Infinite Reload Options: Pricing and Analysis

A. C. Bélanger* P. A. Forsyth†

April 27, 2006

Abstract

Infinite reload options allow the user to exercise his reload right as often as he chooses during the lifetime of the contract. Each time a reload occurs, the owner receives new options where the strike price is set to the current stock price. We consider a modified version of the infinite reload option contract where the strike price of the new options received by the owner is increased by a certain percentage; we refer to this new contract as an increased reload option. The pricing problem for this modified contract is characterized as an impulse control problem resulting in a Hamilton-Jacobi-Bellman equation. We use fully implicit timestepping and prove that the discretized equations are monotone, stable and consistent, implying convergence to the viscosity solution. We also derive a globally convergent iterative method for solving the non-linear discrete equations. Numerical examples show that both the exercise policy and the option value are very sensitive to the percentage increase in the reload strike.

Keywords: Infinite reload options, impulse control problem, viscosity solution, optimal exercise, implicit constraint

1 Introduction

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. As such, companies are looking to establish the *fair* or no-arbitrage value of these contracts using numerical techniques. In addition, companies who have issued more exotic, and hence 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.

In this paper, we will consider a particularly expensive type of employee stock option referred to as a reload option. These contracts include a reload feature which allows the owner to pay

*David Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada, N2L 3G1 (e-mail: acbelang@cs.uwaterloo.ca)

†David Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada, N2L 3G1 (e-mail: paforsyt@cs.uwaterloo.ca)

the current strike price using a certain amount of his company stock and in return receive new options where the strike price is set to the prevailing stock price. In the case of 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. Both [11] and [8] present numerical methods for pricing reload options in the no-arbitrage framework. The authors of [11] use a binomial model (essentially an explicit finite difference method) to price infinite reload options and outline the optimal exercise policy which states that the owner should exercise his option whenever the stock price exceeds the current strike price. Meanwhile, [8] outlines a binomial pricing model for reload contracts with both finite and infinite number of reload opportunities where the reload feature is incorporated using dynamic programming. The work in [9] extends this pricing model to take into consideration possible time vesting requirements.

Numerous companies that have issued infinite reload options are now looking for ways to reduce their no-arbitrage price [15]. One particular contractual change which has been considered by some companies [15] is to increase the strike price of new options received following a reload event by a certain percentage. We will refer to this modified contract as an increased reload option and will demonstrate how this contract modification can reduce the option expense.

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

- The increased reload pricing problem is outlined and characterized as an impulse control problem [5], which results in a Hamilton-Jacobi-Bellman variational inequality. Note that 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. We show that the discretized Hamilton-Jacobi-Bellman equations satisfy the classic stability, monotonicity and consistency requirements as outlined in [2].

- Furthermore, the application method of the reload constraint is considered. While a penalty term is used to impose the reload constraint, we demonstrate how applying the reload constraint implicitly results in more accurate results than applying the constraint explicitly. Note that previous work on reload options involved applying the constraint explicitly [11, 8].

- In addition, we show that both the option value and the optimal exercise policy are highly sensitive to the increased percentage 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.

This paper is structured as follows. Section 2 presents the pricing model for increased reload options while Section 3 outlines some of the analytical properties of the associated discrete equations. Some crucial algorithmic solution details are also outlined. Section 4 then presents numerical results obtained when pricing different increased reload option contracts including infinite reload options. Finally, Section 5 summarizes our findings and presents concluding remarks.

2 Increased Reload Pricing Problem

One of the main goals of this paper is to investigate how a particular contract modification can reduce the no-arbitrage price of infinite reload options. In the case of classic infinite reload options, the following exchange takes place each time a reload occurs: the owner will pay the current strike price K using K/S pre-owned company shares and, in return, will receive one unit of company stock and K/S new reload options where the strike price is set to the current stock price $K = S$. The contractual change considered by some companies [15] 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. We will refer to this modified reload option contract as an increased reload option. 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 contract $V = V(S, K, t)$ will depend on the current company stock price S, the option strike price K and time t. We assume that the company stock price S follows geometric Brownian motion, 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, $\sigma(S, K, t)$ is the volatility of the company stock and dZ is the increment in a Wiener process. Note that the asset volatility 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.

At maturity of the contract ($t = T$), the owner will receive one unit of stock for each increased reload option owned, which he can then sell at market value. Hence, the option payoff received by the employee at expiry is:

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

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 $V^* = V^*(S, K, t)$ 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. Keep in mind that the owner of an increased reload option will only consider reloading if $S > K$. Based on the dynamics of the contract, the increased reload constraint V^* is defined as:

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

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 as stated in [8] and [11] is recovered by setting $p = 0$ in equation (2.3).

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 no-arbitrage value of the increased reload option can be stated as [8, 11]:

$$\min\Big(V_\tau - \mathcal{L}V, V - V^*\Big) = 0, \tag{2.5}$$

where V^* is the constraint 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 the penalized problem:

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

This pricing problem will be solved numerically using the penalty method outlined in [12]. Note that [1] demonstrates that the penalization method is a good viscosity approximation for problems such as that presented in equation (2.6). Well-posedness properties are also outlined in [1].

When option values $V(S, K, t)$ are homogeneous of degree one in both S and K, the increased reload constraint can be simplified by using the following property [14]:

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

for a given value of λ. Setting $\lambda = \frac{K^*}{K}$, we can write:

$$V(S, K, t) = \frac{K}{K^*} V\left(\frac{SK^*}{K}, K^*, t \right). \tag{2.8}$$

In the special case where $K = S(1 + p)$, we then obtain:

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

Assuming $K^* = K$, we can now simplify the reload constraint in equation (2.3) as follows:

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

Therefore, in cases where this similarity reduction can be applied, the constraint in equation (2.10) can be used when solving equation (2.6). 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) only [19]. Since this solution method is only applicable when homogeneity conditions are met, it will be treated as a special case of the general increased reload pricing problem.

Theoretically, the increased reload pricing problem as outlined in equation (2.6) should be solved on an unbounded two-dimensional domain. In the context of this paper, we will however be considering this particular pricing problem on a truncated rectangular two-dimensional $S \times K$ domain: $[0, S_{max}] \times [0, K_{max}]$ where $S_{max} >> K_{max}$.

2.1 Boundary Conditions

To fully define this problem, we need to 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:

$$V_\tau + rV - \frac{1}{\epsilon} \max(V^* - V, 0) = 0. \tag{2.11}$$

Similarly, we note that 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 will apply the following boundary condition:

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

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 simply solve:

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

and the reload constraint in equation (2.3) becomes:

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

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 amounts to assuming that $\sigma(S, K, \tau)$ becomes constant as $S \to K_{max}$. Making this assumption, the solution for $S > K_{max}$ can be approximated by the similarity solution with little error provided K_{max} is selected sufficiently large.

More precisely, we will modify the reload constraint in equation (2.3) to become:

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

To summarize, we solve the following equation:

$$V_\tau - \mathcal{L}V - \frac{1}{\epsilon} \max(V^* - V, 0) = 0 \tag{2.16}$$

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

$$V(S, K, \tau = 0) = \max(S - K, 0), \tag{2.17}$$

and boundary conditions:

$$V_\tau + rV = \frac{1}{\epsilon} \max(V^* - V, 0) \quad \text{for } S = 0, \tag{2.18}$$

$$V = \max(\text{Payoff}(S, K), V^*) \quad \text{for } S = S_{max}, \tag{2.19}$$

where V^* is given by equation (2.15). This fully specifies our option pricing problem. Note that for finite (S_{max}, K_{max}), this is clearly an approximation to the original pricing problem on $[0, \infty] \times [0, \infty]$. We will verify through numerical experiments that the error due to finite K_{max} is easily reduced to negligible values (see Appendix B).

We now study the properties of the discrete equations in the context of the increased reload pricing problem and show that the numerical scheme obtained is stable, monotone and consistent.

3 Analysis of the Discrete Equations

Having described the increased reload pricing problem in Section 2, we now consider the discretization of equation (2.16) on our $[0, S_{max}] \times [0, K_{max}]$ domain. We first describe how the underlying grid is built and then carry out an analysis of the discrete equations. We will check that the discrete scheme is consistent, stable and monotone. As outlined in [2], these three properties ensure that convergence to the viscosity solution [7] is possible provided a strong comparison result applies. Indeed, [16] demonstrates how some reasonable discretization schemes either never converge or converge to the wrong solution if these properties aren't satisfied.

Let us first point out a few details about the mesh construction used for our discrete $[0, S_{max}] \times [0, K_{max}]$ domain. Since equation (2.4) contains no derivatives with respect to K, we can discretize equation (2.5) using a set of one-dimensional grids. Let K^* be the initial strike price. We first build a set of nodes in the K direction $\{K_j\}$ for $j = 0, \ldots, j_{max}$ such that there there exists an index l where $K_l = K^*$.

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

$$\begin{aligned} S_i^j &= \frac{K_j}{K^*}\frac{K_i}{(1+p)} \quad \text{for } i = 0, \ldots, j_{max} - 1 \\ S_{j_{max}}^j &= \frac{K_{j_{max}}}{K^*}\frac{K_{j_{max}}}{(1+p)}. \end{aligned} \tag{3.1}$$

Note that for any given j, the node $\left(\frac{K_j}{1+p}, K_j\right)$ is included in the grid.

As shown in Figure 3.1, this type of grid concentrates nodes near the line $K = (1+p)S$ so that the constraint can be accurately estimated using equation (2.15). As we shall see, this type of *scaled grid* in general requires an interpolation to estimate V^*. This contrasts with the simple idea of defining:

$$S_i^j = K_i \quad \text{for } i = 0, \ldots, j_{max} \tag{3.2}$$

for a given K_j which results in the so-called *repeated grids* discussed in [20] and [21]. In this case, no interpolation is required to estimate V^*. However, tests in [21] show that a scaled grid, along with diagonal interpolation (to be discussed later) is superior to a repeated grid.

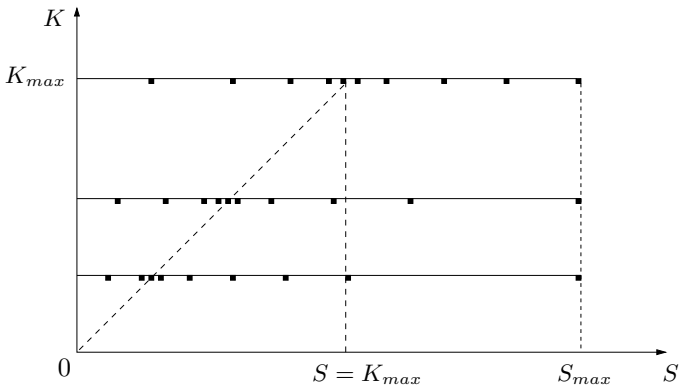


Figure 3.1: Example of a scaled grid construction for the two-dimensional $[0, S_{max}] \times [0, K_{max}]$ domain.

We will now move on to the analysis of the discrete equations for the increased reload option pricing problem. To outline the dependency of the option value on S, K and τ, the following notation will be used for the option value in the discrete domain: $V_{i,j}^n = V(S_i^j, K_j, \tau^n)$. The discrete form of equation (2.6) is obtained by using finite difference approximations and introducing a discrete penalty term $P(V_{i,j}^{n+1}, (V_{i,j}^*)^{n+1})$:

$$\frac{V_{i,j}^{n+1} - V_{i,j}^n}{\Delta\tau} = (1-\theta)[\mathcal{L}V]_{i,j}^{n+1} + \theta[\mathcal{L}V]_{i,j}^n + P(V_{i,j}^{n+1}, (V_{i,j}^*)^{n+1}), \tag{3.3}$$

where $0 \le \theta \le 1$ and $(V_{i,j}^*)^{n+1} = V^*(S_i^j, K_j, \tau^{n+1})$ is the discrete form of the reload constraint. The discrete form of the differential operator $\mathcal{L}$ is defined as:

$$[\mathcal{L}V]_{i,j}^n = \alpha_{i,j}^n V_{i-1,j}^n + \beta_{i,j}^n V_{i+1,j}^n - (\alpha_{i,j}^n + \beta_{i,j}^n + r)V_{i,j}^n, \tag{3.4}$$

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

$$\alpha_{i,j}^n \ge 0 \ ; \ \beta_{i,j}^n \ge 0 \quad \forall i, j, n. \tag{3.5}$$

Consequently, for a given value of S_i^j and K_j, the discrete equation is:

$$\begin{aligned}\frac{V_{i,j}^{n+1} - V_{i,j}^n}{\Delta\tau} &= -(1-\theta)(\alpha_{i,j}^{n+1} + \beta_{i,j}^{n+1} + r)V_{i,j}^{n+1} - \theta(\alpha_{i,j}^n + \beta_{i,j}^n + r)V_{i,j}^n \\ &\quad + (1-\theta)(\alpha_{i,j}^{n+1}V_{i-1,j}^{n+1} + \beta_{i,j}^{n+1}V_{i+1,j}^{n+1}) + \theta(\alpha_{i,j}^n V_{i-1,j}^n + \beta_{i,j}^n V_{i+1,j}^n) + P(V_{i,j}^{n+1}, (V_{i,j}^*)^{n+1}).\end{aligned} \tag{3.6}$$

Note that $\theta = 0$ implies that a fully implicit method is chosen while $\theta = 1/2$ implies that Crank-Nicolson timestepping is used.

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

$$P(V_{i,j}^{n+1}, (V_{i,j}^*)^{n+1}) = L_{i,j}^{n+1}\left[(V_{i,j}^*)^{n+1} - V_{i,j}^{n+1}\right], \tag{3.7}$$

where

$$L_{i,j}^{n+1} = L(V_{i,j}^{n+1}, (V_{i,j}^*)^{n+1}) = \begin{cases} \frac{1}{\epsilon} & \text{if } (V_{i,j}^*)^{n+1} > V_{i,j}^{n+1} \text{ and } S_i^j > K_j \\ 0 & \text{otherwise.} \end{cases} \tag{3.8}$$

Note that we can also write equation (3.7) as:

$$P(V_{i,j}^{n+1}, (V_{i,j}^*)^{n+1}) = \max_{\gamma \in \{0,1\}} \frac{\gamma}{\epsilon}\left[(V_{i,j}^*)^{n+1} - V_{i,j}^{n+1}\right] H\left(S_i^j - K_j\right), \tag{3.9}$$

where

$$H(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{3.10}$$

Writing the penalty term as a control term, as done in equation (3.9), is sometimes useful for carrying out analysis of the discrete equations.

In calculating the constraint $(V^*_{i,j})^{n+1}$ in equation (3.7), we will be using diagonal interpolation along the $K = S(1+p)$ line to determine $V(S, S(1+p), \tau)$ in equation (2.15). Having determined m such that $K_m \leq S_i^j(1+p) \leq K_{m+1}$, we use diagonal interpolation:

$$\begin{aligned} V(S_i^j, S_i^j(1+p), \tau^{n+1}) =& V\left(\frac{K_m}{1+p}, K_m, \tau^{n+1}\right)\left(1 - \frac{S_i^j(1+p) - K_m}{K_{m+1} - K_m}\right) \\ &+ V\left(\frac{K_{m+1}}{(1+p)}, K_{m+1}, \tau^{n+1}\right)\frac{S_i^j(1+p) - K_m}{K_{m+1} - K_m} \\ &+ \mathcal{O}((K_{m+1} - K_m)^2). \end{aligned} \tag{3.11}$$

Tests in [21] show that diagonal interpolation for shout options is superior to the usual bilinear interpolation. Defining the interpolation weight $0 \leq \omega \leq 1$ as:

$$\omega = \frac{S_i^j(1+p) - K_m}{K_{m+1} - K_m}, \tag{3.12}$$

equation (3.11) can be written as:

$$V(S_i^j, S_i^j(1+p), \tau^{n+1}) \simeq V_{l,m}^{n+1}(1-\omega) + V_{l,m+1}^{n+1}\omega, \tag{3.13}$$

where l is an index such that $S_l^j = K_j/(1+p)$ and $V_{l,j}^{n+1} = V(S_l^j, K_j, \tau^{n+1})$. Figure 3.2 shows a graphical representation of diagonal interpolation along the $K = S(1+p)$ line.

Consequently, the discrete penalty term can be written as:

$$P(V_{i,j}^{n+1}, (V_{i,j}^*)^{n+1}) = \begin{cases} \frac{1}{\epsilon}\left[(V_{i,j}^*)^{n+1} - V_{i,j}^{n+1}\right] & \text{when } (V_{i,j}^*)^{n+1} > V_{i,j}^{n+1} \\ 0 & \text{otherwise,} \end{cases} \tag{3.14}$$

where

$$(V_{i,j}^*)^{n+1} = S_i^j - K_j + \frac{K_j}{S_i^j}\left((1-\omega)V_{l,m}^{n+1} + \omega V_{l,m+1}^{n+1}\right). \tag{3.15}$$

Recall that the grid construction ensures that the node S_l^j is included in the one-dimensional grid built for K_j (see equation (3.1)).

In situations where the similarity reduction is applicable, the discrete penalty term $P(V_{i,j}^{n+1}, (V_{i,j}^*)^{n+1})$ will still be of the general form presented in equation (3.14) but $(V_{i,j}^*)^{n+1}$ will be (from equation (2.10)):

$$(V_{i,j}^*)^{n+1} = S_i^j - K^j + (1+p)V_{l,j}^{n+1}. \tag{3.16}$$

Note that no interpolation is required in this case since the data point (S_l^j, K_j) is included in our initial grid.

We will show that the discrete equation in (3.6) satisfies the stability, monotonicity and consistency requirements which generally lead to the convergence of the numerical solution to the unique viscosity solution as shown in [4], [3] and [2].

Note that in addition to satisfying these three properties, the original problem in (2.5) must also satisfy the strong comparison result [2] to conclude that the numerical scheme converges to

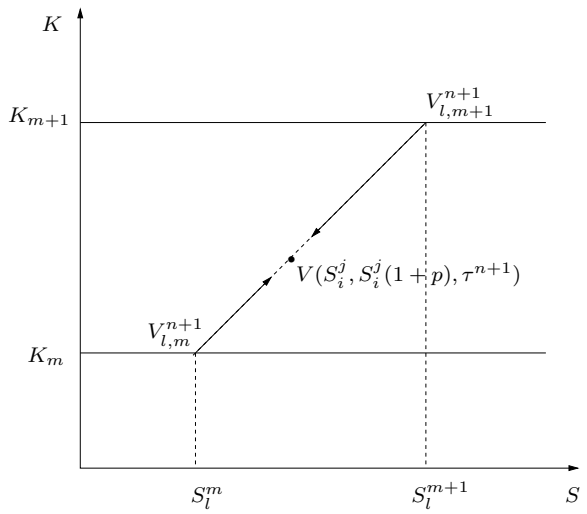


Figure 3.2: Diagonal interpolation is used when determining $V(S_i^j, S_i^j(1+p), \tau^{n+1})$ in the infinite reload constraint presented as equation (2.15).

the unique viscosity solution. 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 $(V^*_{i,j})^{n+1}$ as defined in equations (3.15) and (3.16). However, the authors of [18] consider a quasi-variational Hamilton-Jacobi-Bellman inequality, similar to our pricing problem in equation (2.5), with a non-local impulse operator. Indeed, the authors of [18] study a portfolio optimization problem with a non-linear impulse transaction function. The general equation for the value function is quite similar to our pricing problem presented in equation (2.5). The authors show that the solution of this control problem is a constrained viscosity solution as introduced in [17] which satisfies a comparison property. Similarly, [6] presents an iterative method for solving quasi-variational inequalities (with a non-local impulse operator) after having shown that the solution will converge to the viscosity solution.

While both [6] and [18] indicate that a comparison result is applicable to our pricing problem, we nonetheless need to verify the stability, monotonicity and consistency requirements as done in the following sections.

3.1 Stability

We begin by demonstrating that the discrete equation in (3.6) satisfies the l_∞-stability requirement which involves showing that the discrete option value $V^{n+1}_{i,j}$ is bounded. The following notation will be used when defining the stability requirement:

$$\Delta\tau = \frac{T}{N}, \tag{3.17}$$

$$\Delta S^j = \max_i (S^j_{i+1} - S^j_i), \tag{3.18}$$

where T is the contract maturity.

Definition 3.1 (Stability)**.** *The discretization presented as equation* (3.6) *is* l_∞*-stable if*

$$||V_j^{n+1}||_\infty < C \tag{3.19}$$

for $0 \leq n \leq N$*, as* $\Delta\tau \to 0$ *and* $\max_j \Delta S^j \to 0$*, where* C *is a constant independent of* $\Delta\tau$ *and* ΔS^j*.*

The stability of the discrete scheme in (3.6) will be a consequence of the following Lemma.

Lemma 3.2 (Bound for $V_{i,j}^{n+1}$)**.** *Assuming that the numerical scheme satisfies the positive coefficient condition presented in equation* (3.5)*, that the boundary conditions are applied as outlined in Section 2.1 and that the initial conditions are given by equation* (2.2)*, the value of the option contract will always satisfy:*

$$0 \leq V_{i,j}^n \leq S_i^j \quad \forall i, j, n, \tag{3.20}$$

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

Proof. For a given $j \in \{0, \ldots, j_{max}\}$, the discrete scheme as presented in equation (3.6) can be written out as (for fully implicit timestepping , i.e. $\theta = 0$):

$$\begin{aligned} V_{i,j}^{n+1} = V_{i,j}^n - \Delta\tau(\alpha_{i,j}^{n+1} + \beta_{i,j}^{n+1} + r)V_{i,j}^{n+1} + \Delta\tau\alpha_{i,j}^{n+1}V_{i-1,j}^{n+1} + \Delta\tau\beta_{i,j}^{n+1}V_{i+1,j}^{n+1} \\ + \Delta\tau L_{i,j}^{n+1}\left[S_i^j - V_{i,j}^{n+1} - K_j + \frac{K_j}{S_i^j}\left((1-\omega)V_{l,m}^{n+1} + \omega V_{l,m+1}^{n+1}\right)\right], \end{aligned} \tag{3.21}$$

for all $i < j_{max}$.

For the pricing problem where $j = j_{max}$, it is a simple exercise to show that $V_{i,j_{max}}^{n+1} \geq \max(S_i^{j_{max}} - K_{j_{max}}, 0)$. Consequently, based on the definition of V^* in equation (2.15) and recalling that $S_{max} >> K_{max}$, we have that $V^*(S_{j_{max}}^j, K_j) \geq \max(S_{j_{max}}^j - K_j, 0)$ when $i = j_{max}$, for all j. Consequently, we can write the boundary condition (2.12) in penalized form for $i = j_{max}$ as follows:

$$V_{j_{max},j}^{n+1} = \text{Payoff}(S_{j_{max}}^j, K_j) + \Delta\tau L_{j_{max},j}^{n+1}\left[S_{j_{max}}^j - V_{j_{max},j}^{n+1} - K_j + \frac{K_j}{S_l^{j_{max}}}V_{l,j_{max}}^{n+1}\right], \tag{3.22}$$

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

When the differential operator $\mathcal{L}$ is applied to S, the following equation is satisfied:

$$\mathcal{L}S = qS. \tag{3.23}$$

The discrete form of equation (3.23) for $i < j_{max}$ is:

$$S_i^j = S_i^j(1 + q\Delta\tau) - \Delta\tau(\alpha_{i,j}^{n+1} + \beta_{i,j}^{n+1} + r)S_i^j + \Delta\tau\alpha_{i,j}^{n+1}S_{i-1}^j + \Delta\tau\beta_{i,j}^{n+1}S_{i+1}^j. \tag{3.24}$$

To facilitate our demonstration, we rewrite equation (3.24) so that it has a similar form to equation (3.21):

$$\begin{aligned} S_i^j = S_i^j(1 + q\Delta\tau) - \Delta\tau(\alpha_{i,j}^{n+1} + \beta_{i,j}^{n+1} + r)S_i^j + \Delta\tau\alpha_{i,j}^{n+1}S_{i-1}^j + \Delta\tau\beta_{i,j}^{n+1}S_{i+1}^j \\ + \Delta\tau L_{i,j}^{n+1}\left[S_i^j - S_i^j - K_j + \frac{K_j}{S_i^j}\left((1-\omega)S_l^m + \omega S_l^{m+1}\right)\right], \end{aligned} \tag{3.25}$$

which follows since $S_i^j = (1-\omega)S_l^m + \omega S_l^{m+1}$.

When $i = j_{max}$, we can also write:

$$S^j_{j_{max}} = S^j_{j_{max}} + \Delta\tau L^{n+1}_{i,j}(S^j_{j_{max}} - S^j_{j_{max}} - K_j + \frac{K_j}{S_l^{j_{max}}} S_l^{j_{max}}). \tag{3.26}$$

When $i < j_{max}$, we now subtract equation (3.21) from (3.25) which leads us to:

$$\begin{aligned} S^j_i - V^{n+1}_{i,j} =& (S^j_i(1+q\Delta\tau) - V^n_{i,j}) - \Delta\tau(\alpha^{n+1}_{i,j} + \beta^{n+1}_{i,j} + r)(S^j_i - V^{n+1}_{i,j}) \\ &+ \Delta\tau\alpha^{n+1}_{i,j}(S^j_{i-1} - V^{n+1}_{i-1,j}) + \Delta\tau\beta^{n+1}_{i,j}(S^j_{i+1} - V^{n+1}_{i+1,j}) \\ &+ \Delta\tau L^{n+1}_{i,j}\left[-(S^j_i - V^{n+1}_{i,j}) + \frac{K_j}{S^j_i}\left((1-\omega)(S^m_l - V^{n+1}_{l,m}) + \omega(S^{m+1}_l - V^{n+1}_{l,m+1})\right)\right]. \end{aligned} \tag{3.27}$$

Similarly, when $i = j_{max}$, we can subtract equation (3.22) from (3.26) and obtain:

$$(S^j_{j_{max}} - V^{n+1}_{j_{max},j}) = (S^j_{j_{max}} - \text{Payoff}(S^j_{j_{max}}, K_j)) + \Delta\tau L^{n+1}_{j_{max},j}\left[-(S^j_{j_{max}} - V^{n+1}_{j_{max},j}) + \frac{K_j}{S_l^{j_{max}}}(S_l^{j_{max}} - V^{n+1}_{l,j_{max}})\right]. \tag{3.28}$$

Defining $E^n_{i,j} = S^j_i - V^n_{i,j}$ and $\hat{E}^n_{i,j} = S^j_i(1+q\Delta\tau) - V^n_{i,j}$ for $i < j_{max}$, we can rewrite equation (3.27) as:

$$\begin{aligned} E^{n+1}_{i,j}(1 + \Delta\tau(\alpha^{n+1}_{i,j} + \beta^{n+1}_{i,j} + r + L^{n+1}_{i,j})) - \Delta\tau\alpha^{n+1}_{i,j}E^{n+1}_{i-1,j} - \Delta\tau\beta^{n+1}_{i,j}E^{n+1}_{i+1,j} \\ - \Delta\tau L^{n+1}_{i,j}\left[\frac{K_j}{S^j_i}\left((1-\omega)E^{n+1}_{l,m} + \omega E^{n+1}_{l,m+1}\right)\right] = \hat{E}^n_{i,j}. \end{aligned} \tag{3.29}$$

Similarly, defining $\hat{E}^n_{j_{max},j} = S^j_{j_{max}} - \text{Payoff}(S^j_{j_{max}}, K_j)$ (when $i = j_{max}$), equation (3.28) can be written as:

$$E^{n+1}_{j_{max},j}(1 + \Delta\tau L^{n+1}_{j_{max},j}) - \Delta\tau L^{n+1}_{j_{max},j}\frac{K_j}{S_l^{j_{max}}}E^{n+1}_{l,j_{max}} = \hat{E}^n_{j_{max},j}. \tag{3.30}$$

We now define the following vectors:

$$V^{n+1}_j = \begin{bmatrix} V^{n+1}_{0,j} \\ V^{n+1}_{1,j} \\ \vdots \\ V^{n+1}_{j_{max},j} \end{bmatrix}; \; E^{n+1}_j = \begin{bmatrix} E^{n+1}_{0,j} \\ E^{n+1}_{1,j} \\ \vdots \\ E^{n+1}_{j_{max},j} \end{bmatrix}; \; \hat{E}^{n+1}_j = \begin{bmatrix} \hat{E}^{n+1}_{0,j} \\ \hat{E}^{n+1}_{1,j} \\ \vdots \\ \hat{E}^{n+1}_{j_{max},j} \end{bmatrix} \tag{3.31}$$

and further define:

$$V^{n+1} = \begin{bmatrix} V^{n+1}_0 \\ V^{n+1}_1 \\ \vdots \\ V^{n+1}_{j_{max}} \end{bmatrix}; \; E^{n+1} = \begin{bmatrix} E^{n+1}_0 \\ E^{n+1}_1 \\ \vdots \\ E^{n+1}_{j_{max}} \end{bmatrix}; \; \hat{E}^{n+1} = \begin{bmatrix} \hat{E}^{n+1}_0 \\ \hat{E}^{n+1}_1 \\ \vdots \\ \hat{E}^{n+1}_{j_{max}} \end{bmatrix}. \tag{3.32}$$

Using the notation outlined in (3.32), equations (3.29) and (3.30) can be combined and written as:

$$\mathcal{Q}^{n+1}E^{n+1} = \hat{E}^n, \tag{3.33}$$

where $\mathcal{Q}^{n+1}$ is a sparse matrix where each row holds the coefficients for the $E_{i,j}^{n+1}$ equation, and the entries are such that:

$$\left[\mathcal{Q}^{n+1}E^{n+1}\right]_{row\ (i,j)} = E_{i,j}^{n+1}(1+\Delta\tau(\alpha_{i,j}^{n+1}+\beta_{i,j}^{n+1}+r+L_{i,j}^{n+1})) - \Delta\tau\alpha_{i,j}^{n+1}E_{i-1,j}^{n+1} - \Delta\tau\beta_{i,j}^{n+1}E_{i+1,j}^{n+1} - \frac{(1-\omega)\Delta\tau L_{i,j}^{n+1}K_j}{S_i^j}E_{l,m}^{n+1} - \frac{\omega\Delta\tau L_{i,j}^{n+1}K_j}{S_i^j}E_{l,m+1}^{n+1}, \tag{3.34}$$

when $i < j_{max}$, and similarly the following is satisfied when $i = j_{max}$:

$$\left[\mathcal{Q}^{n+1}E^{n+1}\right]_{row\ (j_{max},j)} = E_{j_{max},j}^{n+1}(1+\Delta\tau L_{j_{max},j}^{n+1}) - \Delta\tau L_{j_{max},j}^{n+1}\frac{K_j}{S_l^{j_{max}}}E_{l,j_{max}}^{n+1}. \tag{3.35}$$

Note that if $L_{i,j}^{n+1} > 0$, then from equation (3.8), we have that $S_i^j > K_j$. Consequently, taking into consideration equations (3.34) and (3.35), we note the following concerning the matrix $\mathcal{Q}^{n+1}$:

- the diagonal entries in $\mathcal{Q}^{n+1}$ are positive,
- the off-diagonal entries in $\mathcal{Q}^{n+1}$ are non-positive,
- the row sum of the entries in $\mathcal{Q}^{n+1}$ is strictly positive for all rows.

Hence, we conclude that $\mathcal{Q}^{n+1}$ is an M-matrix.

Recalling that we defined $\hat{E}_{i,j}^n = S_i^j(1+q\Delta\tau) - V_{i,j}^n$ for $i < j_{max}$, and $\hat{E}_{j_{max},j}^n = S_{j_{max}}^j -$ Payoff$(S_{j_{max}}^j, K_j)$ when $i = j_{max}$, we note that, since $q \geq 0$:

$$\hat{E}_{i,j}^n \geq S_i^j - V_{i,j}^n \quad \text{for } i < j_{max}\ , \tag{3.36}$$

$$\hat{E}_{j_{max},j}^n \geq 0 \quad \text{for } i = j_{max}\ , \tag{3.37}$$

so that:

$$\hat{E}_{i,j}^n \geq E_{i,j}^n \quad \text{for } i < j_{max},\ j = 0,\ldots,j_{max}\ , \tag{3.38}$$

$$\hat{E}_{j_{max},j}^n \geq 0 \quad \text{for } i = j_{max},\ j = 0,\ldots,j_{max}\ . \tag{3.39}$$

If we assume that $\hat{E}^n \geq 0$, equation (3.33) then implies:

$$E^{n+1} = (\mathcal{Q}^{n+1})^{-1}\hat{E}^n \geq 0, \tag{3.40}$$

since the matrix $\mathcal{Q}^{n+1}$ was shown to be an M-matrix. Combining equations (3.38), (3.39) and (3.40), we also find that $\hat{E}^{n+1} \geq 0$. As $\hat{E}^0 \geq 0$ (a consequence of the initial payoff condition; see equation (2.2)), we find that:

$$V_{i,j}^{n+1} \leq S_i^j \quad \forall i,j,n. \tag{3.41}$$

Using the definition of $\mathcal{Q}^{n+1}$ (see equations (3.34) and (3.35)), equation (3.21) can be rewritten as:

$$\mathcal{Q}^{n+1}V^{n+1} = V^n + A^{n+1}, \tag{3.42}$$

where the vectors A_j^{n+1} and A^{n+1} are defined as:

$$A_j^{n+1} = \begin{bmatrix} \Delta\tau L_{0,j}^{n+1}(S_0^j - K_j) \\ \Delta\tau L_{1,j}^{n+1}(S_1^j - K_j) \\ \vdots \\ \Delta\tau L_{j_{max},j}^{n+1}(S_{j_{max}}^j - K_j) \end{bmatrix} ; A^{n+1} = \begin{bmatrix} A_0^{n+1} \\ A_1^{n+1} \\ \vdots \\ A_{j_{max}}^{n+1} \end{bmatrix}. \tag{3.43}$$

Based on the definition of $L_{i,j}^{n+1}$ in equation (3.8), we see that the entries in A_j^{n+1} are non-negative:

$$\Delta\tau L_{i,j}^{n+1}(S_i^j - K_j) \geq 0 \quad \forall i, j. \tag{3.44}$$

Consequently, since $A_j^{n+1} \geq 0$ for all values of j and n, we have that $A^{n+1} \geq 0$.

Furthermore, if $V_{i,j}^n \geq 0$ for all i, j and n, then equation (3.42) implies the following:

$$V^n + A^{n+1} \geq 0 \rightarrow \mathcal{Q}^{n+1}V^{n+1} \geq 0. \tag{3.45}$$

Since the matrix $\mathcal{Q}^{n+1}$ was shown to be an M-matrix, we have that:

$$V^{n+1} \geq 0 \quad \forall n. \tag{3.46}$$

Since the option value is initially set to the payoff (as defined in equation (2.2)), we have that $V^0 \geq 0$ which implies:

$$V_{i,j}^{n+1} \geq 0 \quad \forall i, j, n. \tag{3.47}$$

Hence, combining this result with equation (3.41) leads us to the following conclusion:

$$0 \leq V_{i,j}^{n+1} \leq S_i^j \text{ for all } i, j, n. \tag{3.48}$$

□

Since $S_i^j \leq S_{max}$, then the discrete numerical scheme satisfies the stability requirement presented as Definition 3.1.

Remark 3.3 (Stability for Crank-Nicolson)**.** *We can extend the above analysis when Crank-Nicolson is used ($\theta = 1/2$) to show that Crank-Nicolson timestepping is l_∞-stable if the following timestepping condition is satisfied:*

$$\Delta\tau \leq \frac{2}{\alpha_{i,j}^n + \beta_{i,j}^n + r} \quad \forall i, j. \tag{3.49}$$

3.2 Consistency

For the purposes of defining consistency in a concise manner, we will outline some notational details. We first let $\Delta K_{max} = \max_j(K_{j+1} - K_j)$, $\Delta S_{max}^j = \max_i(S_{i+1}^j - S_i^j)$ and $\Delta\tau_{max} = \max_n(\tau^{n+1} - \tau^n)$. Furthermore, we consider that ΔK_{max}, ΔS_{max}^j and $\Delta\tau_{max}$ will be parametrized as follows:

$$\Delta K_{max} = Ch_{max} \tag{3.50}$$

$$\Delta S_{max}^j = Dh_{max} \tag{3.51}$$

$$\Delta\tau_{max} = Eh_{max} \tag{3.52}$$

where C, D and E are constants. Finally, we use the following notation for the numerical scheme in equation (3.6):

$$\begin{aligned} g_{i,j}(V_{i,j}^{n+1}, \{V_{k,j}^{n+1}\}_{k\neq i}, \{V_{i,j}^{n}\}, V_{l,m}^{n+1}, V_{l,m+1}^{n+1}) &= V_{i,j}^{n+1} - V_{i,j}^{n} + (1-\theta)\Delta\tau(\alpha_{i,j}^{n+1} + \beta_{i,j}^{n+1} + r)V_{i,j}^{n+1} \\ &+ \theta\Delta\tau(\alpha_{i,j}^{n} + \beta_{i,j}^{n} + r)V_{i,j}^{n} - (1-\theta)\Delta\tau(\alpha_{i,j}^{n+1}V_{i-1,j}^{n+1} + \beta_{i,j}^{n+1}V_{i+1,j}^{n+1}) \\ &- \theta\Delta\tau(\alpha_{i,j}^{n}V_{i-1,j}^{n} + \beta_{i,j}^{n}V_{i+1,j}^{n}) - \Delta\tau L(V_{i,j}^{n+1}, (V_{i,j}^{*})^{n+1})\Big[(V_{i,j}^{*})^{n+1} - V_{i,j}^{n+1}\Big] \\ &= 0 \end{aligned} \tag{3.53}$$

where $0 \leq \theta \leq 1$ and $(V_{i,j}^{*})^{n+1}$ is defined as in equation (3.15) for $i < j_{max}$ and as in equation (3.16) for $i = j_{max}$. Note that $(V_{i,j}^{*})^{n+1} = V_{i,j}^{*}(V_{l,m}^{n+1}, V_{l,m+1}^{n+1})$.

Definition 3.4 (Consistency)**.** *The numerical scheme $g_{i,j}(V_{i,j}^{n+1}, \{V_{k,j}^{n+1}\}_{k\neq i}, \{V_{i,j}^{n}\}, V_{l,m}^{n+1}, V_{l,m+1}^{n+1})$ presented in equation* (3.53) *will be consistent if, for any smooth test function ϕ, where $\phi_{i,j}^{n} = \phi(S_i^j, K_j, \tau^n)$, we have that:*

$$\lim_{h_{max}\to 0}\left|\big(\phi_\tau - \mathcal{L}\phi + P\big)_{i,j}^{n+1} - \frac{1}{\Delta\tau}g_{i,j}\big(\phi_{i,j}^{n+1}, \{\phi_{k,j}^{n+1}\}_{k\neq i}, \{\phi_{i,j}^{n}\}, \phi_{l,m}^{n+1}, \phi_{l,m+1}^{n+1}\big)\right| = 0. \tag{3.54}$$

Lemma 3.5 (Consistent Discretization)**.** *The numerical scheme $g_{i,j}(V_{i,j}^{n+1}, \{V_{k,j}^{n+1}\}_{k\neq i}, \{V_{i,j}^{n}\}, V_{l,m}^{n+1}, V_{l,m+1}^{n+1})$ presented in equation* (3.53) *is consistent according to Definition 3.4.*

Proof. In order to demonstrate that our discretization presented as equation (3.53) is consistent, we will first determine the truncation error of this discretization. We once again carry out our analysis assuming that fully implicit timestepping is used.

Let $\phi(S, \tau)$ denote a smooth function with bounded derivatives of all orders with respect to both S and τ. Using Taylor series and ignoring the penalty term for the moment, we find that:

$$|\mathcal{L}\phi_i^{n+1} - [\mathcal{L}\phi]_i^{n+1}| = \mathcal{O}(\Delta S_{max}^{j}). \tag{3.55}$$

Similarly, the error stemming from the calculation of the infinite reload constraint can be determined. The constraint will generally be obtained as outlined in (3.15) using diagonal interpolation (see equation (3.11)). This will result in an error of the form $\mathcal{O}((\Delta K_{max})^2)$. Since this interpolation occurs when calculating $(V_{i,j}^{*})^{n+1}$ as outlined in equation (3.14), the full error term (for fixed ϵ) will be $\frac{1}{\epsilon}\mathcal{O}((\Delta K_{max})^2)$.

Consequently, including the error from the timestepping method, we have that:

$$\begin{aligned} \left|\big(\phi_\tau - \mathcal{L}\phi + P\big)_{i,j}^{n+1} - \frac{1}{\Delta\tau}g_{i,j}\big(\phi_{i,j}^{n+1}, \{\phi_{k,j}^{n+1}\}_{k\neq i}, \{\phi_{i,j}^{n}\}, \phi_{l,m}^{n+1}, \phi_{l,m+1}^{n+1}\big)\right| = \\ \mathcal{O}(\Delta S_{max}^{j}) + \mathcal{O}(\Delta\tau_{max}) + \mathcal{O}((\Delta K_{max})^2) \end{aligned} \tag{3.56}$$

when fully implicit timestepping is chosen. This enables us to conclude that the numerical scheme presented in equation (3.53) satisfies Definition 3.4 and is consistent. □

Remark 3.6. *In the case when no constraint is active, Crank-Nicolson timestepping is used, and central weighting is active at node (i, j), then the truncation error is locally second order in $\Delta\tau$, ΔS_{max}^{j}.*

3.3 Monotonicity

We now move on to demonstrate that the numerical scheme in equation (3.6) is monotone.

Definition 3.7 (Monotonicity)**.** *The numerical scheme* $g_{i,j}(V_{i,j}^{n+1}, \{V_{k,j}^{n+1}\}_{k\neq i}, V_{i,j}^{n}, V_{l,m}^{n+1}, V_{l,m+1}^{n+1})$ *presented in equation* (3.53) *is monotone if*

$$\begin{aligned} g_{i,j}(V_{i,j}^{n+1},&\{V_{k,j}^{n+1}+\epsilon_{k,j}^{n+1}\}_{k\neq i}, \{V_{i,j}^{n}+\epsilon_{i,j}^{n}\}, V_{l,m}^{n+1}+\epsilon_{l,m}^{n+1}, V_{l,m+1}^{n+1}+\epsilon_{l,m+1}^{n+1}) \\ &- g_{i,j}(V_{i,j}^{n+1}, \{V_{k,j}^{n+1}\}_{k\neq i}, V_{i,j}^{n}, V_{l,m}^{n+1}, V_{l,m+1}^{n+1}) \leq 0 \quad ; \quad \forall \epsilon_{i,j}^{n} \geq 0, \end{aligned} \tag{3.57}$$

and, for any constants $a, a_0 \geq 0$ *we have:*

$$\begin{aligned} g_{i,j}(V_{i,j}^{n+1}+a+a_0 t,&\{V_{k,j}^{n+1}+a+a_0 t\}_{k\neq i}, \{V_{i,j}^{n}+a+a_0 t\}, V_{l,m}^{n+1}+a+a_0 t, V_{l,m+1}^{n+1}+a+a_0 t) \\ &- g_{i,j}(V_{i,j}^{n+1}, \{V_{k,j}^{n+1}\}_{k\neq i}, V_{i,j}^{n}, V_{l,m}^{n+1}, V_{l,m+1}^{n+1}) \geq a_0 \quad ; \quad \forall a, a_0 \geq 0. \end{aligned} \tag{3.58}$$

Note that this definition of monotonicity is equivalent to that presented in [13].

Remark 3.8 (Note on the definition of monotonicity)**.** *Though the monotonicity definition involves both equations* (3.57) *and* (3.58)*, the crucial requirement lies in equation* (3.57)*. As previously noted in [10], when* $a > 0$ *and* $a_0 = 0$*, condition* (3.58) *will be satisfied by any consistent discretization of equation* (2.6) *assuming* $r > 0$*. When* $a = 0$ *and* $a_0 > 0$*, equation* (3.58) *is equivalent to requiring that the discrete equations contain a consistent discretization of* V_τ *in equation* (2.6)*. As observed in [10], equations* (3.57) *and* (3.58) *require* $g_{i,j}(V_{i,j}^{n+1}, \{V_{k,j}^{n+1}\}_{k\neq i}, \{V_{i,j}^{n}\}, V_{l,m}^{n+1}, V_{l,m+1}^{n+1})$ *to be increasing in* $V_{i,j}^{n+1}$ *and non-increasing in* $\{V_{k,j}^{n+1}\}_{k\neq i}$*,* $\{V_{i,j}^{n}\}$*,* $V_{l,m}^{n+1}$ *and* $V_{l,m+1}^{n+1}$*.*

Lemma 3.9 (Monotone Discretization)**.** *Assuming that the discretization satisfies condition* (3.5)*, the numerical scheme* $g_{i,j}(V_{i,j}^{n+1}, \{V_{k,j}^{n+1}\}_{k\neq i}, \{V_{i,j}^{n}\}, V_{l,m}^{n+1}, V_{l,m+1}^{n+1})$*, as presented in equation* (3.53)*, is monotone according to Definition 3.7.*

Proof. We will once again outline the proof in the case where fully implicit timestepping is used ($\theta = 0$). In this case, $g_{i,j}(V_{i,j}^{n+1}, \{V_{k,j}^{n+1}\}_{k\neq i}, V_{i,j}^{n}, V_{l,m}^{n+1}, V_{l,m+1}^{n+1})$ is defined as:

$$\begin{aligned} g_{i,j}\left(V_{i,j}^{n+1}, \{V_{k,j}^{n+1}\}_{k\neq i}, V_{i,j}^{n}, V_{l,m}^{n+1}, V_{l,m+1}^{n+1}\right) &= \left(1+\Delta\tau(\alpha_{i,j}^{n+1}+\beta_{i,j}^{n+1}+r)\right) V_{i,j}^{n+1} \\ &- \Delta\tau\alpha_{i,j}^{n+1} V_{i-1,j}^{n+1} - \Delta\tau\beta_{i,j}^{n+1} V_{i+1,j}^{n+1} - V_{i,j}^{n} \\ &- \Delta\tau L(V_{i,j}^{n+1}, (V_{i,j}^{*})^{n+1})\Big[(V_{i,j}^{*})^{n+1} - V_{i,j}^{n+1}\Big]. \end{aligned} \tag{3.59}$$

Thus, we first perturb $\{V_{k,j}^{n+1}\}_{k\neq i}$ in the numerical scheme in equation (3.59):

$$\begin{aligned} g_{i,j}(V_{i,j}^{n+1},&\{V_{k,j}^{n+1}+\epsilon_{k,j}^{n+1}\}_{k\neq i}, V_{i,j}^{n}, V_{l,m}^{n+1}, V_{l,m+1}^{n+1}) \\ &- g_{i,j}(V_{i,j}^{n+1}, \{V_{k,j}^{n+1}\}_{k\neq i}, V_{i,j}^{n}, V_{l,m}^{n+1}, V_{l,m+1}^{n+1}) = -\Delta\tau\alpha_{i,j}^{n+1}\epsilon_{i-1,j}^{n+1} - \Delta\tau\beta_{i,j}^{n+1}\epsilon_{i+1,j}^{n+1} \leq 0 \end{aligned} \tag{3.60}$$

Next, we perturb $V_{i,j}^{n}$ in equation (3.59) and obtain:

$$\begin{aligned} g_{i,j}(V_{i,j}^{n+1},&\{V_{k,j}^{n+1}\}_{k\neq i}, V_{i,j}^{n}+\epsilon_{i,j}^{n}, V_{l,m}^{n+1}, V_{l,m+1}^{n+1}) \\ &- g_{i,j}(V_{i,j}^{n+1}, \{V_{k,j}^{n+1}\}_{k\neq i}, V_{i,j}^{n}, V_{l,m}^{n+1}, V_{l,m+1}^{n+1}) = -\epsilon_{i,j}^{n} \leq 0 \end{aligned} \tag{3.61}$$

Defining $\epsilon_1 = (1-\omega)\frac{K_j}{S_i^j}\epsilon_{l,m}^{n+1}$ and perturbing $V_{l,m}^{n+1}$ in equation (3.59), we obtain:

$$\begin{aligned} &g_{i,j}(V_{i,j}^{n+1}, \{V_{k,j}^{n+1}\}_{k\neq i}, V_{i,j}^{n}, V_{l,m}^{n+1} + \epsilon_{l,m}^{n+1}, V_{l,m+1}^{n+1}) - g_{i,j}(V_{i,j}^{n+1}, \{V_{k,j}^{n+1}\}_{k\neq i}, V_{i,j}^{n}, V_{l,m}^{n+1}, V_{l,m+1}^{n+1}) \\ &\quad = -\Delta\tau L(V_{i,j}^{n+1}, (V_{i,j}^{*})^{n+1})\epsilon_1 - \Delta\tau\left[L(V_{i,j}^{n+1}, (V_{i,j}^{*})^{n+1} + \epsilon_1) - L(V_{i,j}^{n+1}, (V_{i,j}^{*})^{n+1})\right]((V_{i,j}^{*})^{n+1} + \epsilon_1 - V_{i,j}^{n+1}) \\ &\quad \leq 0 \end{aligned} \tag{3.62}$$

which follows from the definition of $\epsilon_1 \geq 0$ and the definition of $L(V_{i,j}^{n+1}, (V_{i,j}^{*})^{n+1})$ in equation (3.8).

Similarly, we define $\epsilon_2 = \omega\frac{K_j}{S_i^j}\epsilon_{l,m+1}^{n+1}$ and perturb $V_{l,m+1}^{n+1}$ to obtain:

$$\begin{aligned} &g_{i,j}(V_{i,j}^{n+1}, \{V_{k,j}^{n+1}\}_{k\neq i}, V_{i,j}^{n}, V_{d,m}^{n+1}, V_{d,m+1}^{n+1} + \epsilon_{d,m+1}^{n+1}) - g_{i,j}(V_{i,j}^{n+1}, \{V_{k,j}^{n+1}\}_{k\neq i}, V_{i,j}^{n}, V_{d,m}^{n+1}, V_{d,m+1}^{n+1}) \\ &\quad = -\Delta\tau L(V_{i,j}^{n+1}, (V_{i,j}^{*})^{n+1})\epsilon_2 - \Delta\tau\left[L(V_{i,j}^{n+1}, (V_{i,j}^{*})^{n+1} + \epsilon_2) - L(V_{i,j}^{n+1}, (V_{i,j}^{*})^{n+1})\right]((V_{i,j}^{*})^{n+1} + \epsilon_2 - V_{i,j}^{n+1}) \\ &\quad \leq 0 \end{aligned} \tag{3.63}$$

which once again follows from the definition of $\epsilon_2 \geq 0$ and $L(V_{i,j}^{n+1}, (V_{i,j}^{*})^{n+1})$ in equation (3.8).

Finally, since we have previously shown that the discretization is consistent, then condition (3.58) is satisfied.

□

Remark 3.10 (Similarity Reduction Case)**.** *Note that a similar proof of monotonicity can be carried out in the case where a similarity reduction is possible.*

Remark 3.11 (Crank-Nicolson)**.** *We can once again extend the above analysis when Crank-Nicolson is used ($\theta = 1/2$) provided the condition in equation* (3.49) *is satisfied.*

3.4 Solution Algorithm

Before moving on to consider numerical results obtained from the increased reload pricing model, we take a moment to specify additional algorithmic details about the solution process. Recall that at each timestep, we will be solving a set of one-dimensional problems each with a different strike value K_j. More specifically, when the reload constraint is applied implicitly, we will solve the following equation at each timestep (assuming that fully implicit timestepping is used):

$$\mathcal{B}_j^{n+1}V_j^{n+1} = V_j^n + \Delta\tau L_j^{n+1}(V_j^*)^{n+1}, \tag{3.64}$$

where

$$L_j^{n+1} = \begin{bmatrix} L_{0,j}^{n+1} \\ L_{1,j}^{n+1} \\ \vdots \\ L_{j_{max},j}^{n+1} \end{bmatrix}; (V_j^*)^{n+1} = \begin{bmatrix} (V_{0,j}^*)^{n+1} \\ (V_{1,j}^*)^{n+1} \\ \vdots \\ (V_{j_{max},j}^*)^{n+1} \end{bmatrix}, \tag{3.65}$$

$L_{i,j}^{n+1} = L(V_{i,j}^{n+1}, (V_{i,j}^*)^{n+1})$ is defined as in equation (3.8) and $(V_{i,j}^*)^{n+1}$ is defined as in equation (3.15) (or (3.16) in the similarity reduction case). Note that the matrix $\mathcal{B}_j^{n+1}$ is built such that:

$$[\mathcal{B}_j^{n+1}V_j^{n+1}]_i = V_{i,j}^{n+1}(1 + \Delta\tau(\alpha_{i,j}^{n+1} + \beta_{i,j}^{n+1} + r + L_{i,j}^{n+1})) - \Delta\tau\alpha_{i,j}^{n+1}V_{i-1,j}^{n+1} - \Delta\tau\beta_{i,j}^{n+1}V_{i+1,j}^{n+1}, \tag{3.66}$$

for rows where the diagonal interpolation doesn't involve the current pricing problem, namely when $S_i^j > K_{j+1}$.

However, for some points S_i^j near K_j, the diagonal interpolation will involve data from the current pricing problem. Indeed, when $K_j \leq S_i^j \leq K_{j+1}$, the diagonal interpolation as outlined in equation (3.15) 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 (3.13). Hence, the rows in $\mathcal{B}_j^{n+1}$ corresponding to these points will contain an extra entry such that:

$$[\mathcal{B}_j^{n+1}V_j^{n+1}]_i = V_{i,j}^{n+1}(1+\Delta\tau(\alpha_{i,j}^{n+1}+\beta_{i,j}^{n+1}+r+L_{i,j}^{n+1})) - \Delta\tau\alpha_{i,j}^{n+1}V_{i-1,j}^{n+1} - \Delta\tau\beta_{i,j}^{n+1}V_{i+1,j}^{n+1} - \Delta\tau L_{i,j}^{n+1}\frac{K_j}{S_i^j}(1-\omega)V_{l,j}^{n+1}, \tag{3.67}$$

and consequently $(V_{i,j}^*)^{n+1}$ for these points will be defined as:

$$(V_{i,j}^*)^{n+1} = S_i^j - K_j + \frac{K_j}{S_i^j}\omega V_{l,j+1}^{n+1}. \tag{3.68}$$

Since interpolation is used in calculating the constraint $(V_j^*)^{n+1}$, it's value will generally depend on the solution from pricing problems with higher strike values: V_h^{n+1} where $K_h > K_j$. Hence, we will need to solve each of the V_h^{n+1} problems first before proceeding to value V_j^{n+1}. 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 3.69.

Implicit Constraint

For $j = 0, \ldots, j_{max}$
  For $i = 0, \ldots, j_{max}$
    $V_{i,j}^0 = \text{Payoff}(S_i^j, K_j)$
  EndFor i
EndFor j
For $n = 0, \ldots, T/dt$
  For $j = j_{max}, \ldots, 0$
    Solve: $\mathcal{B}_j^{n+1}V_j^{n+1} = V_j^n + \Delta\tau L_j^{n+1}(V_j^*)^{n+1}$
  EndFor j
EndFor n

(3.69)

Having noted that equation (3.64) is non-linear, we will use a non-linear iteration to determine V_j^{n+1} for each j value. We will denote the kth estimate for V_j^{n+1} as $(V_j^{n+1})^k = \bar{V}_j^k$. Similarly, we will define $(L_j^{n+1})^k = \bar{L}_j^k$ and $(\mathcal{B}_j^{n+1})^k = \bar{\mathcal{B}}_j^k$. Algorithm 3.70 outlines the iteration algorithm to determine V_j^{n+1} for a given j value. Note that the convergence tolerance, denoted by tol, is chosen adequately small i.e. $tol << 1$.

Non-Linear Iteration

$\bar{V}_j^0 = V_j^n$

For $k = 0, \ldots,$ until convergence

Solve $\bar{\mathcal{B}}_j^k \bar{V}_j^{k+1} = V_j^n + \Delta\tau \bar{L}_j^k (\bar{V}_j^*)^k$

If $\max_i \left[\dfrac{|\bar{V}_{i,j}^{k+1} - \bar{V}_{i,j}^k|}{\max(1, |V_{i,j}^k|)} \right] < tol$ (3.70)

Stop iteration - Exit For loop

EndFor k

$V_j^{n+1} = \bar{V}_j^{k+1}$

Theorem 3.12 (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 used to solve* (3.64) *is globally convergent.*

Proof. Noting equation (3.9), the proof of Theorem 3.12 follows the same steps as in [12], where the authors prove that the penalty iteration for simple American options is globally convergent. □

Remark 3.13. *Note that in general $\bar{\mathcal{B}}_j^k$ is not a tri-diagonal matrix but $\bar{\mathcal{B}}_j^k \bar{V}_j^{k+1} = V_j^n + \Delta\tau \bar{L}_j^{k+1} (\bar{V}_j^*)^k$ can be easily solved using a direct sparse solver.*

An alternate way of applying the reload constraint in equation (2.15) is to apply the constraint explicitly. In this case, an intermediate solution value $\hat{V}_j^{n+1}$ is determined by solving the following equation:

$$\hat{\mathcal{B}}_j^{n+1} \hat{V}_j^{n+1} = V_j^n, \tag{3.71}$$

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

$$[\hat{\mathcal{B}}_j^{n+1} \hat{V}_j^{n+1}]_i = \hat{V}_{i,j}^{n+1}(1 + \Delta\tau(\alpha_{i,j}^{n+1} + \beta_{i,j}^{n+1} + r)) - \Delta\tau\alpha_{i,j}^{n+1}\hat{V}_{i-1,j}^{n+1} - \Delta\tau\beta_{i,j}^{n+1}\hat{V}_{i+1,j}^{n+1}. \tag{3.72}$$

Note that the matrix $\hat{\mathcal{B}}_j^{n+1}$ is a tri-diagonal matrix and that the resulting system for a given j value is now linear. As such, no iteration is required when solving equation (3.71).

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

$$V_{i,j}^{n+1} = \max((V_{i,j}^*)^{n+1}, \hat{V}_{i,j}^{n+1}), \tag{3.73}$$

where $(V_{i,j}^*)^{n+1}$ is defined as:

$$(V_{i,j}^*)^{n+1} = S_i^j - K_j + \frac{K_j}{S_i^j}\left[(1-\omega)\bar{V}_{l,m}^{n+1} + \omega V_{l,m+1}^{n+1}\right], \tag{3.74}$$

with

$$\bar{V}_{l,m}^{n+1} = \begin{cases} V_{l,m}^{n+1} & \text{if } m \neq j \\ \hat{V}_{l,m}^{n+1} & \text{if } m = j, \end{cases} \tag{3.75}$$

where we are using the most recent information to compute $(V^*_{i,j})^{n+1}$. Note that in the case when the similarity reduction is used, $(V^*_{i,j})^{n+1}$ is defined as:

$$(V^*_{i,j})^{n+1} = S^j_i - K_j + (1+p)\hat{V}^{n+1}_{l,j} \tag{3.76}$$

and no interpolation is required to determine $(V^*_{i,j})^{n+1}$.

When no similarity reduction is possible, the calculation of $(V^*_{i,j})^{n+1}$ still requires interpolation and use of the data from pricing problems with higher strike values. Consequently, the one-dimensional problems will once again be solved in decreasing order, namely from $j = j_{max}$ to $j = 0$. As such, we can write the complete solution process as done in Algorithm 3.77.

In Section 4, 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. Note that previous work on reload options in both [11] and [8] 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 of the numerical results to the analytical values. Furthermore, as shown in [12], an explicit application of the constraint when pricing American options can result in oscillations in the gamma (V_{SS}) of the option value.

Explicit Constraint

For $j = 0, \ldots, j_{max}$
  For $i = 0, \ldots, j_{max}$
    $V^0_{i,j} = \text{Payoff}(S^j_i, K_j)$
  EndFor i
EndFor j
For $n = 0, \ldots, T/dt$
  For $j = j_{max}, \ldots, 0$
    Determine $\hat{V}^{n+1}_j$ by solving $\hat{\mathcal{B}}^{n+1}_j \hat{V}^{n+1}_j = V^n_j$
    For $i = 0, \ldots, j_{max}$
      $V^{n+1}_{i,j} = \max((V^*_{i,j})^{n+1}, \hat{V}^{n+1}_{i,j})$
    EndFor i
  EndFor j
EndFor n (3.77)

Remark 3.14. *It is straightforward to show that the explicit constraint (Algorithm 3.77) is unconditionally stable and monotone when fully implicit timestepping is used.*

4 Numerical Results

Having examined the analytical properties of the increased reload pricing equations, we now move on to consider numerical results obtained when pricing such contracts. We begin by carrying out a

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

Table 4.1: Parameter values used when pricing increased reload option contracts.

convergence study of the increased reload pricing model in Section 4.1. We also show that companies can reduce their option expense by replacing infinite reload options ($p = 0$) by increased reload options (with $p > 0$). Next, we demonstrate how the implicit application of the reload constraint in equation (2.15) is superior to the explicit application of this same constraint when pricing increased reload options. Finally, we introduce a volatility surface and demonstrate it's effect on the value of increased reload options with $p = 0$.

4.1 Convergence Study

In order to verify the legitimacy of our pricing model for increased reload options, we carry-out a convergence analysis for both the general case when $p \neq 0$ and the special case of $p = 0$. The numerical values obtained for infinite reload options ($p = 0$) will be compared with analytical values for these contracts presented in [8]. The parameter values chosen for the convergence analysis are presented in Table 4.1 and will be used throughout this section unless otherwise specified. Also, note that the convergence tolerance in Algorithm 3.70 is set to 1×10^{-8} and that $\epsilon = \Delta\tau_0 \times 10^{-7}$ in the penalty term (see equation (3.8)) where $\Delta\tau_0$ is the initial timestep size on the coarsest grid.

Table 4.2 holds the results of a convergence study carried out for increased reload options when $p = 5\%$. Let us first shed some light on the content of each column of Table 4.2. The first column *Refinement* contains the refinement level used when pricing the 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 4.2. 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. Finally, 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. We note that the convergence ratio obtained in Table 4.2 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 were taken when fully implicit is used while variable timesteps were taken in the case of Crank-Nicolson. See [12] 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 also carry out a convergence study in the special case when $p = 0$, the results of which are presented in Table 4.3. We see that in this case, linear

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 4.2: 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. Note that the constraint is applied implicitly as specified in Algorithm 3.69. The initial timestep is $\Delta\tau_0 = 0.1$ years. Other parameter values are presented in Table 4.1.

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 4.3: 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. Note that the reload constraint is applied implicitly as specified in Algorithm 3.69. The initial timestep is $\Delta\tau_0 = 0.1$ years. Other parameter values are presented in Table 4.1.

convergence was obtained when fully implicit timestepping is used but quadratic convergence was not obtained when Crank-Nicolson timestepping was chosen. Indeed, Crank-Nicolson only provided linear convergence. Additional tests using a second order BDF scheme were also carried out with similar results. Nonetheless, we note that the results in Table 4.3 appear to be converging to the analytic values for infinite reload option contracts obtained in [8]: \$64.67 at $S = \$100$.

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

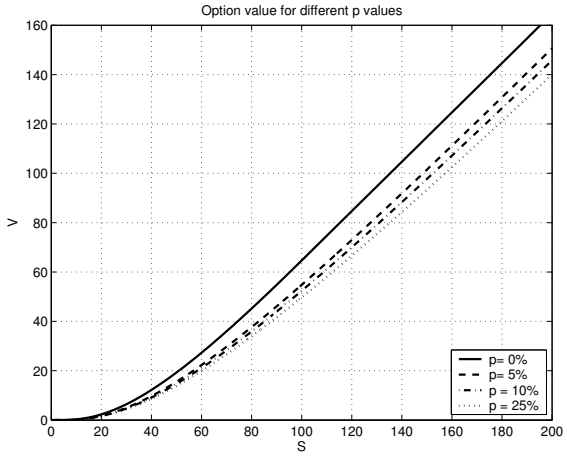
Table 4.4: Price of an increased reload option at $S = \$100$ for different p values. Additional parameters used are presented in Table 4.1. Crank-Nicolson timestepping with variable timesteps was used; the initial timestep is set to $\Delta\tau_0 = 0.1$ years.

To complete our analysis, we consider the particular features of the option delta (V_S) and gamma (V_{SS}). Note that these quantities are hedging parameters, and hence are of practical importance. Figure 4.1 presents the option value, the delta and the gamma of increased reload option contracts with different values of p. We 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. Note that the distinct shape of the gamma curve when $p = 0$ reflects the optimal exercise policy for this contract. Indeed, the kink in the option gamma curve implies that it is optimal to exercise the reload option whenever $S > K$ [11, 8]. 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.

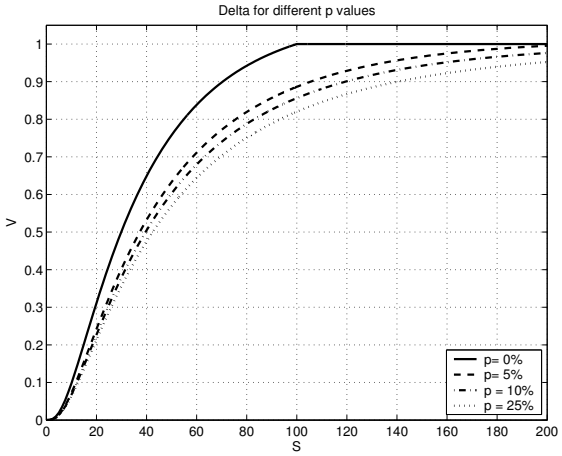
Finally, we investigate how the option value is affected by the value of p. Recall that this contract modification is suggested as a tool to reduce option expense for companies issuing infinite reload options. As such, Table 4.4 presents the value of increased reload options for five different choices of p. We see that increasing p from 0% to 1% results in an 8% price reduction of the option contracts. However, we see that setting p to larger values has a less significant impact on the option value. This trend is confirmed by Figure 4.2 which demonstrates that setting p to any value greater than 30% results in the same option value. Indeed, beyond that point the value of the reload contract is essentially identical to the value of a 10 year European option.

Hence, this leads us to conclude that transforming infinite reload options ($p = 0$) into increased reload option contracts (with $p > 0$) results in a significant price reduction for small 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.

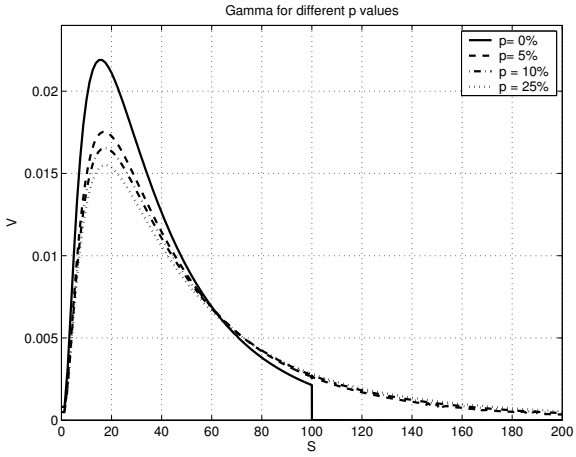
In addition, it is also interesting to note that for standard infinite reload options ($p = 0\%$), it is always optimal to reload whenever $S > K$ [11]. However, for small values of p, i.e. $p = 5\%$, we can see from Figure 4.1 that it is not optimal to reload for $S < \$200$ (when $K = \$100$). In fact, when $p = 5\%$, it is optimal to reload only for $S \simeq \$214$. Consequently, the optimal reload policy is extremely sensitive to small changes in p.



(a) Option value for different p values.



(b) Option delta for different p values.



(c) Option gamma for different p values.

Figure 4.1: 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 4.1.

4.2 Implicit vs Explicit Application of the Reload Constraint

All numerical results presented thus far have been obtained when the reload constraint in equation (2.15) is applied implicitly (as in Algorithm 3.69). An alternative to this choice is to apply the reload constraint explicitly at each timestep as outlined in Algorithm 3.77. If we are using a fully implicit timestepping method, which is only $\mathcal{O}(\Delta\tau)$, then one might argue that we might as well use the simpler Algorithm 3.77 as done in both [11] and [8]. Though much simpler to implement, this explicit method results in poor accuracy. Indeed, Table 4.5 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 $(V^*_{i,j})^{n+1}$ (see equation (3.76)). Note that the numerical results obtained when the constraint is applied explicitly are very far from the analytic values for infinite reload option contracts obtained

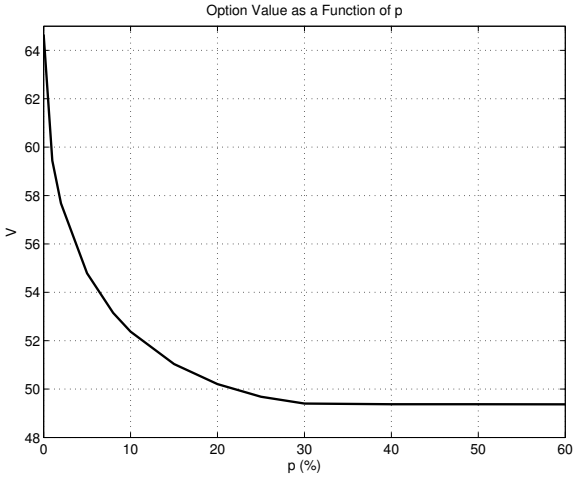


Figure 4.2: Value of an increased reload option (at $S = \$100$) as a function of the percentage increase p. The parameters used are presented in Table 4.1.

		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 4.5: Value of an increased reload option contract with $p = 0\%$ when the reload constraint (2.3) is applied both implicitly and explicitly. Note that these results are obtained using a similarity reduction. Also, fully implicit timestepping is chosen. The parameter values used in these calculations are presented in Table 4.1.

in [8]: \$54.79 at $S = \$90$, \$64.67 at $S = \$100$ and \$74.67 at $S = \$110$.

These comments are confirmed by the convergence analysis results presented in Table 4.6. These results demonstrate that the numerical solution remains far from the analytical option value for reasonable refinement levels. Indeed, the option value obtained with refinement level 4 when Crank-Nicolson timestepping is used is still about \$0.40 below the analytical value of \$64.67 from [8].

As a side note, both the similarity reduction and the full two-dimensional approach provide identical results as shown in Table 4.7. Thus, when applicable, the similarity reduction may be considered as an alternate and less computationally expensive solution method. Furthermore, we show in Appendix B that the error associated with the boundary condition at K_{max} (see equation (2.15)) can be practically eliminated provided that K_{max} is chosen appropriately, i.e. $K_{max} > 1000$.

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 4.6: Value of an increased reload option with $p = 0\%$ at $S = \$100$ using both fully implicit and Crank-Nicolson (constant timesteps) timestepping for different refinement levels. Note that the reload constraint is applied explicitly in the context of a similarity reduction. Other parameter values chosen are presented in Table 4.1.

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 4.7: Value of an increased reload option contract 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.). Note that fully implicit timestepping is used and that the reload constraint is applied implicitly. The parameter values used in these calculations are presented in Table 4.1.

4.3 Adding a Volatility Surface

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. It is well known that constant volatility models cannot reproduce observed market option prices. A standard approach is to assume that $\sigma = \sigma(S, \tau)$ and to determine $\sigma(S, \tau)$ by calibration [19]. In addition, 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 modelling assumption implies 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 combina-

		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 4.8: Value of an increased reload option contract 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 (4.2) holds and *Case 2* implies that both equations (4.2) and (4.3) hold. For comparison purposes, we include results for two different constant volatility values (*Const. Vol.*). Note that fully implicit timestepping is chosen. Additional parameter values are presented in Table 4.1.

tions of strike and asset values, this modelling assumption basically implies that the volatility skew will always be centered around the current strike of the contract considered.

The *sticky delta* property stems from the following assumption about asset volatility:

$$\sigma(S, K, t) = \sigma(\rho S, \rho K, t). \tag{4.1}$$

Setting $\rho = \frac{K^*}{K}$, we find that:

$$\sigma(S, K, t) = \sigma\left(\frac{S}{K}K^*, K^*, t\right), \tag{4.2}$$

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

The second modelling 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 modelling assumption implies that:

$$\sigma(S, K, t) = \sigma(S, K, t - t_r), \tag{4.3}$$

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

Table 4.8 presents numerical results obtained when pricing infinite reload options using a volatility surface and assuming that equation (4.2) and/or equation (4.3) hold. The local volatility surface used to price these contracts was obtained by calibration to synthetic option prices. As outlined in [20], synthetic market prices were generated for both vanilla call and put options using the exact European prices under a Merton jump diffusion model. 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 4.3 for a graphical representation of the volatility surface obtained. More details regarding the parameters used to generate this volatility surface are specified in [20].

Table 4.8 shows values obtained when using a volatility surface in pricing 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 (4.2) holds, and *Case 2* where the same volatility surface is

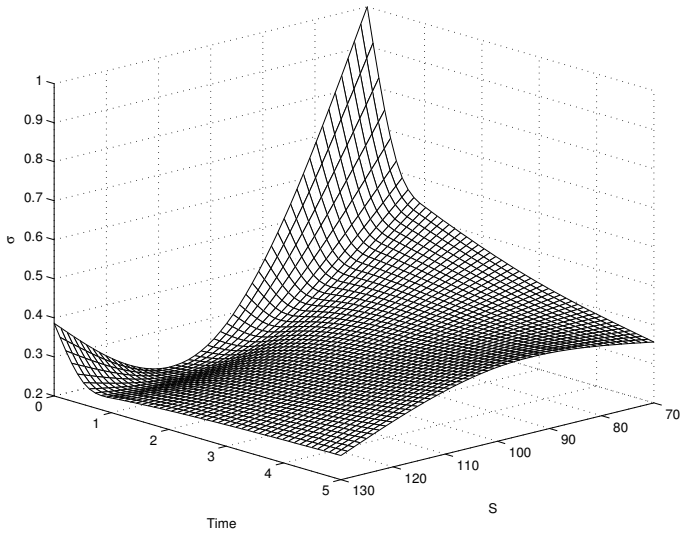


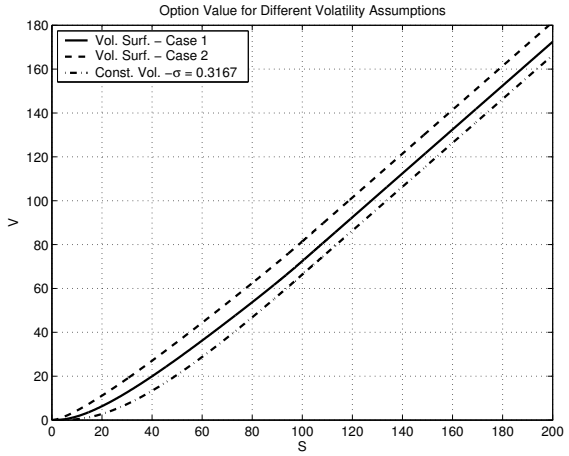
Figure 4.3: Plot of volatility surface obtained by calibrating to synthetic market prices.

used but now both equations (4.2) and (4.3) hold. Note that 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 4.8 shows the impact of updating the volatility surface on the contract value. Indeed, we see that *rolling forward* 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, we can see that the use of a volatility surface, compared to the assumption of constant volatility, results in increased option values.

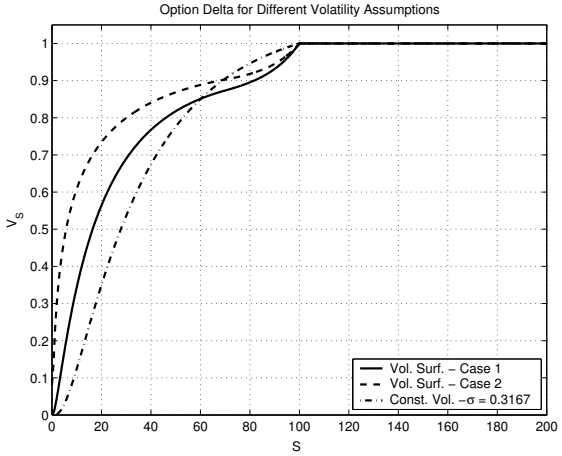
Figure 4.4 shows the option value (V), the delta (V_S) and the gamma (V_{SS}) for the cases considered in Table 4.8. We see that while the shape of the option value curve is relatively unaffected by the use of a local volatility surface, this is not the case for the delta and gamma of the option. Indeed, 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.

5 Conclusion

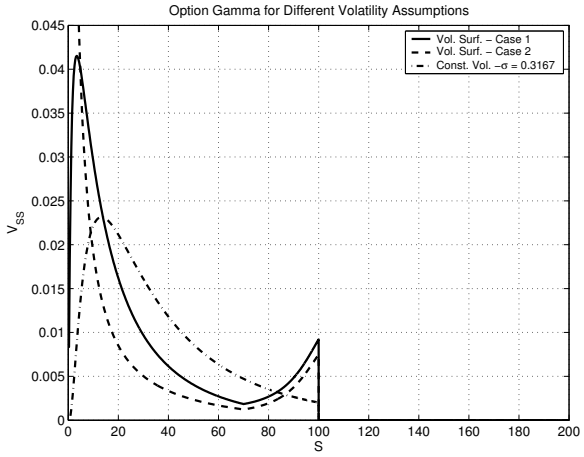
Infinite reload options are considered as some of the more complex and costly types of employee stock options. As such, issuing companies who need to include these contracts in their balance sheets may look for ways to reduce their no-arbitrage value. In this paper, we specify the increased reload option contract 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 $p = 0$. This increased reload option contract can then be used as a tool to help companies reduce their option expenses.



(a) Option value.



(b) Option delta.



(c) Option gamma.

Figure 4.4: Plot of the option value (V), the delta (V_S) and the gamma (V_{SS}) obtained when pricing a 10 year increased reload option contract with $p = 0\%$ under different modelling assumptions. *Vol.Surf.- Case 1* implies that a local volatility surface is used and equation (4.2) holds. *Vol.Surf.- Case 2* implies that a local volatility surface is used and equations (4.2) and (4.3) both hold. The update interval in this case is 1 year. *Const. Vol.* makes the assumption that $\sigma = .3167$ is a constant parameter. The other parameters used in the pricing process are presented in Table 4.1.

More specifically, the following contributions were made:

- A detailed pricing model for increased reload options was outlined as an impulse control problem which was expressed as a Hamilton-Jacobi-Bellman equation. A penalization method was also used to apply the reload constraint.
- In the context of convergence to the viscosity solution, we demonstrated that the discrete equations obtained from the pricing model are consistent, stable and monotone [2].
- It was also demonstrated 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 reload 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.

Consequently, 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 and still provide 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. Consequently, the value of infinite reload options in this context would drop significantly as this would effectively turn infinite reload options into reload options with limited reload rights. These different contract modifications will surely be considered more seriously by companies that have issued stock options that are now looking to reduce their stock option expense recorded on their balance sheet.

APPENDICES

A Discretization

The partial differential equation (2.6) can be approximated by replacing derivatives by finite difference approximations. Using the following notation $V_{i,j}^n = V(S_i^j, K_j, \tau^n)$, the discrete version of equation (2.6) can be written as:

$$V_{i,j}^{n+1}\left[1+(\alpha_{i,j}^{n+1}+\beta_{i,j}^{n+1}+r)(1-\theta)\Delta\tau\right]-(1-\theta)\Delta\tau\beta_{i,j}^{n+1}V_{i+1,j}^{n+1}-(1-\theta)\Delta\tau\alpha_{i,j}^{n+1}V_{i-1,j}^{n+1}=$$
$$V_{i,j}^{n}\left[1-(\alpha_{i,j}^{n}+\beta_{i,j}^{n}+r)\theta\Delta\tau\right]+\theta\Delta\tau\beta_{i,j}^{n}V_{i+1,j}^{n}+\theta\Delta\tau\alpha_{i,j}^{n}V_{i-1,j}^{n}+P(V_{i,j}^{n+1},(V_{i,j}^{*})^{n+1}), \quad \text{(A.1)}$$

where $0 \le \theta \le 1$, $P(V_{i,j}^{n+1},(V_{i,j}^{*})^{n+1})$ is the discrete penalty term (as defined in equation (3.7)) and $\alpha_{i,j}^n, \beta_{i,j}^n$ depend on the type of approximation used for the first and second derivatives. Note that choosing $\theta = 0$ corresponds to fully implicit timestepping while $\theta = 1/2$ results in Crank-Nicolson timestepping.

The choice of discretization for the derivative terms in equation (2.6) will determine the value of both $\alpha_{i,j}^n$ and $\beta_{i,j}^n$. For example, choosing the higher order central difference scheme leads to the following values of $\alpha_{i,j}^n$ and $\beta_{i,j}^n$:

$$\alpha_{i,j,central}^{n} = \frac{(\sigma_{i,j}^{n}S_i^j)^2}{(S_i^j-S_{i-1}^j)(S_{i+1}^j-S_{i-1}^j)} - \frac{(r-q)S_i^j}{S_{i+1}^j-S_{i-1}^j},$$
$$\beta_{i,j,central}^{n} = \frac{(\sigma_{i,j}^{n}S_i^j)^2}{(S_{i+1}^j-S_i^j)(S_{i+1}^j-S_{i-1}^j)} + \frac{(r-q)S_i^j}{S_{i+1}^j-S_{i-1}^j}, \quad \text{(A.2)}$$

where $\sigma^n_{i,j} = \sigma(S^j_i, K_j, \tau^n)$. However, if either $\alpha^n_{i,j,central}$ or $\beta^n_{i,j,central}$ is negative, oscillations may appear in the solution. In order to alleviate these oscillations, it is preferable to choose other discretization techniques at the problem nodes such as forward or backward differences. Forward differences produces:

$$\begin{aligned}
\alpha^n_{i,j,forward} &= \frac{(\sigma^n_{i,j} S^j_i)^2}{(S^j_i - S^j_{i-1})(S^j_{i+1} - S^j_{i-1})}, \\
\beta^n_{i,j,forward} &= \frac{(\sigma^n_{i,j} S^j_i)^2}{(S^j_{i+1} - S^j_i)(S^j_{i+1} - S^j_{i-1})} + \frac{(r-q)S^j_i}{S^j_{i+1} - S^j_i},
\end{aligned} \tag{A.3}$$

while backward differences delivers:

$$\begin{aligned}
\alpha^n_{i,j,backward} &= \frac{(\sigma^n_{i,j} S^j_i)^2}{(S^j_i - S^j_{i-1})(S^j_{i+1} - S^j_{i-1})} - \frac{(r-q)S^j_i}{S^j_{i+1} - S^j_i}, \\
\beta^n_{i,j,backward} &= \frac{(\sigma^n_{i,j} S^j_i)^2}{(S^j_{i+1} - S^j_i)(S^j_{i+1} - S^j_{i-1})}.
\end{aligned} \tag{A.4}$$

Algorithmically, the decision between a central or forward discretization at each node is made based on the criteria presented below:

$$\begin{array}{l}
\text{If } \left[\alpha^n_{i,j,central} \geq 0 \text{ and } \beta^n_{i,j,central} \geq 0\right] \text{ then} \\
\quad \alpha^n_{i,j} = \alpha^n_{i,j,central} \\
\quad \beta^n_{i,j} = \beta^n_{i,j,central} \\
\text{ElseIf } \left[\beta^n_{i,j,forward} \geq 0\right] \text{ then} \\
\quad \alpha^n_{i,j} = \alpha^n_{i,j,forward} \\
\quad \beta^n_{i,j} = \beta^n_{i,j,forward} \\
\text{Else} \\
\quad \alpha^n_{i,j} = \alpha^n_{i,j,backward} \\
\quad \beta^n_{i,j} = \beta^n_{i,j,backward} \\
\text{EndIf}
\end{array} \tag{A.5}$$

Note that the criteria (A.5) guarantees that both $\alpha^n_{i,j}$ and $\beta^n_{i,j}$ are non-negative. Thus, since $r \geq 0$, we have that the discretization presented in (A.1) will result in a positive coefficient scheme. Note that a positive coefficient scheme implies the following condition:

$$\alpha^n_{i,j} \geq 0 \; ; \; \beta^n_{i,j} \geq 0 \quad \text{for } i = 0, \ldots j_{max}. \tag{A.6}$$

B Error Introduced by Choice of K_{max}

In this section, we carry out a test to determine the error associated with choosing a finite value for K_{max}. Recall that the discrete domain used to price increased reload options is of the form

	Nbr. Nodes	K_{max}	S_{max}
Grid 1	58	300	900
Grid 2	59	500	2500
Grid 3	60	1000	10000
Grid 4	61	2000	40000

Table B.1: Characteristics of four different grid constructions. *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.

$[0, S_{max}] \times [0, K_{max}]$ where $K_{max} = K_{j_{max}}$ and

$$S_{max} = \frac{K_{max}}{K^*}\frac{K_{max}}{1+p}, \tag{B.1}$$

where K^* is the initial contract strike price and p is the percentage increase (see equation (3.1) for more details). In Section 2.1, we specify the boundary condition at $K = K_{max}$ which effectively assumes that $\sigma(S, K, \tau)$ becomes constant as $S \to K_{max}$. This assumption enables us to apply a similarity reduction at the boundary points. However, such an assumption will introduce a given amount of error, which is what we now try to determine.

Table B.1 presents the four different initial grids that we will consider in our test. Each grid is characterized by the number of nodes it contains (*Nbr. Nodes*) in each of the S and K directions, as well as the value of both K_{max} and S_{max}

Table B.2 presents the value of a 10 year infinite reload option contract on each of the four initial grids considered. Recall that the infinite reload option contract is a special case of the increased reload option contract where $p = 0$. We evaluate the infinite reload option using a volatility surface (without the sticky delta property) that is updated every year. The values in Table B.2 demonstrate that the option value is not significantly affected by the choice of K_{max} as long as this parameter is chosen large enough, i.e. $K_{max} \geq 1000$. Indeed, the results obtained for grids 3 and 4 are identical. In addition, note that the error associated with the grid design is negligible for as the grid is refined and timestep size is reduced. Hence, choosing a reasonable value for K_{max} such as $K_{max} \geq 1000$ results in minimal error.

References

[1] 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.

[2] G. Barles. Convergence of numerical schemes for degenerate parabolic equations arising in finance theory. In L.C.G. Rogers and D. Talay, editors, *Numerical Methods in Finance*, pages 1–21. Cambridge University Press, Cambridge, 1997.

[3] G. Barles, Ch. Daher, and M. Romano. Convergence of numerical schemes for parabolic equations arising in finance theory. *Mathematical Models and Methods in Applied Sciences*, 5(1):125–143, 1995.

Refinement	Nodes	Timesteps	Option Value
Grid 1			
0	58	100	66.048232
1	115	200	65.922058
2	229	400	65.836968
3	457	800	65.789509
Grid 2			
0	59	100	66.052507
1	117	200	65.924701
2	233	400	65.839258
3	465	800	65.791605
Grid 3			
0	60	100	66.052514
1	119	200	65.924707
2	237	400	65.839261
3	472	800	65.791606
Grid 4			
0	61	100	66.052514
1	121	200	65.924707
2	241	400	65.839261
3	481	800	65.791606

Table B.2: Value of an increased reload option with $p = 0\%$ at $S = \$100$ when a volatility surface is used (without the sticky delta property). The volatility surface is *rolled forward* every year and the contract maturity is 10 years. Excluding volatility, the other parameter values chosen are specified in Table 4.1. Note that fully implicit timestepping was used with constant timesteps where $\Delta\tau_0 = 0.1$ years.

[4] G. Barles and P.E. Souganidis. Convergence of approximation schemes for fully nonlinear equations. *Asymptotic Analysis*, (4):271–283, 1991.

[5] A. Bensoussan and J.-L. Lions. *Impulse Control and Quasi-Variational Inequalities*. Gauthier-Villars, 1984.

[6] J.-P. Chancelier, B. Øksendal, and A. Sulem. Combined stochastic control and optimal stopping, and application to numerical approximation of combined stochastic and impluse control. *Proceedings of the Steklov Institute of Mathematics*, 237:140–163, 2002.

[7] 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 Sociesty*, 27:1–67, July 1992.

[8] 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.

[9] M. Dai and Y. Kwok. Valuing employee reload options under time vesting requirement. Working paper, Hong Kong University of Science and Technology, 2004.

[10] 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.

[11] P. H. Dybvig and M. Loewenstein. Employee reload options: Pricing, hedging, and optimal exercise. *Review of Financial Studies*, 16:145–171, 2003.

[12] 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.

[13] E.R. Jakobsen. On the rate of convergence of approximation schemes for Bellman equations associated with stopping time problems. *Mathematical Models and Methods in Applied Sciences*, (13):613–644, 2003.

[14] R. C. Merton. Theory of rational option pricing. *Bell Journal of Economics and Management Science*, 4:141–183, 1973.

[15] Joe Nedumgottil, Director of Equity Compensation, Unitrin Inc. Private communication (2005).

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

[17] H. Soner. Optimal control with state-space constraint. *SIAM Journal on Control and Optimization*, 24(3):552–561 and 1110–1122, 1986.

[18] V. Ly Vath, M. Mnif, and H. Pham. A model of optimal portfolio selection under liquidity risk and price impact. Working paper, Laboratoire de Probabilités et Modèles Aléatoires, Université Paris 7, France, October 2005.

[19] P. Wilmott. *Derivatives: The Theory and Practice of Financial Engineering*. Wiley, West Sussex, England, 1998.

[20] H. Windcliff, P. A. Forsyth, and K. R. Vetzal. Numerical methods and volatility models for valuing cliquet options. To appear in Applied Mathematical Finance.

[21] 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.