tau^{n-1} : \hat{w}(\tau^{n-1}) = w_k + \frac{(w_kr + qn_sS_i)(e^{r(\tau^{n+1}-\tau^{n-1})} - 1)}{r} \tag{4.21}$$

$$\text{At time } \tau^{n} : \hat{w}(\tau^{n}) = w_k + \frac{(w_kr + qn_sS_i)(e^{r(\tau^{n+1}-\tau^{n})} - 1)}{r} \tag{4.22}$$

Equations (4.21) and (4.22) are valid for $T \geq \tau^{n+1} > \tau^n > \tau^{n-1} \geq 0$ where T is the contract maturity.

Let $U^n_{i,j,k'} = U(S_i, K_j, \hat{w}(\tau^n), \tau^n)$ denote the utility value at the departure point of

the characteristic curve $\hat{w}(\tau^n)$. Limited quadratic interpolation along the w-axis can be used when calculating $U^n_{i,j,k'}$ [41]. Note that linear interpolation can also be used to ensure monotonicity of the discrete scheme. For ease of explanation, we assume linear interpolation is used in the following. Having determined the index p such that $w_p \leq \hat{w}(\tau^n) \leq w_{p+1}$, linear interpolation along the w-axis can be written as:

$$U^n_{i,j,k'} = (1-\xi)U^n_{i,j,p} + \xi U^n_{i,j,p+1}, \tag{4.23}$$

where the interpolation weight $0 \leq \xi \leq 1$ is defined as:

$$\xi = \frac{\hat{w}(\tau^n) - w_p}{w_{p+1} - w_p}. \tag{4.24}$$

The discretization of equation (4.17) along the characteristic curve gives:

$$\frac{U^{n+1}_{i,j,k} - U^n_{i,j,k'}}{\Delta\tau} = (1-\theta)[\mathcal{L}^h U]^{n+1}_{i,j,k} + \theta[\mathcal{L}^h U]^n_{i,j,k'} + P\big(U^{n+1}_{i,j,k}, \mathcal{A}^h U^{n+1}_{i,j,k}\big), \tag{4.25}$$

where $0 \leq \theta \leq 1$ determines the timestepping method used[4] and the discrete penalty term $P(U^{n+1}_{i,j,k}, \mathcal{A}^h U_{i,j,k})^{n+1}$ is defined as:

$$P\big(U^{n+1}_{i,j,k}, \mathcal{A}^h U^{n+1}_{i,j,k}\big) = L^{n+1}_{i,j,k}\left[\mathcal{A}^h U^{n+1}_{i,j,k} - U^{n+1}_{i,j,k}\right], \tag{4.26}$$

where

$$L^{n+1}_{i,j,k} = L\big(U^{n+1}_{i,j,k}, \mathcal{A}^h U^{n+1}_{i,j,k}\big) = \begin{cases} \frac{1}{\epsilon} & \text{if } \mathcal{A}^h U^{n+1}_{i,j,k} > U^{n+1}_{i,j,k}, \\ 0 & \text{otherwise.} \end{cases} \tag{4.27}$$

[4]Recall that $\theta = 0$ implies that fully implicit timestepping is used when solving equation (4.25), while $\theta = 1/2$ implies that Crank-Nicolson is chosen.

The discrete differential operator $\mathcal{L}^h$ can be written as:

$$[\mathcal{L}^h U]^{n+1}_{i,j,k} = \alpha_{i,j} U^{n+1}_{i-1,j,k} + \beta_{i,j} U^{n+1}_{i+1,j,k} - (\alpha_{i,j} + \beta_{i,j}) U^{n+1}_{i,j,k}, \tag{4.28}$$

where $\alpha_{i,j} \geq 0$ and $\beta_{i,j} \geq 0$ are determined according to the algorithm in Appendix A.

4.1.3 Step 2: Monte Carlo Simulations

In this section, we describe the next step of the pricing process where we determine the cash equivalent option value. Using the approach presented in Sections 4.1.1 and 4.1.2, we can determine the employee's expected utility when stock options are received as part of his or her compensation package, which we denote by u^*:

$$u^* = E^P[\mathcal{U}(\mathcal{W}(S, K, w, t = T))]. \tag{4.29}$$

We are now looking to determine the cash equivalent value c^* which satisfies:

$$E^P[\mathcal{U}(n_s S(T) + w(T) + e^{rT} c^*)] = u^*, \tag{4.30}$$

where $S(T)$ and $w(T)$ denote the value at maturity of the company stock and the employee's non-firm related wealth respectively. We choose to carry-out Monte Carlo simulations to solve equation (4.30) and determine c^*. Since equation (4.30) doesn't involve any optimal decisions, the use of Monte Carlo simulations is straight forward. Multiple asset paths are generated and an estimate for the employee's expected utility is obtained using an initial estimate for c^*. A more precise approximation of c^* is determined by using Newton's method.

Generating multiple asset paths translates into updating the following discrete equa-

tions at each timestep t_i, for $i = 0, \ldots, N$, with $t_N = T$:

$$S^{i+1} = S^i + S^i(\mu \Delta t_i + \sigma \phi^i \sqrt{\Delta t_i}), \tag{4.31}$$

$$w^{i+1} = w^i + (w^i r + q n_s S^i)\Delta t_i, \tag{4.32}$$

where the drift rate μ is defined in equation (4.5), ϕ^i is a random number drawn from a standard Normal distribution and $\Delta t_i = t_{i+1} - t_i$.

Since we only consider simple cases involving Brownian motion where μ and σ are constant, we can solve equation (4.31) exactly. Equation (4.31) is then replaced by:

$$S^{i+1} = S^i \exp[(\mu - \frac{\sigma^2}{2})\Delta t_i + \sigma \phi^i \sqrt{\Delta t_i}]. \tag{4.33}$$

However, keep in mind that equation (4.32) will not be solved exactly and will thus incur some timestepping error.

At maturity $t_N = T$, we calculate $\mathcal{U}_j(n_s S^N + w^N + e^{rT} c^*)$ for the jth asset path. Once M asset paths are generated, we compute an estimate for the expected utility:

$$E[\mathcal{U}(n_s(T)S(T) + w(T) + e^{rT} c^*)] \simeq \frac{1}{M} \sum_{j=1}^{M} \mathcal{U}_j(n_s S^N + w^N + e^{rT} c^*). \tag{4.34}$$

Using path-wise differentiation [48], we apply Newton's method and determine an approximation for c^* such that:

$$u^* = \frac{1}{M} \sum_{j=1}^{M} \mathcal{U}_j(n_s S^N + w^N + e^{rT} c^*). \tag{4.35}$$

The final c^* value represents the cash equivalent value to the employee of n_o stock options.

4.1.4 Special case when $q = 0$

When there are no dividends associated with the company stock (i.e. $q = 0$), the use of Monte Carlo simulations to solve equation (4.30) can be replaced by a closed-form solution which is evaluated using numerical integration techniques. This provides a more precise and efficient method for determining the certainty equivalence value of a given stock option contract.

Assuming $q = 0$, the wealth process in equation (4.6) becomes deterministic and simplifies to:

$$dw = wrdt. \tag{4.36}$$

Consequently, the employee's expected utility $U = U(S, w, \tau)$ now follows:

$$U_\tau = \mu S U_S + \frac{1}{2}\sigma^2 S^2 U_{SS} + wrU_w, \tag{4.37}$$

where $\mu = r + \beta(r_m - r)$. Since we are looking to solve equation (4.37) analytically, we closely follow the derivation of the solution to the Black-Scholes equation outlined in [85]. Details of the derivation can be found in Appendix F.

Note that this technique is not applicable when $q > 0$ due to the presence of a partial derivative term added to equation (4.37) containing two variables. Other solution methods, such as partial fractions, were considered for the more general case when $q > 0$ but to no avail.

As shown in Appendix F, the expected utility can be obtained by evaluating:

$$U(S, w, t) = \frac{1}{\sqrt{2\pi(T-t)}\sigma} \int_0^\infty e^{-\frac{(\ln(S/S') + (\mu - \frac{1}{2}\sigma^2)(T-t))^2}{2\sigma^2(T-t)}} \frac{(n_s S' + we^{rT} + ce^{rT})^{1-\alpha}}{1-\alpha} \frac{dS'}{S'}. \tag{4.38}$$

The integral in equation (4.38) can be evaluated numerically using a variety of numerical

integration techniques [5]. Evaluating equation (4.38) can be much more efficient then carrying out numerous Monte Carlo simulations and this analytical technique will be used whenever possible.

4.2 Valuing Classic Employee Stock Options

To validate our pricing model, we determine the cash equivalent value of classic employee stock options and compare our results to those in [50]. Recall that in the no-arbitrage context, a classic employee stock option has the same value as a European call option with identical maturity. We begin by reviewing the utility pricing problem for classic employee stock options (Section 4.2.1) and then present numerical results obtained with this model (Section 4.2.2).

4.2.1 Mathematical Model

As outlined in Section 4.1.2, the first step of the pricing process involves solving a PDE to determine the expected utility of the employee when given classic employee stock options as part of his or her compensation package. When valuing classic employee stock options, the expected utility depends on two path-dependent variables, namely S and w, as well as time: $U = U(S, w, \tau)$. Since there is no American-type constraint, equation (4.8) is solved numerically on a discrete two-dimensional $[0, S_{\max}] \times [0, w_{\max}]$ domain.

Additional boundary and initial conditions are necessary to fully specify the expected utility problem for classic stock options. The initial conditions are outlined in equation (4.12) where the option payoff corresponds to that of a standard call option in equation (4.13).

[5] We chose to use the numerical integration techniques included in the NAG Library of Numerical Algorithms [2].

Notice from equation (4.1) that the initial wealth $\mathcal{W}(S, K, w, 0) = 0$ when $S = w = 0$. This causes a fundamental problem when computing the utility at this point when $\alpha > 1$. To avoid numerical complications, the initial wealth is set to some small value $\mathcal{W}(0, K, 0, 0) = \delta$ when $S = w = 0$, where $0 < \delta \ll 1$. Therefore, the initial conditions can be written as:

$$U(S, w, \tau = 0) = U\left(\max(n_s S + n_o \text{ Payoff}(S, K) + w, \delta)\right). \tag{4.39}$$

Though not presented here, numerical tests were conducted to determine the appropriate value for δ. Based on these results, our choice of $\delta = 1 \times 10^{-8}$ ensures a minimum of 8 digits of accuracy in the numerical solution.

Additional boundary conditions are necessary along the S and w directions. At $S = 0$, we find that equation (4.8) becomes:

$$U_\tau = wrU_w \; ; \; S \to 0. \tag{4.40}$$

As $S \to \infty$, we adopt the common assumption that $U_{SS} = 0$ [89], which implies:

$$U \approx F(w, \tau)S + B(w, \tau), \tag{4.41}$$

where $F(w, \tau)$ and $B(w, \tau)$ are independent of S. We further assume $F(w, \tau)S \gg B(w, \tau)$ which leads to:

$$U \approx F(w, \tau)S. \tag{4.42}$$

Combining equation (4.42) with the differential equation in (4.8), we obtain:

$$F(w, \tau)_\tau S = \mu F(w, \tau)S + (wr + qn_s S)F(w, \tau)_w S. \tag{4.43}$$

Therefore, we obtain the following approximation as $S \to S_{\max}$:

$$U_\tau = \mu U + (wr + qn_s S)U_w. \tag{4.44}$$

Next, we consider the boundary conditions along the w direction. Assuming that $\alpha > 1$, we find that as wealth increases, the utility increases to zero. Therefore, we assume that $U \to 0$ as $w \to w_{\max}$. No boundary condition is required at $w = 0$ since this is an outflow boundary.

To summarize the first step of the pricing process, we solve:

$$U_\tau = \mu S U_S + \frac{1}{2}\sigma^2 S^2 U_{SS} + (wr + qn_s S)U_w, \tag{4.45}$$

on a discrete two-dimensional $[0, S_{\max}] \times [0, w_{\max}]$ domain with initial conditions:

$$U(S, w, \tau = 0) = U(\max(n_s S + n_o \max(S - K, 0) + w, \delta)), \tag{4.46}$$

where $0 < \delta \ll 1$ is some small value. The boundary conditions used are:

$$U_\tau - wrU_w = 0 \qquad \text{for } S = 0, \tag{4.47}$$

$$U_\tau - \mu U - (wr + qn_s S)U_w = 0 \qquad \text{for } S = S_{\max}, \tag{4.48}$$

$$U_\tau - \mathcal{L}U - (wr + qn_s S)U_w = 0 \qquad \text{for } w = 0, \tag{4.49}$$

$$U = 0 \qquad \text{for } w = w_{\max}, \tag{4.50}$$

where $\mathcal{L}U$ is defined in equation (4.9).

Once the employee's expected utility is obtained, we use Monte Carlo simulations exactly as described in Section 4.1.3 to determine the certainty equivalent cash value. No modifications are necessary to this second part of the pricing process. The cash amount

Portfolio Details	Case 1	Case 2	Case 3	Case 4
Employee risk-aversion coefficient - α	2.0	3.0	2.0	3.0
Number of options - n_o	5000	5000	5000	5000
Investment in stock - Sn_s	\$3.35 mil.	\$3.35 mil.	\$2.5 mil.	\$2.5 mil.
Initial non-firm related wealth - w	\$1.65 mil.	\$1.65 mil	\$2.5 mil.	\$2.5 mil.

TABLE 4.1: *Different employee situations or* Cases *as defined in [50]. Each case assumes one of two employee portfolio allocations (Sn_s, n_o and w) and one of two risk-aversion levels (α).*

obtained will correspond to the cash equivalent value of n_o stock options to the employee.

4.2.2 Numerical Results

This section presents numerical results obtained when determining the value of classic employee stock options to the contract holder. We begin by carrying out a convergence analysis of the utility values obtained from the first step of the pricing process, namely when solving the equation (4.45). We then determine the cash equivalent value of a classic stock option (using the numerical scheme presented in Section 4.2.1) for different portfolio compositions. To enable comparison with results in [50], four specific cases are defined in Table 4.1. Since we are assuming that $q = 0$, most of the results in this section are obtained using the analytical solution presented in Section 4.1.4 in lieu of Monte Carlo simulations.

We now examine in more detail the convergence obtained in the first step of the pricing process when solving the PDE in equation (4.45). Table 4.2 presents the expected utility value, and the corresponding cash equivalent option value, for different grid refinement levels. The portfolio allocation of the employee corresponds to *Case 1* as defined in Table 4.1. The other parameters used in the computation are presented in Table 4.3. Note that an unequally spaced grid containing 71 nodes is built in the S direction. Based on numerical tests conducted, it was determined that choosing $S_{\max} = \$20000$ results in 7 digits of accuracy for the utility values obtained. Similarly, an unequally spaced

Expected Utility - Classic Employee Option							
Ref.	Nbr. Timesteps	Nbr. Nodes S	Nbr. Nodes w	Expected Utility	Difference	Ratio	Cash Equivalent Option Value
Fully Implicit							
0	200	71	16	-1.046179e-07	n.a.	n.a.	6.9548
1	400	141	31	-1.045471e-07	7.0764e-11	n.a.	7.4955
2	800	281	61	-1.045102e-07	3.6879e-11	1.92	7.7776
3	1600	561	121	-1.044965e-07	1.3710e-11	2.69	7.8826
4	3200	1121	241	-1.044917e-07	4.8652e-12	2.82	7.9198
Crank-Nicolson							
0	200	71	16	-1.045734e-07	n.a.	n.a.	7.2943
1	400	141	31	-1.045248e-07	4.8609e-11	n.a.	7.6659
2	800	281	61	-1.044991e-07	2.5741e-11	1.89	7.8629
3	1600	561	121	-1.044910e-07	8.1388e-12	3.16	7.9253
4	3200	1121	241	-1.044889e-07	2.0778e-12	3.92	7.9412

TABLE 4.2: *Expected utility of the employee when given classic stock options with* $K = \$30$ *when fully implicit or Crank-Nicolson timestepping is used. The cash equivalent option value is presented in the last column of the table. The portfolio allocation of the employee corresponds to* Case 1 *as defined in Table 4.1. The initial timestep is set to* 0.05 *years on the coarsest grid, and halved at each refinement level. Other parameters used are presented in Table 4.3.*

w grid is built containing 16 nodes where $w_{\max} = \$25$ million. Again, $w_{\max}$ is chosen to ensure 7 digits of accuracy in the utility values obtained. Unless otherwise specified, this same grid construction is used to generate all numerical utility values contained in this section. As shown in Table 4.2, more than linear convergence is obtained when fully implicit timestepping is used. Similarly, close to quadratic convergence is observed when Crank-Nicolson timestepping is chosen.

To enable comparison with the results presented in [50], we consider alternate employee portfolio allocations. Table 4.4 presents the cash equivalent value of a classic employee stock option for each of the four cases described in Table 4.1, as well as the corresponding values presented in [50]. In [50], the authors adopt a certainty equivalence approach and use a binomial tree to determine the employee's expected utility at each node while utilizing the CAPM theory. The authors in [50] use identical parameter values

Parameter	Value
α - Employee's risk aversion coefficient	2.0
β - Firm's systematic risk measure	1.0
σ - Asset volatility	0.30
r - Risk-free interest rate	0.06
r_m - Expected market return	0.125
q - Dividend yield	0
K - Strike price	\$30
T - Contract maturity	10 years
Newton iteration tolerance	1×10^{-6}

TABLE 4.3: *Parameters used when pricing classic employee stock options in a utility context.*

to those in Table 4.3 and take 50 timesteps per year. Although our results may appear to be significantly different from those obtained in [50], we note that the authors do not provide any error estimates. The numerical difference between our results and those in [50] would appear to be due to the use of a more refined pricing grid on our part. Indeed, the results presented in Table 4.4 support this conjecture. Table 4.4 presents cash equivalent option values obtained after successive grid refinement operations during the PDE solution process and enables us to gain some insight into the accuracy of the numerical solution of equation (4.8). Focusing on *Case 1*, we note that the value of \$7.41 reported in [50] is between the values of \$7.29 (level 0) and \$7.67 (level 1) presented in Table 4.4. Extrapolation of the results in Table 4.4 for *Case 1* indicate that the converged solution is about \$7.95. In addition, we note that the company cost for the classic stock option presented in [50], which is simply the value of a European call option, differs by over \$0.15 from the value obtained when using our pricing software and/or Matlab. Consequently, we are of the opinion that the results in [50] are simply inaccurate due to the use of a coarse lattice.

Furthermore, the results in Table 4.4 confirm the now well-accepted theory that employees value stock options well below their no-arbitrage price [60, 70]. Keeping in mind that the no-arbitrage price of a classic employee stock option is about \$16.71, the data

Refinement	Case 1	Case 2	Case 3	Case 4
0	7.2943	9.2419	11.8030	16.4525
1	7.6659	5.9087	11.3631	10.5243
2	7.8629	5.0353	11.4057	9.1340
3	7.9253	4.8205	11.4351	8.7902
4	7.9412	4.7649	11.4422	8.6996
Values from [50]	7.41	3.49	10.51	6.41

TABLE 4.4: *Cash equivalent value of a classic employee stock option as a function of the employee's risk aversion coefficient and investment profile (see Table 4.1). Crank-Nicolson timestepping (with 2 Rannacher steps) was used in the PDE portion of the solution process. Other parameters used are presented in Table 4.3.*

in Table 4.4 demonstrates that an employee under-values his or her stock options by as much as 72% (*Case 2*). Overall, the option values presented in Table 4.4 are consistently higher (by \$1 on average in each case) than those presented in [50]. Thus, while the results in Table 4.4 are in accordance with those presented in [50], we obtain a more accurate estimate of the gap between the no-arbitrage price and the cash equivalent value of classic stock options.

4.3 Valuing Infinite Reload Options

In this section, we focus on valuing infinite reload options from a holder's perspective using the model described previously in Section 4.1. Limited work has been done regarding the utility pricing of reload options. Only [64] and [61] present a method for valuing reload options from an employee's perspective. In [64], the authors adopt a utility maximizing approach and use a trinomial tree coupled with backward induction to determine the value of reload options to employees. Note that the numerical results in [64] are limited to reload options with a finite number of reload opportunities. In [61], the author takes a different approach and extends a no-arbitrage analytical pricing formula for infinite reload options to determine the subjective option value. This is done by assuming that

the interest-rate and dividend yield are a function of the employee's risk aversion level and personal wealth held in company stock.

The work on reload options presented in this section differs from what is done in [64] in the following manner:

- Focusing exclusively on infinite reload options, we modify our general utility maximizing scheme presented in Section 4.1 to determine the cash equivalent value of infinite reload options. For this contract, the first step of the pricing process involves solving an impulse control problem with a penalty scheme. The second step uses Monte Carlo simulations (described in Section 4.1.3) or an analytical solution (described in Section 4.1.4) to determine the cash equivalent option value.

- In addition to demonstrating how significantly infinite reload options are undervalued by employees, we determine the effect of portfolio composition on the cash equivalent value of infinite reload options.

In Sections 4.3.1 and 4.3.2, we specify details of the pricing model when applied to infinite reload options and then present results from numerical experiments in Section 4.3.3.

4.3.1 Mathematical Model

While the core of the valuation model described in Section 4.1 remains intact when applied to infinite reload options, the additional reload feature must be taken into consideration when determining the employee's expected utility. Recall that the reload event provides the employee with new reload options with strike price $K = S$. As such, the expected utility is now a function of K and denoted as: $U = U(S, K, w, \tau)$. The reload constraint

is denoted by $\mathcal{A}U = \mathcal{A}U(S, K, w, \tau)$ and can be written as [64]:

$$\mathcal{A}U(S, K, w, \tau) = \begin{cases} U(S, S, w, \tau) & \text{when } S > K, \\ 0 & \text{otherwise.} \end{cases} \tag{4.51}$$

The operator $\mathcal{A}$ defined in equation (4.51) can be considered as a non-local impulse operator (see Chapter 1). In this case, the impulse operator $\mathcal{A}$ contains an intervention term $U(S, S, w, \tau)$ but no cash-flow term (see equation (1.1) in Chapter 1). While the number of shares held by the employee changes following a reload event, this information is encapsulated in the utility value. Indeed, the strike price K essentially keeps track of the number of shares held by the employee since K is the only variable required when computing this quantity (see discussion below). Furthermore, since we assume that the employee will not sell stock units received following a reload event, the reload constraint $\mathcal{A}U$ as defined in equation (4.51) will not include a cash-flow term.

When applying the utility pricing model to infinite reload options, the following impulse control problem is solved:

$$\min\left(U_\tau - (wr + qn_s(K)S)U_w - \mathcal{L}U, U - \mathcal{A}U\right) = 0, \tag{4.52}$$

on a truncated three-dimensional $S \times K \times w$ domain where the differential operator $\mathcal{L}$ is defined in equation (4.9).

As noted in equation (4.52), the number of stock units $n_s(K)$, as well as the number of reload options $n_o(K)$, is no longer constant. Both $n_o(K)$ and $n_s(K)$ depend on the current strike price K due to previous reload events. Let us assume that the employee initially owns $\overline{n_o}$ reload options with initial strike price K_{init} and $\overline{n_s}$ units of company stock. For those nodes with $K > K_{\text{init}}$, we know that at least one reload event has

occurred. Assuming the employee only reloaded once, the owner paid the strike price K_{init} of each option using $\frac{K_{\text{init}}}{K}$ stock units. The employee receives one unit of stock for each option exercised and $\frac{K_{\text{init}}}{K}$ new stock options with strike price K. Therefore, the current number of stock options $n_o(K)$ can be written as [64]:

$$n_o(K) = \overline{n_o}\frac{K_{\text{init}}}{K}, \tag{4.53}$$

while the current number of stock units is:

$$n_s(K) = \overline{n_s} + \overline{n_o}\left(1 - \frac{K_{\text{init}}}{K}\right). \tag{4.54}$$

While we have assumed that the employee only reloaded his or her option once, the authors of [64] show that equations (4.53) and (4.54) hold even if the employee reloads a larger number of times to obtain reload options with strike price K. Note that equations (4.53) and (4.54) are only applicable to nodes where $K > K_{\text{init}}$ since it is only optimal for the employee to reload his or her option if the strike price of the options received following a reload event is greater than the current strike price [39].

Additional boundary and initial conditions are necessary to fully specify the expected utility problem. At maturity, the initial conditions are:

$$U(S, w, \tau = 0) = U\big(\max(n_s(K)S + n_o(K)\ \text{Payoff}(S, K) + w, \delta)\big),$$

where $n_o(K)$ and $n_s(K)$ are defined in equations (4.53) and (4.54) respectively, $\text{Payoff}(S, K)$ corresponds to that of a standard call option in equation (4.13) and $0 < \delta \ll 1$ is included to avoid numerical issues when $S = w = 0$ (see Section 4.2.1).

Since the employee's expected utility depends on three path-dependent variables and time, a truncated three-dimensional $S \times K \times w$ domain is used: $[0, S_{\max}] \times [K_{\text{init}}, K_{\max}] \times$

$[0, w_{\max}]$ where K_{init} is the initial contract strike price. The domain in the K direction is limited to $[K_{\text{init}}, K_{\max}]$ since only information from the problems where $K \geq K_{\text{init}}$ are required to determine the expected utility for options of strike K_{init}.

To localize the utility maximization problem, we need to determine boundary conditions in the K direction in addition to those outlined in Section 4.2.1 for classic stock options. As $K \to K_{\text{init}}$, no additional boundary condition is necessary since this is an outflow boundary. However, the boundary at $K \to \infty$ needs to be handled with some care. While solving equation (4.52) adequately for large values of K remains feasible, applying the reload constraint in equation (4.51) becomes problematic for some outlying points. Since we truncate the K domain to $[K_{\text{init}}, K_{\max}]$, we lack the necessary information to apply the reload constraint correctly for points where $S > K_{\max}$. A similar difficulty was encountered when pricing infinite reload options in the no-arbitrage context in Chapter 2; this problem was alleviated by using a similarity reduction property to determine the reload constraint for outlying points [13]. Such a property is not applicable here due to the dependence of $n_o(K)$ and $n_s(K)$ on K.

Therefore, an alternate approach is used to determine the reload constraint for points where $S > K_{\max}$, which essentially involves placing a cap on the stock price at which the employee can reload. Instead of applying the reload constraint in equation (4.51), a modified version of the reload constraint will be used to determine the employee's expected utility:

$$\mathcal{A}U(S, K, w, \tau) = U(S = \min(S, K_{\max}), K = \min(S, K_{\max}), w, \tau). \tag{4.55}$$

Note that as $K_{\max} \to \infty$, the expected utility obtained using the modified reload constraint in equation (4.55) will tend to the value obtained if the original reload constraint, presented in equation (4.51), was applied to all grid nodes.

In summary, the first step of the option pricing process involves solving the impulse control problem:

$$\min\left(U_\tau - (wr + qn_s(K)S)U_w - \mathcal{L}U, U - \mathcal{A}U\right) = 0, \tag{4.56}$$

where $\mathcal{L}U$ is defined in equation (4.9) and the reload constraint $\mathcal{A}U$ is defined as:

$$\mathcal{A}U(S, K, w, \tau) = U(S = \min(S, K_{\max}), K = \min(S, K_{\max}), w, \tau). \tag{4.57}$$

Equation (4.56) is solved using a penalty method [44] on the discrete domain $[0, S_{\max}] \times [K_{\text{init}}, K_{\max}] \times [0, w_{\max}]$ with initial conditions:

$$U(S, w, \tau = 0) = U(\max(n_s(K)S + n_o(K)\max(S - K, 0) + w, \delta)), \tag{4.58}$$

where $0 < \delta \ll 1$ and $n_s(K)$ and $n_o(K)$ are defined in equations (4.53) and (4.54) respectively. The following boundary conditions are used:

$$U_\tau - wrU_w - \frac{1}{\epsilon}\max(\mathcal{A}U - U, 0) = 0 \quad \text{for } S = 0, \tag{4.59}$$

$$U_\tau - \mu U - (wr + qn_s(K)S)U_w - \frac{1}{\epsilon}\max(\mathcal{A}U - U, 0) = 0 \quad \text{for } S = S_{\max}, \tag{4.60}$$

$$U_\tau - \mathcal{L}U - (wr + qn_s(K)S)U_w - \frac{1}{\epsilon}\max(\mathcal{A}U - U, 0) = 0 \quad \text{for } w = 0, \tag{4.61}$$

$$U = 0 \quad \text{for } w = w_{\max}, \tag{4.62}$$

$$U_\tau - \mathcal{L}U - (wr + qn_s(K)S)U_w - \frac{1}{\epsilon}\max(\mathcal{A}U - U, 0) = 0 \quad \text{for } K = K_{\text{init}}, \tag{4.63}$$

$$U_\tau - \mathcal{L}U - (wr + qn_s(K)S)U_w = 0 \quad \text{for } K = K_{\max}. \tag{4.64}$$

Having determined the employee's expected utility, we can approximate the value of the infinite reload option to the holder by using Monte Carlo simulations as outlined

in Section 4.1.3. No modification needs to be made to the second step of the solution process outlined in Section 4.1.3 since this step does not depend on the type of stock option considered. Keep in mind that the portfolio content remains constant which means $n_s = \overline{n_s}$ (in Section 4.1.3) for infinite reload options. Similarly, the analytical solution presented in Section 4.1.4, which can replace Monte Carlo simulations, remains applicable for cases where the underlying stock does not generate any dividends (i.e. $q = 0$).

4.3.2 Underlying Grid and Discrete Equations

This section specifies some of the solution details when determining the employee's expected utility for infinite reload options in the first step of the pricing process. We first describe how the underlying $S \times K \times w$ grid is constructed and then specify the discrete equations solved on this domain.

As previously stated, the employee's expected utility when given infinite reload options will be obtained by solving equation (4.52) on a $[0, S_{\max}] \times [K_{\text{init}}, K_{\max}] \times [0, w_{\max}]$ domain. To construct the underlying grid, we build a set of discrete grid nodes $\{w_k\}$ for $k = 0, \ldots, k_{\max}$, where $w_0 = 0$, $w_{k_{\max}} = w_{\max}$ and one of the nodes in $\{w_k\}$ corresponds to the initial wealth of the employee.

Associated with each w_k is a two-dimensional $[0, S_{\max}] \times [K_{\text{init}}, K_{\max}]$ grid. Since equation (4.52) contains no K derivatives, the $S \times K$ domain is built using a set of one-dimensional S-grids. A *scaled grid* is built on the $[0, S_{\max}] \times [0, K_{\max}]$ domain as described in Section 2.2.1 and subsequently truncated by eliminating the one-dimensional problems with $K < K_{\text{init}}$ and renumbering the K_j nodes accordingly.

As outlined in Section 4.1.2, the discrete equations solved during the first step of the

pricing process can be written as:

$$\frac{U^{n+1}_{i,j,k} - U^{n}_{i,j,k'}}{\Delta\tau} = (1-\theta)[\mathcal{L}^h U]^{n+1}_{i,j,k} + \theta[\mathcal{L}^h U]^{n}_{i,j,k'} + P(U^{n+1}_{i,j,k}, \mathcal{A}^h U^{n+1}_{i,j,k}), \tag{4.65}$$

where $0 \leq \theta \leq 1$ determines the timestepping method used, $[\mathcal{L}^h U]^{n+1}_{i,j,k}$ is defined in equation (4.28) and $P(U^{n+1}_{i,j,k}, \mathcal{A}^h U^{n+1}_{i,j,k})$ represents the discrete penalty term in equation (4.26). Note that $U^{n}_{i,j,k'}$ may be obtained by linear interpolation as outlined in equation (4.23).

When calculating the reload constraint $\mathcal{A}^h U^{n+1}_{i,j,k}$ defined in equation (4.57), diagonal interpolation along the $S = K$ line is used to determine $U(S = \min(S, K_{\max}), K = \min(S, K_{\max}), w, \tau)$ as described in equation (2.30) (with $p = 0$). Defining the interpolation weight $0 \leq \omega \leq 1$ as in equation (2.31) (with $p = 0$), we obtain:

$$U(S^j_i, S^j_i, w_k, \tau^{n+1}) \simeq (1-\omega) U^{n+1}_{f,m,k} + \omega U^{n+1}_{f,m+1,k}\ , \tag{4.66}$$

where f is an index such that $S^j_f = K_j$ and $U^{n+1}_{f,m,k} = U(S^m_f, K_m, w_k, \tau^{n+1})$. Recall that $U^{n+1}_{f,j,k}$ corresponds to a node in the grid for all pricing problems (see Section 2.2.1).

4.3.3 Numerical Results

This section presents numerical results obtained when valuing infinite reload options in a utility framework. We start by examining the convergence of the PDE solution step of the pricing process. Table 4.6 presents the expected utility value and the corresponding cash equivalent option value for an infinite reload option with $K_{\text{init}} = \$100$ and a maturity of $T = 10$ years. The employee is assumed to have a total wealth of \$17 million divided between company stock, infinite reload options and a risk-free bank account. We assume that $w = \$5$ million, while \$8.4 million is invested in company stock and \$3.6 million in infinite reload options. Other parameter values are outlined in Table 4.5 and were

Parameter	Value
α - Employee's risk aversion coefficient	2.0
β - Firm's systematic risk measure	0.70
σ - Asset volatility	0.30
r - Risk-free interest rate	0.04
r_m - Expected market return	0.10
q - Dividend yield	0
K_{init} - Initial strike price	\$100
T - Contract maturity	10 years

TABLE 4.5: *Parameters used when pricing infinite reload options in a utility framework.*

chosen to be consistent with the *benchmark* firm presented in [65]. The underlying grid is built as described in Section 4.3.2 with 90 nodes in the S direction, 62 nodes in the K direction and 16 nodes in the w direction. As when pricing classic employee stock options, numerical tests were conducted to determine the appropriate values for $K_{\max}$ and $w_{\max}$. To ensure that the utility values obtained had 6 digits of accuracy, we chose $K_{\max} = \$20000$ and set $w_{\max} = \$25$ million. Our choice of $K_{\max}$ implies that $S_{\max} = \$4$ million according to scaled grid construction (see Section 2.2.1 for more details).

The expected utility values in Table 4.6 enable us to compute the convergence rate for the PDE solution step of the pricing process. Results for only three refinement levels were obtained due to the high dimensionality of the problem. Since a three-dimensional domain is required to price infinite reload options, a refinement level of 3 or higher required more memory than was available. Nonetheless, the results in Table 4.6 give some indication of the convergence obtained when valuing infinite reload options. Close to quadratic convergence is observed when using fully implicit and Crank-Nicolson timestepping during the PDE solution step, as indicated by the convergence ratios in Table 4.6. Note that the cash equivalent value, included in the last column of Table 4.6, converges at a similar rate.

The results in Table 4.6 confirm the well-established notion that employees under-

Infinite Reload Options								
	Nbr.	Nbr. Nodes			Expected			Cash Equivalent
Ref.	Timesteps	S	K	w	Utility	Difference	Ratio	Option Value
Fully Implicit								
0	500	90	62	16	-4.667776e-08	n.a.	n.a.	26.9406
1	1000	179	123	31	-4.709204e-08	-4.1428e-10	n.a.	25.0409
2	2000	357	245	61	-4.720267e-08	-1.1063e-10	3.74	24.5399
Crank-Nicolson								
0	500	90	62	16	-4.667253e-08	n.a.	n.a.	26.9649
1	1000	179	123	31	-4.708504e-08	-4.1252e-10	n.a.	25.0727
2	2000	357	245	61	-4.720186e-08	-1.1682e-10	3.53	24.5435

TABLE 4.6: *Expected utility of an employee when given infinite reload option at* $S = \$100$ *when fully implicit and Crank-Nicolson timestepping is used. The cash equivalent option value is included in the last column. The employee is assumed to have* \$8.4 *million in company stock,* \$3.6 *million in options and* \$5 *million of non-firm related wealth. Other pricing parameters used and contract details are presented in Table 4.5. Constant timesteps were taken in both cases and the initial timestep is* $\Delta\tau = 0.02$ *years on the coarsest grid.*

value stock options. Knowing that the corresponding no-arbitrage price for an infinite reload option is \$64.67 [13], we see that the value to the employee is only about 40% of the no-arbitrage price. Such a significant difference suggests that the company could negotiate with employees to replace the infinite reload options by simpler stock options with a lower no-arbitrage value or even a cash settlement.

However, the cash equivalent value of stock options is closely linked to the assumptions made regarding the issuing firm and the portfolio composition of the employee. As such, we consider different factors which affect the cash equivalent option value such as the non-firm related wealth of the employee w, the dividend yield of the underlying stock q and the balance between options and company stock in the employee's portfolio. We begin by considering different values of w to determine the effect on the employee's option value. Table 4.7 presents the cash equivalent value of an infinite reload option when the employee is assumed to own \$3.6 million of options, \$8.4 million of company stock and has non-firm related wealth ranging from \$3 to \$10 million. The results presented in

	Initial Wealth Level (w)			
Refinement	\$3 mil.	\$5 mil.	\$7 mil.	\$10 mil.
0	22.1378	26.9649	30.3471	31.1250
1	21.0750	25.0727	27.9402	29.9378
2	21.0616	24.5435	26.9295	29.3030

TABLE 4.7: *Cash equivalent value of an infinite reload option at* $S = \$100$ *as a function of the employee's external wealth level (w). Recall that the no-arbitrage price of this infinite reload option is* \$64.67. *Crank-Nicolson timestepping (with 2 Rannacher steps) was used in the PDE portion of the solution process. The employee is assumed to have* \$8.4 *million in company stock and* \$3.6 *million in options. Other parameters used are presented in Table 4.5.*

Table 4.7 indicate that the option value increases significantly as w is increased. For example, the value to the employee of an infinite reload option increases by 20% when w goes from \$3 to \$5 million.

While our analysis has thus far focused on cases where the dividend yield is zero, we now consider the effect of non-zero dividends on the executive's valuation of infinite reload options. Table 4.8 presents the cash equivalent value of an infinite reload option with $q = 1\%$ or $q = 2\%$. The employee is assumed to have \$3.6 million in reload options, \$8.4 million of company stock and \$3 million of non-firm related wealth. Crank-Nicolson timestepping is used when solving the PDE in the first part of the pricing process and Monte Carlo simulations are used to determine the final cash equivalent option value when $q > 0$. Note that 5 million simulations were conducted in each case with a timestep of $\Delta t = 5 \times 10^{-4}$ years. Though not included here, results from previous refinement levels enable us to estimate that the values in Table 4.8 for $q > 0$ are accurate to within $\pm\$0.50$. The cash equivalent value of an infinite reload options when $q = 0$, obtained with the analytical formula presented in Section 4.1.4, is also included in Table 4.8 for comparison purposes. The numerical results presented in Table 4.8 illustrate that the effect of non-zero dividend yield on the cash equivalent value of an infinite reload option is limited. Indeed, increasing the dividend yield from $q = 0$ to $q = 2\%$ only reduces the

Dividend Yield			
S	$q = 0\%$	$q = 1\%$	$q = 2\%$
90	17.3191	16.8533	16.3373
100	21.0616	20.8525	20.5895
110	24.8836	24.9708	25.1086

TABLE 4.8: *Cash equivalent value of an infinite reload option as a function of the dividend yield (q). Note that* 5 *million Monte Carlo simulations were conducted to obtain the option cash value when* $q > 0$. *The employee is assumed to have* \$8.4 *million in company stock,* \$3.6 *million in options and* \$3 *million of non-firm related wealth. Crank-Nicolson timestepping (with 2 Rannacher steps) was used in the PDE portion of the solution process. Results when* $q > 0$ *are accurate to within* $\pm$\$0.50. *Other parameters used are presented in Table 4.5.*

Error from Monte Carlo Simulations			
		Cash Equivalent Option Value	
S	Expected Utility	Monte Carlo	Analytical
90	-6.313089e-08	17.3261	17.3191
100	-5.826764e-08	21.0346	21.0616
110	-5.408418e-08	24.9134	24.8836

TABLE 4.9: *Error when using Monte Carlo simulations to determine the cash equivalent value of an infinite reload option when* $q = 0$. *For comparison, the analytical option value is presented in the last column. When using Monte Carlo simulation, we conducted* 5 *million simulations with* $\Delta t = 5 \times 10^{-4}$ *years. The employee is assumed to have* \$8.4 *million in company stock,* \$3.6 *million in options and* \$3 *million of non-firm related wealth. Other parameters used are presented in Table 4.5.*

cash equivalent option value by about \$0.40 when $S = \$100$.

To estimate the error resulting from the use of Monte Carlo simulations, Table 4.9 presents the cash equivalent value for an infinite reload option when using the analytical solution and Monte Carlo simulations with $q = 0$ in the second step of the pricing process. The results in Table 4.9 indicate that carrying out 5 million Monte Carlo simulations with the parameters stated above generates errors of about $\pm$\$0.05 which is much smaller that the corresponding PDE solution error in some cases (see Table 4.7).

Thus far, we have assumed that 30% of the total firm-related wealth (\$12 million) is invested in options and 70% is invested in company stock. Assuming that $w = \$5$ million, we vary the proportion of the firm-related wealth invested in options and see how this

	Option and Stock percentage		
	30% options	40% options	60% options
Refinement	70% stock	60% stock	40% stock
0	26.9649	28.1301	28.7251
1	25.0727	26.9679	29.1311
2	24.5435	26.8579	29.3951

TABLE 4.10: *Cash equivalent value of an infinite reload option as a function of the distribution of the firm-related wealth (estimated at* \$12 *million) between company stock and options. The employee is assumed to have* \$5 *million of non-firm related wealth. Recall that the no-arbitrage price of this infinite reload option is* \$64.67. *Note that Crank-Nicolson timestepping (with 2 Rannacher steps) was used in the PDE portion of the solution process. Other parameters used are presented in Table 4.5.*

affects the cash equivalent value of the infinite reload option to the employee. Table 4.10 presents the value of an infinite reload option obtained when the proportion of options ranges from 30% to 60% of the firm-related wealth, which we assume to be \$12 million. Based on the results in Table 4.10, we see that the value of an infinite reload option to the employee increases as a larger portion of the firm-related wealth is held in reload options. For example, increasing the options from 30% to 60% of the firm-related wealth results in an increase of about 20% of the option value.

4.4 Summary

In view of the wide range of existing employee stock options, we have developed a general utility-based pricing framework that can be used to value stock options from a holder's perspective. Our pricing model is a multi-step process that involves computing two expected utility values. First, we determine the employee's expected utility when given stock options as part of his or her compensation package by solving the appropriate partial differential equation (or impulse control problem as the case may be). Second, we determine the initial cash compensation that, when given instead of stock options, provides the same expected utility level by using Monte Carlo simulations, or an analytical

solution if applicable. This cash amount is an estimate of the value to the executive of the stock options held in his or her portfolio.

To highlight our method's versatility, we applied the pricing model to value simple classic stock options and more complex infinite reload options. The numerical results obtained when pricing classic employee stock options were consistent with previous literature and clearly show that employees place a much lower value on stock options when compared to their no-arbitrage price. However, there is some question about the accuracy of previously reported results for classic stock options. A similar gap was observed when valuing infinite reload options.

Although it is now recognized that, from a no-arbitrage perspective, the reload feature is very expensive, from an employee point of view, these features are not valued as highly. The results included in this chapter suggest that there is an opportunity to negotiate with employees, perhaps by paying a cash settlement, to eliminate some of the expensive (from a no-arbitrage point of view) contractual features. Our research suggests that employees would be willing to accept cash settlements at significantly less than the no-arbitrage price of the option contracts. This would result in a net improvement in the company balance sheet.

While the pricing model presented in this chapter is applicable to many different employee stock options, a possible weakness of this utility approach lies in the fact that the value depends on the individual situation of each employee. Since the pricing model uses individual parameters such as the employee's portfolio composition and risk-aversion coefficient, it may be difficult to obtain a reliable cash equivalent option value. Often, these values may need to be estimated as there may only be partial information available. Nonetheless, once reasonable parameter values are chosen, this model provides a general framework to obtain the utility-maximizing value of employee stock options.

Chapter 5

Valuing the Guaranteed Minimum Death Benefit Clause with Partial Withdrawals

A variable annuity or equity-linked insurance contract is a retirement and/or investment vehicle created by insurance companies. It is a contract between the customer and the insurance company where the insurer generally agrees to make periodic payments to the client starting at a given date. These contracts are particularly popular in the United States and the United Kingdom since the investment gains are tax-deferred until the funds are withdrawn or annuitized at retirement. In addition to the tax deferral these contracts may also include a death benefit. Specific examples of variable annuity contracts include guaranteed minimum income benefits, guaranteed minimum withdrawal benefits [73, 34, 25] and guaranteed minimum death benefits.

We are interested in the latter contract, the guaranteed minimum death benefit (GMDB). In this case, if the customer passes away before the maturity of the contract,

then the beneficiary receives the greater of the investment account value or the death benefit. We consider the case of *market guarantees*, where some form of market returns are guaranteed through periodic ratchet dates [74]. A GMDB contract has two phases: the accumulation phase and the continuation phase. During the accumulation phase, the value of the death benefit is reset periodically to the maximum of the current account value or the prior death benefit value[1]. This initial phase begins when the contract is purchased and typically ends when the client reaches 80 years of age [79]. Once the accumulation phase is over, the continuation phase begins with the value of the death benefit now remaining constant. The contract usually expires when the client turns 90 years old [79]. At this time, the client receives the market value of the invested capital. Should the client pass away prior to the contract maturity, the beneficiary receives the maximum of either the guaranteed death benefit or the current investment account value. We also allow partial withdrawals from the account, a common feature in GMDB contracts.

Determining the fair insurance fee for a GMDB contract allowing partial withdrawals is a challenging and important problem. The first difficulty arises from the stochastic nature of the contract maturity caused by the death benefit. These market guarantees expose insurance companies to considerable risk during prolonged periods of weak equity markets. Indeed, due to the stochastic nature of the contract maturity, GMDB contracts have also been referred to as *Titanic options* [72]. Allowing for partial withdrawal of funds introduces a second level of uncertainty to these contracts. Here the problem is that some withdrawals are better than others in terms of overall return. As discussed in [30], a conservative approach to pricing these guarantees is based on the *optimal* withdrawal at any given instant (i.e. the worst case from the hedger's point of view). Thus, determining insurance fees for GMDB contracts with partial withdrawal becomes an optimal control

[1]Intuitively, this can be viewed as a discretely observed lookback option based on the maximum value of the underlying [85].

problem.

While many variable annuities have been well studied in the literature, the same cannot be said for GMDB contracts. This is even more pronounced when partial withdrawal is considered. Bauer, Kling and Russ [12] give a solution to the GMDB problem allowing optimal withdrawal at discrete instances under a constant volatility Brownian motion pricing model. In between the withdrawal times, the solution of a modified Black-Scholes PDE is determined by a Green's function integral, which is approximated numerically. The optimal withdrawal at each withdrawal time is determined by a grid search. Previous to this, Mudavanhu and Zhuo [74] determine valuations of GMDB contracts where the owner has the option to lapse after paying lapsation or surrender charges[2]. Thus, withdrawals are allowed but they are of the all or nothing nature. Lapsation can then be considered to be an American type option, which allows the authors in [74] to use a numerical method based on Monte Carlo simulations for American options to price the guarantee. Note that it is not clear how to extend this approach to the case of partial withdrawals. Milevsky and Posner [72] determine the fair insurance charge such that the risk-neutral present value of the contract benefits and the contract costs are equivalent. However, partial withdrawals are not considered. Work in this area includes [45] (where both actuarial and financial approaches are studied) and [28] (which focuses on the hedging of GMDB contracts) where again in these papers partial withdrawals are not allowed. In addition, all the above papers rely on a constant volatility Brownian motion pricing model.

In this chapter, we determine the fair insurance charge for a GMDB contract from a combined no-arbitrage and actuarial approach (see [88]). We characterize the GMDB pricing problem as an impulse control problem and develop a pricing model based on partial

[2]When a policy owner decides to lapse his policy, the investments are liquidated and the variable annuity is terminated.

differential inequalities. While valuing the GMDB guarantee from an issuer's perspective, this approach takes into account the partial withdrawal feature. Not surprisingly, our results show that the withdrawal feature is very valuable and results in significantly higher insurance fees than found previously in the literature when withdrawals are ignored.

The assumption of a constant volatility pricing model used in [12, 74, 72, 45, 28] is well-known to be inconsistent with the implied volatility observed in the market. In this chapter, we value the insurance fee for GMDB contracts using both a constant volatility pricing model (for comparative purposes) and a pricing model which switches randomly between a finite number of states or regimes. Such regime-switching models can be viewed as representing different economic states (e.g. economic growth or recession), where each state has a different risk factor (i.e. volatility) and the underlying switches randomly between these discrete states. Regime-switching is known to provide more realistic models for market conditions, especially for long-term contracts.

Our valuations for the fair insurance fee of GMDB contracts are determined using a partial differential equation (PDE) framework. We use a fully implicit penalty method to solve this impulse control problem, where we allow both complete lapsation and partial withdrawal. We are able to give a complete solution to the GMDB problem. By this we mean that we: (a) give a complete specification of the problem in terms of PDEs, including localized boundary conditions; (b) discretize the system of PDEs using a fully implicit method; (c) compare the numerical results to those of previous work; and (d) prove that the discrete equations converge to the viscosity solution [31] (assuming it exists) away from ratchet dates. The last named property follows from proving that our discrete equations are monotone, stable and consistent [8]. It is well known that the viscosity solution is the financially relevant solution of option pricing problems.

5.1 GMDB Model with Constant parameters

In this section, we derive the GMDB pricing model in the classic Black-Scholes framework [85]. Since our goal is to compute the no-arbitrage insurance charge for such contracts, the pricing model developed here determines the cost of the GMDB guarantee from the issuer's perspective, as opposed to the value of the whole contract (i.e. investment account plus guarantee). Let $\hat{V} = \hat{V}(S, B, D, t)$ represent the cost of the GMDB guarantee where:

- S is the current value of the underlying investment account,
- B is the current death benefit level,
- D is the current amount deposited in the investment account, and
- t denotes time.

When the GMDB contract is issued ($t = 0$), the death benefit is set to the initial deposit $\mathcal{D}_0$ made by the policy owner, i.e. $B = \mathcal{D}_0$ at $t = 0$. The death benefit can then be reset at each ratchet date to the maximum of the current investment account value or the current benefit level. Generally, ratchet events only occur during the accumulation phase of the contract and the last ratchet date is typically scheduled at the end of the policy year when the owner turns 80 years old [79]. We denote a ratchet date by t_o and define t_o^- and t_o^+ as the times just before and after t_o. By standard no-arbitrage arguments, the following jump condition is obtained:

$$\hat{V}(S, B^+, D, t_o^+) = \hat{V}(S, B^-, D, t_o^-), \tag{5.1}$$

where $B^+ = \max(B^-, S)$.

Should the policy owner pass away prior to the expiry of the GMDB contract, the death benefit is exercised and the beneficiary receives the greater of the current benefit level or the current investment account value. Consequently, the issuing company is liable for any excess payment when the current death benefit is higher than the investment account value.

When the holder of the contract makes a partial or full withdrawal (lapsing), a surrender charge, denoted as $\hat{\gamma}(t)$, is imposed. When the death benefit is exercised, the owner's estate does not pay a surrender charge. However, the issuer may have to pay a surrender charge to the re-seller [79]. In this chapter, we consider the value of the guarantee from the issuer's perspective. To be concrete, we can think of the issuer of the guarantee as a re-insurer, and the re-seller as an insurance company selling the guarantee to retail customers. We assume that the surrender charge is calculated as a percentage of the current deposit level D [79]. Generally, the surrender charge is highest at the start of the contract and decreases annually. After the initial t_s years of the contract, the surrender charge disappears: $\hat{\gamma}(t) = 0$ when $t > t_s$ years. Typically, $t_s = 7$ years. Hence, the death benefit exposure of the issuer, denoted by $f = f(S, B, D, t)$, is defined as:

$$f(S, B, D, t) = \max(B - S, 0) + \hat{\gamma}(t)D. \tag{5.2}$$

Note that when modelling the death benefit exposure, we use a mortality distribution as discussed subsequently.

Annual fees associated with variable annuity contracts, referred to as mortality and expense (M&E) fees, are charged to the policy owner. The mortality and expense fees, denoted by ρ_{total}, are calculated as a predetermined percentage of the account value S,

and include both management fees (ρ_{man}) and insurance charges (ρ_{ins}):

$$\rho_{\text{total}} = \rho_{man} + \rho_{ins}. \tag{5.3}$$

Assuming the management fees (ρ_{man}) are known, we will determine in Section 5.5 the value of ρ_{ins} such that the issuer does not incur any loss, assuming the contract is hedged. As outlined in [74], these M&E charges can be modeled similarly to dividends.

In this section, we will assume that the value of the underlying investment account (S) follows a classic geometric Brownian motion process (under the risk-neutral measure)[54]:

$$\frac{dS}{S} = (r - \rho_{\text{total}})dt + \sigma dZ, \tag{5.4}$$

where r is the risk-free rate, σ is the asset volatility and dZ is the increment of a Wiener process [85].

When the GMDB contract expires at time $t = T$, the owner, if still alive, receives a payoff corresponding to the value of the invested capital at contract maturity. As such, the issuing company is not liable for any additional payment at maturity beyond the current investment account value. Consequently, at time $t = T$, we have:

$$\hat{V}(S, B, D, t = T) = 0. \tag{5.5}$$

Following the derivation in [86, 88], the cost of the GMDB guarantee in the Black-Scholes framework is given by:

$$\hat{V}_t + \frac{1}{2}\sigma^2 S^2 \hat{V}_{SS} + (r - \rho_{\text{total}})S\hat{V}_S - r\hat{V} - \hat{\mathcal{R}}(t)\rho_{ins}S + \hat{\mathcal{M}}(t)f = 0, \tag{5.6}$$

where $\hat{\mathcal{M}}(t)$ represents the mortality function of the policy owners and $\hat{\mathcal{R}}(t)$ is the survival probability of policy owners. The mortality function is defined such that the fraction of original owners who pass away during the time interval $[t, t+dt]$ is $\hat{\mathcal{M}}(t)dt$. Consequently, the portion of policy owners still alive at time t, denoted by $\hat{\mathcal{R}}(t)$, is:

$$\hat{\mathcal{R}}(t) = 1 - \int_0^t \hat{\mathcal{M}}(n)dn, \tag{5.7}$$

where the integral term represents the owners who have died during the period $[0, t]$. Note that equation (5.6) is derived under the assumption that mortality risk is diversifiable amongst many policy owners [72].

Many GMDB contracts include a feature allowing the policy owner to make partial withdrawals from the invested capital at any time prior to the maturity of the contract (during both the accumulation and continuation phase). When the owner makes a withdrawal, both the deposit D and the death benefit B are reduced [79]. In this work, we assume that D and B are reduced on a dollar-for-dollar basis following a partial withdrawal. However, our PDE approach can easily be extended to model different withdrawal policies. For example, an alternate withdrawal policy whereby the deposit is reduced by the amount withdrawn but the death benefit is reduced on a proportional basis, could be easily implemented.

The partial withdrawal feature enables the contract owner to withdraw any cash amount up to the current account value S. However, to keep the policy active, a minimal deposit amount must remain in the investment account. We denote the partial withdrawal amount as $W \in [0, S - \omega]$, where ω is the minimal deposit amount. For each partial withdrawal, a surrender charge, denoted by $\hat{\gamma}(t)$ and calculated as a percentage of W, is imposed. The surrender charge $\hat{\gamma}(t)$ is also applied when the owner chooses to lapse his policy. Recall that when an investor decides to lapse his policy, the investment account

is liquidated and the GMDB policy cancelled. In this case, the surrender charge is a percentage of the investment account value S.

While our goal is to determine the no-arbitrage insurance charge for the GMDB guarantee, for explanatory purposes, we first consider the effect of partial withdrawals on the entire GMDB contract (investment account plus guarantee) and determine the appropriate withdrawal constraint. The withdrawal constraint for the entire GMDB contract is then used as a tool to derive the withdrawal constraint for the GMDB guarantee.

Let $\hat{\mathcal{V}} = \hat{\mathcal{V}}(S, B, D, t)$ represent the value of the entire GMDB contract (investment account plus guarantee). Assuming optimal behavior and ignoring mortality effects for the moment, the policy owner will maximize his return and choose W such that:

$$W = \underset{W' \in [0, S-\omega]}{\operatorname{argmax}} \big((1 - \hat{\gamma}(t)) W' + \hat{\mathcal{V}}(S - W', \max(B - W', 0), \max(D - W', 0), t) \big). \quad (5.8)$$

Taking into consideration the option to lapse, the value of the total GMDB contract satisfies (after optimal withdrawal or lapsing):

$$\hat{\mathcal{V}} = \max \Bigg((1 - \hat{\gamma}(t)) S, \quad (5.9)$$
$$\max_{W \in [0, S-\omega]} \big((1 - \hat{\gamma}(t)) W + \hat{\mathcal{V}}(S - W, \max(B - W, 0), \max(D - W, 0), t) \big) \Bigg).$$

While we have assumed in equation (5.9) that the contract owner will lapse whenever it is optimal to do so, alternate assumptions could be made whereby the contract owner would lapse at a pre-determined rate. See [86, 88] for more details on modeling investor lapsing.

Since our goal is to determine the value of the GMDB guarantee, we now derive the equivalent withdrawal constraint from the issuer's perspective. Recall that we are looking to value the GMDB guarantee in an aggregate sense by assuming that contracts are sold

to a given population. As such, the mortality/survival function defined in equation (5.7) must be taken into consideration when determining the withdrawal constraint. More precisely, we redefine $\hat{\mathcal{V}}(S,B,D,t)$ as the value of the whole contract to the issuer which can be written as: $\hat{\mathcal{V}}(S,B,D,t) = \hat{V}(S,B,D,t) + \mathcal{R}(t)S$. Notice that only the investment account is affected by the survival probability since investor mortality is already included in the differential equation for $\hat{V}(S,B,D,t)$ presented as (5.6). Since only those owners that are alive can conduct a withdrawal or choose to lapse, the cash flows associated with both actions will also be scaled by the survival probability.

As such, integrating our cash flow assumption into equation (5.9), we obtain:

$$\begin{aligned}\hat{\mathcal{V}} &= \max\Bigg(\hat{\mathcal{R}}(t)(1-\hat{\gamma}(t))S, \\ &\qquad \max_{W\in[0,S-\omega]}\big(\hat{\mathcal{R}}(t)(1-\hat{\gamma}(t))W + \hat{\mathcal{V}}(S-W,\max(B-W,0),\max(D-W,0),t)\big)\Bigg) \\ &= \max\Bigg(\hat{\mathcal{R}}(t)(1-\hat{\gamma}(t))S, \\ &\qquad \max_{W\in[0,S-\omega]}\big(-\hat{\mathcal{R}}(t)\hat{\gamma}(t)W + \hat{V}(S-W,\max(B-W,0),\max(D-W,0),t) + \hat{\mathcal{R}}(t)S\big)\Bigg).\end{aligned}$$

Since $\hat{\mathcal{V}}(S,B,D,t) = \hat{V}(S,B,D,t) + \hat{\mathcal{R}}(t)S$, we have:

$$\hat{V} = \max\left(-\hat{\mathcal{R}}(t)\hat{\gamma}(t)S, \max_{W\in[0,S-\omega]}\big(\hat{V}(S-W,\max(B-W,0),\max(D-W,0),t) - \hat{\mathcal{R}}(t)\hat{\gamma}(t)W\big)\right).$$

Thus, we can denote the withdrawal constraint by $\mathcal{A}\hat{V} = \mathcal{A}\hat{V}(S,B,D,t)$ with:

$$\begin{aligned}\mathcal{A}\hat{V} \equiv \max\Bigg(&-\hat{\mathcal{R}}(t)\hat{\gamma}(t)S, \\ &\max_{W\in[0,S-\omega]}\big(\hat{V}(S-W,\max(B-W,0),\max(D-W,0),t) - \hat{\mathcal{R}}(t)\hat{\gamma}(t)W\big) - c\Bigg),\end{aligned} \tag{5.10}$$

where $c > 0$ is a small fixed cost added to the constraint to ensure that the impulse control problem is well-posed. While our theoretical formulation requires that $c > 0$, the numerical scheme presented subsequently accepts both $c = 0$ and $c > 0$. We expect in practice that very small values of c will have little effect on the numerical solution obtained. This is confirmed by the examples included in Section 5.5.

We can write the pricing problem for the GMDB guarantee (away from the ratchet dates) as an impulse control problem:

$$\min\left(\hat{V}_t + \mathcal{L}\hat{V} - \hat{\mathcal{R}}(t)\rho_{ins}S + \hat{\mathcal{M}}(t)f, \hat{V} - \mathcal{A}\hat{V}\right) = 0, \tag{5.11}$$

where the differential operator $\mathcal{L}$ is defined as:

$$\mathcal{L}\hat{V} = \frac{1}{2}\sigma^2 S^2 \hat{V}_{SS} + (r - \rho_{\text{total}})S\hat{V}_S - r\hat{V}. \tag{5.12}$$

Note that using equation (5.10) with $c > 0$ ensures that the second term in equation (5.11) is not trivially zero which for example occurs when $W = 0$ since $\mathcal{A} = I$ in this case.

The GMDB pricing problem can also be written as a penalized problem:

$$\lim_{\epsilon \to 0}\left(V_\tau - \mathcal{L}V + \mathcal{R}(\tau)\rho_{ins}S - \mathcal{M}(\tau)f - \frac{1}{\epsilon}\max(\mathcal{A}V - V, 0)\right) = 0, \tag{5.13}$$

where $\tau = T - t$ is defined as the time to maturity and

$$V(S,B,D,\tau) = \hat{V}(S,B,D,T-\tau)\ ,\ \mathcal{R}(\tau) = \hat{\mathcal{R}}(T-\tau)\ ,\ \mathcal{M}(\tau) = \hat{\mathcal{M}}(T-\tau)\ ,\ \gamma(\tau) = \hat{\gamma}(T-\tau).$$

The pricing problem in equation (5.13) is solved numerically using the penalty method outlined in [44, 34]. More specifically, using the terminal payoff in equation (5.5) as initial conditions, we work backward in time to solve equation (5.13) while taking into

consideration the jump condition in equation (5.1). Note that [3] demonstrates that the penalty method is a good viscosity approximation for impulse control problems obtained when pricing American options. In [4], the author also outlines properties for well-posed impulse control problems.

5.1.1 Boundary Conditions

The GMDB guarantee pricing problem in equation (5.11) is solved on an $S \times B \times D \times \tau$ domain. Since $B = \mathcal{D}_0$ at $\tau = T$ (or $t = 0$), equation (5.1) indicates that the benefit level B can only increase, unless a withdrawal occurs. Similarly, $D = \mathcal{D}_0$ at $\tau = T$ and the deposit D decreases only when a partial withdrawal occurs. Since D is reduced by the same amount as B following a withdrawal, we have that $B \geq D$. As such, the solution domain for $V(S, B, D, \tau)$ is defined as:

$$[0, \infty] \times [D, \infty] \times [0, \mathcal{D}_0] \times [0, T], \tag{5.14}$$

where $\mathcal{D}_0$ is the initial investment deposit and T is the contract maturity.

To localize the GMDB pricing problem, additional boundary conditions are necessary. As $S \to 0$, the partial withdrawal policy is no longer applicable and the penalized problem in equation (5.13) reduces to (noting the definition of $f = f(S, B, D, \tau)$ in equation (5.2)):

$$V_\tau + rV - \mathcal{M}(\tau)(B + \gamma(\tau)D) = 0. \tag{5.15}$$

As $S \to \infty$, we make the common assumption that $V_{SS} \to 0$ [89], which implies that V is a linear function of S, along with the additional assumption that the linear term dominates in size (see Appendix G). Using these assumptions, we obtain the following

approximation to equation (5.13):

$$V_\tau + \rho_{\text{total}} V + \mathcal{R}(\tau)\rho_{ins} S - \frac{1}{\epsilon}\max(\mathcal{A}V - V, 0) = 0 \; ; \; S = S_{\max}. \tag{5.16}$$

As $B \to D$, no additional boundary condition is required and the pricing equation in (5.13) is solved. As $B \to \infty$, equation (5.13) is solved but the jump condition in equation (5.1) needs to be modified to take into consideration the discrete solution domain. For those grid nodes where $S > B_{\max}$, the discrete $S \times B$ plane does not contain the required data to calculate the jump condition outlined in equation (5.1). We assume that no ratchet events occur for those nodes where $S > B_{\max}$, which implies (in terms of $\tau = T - t$):

$$V(S, B, D, \tau_o^+) = \begin{cases} V(S, B, D, \tau_o^-) & \text{if } S \le B, \\ V(S, S, D, \tau_o^-) & \text{if } B < S \le B_{\max}, \\ V(S, B, D, \tau_o^-) & \text{if } S > B_{\max}, \end{cases} \tag{5.17}$$

where τ_o denotes the ratchet date, while τ_o^- and τ_o^+ denote the instants immediately before and after a ratchet event. This is clearly an approximation but the resulting error will be small, assuming $B_{\max}$ is chosen sufficiently large. Numerical tests conducted in Section 5.5 verify this to be the case.

In the D direction, no additional boundary condition is required as $D \to \mathcal{D}_0$, since $\mathcal{A}V$ requires information only from problems where $D < \mathcal{D}_0$ (from equation (5.10)). As $D \to 0$, the partial withdrawal feature remains applicable and the usual pricing equation (5.13) is solved.

In summary, when valuing the GMDB guarantee, the following equation is solved

numerically on a three-dimensional $[0, S_{\max}] \times [D, B_{\max}] \times [0, \mathcal{D}_0]$ domain:

$$V_\tau - \mathcal{L}V + \mathcal{R}(\tau)\rho_{ins}S - \mathcal{M}(\tau)f - \frac{1}{\epsilon}\max(\mathcal{A}V - V, 0) = 0, \tag{5.18}$$

where the differential operator $\mathcal{L}$ is defined in equation (5.12), $f = f(S, B, D, \tau)$ is defined in equation (5.2) and the withdrawal constraint $\mathcal{A}V$ is defined in equation (5.10). The initial conditions are:

$$V(S, B, D, \tau = 0) = 0. \tag{5.19}$$

The boundary conditions used are:

$$V_\tau + rV - \mathcal{M}(\tau)(B + \gamma(\tau)D) = 0 \text{ for } S = 0, \tag{5.20}$$

$$V_\tau + \rho_{\text{total}}V + \mathcal{R}(\tau)\rho_{ins}S - \frac{1}{\epsilon}\max(\mathcal{A}V - V, 0) = 0 \text{ for } S = S_{\max}, \tag{5.21}$$

while the usual pricing equation in (5.18) is solved on the boundaries of the $B \times D$ plane. Finally, the benefit jump condition in equation (5.17) is applied on pre-specified ratchet dates.

5.2 Pricing the GMDB Guarantee with Regime-Switching

The assumption of constant volatility for option contracts is well-known to be inconsistent with the implied volatility observed in the market. In this section, we introduce the concept of regime-switching to the GMDB impulse control problem in equation (5.13). The underlying assumption is that the volatility switches randomly between a finite number of states or regimes. Each regime has a different volatility value and is meant to represent a different economic state. While the underlying account value follows a log-normal process within a given state, a jump in S occurs when the state of the economy changes.

One could argue that stochastic volatility [85] is a valid alternative to regime-switching when dealing with long-term contracts such as variable annuities. However, using a stochastic volatility model implies solving a higher dimensional PDE. While both models are valid remedies to unrealistic constant volatility models for long-term contracts, regime-switching appears to be less expensive from a computational point of view and may be somewhat more intuitive.

Introduced in [52], the concept of regime-switching has since been used extensively when modeling both interest rates [49, 91, 27] and pricing option contracts [15, 38, 92, 18, 16]. In [15], the author develops a pentanomial lattice for valuing both European and American type options with regime-switching. In [38], a family of option pricing models is developed which incorporates regime-switching as well as a feedback effect on volatility. Meanwhile, both [18] and [92] derive a system of option pricing PDEs that incorporates the concept of regime-switching but apply differing boundary conditions. In [16], the authors provide a regime-switching pricing model, based on PDEs, for more exotic options such as Asian and lookback options. Working in another vein, the author of [53] determines the appropriate parameters for a regime-switching lognormal model using maximum likelihood estimation.

To extend our modelling framework to regime-switching, we introduce an additional modeling variable E which represents the current state of the economy and define M distinct states: $E \in \{e_1, e_2, \ldots, e_M\}$. Associated with each state e_m is a constant volatility value denoted as σ_m. Assuming we are in state e_m, the value of the GMDB guarantee is denoted as:

$$\hat{V}^m = \hat{V}(S, B, D, E = e_m, \sigma_m, t). \tag{5.22}$$

For a given regime e_m, the value of the underlying investment account S follows (under

the risk neutral measure):

$$\frac{dS}{S} = \left(r - \rho_{\text{total}} - \sum_{\substack{l=1 \\ l\neq m}}^{M} \lambda^{m\to l}(J^{m\to l} - 1)\right)dt + \sigma_m dZ + \sum_{\substack{l=1 \\ l\neq m}}^{M}(J^{m\to l} - 1)dq^{m\to l}, \quad (5.23)$$

where $dq^{m\to l}$ is an independent Poisson process and $J^{m\to l} \geq 0$ $(l \neq m)$ is an impulse function producing a jump from S to $J^{m\to l}S$ when the state of the economy changes from e_m to e_l. We define $\lambda^{m\to l}$ $(l \neq m)$ as the risk-neutral probability of a jump from economic state e_m to state e_l and have (for $l \neq m$):

$$dq^{m\to l} = \begin{cases} 0 & \text{with probability } 1 - \lambda^{m\to l}dt, \\ 1 & \text{with probability } \lambda^{m\to l}dt. \end{cases} \quad (5.24)$$

A system of coupled PDEs can then be derived to determine the value of the GMDB guarantee in the regime-switching context. Each PDE represents a different economic state and can be written as (see [27]):

$$\hat{V}_t^m + \left(r - \rho_{\text{total}} - \sum_{\substack{l=1 \\ l\neq m}}^{M} \lambda^{m\to l}(J^{m\to l} - 1)\right)S\hat{V}_S^m + \frac{1}{2}\sigma_m^2 S^2 \hat{V}_{SS}^m - r\hat{V}^m$$
$$- \hat{\mathcal{R}}(t)\rho_{ins}S + \hat{\mathcal{M}}(t)f + \sum_{\substack{l=1 \\ l\neq m}}^{M} \lambda^{m\to l}(\hat{V}^l(SJ^{m\to l}, B, D, e_l, t) - \hat{V}^m) = 0. \quad (5.25)$$

For a given regime e_m, the withdrawal constraint $\mathcal{A}\hat{V}^m = \mathcal{A}\hat{V}^m(S, B, D, e_m, t)$ can be written as:

$$\mathcal{A}\hat{V}^m \equiv \max\Bigg(- \hat{\mathcal{R}}(t)\hat{\gamma}(t)S, \max_{W\in[0,S-\omega]}\big(\hat{V}^m(S - W, \max(B - W, 0), \max(D - W, 0), e_m, t)$$
$$- \hat{\mathcal{R}}(t)\hat{\gamma}(t)W\big) - c\Bigg), \quad (5.26)$$

where c is a small fixed cost. We remark that determining the optimal withdrawal amount in equation (5.26) is a local optimization problem whose solution is discussed in Section 5.3.3.

The jump condition applied at each ratchet date can be written as:

$$\hat{V}(S, B^+, D, e_m, t_o^+) = \hat{V}(S, B^-, D, e_m, t_o^-), \tag{5.27}$$

where $B^+ = \max(B^-, S)$. The initial conditions for this pricing problem are similar to those outlined in equation (5.5) and can be written as:

$$\hat{V}(S, B, D, e_m, t = T) = 0. \tag{5.28}$$

Consequently, we obtain a set of M impulse control problems which are solved simultaneously to determine the value of the GMDB guarantee. Assuming the economy is in state e_m, we solve the following equation in terms of time to maturity ($\tau = T - t$):

$$\min\left(V_\tau^m - \mathcal{L}V^m + \mathcal{R}(\tau)\rho_{ins}S - \mathcal{M}(\tau)f, V^m - \mathcal{A}V^m\right) = 0, \tag{5.29}$$

where $V(S, B, D, e_m, \tau) = \hat{V}(S, B, D, e_m, T - \tau)$ and $\mathcal{L}V^m$ is now defined as:

$$\begin{aligned}\mathcal{L}V^m = \frac{1}{2}\sigma_m^2 S^2 V_{SS}^m + \left(r - \rho_{\text{total}} - \sum_{\substack{l=1 \\ l\neq m}}^{M} \lambda^{m\to l}(J^{m\to l} - 1)\right) SV_S^m - rV^m \\ + \sum_{\substack{l=1 \\ l\neq m}}^{M} \lambda^{m\to l}(V^l(SJ^{m\to l}, B, D, e_l, t) - V^m).\end{aligned} \tag{5.30}$$

Equation (5.29) can also be written in penalized form:

$$\lim_{\epsilon \to 0} \left(V_\tau^m - \mathcal{L}V^m + \mathcal{R}(\tau)\rho_{ins} S - \mathcal{M}(\tau) f - \frac{1}{\epsilon} \max\big(\mathcal{A}V^m - V^m, 0\big) \right) = 0. \tag{5.31}$$

This set of coupled PDEs is solved, working backward in time, using an iterative penalty scheme [44] to determine the value of the guarantee at each timestep.

5.2.1 Boundary Conditions

Additional boundary conditions need to be specified to localize the regime-switching GMDB pricing problem. For each regime e_m, equation (5.31) is solved on a truncated domain: $[0, S_{\max}] \times [D, B_{\max}] \times [0, \mathcal{D}_0]$ (see Section 5.1.1).

As $S \to \infty$, the boundary condition must be considered carefully due to the presence of jumps in S when the state of the economy changes. More specifically, the case when S jumps outside the discrete domain following a regime change, i.e. $SJ^{m \to l} > S_{\max}$, must be dealt with in an appropriate manner. We assume that any asset value that jumps outside the discrete S grid is set to $S_{\max}$, which implies that the jump size $J^{m \to l}$ $(l \neq m)$ is a function of S:

$$J^{m \to l}(S) = \begin{cases} J^{m \to l} & \text{when } \ 0 \leq S \leq \frac{S_{\max}}{J^{m \to l}}, \\ \frac{S_{\max}}{S} & \text{when } \ \frac{S_{\max}}{J^{m \to l}} < S \leq S_{\max}. \end{cases} \tag{5.32}$$

Again, this is an approximation, where we expect the error to be small as $S_{\max} \to \infty$. This will be verified in some numerical tests in Section 5.5.

The penalized GMDB pricing equation with regime-switching can then be written as:

$$V_\tau^m - \mathcal{L}V^m + \mathcal{R}(\tau)\rho_{ins} S - \mathcal{M}(\tau) f - \frac{1}{\epsilon} \max\big(\mathcal{A}V^m - V^m, 0\big) = 0, \tag{5.33}$$

where:

$$\mathcal{L}V^m = \frac{1}{2}\sigma_m^2 S^2 V_{SS}^m + \left(r - \rho_{\text{total}} - \sum_{\substack{l=1 \\ l\neq m}}^{M} \lambda^{m\to l}(J^{m\to l}(S)-1)\right) S V_S^m - rV^m \tag{5.34}$$
$$+ \sum_{\substack{l=1 \\ l\neq m}}^{M} \lambda^{m\to l}(V^l(J^{m\to l}(S)S, B, D, e_l, \tau) - V^m).$$

Making the same assumption as in Section 5.1.1, the jump condition in equation (5.27) becomes:

$$V(S, B, D, e_m, \tau_o^+) = \begin{cases} V(S, B, D, e_m, \tau_o^-) & \text{if } S \leq B, \\ V(S, S, D, e_m, \tau_o^-) & \text{if } B < S \leq B_{\max}, \\ V(S, B, D, e_m, \tau_o^-) & \text{if } S > B_{\max}. \end{cases} \tag{5.35}$$

The boundary conditions for each regime are similar to those outlined in Section 5.1.1 and are:

$$V_\tau^m + rV^m - \mathcal{M}(\tau)(B + \gamma(\tau)D) - \sum_{\substack{l=1 \\ l\neq m}}^{M} \lambda^{m\to l}(V^l(0, B, D, e_l, \tau) - V^m) = 0 \text{ for } S = 0, \tag{5.36}$$

$$V_\tau^m + \mathcal{R}(\tau)\rho_{ins}S + \rho_{\text{total}}V^m - \sum_{\substack{l=1 \\ l\neq m}}^{M} \lambda^{m\to l} J^{m\to l}(S)\big(V^l(S, B, D, e_l, \tau) - V^m\big) \tag{5.37}$$
$$-\frac{1}{\epsilon}\max(\mathcal{A}V^m - V^m, 0) = 0 \text{ for } S = S_{\max},$$

while the usual pricing equation in (5.33) is solved on the boundaries of the $B \times D$ plane.

5.3 Numerical Solution of the GMDB Problem with Regime-Switching

In this section, we present details for the numerical solution of the GMDB pricing problem. Section 5.3.1 describes the construction of the underlying grid in each regime, while Section 5.3.2 presents the discrete equations for the GMDB pricing problem. Finally, Section 5.3.3 discusses how the local optimization problem is handled when determining the value of the partial withdrawal constraint.

5.3.1 Construction of the Underlying Grid

The regime-switching pricing problem requires the addition of a fourth dimension to the solution domain. As such, we build a discrete $S \times B \times D \times E$ domain where the E grid contains M entries. The pricing equations for each economic state e_m are solved on a discrete three-dimensional $[0, S_{\max}] \times [D, B_{\max}] \times [0, \mathcal{D}_0]$ grid. The three-dimensional $S \times B \times D$ grid is identical for each economic state e_m and its construction is now described in more detail.

A set of discrete nodes is built in the D direction and denoted as $\{D_k\}$, for $k = 0, \ldots, k_{\max}$. Note that $D_0 = 0$ and $D_{k_{\max}} = \mathcal{D}_0$, where $\mathcal{D}_0$ is the initial deposit made by the policy owner. Associated with each D_k value is a two-dimensional $S \times B$ plane. Since equation (5.13) contains no derivatives in the B direction, each $S \times B$ plane is a collection of one-dimensional problems. As the domain definition for B depends on D_k, we begin by building a $S \times B$ grid on the larger $[0, B_{\max}] \times [0, S_{\max}]$ domain which is then truncated to the required size.

To construct the $S \times B$ grid, we build a set of nodes over the domain $[0, B_{\max}]$ which we denote by $\{B_l\}$ for $l = 0, \ldots, l_{\max}$ where:

- $B_0 = 0$ and $B_{l_{\max}} = B_{\max}$,

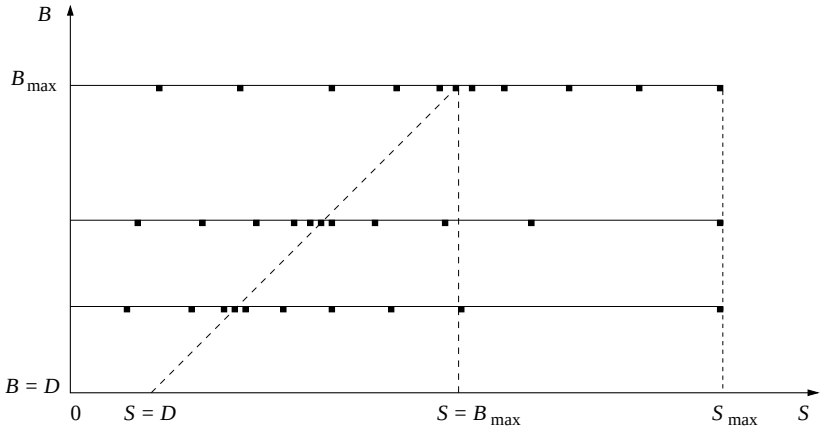


FIGURE 5.1: *Representation of a* $[0, S_{\max}] \times [D, B_{\max}]$ *plane where each one-dimensional* S *grid is built using the scaled grid technique defined in equation* (5.38).

- each of the nodes in $\{D_k\}$ is also included in the grid $\{B_l\}$, i.e. $\{D_k\} \subset \{B_l\}$, and
- the bulk of the nodes in $\{B_l\}$ are placed around the initial deposit amount $\mathcal{D}_0$.

For a fixed B_l, we construct a set of S nodes $\{S_i^l\}$ using the *scaled grid* construction (see Section 2.2.1):

$$S_i^l = B_i \frac{B_l}{\mathcal{D}_0} \qquad \text{for } i = 0, \ldots, l_{\max} - 1, \qquad \text{and} \qquad S_{i_{\max}}^l = \frac{(B_{l_{\max}})^2}{\mathcal{D}_0}. \tag{5.38}$$

This *scaled grid* construction enables a more precise calculation of the jump condition in equation (5.17). Note that interpolation is generally required when calculating the jump condition in (5.17) on a *scaled grid*.

As specified previously, the discrete $S \times B$ grid is truncated according to the deposit value D_k to which it is associated. More specifically, we will eliminate those entries in $\{B_l\}$ (and their associated S-grid) such that $B_l < D_k$. Therefore, assuming we have determined the index p such that $B_p = D_k$, the grid along the B direction is truncated

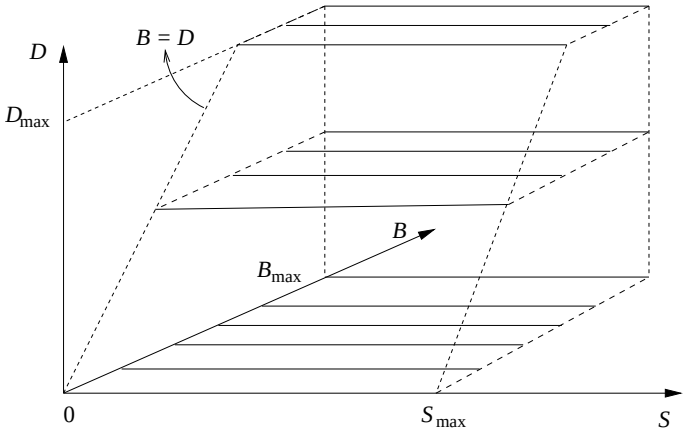


FIGURE 5.2: *Three dimensional solution domain to price the GMDB guarantee in economic state e_m. Each $S \times B$ plane is constructed as in Figure 5.1.*

such that:

$$B_j^k = B_{p+j} \qquad \text{for } j = 0, \ldots, j_{\max}. \tag{5.39}$$

Note that the notation B_j^k in (5.39) was chosen to highlight the dependence of the truncated B-grid on D_k. The resulting $S \times B$ grid for a fixed deposit amount D_k is shown in Figure 5.1.

This grid construction ensures that we use the minimum number of nodes to solve the GMDB pricing problem for each economic state e_m. The final three-dimensional $S \times B \times D$ domain is presented in Figure 5.2.

5.3.2 Discrete Equations

Let us now consider the discretization of equation (5.29) on the $S \times B \times D \times E$ domain. We denote the discrete value as $V_{i,j,k,m}^{n+1} = V(S_i^j, B_j^k, D_k, e_m, \tau^{n+1})$, while $\mathcal{A}^h V_{i,j,k,m}^{n+1} = \mathcal{A}V(S_i^j, B_j^k, D_k, e_m, \tau^{n+1})$ represents the discrete version of the withdrawal constraint defined in equation (5.26). In terms of notation, discrete operators will be denoted as $\mathcal{A}^h$ and $\mathcal{L}^h$ where the superscript h represents the space discretization parameter.

Assuming fully implicit timestepping is used, the discrete form of equation (5.29) is obtained by applying standard finite difference approximations:

$$\frac{V^{n+1}_{i,j,k,m} - V^{n}_{i,j,k,m}}{\Delta\tau} = [\mathcal{L}^h V]^{n+1}_{i,j,k,m} - \mathcal{R}^{n+1}\rho_{ins}S^j_i + \mathcal{M}^{n+1} f^{n+1}_{i,j,k} + \frac{\mu^{n+1}_{i,j,k,m}}{\epsilon}\big(\mathcal{A}^h V^{n+1}_{i,j,k,m} - V^{n+1}_{i,j,k,m}\big), \tag{5.40}$$

where

$$\mathcal{M}^{n+1} = \mathcal{M}(\tau^{n+1}), \quad \mathcal{R}^{n+1} = \mathcal{R}(\tau^{n+1}), \quad \gamma^{n+1} = \gamma(\tau^{n+1}), \tag{5.41}$$

$$f^{n+1}_{i,j,k} = f(S^j_i, B^k_j, D_k, \tau^{n+1}) = \max(B^k_j - S^j_i, 0) + \gamma^{n+1} D_k, \tag{5.42}$$

$$\mu^{n+1}_{i,j,k,m} = \begin{cases} 1 & \text{if } \mathcal{A}^h V^{n+1}_{i,j,k,m} > V^{n+1}_{i,j,k,m}, \\ 0 & \text{otherwise.} \end{cases} \tag{5.43}$$

The discrete differential operator $\mathcal{L}^h$ can be written as:

$$\begin{aligned}[\mathcal{L}^h V]^{n+1}_{i,j,k,m} = {} & \alpha_{i,j,m} V^{n+1}_{i-1,j,k,m} + \beta_{i,j,m} V^{n+1}_{i+1,j,k,m} - (\alpha_{i,j,m} + \beta_{i,j,m} + r) V^{n+1}_{i,j,k,m} \\ & + \sum_{\substack{l=1 \\ l \neq m}}^{M} \lambda^{m\to l} (\mathcal{H}(J^{m\to l})_i V^{n+1}_{j,k,l} - V^{n+1}_{i,j,k,m}),\end{aligned} \tag{5.44}$$

where $\alpha_{i,j,m}, \beta_{i,j,m} \geq 0$ are defined in Appendix A and $\mathcal{H}(J^{m\to l})_i V^{n+1}_{j,k,l}$ represents the interpolated guarantee value in regime e_l when the asset price jumps to $J^{m\to l}(S)S$. Assuming linear interpolation is chosen, we have:

$$\mathcal{H}(J^{m\to l})_i V^{n+1}_{j,k,l} = (1 - w_{i,j,m}) V^{n+1}_{a,j,k,l} + w_{i,j,m} V^{n+1}_{a+1,j,k,l}\ , \tag{5.45}$$

where $S^j_a \leq J^{m\to l}(S^j_i) S^j_i \leq S^j_{a+1}$ and the interpolation weight $0 \leq w_{i,j,m} \leq 1$ can be

written as:

$$w_{i,j,m} = \frac{J^{m\to l}(S_i^j)S_i^j - S_a^j}{S_{a+1}^j - S_a^j}. \tag{5.46}$$

Since the node $(S_i^j - W, \max(B_j^k - W, 0), \max(D_k - W, 0))$ does not always coincide with an existing grid node, interpolation must be used when calculating the discrete withdrawal constraint $\mathcal{A}^h V_{i,j,k,m}^{n+1}$. We define the vector $\mathcal{I}(W)_{i,j,k}$ as the interpolation operator used when calculating the value of the GMDB guarantee following a withdrawal W. Thus, we have:

$$\mathcal{A}^h V_{i,j,k,m}^{n+1} = \max\left(-R^{n+1}\gamma^{n+1}S_i^j, \max_{W\in[0,S_i^j-\omega]}\left[\mathcal{I}(W)_{i,j,k}V_m^{n+1} - R^{n+1}\gamma^{n+1}W\right] - c\right), \tag{5.47}$$

where V_m^{n+1} is a vector containing the GMDB values for regime e_m:

$$V_m^{n+1} = \begin{bmatrix} V_{0,0,0,m}^{n+1} \\ V_{1,0,0,m}^{n+1} \\ \vdots \\ V_{i_{\max}-1,j_{\max},k_{\max},m}^{n+1} \\ V_{i_{\max},j_{\max},k_{\max},m}^{n+1} \end{bmatrix}, \tag{5.48}$$

and $\mathcal{I}(W)_{i,j,k}$ can be written as follows assuming linear interpolation:

$$\mathcal{I}(W)_{i,j,k}V_m^{n+1} = \sum_{i,j,k}\eta_{i,j,k,m}V_{i,j,k,m}^{n+1}, \tag{5.49}$$

where $0 \le \eta_{i,j,k,m} \le 1$ are the interpolation weights and:

$$\sum_{i,j,k}\eta_{i,j,k,m} = 1. \tag{5.50}$$

Letting $W^{n+1}_{i,j,k,m}$ denote the optimal withdrawal amount at node (S^j_i, B^k_j, D_k, e_m) and time τ^{n+1}, and defining the indicator variable $a^{n+1}_{i,j,k,m}$ as:

$$a^{n+1}_{i,j,k,m} = \begin{cases} 1 & \text{if it is optimal to lapse,} \\ 0 & \text{if it is optimal to withdraw } W^{n+1}_{i,j,k,m}, \end{cases} \tag{5.51}$$

we can rewrite equation (5.47) as:

$$\mathcal{A}^h V^{n+1}_{i,j,k,m} = -a^{n+1}_{i,j,k,m} R^{n+1}\gamma^{n+1} S^j_i + (1 - a^{n+1}_{i,j,k,m}) \left(\mathcal{I}(W)_{i,j,k} V^{n+1}_m - R^{n+1}\gamma^{n+1} W^{n+1}_{i,j,k,m} - c \right). \tag{5.52}$$

The numerical scheme in equation (5.40) is a positive coefficient discretization [43] when the following definition is satisfied.

Definition 5.1 (Positive Coefficient Scheme). *The numerical scheme defined in equation* (5.40) *is a positive coefficient discretization when:*

$$\begin{aligned} \alpha_{i,j,m}\ ,\ \beta_{i,j,m} &\geq 0, \qquad \forall i, j, m\ , \\ r &\geq 0\ , \\ \lambda^{m \to l} &\geq 0, \qquad \textit{when } m \neq l\ , \end{aligned}$$

and the interpolation operators $\mathcal{H}(J^{m\to l})_i$ and $\mathcal{I}(W)_{i,j,k}$ represent linear interpolation.

Since $\alpha_{i,j,m},\ \beta_{i,j,m} \geq 0$ by construction (see Appendix A), $\lambda^{m\to l} \geq 0$, when $m \neq l$ and $r \geq 0$ for all problems considered, the numerical scheme in (5.40) is a positive coefficient scheme.

5.3.3 Optimal Withdrawal

In this section, we outline the method used to determine the optimal withdrawal W when calculating the constraint in equation (5.26) at a discrete grid node (S_i^j, B_j^k, D_k, e_m). We solve this local optimization problem by considering all possible discrete withdrawals.

After checking that a withdrawal is possible by verifying $S_i^j > \omega$, where ω is the minimal deposit amount, we carry out a linear search over all possible discrete withdrawals $\bar{W}$ where:

$$\bar{W} = \min(S_l^j, S_i^j - \omega), \tag{5.53}$$

assuming $S_l^j < S_i^j$. For each $\bar{W}$ considered, we calculate the effect of the partial withdrawal to the issuer, denoted by $A(\bar{W})$:

$$A(\bar{W}) = \mathcal{I}(\bar{W})_{i,j,k} V_m^{n+1} - \mathcal{R}^{n+1}\gamma^{n+1}\bar{W}, \tag{5.54}$$

where $\mathcal{I}(\bar{W})_{i,j,k}$ is defined in (5.49).

The optimal withdrawal is determined by taking the maximum of $A(\bar{W})$ over all discrete withdrawals $\bar{W}$ and the final withdrawal constraint for node (S_i^j, B_j^k, D_k, e_m) is computed as:

$$\mathcal{A}^h V_{i,j,k,m}^{n+1} = \max\left(-\mathcal{R}^{n+1}\gamma^{n+1}S_i^j, \max_{\bar{W}}\left[A(\bar{W})\right] - c\right). \tag{5.55}$$

This search procedure is summarized in Algorithm 5.1.

5.4 Convergence to the Viscosity Solution

In [81], the authors demonstrate how some reasonable discretization schemes either never converge or converge to a wrong solution. Thus, it is important to ensure that our

$\bar{W} = 0\ ;\ A = 0\ ;\ A_{\max} = 0$

if $S_i^j > \omega$ **then**

Determine maximum withdrawal: $\bar{W} = S_i^j - \omega$

Calculate: $A_{\max} = \mathcal{I}(\bar{W})_{i,j,k} V_m^{n+1} - \mathcal{R}^{n+1}\gamma^{n+1}\bar{W}$

Determine index $i_{\max}$ s.t.: $S_{i-1}^j < S_{i_{\max}}^j < S_i^j - \omega$

for $l = 0, \ldots, i_{\max}$ **do**

Determine withdrawal: $\bar{W} = S_l^j$

Calculate: $A = \mathcal{I}(\bar{W})_{i,j,k} V_m^{n+1} - \mathcal{R}^{n+1}\gamma^{n+1}\bar{W}$

$A_{\max} = \max(A, A_{\max})$

end for

end if

$\mathcal{A}^h V_{i,j,k}^{n+1} = \max\left(A_{\max} - c, -\mathcal{R}^{n+1}\gamma^{n+1}S_i^j\right)$

ALGORITHM 5.1: *Calculation of Withdrawal Constraint for GMDB Contracts*

discretization method converges to the unique viscosity solution [31]. Assuming that a unique, continuous viscosity solution to equation (5.40) exists, the numerical scheme in (5.40) converges to the viscosity solution away from the ratchet dates if it satisfies certain stability, consistency and monotonicity requirements [8, 11].

Assuming a given state e_m, the solution domain for the GMDB pricing problem in equation (5.29) is $[0, S_{\max}] \times [D, B_{\max}] \times [0, \mathcal{D}_0]$. When working backward in time, we denote the ratchet dates as τ_o^u for $u = 0, \ldots, u_{\max}$, and use τ_o^{u-} and τ_o^{u+} to denote the times right before and after a ratchet event. Thus, we define the solution domains Π_u and Π by:

$$\Pi_u = [0, S_{\max}] \times [D, B_{\max}] \times [0, \mathcal{D}_0] \times [\tau_o^{u+}, \tau_o^{(u+1)-}] \text{ for } u = 0, \ldots, u_{\max} - 1, \text{ and}$$

$$\Pi = \bigcup_u \Pi_u = [0, S_{\max}] \times [D, B_{\max}] \times [0, \mathcal{D}_0] \times \bigcup_u [\tau_o^{u+}, \tau_o^{(u+1)-}].$$

This enables us to define the pricing problem for the GMDB guarantee in detail.

Definition 5.1 (GMDB Pricing Problem with Discrete Ratchets). *The pricing*

problem for the GMDB guarantee with discrete ratchet events is defined in Π *as follows: within each domain* Π_u*, for* $u = 0, \ldots, u_{\max} - 1$*, we determine the solution to the pricing problem presented in equation* (5.29) *with initial conditions expressed in equation* (5.28) *when* $u = 0$ *or in equation* (5.35) *when* $u > 0$*, boundary conditions described in equations* (5.36)–(5.37) *and localization conditions in equations* (5.32) *and* (5.35)*.*

Note that we have not defined the pricing problem for the GMDB guarantee over the entire contract lifetime $\tau \in [0, T]$ since the solution can be discontinuous across ratchet dates τ_o^u, for $u = 0, \ldots, u_{\max} - 1$, due to the no-arbitrage condition in equation (5.35).

Assumption 5.2. *We assume that a unique, continuous viscosity solution exists [8, 67, 78] for the localized pricing problem in Definition 5.1 which satisfies equations* (5.36)–(5.37) *and localization conditions in equations* (5.32) *and* (5.35)*. More specifically, we assume that the unique viscosity solution is continuous within each domain* Π_u*, for* $u = 0, \ldots, u_{\max} - 1$*.*

A unique, continuous viscosity solution exists if the PDE satisfies a strong comparison property (see Section 2.3). In the regime switching case, existence of a continuous, viscosity solution is shown using properties of the value function in [78]. Note that the definition of viscosity solution has to be generalized for systems of weakly coupled PDEs, such as regime switching models [67, 78].

If Assumption 5.2 holds, then showing that the discrete equations are monotone, stable and consistent will enable us to conclude that the solution of the numerical scheme in equation (5.40) converges to the unique viscosity solution of the pricing problem outlined in Definition 5.1.

5.4.1 Stability

We begin by showing that the discrete equations in (5.40) satisfy the l_∞-stability requirement which involves demonstrating that the discrete contract value $V^{n+1}_{i,j,k,m}$ is bounded. We define:

$$\Delta S^j_{\max} = \max_i (S^j_{i+1} - S^j_i),\ \Delta B^k_{\max} = \max_j (B^k_{j+1} - B^k_j),\ \Delta D_{\max} = \max_k (D_{k+1} - D_k),\ \Delta\tau = \frac{T}{N}.$$

Definition 5.3 (Stability). *For fixed $S_{\max}$, $B_{\max}$ and T, the numerical scheme presented in equation* (5.40) *is l_∞-stable if:*

$$||V^n||_\infty \leq C \tag{5.56}$$

for $0 \leq n \leq N$, as $\Delta\tau \to 0$, $\max_j \Delta S^j_{\max} \to 0$, $\max_k \Delta B^k_{\max} \to 0$, $\Delta D_{\max} \to 0$ and $\epsilon \to 0$. The constant C is independent of $\Delta\tau$, $\Delta S^j_{\max}$, $\Delta B^k_{\max}$, $\Delta D_{\max}$ and ϵ.

For notational convenience, we make the following assumption.

Assumption 5.4. *We assume that $\Delta B^k_{\max}$, $\Delta S^j_{\max}$, $\Delta\tau$ and ϵ are parametrized by:*

$$\Delta B^k_{\max} = c_0 h,\ \Delta S^j_{\max} = c_1 h,\ \Delta\tau = c_2 h \text{ and } \epsilon = c_3 h,$$

with c_0, c_1, c_2 and c_3 constants.

Theorem 5.5. *Assume the numerical scheme satisfies Definition 5.1, that the boundary conditions are described by the discrete version of equations* (5.36)–(5.37)*, that the initial conditions are given by the discrete version of equation* (5.28) *and that fully implicit timestepping is used. Then:*

$$-S^j_i \leq V^{n+1}_{i,j,k,m} \leq \mathcal{C}^{n+1}_0 B_{\max} + \mathcal{C}^{n+1}_1 D_{\max} \quad \forall i,j,k,m,n, \tag{5.57}$$

where the parameters $0 \leq \mathcal{C}_0^{n+1} \leq 1$ *and* $0 \leq \mathcal{C}_1^{n+1}$ *are defined as:*

$$\mathcal{C}_0^{n+1} = \Delta\tau \sum_{i=0}^{n+1} \mathcal{M}^i \quad \text{and} \quad \mathcal{C}_1^{n+1} = \Delta\tau \sum_{i=0}^{n+1} \mathcal{M}^i \gamma^i. \tag{5.58}$$

Proof. See Appendix H. □

Theorem 5.5 enables us to conclude that the numerical scheme for $V_{i,j,k,m}^{n+1}$, as defined in equation (5.40), is stable according to Definition 5.3.

5.4.2 Monotonicity

In this section, we show that the discrete equations presented in (5.40) are monotone. To facilitate exposition, we denote the discrete equations on interior nodes (when $S_i^j < S_{\max}$) as:

$$\begin{aligned}\mathcal{G}\big(h, x, V_{i,j,k,m}^{n+1}, V_{i,j,k,m}^{n}, \{V_{a,p,u,l}^{n+1}\}\big) = \frac{V_{i,j,k,m}^{n+1} - V_{i,j,k,m}^{n}}{\Delta\tau} - [\mathcal{L}^h V]_{i,j,k,m}^{n+1} + \mathcal{R}^{n+1}\rho_{ins} S_i^j \\ - \mathcal{M}^{n+1} f_{i,j,k}^{n+1} - \frac{1}{\epsilon}\max\left(\mathcal{A}^h V_{i,j,k,m}^{n+1} - V_{i,j,k,m}^{n+1}, 0\right),\end{aligned} \tag{5.59}$$

where $x = (S_i^j, B_j^k, D_k, e_m, \tau^{n+1})$, h is the discretization parameter, and $\{V_{a,p,u,l}^{n+1}\}$ represents all discrete nodes, other than $V_{i,j,k,m}^{n+1}$ and $V_{i,j,k,m}^{n}$, included in the discrete equations. Similarly, at the boundary when $S_i^j = S_{\max}$, the discretization is given as:

$$\begin{aligned}\mathcal{G}\big(h, x, V_{i_{\max},j,k,m}^{n+1}, V_{i_{\max},j,k,m}^{n}, \{V_{a,p,u,l}^{n+1}\}\big) = \frac{V_{i_{\max},j,k,m}^{n+1} - V_{i_{\max},j,k,m}^{n}}{\Delta\tau} + \rho_{\text{total}} V_{i_{\max},j,k,m}^{n+1} \\ + \mathcal{R}^{n+1}\rho_{ins} S_{i_{\max}}^j - \sum_{\substack{l=1 \\ l\neq m}}^{M} \lambda^{m\to l} J_{i_{\max}}^{m\to l} (V_{i_{\max},j,k,l}^{n+1} - V_{i_{\max},j,k,m}^{n+1}) \\ - \frac{1}{\epsilon}\max\left(\mathcal{A}^h V_{i_{\max},j,k,m}^{n+1} - V_{i_{\max},j,k,m}^{n+1}, 0\right).\end{aligned} \tag{5.60}$$

Definition 5.6 (Monotonicity). *The numerical scheme* $\mathcal{G}(h, x, V_{i,j,k,m}^{n+1}, V_{i,j,k,m}^{n}, \{V_{a,p,u,l}^{n+1}\})$

presented in equations (5.59) *and* (5.60) *is monotone if for all* $Y^n_{i,j,k,m} \geq V^n_{i,j,k,m}$*:*

$$\mathcal{G}(h, x, V^{n+1}_{i,j,k,m}, Y^n_{i,j,k,m}, \{Y^{n+1}_{a,p,u,l}\}) - \mathcal{G}(h, x, V^{n+1}_{i,j,k,m}, V^n_{i,j,k,m}, \{V^{n+1}_{a,p,u,l}\}) \leq 0. \tag{5.61}$$

Note that this definition of monotonicity is equivalent to the one presented in [8].

Theorem 5.7 (Monotone Discretization). *Assuming that the discretization satisfies Condition* (5.1)*, the numerical scheme* $\mathcal{G}(h, x, V^{n+1}_{i,j,k,m}, V^n_{i,j,k,m}, \{V^{n+1}_{a,p,u,l}\})$ *defined in equations* (5.59) *and* (5.60)*, is monotone.*

Proof. Notice that the numerical scheme presented in equations (5.59) and (5.60) is a positive coefficient discretization since it satisfies Condition 5.1. In [43], the authors demonstrate that a positive coefficient discretization of a control problem, such as the one considered here, is monotone. Using the same technique as in [43], it is straightforward to show that the numerical scheme presented in equations (5.59) and (5.60) is monotone and satisfies Definition 5.6. □

5.4.3 Consistency

We now show that the numerical scheme in equation (5.40) is consistent.

For the GMDB pricing problem, the impulse control problem can be written in compact form as:

$$F(V(x)) = 0 \text{ for all } x = (S, B, D, e_m, \tau), \tag{5.62}$$

where

$$F(V(x)) = \begin{cases} F_{in}(V(x)) & \text{if } S < S_{\max}, \\ F_{bound}(V(x)) & \text{if } S = S_{\max}. \end{cases} \tag{5.63}$$

The continuous problem evaluated at discrete interior nodes when $S_i^j < S_{\max}$ is then:

$$F_{in}(V)_{i,j,k,m}^{n+1} = \left[\min\left(V_\tau - \mathcal{L}V + \mathcal{R}(\tau)\rho_{ins}S - \mathcal{M}(\tau)f, V - \mathcal{A}V\right)\right]_{i,j,k,m}^{n+1} = 0, \quad (5.64)$$

while at boundary nodes when $S_i^j = S_{\max}$ we have:

$$F_{bound}(V)_{i_{\max},j,k,m}^{n+1} = \left[\min\left(V_\tau + \rho_{\text{total}}V - \sum_{\substack{l=1 \\ l\neq m}}^{M} \lambda^{m\to l} J^{m\to l}(S)\big(V(S,B,D,e_l,\tau) - V\big) + \mathcal{R}(\tau)\rho_{ins}S, V - \mathcal{A}V\right)\right]_{i_{\max},j,k,m}^{n+1} = 0, \quad (5.65)$$

where the continuous operator $\mathcal{L}$ is defined in equation (5.30) and $f = f(S,B,D,\tau)$ is defined in equation (5.2).

Since $\epsilon > 0$, the discrete scheme in equation (5.59) can be rewritten as:

$$\begin{aligned}
&\hat{\mathcal{G}}\big(h, x, V_{i,j,k,m}^{n+1}, V_{i,j,k,m}^{n}, \{V_{a,p,u,l}^{n+1}\}\big) = \\
&\min\left(\epsilon\left(\frac{V_{i,j,k,m}^{n+1} - V_{i,j,k,m}^{n}}{\Delta\tau} - [\mathcal{L}^h V]_{i,j,k,m}^{n+1} + \mathcal{R}^{n+1}\rho_{ins}S_i^j - \mathcal{M}^{n+1}f_{i,j,k}^{n+1}\right) + V_{i,j,k,m}^{n+1} - \mathcal{A}^h V_{i,j,k,m}^{n+1},\right. \\
&\left.\frac{V_{i,j,k,m}^{n+1} - V_{i,j,k,m}^{n}}{\Delta\tau} - [\mathcal{L}^h V]_{i,j,k,m}^{n+1} + \mathcal{R}^{n+1}\rho_{ins}S_i^j - \mathcal{M}^{n+1}f_{i,j,k}^{n+1}\right) = 0, \quad (5.66)
\end{aligned}$$

at interior nodes when $S_i^j < S_{\max}$, while equation (5.60) can be rewritten as:

$$\begin{aligned}\hat{\mathcal{G}}\big(h, x, V^{n+1}_{i_{\max},j,k,m}, V^{n}_{i_{\max},j,k,m}, \{V^{n+1}_{a,p,u,l}\}\big) = \min\Bigg(\epsilon\Bigg(&\frac{V^{n+1}_{i_{\max},j,k,m} - V^{n}_{i_{\max},j,k,m}}{\Delta\tau} + \rho_{\text{total}} V^{n+1}_{i_{\max},j,k,m} \\ &- \sum_{\substack{l=1 \\ l\neq m}}^{M} \lambda^{m\to l} J^{m\to l}_{i_{\max}} (V^{n+1}_{i_{\max},j,k,l} - V^{n+1}_{i_{\max},j,k,m}) + \mathcal{R}^{n+1}\rho_{ins} S^j_{i_{\max}}\Bigg) \\ &+ V^{n+1}_{i_{\max},j,k,m} - \mathcal{A}^h V^{n+1}_{i_{\max},j,k,m}, \frac{V^{n+1}_{i_{\max},j,k,m} - V^{n}_{i_{\max},j,k,m}}{\Delta\tau} + \rho_{\text{total}} V^{n+1}_{i_{\max},j,k,m} \\ &- \sum_{\substack{l=1 \\ l\neq m}}^{M} (\lambda^{m\to l} J^{m\to l})_{i_{\max}} (V^{n+1}_{i_{\max},j,k,l} - V^{n+1}_{i_{\max},j,k,m}) + \mathcal{R}^{n+1}\rho_{ins} S^j_{i_{\max}}\Bigg) = 0, \quad (5.67)\end{aligned}$$

on the boundary when $S_i^j = S_{\max}$.

To formally define the notion of consistency, we require the concept of upper and lower semi-continuous envelope of a function from Definition 2.7.

Definition 5.8 (Consistency). *For any smooth test function ϕ with bounded derivatives of all orders with respect to S and τ, the numerical scheme $\hat{\mathcal{G}}(h, x, \phi^{n+1}_{i,j,k,m}, \phi^{n}_{i,j,k,m}, \{\phi^{n+1}_{a,p,u,l}\})$ is consistent if, for all points $\hat{x} = (\hat{S}, \hat{B}, \hat{D}, e_m, \hat{\tau})$ with $x = (S^j_i, B^k_j, D_k, e_m, \tau^{n+1})$, we have:*

$$\limsup_{\substack{h,\xi\to 0 \\ x\to\hat{x}}} \hat{\mathcal{G}}\big(h, x, \phi^{n+1}_{i,j,k,m} + \xi, \phi^{n}_{i,j,k,m} + \xi, \{\phi^{n+1}_{a,p,u,l} + \xi\}\big) \leq F^*(\phi(\hat{x})), \quad (5.68)$$

$$\liminf_{\substack{h,\xi\to 0 \\ x\to\hat{x}}} \hat{\mathcal{G}}\big(h, x, \phi^{n+1}_{i,j,k,m} + \xi, \phi^{n}_{i,j,k,m} + \xi, \{\phi^{n+1}_{a,p,u,l} + \xi\}\big) \geq F_*(\phi(\hat{x})), \quad (5.69)$$

where $\phi^n_{i,j,k,m} = \phi(S^j_i, B^k_j, D_k, e_m, \tau^n)$ and $\xi \geq 0$.

Note that Remark 2.9 also applies to the GMDB pricing problem.

Theorem 5.9 (Consistent Discretization). *The numerical scheme presented in equation* (5.40) *is consistent according to Definition 5.8.*

Proof. See Appendix I. □

5.5 Results from Numerical Experiments

This section focuses on determining the *fair insurance charge* associated with a GMDB guarantee from the issuer's perspective. More specifically, we are looking for ρ_{ins} such that:

$$V(\rho_{ins}; S = \mathcal{D}_0, B = \mathcal{D}_0, D = \mathcal{D}_0, E = e_m, \tau = T) = 0, \tag{5.70}$$

where $\mathcal{D}_0$ is the initial deposit made by the contract owner and T is the contract maturity in years. Newton iteration is used to determine the fair insurance charge ρ_{ins} that satisfies equation (5.70) assuming an economic state e_m. The Newton iteration tolerance, denoted by tol, ensures that:

$$|\rho_{ins}^{k+1} - \rho_{ins}^{k}|_\infty \leq tol, \tag{5.71}$$

where $tol = 1 \times 10^{-6}$ and k is the iteration index. Unless otherwise stated, this tolerance level is used for all numerical results included in this section.

As a validation exercise, we compared our results with those presented in [72] when valuing a special GMDB contact with continuous ratchet updates, a 25 year maturity and no withdrawals; details of this comparison are presented in Appendix J.

5.5.1 Results for Constant Volatility

Assuming volatility is constant, we present numerical results when pricing the GMDB guarantee. In a regime-switching sense, this is equivalent to assuming that only one economic state, e_0, exists. The volatility associated with e_0, as well as other contract parameters, are presented in Table 5.1. We are looking to determine the insurance fee

State Information - e_0	
σ_0 - Volatility	0.20
Contract Information	
r - Interest rate	0.06
ρ_{man} - Management fees	0.015
Ratchet interval	1 year
Last Ratchet Date	30 years
T - Contract maturity	40 years
Grid Construction	
$\mathcal{D}_0$ - Initial deposit	\$100
$S_{\max}$ - Grid parameter	$\$3.6 \times 10^7$
$B_{\max}$ - Grid parameter	\$60000

TABLE 5.1: *Parameter values used when pricing the GMDB guarantee in the classic Black-Scholes context.*

ρ_{ins} which satisfies:

$$V(\rho_{ins}; S = \$100, B = \$100, D = \$100, E = e_0, \tau = T) = 0. \tag{5.72}$$

Additional assumptions are necessary regarding the owner of the GMDB contract. We assume that the owner of the variable annuity is a male of 50 years of age at the time of purchase. As such, the accumulation period of the contract, during which there are periodical ratchet events, will last 30 years. The contract is assumed to come to maturity when the owner turns 90 years old which implies that $T = 40$ years, as reflected in Table 5.1. The mortality data used to price the GMDB guarantee is taken from the *Complete life table, Canada, 1995-1997* for males and females found in [29].

Table 5.1 also specifies some grid construction details. While an unequally spaced grid containing 36 nodes is built along S, the grid built in the D direction contains 21 nodes spanning $[0, \mathcal{D}_0]$. Though not presented here, numerical tests were carried out to ensure that the choice of $B_{\max}$, and consequently $S_{\max}$, provides a minimum of 6 digits of accuracy. Recall that $S_{\max} = B_{\max}^2/\mathcal{D}_0$, where $\mathcal{D}_0$ is the initial deposit (see Section 5.3.1

for more details). Similarly, numerical tests show that choosing a sufficiently small fixed cost, such as $c = 1 \times 10^{-10}$, results in values identical to those obtained when $c = 0$ up to at least 6 digits. Consequently, for all numerical experiments in this section, we set $c = 1 \times 10^{-10}$. The penalty parameter ϵ (see equations (5.59) and (5.60)) is set to $\epsilon = \Delta\tau \times 10^{-6}$.

In addition to the parameters in Table 5.1, the surrender charge imposed when a withdrawal occurs (denoted as $\hat{\gamma}(t)$ in equation (5.10)) is defined as in [74]:

$$\hat{\gamma}(t) = \begin{cases} 0.08 - 0.01\lceil t \rceil & t \leq 7 \text{ years}, \\ 0.00 & t > 7 \text{ years}, \end{cases} \tag{5.73}$$

where $\lceil \cdot \rceil$ represents the ceiling function.

To determine the accuracy level that can be attained, we carry out a convergence analysis when pricing the GMDB guarantee. Table 5.2 holds the cost of the GMDB guarantee assuming $\omega = \$80$ for different refinement levels when the parameters in Table 5.1 are used. Note that we have set $\rho_{ins} = 0.008$ for the time being. The top section of Table 5.2 contains the values obtained when fully implicit timestepping is used while the bottom panel presents the values recovered when Crank-Nicolson timestepping is used. Constant timesteps are taken for both fully implicit and Crank-Nicolson timestepping and the initial timestep is $\Delta\tau = 0.05$ years on the coarsest grid. To eliminate oscillations in the final Crank-Nicolson solution, two fully implicit timesteps are taken at the start of the solution process [82]. Note that Crank-Nicolson is not monotone, and hence is not guaranteed to converge to the viscosity solution.

We see that the results for the highest refinement level in Table 5.2 provide an acceptable level of accuracy. However, results from higher refinement levels would be required to establish a definite conclusion about the convergence of the numerical scheme with

Cost of a GMDB guarantee						
Refinement Level	Nodes S	B	D	Option Value	Difference	Ratio
Fully Implicit						
0	36	36	21	1.653844	n.a.	n.a.
1	71	71	41	1.728004	0.074161	n.a.
2	141	141	81	1.752456	0.024452	3.03
Crank-Nicolson						
0	36	36	21	1.711003	n.a.	n.a.
1	71	71	41	1.761588	0.050585	n.a.
2	141	141	81	1.769926	0.008338	6.07

TABLE 5.2: *Cost of the GMDB guarantee when the owner is assumed to be a male of* 50 *years old at the time of purchase,* $\omega = \$80$ *and* $\rho_{ins} = 0.008$*. Other contract parameters are presented in Table 5.1. Nodes - B indicates the maximum number of nodes in the B direction (i.e. when* $D = 0$*). The initial timestep is* $\Delta\tau = 0.05$ *years on the coarsest grid.*

Fair Insurance Fee for GMDB Guarantee				
Refinement Level	Nodes S	B	D	Insurance Fee (ρ_{ins})
0	36	36	21	0.009255
1	71	71	41	0.009225
2	141	141	81	0.009216

TABLE 5.3: *Fair insurance fee* (ρ_{ins}) *for a GMDB guarantee for different grid refinement levels when the owner is assumed to be a male of* 50 *years old at the time of purchase and* $\omega = \$80$*. Crank-Nicolson timestepping is used and the initial timestep is* $\Delta\tau = 0.05$ *years on the coarsest grid. Other contract parameters are presented in Table 5.1. Nodes - B indicates the maximum number of nodes in the B direction (i.e. when* $D = 0$*).*

both timestepping methods considered. Clearly the results in Table 5.2 show that the convergence has not settled down to the asymptotic rate. Results from higher refinement levels were not generated due to the prohibitive running time for such large problems (recall that this is a three-dimensional problem). Nonetheless, since our interest lies in determining the fair insurance fee associated with the contract, the results in Table 5.2 provide adequate accuracy for our purposes.

Table 5.3 presents the convergence of the fair risk charge obtained when Crank-Nicolson timestepping is used. We again assume that the owner is a male of 50 years old

Owner	Minimal Deposit ω						No withdrawal or lapsing
	\$90	\$80	\$60	\$40	\$20	\$10	
Male	0.0090	0.0092	0.0097	0.0105	0.0122	0.0136	0.0077
Female	0.0067	0.0069	0.0073	0.0081	0.0095	0.0107	0.0053

TABLE 5.4: *Fair insurance charge (ρ_{ins}) for contracts containing a GMDB clause with annual ratchet events as a function of the minimal deposit amount (ω). Contract owners are assumed to be 50 years old at the time of purchase. The parameters in Table 5.1 are used in the pricing process.*

when the contract is purchased and that $\omega = \$80$. Other contract parameters are set to the values presented in Table 5.1. Results for the highest refinement level in Table 5.3 are accurate to within 2×10^{-5}.

We now examine how the minimum deposit amount (ω) affects the fair insurance charge ρ_{ins} obtained when solving equation (5.72). Table 5.4 presents the fair insurance charge for the GMDB clause with annual ratchet events when the minimum deposit ω ranges from \$10 to \$90. For comparison purposes, we also include the fair insurance charge for the GMDB clause when no withdrawals or contract lapsing are allowed. The results for both male and female owners are presented in Table 5.4. Other parameter values are specified in Table 5.1. In observing the results contained in Table 5.4, we see that the minimum deposit amount ω significantly impacts the fair insurance charge for the GMDB clause. Intuitively, as ω decreases, larger withdrawals can occur which is more detrimental to the issuing company and, as such, results in a higher insurance charge. The results in Table 5.4 show that the withdrawal feature is very valuable.

Table 5.4 also demonstrates the impact of the gender of the contract owner on the required insurance charge. Since female owners generally live longer than their male counterparts, a lower insurance fee is required. As shown in Table 5.4, this can be observed for different values of ω, as well as when the GMDB does not allow withdrawals or lapsing.

In [72], the authors state that certain contracts with a GMDB clause include longer

	Ratchet Interval				
Owner	0.5 year	1 year	2 years	5 years	10 years
Male	0.0137	0.0122	0.0105	0.0080	0.0059
Female	0.0107	0.0095	0.0081	0.0062	0.0046

TABLE 5.5: *Fair insurance charge (ρ_{ins}) for a GMDB guarantee with different ratchet intervals ranging from* 0.5 *to* 10 *years. The owner is assumed to be* 50 *years old at the time of purchase and* $\omega = \$20$. *Other contract parameters used when solving equation* (5.72) *are presented in Table 5.1.*

time intervals between ratchet dates such as 2 or 5 years. As such, numerical results for pricing GMDB contracts with $\omega = \$20$ for different ratchet intervals ranging from 6 months to 10 years are presented in Table 5.5. Note that the parameter values presented in Table 5.1 are used and that the owner is assumed to be 50 years of age when the contract is purchased. The results of Table 5.5 demonstrate that a lower insurance charge is imposed by the issuer as the ratchet interval is increased. With fewer ratchet events during the contract lifetime, the death benefit exposure of the issuing company is generally reduced resulting in a lower insurance fee. This relation is observed for both male and female owners. Clearly, modifying the ratchet interval also significantly impacts the fair insurance charge associated with the GMDB clause. It would appear that both the withdrawal and ratchet features are very valuable when included in a GMDB contract.

5.5.2 Numerical Results with Regime Switching

We now consider results from numerical experiments where regime-switching is added to the pricing model, as described in Section 5.2. In accordance with the calibration carried out in [5], we assume that there are three economic regimes which we denote as e_1, e_2 and e_3. In [5], the authors assume that the underlying is in one of three regimes of Brownian volatility and calibrate this model to an existing volatility smile. Therefore,

State Information - e_1	
σ_1 - Volatility	0.0955
Jump sizes:	$J^{1\to 2} = 0.9095$; $J^{1\to 3} = 1.0279$
Jump intensities:	$\lambda^{1\to 2} = 0.2405$; $\lambda^{1\to 3} = 3.3208$
State Information - e_2	
σ_2 - Volatility	0.0644
Jump sizes:	$J^{2\to 1} = 1.2502$; $J^{2\to 3} = 1.6512$
Jump intensities:	$\lambda^{2\to 1} = 1.1279$; $\lambda^{2\to 3} = 0.0729$
State Information - e_3	
σ_3 - Volatility	0.0241
Jump sizes:	$J^{3\to 1} = 0.9693$; $J^{3\to 2} = 0.7732$
Jump intensities:	$\lambda^{3\to 1} = 2.9882$; $\lambda^{3\to 2} = 0.2025$
Contract Information	
r - Interest rate	0.06
ρ_{man} - Management fees	0.015
Ratchet interval	1 year
Last Ratchet Date	30 years
T - Contract maturity	40 years
Grid Construction	
$\mathcal{D}_0$ - Initial deposit	\$100
$S_{\max}$ - Grid parameter	$\$3.6 \times 10^7$
$B_{\max}$ - Grid parameter	\$60000

TABLE 5.6: *Parameter values used when pricing GMDB contracts with regime-switching. Jump sizes and intensities taken from [5].*

we will determine the fair insurance charge ρ_{ins} that satisfies:

$$V(\rho_{ins}; S = \$100, B = \$100, D = \$100, E = e_1, \tau = 40 \text{ years}) = 0. \tag{5.74}$$

The data for all three states, e_1, e_2 and e_3, is presented in Table 5.6 and is taken from [5]. Table 5.6 also includes additional information about contract parameters and details on the grid construction used when solving equation (5.74) for different values of ω. We have verified that our choice for $B_{\max}$, and consequently $S_{\max}$, still provides a minimum of 5 digits of accuracy in the numerical results obtained. We choose to set the small fixed cost to $c = 1 \times 10^{-10}$ to ensure accuracy of at least 6 digits in the numerical results obtained.

Fair Insurance Fee for GMDB Guarantee with Regime-Switching				
Refinement Level	Nodes			Insurance Fee (ρ_{ins})
	S	B	D	
0	119	36	11	0.006286
1	237	71	21	0.006085
2	473	141	41	0.005931

TABLE 5.7: *Fair insurance fee (ρ_{ins}) for a GMDB guarantee with regime-switching for different grid refinement levels. The owner is assumed to be a male of* 50 *years old at the time of purchase and* $\omega = \$80$. *Fully implicit timestepping is used and the initial timestep is* $\Delta\tau = 0.05$ *years on the coarsest grid. Other contract parameters are presented in Table 5.1. Nodes - B indicates the maximum number of nodes in the B direction (i.e. when* $D = 0$*).*

Table 5.7 holds the fair insurance fee for a GMDB guarantee with regime-switching assuming $\omega = \$80$ for different grid refinement levels. We further assume that the contract owner is a male of 50 years of age when the contract is purchased. Additional contract parameters are presented in Table 5.6 and constant timesteps are used with fully implicit timestepping. The initial timestep is $\Delta\tau = 0.05$ years on the coarsest grid. Due to the high dimensionality of the pricing problem considered, the coarsest grid in the D direction is limited to 11 nodes and results from only 2 refinement levels were obtained. We estimate that the results are correct to within 2×10^{-4} when using a grid refinement of 2. While results from higher refinement levels would be necessary to establish a more definite convergence analysis, problem size and running time would be unmanageable. We remind the reader that the regime switching HJB problem is four dimensional. Note that typically, one obtains convergence estimates for nonlinear HJB equations which are of the form $O(h^\rho)$ where h is the discretization parameter. Estimates of ρ vary from 1/27 to 1/2 depending on assumptions about regularity of the solution and the PDE coefficients. See [10] for an overview of recent work along these lines. Nonetheless, the level of accuracy attained in Table 5.7 is deemed sufficient for our purposes.

Table 5.8 holds the fair insurance charge associated with the GMDB guarantee as a

function of ω assuming the economy is in state e_1. Based on previous comments, the results in Table 5.8 are obtained with a grid refinement level 2. Note that the owner is once again assumed to be 50 years of age when the contract is purchased. Other contract parameters used during the pricing process are presented in Table 5.6. For comparison purposes, the fair insurance charge associated with the GMDB guarantee when no withdrawal or lapsing is allowed is included in the last column of Table 5.8. As noted previously in Section 5.5.1, decreasing the minimum deposit amount ω increases the insurance fee charged by the issuing company. For example, setting $\omega = \$10$ when the contract owner is a man, requires a fee close to twice as large as that charged when no partial withdrawals are allowed. Notice that this remark applies equally to both male and female contract owners. In addition, the gender of the contract owner still affects the fair insurance charge for a given value of ω. Assuming $\omega = \$40$, the fair insurance charge for the GMDB guarantee when owned by a woman is about 25% less than what is charged for a male contract owner. Thus, even when more realistic assumptions are made regarding the state of the economy, we see that both the gender of the contract owner and the value of ω have a significant impact on the fair insurance fee for the GMDB guarantee.

The results presented in Table 5.8 are significantly different from those included in previous work on the topic such as [72]. In [72], the authors consider a GMDB contract with continuous ratchet events, no partial withdrawals and a shorter maturity period, resulting in much lower insurance fees than those presented in Table 5.8. Thus, Table 5.8 clearly demonstrates that higher fees are required for GMDB contracts with a partial withdrawal feature in a more realistic regime-switching context.

	Minimal Deposit ω						No withdrawal
Owner	\$90	\$80	\$60	\$40	\$20	\$10	or lapsing
Male	0.0058	0.0059	0.0063	0.0070	0.0082	0.0091	0.0049
Female	0.0044	0.0045	0.0049	0.0054	0.0065	0.0073	0.0034

TABLE 5.8: *Fair insurance charge (ρ_{ins}) for contracts containing a GMDB clause with annual ratchet events as a function of the minimal deposit amount (ω) assuming the economy is in regime e_1. Contract owners are assumed to be 50 years old at the time of purchase. The parameters in Table 5.6 are used in the pricing process.*

5.6 Summary

Increasingly popular in both the United States and the United Kingdom, variable annuity contracts include many different features. Focusing on contracts with a guaranteed minimum death benefit (GMDB) clause, we characterize the pricing problem as an impulse control problem. A pricing model based on partial differential equations was developed to determine the *fair or no-arbitrage insurance charge* for contracts with a GMDB clause. Regime-switching is also included in the pricing model due to the longer maturity of the contract considered. A numerical scheme was given which was shown to converge to the viscosity solution away from the ratchet dates. Based on results from numerical experiments, we have also shown that a much higher insurance charge is required when partial withdrawals are added to the GMDB guarantee. Previous work in the area [72] which ignores the possibility of partial withdrawals results in lower insurance fees.

While we have focused on pricing the GMDB guarantee with periodical ratchet dates and dollar-for-dollar withdrawals, alternate modelling assumptions are possible. For example, some contracts may include a proportional withdrawal policy, which is sure to impact the fair insurance fee. The most costly aspect of the computation of the guarantee involves the linear search for finding the optimal withdrawal. Further work will focus on techniques for speeding up this computation. We expect that parallelization will be extremely helpful here. Finally, while we have shown that our procedure converges to the

viscosity solution, we are not able to determine the rate of convergence. More research is needed on this topic.

Chapter 6

Conclusion

The impulse control framework is considered by some as a rather general approach which allows us to handle complex contractual features. In this thesis, two long-term option-type contracts were characterized as impulse control problems, namely: a complex employee stock option referred to as an infinite reload option and a variable annuity containing a guaranteed minimum death benefit (GMDB) clause. Implicit numerical schemes based on either a penalty method [44] or an operator splitting scheme [55] were developed and tailored to both contracts. From a theoretical perspective, convergence to the viscosity solution [31] was verified for both contracts. Assuming a strong comparison result applies, the numerical solution is guaranteed to converge to the unique viscosity solution if the discrete equations are stable, monotone and consistent [8].

While the main goal of this thesis was to show how impulse control problems can be valued with an implicit timestepping method, additional modelling details for each contract considered were also investigated. For infinite reload options, vesting periods were successfully added to the infinite reload pricing model in Chapter 3. Furthermore, Chapter 4 focused on determining the contract value to the holder by using a certainty equivalence approach. Similarly, when pricing variable annuities with a GMDB clause

in Chapter 5, the partial withdrawal feature was successfully integrate into the pricing model.

Hence, the main contributions of this thesis can be summarized as follows:

- The increased reload pricing problem was outlined and characterized as an impulse control problem [14], resulting in a Hamilton-Jacobi-Bellman variational inequality. In our formulation, the infinite reload pricing problem becomes a special case of the increased reload pricing problem where there is no increase in the reload strike. From a financial perspective, we demonstrated that making small changes to the reload contract can dramatically reduce the no-arbitrage contract value.

- A multi-step valuation model was developed to determine the value of employee stock options to the contract holder based on a certainty equivalence approach [64]. Both simple stock options and infinite reload options were valued from an employee's perspective. For infinite reload options, it was shown that the contract value to the employee is less than half of the no-arbitrage price and is highly sensitive to assumptions made regarding the employee's portfolio composition.

- A robust pricing model for the GMDB guarantee was developed. Cast as an impulse control problem, the pricing model includes the partial withdrawal feature and makes use of a regime-switching model [52]. Numerical results for the fair insurance charge were used in showing that typical fees being charged by insurance companies are not enough to fund hedging costs. This contrasts with previous results in the literature [72, 74] which did not take into account optimal withdrawals.

Thus, we have shown that the numerical solution to financial impulse control problems can be computed with an implicit solution method. While our focus was on option pricing problems, the numerical methods considered herein could be easily applied to other areas of finance where impulse control problems occur.

6.1 Future Work

Some directions for future research in this area include:

- Since employee stock options are known to be under-valued by their owners, there is room to negotiate new contract terms which would result in a reduced option expense for issuing companies. Alternate contract modifications to the one considered in this thesis are sure to reduce the no-arbitrage price of employee stock options. For example, a minimum holding period for new stock received following a reload event may reduce the value of an infinite reload option.

- For variable annuity pricing, the numerical method developed here could be used to determine the fair insurance fee for other complex contracts such as those containing a GMDB with interest guarantees [71] or those with guaranteed minimum income benefits (GMIB).

- Furthermore, it was observed that over half of the computation time when valuing the GMDB clause was spent determining the optimal withdrawal. While we used a linear search algorithm to accomplish this task, variable annuity pricing would benefit from improvements to speed up this computation.

- Since singular stochastic control problems can be viewed as a subset of impulse control problems (with infinitely small fixed cost, see Chapter 1), the implicit solution methods considered here may potentially be applied to many such problems occurring in finance.

Appendix A

Discretization

Similar discretization schemes are used when pricing infinite reload options in different contexts and when determining the fair insurance charge for the GMDB guarantee. Therefore, this appendix describes the discretization scheme in a high-level manner which encompasses all pricing problems presented in this thesis.

Let us assume that we are considering a derivative contract $V^m = V(S, K, E = e_m, \tau)$ whose value is a function of the underlying stock price S, the contract strike price K and the current economic regime denoted as $E = e_m$. We also assume that S follows a regime-switching process as described in equation (5.23). The value of the derivative contract can be obtained by solving the partial differential equation:

$$\begin{aligned} V_\tau^m =& \frac{(a(S,K,e_m,\tau))^2}{2} V_{SS}^m + b(S,K,e_m,\tau) V_S^m - c(S,K,e_m,\tau) V^m \\ &+ d(S,K,e_m,\tau)\left(V^l - V^m\right) + \frac{1}{\epsilon}\max(\mathcal{A}V^m - V^m, 0), \end{aligned} \tag{A.1}$$

where $\mathcal{A}V^m = \mathcal{A}V(S, K, e_m, \tau)$ represents the American-type constraint or impulse operator and $V^l = V(J^{m\to l}S, K, e_l, \tau)$ represents the contract value in regime e_l following a jump in S of amplitude $J^{m\to l}$ when the regime changes from e_m to e_l (see Section (5.2)).

The partial differential equation in (A.1) can be approximated by replacing derivatives by finite difference approximations. Assuming $V^n_{i,j,m} = V(S^j_i, K_j, e_m, \tau^n)$ denotes the discrete option value at $(S^j_i, K_j, e_m, \tau^n)$, the discrete version of equation (A.1) can be written as:

$$\begin{aligned} V^{n+1}_{i,j,m}\Big[1 + (1-\theta)\Delta\tau\big(\alpha^{n+1}_{i,j,m} + \beta^{n+1}_{i,j,m} + c(S^j_i, K_j, e_m, \tau^{n+1}) + d(S^j_i, K_j, e_m, \tau^{n+1})\big)\Big] = \\ V^n_{i,j,m}\Big[1 - \theta\Delta\tau(\alpha^n_{i,j,m} + \beta^n_{i,j,m} + c(S^j_i, K_j, e_m, \tau^n) + d(S^j_i, K_j, e_m, \tau^n))\Big] \\ + (1-\theta)\Delta\tau\left(\beta^{n+1}_{i,j,m}V^{n+1}_{i+1,j,m} + \alpha^{n+1}_{i,j,m}V^{n+1}_{i-1,j,m} + d(S^j_i, K_j, e_m, \tau^{n+1})V^{n+1}_{i,j,l}\right) \\ + \theta\Delta\tau\left(\beta^n_{i,j,m}V^n_{i+1,j,m} + \alpha^n_{i,j,m}V^n_{i-1,j,m} + d(S^j_i, K_j, e_m, \tau^n)V^n_{i,j,l}\right) \\ + P(V^{n+1}_{i,j,m}, \mathcal{A}^h V^{n+1}_{i,j,m}), \end{aligned} \tag{A.2}$$

where $0 \leq \theta \leq 1$ indicates which timestepping method is used, $P(V^{n+1}_{i,j,m}, \mathcal{A}^h V^{n+1}_{i,j,m})$ is the discrete penalty term and the value of both $\alpha^n_{i,j,m}, \beta^n_{i,j,m}$ depends on the choice of discretization for the derivative terms in equation (A.1). For example, choosing the higher order central difference scheme leads to the following values of $\alpha^n_{i,j,m}$ and $\beta^n_{i,j,m}$:

$$\begin{aligned} \alpha^n_{i,j,m,\text{central}} &= \frac{(a(S^j_i, K_j, e_m, \tau^n))^2}{(S^j_i - S^j_{i-1})(S^j_{i+1} - S^j_{i-1})} - \frac{b(S^j_i, K_j, e_m, \tau^n)}{S^j_{i+1} - S^j_{i-1}}, \\ \beta^n_{i,j,\text{central}} &= \frac{(a(S^j_i, K_j, e_m, \tau^n))^2}{(S^j_{i+1} - S^j_i)(S^j_{i+1} - S^j_{i-1})} + \frac{b(S^j_i, K_j, e_m, \tau^n)}{S^j_{i+1} - S^j_{i-1}}. \end{aligned} \tag{A.3}$$

To ensure a positive coefficient discretization, assuming $c(S^j_i, K_j, e_m, \tau^n) \geq 0$ and $d(S^j_i, K_j, e_m, \tau^n) \geq 0$ for all n, it is preferable to choose other discretization techniques at

if $\alpha^n_{i,j,m,\text{central}} \geq 0$ and $\beta^n_{i,j,m,\text{central}} \geq 0$ **then**
  $\alpha^n_{i,j,m} = \alpha^n_{i,j,m,\text{central}}$
  $\beta^n_{i,j,m} = \beta^n_{i,j,m,\text{central}}$
else if $\beta^n_{i,j,m,\text{forward}} \geq 0$ **then**
  $\alpha^n_{i,j,m} = \alpha^n_{i,j,m,\text{forward}}$
  $\beta^n_{i,j,m} = \beta^n_{i,j,m,\text{forward}}$
else
  $\alpha^n_{i,j,m} = \alpha^n_{i,j,m,\text{backward}}$
  $\beta^n_{i,j,m} = \beta^n_{i,j,m,\text{backward}}$
end if

ALGORITHM A.1: *Coefficient Discretization*

the problem nodes such as forward or backward differences. Forward differences produces:

$$\alpha^n_{i,j,m,\text{forward}} = \frac{(a(S^j_i, K_j, e_m, \tau^n))^2}{(S^j_i - S^j_{i-1})(S^j_{i+1} - S^j_{i-1})},$$
$$\beta^n_{i,j,m,\text{forward}} = \frac{(a(S^j_i, K_j, e_m, \tau^n))^2}{(S^j_{i+1} - S^j_i)(S^j_{i+1} - S^j_{i-1})} + \frac{b(S^j_i, K_j, e_m, \tau^n)}{S^j_{i+1} - S^j_i}, \tag{A.4}$$

while backward differences delivers:

$$\alpha^n_{i,j,m,\text{backward}} = \frac{(a(S^j_i, K_j, e_m, \tau^n))^2}{(S^j_i - S^j_{i-1})(S^j_{i+1} - S^j_{i-1})} - \frac{b(S^j_i, K_j, e_m, \tau^n)}{S^j_{i+1} - S^j_i},$$
$$\beta^n_{i,j,m,\text{backward}} = \frac{(a(S^j_i, K_j, e_m, \tau^n))^2}{(S^j_{i+1} - S^j_i)(S^j_{i+1} - S^j_{i-1})}. \tag{A.5}$$

Algorithmically, the decision between a central, forward or backward discretization at each node is made based on the criteria presented as Algorithm A.1. The criteria in Algorithm A.1 guarantees that both $\alpha^n_{i,j,m}$ and $\beta^n_{i,j,m}$ are non-negative:

$$\alpha^n_{i,j,m} \geq 0 \; ; \; \beta^n_{i,j,m} \geq 0 \quad \text{for all } i,\, j,\, m,\, n. \tag{A.6}$$

Appendix B

Proof of Theorem 2.4

In this appendix, we prove the following Theorem.

Theorem 2.4 (Bound for $V_{i,j}^{n+1}$). *Assuming the numerical scheme satisfies Definition 2.1, that the boundary conditions are applied as outlined in Section 2.1.1 and that the initial conditions are given by the discrete version of equation* (2.18)*, the value of the increased reload contract satisfies:*

$$0 \leq V_{i,j}^{n} \leq S_i^j \quad \forall i, j, n, \tag{B.1}$$

for fully implicit timestepping ($\theta = 0$).

Before proving Theorem 2.4, we prove some utility lemmas. We define the vector $\mathcal{Z}^{n+1}$ as:

$$\mathcal{Z}^{n+1} = \begin{bmatrix} \mathcal{Z}_{0,0}^{n+1} \\ \mathcal{Z}_{1,0}^{n+1} \\ \vdots \\ \mathcal{Z}_{j_{\max}-1,j_{\max}}^{n+1} \\ \mathcal{Z}_{j_{\max},j_{\max}}^{n+1} \end{bmatrix}, \tag{B.2}$$

and denote the the μ^{th} entry in $\mathcal{Z}^{n+1}$ as $[\mathcal{Z}^{n+1}]_{i,j}$ where $\mu = (i+1) + j(j_{\max}+1)$.

Let $\mathcal{Q}^{n+1}(V^{n+1})$ be the matrix of coefficients involving elements of V^{n+1} in discretization (2.23). The matrix $\mathcal{Q}^{n+1}(V^{n+1})$ is defined as:

$$\begin{aligned}\left[\mathcal{Q}^{n+1}(V^{n+1})\mathcal{Z}^{n+1}\right]_{i,j} = \mathcal{Z}^{n+1}_{i,j}(1+\Delta\tau(\alpha^{n+1}_{i,j}+\beta^{n+1}_{i,j}+r+L^{n+1}_{i,j})) - \Delta\tau\alpha^{n+1}_{i,j}\mathcal{Z}^{n+1}_{i-1,j} \\ - \Delta\tau\beta^{n+1}_{i,j}\mathcal{Z}^{n+1}_{i+1,j} - \Delta\tau L^{n+1}_{i,j}\frac{K_j}{S^j_i}\left((1-\omega)\mathcal{Z}^{n+1}_{l,m} - \omega\mathcal{Z}^{n+1}_{l,m+1}\right), \quad \text{(B.3)}\end{aligned}$$

when $i < j_{\max}$, and we ignore for the moment any nodes where a similarity boundary condition is applied (see equation (2.16)), and when $i = j_{\max}$:

$$\left[\mathcal{Q}^{n+1}(V^{n+1})\mathcal{Z}^{n+1}\right]_{j_{\max},j} = \mathcal{Z}^{n+1}_{j_{\max},j}(1+L^{n+1}_{j_{\max},j}) - L^{n+1}_{j_{\max},j}\min\left(1, \frac{K_j(1+p)}{K_{j_{\max}}}\right)\mathcal{Z}^{n+1}_{l,j_{\max}}. \quad \text{(B.4)}$$

Remark B.1. *For ease of exposition, we will not consider the case for interior nodes where the similarity reduction boundary condition is applied. The following argument is straightforward to apply to those nodes where* $S^j_i(1+p) > K_{j_{\max}}$ *and is omitted. Note that* $\frac{K_j}{S^j_i} \leq 1$ *and of course* $\min\left(1, \frac{K_j(1+p)}{K_{j_{\max}}}\right) \leq 1$.

Here, ω is defined in equation (2.31), $L^{n+1}_{i,j}$ is defined in equation (2.27) and $\alpha^n_{i,j}$ and $\beta^n_{i,j}$ are determined according to the algorithm in Appendix A. Notice that $\mathcal{Q}^{n+1}(V^{n+1})$ is a function of the solution vector V^{n+1} since the value of $L^{n+1}_{i,j}$ depends on the solution.

It is useful to note the following property of the matrix $\mathcal{Q}^{n+1}(V^{n+1})$.

Lemma B.2 (M-matrix). *The matrix* $\mathcal{Q}^{n+1}(V^{n+1})$ *defined in equations* (B.3) *and* (B.4) *is an M-matrix.*

Proof. For any V^{n+1}, the diagonal entries in $\mathcal{Q}^{n+1}(V^{n+1})$ are positive while the off-diagonal entries are negative or equal to zero. In addition, the row sum of the entries in $\mathcal{Q}^{n+1}(V^{n+1})$ is strictly positive for all rows. Thus, $\mathcal{Q}^{n+1}(V^{n+1})$ is an M-matrix. □

Remark B.3 (Positive Inverse). *Note that an M-matrix has the important property that it is invertible with a positive inverse. In particular, for any vector V, $\mathcal{Q}^{n+1}(V^{n+1})V \geq 0$ implies that $V \geq 0$.*

Lemma B.4. *The following are true.*

(a) Let $[\mathcal{Z}^{n+1}]_{i,j} = V_{i,j}^{n+1}$ where $V_{i,j}^{n+1}$ is the solution to equation (2.23) and define the vector $\hat{\mathcal{Z}}^{n+1}$ as:

$$[\hat{\mathcal{Z}}^{n+1}]_{i,j} = \begin{cases} V_{i,j}^{n+1} & \text{if } i < j_{\max}, \\ \text{Payoff}(S_{j_{\max}}^{j}, K_j) & \text{if } i = j_{\max}, \end{cases} \tag{B.5}$$

where $\text{Payoff}(S_i^j, K_j)$ is defined in equation (2.2). Then:

$$\mathcal{Q}^{n+1}(V^{n+1})\mathcal{Z}^{n+1} = \hat{\mathcal{Z}}^{n} + F^{n+1},$$

where

$$[F^{n+1}]_{i,j} = \begin{cases} \Delta\tau L_{i,j}^{n+1}(S_i^j - K_j) & \text{if } i < j_{\max}, \\ L_{j_{\max},j}^{n+1}(S_{j_{\max}}^{j} - K_j) & \text{if } i = j_{\max}. \end{cases} \tag{B.6}$$

(b) Let $[\mathcal{Z}^{n+1}]_{i,j} = S_i^j$. Then:

$$\mathcal{Q}^{n+1}(V^{n+1})\mathcal{Z}^{n+1} = \mathcal{Z}^{n} + F^{n+1},$$

where

$$[F^{n+1}]_{i,j} = \begin{cases} \Delta\tau(qS_i^j + L_{i,j}^{n+1}(S_i^j - K_j)) & \text{if } i < j_{\max}, \\ L_{j_{\max},j}^{n+1}(S_{j_{\max}}^{j} - K_j) & \text{if } i = j_{\max}. \end{cases} \tag{B.7}$$

Here, recall that q is the dividend yield.

Proof. The first identity follows directly from the definition of $\mathcal{Q}^{n+1}(V^{n+1})$ in equations (B.3) and (B.4) and the discretization in equation (2.23). The second identity follows from looking at the i, j components of the matrix $\mathcal{Q}^{n+1}(V^{n+1})$. □

We now present the proof of Theorem 2.4.

Proof. (of Theorem 2.4)

Let $[\mathcal{Z}^n]_{i,j} = V^n_{i,j}$ for all i, j and define the vector $\hat{\mathcal{Z}}^{n+1}$ as in equation (B.5). We will show using induction that $\mathcal{Z}^n > 0$ for all n.

Since the option value is initially set to the payoff (defined in equation (2.2)), notice that $\mathcal{Z}^0 \geq 0$ and, similarly, $\hat{\mathcal{Z}}^0 \geq 0$. Assume now that $n > 0$ and that $\mathcal{Z}^n \geq 0$. These assumptions imply that $\hat{\mathcal{Z}}^n \geq 0$.

Then, from Lemma B.2, we have:

$$\mathcal{Q}^{n+1}(V^{n+1})\mathcal{Z}^{n+1} = \hat{\mathcal{Z}}^n + F^{n+1}, \tag{B.8}$$

where F^{n+1} is defined in equation (B.6). From the definition of $L^{n+1}_{i,j}$ in equation (2.27), we see that:

$$L^{n+1}_{i,j}(S^j_i - K_j) \geq 0 \quad \forall i, j. \tag{B.9}$$

Hence, equation (B.9) implies $F^{n+1} \geq 0$.

Since $\hat{\mathcal{Z}}^n \geq 0$, we see that $\mathcal{Q}^{n+1}(V^{n+1})\mathcal{Z}^{n+1} \geq 0$. Since $\mathcal{Q}^{n+1}(V^{n+1})$ is an M-matrix, we make use of Remark B.3 and deduce $\mathcal{Z}^{n+1} \geq 0$. Thus, by induction, we have $\mathcal{Z}^n \geq 0$ for all n, proving the first inequality of Theorem 2.4.

We now set $[\mathcal{Z}^{n+1}]_{i,j} = S^j_i - V^{n+1}_{i.j}$ for all i, j and define $\hat{\mathcal{Z}}^{n+1}$ as:

$$[\hat{\mathcal{Z}}^{n+1}]_{i,j} = \begin{cases} S^j_i - V^{n+1}_{i,j} & \text{if } i < j_{\max}, \\ S^j_{j_{\max}} - \text{Payoff}(S^j_{j_{\max}}, K_j) & \text{if } i = j_{\max}. \end{cases} \tag{B.10}$$

We will once again use induction to show that $\mathcal{Z}^n \geq 0$ for all n.

Since $V_{i,j}^0 = \max(S_i^j - K_j, 0)$, both $\mathcal{Z}^0 \geq 0$ and $\hat{\mathcal{Z}}^0 \geq 0$. We now assume that $n > 0$ and that $\mathcal{Z}^n \geq 0$. Since $S_{j_{\max}}^j \geq \text{Payoff}(S_{j_{\max}}^j, K_j)$ and $\mathcal{Z}^n \geq 0$, we deduce from equation (B.10) that $\hat{\mathcal{Z}}^n \geq 0$.

From Lemma B.4, we have:

$$\mathcal{Q}^{n+1}(V^{n+1})\mathcal{Z}^{n+1} = \hat{\mathcal{Z}}^n + F^{n+1}, \tag{B.11}$$

where

$$[F^{n+1}]_{i,j} = \begin{cases} \Delta\tau q S_i^j & \text{if } i < j_{\max}, \\ 0 & \text{if } i = j_{\max}. \end{cases} \tag{B.12}$$

Since the dividend yield $q \geq 0$, we have $F^{n+1} \geq 0$ which implies $\mathcal{Q}^{n+1}(V^{n+1})\mathcal{Z}^{n+1} \geq 0$. Recalling that $\mathcal{Q}^{n+1}(V^{n+1})$ is an M-matrix and making use of Remark B.3, we conclude that $\mathcal{Z}^{n+1} \geq 0$. Hence, by induction, we have shown that $\mathcal{Z}^n \geq 0$ for all n, which implies $V_{i,j}^n \leq S_i^j$ for all i, j, n.

Thus, we have shown that $V_{i,j}^n$ is bounded:

$$0 \leq V_{i,j}^n \leq S_i^j \text{ for all } i, j, n. \tag{B.13}$$

□

Appendix C

Proof of Theorem 2.10

Before outlining the proof for Theorem 2.10, we prove a utility lemma. Note once again that in the following we will not consider the consistency for interior nodes with a similarity boundary condition (see equation (2.16)). The following argument can be applied to these nodes in a straightforward manner but we omit this for brevity.

Lemma C.1. *For any smooth test function ϕ with bounded derivatives of all orders with respect to S and τ, with $x = (S_i^j, K_j, \tau^{n+1})$, we have:*

$$\hat{g}_{i,j}(h, x, \phi_{i,j}^{n+1} + \xi, \{\phi_{k,j}^{n+1} + \xi\}_{k \neq i}, \{\phi_{i,j}^{n} + \xi\}, \phi_{l,m}^{n+1} + \xi, \phi_{l,m+1}^{n+1} + \xi) - F(\phi)_{i,j}^{n+1} = O(h) + \xi b(x), \tag{C.1}$$

where $b(x)$ is a bounded function of x with

$$|b(x)| = \begin{cases} \max\left(r, 1 - \frac{K_j}{S_i^j}\right) & \text{when } i < j_{\max}, \\ \max\left(1, 1 - \min\left(1, \frac{K_j(1+p)}{K_{\max}}\right)\right) & \text{when } i = j_{\max}. \end{cases}$$

Proof. To prove Lemma C.1, we determine the truncation error of the differential operator

$\mathcal{L}$ and the penalty term, assuming fully implicit timestepping is used.

We denote the continuous operator $\mathcal{L}$ at node (S_i^j, K_j, τ^n) as:

$$(\mathcal{L}\phi)_{i,j}^{n+1}, \tag{C.2}$$

while the discrete version of the operator is denoted by:

$$[\mathcal{L}^h\phi]_{i,j}^{n+1}. \tag{C.3}$$

Using Taylor series expansion, we find:

$$[\mathcal{L}^h(\phi+\xi)]_{i,j}^{n+1} - (\mathcal{L}\phi)_{i,j}^{n+1} = \mathcal{O}(\Delta S_{\max}^j) - r\xi. \tag{C.4}$$

Similarly, we assume that:

$$[\mathcal{A}\phi]_{i,j}^{n+1} \tag{C.5}$$

denotes the continuous reload constraint evaluated at node (S_i^j, K_j, τ^{n+1}), while the discrete version of the constraint is denoted by:

$$\mathcal{A}^h\phi_{i,j}^{n+1}. \tag{C.6}$$

The discrete reload constraint is generally obtained using diagonal interpolation (see equations (2.30) and (2.34)), which results in an error of the form:

$$\mathcal{A}^h(\phi_{i,j}^{n+1}+\xi) - [\mathcal{A}\phi]_{i,j}^{n+1} = \mathcal{O}((\Delta K_{\max})^2) + \frac{K_j}{S_i^j}\xi. \tag{C.7}$$

Using the formulation in equation (2.50) and the discretization error estimates in

equations (C.4) and (C.7), we find that for interior nodes ($i < j_{\max}$):

$$\begin{aligned}
&\left|\hat{g}_{i,j}(h, x\phi_{i,j}^{n+1}+\xi, \{\phi_{k,j}^{n+1}+\xi\}_{k\neq i}, \{\phi_{i,j}^{n}+\xi\}, \phi_{l,m}^{n+1}+\xi, \phi_{l,m+1}^{n+1}+\xi) - F_{in}(\phi)_{i,j}^{n+1}\right| \qquad \text{(C.8)}\\
&\quad \leq \max\Bigg(\left|\epsilon\left(\frac{\phi_{i,j}^{n+1}-\phi_{i,j}^{n}}{\Delta\tau} - [\mathcal{L}^h(\phi+\xi)]_{i,j}^{n+1}\right) + \phi_{i,j}^{n+1}+\xi - \mathcal{A}^h(\phi_{i,j}^{n+1}+\xi) - [\phi-\mathcal{A}\phi]_{i,j}^{n+1}\right|,\\
&\qquad\quad \left|\frac{\phi_{i,j}^{n+1}-\phi_{i,j}^{n}}{\Delta\tau} - [\mathcal{L}^h(\phi+\xi)]_{i,j}^{n+1} - [\phi_\tau - \mathcal{L}\phi]_{i,j}^{n+1}\right|\Bigg)\\
&\quad = \max\Bigg(\left|\epsilon\left(\frac{\phi_{i,j}^{n+1}-\phi_{i,j}^{n}}{\Delta\tau} - [\mathcal{L}^h(\phi)]_{i,j}^{n+1} + r\xi\right) + \mathcal{O}((\Delta K_{\max})^2) + \xi\left(1-\frac{K_j}{S_i^j}\right)\right|,\\
&\qquad\quad \left|\mathcal{O}(\Delta S_{\max}^j) + \mathcal{O}(\Delta\tau) + r\xi\right|\Bigg).
\end{aligned}$$

assuming fully implicit timestepping is chosen [1]. Similarly, for the boundary nodes ($i = j_{\max}$), we have:

$$\begin{aligned}
&\left|\hat{g}_{j_{\max},j}(h, x, \phi_{j_{\max},j}^{n+1}+\xi, \{\phi_{k,j}^{n+1}+\xi\}_{k\neq j_{\max}}, \{\phi_{j_{\max},j}^{n}+\xi\}, \phi_{l,m}^{n+1}+\xi, \phi_{l,m+1}^{n+1}+\xi) - F_{bound}(\phi)_{j_{\max},j}^{n+1}\right|\\
&\leq \max\Bigg(\left|\epsilon\left(\phi_{j_{\max},j}^{n+1}+\xi - \text{Payoff}(S_{j_{\max}}^j, K_j)\right) + \phi_{j_{\max},j}^{n+1}+\xi - \mathcal{A}^h(\phi_{j_{\max},j}^{n+1}+\xi) - [\phi-\mathcal{A}\phi]_{j_{\max},j}^{n+1}\right|,\\
&\qquad \left|\phi_{j_{\max},j}^{n+1}+\xi - \text{Payoff}(S_{j_{\max}}^j, K_j) - [\phi - \text{Payoff}(S,K)]_{j_{\max},j}^{n+1}\right|\Bigg)\\
&= \max\Bigg(\left|\epsilon\left(\phi_{j_{\max},j}^{n+1}+\xi - \text{Payoff}(S_{j_{\max}}^j, K_j)\right) + \mathcal{O}((\Delta K_{\max})^2)\right.\\
&\qquad \left. + \xi\left(1 - \min\left(1, \frac{K_j(1+p)}{K_{\max}}\right)\right)\right|, \left|\xi\right|\Bigg). \qquad \text{(C.9)}
\end{aligned}$$

Hence, recalling Assumption 2.3, we have shown that equation (C.1) holds. □

We now provide the proof of Theorem 2.10.

Proof. (of Theorem 2.10)

[1] When no constraint is active and central weighting is active at node (i, j), then the truncation error is locally second order in $\Delta\tau$, $\Delta S_{\max}^j$ when Crank-Nicolson timestepping is used.

We begin by proving that equation (2.53) holds. From the definition of lim sup, there exists sequences $h_d, i_d, j_d, n_d, \xi_d$ where:

$$h_d \to 0,\ \xi_d \to 0,\ x_d = (S_{i_d}^{j_d}, K_{j_d}, \tau^{n_d+1}) \to \hat{x} = (\hat{S}, \hat{K}, \hat{\tau}) \ \text{ as } \ d \to \infty \tag{C.10}$$

and

$$\begin{aligned}\limsup_{d\to\infty} \hat{g}_{i_d,j_d}(h_d, x_d, \phi_{i_d,j_d}^{n_d+1} + \xi_d, \{\phi_{k,j_d}^{n_d+1} + \xi_d\}_{k\neq i_d}, \{\phi_{i_d,j_d}^{n_d} + \xi_d\}, \phi_{l,m}^{n_d+1} + \xi_d, \phi_{l,m+1}^{n_d+1} + \xi_d) \\ = \limsup_{\substack{\xi,\, h\to 0 \\ x\to\hat{x}}} \hat{g}_{i,j}(h, x, \phi_{i,j}^{n+1} + \xi, \{\phi_{k,j}^{n+1} + \xi\}_{k\neq i}, \{\phi_{i,j}^{n} + \xi\}, \phi_{l,m}^{n+1} + \xi, \phi_{l,m+1}^{n+1} + \xi).\end{aligned} \tag{C.11}$$

From Lemma C.1, we have:

$$\begin{aligned}\hat{g}_{i_d,j_d}(h_d, \phi_{i_d,j_d}^{n_d+1} + \xi_d, \{\phi_{k,j_d}^{n_d+1} + \xi_d\}_{k\neq i_d}, \{\phi_{i_d,j_d}^{n_d} + \xi_d\}, \phi_{l,m}^{n_d+1} + \xi_d, \phi_{l,m+1}^{n_d+1} + \xi_d) \\ = F(\phi)_{i_d,j_d}^{n_d+1} + \xi_d b(x_d) + O(h_d).\end{aligned} \tag{C.12}$$

Now consider a sequence of nodes x_d as defined in equation (C.10) possibly containing both interior ($i < j_{\max}$) and boundary nodes ($i = j_{\max}$). Combining equation (C.11) with equation (C.12), we obtain:

$$\begin{aligned}\limsup_{\substack{\xi,\, h\to 0 \\ x\to\hat{x}}} \hat{g}_{i,j}(h, x, \phi_{i,j}^{n+1} + \xi, \{\phi_{k,j}^{n+1} + \xi\}_{k\neq i}, \{\phi_{i,j}^{n} + \xi\}, \phi_{l,m}^{n+1} + \xi, \phi_{l,m+1}^{n+1} + \xi) \\ \leq \limsup_{d\to\infty} F(\phi)_{i_d,j_d}^{n_d+1} + \lim_{d\to\infty} [O(h_d) + \xi_d b(x_d)] \leq F^*(\phi(\hat{x})),\end{aligned} \tag{C.13}$$

since

$$\lim_{d\to\infty} [O(h_d) + \xi_d b(x_d)] = 0. \tag{C.14}$$

The same technique can be used to verify equation (2.53) holds; this is omitted for brevity. □

Thus, we conclude that the numerical scheme in equation (2.23) satisfies Definition 2.8 and is consistent.

Appendix D

Reload Options with Capped Boundary Condition

In Section 2.1.1, one of the boundary conditions considered at $K = K_{\max}$ for the increased reload problem is to assume that the contract contains a cap. This implies that no reload is possible when $K \geq K_{\max}$. In this section, we present numerical results obtained when pricing increased reload options using this alternate boundary condition at $K = K_{\max}$.

When $K = K_{\max}$, the pricing problem is:

$$V_\tau - \mathcal{L}V = 0, \tag{D.1}$$

while the following is solved when $K < K_{\max}$:

$$\min(V_\tau - \mathcal{L}V, V - \mathcal{A}V) = 0, \tag{D.2}$$

where the reload constraint $\mathcal{AV} = \mathcal{AV}(S, K, t)$ is defined as:

$$\mathcal{AV}(S,K,t) = \begin{cases} (S-K) + \frac{K}{S} V(S, S^*(1+p), t) & \text{if } S > K, \\ 0 & \text{otherwise,} \end{cases} \tag{D.3}$$

where $S^* = \min\left(S, \frac{K_{\max}}{(1+p)}\right)$. When the capped boundary condition is used at $K = K_{\max}$, no similarity reduction is possible which implies that the reload pricing problem will be solved on a two-dimensional $S \times K$ domain. The underlying grid will still be built using the *scaled grid* technique described in Section 2.2.1.

In the context of the capped boundary condition, the choice of $K_{\max}$ is crucial in ensuring accurate numerical results. As such, we consider four different grids each with different values of $K_{\max}$ and determine the error introduced by this choice. Table D.1 presents values of $K_{\max}$ and $S_{\max}$ as well as the number of nodes (in both the S and K directions) for each grid considered. Table D.2 presents the value of an increased reload option contract when $p = 0\%$ on each of the four grids defined in Table D.1 for different refinement levels. The contract considered has a maturity of 10 years and an initial strike of $\overline{K} = \$100$. Other contract parameters used are presented in Table 2.1.

We see that the choice of $K_{\max}$ has a significant impact on the numerical results obtained and that even larger choices of $K_{\max}$ introduce some numerical error. Recall that the analytical value of an infinite reload contract is \$64.67 at $S = \$100$ [32]. While the accuracy level attained for a refinement of level 3 for grids C and D is comparable to the values presented in Section 2.5, numerical results from higher refinement levels are not presented due to the impracticality of solving the reload pricing problem on such a large grid. Note that grid B was used in Section 2.5 with a similarity boundary condition (see equation (2.16)). In that case, the effect of increasing $K_{\max}$ was only seen in the sixth digit. When the capped boundary condition is applied, a larger value of $K_{\max}$ is

	Nbr. Nodes	$K_{\max}$	$S_{\max}$
Grid A	60	1000	10000
Grid B	61	2000	40000
Grid C	62	5000	250000
Grid D	63	10000	1000000

TABLE D.1: *Characteristics of four different grid constructions.* Nbr. Nodes *refers to the initial number of nodes in each of the S and K directions. Also, we have $K_{\max} = K_{j_{\max}}$ and $S_{\max} = S^j_{j_{\max}}$ for any j.*

Refinement	Nodes	Timesteps	Option Value
Grid A			
0	60	105	63.812320
1	119	232	64.362905
2	237	515	64.565822
3	472	1107	64.633061
Grid B			
0	61	108	64.214477
1	121	237	64.533236
2	241	518	64.628285
3	481	1108	64.655974
Grid C			
0	62	105	64.418463
1	123	232	64.595394
2	245	515	64.645230
3	489	1107	64.660752
Grid D			
0	63	105	64.498454
1	123	235	64.613959
2	245	515	64.648246
3	489	1107	64.661248

TABLE D.2: *Value of an increased reload option with $p = 0\%$ at $S = \$100$ with a capped boundary condition for different underlying grids defined in Table D.1. Note that Crank-Nicolson timestepping was used with variable timesteps where $\Delta\tau_0 = 0.1$ years on the coarsest grid. The other parameter values chosen are specified in Table 2.1.*

required to limit the error introduced by the modified reload constraint in (D.3). Based on the results in Table D.2, choices such as $K_{\max} = \$5000$ or $K_{\max} = \$10000$ would appear appropriate in this case.

In summary, we have shown that the capped boundary condition remains an acceptable choice for infinite reload options since a reasonable level of accuracy can be obtained, assuming $K_{\max}$ is chosen sufficiently large. However, there are two significant drawbacks stemming from this alternate boundary condition which need to be taken into consideration. First, no similarity reduction is possible when a capped boundary condition is chosen. This implies that the full two-dimensional $S \times K$ grid will always be used when pricing increased reload options. Second, a higher value of $K_{\max}$ is required to limit the error introduced by the boundary condition, resulting in a larger underlying grid. Both of these observations imply higher computation times when the capped reload constraint is used. Hence, it would appear that the use of a similarity reduction at $K_{\max}$ is the better of the two boundary conditions considered.

Appendix E

Pricing American Options with Operator Splitting

In this section, we present numerical results obtained when pricing an American put option using the operator splitting method. More specifically, we are looking to determine the accuracy and convergence rate obtained when using the operator splitting method.

Denoting the option value as $V = V(S, K, \tau)$ where $\tau = T - t$, we solve the following pricing problem where V satisfies the Black-Scholes equation [85]:

$$\min\Big(V_\tau - \mathcal{L}V, V - \mathcal{A}V\Big) = 0, \tag{E.1}$$

and $\mathcal{L}V$ is defined in equation (2.4). In this case, the American constraint $\mathcal{A}V = \mathcal{A}V(S, K, \tau)$ is set to the payoff:

$$\mathcal{A}V(S, K, \tau) = \text{Payoff}(S, K) = \max(K - S, 0), \tag{E.2}$$

with $V(S, K, \tau = 0) = \text{Payoff}(S, K)$.

To get a fair appraisal of the operator splitting method, the test contract is assumed to have a maturity of 10 years. The other contract parameters are presented in Table 3.1. The numerical values obtained when using both fully implicit and Crank-Nicolson timestepping for different refinement levels are presented in Table E.1. Recall that each refinement operation nearly doubles the number of grid nodes and cuts the timestep size in half. Note that Table E.1 includes the values for Crank-Nicolson timestepping when both constant and variable timesteps are used. See [44] for details on the timestep selector used and an explanation of the importance of variable timestepping for American-type constraints. Also, since the initial payoff is non-smooth, a few fully implicit timesteps are taken at the start of the solution process when Crank-Nicolson timestepping is used, as suggested in [82]. For the values in Table E.1, four such timesteps were taken. While past research suggests that two Rannacher steps be taken, this was not sufficient to ensure solution smoothness. It was found that four initial Rannacher steps were required to eliminate oscillations from the final solution. This is consistent with the results in [47].

In observing the results in Table E.1, we first note that linear convergence is obtained when the operator splitting method is used with fully implicit timestepping. This result is consistent with local truncation error analysis. However, when Crank-Nicolson timestepping is used with constant or variable timesteps, the convergence ratios obtained do not indicate quadratic convergence. When variable timesteps are used, we see that the convergence ratios obtained are somewhat erratic and do not demonstrate consistent quadratic convergence. When constant timesteps are taken, the convergence ratios are more stable but tend towards linear rather than quadratic convergence. Thus, it is highly unlikely that the operator splitting method will provide quadratic convergence when pricing more complex contracts such as infinite reload options with Crank-Nicolson timestepping.

For comparison purposes, Table E.2 presents numerical values obtained when pricing

American Put Option - Operator Splitting				
Refinement	Nodes	Option Value	Difference	Ratio
Fully Implicit				
0	61	19.941103	n.a.	n.a.
1	121	20.041845	0.100741	n.a.
2	241	20.079060	0.037215	2.71
3	481	20.092135	0.013075	2.85
4	961	20.096336	0.004201	3.11
5	1921	20.098177	0.001841	2.28
6	3841	20.099011	0.000834	2.21
Crank-Nicolson (variable timesteps)				
0	61	19.959293	n.a.	n.a.
1	121	20.027473	0.068179	n.a.
2	241	20.062075	0.034602	1.97
3	481	20.070472	0.008398	4.12
4	961	20.082887	0.012414	0.68
5	1921	20.091421	0.008534	1.45
6	3841	20.095639	0.004218	2.02
Crank-Nicolson (constant timesteps)				
0	61	19.966965	n.a.	n.a.
1	121	20.063030	0.096065	n.a.
2	241	20.088208	0.025178	3.82
3	481	20.096883	0.008675	2.90
4	961	20.098706	0.001823	4.76
5	1921	20.099381	0.000675	2.70
6	3841	20.099618	0.000237	2.85

TABLE E.1: *Value of an American put option at* $S = \$100$ *using fully implicit and Crank-Nicolson timestepping, with both constant and variable timesteps, for different refinement levels. The initial timestep is set to* 0.05 *years on the coarsest grid. The* Nodes *column indicates the number of nodes in the underlying* S*-grid. Other parameter values are presented in Table 3.1.*

the same American contract using the penalty scheme [44]. Results for fully implicit timestepping as well as Crank-Nicolson timestepping, using variable timesteps, are presented. Once again, the parameters in Table 3.1 are used and four Rannacher timesteps are taken when Crank-Nicolson is chosen. We first note that the sequence of convergence ratios obtained when fully implicit timestepping is used tends to 2, which implies linear convergence. Similarly, when Crank-Nicolson timestepping is used, the convergence ra-

American Put Option - Penalty Method				
Refinement	Nodes	Option Value	Difference	Ratio
Fully Implicit				
0	61	19.949307	n.a.	n.a.
1	121	20.061658	0.112351	n.a.
2	241	20.086357	0.024699	4.55
3	481	20.095559	0.009202	2.68
4	961	20.098164	0.002605	3.53
5	1921	20.099094	0.000930	2.80
6	3841	20.099470	0.000376	2.48
Crank-Nicolson (variable timesteps)				
0	61	19.965757	n.a.	n.a.
1	121	20.070863	0.105105	n.a.
2	241	20.091085	0.020222	5.20
3	481	20.097917	0.006832	2.96
4	961	20.099333	0.001416	4.83
5	1921	20.099676	0.000344	4.12
6	3841	20.099762	8.524000e-05	4.03

TABLE E.2: *Value of an American put option at* $S = \$100$ *using fully implicit and Crank-Nicolson timestepping for different refinement levels when the penalty method is used. The initial timestep is set to* 0.05 *years on the coarsest grid. The* Nodes *column indicates the number of nodes in the underlying* S*-grid. Other parameter values are presented in Table 3.1.*

tios in Table E.2 imply quadratic convergence. Thus, the penalty method would appear to provide better convergence than the operator splitting method when Crank-Nicolson timestepping is used.

As a side note, the results in Table E.1 indicate that using constant timesteps in the context of the operator splitting method provides superior convergence for Crank-Nicolson timestepping. Thus, only constant timestepping is used when generating the numerical option values using the operator splitting method in Section 3.3.

Appendix F

Derivation of Analytical Solution when $q = 0$

This section contains the derivation of the analytical solution that can be used to determine the certainty equivalence value of employee stock options when the underlying company stock does not have any dividends (i.e. $q = 0$). The analytical solution presented here can replace Monte Carlo simulations in the second step of the pricing process when determining the expected utility when the employee's portfolio contains cash (instead of options) as outlined in Section 4.1.3.

Assuming the underlying company stock has no dividends (i.e. $q = 0$), the expected utility of the employee, $U = U(S, w, \tau)$, follows the differential equation:

$$U_\tau = \mu S U_S + \frac{1}{2}\sigma^2 S^2 U_{SS} + wrU_w, \tag{F.1}$$

where $\mu = r + \beta(r_m - r)$. We are looking to solve equation (F.1) analytically and provide an alternative to Monte Carlo simulations when solving the right-hand side of equation (4.3). We follow the derivation of the solution to the Black-Scholes equation as

outlined in [85].

We begin by setting $\xi = \ln(S)$ such that the employee's expected utility is now denoted as $U = U(\xi, w, \tau)$. Using the differential expressions:

$$\frac{\partial}{\partial S} = e^{-\xi}\frac{\partial}{\partial \xi}, \tag{F.2}$$

$$\frac{\partial^2}{\partial S^2} = e^{-2\xi}\frac{\partial^2}{\partial \xi^2} - e^{-2\xi}\frac{\partial}{\partial \xi}, \tag{F.3}$$

equation (F.1) is updated and we obtain:

$$U_\tau = (\mu - \frac{1}{2}\sigma^2)U_\xi + \frac{1}{2}\sigma^2 U_{\xi\xi} + wrU_w. \tag{F.4}$$

Defining $y = \ln(w)$, we now use this change of variable to simplify equation (F.4) further. Consequently, the expected utility denoted by $U = U(\xi, y, \tau)$ and the associated PDE now becomes:

$$U_\tau = (\mu - \frac{1}{2}\sigma^2)U_\xi + \frac{1}{2}\sigma^2 U_{\xi\xi} + rU_y. \tag{F.5}$$

Next, we use the following variable definitions to further simplify equation (F.5):

$$x(\tau) = y + r\tau, \tag{F.6}$$

$$z(\tau) = \xi + (\mu - \frac{1}{2}\sigma^2)\tau. \tag{F.7}$$

For notational purposes, we let $W(x(\tau), z(\tau), \tau) = U(y, \xi, \tau)$ and obtain the following simplified PDE:

$$W_\tau = \frac{1}{2}\sigma^2 W_{zz}. \tag{F.8}$$

Following the derivation outlined in [85], we obtain the following Green's function:

$$G(z, z', \tau) = \frac{1}{\sqrt{2\pi\tau}\sigma} e^{-\frac{(z-z')^2}{2\sigma^2\tau}}, \tag{F.9}$$

which is a solution of:

$$G_\tau - \frac{1}{2}\sigma^2 G_{zz} = \delta(\tau)\delta(z - z'), \tag{F.10}$$

where $\delta(x)$ is Dirac's delta function. The solution of equation (F.8) can now be written as:

$$W(x, z, \tau) = \int_{-\infty}^{\infty} G(z, z', \tau) W(x, z', 0) dz', \tag{F.11}$$

with $W(x, z, \tau) = U(y, \xi, \tau)$.

Let us now consider the initial conditions to be used for $U(S, w, \tau)$. At contract maturity (when $\tau = 0$), we can determine the value of the employee's expected utility in our original coordinate system as follows:

$$U(w, S, \tau = 0) = \frac{(we^{rT} + n_s S + ce^{rT})^{1-\alpha}}{1 - \alpha}, \tag{F.12}$$

or, in terms of the coordinate system for $W(x(\tau), z(\tau), \tau)$:

$$W(x, z, \tau = 0) = \frac{(e^x + n_s e^z + ce^{rT})^{1-\alpha}}{1 - \alpha}. \tag{F.13}$$

Recall that α is the employee's personal risk aversion coefficient.

To incorporate these initial conditions, we combine equations (F.11) and (F.13) to obtain:

$$W(x, z, \tau) = \frac{1}{\sqrt{2\pi\tau}\sigma} \int_{-\infty}^{\infty} e^{-\frac{(z-z')^2}{2\sigma^2\tau}} \frac{(e^x + n_s e^{z'} + ce^{rT})^{1-\alpha}}{1 - \alpha} dz'. \tag{F.14}$$

Returning to our original coordinate system with time running forward ($t = \tau + T$), we

find that equation (F.14) becomes:

$$U(S,w,t) = \frac{1}{\sqrt{2\pi(T-t)}\sigma}\int_0^\infty e^{-\frac{(\ln(S/S')+(\mu-\frac{1}{2}\sigma^2)(T-t))^2}{2\sigma^2(T-t)}}\frac{(n_sS'+we^{rT}+ce^{rT})^{1-\alpha}}{1-\alpha}\frac{dS'}{S'}. \tag{F.15}$$

Equation (F.15) provides us with an alternate method for solving the right-hand side of equation (4.3). Indeed, equation (F.15) can be solved using numerical integration techniques thus obtaining an accurate estimate for the employee's expected utility. Using the resulting expected utility, we can in turn determine the certainty equivalence value of a given employee stock option contract.

Appendix G

Derivation of the Boundary Condition as $S \to \infty$

To determine the boundary condition for equation (5.13) as $S \to \infty$, we make the common assumption that $V_{SS} \to 0$ [89], which implies:

$$V \approx H(B, D, \tau)S + F(B, D, \tau), \tag{G.1}$$

where $H(B, D, \tau)$ and $F(B, D, \tau)$ are independent of S. We further assume that S is so large that $H(B, D, \tau)S \gg F(B, D, \tau)$, which leads to:

$$V \approx H(B, D, \tau)S. \tag{G.2}$$

Equation (G.2) implies:

$$V_S \approx H(B, D, \tau), \tag{G.3}$$

and hence, we can rewrite the differential equation in (5.13) as:

$$H_\tau(B,D,\tau)S = (r-\rho_{\text{total}})H(B,D,\tau)S - rH(B,D,\tau)S - \mathcal{R}(\tau)\rho_{ins}S \tag{G.4}$$
$$+ \mathcal{M}(\tau)\max(B-S,0) + \mathcal{M}(\tau)\gamma(\tau)D + \frac{1}{\epsilon}\max\Big(\mathcal{A}(H(B,D,\tau)S) - H(B,D,\tau)S, 0\Big),$$

where

$$\mathcal{A}(H(B,D,\tau)S) =$$
$$\max\Big(-\mathcal{R}(\tau)\gamma(\tau)S, \max_{W\in[0,S-\omega]}\big(H(\max(B-W,0),\max(D-W,0),\tau)(S-W)$$
$$- \mathcal{R}(\tau)\gamma(\tau)W\big) - c\Big). \tag{G.5}$$

Since $B \ll S_{\max}$ and $W \le \mathcal{D}_0 \ll S_{\max}$, we can simplify equation (G.4) as:

$$H_\tau(B,D,\tau)S \approx \tag{G.6}$$
$$- \rho_{\text{total}}H(B,D,\tau)S - \mathcal{R}(\tau)\rho_{ins}S + \frac{1}{\epsilon}\max\Big(\mathcal{A}(H(B,D,\tau)S) - H(B,D,\tau)S, 0\Big).$$

As a result, we obtain the following approximation to equation (G.4):

$$V_\tau = -\rho_{\text{total}}V - \mathcal{R}(\tau)\rho_{ins}S + \frac{1}{\epsilon}\max(\mathcal{A}V - V, 0)\ ;\ S = S_{\max}. \tag{G.7}$$

A similar argument gives the boundary condition for large S when regime switching is used.

Appendix H

Proof of Theorem 5.5

In this appendix, we show that the discrete GMDB cost $V_{i,j,k,m}^{n+1}$ is bounded. Before proving Theorem 5.5, we prove some utility lemmas. We define the vector V^{n+1} as:

$$V^{n+1} = \begin{bmatrix} V_1^{n+1} \\ V_2^{n+1} \\ \vdots \\ V_M^{n+1} \end{bmatrix}, \tag{H.1}$$

where V_m^{n+1} is defined in equation (5.48) and the κ^{th} entry of V^{n+1} is denoted as $[V^{n+1}]_{i,j,k,m}$ where:

$$\kappa = (i+1) + j(i_{\max}+1) + k(i_{\max}+1)(j_{\max}+1) + (m-1)(i_{\max}+1)(j_{\max}+1)(k_{\max}+1).$$

Let $\mathcal{P}^{n+1}$ be defined as:

$$[\mathcal{P}^{n+1}\mathcal{Z}^{n+1}]_{i,j,k,m} = \left(1 + \Delta\tau\left(\alpha_{i,j,m} + \beta_{i,j,m} + r + \sum_{\substack{l=1\\l\neq m}}^{M} \lambda^{m\to l}\right)\right)\mathcal{Z}^{n+1}_{i,j,k,m} - \Delta\tau\alpha_{i,j,m}\mathcal{Z}^{n+1}_{i-1,j,k,m}$$
$$- \Delta\tau\beta_{i,j,m}\mathcal{Z}^{n+1}_{i+1,j,k,m} - \Delta\tau\sum_{\substack{l=1\\l\neq m}}^{M} \lambda^{m\to l}\mathcal{H}(J^{m\to l})_i\mathcal{Z}^{n+1}_{j,k,l} \tag{H.2}$$

when $i < i_{\max}$ and

$$[\mathcal{P}^{n+1}\mathcal{Z}^{n+1}]_{i_{\max},j,k,m} = \left(1 + \Delta\tau\left(\rho_{\text{total}} + \sum_{\substack{l=1\\l\neq m}}^{M} \lambda^{m\to l}J^{m\to l}_{i_{\max}}\right)\right)\mathcal{Z}^{n+1}_{i_{\max},j,k,m}$$
$$- \Delta\tau\sum_{\substack{l=1\\l\neq m}}^{M} \lambda^{m\to l}J^{m\to l}_{i_{\max}}\mathcal{Z}^{n+1}_{i_{\max},j,k,l} \tag{H.3}$$

when $i = i_{\max}$. Also, let $\mathcal{Q}^{n+1}(V^{n+1})$ be defined by:

$$[\mathcal{Q}^{n+1}(V^{n+1})\mathcal{Z}^{n+1}]_{i,j,k,m} = [\mathcal{P}^{n+1}\mathcal{Z}^{n+1}]_{i,j,k,m} + \frac{\Delta\tau\mu^{n+1}_{i,j,k,m}}{\epsilon}\mathcal{Z}^{n+1}_{i,j,k,m}$$
$$- \frac{\Delta\tau\mu^{n+1}_{i,j,k,m}}{\epsilon}(1 - a^{n+1}_{i,j,k,m})\mathcal{I}(W)_{i,j,k}\mathcal{Z}^{n+1}_{m}, \tag{H.4}$$

valid for all i. Here, $\mu^{n+1}_{i,j,k,m}$ is defined in equation (5.43), $a^{n+1}_{i,j,k,m}$ is defined in equation (5.51) and the interpolation operators $\mathcal{H}(J^{m\to l})_i$ and $\mathcal{I}(W)_{i,j,k}$ are defined in equations (5.45) and (5.49) respectively. The matrix $\mathcal{Q}^{n+1}(V^{n+1})$ is the matrix of coefficients for all terms involving elements from V^{n+1} in the discretization (5.40). Note that $\mathcal{Q}^{n+1}(V^{n+1})$ is a function of the solution since the interpolation operators, the μ and a values all depend on the solution.

It is useful to note the following property of the coefficient matrices $\mathcal{P}^{n+1}$ and $\mathcal{Q}^{n+1}(V^{n+1})$.

Lemma H.1 (M-matrix). *The matrices $\mathcal{P}^{n+1}$ and $\mathcal{Q}^{n+1}(V^{n+1})$ as defined in equa-*

tions (H.2),(H.3) *and* (H.4) *are M-matrices for any* V^{n+1}.

Proof. The diagonal entries in $\mathcal{P}^{n+1}$ are positive while the off-diagonal entries are negative or equal to zero. In addition, the row sum of the entries in both matrices are strictly positive for all rows. The above are also true for the matrix $\mathcal{Q}^{n+1}(V^{n+1})$ for any V^{n+1}. Thus both $\mathcal{P}^{n+1}$ and $\mathcal{Q}^{n+1}(V^{n+1})$ are M-matrices. □

Remark H.2. *We remark that an M-matrix has the important property that it is invertible with a positive inverse. In particular, for any vector* Z, $\mathcal{P}^{n+1}Z \geq 0$ *or* $\mathcal{Q}^{n+1}(V^{n+1})Z \geq 0$ *implies that* $Z \geq 0$.

Lemma H.3. *The following are true.*

(a) Let $[\mathcal{Z}^{n+1}]_{i,j,k,m} = C_0^{n+1}B_{\max} + C_1^{n+1}D_{\max}$ *(with* C_0^{n+1}, C_1^{n+1} *defined in (5.58)). Then:*

$$\mathcal{Q}^{n+1}(V^{n+1})\mathcal{Z}^{n+1} > \mathcal{Z}^{n+1}$$

for any V^{n+1}.

(b) Let $[\mathcal{Z}^{n+1}]_{i,j,k,m} = S_i^j$. *Then*[1]*:*

$$\mathcal{P}^{n+1}\mathcal{Z}^{n+1} = (1 + \rho_{\text{total}}\Delta\tau)\mathcal{Z}^n.$$

(c) Let $\mathcal{Z}$ *solve the discrete equations (5.40). Then:*

$$\mathcal{Q}^{n+1}(\mathcal{Z}^{n+1})\mathcal{Z}^{n+1} = \mathcal{Z}^n + \Delta\tau\text{Rest}^{n+1},$$

[1]Note that this is trivially true at $S_0^j = 0$.

where for all i (since $f_{i_{\max},j,k}=0$)

$$[\text{Rest}^{n+1}]_{i,j,k,m} = \mathcal{M}^{n+1} f_{i,j,k} - \mathcal{R}^{n+1}\rho_{ins}S_i^j - \frac{\mu_{i,j,k,m}^{n+1}}{\epsilon}\Big[a_{i,j,k,m}^{n+1}\mathcal{R}^{n+1}\gamma^{n+1}S_i^j + (1-a_{i,j,k,m}^{n+1})(\mathcal{R}^{n+1}\gamma^{n+1}W_{i,j,k,m}^{n+1}+c)\Big] \tag{H.5}$$

denotes the remaining terms of the discretization.

(d) Let $\mathcal{Z}$ solve the discrete equations (5.40). Then:

$$\mathcal{P}^{n+1}\mathcal{Z}^{n+1} = \mathcal{Z}^n + \Delta\tau\text{Rest}^{n+1},$$

where for all i (since $f_{i_{\max},j,k}=0$)

$$\begin{aligned}[\text{Rest}^{n+1}]_{i,j,k,m} &= \mathcal{M}^{n+1} f_{i,j,k} - \mathcal{R}^{n+1}\rho_{ins}S_i^j + \frac{\mu_{i,j,k,m}^{n+1}}{\epsilon}\Big[-a_{i,j,k,m}^{n+1}\mathcal{R}^{n+1}\gamma^{n+1}S_i^j \\ &\quad +(1-a_{i,j,k,m}^{n+1})(\mathcal{I}(W)_{i,j,k}\mathcal{Z}_m^{n+1} - \mathcal{R}^{n+1}\gamma^{n+1}W_{i,j,k,m}^{n+1} - c) - \mathcal{Z}_{i,j,k,m}^{n+1}\Big].\end{aligned} \tag{H.6}$$

Proof. Identity (a) follows by looking at the i,j,k,m components of the matrix form of $\mathcal{P}$ and $\mathcal{Q}$. For example, when $i < i_{\max}$ we have

$$\begin{aligned}[\mathcal{Q}^{n+1}(V^{n+1})\mathcal{Z}^{n+1}]_{i,j,k,m} &= (1+\Delta\tau(r+a_{i,j,k,m}\frac{\mu_{i,j,k,m}}{\epsilon}))[\mathcal{Z}^{n+1}]_{i,j,k,m} \\ &> [\mathcal{Z}^{n+1}]_{i,j,k,m}\end{aligned}$$

with a similar inequality when $i = i_{\max}$. A similar argument holds for identity (b). Identities (c) and (d) follow directly from the definitions of $\mathcal{Q}$ and $\mathcal{P}$ and the discretization in (5.40). □

We now present the proof of Theorem 5.5.

Proof. (of Theorem 5.5)

Let $\mathcal{Z}^n$ be the vector defined by $[\mathcal{Z}^n]_{i,j,k,m} = S_i^j + V_{i,j,k,m}^n$ for all i, j, k, m. We will use induction to show that $\mathcal{Z}^n \geq 0$ for all n.

Notice that $[\mathcal{Z}^0]_{i,j,k,m} = S_i^j + V_{i,j,k,m}^0 = S_i^j \geq 0$. Assume now that $n > 0$ and that $\mathcal{Z}^n \geq 0$. Then, from Lemma H.3(b)(d) we have:

$$[\mathcal{P}^{n+1}\mathcal{Z}^{n+1}] = \mathcal{Z}^n + \Delta\tau\mathcal{G}^{n+1}, \tag{H.7}$$

with (since $f_{i,j,k} \geq 0$)

$$\begin{aligned}[\mathcal{G}^{n+1}]_{i,j,k,m} \geq &(\rho_{total} - \mathcal{R}^{n+1}\rho_{ins})S_i^j + \frac{\mu_{i,j,k,m}^{n+1}}{\epsilon}\Big[-a_{i,j,k,m}^{n+1}\mathcal{R}^{n+1}\gamma^{n+1}S_i^j \\ &+ (1 - a_{i,j,k,m}^{n+1})(\mathcal{I}(W)_{i,j,k}V_m^{n+1} - \mathcal{R}^{n+1}\gamma^{n+1}W_{i,j,k,m}^{n+1} - c) - V_{i,j,k,m}^{n+1}\Big].\end{aligned} \tag{H.8}$$

Note that $\rho_{\text{total}} - \mathcal{R}^{n+1}\rho_{ins} \geq 0$. Furthermore, notice that $\mu_{i,j,k,m}^{n+1} = 1$ only when (see equation (5.43)):

$$-a_{i,j,k,m}^{n+1}\mathcal{R}^{n+1}\gamma^{n+1}S_i^j + (1 - a_{i,j,k,m}^{n+1})(\mathcal{I}(W)_{i,j,k}V_m^{n+1} - \mathcal{R}^{n+1}\gamma^{n+1}W_{i,j,k,m}^{n+1} - c) - V_{i,j,k,m}^{n+1} > 0$$

and $\mu_{i,j,k,m}^{n+1} = 0$ otherwise. Hence, equation (H.8) implies that $[\mathcal{G}^{n+1}]_{i,j,k,m} \geq 0$.

Since $\mathcal{Z}^n \geq 0$, we see that $\mathcal{P}^{n+1}\mathcal{Z}^{n+1} \geq 0$ and, since $\mathcal{P}^{n+1}$ is an M-matrix, $\mathcal{Z}^{n+1} \geq 0$. Thus, by induction $\mathcal{Z}^n \geq 0$ for all n, proving the first inequality of (5.57).

Now let $\mathcal{Z}$ be the vector defined by $[\mathcal{Z}^n]_{i,j,k,m} = C_0^n B_{\max} + C_1^n D_{\max}$ for all i, j, k, m. We will prove the second inequality of (5.57) by using induction to show that $\mathcal{Z}^n - V^n \geq 0$ for all n. Since (see equation (5.58)):

$$[\mathcal{Z}^0 - V^0]_{i,j,k,m} = \Delta\tau\mathcal{M}^0\ B_{\max} + \Delta\tau\mathcal{M}^0\gamma^0 D_{\max} \geq 0, \tag{H.9}$$

the result is true for $n = 0$. Assume that $n > 0$ and that $\mathcal{Z}^n - V^n \geq 0$. From Lemma H.3(a) along with the definition of C_0^n and C_1^n (see equation (5.58)) we have:

$$\mathcal{Q}^{n+1}(V^{n+1})\mathcal{Z}^{n+1} > \mathcal{Z}^{n+1} = \mathcal{Z}^n + \Delta\tau[\mathcal{M}^{n+1}B_{\max} + \mathcal{M}^{n+1}\gamma^{n+1}D_{\max}].$$

Hence, using Lemma H.3(c) gives:

$$\mathcal{Q}^{n+1}(V^{n+1})(\mathcal{Z}^{n+1}-V^{n+1}) > (\mathcal{Z}^n-V^n)+\Delta\tau[\mathcal{M}^{n+1}B_{\max}+\mathcal{M}^{n+1}\gamma^{n+1}D_{\max}]-\Delta\tau\text{Rest}^{n+1},$$

where the components of Rest^{n+1} are given in equation (H.5). Let

$$\mathcal{G} = [\mathcal{M}^{n+1}B_{\max} + \mathcal{M}^{n+1}\gamma^{n+1}D_{\max}] - \text{Rest}^{n+1}.$$

Then, for $i < i_{\max}$, and using:

$$0 \leq f_{i,j,k}^{n+1} = \max(B_j^k - S_i^j, 0) + \gamma^{n+1}D_k \leq B_{\max} + \gamma^{n+1}D_{\max}\ , \tag{H.10}$$

we have:

$$\begin{aligned}
[\mathcal{G}]_{i,j,k,m} &= \mathcal{M}^{n+1}\left(B_{\max} + \gamma^{n+1}D_{\max} - f_{i,j,k}\right) + \mathcal{R}^{n+1}\rho_{ins}S_i^j \\
&\quad + \frac{\mu_{i,j,k,m}^{n+1}}{\epsilon}\left[a_{i,j,k,m}^{n+1}\mathcal{R}^{n+1}\gamma^{n+1}S_i^j + (1 - a_{i,j,k,m}^{n+1})(\mathcal{R}^{n+1}\gamma^{n+1}W_{i,j,k,m}^{n+1} + c)\right] \\
&\geq \mathcal{R}^{n+1}\rho_{ins}S_i^j \\
&\quad + \frac{\mu_{i,j,k,m}^{n+1}}{\epsilon}\left[a_{i,j,k,m}^{n+1}\mathcal{R}^{n+1}\gamma^{n+1}S_i^j + (1 - a_{i,j,k,m}^{n+1})(\mathcal{R}^{n+1}\gamma^{n+1}W_{i,j,k,m}^{n+1} + c)\right] \\
&\geq 0,
\end{aligned} \tag{H.11}$$

since there are only positive terms in the expression. This is also the case when $i = i_{\max}$. As before, $\mathcal{Z}^n - V^n \geq 0$ so that $\mathcal{Q}^{n+1}(V^{n+1})(\mathcal{Z}^{n+1} - V^{n+1}) \geq 0$ and, since $\mathcal{Q}^{n+1}(V^{n+1})$

is an M-matrix, $\mathcal{Z}^{n+1} - V^{n+1} \geq 0$. Hence, by induction, $\mathcal{Z}^n - V^n \geq 0$ for all n.

Thus, we have shown that $V^{n+1}_{i,j,k,m}$ is bounded with:

$$-S^j_i \leq V^{n+1}_{i,j,k,m} \leq \mathcal{C}^{n+1}_0 B_{\max} + \mathcal{C}^{n+1}_1 D_{\max} \text{ for all } i,j,k,m,n. \tag{H.12}$$

Note that the bound presented in equation (H.12) also holds immediately after each ratchet date τ^{u+}_o. Recall that the value of the GMDB guarantee is updated on each ratchet date τ^u_o according to equation (5.35), which implies (for the continuous problem):

$$V^m(S,B,D,e_m,\tau^{u+}_o) = \begin{cases} V^m(S,B,D,e_m,\tau^{u-}_o) & \text{if } S \leq B, \\ V^m(S,S,D,e_m,\tau^{u-}_o) & \text{if } B < S \leq B_{\max}, \\ V^m(S,B,D,e_m,\tau^{u-}_o) & \text{if } S > B_{\max}. \end{cases} \tag{H.13}$$

Equation (H.13) implies that the bound for $V^{n+1}_{i,j,k,m}$ presented in equation (H.12) remains applicable at times τ^{u+}_o.

□

Remark H.4 (Tighter Upper Bound). *We remark that it is possible to obtain the tighter bound for* $V^{n+1}_{i,j,k,m}$*:*

$$-S^j_i \leq V^{n+1}_{i,j,k,m} \leq \mathcal{C}^{n+1}_0 B_{\max} + \mathcal{C}^{n+1}_1 D_k \text{ for all } i,j,k,m,n. \tag{H.14}$$

However, bound (5.57) *is sufficient for our purposes.*

Appendix I

Proof of Theorem 5.9

In this section, we show that the numerical scheme in equation (5.40) is consistent. Before proving Theorem 5.9, we prove an important lemma.

Lemma I.1. *For any smooth test function ϕ with bounded derivatives of all orders with respect to S and τ, with $x = (S_i^j, B_j^k, D_k, e_m, \tau^{n+1})$, we have (see equation (5.66)):*

$$\hat{\mathcal{G}}\big(h, x, \phi_{i,j,k,m}^{n+1} + \xi, \phi_{i,j,k,m}^{n} + \xi, \{\phi_{a,p,u,l}^{n+1} + \xi\}\big) - F(\phi)_{i,j,k,m}^{n+1} = O(h) + \xi b(x), \tag{I.1}$$

where $b(x)$ is a bounded function of x with $|b(x)| \leq \max(r, \rho_{\text{total}})$.

Proof. To prove Lemma I.1, we consider the truncation error for the differential operator $\mathcal{L}$ and the penalty term.

Let

$$[\mathcal{L}\phi]_{i,j,k,m}^{n+1} \tag{I.2}$$

represent the continuous operator $\mathcal{L}$ at node $(S_i^j, B_j^k, D_k, e_m, \tau^{n+1})$, while the discrete

version of the operator is denoted by:

$$\left[\mathcal{L}^h\phi\right]_{i,j,k,m}^{n+1}. \tag{I.3}$$

Using Taylor series expansion, we have:

$$\left[\mathcal{L}^h(\phi+\xi)\right]_{i,j,k,m}^{n+1} - [\mathcal{L}\phi]_{i,j,k,m}^{n+1} = -r\xi + O(\Delta S_{\max}^j), \tag{I.4}$$

when computing $\mathcal{H}_i\phi_{j,k,l}^{n+1}$ using linear interpolation (see equation (5.45)).

Similarly, we assume that:

$$[\mathcal{A}\phi]_{i,j,k,m}^{n+1} \tag{I.5}$$

represents the continuous withdrawal constraint evaluated at node $(S_i^j, B_j^k, D_k, e_m, \tau^{n+1})$, while the discrete version of the withdrawal constraint is denoted as:

$$[\mathcal{A}^h\phi]_{i,j,k,m}^{n+1}. \tag{I.6}$$

Recall that the discrete withdrawal constraint is determined by linear search as in Algorithm 5.1.

The discretization error associated with the penalty term occurs when it is optimal for the owner to conduct a withdrawal, as opposed to lapsing his policy. Indeed, interpolation is required when calculating the penalty term when a withdrawal occurs, but not when the owner lapses (see equation (5.47)). Since the maximum of a linearly interpolated value is obtained at the nodes, the linear interpolation truncation error is $O(h^2)$ (noting Assumption 5.6). Taking the maximum of the linear interpolation function, as done in Algorithm 5.1, is also second order correct. Assuming two-dimensional linear interpolation is used when calculating the withdrawal constraint as described in equa-

tion (5.49), the interpolation error will be $O(\Delta S^j_{\max}\Delta B^u_{\max})$. Therefore, we obtain (from equation (5.52)):

$$[\mathcal{A}^h(\phi+\xi)]^{n+1}_{i,j,k,m} - [\mathcal{A}\phi]^{n+1}_{i,j,k,m} = \xi + O(\Delta S^j_{\max}\Delta B^u_{\max}) + O(h^2) \tag{I.7}$$

when it is optimal to withdraw and zero when it is optimal to lapse.

Recall from equation (5.59) that the discrete scheme $\mathcal{G}\left(h, x, V^{n+1}_{i,j,k,m}, V^n_{i,j,k,m}, \{V^{n+1}_{a,p,u,l}\}\right)$ is denoted as follows on interior nodes when $S^j_i < S_{\max}$:

$$\frac{\phi^{n+1}_{i,j,k,m} - \phi^n_{i,j,k,m}}{\Delta\tau} - [\mathcal{L}^h\phi]^{n+1}_{i,j,k,m} + \mathcal{R}^{n+1}\rho_{ins}S^j_i - \mathcal{M}^{n+1}f^{n+1}_{i,j,k} - \frac{1}{\epsilon}\max\left([\mathcal{A}^h\phi]^{n+1}_{i,j,k,m} - \phi^{n+1}_{i,j,k,m}, 0\right) = 0. \tag{I.8}$$

We re-arrange the penalized problem in equation (I.8) as:

$$\min\Bigg[\frac{\phi^{n+1}_{i,j,k,m} - \phi^n_{i,j,k,m}}{\Delta\tau} - [\mathcal{L}^h\phi]^{n+1}_{i,j,k,m} + \mathcal{R}^{n+1}\rho_{ins}S^j_i - \mathcal{M}^{n+1}f^{n+1}_{i,j,k} - \frac{1}{\epsilon}\left([\mathcal{A}^h\phi]^{n+1}_{i,j,k,m} - \phi^{n+1}_{i,j,k,m}\right),$$
$$\frac{\phi^{n+1}_{i,j,k,m} - \phi^n_{i,j,k,m}}{\Delta\tau} - [\mathcal{L}^h\phi]^{n+1}_{i,j,k,m} + \mathcal{R}^{n+1}\rho_{ins}S^j_i - \mathcal{M}^{n+1}f^{n+1}_{i,j,k}\Bigg] = 0. \tag{I.9}$$

Equation (I.9) implies that one of the following holds with equality:

$$\frac{\phi^{n+1}_{i,j,k,m} - \phi^n_{i,j,k,m}}{\Delta\tau} - [\mathcal{L}^h\phi]^{n+1}_{i,j,k,m} + \mathcal{R}^{n+1}\rho_{ins}S^j_i - \mathcal{M}^{n+1}f^{n+1}_{i,j,k} - \frac{1}{\epsilon}\left([\mathcal{A}^h\phi]^{n+1}_{i,j,k,m} - \phi^{n+1}_{i,j,k,m}\right) \geq 0, \tag{I.10}$$

$$\frac{\phi^{n+1}_{i,j,k,m} - \phi^n_{i,j,k,m}}{\Delta\tau} - [\mathcal{L}^h\phi]^{n+1}_{i,j,k,m} + \mathcal{R}^{n+1}\rho_{ins}S^j_i - \mathcal{M}^{n+1}f^{n+1}_{i,j,k} \geq 0. \tag{I.11}$$

Since $\epsilon > 0$, equation (I.10) is equivalent to:

$$\epsilon\left(\frac{\phi^{n+1}_{i,j,k,m} - \phi^n_{i,j,k,m}}{\Delta\tau} - [\mathcal{L}^h\phi]^{n+1}_{i,j,k,m} + \mathcal{R}^{n+1}\rho_{ins}S^j_i - \mathcal{M}^{n+1}f^{n+1}_{i,j,k}\right) + \phi^{n+1}_{i,j,k,m} - [\mathcal{A}^h\phi]^{n+1}_{i,j,k,m} \geq 0. \tag{I.12}$$

Similarly, equations (I.11) and (I.12) can be combined to obtain:

$$\min\Bigg(\epsilon\left(\frac{\phi^{n+1}_{i,j,k,m}-\phi^{n}_{i,j,k,m}}{\Delta\tau}-[\mathcal{L}^h\phi]^{n+1}_{i,j,k,m}+\mathcal{R}^{n+1}\rho_{ins}S^j_i-\mathcal{M}^{n+1}f^{n+1}_{i,j,k}\right)+\phi^{n+1}_{i,j,k,m}-[\mathcal{A}^h\phi]^{n+1}_{i,j,k,m}\,,$$
$$\frac{\phi^{n+1}_{i,j,k,m}-\phi^{n}_{i,j,k,m}}{\Delta\tau}-[\mathcal{L}^h\phi]^{n+1}_{i,j,k,m}+\mathcal{R}^{n+1}\rho_{ins}S^j_i-\mathcal{M}^{n+1}f^{n+1}_{i,j,k}\Bigg)=0, \tag{I.13}$$

which corresponds to the definition of $\hat{\mathcal{G}}(h,x,V^{n+1}_{i,j,k,m},V^{n}_{i,j,k,m},\{V^{n+1}_{a,p,u,l}\})$ in equation (5.66) for interior nodes. Applying the same technique for the boundary nodes, we can show the equivalence between the original scheme $\mathcal{G}(h,x,V^{n+1}_{i_{\max},j,k,m},V^{n}_{i_{\max},j,k,m},\{V^{n+1}_{a,p,u,l}\})$ in equation (5.60) and $\hat{\mathcal{G}}(h,x,V^{n+1}_{i_{\max},j,k,m},V^{n}_{i_{\max},j,k,m},\{V^{n+1}_{a,p,u,l}\})$ in equation (5.67). This demonstration is omitted for brevity.

Using the result in equation (I.13) and the discretization error estimates in equations (I.4) and (I.7), we find for the interior nodes when $S^j_i < S_{\max}$ (noting that $|\max(x,y)-\max(\alpha,\beta)|\le\max(|x-\alpha|,|y-\beta|)$):

$$\Big|\hat{\mathcal{G}}\big(h,x,\phi^{n+1}_{i,j,k,m}+\xi,\phi^{n}_{i,j,k,m}+\xi,\{\phi^{n+1}_{a,p,u,l}+\xi\}\big)-F_{in}(\phi)^{n+1}_{i,j,k,m}\Big| \tag{I.14}$$
$$\le\max\Bigg(\Bigg|\Big(\phi^{n+1}_{i,j,k,m}+\xi-\mathcal{A}^h(\phi^{n+1}_{i,j,k,m}+\xi)\Big)-\big[\phi-\mathcal{A}\phi\big]^{n+1}_{i,j,k,m}$$
$$+\epsilon\Big(\frac{\phi^{n+1}_{i,j,k,m}-\phi^{n}_{i,j,k,m}}{\Delta\tau}-[\mathcal{L}^h(\phi+\xi)]^{n+1}_{i,j,k,m}+\mathcal{R}^{n+1}\rho_{ins}S^j_i-\mathcal{M}^{n+1}f^{n+1}_{i,j,k}\Big)\Bigg|,$$
$$\Bigg|\Big(\frac{\phi^{n+1}_{i,j,k,m}-\phi^{n}_{i,j,k,m}}{\Delta\tau}-[\mathcal{L}^h(\phi+\xi)]^{n+1}_{i,j,k,m}+\mathcal{R}^{n+1}\rho_{ins}S^j_i-\mathcal{M}^{n+1}f^{n+1}_{i,j,k}\Big)$$
$$-\big[\phi_\tau-\mathcal{L}\phi+\mathcal{R}(\tau)\rho_{ins}S-\mathcal{M}(\tau)f\big]^{n+1}_{i,j,k,m}\Bigg|\Bigg)$$
$$=\max\Bigg(\Bigg|O(\Delta S^j_{\max}\Delta B^u_{\max})+O(h^2)+\epsilon\Big(\frac{\phi^{n+1}_{i,j,k,m}-\phi^{n}_{i,j,k,m}}{\Delta\tau}-[\mathcal{L}^h\phi]^{n+1}_{i,j,k,m}+\mathcal{R}^{n+1}\rho_{ins}S^j_i$$
$$-\mathcal{M}^{n+1}f^{n+1}_{i,j,k}-\xi r\Big)\Bigg|,\Big|O(\Delta\tau)+O(\Delta S^j_{\max})+r\xi\Big|\Bigg)$$

Similarly, for the boundary nodes when $S_i^j = S_{\max}$, we have:

$$
\begin{aligned}
&\left|\hat{\mathcal{G}}\big(h, x, \phi^{n+1}_{i_{\max},j,k,m} + \xi, \phi^{n}_{i_{\max},j,k,m} + \xi, \{\phi^{n+1}_{a,p,u,l} + \xi\}\big) - F_{bound}(\phi)^{n+1}_{i_{\max},j,k,m}\right| \qquad \text{(I.15)}\\
&\le \max\Bigg(\Bigg|\Big(\phi^{n+1}_{i,j,k,m} + \xi - \mathcal{A}^h(\phi^{n+1}_{i,j,k,m} + \xi)\Big) - \big[\phi - \mathcal{A}\phi\big]^{n+1}_{i,j,k,m} + \epsilon\Bigg(\frac{\phi^{n+1}_{i_{\max},j,k,m} - \phi^{n}_{i_{\max},j,k,m}}{\Delta\tau}\\
&\qquad + \rho_{\text{total}}(\phi^{n+1}_{i_{\max},j,k,m} + \xi) - \sum_{\substack{l=1\\ l\neq m}}^{M} \lambda^{m\to l} J^{m\to l}_{i_{\max}}(\phi^{n+1}_{i_{\max},j,k,l} - \phi^{n+1}_{i_{\max},j,k,m}) + \mathcal{R}^{n+1}\rho_{ins} S^j_{i_{\max}}\Bigg)\Bigg|,\\
&\qquad \Bigg|\Bigg(\frac{\phi^{n+1}_{i_{\max},j,k,m} - \phi^{n}_{i_{\max},j,k,m}}{\Delta\tau} + \rho_{\text{total}}(\phi^{n+1}_{i_{\max},j,k,m} + \xi)\\
&\qquad - \sum_{\substack{l=1\\ l\neq m}}^{M} \lambda^{m\to l} J^{m\to l}_{i_{\max}}(\phi^{n+1}_{i_{\max},j,k,l} - \phi^{n+1}_{i_{\max},j,k,m}) + \mathcal{R}^{n+1}\rho_{ins} S^j_{i_{\max}}\Bigg) - \Bigg[\phi_\tau + \rho_{\text{total}}\phi\\
&\qquad - \sum_{\substack{l=1\\ l\neq m}}^{M} \lambda^{m\to l} J^{m\to l}(S)\big(\phi(S, B, D, e_l, \tau) - \phi\big) + \mathcal{R}(\tau)\rho_{ins} S\Bigg]^{n+1}_{i_{\max},j,k,m}\Bigg|\Bigg)\\
&= \max\Bigg(\Bigg|O(\Delta S^j_{\max}\Delta B^u_{\max}) + O(h^2) + \epsilon\Bigg(\frac{\phi^{n+1}_{i_{\max},j,k,m} - \phi^{n}_{i_{\max},j,k,m}}{\Delta\tau} + \rho_{\text{total}}(\phi^{n+1}_{i_{\max},j,k,m} + \xi)\\
&\qquad - \sum_{\substack{l=1\\ l\neq m}}^{M} \lambda^{m\to l} J^{m\to l}_{i_{\max}}(\phi^{n+1}_{i_{\max},j,k,l} - \phi^{n+1}_{i_{\max},j,k,m}) + \mathcal{R}^{n+1}\rho_{ins} S^j_{i_{\max}}\Bigg)\Bigg|,\\
&\qquad \Big|O(\Delta\tau) + O(\Delta S^j_{\max}) + \rho_{\text{total}}\xi\Big|\Bigg).
\end{aligned}
$$

Using Assumption 5.4, we obtain:

$$
\hat{\mathcal{G}}\big(h, x, \phi^{n+1}_{i,j,k,m} + \xi, \phi^{n}_{i,j,k,m} + \xi, \{\phi^{n+1}_{a,p,u,l} + \xi\}\big) = F(\phi)^{n+1}_{i,j,k,m} + O(h) + \xi b(x), \qquad \text{(I.16)}
$$

for both boundary and interior nodes, where $b(x)$ is a bounded function with $|b(x)| \le \max(r, \rho_{\text{total}})$. □

We now present the proof of Theorem 5.9.

Proof. (of Theorem 5.9)

We begin by proving that equation (5.68) holds. From the definition of lim sup, there exists sequences $h_d, i_d, j_d, k_d, n_d, \xi_d$ such that

$$h_d \to 0,\ \xi_d \to 0,\ x_d = (S_{i_d}^{j_d}, B_{j_d}^{k_d}, D_{k_d}, e_m, \tau^{n_d+1}) \to \hat{x} = (\hat{S}, \hat{B}, \hat{D}, e_m, \hat{\tau}) \ \text{ as } \ d \to \infty, \tag{I.17}$$

and

$$\begin{aligned} \limsup_{d\to\infty} \hat{\mathcal{G}}\big(h_d, x_d, \phi_{i_d,j_d,k_d,m}^{n_d+1} + \xi_d, \phi_{i_d,j_d,k_d,m}^{n_d} + \xi_d, \{\phi_{a_d,p_d,u_d,l}^{n_d+1} + \xi_d\}\big) \\ = \limsup_{\substack{\xi,\, h\to 0 \\ x\to\hat{x}}} \hat{\mathcal{G}}\big(h, x, \phi_{i,j,k,m}^{n+1} + \xi, \phi_{i,j,k,m}^{n} + \xi, \{\phi_{a,p,u,l}^{n+1} + \xi\}\big). \end{aligned} \tag{I.18}$$

From our result in equation (I.1), we have:

$$\hat{\mathcal{G}}\big(h_d, x_d, \phi_{i_d,j_d,k_d,m}^{n_d+1} + \xi_d, \phi_{i_d,j_d,k_d,m}^{n_d} + \xi_d, \{\phi_{a_d,p_d,u_d,l}^{n_d+1} + \xi_d\}\big) = F(\phi(x_d)) + O(h_d) + \xi_d b(x_d), \tag{I.19}$$

where $F(\phi(x))$ is defined in equation (5.63) for interior and boundary nodes.

Now consider a sequence of nodes x_d as defined in equation (I.17) which may contain both interior ($S_{i_d}^{j_d} < S_{\max}$) and boundary nodes ($S_{i_d}^{j_d} = S_{\max}$). Combining equation (I.19) with equation (I.18), we get:

$$\begin{aligned} \limsup_{\substack{\xi,\, h\to 0 \\ x\to\hat{x}}} \hat{\mathcal{G}}\big(h, x, \phi_{i,j,k,m}^{n+1} + \xi, \phi_{i,j,k,m}^{n} + \xi, \{\phi_{a,p,u,l}^{n+1} + \xi\}\big) \\ \le \limsup_{d\to\infty} F(\phi(x_d)) + \limsup_{d\to\infty} [O(h_d) + \xi_d b(x_d)] = F^*(\phi(\hat{x})) \end{aligned}$$

where the last equality holds because of:

$$\limsup_{d\to\infty} [O(h_d) + \xi_d b(x_d)] = 0. \tag{I.20}$$

Verifying equation (5.69) can be done in a similar fashion. □

Having shown that equations (5.68) and (5.69) hold, we conclude that the discrete equations in (5.40) are consistent according to Definition 5.8.

Appendix J

Comparison with Previous GMDB Numerical Results

In [72], the authors present an analytical model to price GMDB contracts with different death benefit guarantees including return-of-premium, rising floor and ratchets. More specifically, the authors determine the fair insurance charge that equates the present value of the risk charges with the value of the death benefit guarantee. While mostly focusing on guarantees with a rising floor, basic numerical results for contracts with a continuous lookback or ratchet feature are included in [72]. To validate the GMDB pricing model presented in Section 5.1, we attempt to reproduce the numerical results presented in [72] when valuing a GMDB clause with ratchets.

For consistency with the problem considered in [72], we modify the GMDB pricing problem presented in Section 5.1 to satisfy the following:

- Since the authors of [72] focus on determining the value for ρ_{ins}, the contract considered does not include any management fees. Consequently, we set $\rho_{man} = 0$.
- The contract considered in [72] does not include the partial withdrawal or lapsing

feature. Thus, we will solve the following pricing equation:

$$V_\tau = \frac{1}{2}\sigma^2 S^2 V_{SS} + (r - \rho_{ins})SV_S - rV - \mathcal{R}(\tau)\rho_{ins}S + \mathcal{M}(\tau)\max(B - S, 0), \quad \text{(J.1)}$$

without imposing an impulse control.

- To reproduce the continuous ratchet assumption, we apply the update feature presented in equation (5.17) discretely at each timestep during the solution process. As $\Delta\tau \to 0$, the value of the GMDB guarantee will converge to the contract value with continuous ratchets.

In [72], the authors assume that the contract terminates when the owner is 75 years old and consider a range of values for the age of the contract owner at the time of purchase (namely $30, 40, 50, 60$ and 65 years old). We will focus our analysis on the case most similar to the rest of the results in this paper and assume that the contract owner is 50 years old at the time of purchase; this implies that $T = 25$ years in our pricing model.

To be consistent with [72], we set $\sigma = 0.20$ and $r = 0.06$. In addition, the mortality data is generated with a Gompertz mortality distribution using the parameters presented in [72] corresponding to the age of the contract owner when the contract is purchased. The parameters in [72] are obtained by fitting a Gompertz mortality distribution to the 1994 Group Annuity Mortality Table (Basic) over the contract lifetime. In our case, we approximate the continuous mortality function with a discrete mortality distribution generated with $\Delta\tau = 6.25 \times 10^{-4}$ years. Such a small $\Delta\tau$ is chosen to avoid interpolation issues for higher refinement levels.

Recall that we are looking to determine the fair insurance charge ρ_{ins} that satisfies:

$$V(\rho_{ins}; S = \$100, B = \$100, D = \$100, \tau = T) = 0. \quad \text{(J.2)}$$

GMDB with Continuous Ratchet					
		Nodes		Insurance Charge - ρ_{ins}	
Refinement	Timesteps	S	B	Male	Female
0	2500	80	80	0.003960	0.002334
1	5000	159	159	0.004042	0.002383
2	10000	317	314	0.004086	0.002409
3	20000	633	633	0.004114	0.002426
4	40000	1265	1265	0.004133	0.002437
Value from [72]				0.00418	0.00246

TABLE J.1: *Fair insurance charge ρ_{ins} for a GMDB contract with discrete ratchet events when the owner is assumed to be 50 years old when the contract is purchased. The contract assumptions are chosen to approximate those in [72]. Crank-Nicolson timestepping with constant timesteps was used. We assume $\sigma = 0.20$, $r = 0.06$, $\rho_{man} = 0$ and set the initial timestep is set to $\Delta\tau = 0.01$ years on the coarsest grid.*

Newton iteration is used during the solution process and the tolerance is set to 1×10^{-6}. The resulting insurance charges are presented in Table J.1.

We see that the results obtained in Table J.1 are consistent with those presented in [72] but exhibit slow convergence. Keep in mind that the authors of [72] generate their results with analytical formulas while we approximate the contract considered by using discrete ratchet events. In Table J.1, we are essentially valuing a discrete lookback option which is a difficult problem. As the ratchet interval is reduced, the value of a discrete lookback is known to converge very slowly to the corresponding continuous lookback value [6, 26, 54].

Nonetheless, the numerical results in Table J.1 are certainly sufficient for practical purposes. Similar levels of accuracy were observed when comparing our numerical results to the analytical values in [72] for the remaining cases (i.e. when the owner is assumed to be $30, 40, 60$ and 65 years old).

Bibliography

[1] Statement of Financial Accounting Standards No. 123 (revised 2004) - Share-Based Payment. Financial Accounting Series, Financial Accounting Standards Board, December 2004.

[2] Numerical algorithms group. http://www.nag.co.uk/.

[3] A. L. Amadori. *Differential and Integro-differential Nonlinear Equations of Degenerate Parabolic Type Arising in the Pricing of Derivatives in Incomplete Markets*. PhD thesis, University of Rome, La Sapienza, 2001.

[4] A. L. Amadori. Quasi-variational inequalities with Dirichlet boundary conditions related to exit time problems for impulse control. *SIAM Journal on Control and Optimization*, 43(2):570–589, 2004.

[5] E. Ayache, P. Henrotte, S. Nassar, and X. Wang. Can anyone solve the smile problem? *Wilmott Magazine*, pages 78–96, January 2004.

[6] S. Babbs. Binomial valuation of lookback options. *Journal of Economic Dynamics and Control*, 24:1499–1525, 2000.

[7] S. Baccarin. Optimal impulse control for cash management with quadratic holding-penalty costs. *Decisions in Economics and Finance*, 25(1):19–32, 2002.

[8] G. Barles. *Numerical Methods in Finance*, chapter Convergence of Numerical Schemes for Degenerate Parabolic Equations Arising in Finance Theory, pages 1–21. Cambridge University Press, 1997.

[9] G. Barles, C. Daher, and M. Romano. Convergence of numerical schemes for parabolic equations arising in finance theory. *Mathematical Models and Methods in Applied Sciences*, 5(1):125–143, 1995.

[10] G. Barles and E. Jakobsen. Error bounds for monotone approximation schemes for parabolic Hamilton-Jacobi-Bellman equations. *Mathematics of Computation*, 76:1861–1893, 2007.

[11] G. Barles and P. E. Souganidis. Convergence of approximation schemes for fully nonlinear second order equations. *Asymptotic Analysis*, 4:271–283, 1991.

[12] D. Bauer, A. Kling, and J. Russ. A universal pricing framework for guaranteed minimum benefits in variable annuities. Working paper, Ulm University, 2006.

[13] A. C. Bélanger and P. A. Forsyth. Infinite reload options: Pricing and analysis. To appear in the *Journal of Computational and Applied Mathematics*.

[14] A. Bensoussan and J.-L. Lions. *Impulse Control and Quasi-Variational Inequalities*. Gauthier-Villars, 1984.

[15] N. P. B. Bollen. Valuing options in regime-switching models. *Journal of Derivatives*, 6(1):38–49, 1998.

[16] P. Boyle and T. Draviam. Pricing exotic options under regime switching. *Insurance: Mathematics and Economics*, 40(2):267–282, 2007.

[17] I. R. C. Buckley and R. Korn. Optimal index tracking under transaction costs and

impulse control. *International Journal of Theoretical and Applied Finance*, 1:315–330, 1998.

[18] J. Buffington and R. J. Elliott. American options with regime-switching. *International Journal of Theoretical and Applied Finance*, 5(5):497–514, 2002.

[19] A. Cadenillas, S. Sarkar, and F. Zapatero. Optimal dividend policy with mean-reverting cash reservoir. *Mathematical Finance*, 17(1):81–109, 2007.

[20] A. Cadenillas and F. Zapatero. Optimal central bank intervention in the foreign exchange market. *Journal of Economic Theory*, 87:218–242, 1999.

[21] A. Cadenillas and F. Zapatero. Classical and impulse stochastic control of the exchange rate using interest rates and reserves. *Mathematical Finance*, 10:141–156, 2000.

[22] M. Cao and J. Wei. Incentive stock and options with trading restrictions - Not as restricted as we thought. To appear in *Research in Finance*, 2007.

[23] J. N. Carpenter. The exercise and valuation of executive stock options. *Journal of Financial Economics*, 48:127–158, 1998.

[24] J.-P. Chancelier, B. Øksendal, and A. Sulem. Combined stochastic control and optimal stopping, and application to numerical approximation of combined stochastic and impulse control. *Proceedings of the Steklov Institute of Mathematics*, 237:140–163, 2002.

[25] 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). Working paper, Cheriton School of Computer Science, University of Waterloo, 2007.

[26] T. Cheuk and T. Vorst. Currency lookback options and observation frequency: A binomial approach. *Journal of International Money and Finance*, 16(2):173–187, 1997.

[27] S. Choi. Regime-switching univariate diffusion models of the short-term interest rate. Working paper, School of Economics, University of Adelaide, Australia, 2004.

[28] T. F. Coleman, Y. Li, and M. Patron. Hedging guarantees in variable annuities (under both market and interest rate risks). *Insurance: Mathematics and Economics*, 38:215–228, 2006.

[29] Collection of Life Tables - Receuil des tables de mortalité canadiennes - 1801–1996. CIED (Centre interuniversitaire d'études démographiques), Department of demography, University of Montreal, June 2003.

[30] 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, 2007.

[31] M. G. Crandall, H. Ishii, and P.-L. Lions. User's guide to viscosity solutions of second order differential equations. *Bulletin of the American Mathematical Society*, 27:1–67, July 1992.

[32] M. Dai and Y. Kwok. A tale of two options: Employee reload options and shout call options. Working paper, Hong Kong University of Science and Technology, 2004.

[33] M. Dai and Y. Kwok. Valuing employee reload options under time vesting requirement. *Quantitative Finance*, 5(1):61–69, 2005.

[34] M. Dai, Y. K. Kwok, and J. Zong. Guaranteed minimum withdrawal benefit in variable annuities. To appear in *Mathematical Finance*, 2008.

[35] 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.

[36] Y. d'Halluin, P. A. Forsyth, and G. Labahn. A semi-Lagrangian Approach for American Asian options under jump diffusion. *SIAM Journal on Scientific Computation*, 27:315–345, 2005.

[37] J. Douglas and T. F. Russell. Numerical methods for convection-dominated diffusion problems based on combining the method of characteristics with finite element or finite difference procedures. *SIAM Journal on Numerical Analysis*, 19(5):871–885, 1982.

[38] J.-C. Duan, I. Popova, and P. Ritchken. Option pricing under regime-switching. *Quantitative Finance*, 2:116–132, 2002.

[39] P. H. Dybvig and M. Loewenstein. Employee reload options: Pricing, hedging, and optimal exercise. *Review of Financial Studies*, 16:145–171, 2003.

[40] J. Eastham and K. Hastings. Optimal impulse control of portfolios. *Mathematics of Operations Research*, 13(4):588–605, 1988.

[41] M. Falcone and R. Ferretti. Convergence analysis for a class of high-order semi-Lagrangian advection schemes. *SIAM Journal on Numerical Analysis*, 35(3):909–940, 1998.

[42] W. H. Fleming and R. W. Rishel. *Deterministic and Stochastic Optimal Control*, volume 1 of *Applications of Mathematics*. Springer-Verlag, New York, 1982.

[43] P. A. Forsyth and G. Labahn. Numerical methods for controlled Hamilton-Jacobi-Bellman PDEs in finance. *Journal of Computational Finance*, 11(2):1–44, Winter 2007/2008.

[44] P. A. Forsyth and K. R. Vetzal. Quadratic convergence for valuing American options using a penalty method. *SIAM Journal on Scientific Computing*, 23:2095–2122, 2002.

[45] C. Frantz, X. Chenut, and J. Walhin. Pricing and capital allocation for unit-linked life insurance contracts with minimum death guarantee. *Proceedings of the AFIR Colloquium*, 2003.

[46] J. Gatheral. *The Volatility Surface: A Practitioner's Guide*. Wiley, 2006.

[47] M. B. Giles and R. Carter. Convergence analysis of Crank-Nicolson and Rannacher time-marching. *Journal of Computational Finance*, 9(4):89–112, Summer 2006.

[48] P. Glasserman. *Monte Carlo Methods in Financial Engineering*, volume 53 of *Stochastic Modelling and Applied Probability*. Springer-Verlag, New York, 2003.

[49] S. F. Gray. Modeling the conditional distribution of interest rates as a regime-switching process. *Journal of Financial Economics*, 42:27–62, 1996.

[50] B. J. Hall and K. J. Murphy. Stock options for undiversified executives. *Journal of Accounting and Economics*, 33:3–42, 2002.

[51] B. J. Hall and K. J. Murphy. The trouble with stock options. *Journal of Economic Perspectives*, 17(3):49–70, 2003.

[52] J. D. Hamilton. A new approach to the economic analysis of non-stationary time series. *Econometrica*, 57:357–384, 1989.

[53] M. R. Hardy. A regime-switching model of long-term stock returns. *North American Actuarial Journal*, 5(2):41–53, 2001.

[54] J. C. Hull. *Options, Futures, and Other Derivatives*. Prentice Hall, Upper Saddle River, NJ, 5th edition, 2003.

[55] S. Ikonen and J. Toivanen. Operator splitting methods for American option pricing. *Applied Mathematics Letters*, 17:809–814, 2004.

[56] S. Ikonen and J. Toivanen. Operator splitting methods for pricing American options with stochastic volatility. Technical Report B 11/2004, University of Jyväskylä, 2004.

[57] K. Ishii. Viscosity solutions of nonlinear second order elliptic PDEs associated with impulse control problems II. *Funkcialaj Ekvacioj*, 38:297–328, 1995.

[58] M. Jeanblanc-Picqué and A. N. Shiryaev. Optimization of the flow of dividends. *Russian Mathematical Reviews*, 50:257–277, 1995.

[59] Unitrin Inc. Joe Nedumgottil, Director of Equity Compensation. Private communication.

[60] J. E. Ingersoll Jr. The subjective and objective evaluation of compensation stock options. *Journal of Business*, 79(2):453–487, 2006.

[61] J. E. Ingersoll Jr. Valuing reload options. *Review of Derivatives Research*, 9(1):67–105, 2006.

[62] R. Korn. Portfolio optimization with strictly positive transaction costs and impulse controls. *Finance and Stochastics*, 2:85–114, 1998.

[63] R. Korn. Some applications of impulse control in mathematical finance. *Mathematical Methods of Operations Research*, 50:493–518, 1999.

[64] K. W. Lau and Y. K. Kwok. Valuation of employee reload options in utility maximization framework. *International Journal of Theoretical and Applied Finance*, 8(5):659–674, 2005.

[65] K. Lewellen. Financing decisions when managers are risk averse. Technical Report 4438-03, MIT Sloan School of Management, September 2003.

[66] H. Liu. Optimal consumption and investment with transaction costs and multiple risky assets. *Journal of Finance*, 59:289–338, 2004.

[67] W. Liu, Y. Yang, and G. Lu. Viscosity solutions of fully nonlinear parabolic systems. *Journal of Mathematical Analysis and Applications*, 281:362–381, 2002.

[68] R. C. Merton. Theory of rational option pricing. *Bell Journal of Economics and Management Science*, 4:141–183, 1973.

[69] R. C. Merton. Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics*, 3:125–144, 1976.

[70] L. K. Meulbroek. The efficiency of equity-linked compensation: Understanding the full cost of awarding executive stock options. Harvard Business School Working Paper No. 00-056, 2000.

[71] M. Milevsky and S. Posner. Option-adjusted equilibrium valuation of guaranteed minimum death benefits in variable annuities. Working paper, York University, June 1999.

[72] M. Milevsky and S. Posner. The Titanic Option: Valuation of the guaranteed minimum death benefit in variable annuities and mutual funds. *Journal of Risk and Insurance*, 68(1):91–126, 2001.

[73] M. A. Milevsky and T. S. Salisbury. Financial valuation of guaranteed minimum withdrawal benefits. *Insurance: Mathematics and Economics*, 38(1):21–38, 2006.

[74] B. Mudavanhu and J. Zhuo. Valuing guaranteed minimum death benefits in variable annuities and the option to lapse. Submitted to the North American Actuarial Journal, March 2002.

[75] B. Øksendal. *Stochastic Differential Equations*. Springer-Verlag, third edition, 1992.

[76] B. Øksendal and A. Sulem. Optimal consumption and portfolio with both fixed and proportional transaction costs. *SIAM Journal on Control and Optimization*, 40(6):1765–1790, 2002.

[77] H. Panjer, editor. *Financial Economics: with applications to Investments, Insurance and Pensions*. The Actuarial Foundation, Schaumburg, Illinois, 1998.

[78] M. Pemy and Q. Zhang. Optimal stock liquidation in a regime switching model with finite time horizon. *Journal of Mathematical Analysis and Applications*, 321:537–552, 2006.

[79] P. M. Philipps. Senior Vice President and Director, CIGNA Reinsurance. Private communication, 2004.

[80] O. Pironneau. On the transport-diffusion algorithm and its applications to the Navier-Stokes equations. *Numerische Mathematik*, 38(3):309–332, 1982.

[81] 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.

[82] R. Rannacher. Finite element solution of diffusion problems with irregular data. *Numerische Mathematik*, 43:309–327, 1984.

[83] Y. S. Tian. Too much of a good incentive? The case of executive stock options. *Journal of Banking and Finance*, 28:1225–1245, 2004.

[84] V. L. Vath, M. Mnif, and H. Pham. A model of optimal portfolio selection under liquidity risk and price impact. *Finance and Stochastics*, 11:51–90, 2007.

[85] P. Wilmott. *Derivatives: The Theory and Practice of Financial Engineering*. Wiley, West Sussex, England, 1998.

[86] H. Windcliff. *Computational methods for valuing path-dependent derivatives*. PhD thesis, University of Waterloo, 2003.

[87] H. Windcliff, P. A. Forsyth, and K. R. Vetzal. Shout options: A framework for valuing securities which can be modified by the investor. *Journal of Computational and Applied Mathematics*, 134:213–241, 2001.

[88] H. Windcliff, P. A. Forsyth, and K. R. Vetzal. Valuation of segregated funds: Shout options with maturity extensions. *Insurance: Mathematics and Economics*, 29(1):1–21, 2001.

[89] H. Windcliff, P. A. Forsyth, and K. R. Vetzal. Analysis of the stability of the linear boundary condition for the Black-Scholes equation. *Journal of Computational Finance*, 8(1):65–92, 2004.

[90] H. Windcliff, P. A. Forsyth, and K. R. Vetzal. Numerical methods and volatility models for valuing cliquet options. *Applied Mathematical Finance*, 13:353–386, 2006.

[91] S. Wu and Y. Zeng. A general equilibrium model of the term structure of interest rates under regime-switching risk. *International Journal of Theoretical and Applied Finance*, 8(7):839–869, 2005.

[92] D. D. Yao, Q. Zhang, and X. Y. Zhou. *A regime-switching model for European Option Pricing*, chapter Stochastic Processes, Optimization, and Control Theory: Applications in Financial Engineering, Queuing Networks, and Manufacturing Systems, pages 281–300. Springer US, 2006.

[93] V. I. Zakamouline. A unified approach to portfolio optimization with linear transaction costs. *Mathematical Methods of Operations Research*, 62(2):319–343, 2005.