EFFICIENT AND ACCURATE NUMERICAL PDE METHODS FOR PRICING FINANCIAL DERIVATIVES

by

Mufan Li

Supervisor: Christina Christara

April 2015

Abstract

Efficient and Accurate Numerical PDE Methods for
Pricing Financial Derivatives

Mufan Li
Bachelor of Applied Science and Engineering
Department of Electrical and Computer Engineering
University of Toronto
2015

The main difficulty in pricing American options comes from the early exercise right, creating a non-linear constraint on the Black-Scholes PDE. Under a finite difference discretization of the PDE, the price of an American can be approximated, with several techniques to properly handle the American Constraint. While both an iterative penalty method and a direct operator splitting method are convergent, the efficiency and quality require a comprehensive study. Using Crank-Nicolson time stepping and non-uniform grids, the methods are compared in numerical experiments. The criteria include order of convergence, efficiency, and complexity.

Acknowledgements

I have to thank my advisor Professor Christara, without her careful guidance this thesis would not have been possible. Under Professor Christara's supervision, this research project has been an amazing learning experience. I would like to thank you very much for your support and teaching over the past year.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

A financial derivative, or just *derivative*, is a security whose value depends on the values of one or more other securities, the latter often referred to as the *underlying* securities, or just the *underlying*. An *option* is a type of derivative that gives the holder the right, but not the obligation, to buy or sell an underlying security at a certain price (*strike price*) and time (*expiry time*) in the future. Since it is a tradable security, it should have a price. The accurate calculation of options' prices is critical for the smooth operation of the markets, and has been the subject of much research study [5].

There are several mathematical models that describe the price of an option. According to a well-known model, the option's price $V(S,t)$, where S is the underlying price and t the time, satisfies the Black-Scholes (BS) Partial Differential Equation (PDE) [5], subject to appropriate initial and boundary conditions.

In its simplest form, the option that allows exercise only at a certain future date is called an *European* option. An European option admits an analytical solution formula. However, some of the most commonly traded options allow exercise at all times before the expiry date. Known as the *American* options, these securities do no admit a closed form solution. The complexity that arises from the early exercise right can be formulated as a mathematical problem, common known as *linear complementarity problem* (LCP). This problem involves, not only the BS PDE, but also a related partial differential inequality, which, together, give rise to some non-linearity in the problem.

There are three main numerical approaches to price American options, lattice based discretization, Monte Carlo simulations, and the numerical PDE methods. It should be noted that all three approaches compute approximations to American option prices, the accuracy of which can be controlled by appropriate parameters.

The lattice approach, also known as *binomial* (or *trinomial*) trees, provides an approximation to the option price in discrete time and underlying price points. The lattice is created by constructing all possible price paths of the underlying asset, which forms the shape of a tree. With the asset price at the expiry, the lattice allows for the calculation of the option at the previous step, and this procedure can be taken to the initial node containing the current price. The solution converges to continuous price as the number of nodes in the lattice increases.

Monte Carlo simulations takes full advantage of Martingale pricing, where the option price is the expected discounted payoff under the risk-neutral measure. By simulating a large number of paths, the expected value can be approximated by averaging the simulation results. While Monte Carlo is quick to find the option price at one point of the underlying asset price, the results are not reproducible due to the nature of the random simulation.

The PDE approach models the pricing problem as a PDE with appropriate boundary and initial conditions, and possibly some additional conditions. In this approach the unknown option price is a function of the underlying price (treated as a spatial variable) and of time. The main advantage compared to the lattice and Monte Carlo methods is that the PDE approach provides values at many points of the underlying price and time domains, as well as a spectrum of the price derivatives (the "*Greeks*"). The most common PDE approach is the finite difference method. By discretizing the time and space (underlying price) dimensions as a grid, one can infer the values at each grid point from the boundary points and relations between nearby points arising from the PDE.

The methods of interest in this study are the finite difference PDE methods for solving the LCP arising in American option pricing. While there are many approaches to resolve the non-linearity, there is no commonly accepted optimal method. These methods can be categorized into two types, direct and iterative. The direct solvers produce an explicit solution with the number of calculations being known in advance. Iterative methods approximate a solution by continuous improvement of an initial guess, and use a stopping criterion to decide when enough improvement has been achieved. The number of improvement steps and, therefore the number of calculations is not known in advance. Although most methods of both types will continue to improve (i.e. reduce) the error with increased computation complexity, the order of convergence of a PDE discretization method is crucial in quantifying the rate of error reduction as compared to the increase in computational complexity. Additionally, the placement of discrete points can also affect the performance and accuracy of the solver.

The main gap in this research area is a comprehensive evaluation of these methods on a complete set of criteria. This thesis' primary objective is to study and compare different methods for the solution of the LCP, using criteria including accuracy, order of convergence, efficiency, and complexity.

Chapter 2

Formulation of problem

2.1 Asset Price Model and Black-Scholes PDE

An asset's price movements is commonly modeled by a geometric Brownian motion,

$$dS = \mu S dt + \sigma S dW \tag{2.1}$$

where S is the asset price, μ is the drift, σ is the volatility, t is the (forward) time, and W is the standard Wiener process. For simplification of later equations, a change of variable $\tau = T - t$ is introduced, where T is the expiry of the option. Thus, $\tau = 0$ and $\tau = T$ correspond to the expiry and the current times, respectively. It can be shown [5], that, if S follows (2.1), and if certain additional conditions hold, the price $V(S, \tau)$ of an European option satisfies the Black-Scholes (BS) equation

$$\frac{\partial V}{\partial \tau} = \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV \tag{2.2}$$

where r is the risk-free interest rate, and the equation is subjected to boundary and initial conditions depending on the specific type of derivative.

For later convenience, we define

$$\mathcal{L}V \equiv \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV. \tag{2.3}$$

Thus, the BS PDE takes the form

$$\frac{\partial V}{\partial \tau} = \mathcal{L}V. \tag{2.4}$$

The S variable, considered as space variable spans the domain $[0, \infty)$. The τ variable, which is the (backward) time variable, spans the domain $[0, T]$. In the typical option pricing problem, we are interested in calculating the current price of the option for any S in the domain, i.e. $V(S, \tau)$, for $\tau = T$ and $S \in [0, \infty)$.

We next discuss initial and boundary conditions that the European option prices satisfy. To this end, we distinguish between *call* options, that allow the holder to buy, and *put* options that allow the holder to sell. Let K be the strike price.

To present the initial conditions for European call and put options, we consider the case $\tau = 0$ (or $t = T$), i.e. the option value at expiry. The price of an option at expiry is determined by a payoff function $\Phi(S)$, which, for European call and put options is given by

$$\begin{aligned} V_{call}(S,\tau)|_{\tau=0} &= \Phi_{call}(S) = \max(S-K, 0) \\ V_{put}(S,\tau)|_{\tau=0} &= \Phi_{put}(S) = \max(K-S, 0) \end{aligned} \tag{2.5}$$

respectively.

The price of European call and put options at the boundary of the spatial domain is given by

$$\begin{aligned} V_{call}(0,\tau) = 0, &\quad V_{call}(S,\tau) \xrightarrow[S\to\infty]{} S - Ke^{-r\tau} \\ V_{put}(0,\tau) = Ke^{-r\tau}, &\quad V_{put}(S,\tau) \xrightarrow[S\to\infty]{} 0 \end{aligned} \tag{2.6}$$

respectively.

2.2 American options

For American options, there is an additional key constraint due to the ability to exercise at anytime before expiry T, which adds complexity to the problem:

$$V(S,\tau) \geq \Phi(S), \quad 0 \leq \tau \leq T. \tag{2.7}$$

At any point in time, the holder of an American option can always decide whether to exercise the right or to continue holding. This leads to a decision (interior or free) boundary at $S_f(\tau)$, where for each τ it is optimal to exercise on one side of the boundary (left), and optimal to hold on the other (right). More specifically, the American put pricing problem is described as below:

$$\begin{aligned}
V(S,0) = \Phi_{put}(S) = \max(K-S,0), &\qquad \text{for } 0 \le S \le S_f(\tau)\\
\frac{\partial V}{\partial \tau} - \mathcal{L}V > 0, &\qquad \text{for } 0 \le S < S_f(\tau)\\
\frac{\partial V}{\partial S}(S_f(\tau),\tau) = -1 &\\
V(S,\tau) > \Phi_{put}(S) = \max(K-S,0), &\qquad \text{for } S_f(\tau) < S < \infty\\
\frac{\partial V}{\partial \tau} - \mathcal{L}V = 0, &\qquad \text{for } S_f(\tau) < S < \infty\\
V(\infty,\tau) = 0 &
\end{aligned} \tag{2.8}$$

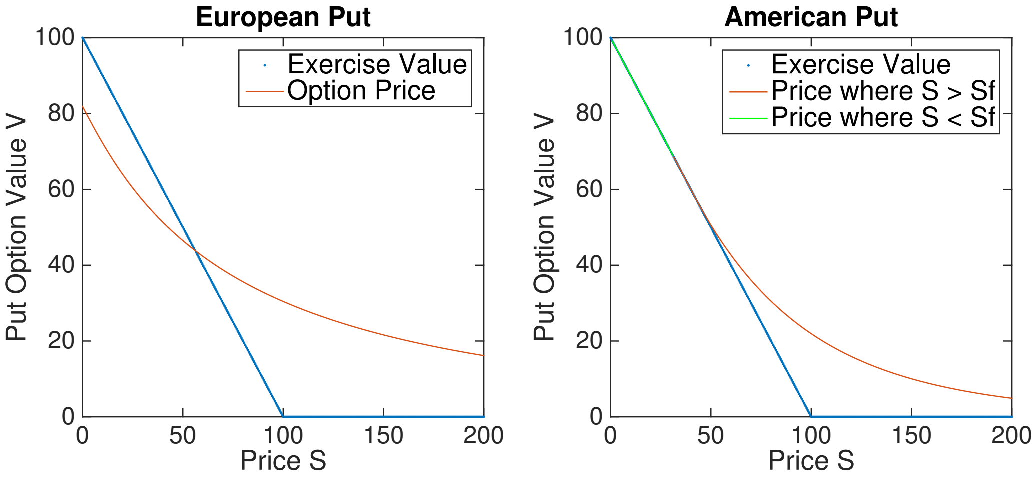

Figure 2.1: Comparison of an European and an American put option prices with the same parameters $K = 100$, $\sigma = 0.8$, $T = 5$, and $r = 0.1$. The American option follows different shapes on the two sides of the free boundary $S_f(\tau) \approx 51.57$, where it is linear as the exercise value on the left, and curved on the right.

The respective linear complementarity problem (LCP) is

$$\begin{aligned} \frac{\partial V}{\partial \tau} - \mathcal{L}V &\geq 0, \\ V - \Phi &\geq 0, \\ (\frac{\partial V}{\partial \tau} - \mathcal{L}V)(V - \Phi) &= 0. \end{aligned} \tag{2.9}$$

While the initial conditions for the American call and put options remain the same as for their European counterparts, some boundary conditions need to be adjusted to incorporate the extra constraint (2.7).

The price of American call and put options at the boundary of the spatial domain is given by

$$\begin{aligned} V_{call}(0,\tau) = 0, \quad & V_{call}(S,\tau) \xrightarrow[S\to\infty]{} S - Ke^{-r\tau} \\ V_{put}(0,\tau) = K, \quad & V_{put}(S,\tau) \xrightarrow[S\to\infty]{} 0 \end{aligned} \tag{2.10}$$

respectively.

Chapter 3

Numerical Methods

3.1 Finite Difference Method

To numerically solve a PDE like the Black-Scholes equation, the time and price dimensions of the problem are discretized using finite differences. Instead of working with the full continuous price function $V(S, \tau)$, the function is sampled at specific points that form a grid. While the spatial domain of the continuous problem is $[0, \infty)$, i.e. semi-infinite, for the discretized problem, the spatial domain is truncated to $[0, S_{\max}]$, for an appropriately large value of $S_{\max}$.

Given that $S \in [0, S_{\max}]$, $\tau \in [0, T]$, each sample is labeled with indices based on the position in the grid. Suppose there are N_S steps in the price direction and N_τ in the time direction. For simplicity of presentation, we first assume that the points in both the spatial and time directions are uniform, i.e, with $h_S = \frac{S_{\max}}{N_S}$ and $h_\tau = \frac{T}{N_\tau}$, we have $S_i = ih_S, i = 0, 1, \ldots, N_S$, and $\tau_j = jh_\tau, j = 0, 1, \ldots, N_\tau$. (Later in the thesis, we consider non-uniform grids.) Then the option price at the (i, j)th point is denoted by $V_{i,j}$, and we have $V_{i,j} = V(S_i, \tau_j)$ $\forall i = 0, 1, \ldots, N_S,\ j = 0, 1, \ldots, N_\tau$

The initial conditions of the discretized call and put pricing problem are

$$\begin{aligned} V_{call}(S_i, \tau)|_{\tau=0} &= \Phi_{call}(S_i) = \max(S_i - K, 0) \\ V_{put}(S_i, \tau)|_{\tau=0} &= \Phi_{put}(S_i) = \max(K - S_i, 0) \\ & i = 0, 1, \ldots, N_S, \end{aligned} \tag{3.1}$$

respectively.

For the European options, the boundary conditions of the discretized call and put pricing

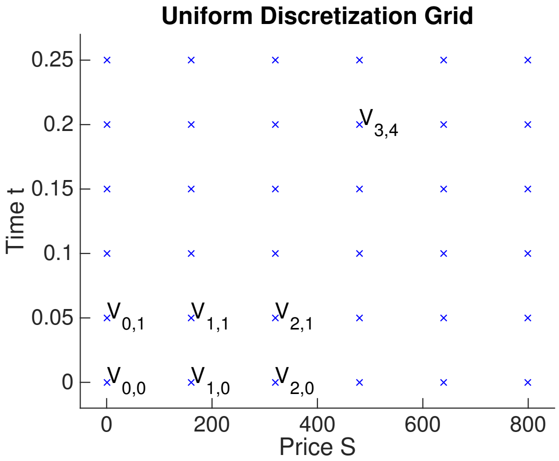

Figure 3.1: A sample uniform price discretization grid with $S_{\max} = 800$, $K = 100$, and $N_S = 5$ combined with a uniform time discretization with $T = 0.25$ and $N_\tau = 5$.

problem are

$$\begin{aligned} V_{call}(0, \tau_j) = 0, \quad & V_{call}(S_{\max}, \tau_j) = S_{\max} - Ke^{-r\tau_j} \\ V_{put}(0, \tau_j) = Ke^{-r\tau_j}, \quad & V_{put}(S_{\max}, \tau_j) = 0 \\ j = 0, 1, \ldots, N_\tau, & \end{aligned} \tag{3.2}$$

respectively.

We then discuss how the partial derivative terms in the BS PDE are discretized. We consider the forward and backward Euler discretizations of the first derivative $\frac{\partial V}{\partial \tau}$, and the centered differences discretizations of the first and second spatial derivatives $\frac{\partial V}{\partial S}$ and $\frac{\partial^2 V}{\partial S^2}$ given by

$$\left.\frac{\partial V}{\partial \tau}\right|_{i,j} = \frac{V_{i,j+1} - V_{i,j}}{h_\tau} + O(h_\tau) \tag{3.3a}$$

$$\left.\frac{\partial V}{\partial \tau}\right|_{i,j+1} = \frac{V_{i,j+1} - V_{i,j}}{h_\tau} + O(h_\tau) \tag{3.3b}$$

$$\left.\frac{\partial V}{\partial S}\right|_{i,j} = \frac{V_{i+1,j} - V_{i-1,j}}{2h_S} + O(h_S^2) \tag{3.3c}$$

$$\left.\frac{\partial^2 V}{\partial S^2}\right|_{i,j} = \frac{V_{i+1,j} - 2V_{i,j} + V_{i-1,j}}{h_S^2} + O(h_S^2). \tag{3.3d}$$

In the above discretization schemes, $O(h_\tau)$ and $O(h_S^2)$ represent the residual terms. With the above discretizations, $\mathcal{L}V$ can be discretized as below, where $c_{i,k}$ are scalars:

$$\begin{aligned}
\mathcal{L}V|_{i,j} &= (\frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV)\bigg|_{i,j} \\
&= \frac{1}{2}\sigma^2 S^2 \frac{V_{i+1,j} - 2V_{i,j} + V_{i-1,j}}{2h_S} + rS\frac{V_{i+1,j} - V_{i-1,j}}{2h_S} - rV_{i,j} + O(h_S^2) \\
&= c_{i,i+1}V_{i+1,j} + c_{i,i}V_{i,j} + c_{i,i-1}V_{i-1,j} + O(h_S^2).
\end{aligned} \tag{3.4}$$

Since $\mathcal{L}V|_{i,j}$ can be reduced to a linear combination of $V_{i,j}$'s, then for each time step τ_j, the above equation can be vectorized, and rewritten in matrix form as

$$\mathcal{L}\mathbf{V}_j = \mathbf{C}\mathbf{V}_j + \mathbf{D}_j + O(h_S^2) \tag{3.5}$$

with

$$\begin{aligned}
\mathbf{V}_j &= \begin{bmatrix} V_{1,j} & V_{2,j} & \cdots & V_{N_S-1,j} \end{bmatrix}^T \\
\mathbf{D}_j &= \begin{bmatrix} c_{1,0}V_{0,j} & 0 & \cdots & 0 & c_{N_S-1,N_S}V_{N_S,j} \end{bmatrix}^T \\
\mathbf{C} &= \begin{bmatrix} c_{1,1} & c_{1,2} & 0 & 0 & \cdots & 0 \\ c_{2,1} & c_{2,2} & c_{2,3} & 0 & \cdots & 0 \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & \cdots & 0 & 0 & c_{N_S-1,N_S-2} & c_{N_S-1,N_S-1} \end{bmatrix}
\end{aligned} \tag{3.6}$$

where the value of $\mathbf{D}_j$ is given by the boundary conditions, therefore known a priori.

For the time derivative, the approximation can be approached either as a forward step or a backward step. At the j-th step, the forward approach will relate one unknown point $V_{i,j+1}$ to three known points $V_{i-1,j}$, $V_{i,j}$, and $V_{i+1,j}$. As a result, $V_{i,j+1}$ can be found explicitly:

$$\begin{aligned}
\mathcal{L}V|_{i,j} &= \frac{V_{i,j+1} - V_{i,j}}{h_\tau} + O(h_\tau) \\
\mathbf{V}_{j+1} &= \mathbf{V}_j + h_\tau[\mathcal{L}\mathbf{V}_j + O(h_\tau)] \\
&= \mathbf{V}_j + h_\tau\left[\mathbf{C}\mathbf{V}_j + \mathbf{D}_j + O(h_\tau + h_S^2)\right]
\end{aligned} \tag{3.7}$$

Similarly, if a backward step was taken, one known point $V_{i,j}$ will be related to three unknown points $V_{i-1,j+1}$, $V_{i,j+1}$, and $V_{i+1,j+1}$. This setup forms a system of equations, which can be solved implicitly with given boundary conditions:

$$\begin{aligned}
\mathcal{L}V\big|_{i,j+1} &= \frac{V_{i,j+1}-V_{i,j}}{h_\tau} + O(h_\tau) \\
\mathbf{V}_j &= \mathbf{V}_{j+1} - h_\tau\big[\mathcal{L}\mathbf{V}_{j+1} + O(h_\tau)\big] \\
&= \mathbf{V}_{j+1} - h_\tau\left[\mathbf{C}\mathbf{V}_{j+1} + \mathbf{D}_{j+1} + O(h_\tau + h_S^2)\right] \\
\mathbf{V}_{j+1} &= \big[\mathbf{I} - h_\tau\mathbf{C}\big]^{-1}\big[\mathbf{V}_j + h_\tau\big(\mathbf{D}_{j+1} + O(h_\tau + h_S^2)\big)\big]
\end{aligned} \tag{3.8}$$

While the implicit (backward) scheme is unconditionally stable, the explicit (forward) scheme is stable only when the constraint $\frac{h_\tau}{h_S^2} \leq \tilde{c}$, where $\tilde{c}$ some constant, is satisfied. The two methods can be used in combination, with weight $\theta \in [0,1]$. Setting $\theta = 0$ gives the explicit scheme, while setting $\theta = 1$ gives the implicit scheme. Once again, the equation can be rewritten in matrix form:

$$\begin{aligned}
\frac{V_{i.j+1}-V_{i,j}}{h_\tau} &= \theta\mathcal{L}V|_{i.j+1} + (1-\theta)\mathcal{L}V|_{i.j} + O(h_\tau) \\
\left[\mathbf{I} - h_\tau\theta\mathbf{C}\right]\mathbf{V}_{j+1} - h_\tau\theta\mathbf{D}_{j+1} &= [\mathbf{I} + h_\tau(1-\theta)\mathbf{C}]\mathbf{V}_j + h_\tau(1-\theta)\mathbf{D}_j + h_\tau O(h_\tau + h_S^2).
\end{aligned} \tag{3.9}$$

Combining the terms into single matrices, we get

$$\mathbf{C}_{im}\mathbf{V}_{j+1} + \mathbf{D}_{im,j+1} = \mathbf{C}_{ex}\mathbf{V}_j + \mathbf{D}_{ex,j} + h_\tau O(h_\tau + h_S^2) \tag{3.10}$$

where

$$\begin{aligned}
\mathbf{C}_{im} &= \mathbf{I} - h_\tau\theta\mathbf{C}, \quad \mathbf{D}_{im,j+1} = -h_\tau\theta\mathbf{D}_{j+1} \\
\mathbf{C}_{ex} &= \mathbf{I} + h_\tau(1-\theta)\mathbf{C}, \quad \mathbf{D}_{ex,j} = h_\tau(1-\theta)\mathbf{D}_j.
\end{aligned} \tag{3.11}$$

The scheme is most optimal at $\theta = 0.5$ which gives the Crank-Nicolson (CN) scheme, reducing the error term in the above equation to $O(h_\tau^2 + h_S^2)$.

Both the fully implicit and the CN schemes are stable without a condition of the form $\frac{h_\tau}{h_S^2} \leq \tilde{c}$. However, it is known that the fully implicit scheme has stronger stability properties. Therefore, it is a better choice when we deal with the discontinuity in the initial condition's first derivative. This leads to the *Rannacher* smoothing scheme, where the first two time steps use fully implicit scheme, smoothing out the discontinuity. [4]

For the American options, the discretization steps are similar, with a slightly different boundary condition for the call and put respectively:

$$\begin{aligned} V_{call}(0,\tau_j) = 0, &\quad V_{call}(S_{\max},\tau_j) = S_{\max} - Ke^{-r\tau_j} \\ V_{put}(0,\tau_j) = K, &\quad V_{put}(S_{\max},\tau_j) = 0 \\ j = 0, 1, \ldots, N_\tau. & \end{aligned} \tag{3.12}$$

The discretization of (2.9) also yields a matrix form for the LCP as

$$\begin{aligned} (\mathbf{C}_{im}\mathbf{V}_{j+1} + \mathbf{D}_{im,j+1}) - (\mathbf{C}_{ex}\mathbf{V}_j + \mathbf{D}_{ex,j}) \geq 0, \\ \mathbf{V}_j - \mathbf{\Phi} \geq 0, \\ \big[(\mathbf{C}_{im}\mathbf{V}_{j+1} + \mathbf{D}_{im,j+1}) - (\mathbf{C}_{ex}\mathbf{V}_j + \mathbf{D}_{ex,j})\big] \cdot \big[\mathbf{V}_j - \mathbf{\Phi}\big] = 0 \end{aligned} \tag{3.13}$$

where the $(\cdot)$ operator denotes element wise multiplication. Note for the explicit scheme ($\theta = 0$), the $\mathbf{C}_{im}$ matrix becomes the identity, and the LCP constraint can be enforced directly by setting all $V_{i,j}$, where $V_{i,j} < \Phi(S_i)$, to $\Phi(S_i)$ at each time step.

3.2 Choice of Grids

Generally, using a uniform grid is the safest choice when the solution shape is unknown. For the Black-Scholes PDE, the solutions tend to follow a certain common shape, allowing to take advantage of the relatively linear areas.

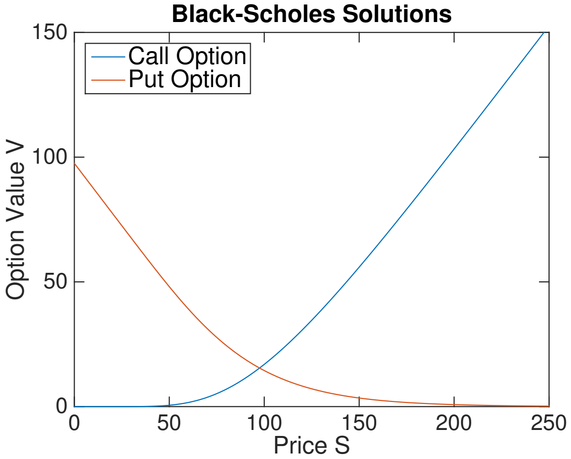

Figure 3.2: Typical solutions to the Black-Scholes PDE when strike price $K = 100$.

Observe from the figure above, the solution only takes a complex curved shape near the strike price, while the majority stays mostly straight. As a result, the finite difference approximation error is minimal at these straight parts, but rather bottlenecks near the strike. Thus, by taking smaller steps near the strike and larger steps away, i.e. having denser points near the strike and sparser points away from it, the discretization error can be reduced even with the same number of points, and therefore, computations.

We then define a general (possibly non-uniform) grid using similar notation:

$$\begin{aligned} 0 &= S_0 < S_1 < \ldots < S_{N_S} = S_{\max} \\ 0 &= \tau_0 < \tau_1 < \ldots < \tau_{N_\tau} = T \\ h_{S_i} &= S_{i+1} - S_i, \quad h_{\tau_j} = \tau_{j+1} - \tau_j \\ V_{i,j} &= V(S_i, \tau_j) \\ \forall i &= 0, 1, \ldots, N_S, \; j = 0, 1, \ldots, N_\tau \end{aligned} \tag{3.14}$$

The finite difference approximations in the price dimension then take on a different form:

$$\left.\frac{\partial V}{\partial S}\right|_{i,j} = \frac{h_{S_i}^2 V_{i+1,j} + (h_{S_{i+1}}^2 - h_{S_i}^2) V_{i,j} - h_{S_{i+1}}^2 V_{i-1,j}}{h_{S_i}(h_{S_i} + h_{S_{i+1}}) h_{S_{i+1}}} + O(h_{S_i} \cdot h_{S_{i+1}}) \tag{3.15a}$$

$$\begin{aligned} \left.\frac{\partial^2 V}{\partial S^2}\right|_{i,j} =& \frac{2h_{S_i} V_{i+1,j} - 2(h_{S_{i+1}} + h_{S_i}) V_{i,j} + 2h_{S_{i+1}} V_{i-1,j}}{h_{S_i}(h_{S_i} + h_{S_{i+1}}) h_{S_{i+1}}} \\ & + O(h_{S_{i+1}} - h_{S_i}) + O(\max\{h_{S_{i+1}}^2, h_{S_i}^2\}). \end{aligned} \tag{3.15b}$$

A non-uniform grid scheme can be defined by a monotonically increasing function

$$W : [0, S_{\max}] \mapsto [0, S_{\max}] \tag{3.16}$$

where W maps a uniform grid $[0, h_S, 2h_S, \cdots, N_S h_S]$ to a non-uniform grid $[0, S_1, S_2, \cdots, S_{N_S}]$.

In this thesis, we consider two discretization grids in the price direction. The first grid, referred to as $W_1(S)$-grid, arises from the mapping function

$$W_1(S) = \left[1 + \frac{\sinh\left(b(\frac{S}{S_{\max}} - a)\right)}{\sinh(ba)} \right] K \tag{3.17}$$

where a is a parameter determining the concentration near the strike K, and b is chosen such that $W_1(S_{\max}) = S_{\max}$. Choosing larger a generates more concentration near the strike. When $a = \frac{i}{N_S}$ for some integer $i < N_S$, the point S_{i+1} is placed at the strike K; when $a = \frac{i+\frac{1}{2}}{N_S}$, the strike K falls at the midpoint between S_i and S_{i+1}.

The second grid, referred to as $W_2(S)$-grid, arises from the mapping function

$$\begin{aligned} W_2(S) &= K + c_0 \sinh\left(c_1 \frac{S}{S_{\max}} + c_2(1 - \frac{S}{S_{\max}})\right) \\ c_1 &= \sinh^{-1}\left(\frac{S_{\max} - K}{c_0}\right), \quad c_2 = \sinh^{-1}\left(\frac{-K}{c_0}\right) \end{aligned} \tag{3.18}$$

where c_0 is a parameter determining the concentration near the strike K, c_1 and c_2 are chosen based on c_0. Choosing a smaller c_0 generates more concentration near the strike.

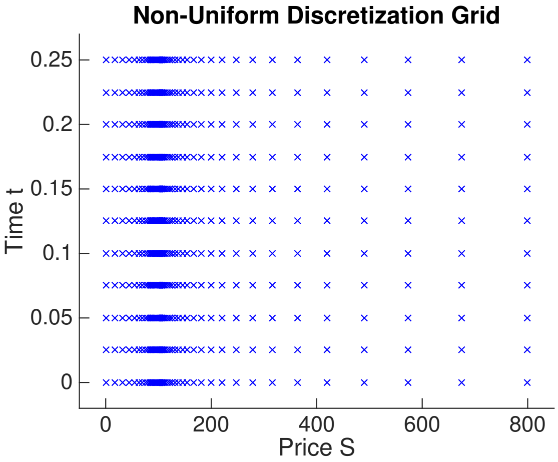

Figure 3.3: A sample non-uniform price discretization grid with $S_{\max} = 800$, $K = 100$, $N_S = 50$, and $c_0 = 8$ under the $W_2(S)$ scheme combined with a uniform time discretization with $T = 0.25$ and $N_\tau = 10$.

Notice in Figure 3.3, the grid points are placed in high concentration near the strike K as desired. Since the parameters a and c_0 can be chosen such that the two mapping functions create highly similar grids, this thesis will use only $W_2(S)$-grid to avoid repetition.

Similar to the price derivative, the time derivative also display consistent patterns, but the concentration required is less obvious to determine before solving the PDE. We consider the method suggested in [2], where the time step is selected adaptively based on the intuition

$$h_{\tau_{j+1}} \sim \left[\frac{\partial V}{\partial \tau}\right]_j^{-1} \tag{3.19}$$

and specifically the time step is selected by

$$h_{\tau_{j+1}} = h_{\tau_j} \min_i \left[d_{norm} \frac{\max(d_0, |V_{i,j+1}|, |V_{i,j}|)}{|V_{i,j+1} - V_{i,j}|} \right] \tag{3.20}$$

where d_{norm} is the target relative change per time step, and d_0 is chosen as a scale so that h_τ is not reduced due to the value $V_{i,j}$ being close to zero. Although the scheme also requires a selection of the initial step $h_{\tau 0}$, the choice of d_{norm} will dominate the behavior of this grid selector given small initial step. Once d_{norm} is selected for the coarsest (most sparse) grid, d_{norm} is then halved for each doubling of the number of steps.

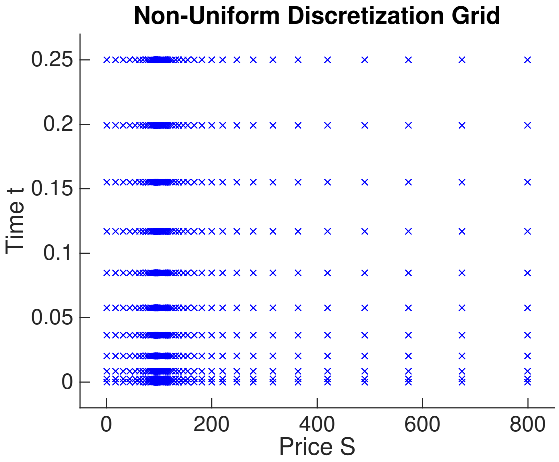

Figure 3.4: A sample non-uniform price discretization grid with $S_{\max} = 800$, $K = 100$, $N_S = 50$, and $c_0 = 8$ under the $W_2(S)$ scheme combined with an adaptive non-uniform time discretization with $T = 0.25$, $N_\tau = 10$, and $d_{norm} = 0.2$

From Figure 3.4, we can observe that more grid points are concentrated near the beginning where the time derivative is the greatest.

3.3 Discrete Penalty

The setup up to this point is sufficient for the pricing of an European option, however we need to address the non-linearity caused by the LCP (2.9) . Instead of enforcing the LCP explicitly by setting $V_{i,j} \geq \Phi(S_i), i = 1, \ldots, N_S - 1, j = 1, \ldots, N_\tau$, the penalty method in [2] adds a term that punishes the difference arising from not satisfying the constraint instead. The formulation results in the relation

$$\frac{\partial V}{\partial \tau} = \mathcal{L}V + \rho \max(\Phi - V, 0) \tag{3.21}$$

where ρ is the penalty parameter, and is typically chosen to be a large positive number.

Observe if $V \geq \Phi$, then $\frac{\partial V}{\partial \tau} = \mathcal{L}V$; otherwise if $V < \Phi$, the derivative $\frac{\partial V}{\partial \tau}$ grows arbitrarily large, forcing V to grow until $V \geq \Phi$ is satisfied. Given the initial condition $V|_{\tau=0} = \Phi$, then as the penalty parameter $\rho \to \infty$, the LCP constraint $V \geq \Phi$ is satisfied.

In the discrete case, a penalty term is placed on the next step at $\mathbf{V}_{j+1}$, resulting in the relation

$$\mathbf{V}_{j+1} - \mathbf{V}_j = (1-\theta)h_\tau(\mathbf{C}\mathbf{V}_j + \mathbf{D}_j) + \theta h_\tau(\mathbf{C}\mathbf{V}_{j+1} + \mathbf{D}_{j+1}) + \mathbf{P}_{j+1}(\mathbf{\Phi} - \mathbf{V}_{j+1}) \quad (3.22)$$

where $\mathbf{P}_{j+1}$ is the penalty matrix, with the (i,n)th element defined as

$$\left[\mathbf{P}_j\right]_{i,n} = \begin{cases} \rho & \text{if } i = n \text{ and } V_{i,j} < \Phi_i \\ 0 & \text{otherwise} \end{cases} \quad (3.23)$$

Observe $\mathbf{P}_j$ is a diagonal matrix, with diagonal elements either ρ or 0. In other words, a penalty is only applied to the terms where $V_{i,j} < \Phi_i$.

Relation (3.22) can be further simplified to

$$\left[\mathbf{C}_{im} + \mathbf{P}_{j+1}\right]\mathbf{V}_{j+1} + \mathbf{D}_{im,j+1} = \mathbf{C}_{ex}\mathbf{V}_j + \mathbf{D}_{ex,j} + \mathbf{P}_{j+1}\mathbf{\Phi}. \quad (3.24)$$

Note that $\mathbf{P}_{j+1}$ depends on $\mathbf{V}_{j+1}$, so (3.24) is non-linear.

In order to solve the above discrete non-linear equation, a Newton iteration will be used. This approach requires an initial guess of $\mathbf{V}_{j+1}$. The iteration will continue until a convergence criterion is reached.

For simplification of multiple indices, new notations will be introduced for the penalty iterations only. Let the iteration occur at the $(j+1)$th time step, define $\mathbf{V}^k$ as the kth estimate of $\mathbf{V}_{j+1}$, and $\mathbf{P}^k$ as the penalty for $\mathbf{V}^k$. Note the boundary points in $\mathbf{D}_j$ do not change as these points always satisfy the LCP constraint. The iteration algorithm then follows:

Penalty Iteration for American Options

```
Initialize $\mathbf{V}^0 = \mathbf{V}_j$
For $k = 1, 2, \cdots$
    solve $\left[\mathbf{C}_{im} + \mathbf{P}^k\right]\mathbf{V}^k + \mathbf{D}_{im,j+1} = \mathbf{C}_{ex}\mathbf{V}_j + \mathbf{D}_{ex,j} + \mathbf{P}^k\mathbf{\Phi}$
    if $\left[\max_i \frac{|\mathbf{V}^{k+1} - \mathbf{V}^k|}{\max(1, |\mathbf{V}^{k+1}|)} < \frac{1}{\rho}\right]$ or $[\mathbf{P}^{k+1} = \mathbf{P}^k]$ quit        (3.25)
EndFor
$\mathbf{V}_{j+1} = \mathbf{V}^k$
```

Observe the two convergence criteria. The first criterion stops the iteration when improvements become small. Since the size of ρ determines the size of improvements, it is reasonable to use $\frac{1}{\rho}$ as the tolerance. The second criterion stops the iteration when the penalty matrix no longer changes, which further iterations will yield no difference. It is important to note the convergence of Newton iterations is significantly improved by a good initial guess. In the case of American options, $\mathbf{V}_j$ is usually very close to $\mathbf{V}_{j+1}$, which allows the algorithm to converge within one to two iterations.

3.4 Operator Splitting

In contrast to the penalty iteration, the operator splitting method used in [3] is a direct solver. The key idea is to decouple the complex operators into two intermediate steps. Here the $\mathcal{L}$ operator (2.3) and the LCP (2.9) are treated separately, resulting in a simple correction at each time step. Compared to explicitly setting $V_{i,j} \geq \Phi(S_i)$, this approach involves a Crank-Nicolson intermediate step, which improves the residual term to $O(h_\tau^2 + h_S^2)$.

Similar to the penalty scheme, an additional term is used to resolve the inequality arising from the LCP ($\frac{\partial V}{\partial \tau} - \mathcal{L}V \geq 0$). Let the auxiliary term be

$$\lambda \equiv \frac{\partial V}{\partial \tau} - \mathcal{L}V \tag{3.26}$$

The LCP can be rewritten in terms of λ as

$$\begin{aligned} \lambda &\geq 0, \\ V - \Phi &\geq 0, \\ \lambda(V - \Phi) &= 0 \end{aligned} \tag{3.27}$$

Once again, the PDE recovered with λ can be discretized and rewritten in matrix form

$$(\mathbf{C}_{im}\mathbf{V}_{j+1} + \mathbf{D}_{im,j+1}) - (\mathbf{C}_{ex}\mathbf{V}_j + \mathbf{D}_{ex,j}) - h_\tau \boldsymbol{\lambda}_{j+1} = 0 \tag{3.28}$$

where $\mathbf{V}_{j+1}$ and $\boldsymbol{\lambda}_{j+1}$ are both unknown vectors. Instead of solving the above equation, an intermediate term $\mathbf{U}_j$ is introduced such that,

$$(\mathbf{C}_{im}\mathbf{U}_{j+1} + \mathbf{D}_{im,j+1}) - (\mathbf{C}_{ex}\mathbf{V}_j + \mathbf{D}_{ex,j}) - h_\tau \boldsymbol{\lambda}_j = 0 \tag{3.29a}$$

$$\mathbf{C}_{im}(\mathbf{V}_{j+1} - \mathbf{U}_{j+1}) - h_\tau(\boldsymbol{\lambda}_{j+1} - \boldsymbol{\lambda}_j) = 0 \tag{3.29b}$$

Observe adding (3.29a) and (3.29b) results in the original setup (3.28) . Next (3.29b) is approximated by setting $\theta = 0$, hence $\mathbf{C}_{im}$ becomes the identity matrix, and we get

$$\mathbf{V}_{j+1} - \mathbf{U}_{j+1} - h_\tau(\boldsymbol{\lambda}_{j+1} - \boldsymbol{\lambda}_j) = h_\tau \theta \mathbf{C}(\mathbf{V}_{j+1} - \mathbf{U}_{j+1}) \approx 0 \tag{3.30}$$

As a result, the operator splitting method takes a Crank-Nicolson step to the intermediate term $\mathbf{U}_{j+1}$, then takes an explicit step to the final solution $\mathbf{V}_{j+1}$. In equation form,

$$(\mathbf{C}_{im}\mathbf{U}_{j+1} + \mathbf{D}_{im,j+1}) - (\mathbf{C}_{ex}\mathbf{V}_j + \mathbf{D}_{ex,j}) - h_\tau \boldsymbol{\lambda}_j = 0 \tag{3.31a}$$

$$\mathbf{V}_{j+1} - \mathbf{U}_{j+1} - h_\tau(\boldsymbol{\lambda}_{j+1} - \boldsymbol{\lambda}_j) = 0 \tag{3.31b}$$

Since Equation (3.31b) contains only vectors, it can be detached into independent equations. In other words, for each i, the pair of $V_{i,j+1}$ and $\lambda_{i,j+1}$ can be analyzed as a stand alone equation:

$$V_{i,j+1} - U_{i,j+1} - h_\tau(\lambda_{i,j+1} - \lambda_{i,j}) = 0. \tag{3.32}$$

From the LCP constraints (3.27), we know that for each i, there exist two possible cases, either $V_{i,j+1} - \Phi_i = 0$ or $\lambda_{i,j+1} = 0$. We first suppose $V_{i,j+1} - \Phi_i = 0$, from Equation (3.32) we get

$$\begin{aligned} V_{i,j+1} - \Phi_i &= U_{i,j+1} - \Phi_i + h_\tau(\lambda_{i,j+1} - \lambda_{i,j}) = 0 \\ \lambda_{i,j+1} &= \lambda_{i,j} - \frac{1}{h_\tau}(U_{i,j+1} - \Phi_i) \end{aligned} \tag{3.33}$$

Note it is possible to have $\lambda_{i,j+1} < 0$ in computation if λ decreases to zero in between two time steps, i.e. $\tau \in (\tau_j, \tau_{j+1})$. Due to the LCP constraints (3.27), the (continuous) λ stops decreasing once it reaches zero, therefore has a value of $\lambda_{i,j+1} = 0$. Also note since the alternate case leads to $\lambda_{i,j+1} = 0$, we get an explicit solution

$$\lambda_{i,j+1} = \max\left(\lambda_{i,j} - \frac{1}{h_\tau}\big[U_{i,j+1} - \Phi_i\big], 0\right) \tag{3.34}$$

Alternatively, we suppose $\lambda_{i,j+1} = 0$. Similarly, from Equation (3.32) we get

$$V_{i,j+1} = U_{i,j+1} - h_\tau\lambda_{i,j} \tag{3.35}$$

Once again, it is possible to have $V_{i,j+1} - \Phi_i < 0$ in computation. Therefore we apply the LCP constraints (3.27) to get

$$V_{i,j+1} = \max\left(U_{i,j+1} - h_\tau\lambda_{i,j}, \Phi_i\right) \tag{3.36}$$

Observe that if $\lambda_{i,j+1} > 0$, we have the first case where $V_{i,j+1} - \Phi_i = 0$. This leads to

$$V_{i,j+1} = U_{i,j+1} + h_\tau(\lambda_{i,j+1} - \lambda_{i,j}) = \Phi_i \tag{3.37}$$

Therefore, there exist an explicit solution that satisfy both scenarios:

$$V_{i,j+1} = \max\left(U_{i,j+1} + h_\tau(\lambda_{i,j+1} - \lambda_{i,j}), \Phi_i\right) \tag{3.38}$$

Altogether, we can find the intermediate term $\mathbf{U}_{j+1}$, the auxiliary term $\lambda_{i,j+1}$, and the price term $\mathbf{V}_{j+1}$ explicitly

$$\begin{aligned} \mathbf{U}_{j+1} &= \mathbf{C}_{im}^{-1}\big[\mathbf{C}_{ex}\mathbf{V}_j + \mathbf{D}_{ex,j} + \mathbf{D}_{im,j+1} + h_\tau\lambda_j\big] \\ \lambda_{i,j+1} &= \max\left(\lambda_{i,j} - \frac{1}{h_\tau}\big[U_{i,j+1} - \Phi_i\big], 0\right) \\ V_{i,j+1} &= \max\left(U_{i,j+1} + h_\tau(\lambda_{i,j+1} - \lambda_{i,j}), \Phi_i\right) \end{aligned} \tag{3.39}$$

In summary, the penalty method takes a direct route to the solution, but the process may require more than one iteration per time step, therefore more than one set of computations (a set of computations being mainly the solution of a tridiagonal system). In contrast, operator splitting takes an indirect route, but only requires one set of computations. This trade-off allows the splitting method to work with more grid points in the space direction and more steps in the time direction at the same computation cost as the penalty scheme. The next section describes how the two approaches compare numerically.

Chapter 4

Results

4.1 European Options

The PDE solver was implemented and tested using MATLAB. Numerical experiments were conducted with the following parameters on an European Put option:

$$\begin{aligned} \text{Put Option: } & K = 100, T = 0.25, \sigma = 0.8, r = 0.1 \\ \text{Finite Difference: } & \theta = 0.5 \\ \text{Grid: } & S_{\max} = 1000, c_0 = 20, d_0 = 1 \\ & d_{\text{norm}} = [0.003, 0.0015, 0.00075, \ldots] \end{aligned} \tag{4.1}$$

Recall d_{norm} is halved each time the number of time steps is expected to approximately double. For the adaptive time steps, the initial time step is chosen by $h_{\tau_0} = h_{\tau,uniform}/1000$, and d_{norm} will dominate the rate of step size growth. Hence each experiment using adaptive time steps correspond to an experiment using uniform time steps. Rannacher smoothing is also used for all the experiments.

Here, we introduce V_{Change} as the difference between the values of V at a specific point obtained in the past two experiments with increasingly finer grids, and V_{Ratio} as the ratio of the past two V_{Change}.

We also introduce the price derivatives, also known as the Greeks:

$$\begin{aligned}\text{Delta: } \Delta &= \frac{\partial V}{\partial S} \\ \text{Gamma: } \Gamma &= \frac{\partial^2 V}{\partial S^2}.\end{aligned} \tag{4.2}$$

Here, Δ_{Change}, Δ_{Ratio}, Γ_{Change}, and Γ_{Ratio} are defined in the same sense as V_{Change} and V_{Ratio}. The Greeks are commonly used parameters in options trading, hence their accurate calculation is just as important as that of the price.

We also introduce N_{it} as the number of iterations, where each iteration involve the solution of a tridiagonal matrix system. Notice that $N_{it} \geq N_\tau$ when the penalty iteration method is used. The *computation cost* is defined as the number of price grid points N_S (cost of inverting a tridiagonal matrix is $O(N_S)$) multiplied by the number of iterations taken in the run N_{it}.

A small adjustment to c_0 is applied to avoid the strike K landing on a specific grid point. This is done by a local search through nearby points of c_0 that places the strike K closest to the midpoint between two grid points. Most adjustments are less than 10%, therefore having minimal effect on concentration.

In Tables 4.1 to 4.5, we have numerical results for the European put at $S = K = 100$ (at-the-money). Observe also in Figure 4.1, both uniform and non-uniform grids converge at the same rate. While the non-uniform grid provides the same order of convergence, the error is significantly improved for V, Δ, and Γ.

Observe when both experiments used uniform time steps, V_{Change} is significantly improved with $W_2(S)$ scheme for non-uniform grid in Table 4.2, compared the uniform grid in Table 4.1. Furthermore, V_{Change} is improved again by using adaptive time stepping in 4.3.

We also observe the behaviour of adaptive time steps over time in Figure 4.2. The time steps is observed to increase aggressively in the beginning, as the initial time step at approximately 10^{-7} increased very quickly to the scale of the plot at approximately 10^{-4}. This explosive growth then slowed down drastically as the time increased and the time derivative decreased in value.

N_S	N_τ	N_{it}	V	V_{Change}	V_{Ratio}
54	25	25	14.603887		
108	50	50	14.519163	-0.084724	
216	100	100	14.435627	-0.083536	1.01
432	200	200	14.447840	0.012213	-6.84
864	400	400	14.450890	0.003049	4.01
1728	800	800	14.451652	0.000762	4.00

Table 4.1: Values of European put option with parameters (4.1) at-the-money using uniform S grid and uniform time steps.

N_S	N_τ	N_{it}	V	V_{Change}	V_{Ratio}
54	25	25	14.435351		
108	50	50	14.448452	0.013101	
216	100	100	14.451101	0.002649	4.95
432	200	200	14.451712	0.000611	4.34
864	400	400	14.451858	0.000146	4.18
1728	800	800	14.451894	0.000036	4.12

Table 4.2: Values of European put option with parameters (4.1) at-the-money using $W_2(S)$-grid and uniform time steps.

N_S	N_τ	N_{it}	V	V_{Change}	V_{Ratio}
54	38	38	14.442542		
108	62	62	14.450322	0.007781	
216	110	110	14.451575	0.001252	6.21
432	207	207	14.451831	0.000257	4.88
864	403	403	14.451888	0.000057	4.50
1728	795	795	14.451901	0.000013	4.36

Table 4.3: Values of European put option with parameters (4.1) at-the-money using $W_2(S)$-grid, and adaptive time stepping with initial time step $h_{\tau_0} = h_{\tau,uniform}/1000$.

N_S	N_τ	N_{it}	Δ	Δ_{Ratio}	Γ	Γ_{Ratio}
54	25	25	-0.40364025		0.0096432396	
108	50	50	-0.39844661		0.0096300076	
216	100	100	-0.39674885	3.06	0.0096482509	-0.73
432	200	200	-0.39653807	8.05	0.0096388955	-1.95
864	400	400	-0.39648550	4.01	0.0096365650	4.01
1728	800	800	-0.39647237	4.00	0.0096359828	4.00

Table 4.4: Delta and Gamma values of European put option with parameters (4.1) at-the-money using uniform S grid and uniform time steps.

N_S	N_τ	N_{it}	Δ	Δ_{Ratio}	Γ	Γ_{Ratio}
54	38	38	-0.39680141		0.0096237397	
108	62	62	-0.39655588		0.0096323359	
216	110	110	-0.39649054	3.76	0.0096349288	3.32
432	207	207	-0.39647370	3.88	0.0096355739	4.02
864	403	403	-0.39646943	3.94	0.0096357352	4.00
1728	795	795	-0.39646835	3.96	0.0096357753	4.02

Table 4.5: Delta and Gamma values of European put option with parameters (4.1) at-the-money using $W_2(S)$-grid, and adaptive time stepping with initial time step $h_{\tau_0} = h_{\tau,uniform}/1000$.

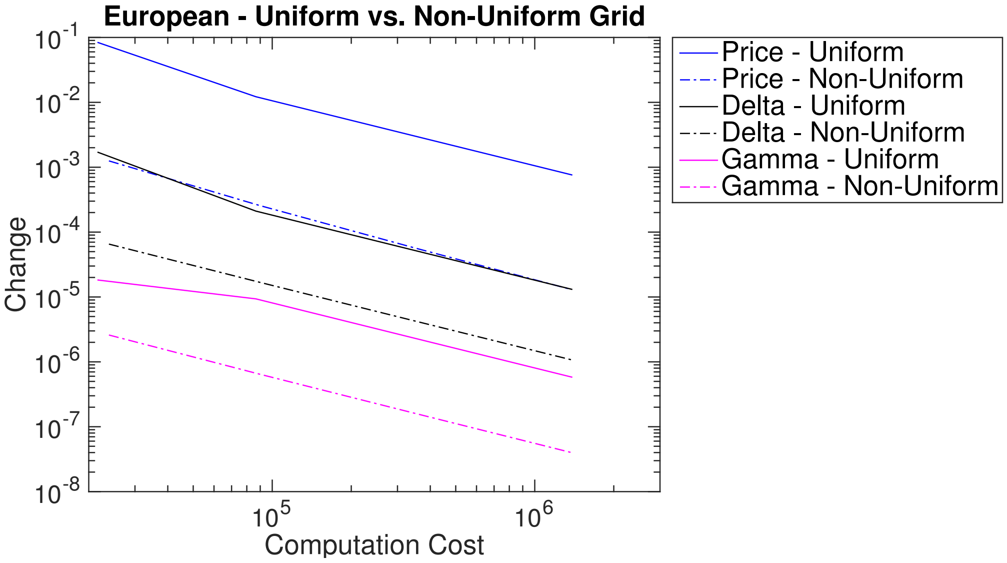

Figure 4.1: Performance comparison between the uniform and non-uniform grids for computing the Price, Delta, and Gamma values at-the-money of the European put option with parameters (4.1) on a log-log plot. Here, "uniform" applies to both the S and time directions, while "non-uniform" means that the $W_2(S)$-grid and adaptive time stepping were used with initial time steps $h_{\tau_0} = h_{\tau,uniform}/1000$.

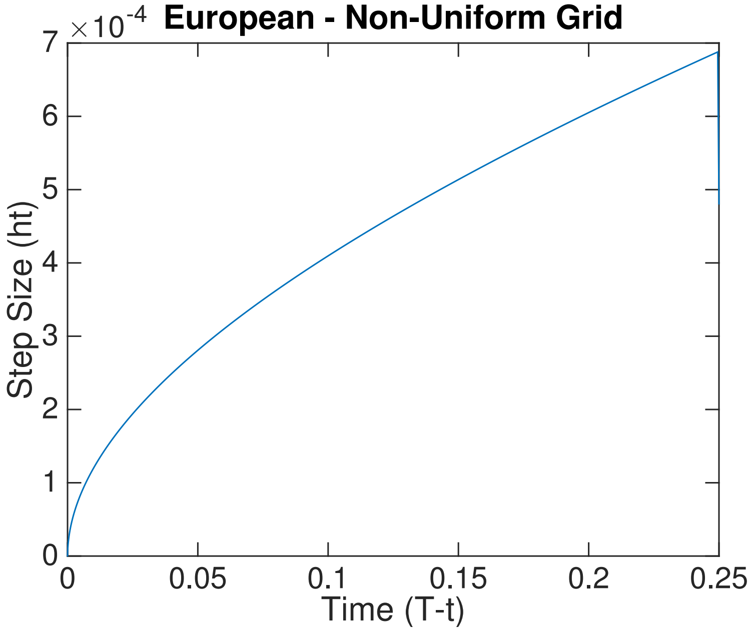

Figure 4.2: The size of time steps over time for the adaptive time step method for the European put with parameters (4.1) using initial time step $h_{\tau_0} = 3.125 \times 10^{-7}$ and $d_{norm} = 1.375 \times 10^{-4}$.

4.2 American Options

Similar numerical experiments were conducted with the following parameters on an American Put option:

$$\begin{aligned} \text{Put Option: } & K = 100, T = 0.25, \sigma = 0.8, r = 0.1 \\ \text{Finite Difference: } & \theta = 0.5, \rho = 1e6 \\ \text{Grid: } & S_{\max} = 1000, c_0 = 20, d_0 = 1 \\ \text{Penalty: } & d_{\text{norm}} = [0.003, 0.0015, 0.00075, \ldots] \\ \text{Operator: } & d_{\text{norm}} = [0.0022, 0.0011, 0.00055, \ldots] \end{aligned} \tag{4.3}$$

We chose different values of d_{norm} for the two methods so that the number of iterations N_{it} are closer and easier to compare. Rannacher smoothing is also used for all the experiments.

Here, we introduce $S_{f,Change}$ and $S_{f,Ratio}$ for the free boundary in the same sense as V_{Change} and V_{Ratio}.

First, we observe that the Penalty and Operator Splitting methods behave quite similarly on uniform grids as shown in Figure 4.3. In Tables 4.6 to 4.7, we can observe a slightly better V_{Change} for the operator splitting method compared to penalty given less computation cost. The two methods were able to find the exact same free boundaries S_f given the same grid in Tables 4.8 to 4.9. In the same tables, we also observe that the Delta and Gamma values computed by the two methods are very close to each other.

N_S	N_τ	N_{it}	V	V_{Change}	V_{Ratio}
54	25	28	14.794245		
108	50	55	14.729485	-0.064760	
216	100	111	14.657730	-0.071755	0.90
432	200	222	14.673097	0.015367	-4.67
864	400	445	14.677253	0.004156	3.70
1728	800	894	14.678396	0.001143	3.64

Table 4.6: Penalty iteration results for values of American put option with parameters (4.3) at-the-money using uniform S grid and uniform time steps.

Next from Figure 4.4 we infer the non-uniform grid scheme consistently improves the error of V, Δ, and Γ for the penalty scheme. However, we note that with the non-uniform grid, S_f did not achieve a significant improvement over the uniform grid case. This is most likely due to the concentration of grid points being focused on the strike rather than on the free boundary. In this case, the two values $S_f \approx 51$ and $K = 100$ are quite far apart, resulting in the free boundary losing in accuracy.

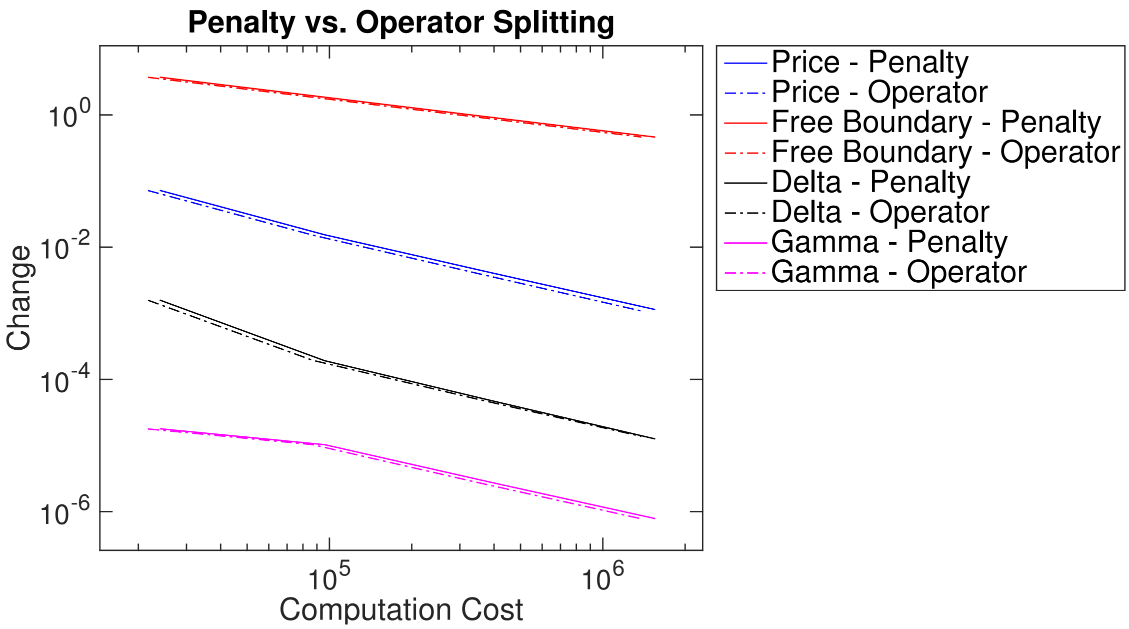

Figure 4.3: Performance comparison between the penalty and the operator splitting methods on a uniform S grid and uniform time stepping for computing the Price, free boundary, Delta, and Gamma values at-the-money of the American put option with parameters (4.3) on a log-log plot.

N_S	N_τ	N_{it}	V	V_{Change}	V_{Ratio}
54	25	25	14.794513		
108	50	50	14.729943	-0.064570	
216	100	100	14.658126	-0.071817	0.90
432	200	200	14.673365	0.015239	-4.71
864	400	400	14.677389	0.004024	3.79
1728	800	800	14.678471	0.001082	3.72

Table 4.7: Operator splitting results for values of American put option with parameters (4.3) at-the-money using uniform S grid and uniform time steps.

N_S	N_τ	N_{it}	S_f	$S_{f,Ratio}$	Δ	Δ_{Ratio}	Γ	Γ_{Ratio}
54	25	28	59.2593		-0.41252800		0.0100255188	
108	50	55	59.2593		-0.40745545		0.0100202263	
216	100	111	55.5556		-0.40588568	3.23	0.0100381027	-0.30
432	200	222	53.7037	2.00	-0.40569427	8.20	0.0100278161	-1.74
864	400	445	52.7778	2.00	-0.40564580	3.95	0.0100250119	3.67
1728	800	894	52.3148	2.00	-0.40563316	3.84	0.0100242240	3.56

Table 4.8: Penalty iteration results for free boundary, Delta, and Gamma values of American put option with parameters (4.3) at-the-money using uniform S grid and uniform time steps.

N_S	N_τ	N_{it}	S_f	$S_{f,Ratio}$	Δ	Δ_{Ratio}	Γ	Γ_{Ratio}
54	25	25	59.2593		-0.41254621		0.0100259027	
108	50	50	59.2593		-0.40746724		0.0100202821	
216	100	100	55.5556		-0.40589316	3.23	0.0100380743	-0.32
432	200	200	53.7037	2.00	-0.40569828	8.08	0.0100277540	-1.72
864	400	400	52.7778	2.00	-0.40564787	3.87	0.0100249811	3.72
1728	800	800	52.3148	2.00	-0.40563413	3.67	0.0100242023	3.56

Table 4.9: Operator splitting results for free boundary, Delta, and Gamma values of American put option with parameters (4.3) at-the-money using uniform S grid and uniform time steps.

Ideally, we would like adaptive changes in the S grid such that there is consistent concentration of points near the value of $S_f(\tau)$ as τ varies. This type of grid will ultimately improve the accuracy of S_f at a cost of some increase in computation and complexity. In [1], certain adaptive S-grid techniques are used with the penalty method, and indicate that two concentrations of S-points, one for the strike and one for the free boundary (which moves to the left over time) may better capture the behaviour of the American price problem.

When comparing the two methods, namely penalty and operator splitting with the non-uniform scheme, we can observe that the operator splitting method begins to show some instability. In Table 4.11, we can see the operator splitting method is not converging monotonically. Although operator splitting can exhibit better V_{Change} as seen in Figure 4.5, the behavior is not as consistent. Otherwise, the accuracies of both methods are quite comparable.

One important distinction between the two methods is the number of linear system solutions. The penalty method is iterative, sometimes requiring the solution of more one linear systems. On the other hand, the operator splitting method is a direct solver, requiring only one linear system solution. While the difference is minor in one price dimension, operator splitting will naturally scale better in higher dimensions.

Finally we study the behavior of $S_f(\tau)$ over time. Both methods showed consistent shape of the free boundary in Figure 4.6. Note the plot used an experiment with very sparse grid for demonstration purposes. In practice the grid can be much more refined. For the penalty method, the movement in $S_f(\tau)$ is driven entirely by the penalty matrix $\mathbf{P}_j$, while for the operator splitting method by the auxiliary term λ_j. The diagonal of the penalty matrix $\mathbf{P}_j$ penalizes the region right of $S_f(\tau)$ for falling below the exercise value $\Phi(S)$, therefore the value of S corresponding to the diagonal entry of $\mathbf{P}_j$ switching from ρ to 0 marks the free boundary $S_f(\tau)$. Observe Figure 4.7, the diagonal of $\mathbf{P}_j$ takes discrete values of either 0 or ρ, and the boundary dividing the discrete values is exactly the free boundary $S_f(\tau)$.

Not surprisingly, in the operator splitting method, the auxiliary term λ gives rise to a surface very similar to the diagonal of $\mathbf{P}_j$, as shown in Figure 4.8. Observe here majority of the points $\lambda_{i,j}$ takes a value of 0 or 10 as well, however without a sharp edge compared to the diagonal of the penalty matrix. The reason for the initialization of $\lambda \approx 10$ is due to its approximation of the time derivative

$$\begin{aligned} \lambda_{i,1} &= \max\left(-\frac{U_{i,1}-\Phi_i}{h_\tau}, 0\right) \\ &\approx \max\left(-\left.\frac{\partial V}{\partial \tau}\right|_{i,0}, 0\right) \\ &= \max\left(-\left[\frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV\right]_{i,0}, 0\right) \end{aligned} \tag{4.4}$$

For $S_i < K$, the price of the put is approximately $V_{i,0} \approx K - S_i$, which leads to

$$\begin{aligned} \lambda_{i,1} &\approx \max\left(0 - rS_i(-1) + r(K - S_i), 0\right) \\ &= rK \\ &= 10 \end{aligned} \tag{4.5}$$

which is the value of λ observed in Figure 4.8 when it is non-zero.

N_S	N_τ	N_{it}	V	V_{Change}	V_{Ratio}
54	38	51	14.666084	0.000000	0.00
108	62	88	14.676292	0.010207	0.00
216	110	162	14.678275	0.001984	5.15
432	208	313	14.678733	0.000458	4.33
864	405	615	14.678842	0.000109	4.22
1728	799	1183	14.678869	0.000027	4.06

Table 4.10: Penalty iteration results for values of American put option with parameters (4.3) at-the-money using $W_2(S)$-grid and adaptive time stepping with initial time steps $h_{\tau_0} = h_{\tau,uniform}/1000$.

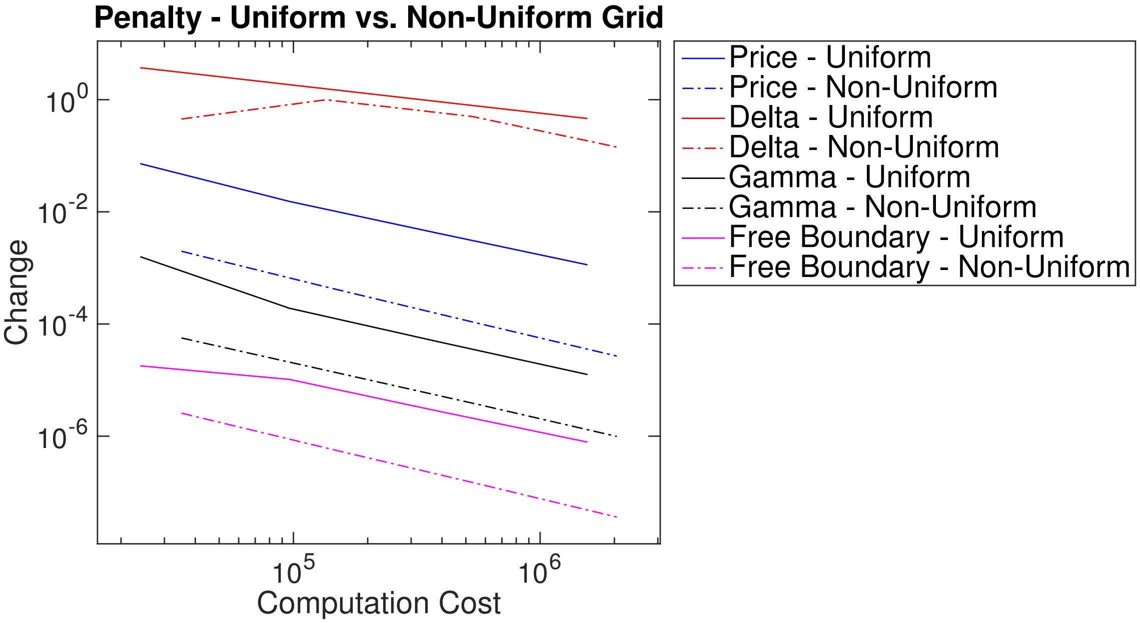

Figure 4.4: Performance comparison between the uniform and non-uniform grids using the penalty method for computing the Price, free boundary, Delta, and Gamma values at-the-money of the American put option with parameters (4.3) on a log-log plot. Here, the $W_2(S)$-grid and adaptive time stepping were used with initial time steps $h_{\tau_0} = h_{\tau,uniform}/1000$.

N_S	N_τ	N_{it}	V	V_{Change}	V_{Ratio}
54	46	46	14.667658	0.000000	0.00
108	78	78	14.677313	0.009656	0.00
216	145	145	14.678802	0.001489	6.48
432	279	279	14.678981	0.000179	8.31
864	547	547	14.678952	-0.000029	-6.20
1728	1085	1085	14.678916	-0.000036	0.79

Table 4.11: Operator splitting results for values of American put option with parameters (4.3) at-the-money using $W_2(S)$-grid and adaptive time stepping with initial time step $h_{\tau_0} = h_{\tau,uniform}/1000$.

N_S	N_τ	N_{it}	S_f	$S_{f,Ratio}$	Δ	Δ_{Ratio}	Γ	Γ_{Ratio}
54	38	51	60.0120	0.00	-0.40591239	0.00	0.0100117440	0.00
108	62	88	54.1134	0.00	-0.40570493	0.00	0.0100204951	0.00
216	110	162	53.6617	13.06	-0.40564854	3.68	0.0100230679	3.40
432	208	313	52.6657	0.45	-0.40563361	3.78	0.0100236843	4.17
864	405	615	52.1655	1.99	-0.40562973	3.85	0.0100238336	4.13
1728	799	1183	52.0223	3.50	-0.40562874	3.92	0.0100238697	4.14

Table 4.12: Penalty iteration results for free boundary, Delta, and Gamma values of American put option with parameters (4.3) at-the-money using $W_2(S)$-grid and adaptive time stepping with initial time step $h_{\tau_0} = h_{\tau,uniform}/1000$.

N_S	N_τ	N_{it}	S_f	$S_{f,Ratio}$	Δ	Δ_{Ratio}	Γ	Γ_{Ratio}
54	46	46	60.0120	0.00	-0.40597656	0.00	0.0100150027	0.00
108	78	78	54.1134	0.00	-0.40573957	0.00	0.0100218326	0.00
216	145	145	53.6617	13.06	-0.40566477	3.17	0.0100236014	3.86
432	279	279	52.6657	0.45	-0.40564086	3.13	0.0100238967	5.99
864	547	547	52.1655	1.99	-0.40563284	2.98	0.0100239172	14.34
1728	1085	1085	52.0223	3.50	-0.40563003	2.86	0.0100239026	-1.40

Table 4.13: Operator splitting results for free boundary, Delta, and Gamma values of American put option with parameters (4.3) at-the-money using $W_2(S)$-grid and adaptive time stepping with initial time step $h_{\tau_0} = h_{\tau,uniform}/1000$.

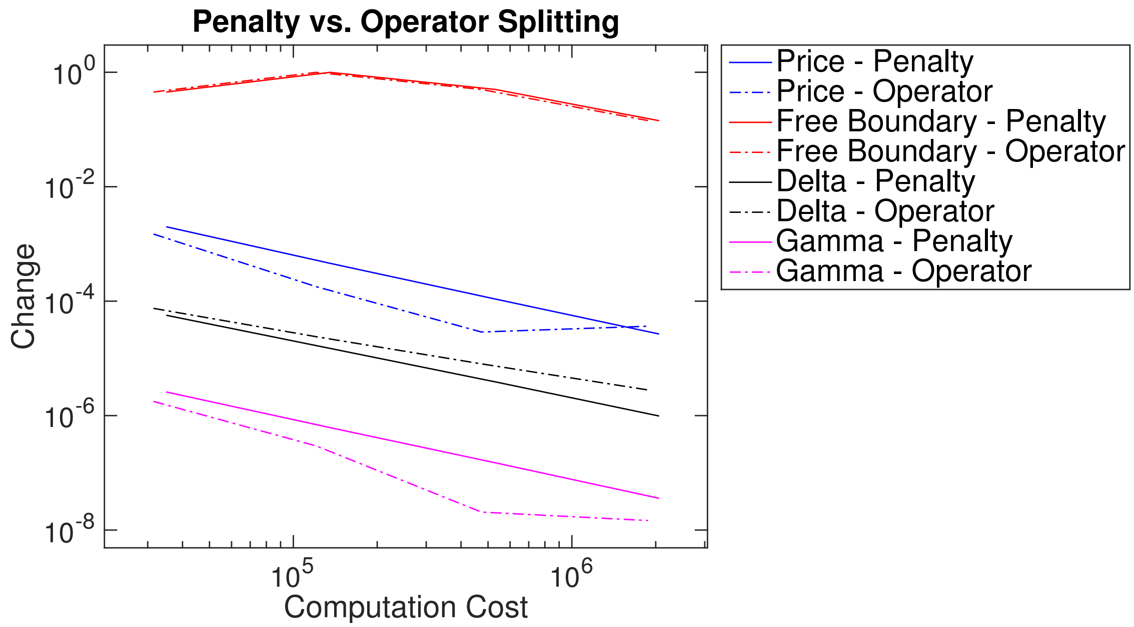

Figure 4.5: Performance comparison between the penalty and operator splitting methods on a $W_2(S)$-grid with adaptive time stepping for computing the Price, free boundary, Delta, and Gamma values at-the-money of the American put option with parameters (4.3) on a log-log plot. The initial time step was $h_{\tau_0} = h_{\tau,uniform}/1000$.

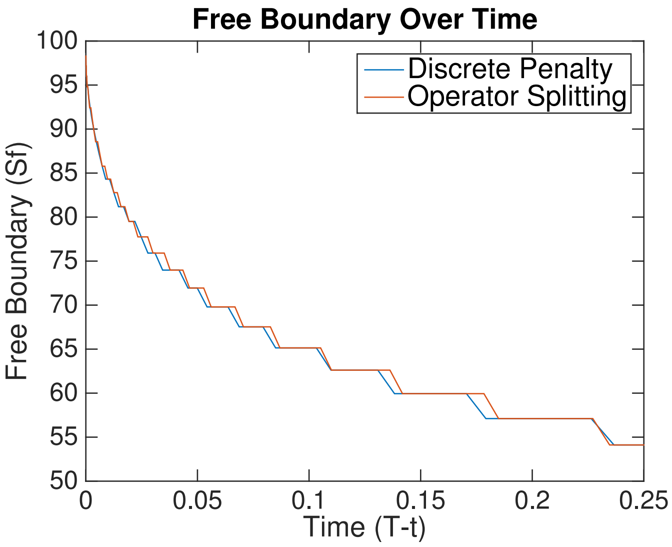

Figure 4.6: The free boundary is plotted over time for both the penalty and operator splitting methods solving the American put option problem with parameters (4.3). Here, the $W_2(S)$-grid and adaptive time stepping were used. The specific penalty method experiment corresponds to parameters $N_S = 109$, $N_\tau = 62$, and $d_{norm} = 0.0011$, or row 2 of Table 4.10. The specific operator splitting experiment corresponds to parameters $N_S = 109$, $N_\tau = 78$, and $d_{norm} = 0.0011$, or row 2 of Table 4.11.

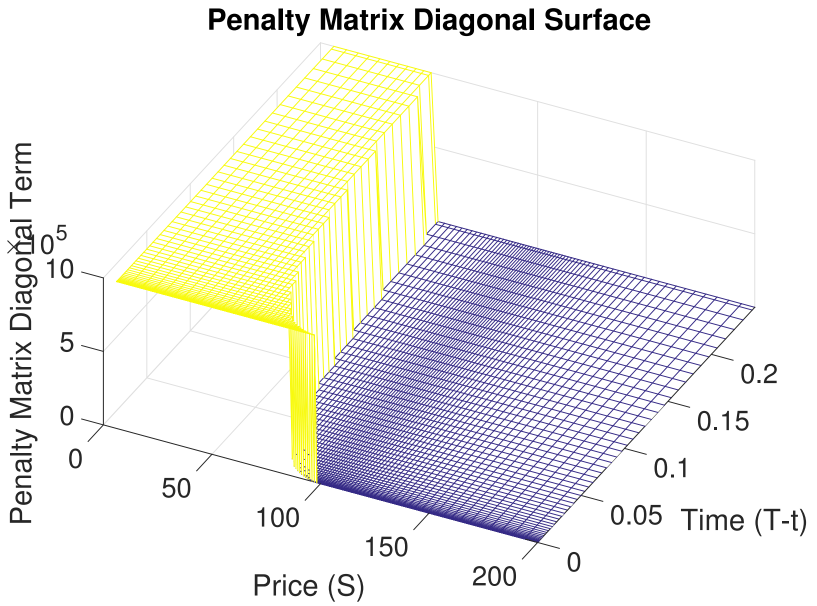

Figure 4.7: The diagonal of the penalty matrix $\mathbf{P}_j$ is plotted as a surface over time and price for the penalty method solving the American put option problem with parameters (4.3). Here, the $W_2(S)$-grid and adaptive time stepping were used. The specific experiment corresponds to parameters $N_S = 109$, $N_\tau = 62$, and $d_{norm} = 0.0011$, or row 2 of Table 4.10.

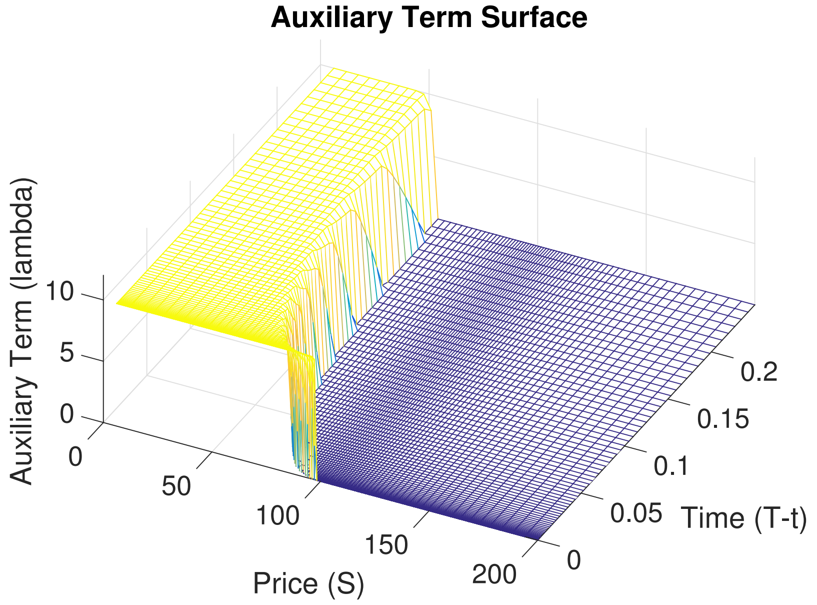

Figure 4.8: The auxiliary term $\lambda_{i,j}$ is plotted as a surface over time and price for the operator splitting method solving the American put option problem with parameters (4.3). Here, the $W_2(S)$-grid and adaptive time stepping were used. The specific experiment corresponds to parameters $N_S = 109$, $N_\tau = 78$, and $d_{norm} = 0.0011$, or row 2 of Table 4.11.

Chapter 5

Conclusions and Future Works

In this thesis, we have studied several numerical methods for pricing European and American options in detail. First, we studied the finite difference method for pricing the European option through the solution of the Black-Scholes PDE. By increasing the computation cost, we have shown the error in price and the Greeks are converging at a consistent rate. Furthermore, the use of non-uniform grid and adaptive time steps significantly improve the absolute error at the same computation cost.

We then studied the penalty iteration and operator splitting methods for pricing the American put option. While both methods showed consistent convergence on uniform grids in price and time, the penalty method has shown more stable convergence rate when using non-uniform grid and adaptive time steps. Overall, the two methods are quite comparable in terms of absolute error in price, free boundary, and the Greeks.

There are several areas of possible extensions to this study. Since the penalty iteration requires more than one solution of a linear system per time step, its computational cost is expected to scale faster in multiple dimensions than the operator splitting method, which requires only one linear system solution per time step. As a result, the operator splitting method could potentially improve the performance of a high dimensional pricing problem.

Additionally, there is no available guideline to optimally select parameters for the non-uniform grid and adaptive time steps. Optimization of these parameters will continue to improve the error beyond the current results. The use of an adaptive grid seen in [1] could also be explored to improve the accuracy of the free boundary. Finally, many techniques studied in this thesis is not limited to European and American options, but also applicable for a wide range of derivative products.

Bibliography

[1] C. Christara and D. M. Dang. Adaptive and high-order methods for valuing American options. *J. Comput. Finance*, 14(4):73–113, Summer 2011.

[2] P. A. Forsyth and K. R. Vetzal. Quadratic convergence for valuing American options using a penalty method. *SIAM J. Sci. Comput.*, 23(6):2095–2122, 2002.

[3] S. Ikonen and J. Toivanen. Operator splitting methods for American option pricing. *Appl. Math. Let.*, 17:809–814, 2004.

[4] R. Rannacher. Finite element solution of diffusion problems with irregular data. *Numer. Math.*, 43:309–327, 1984.

[5] P. Wilmott, S. Howison, and J. Dewynne. *The Mathematics of financial derivatives: a student introduction*. Cambridge University Press, 1995.