Hedging and Pricing in Non-probabilistic Models with Transaction Costs.

by

Massoome Rahsepar

Master of Science, Isfahan University of Technology , 2004

A thesis

presented to Ryerson University

in partial fulfillment of the

requirements for the degree of

Master of Science

in the Program of

Applied Mathematics

Toronto, Ontario, Canada, 2011

I hereby declare that I am the sole author of this thesis.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Hedging and Pricing in Non-probabilistic Models with Transaction Costs.

Master of Science 2011

Massoome Rahsepar

Applied Mathematics

Ryerson University

The thesis extends a non probabilistic market model proposed by Britten-Jones and Neuberger by incorporating transaction costs into their model. The original model is of rather general applicability as it incorporates the discrete nature of the market by allowing only a finite number of transactions and discrete jumps and requires few observable parameters to be deployed.

Our addition of transaction costs gives the model an even more realistic character and, in this way, allows to use the model as an instrument to look for arbitrage opportunities in the market. The main output of the resulting model is a pair of numbers acting as lower and upper bounds to prices of financial instruments. The thesis does perform a limited search for arbitrage opportunities in market data and finds several interesting phenomena. A detailed analysis of several analytical properties, optimization and computational issues, along with a software implementation, are also fully developed in the thesis.

Keywords: Incomplete market, No-arbitrage bounds, Proportional transaction costs, Fixed transaction costs.

Dedication

This thesis is dedicated to my wonderful husband Ali who has been a great source of motivation and inspiration. And to my parents Farkhonde and Hassan and my sister Bahar who have supported me all the way since the beginning of my studies.

"To us, family means putting your arms around each other and being there."

—Barbara Bush

4.1.2 J-N Discretization for a Special Case . . . 40
4.1.3 J-N Discretization for the General Case . . . 40
4.1.4 Option Price at Each Stage in J-N Algorithm . . . 42
4.1.5 Optimal Hedging and Convex Hull . . . 44

5 Numerical Analysis **49**
5.1 Artificial Paths and General Results . . . 49
5.1.1 Comparison to Black-Scholes . . . 50
5.1.2 Arbitrage Bounds: Basic Model . . . 51
5.1.3 Arbitrage Bounds: Transaction Costs Model . . . 55
5.2 Market Data and No-Arbitrage Bounds . . . 59
5.2.1 Forecasting the Quadratic Variation . . . 61
5.2.2 Investment Strategy . . . 65
5.2.3 Smile Quadratic Variation . . . 70

Conclusions **73**

Bibliography **100**

List of Tables

5.1 No arbitrage bounds with different the jump units. 53
5.2 No arbitrage bounds with different the jump units for a derivative with a butterfly payoff. 54
5.3 No arbitrage bounds with different the jump units for a derivative with a strangle payoff. 54
5.4 The convergence behavior of the proportional transaction cost model as the number of trades N_2 increases and also the relation between δ and round-trip α 59
5.5 This table shows the option price with different strike prices. 62
5.6 profits/losses for price path of Yahoo inc! 70

List of Figures

1.1 The boundary region for a call option's prices 3

2.1 Case A: A lazy permissible path 8

2.2 Case B: A possible permissible path 8

4.1 The sample grid with $N = 11$ steps tree construction 39

4.2 All destination points when $n = 3$ 41

4.3 All destination points for the last two stages 42

4.4 Multiple destination points from fourth to last step and beyond 43

4.5 Convex Hull optimization and the Hull Line 45

5.1 The minimum upper bound $\overline{V}$ as a function N_2 when $\frac{d}{\delta} = 1$ 50

5.2 Convergence of $\overline{V}$ as a function N_2 for different values of $\frac{d}{\delta}$. 51

5.3 The minimum upper bound and the maximum lower bound for the basic model with different jump units 52

5.4 A Butterfly and Long Strangle payoffS 53

5.5 The upper and lower bounds for a derivative with a Butterfly and Long Strangle payoffS with different jump units 54

5.6 Upper and lower bounds for the fixed transaction costs $\mathcal{K}$ 55

5.7 Upper and lower bounds become wider by increasing the rate $\mathcal{K}$ with the fixed jump units $\frac{d}{\delta} = 3$. 56

5.8 This plot shows the option bounds when the fixed transaction cost is $10 by assuming 1000 call options. 57

5.9 The minimum upper bound $\overline{V}$ as a function of the quadratic variation in both case, with and without transaction costs. 58

5.10 The minimum upper and the maximum lower bounds for the proportional transaction costs. 60

5.11 Time Series: The sequence of closing price between May 2009-May 2011 for the chosen stock is shown in this figure 60

5.12 This figure shows the quadratic variation for different windows 1, .., 16 which start at times t_s such that $S_0 = S(t_s) \approx 16.69$ and all window's sizes are $N_2 = 43$ 61

5.13 The minimum upper bound for the maximum quadratic variation 0.0326, in the case of no jump occurs. 63
5.14 The minimum upper bound with no jumps and maximum jump units equal to 6 compared with the real option value show in this figure. 63
5.15 This plot shows the upper bound with the minimum fixed transaction $\mathcal{K} = 0.045$ which can remove the arbitrage in this market for all strike price. 64
5.16 In the case of proportional transaction costs, we are not able to remove the gap, because of limitation on $\alpha < 0.027$. 65
5.17 Yahoo! Inc.path price and The J-N price path 67
5.18 Yahoo! Inc.path price and The J-N price path 68
5.19 Yahoo! Inc.path price and The J-N price path when $N_2 = 80$ and $v_0 = Qv_{max}$ 69
5.20 Yahoo! Inc.path price and The J-N price path 69
5.21 The behavior of the quadratic variation as the strike price increases with fixed transaction costs. 71
5.22 The behavior of the quadratic variation as the strike price increases with proportional transaction cost. 72

List of Appendices

A Some Basic Concepts in Finance **75**

B Binomial Tree Model **77**

C Dynamic Programming **81**

C.1 What Is Dynamic Programming? 81

C.2 Analytical Concepts in DP 81

C.2.1 Utility or payoff Function 82

C.2.2 Control and State Variable 82

C.2.3 Policy 83

C.2.4 Basic Model 83

D The Optimal Hedging **87**

D.1 Basic Model 87

D.2 Proportional Transaction Costs 89

E MATLAB Code **91**

Chapter 1

Introduction

1.1 Background Material

A fundamental question in finance is to determine the value of a financial option today[1] Black and Scholes [Black and Scholes, 1973] studied this question in a path breaking paper which showed how one can replicate the payoff of an option using a dynamic trading strategy. Based on this results they provided a unique value of the option. Their approach uses a no arbitrage condition and is based on the assumption that the logarithm of the stock price follows a certain continuous time version of a random walk. They also assume that the stock and the risk free bond can be traded continuously.

The Black-Scholes model provides the unique price of a contingent claim in an ideal, complete and unconstrained market based on the fundamental principle of absence of arbitrage opportunities. In other words, this price is the unique one for which there are no arbitrage opportunities by taking either a short or long position in the claim and investing wisely in the market. The Black-Scholes model also provides a single hedging portfolio which one can use to exactly duplicate the claim.

However, the assumptions of the model may not be closely met in practice. This could be because markets are incomplete. Market incompleteness generally comes from two main resources [Hao, 2008]:

1. There are not enough assets in the market to span the uncertainty.
2. Trading strategies are limited or not ideal because of discrete trading, jumps or transaction costs, short selling constraints, etc.

In incomplete markets, instead of a single arbitrage-free price there appears an arbitrage-free interval $[\underline{V}, \overline{V}]$ which contains a spectrum of market prices. Here, $\underline{V}$ is the greatest price the buyer can afford to pay without risk and $\overline{V}$ represents the least price the seller can accept without risk. This interval has the following properties:

[1] For the reader's convenience, Appendix A provides a brief introduction to the basic finance concepts which are used in this thesis.

- Every price level outside the interval leads to an arbitrage opportunity.

- There are no arbitrage opportunities for price level in the interior of the interval.

There is no single widely accepted method to calculate bounds prices in an incomplete market. One approach is to find a super-replicating strategy which is a portfolio whose payoffs are always at least as big as the payoff of the option being hedged. The value of the option is then bounded by the value of the super-replicating portfolio. The super-replication price is the minimal initial wealth needed to hedge without risk the contingent claim. A second approach is the utility-based valuation; in this case the investor of the contingent claim assumes unhedgable risk, which will affect the probability distribution of his/her consumption and final wealth level, and his/her utility function.

Pricing bounds are useful in many situations in which a relative pricing approach is appropriate but perfect replication is not possible. A few examples follow [Hao, 2008]:

1. A trader can use the bounds as buy and sell points in the search for good deals in asset markets.

2. A bank or other institution that markets or synthesizes non traded securities can use bounds as bid and ask prices.

3. Bounds can be used as economic measure of the accuracy of an option pricing formula.

4. Option pricing formulas are often used in risk assessments to quantify the exposure of a position to various risk factors. It is useful to assess such risks when perfect replication is impossible and to quantify the importance of the market price of risk assumptions.

Obtaining bounds for option prices can be traced all the way back to Merton's paper [Merton, 1973] on option pricing in 1973. The reference addresses this question by asking what can be said about the value of a call option without making any additional assumptions about the price path. The answer is that a European call option on a risky asset with zero interest rate, the price V_0 of this option must satisfy:

$$\max\left[S-K,0\right] \leq V_0 \leq S, \tag{1.1}$$

where S is the spot price of the underlying.

Figure 1.1 shows the boundary as the 45 degree line from the origin. Along this line, the call option is worth the same as the stock. The lower bound is the value of the call option at expiration by using a European call option.

Recall that the above bound is based on an arbitrage condition; if the price of the call option is outside this range then one can construct an arbitrage strategy that is guaranteed to make money without risk.

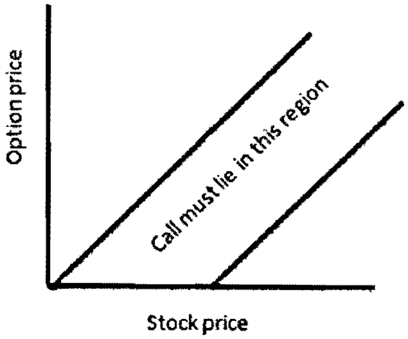


Figure 1.1: The boundary region for a call option's prices

On the mathematics and financial assumptions on the standard Merton bounds, there are two things one can consider.

First, the Merton bounds (1.1) requires no knowledge of the underlying asset's terminal price distribution or investor's behavior that will produce increasingly tighter bounds on option prices if additional assumptions are placed on investor preferences.

Second, a crucial simplification in Merton's work is the absence of transaction costs. In such idealized model the investor would optimally maintain a proportion of wealth in the stock by trading continuously. Such continuous strategies are no longer admissible once the transaction costs are introduced. The investor must then determine when the stock position is sufficiently out of line to make the trading worthwhile.

The present thesis builds on the approach from [Britten and Neuberger, 1996] (and expanded in [Rebonato, 2004]). An specific objective of the study is to add two types of transaction costs to the basic model from [Britten and Neuberger, 1996].

The approach in [Britten and Neuberger, 1996] manages to obtain tighter bounds in an incomplete market model by introducing realistic restrictions on the set of values taken by the underlying process. That reference develop this approach in a non probabilistic setting. The most important constraint of this model is that the (sampled) quadratic log variation of the prices of the underlying is known. They also assume that trading will take place at a finite number of occasions but it is not necessary to occur in equally spaced time intervals. An upper bound restriction on jumps size is also imposed.

As indicated, as a main goal of our thesis, we introduce transaction costs in the framework from [Britten and Neuberger, 1996]. As a result, we propose an approach to the pricing and hedging of contingent claims in the presence of transaction costs, under a general incomplete market in discrete time with special reference to the models introduced in [Bensaid and Scheinkman, 1992] and

[Avellaneda and Parás, 1995], based on the framework described in [Britten and Neuberger, 1996].

1.2 Overview of the thesis

This work is structured as follows. Chapter 2 introduces some of the notation to be used in the remaining of the thesis. We assume that there is a forward market in the asset S where the forward price for delivery of stock at time t_n will be denoted by S_n[2]. No probability structure is imposed at the outset. We also introduce the basic model from [Britten and Neuberger, 1996], the lower and upper bounds for the derivative's prices are also introduced. Section 2.3 uses the dynamic programming optimization methodology to make the problem computationally tractable.

Chapter 3 formulates two models for transaction costs in the basic framework of [Britten and Neuberger, 1996]. The investor's portfolio model in this chapter consists of one risky and one risk-free asset. Whenever the investor re-balances his/her portfolio in favor for one or the other asset class, he/she faces transaction costs. Transaction costs are either assumed to be of a fixed value or proportional to the volume of the risky asset traded. Section 3.1 provides a review of the transaction costs models. Fixed costs are described in Section 3.2, there it is shown how to compute the minimum upper bound. The dynamic programming formulation of the problem is also introduced. Section 3.3 develops the proportional transaction cost model, this means that transaction costs are proportional to the wealth transferred. This extends the basic framework from [Britten and Neuberger, 1996]. We also encode the optimization problem in terms of dynamic programming.

Chapter 4 describes the construction of the pricing algorithm, which is referred as the **J-N** algorithm and presents the mathematical setup of this algorithm.

Chapter 5 discusses numerical tests of the methods which were introduced in Chapter 2 and Chapter 3. We also study the approach in real market data, one interesting aspect of this investigation is to analyze the effects of the transaction costs on removing any available arbitrage opportunities.

Appendix A presents a very short introduction to financial concepts , Appendices B and C contain an introduction to the Binomial Model and the dynamic programming respectively. Appendix D contains analytical derivations of the optimal hedging ratios for the different models, these are not needed in the rest of the thesis. Finally all Matlab code is available in Appendix E.

[2] We will assume zero interest rates, this implies that we will end up dealing directly with the stock value S_n.

Chapter 2

Basic Model

A central question in finance consists in finding the price of an option given information on the underlying asset. The approach which is taken here is not a traditional approach as it does not rely on a probability model. We also allow for some market imperfections. More precisely, we are interested in determining the price of an option using a non probabilistic approach with a no-arbitrage condition. Since our model is incomplete because jumps are allowed, one cannot use an exact pricing rule. Independently of this state of affairs, one can always search for a bounding principle for the price of an option. Britten-Jones and Neuberger [Britten and Neuberger, 1996] (in some instances, during the following pages, these reference/authors are abbreviated by *J-N*) introduced an interesting way of looking at option pricing by assuming that the (sampled) log quadratic variation of price changes over the option's life is known. This chapter contains a review of this arbitrage-free approach to the pricing of derivatives.

2.1 Framework

The objective of this section is to model an incomplete financial market with a non probabilistic approach in a market containing one risk-free (bank account) and a risky assets (stock). In order to simplify our formulas and developments, we assume:

1. There are no taxes and transactions costs.

2. The riskless borrowing and lending rats are zero.

We discuss a more general framework in Chapter 3.

According to Merton's bound,[1] discussed in the previous chapter, the option is not more valuable than the underlying asset. This is a very weak bound but it is tight in the case that no extra restrictions are placed on the price path of the risky asset. In this work, we show through computer experimentation,

[1] Merton's bound is based on an arbitrage condition; if the price of the call option is outside this range then one can construct an arbitrage strategy that is guaranteed to make money without risk.

that practical bounds are obtained by assuming a known (log) quadratic variation and a bound on the jump size as constraints on allowed paths for the risky asset.

2.1.1 Non-probabilistic Asset Price Models

Assume a finite discrete model such that trading will take place at a finite number of occasions which is not fixed in number or necessarily equally spaced in time. There is a risky asset (e.g. stock) with $S(0)$ being today's price, and its value at $t_n \leq T$ is denoted by S_n. Moreover, at any trading position $t_n \leq T$, we have an amount of cash B_n in a bank account with zero interest rate.

Definition 1. *Assume S_i is the price of the underlying at trading position i, an asset price path π is any finite sequence of positive numbers $\{S_i\}_{i=0}^{N}$, where N is not set a priori but may equal to any finite positive integer and at maturity S_N is equal to S_T ($S_T \equiv S(T)$).*

In addition, we consider a European contingent claim, V, a security whose payoff at time T is only dependent on S_T. It means that the forward price path $\{S_i\}_{i=0}^{N}$ has a payoff at T which depends only on S_T, written as $V(S_T)$. For example, a European call option payoff is given by:

$$V(S_T) = max(S_T - K, 0),$$

where K is the strike price.

Now, we provide constraints on path's prices $\{S_i\}_{i=0}^{N}$ which will allow us to calculate no arbitrage price bounds which will be tighter that the Merton's bounds in the case of an incomplete market. In order to reach our goal, we make two assumptions regarding the price path. We assume that the (sampled log) quadratic variation of the risky asset is constant (i.e. it does not depend on the specific path) and we have a jump restriction between tradings.

Definition 2. *For a given constant $v > 0$, $d > 0$, and $S(0)$, a price sequence $\pi = \{S_0, S_1, ...S_N\}$, where $S_0 = S(0)$, belongs to the set of permissible price sequences $P(S_0, v, d)$ if and only if*

1. $S_0 = S(0)$,
2. $|\ln S_{i+1} - \ln S_i| \leq d, \qquad 0 \leq i \leq N-1,$
3. $\sum_{i=0}^{N-1} (\ln S_{i+1} - \ln S_i)^2 = v.$

The first condition simply requires that, for a price sequence to be permissible, its first value should be equal to the price of the underlying asset today. The second condition implies that the jump in the log of the prices between any two steps should not be larger than the given value d.

As far as the last condition is concerned, it is easy to recognize that $\sum_{i=0}^{N-1} (\ln S_{i+1} - \ln S_i)^2$ is the sampled (log) quadratic variation generated by the price sequence. This sampled quadratic variation should be known and be exactly equal to v.

Definition 3. *For a given price sequence $\pi \in P(S_0, v, d)$ at step i, define:*

1. $v_0 = v$,
2. $v_{i+1} = v_i - (\ln S_{i+1} - \ln S_i)^2, \qquad 0 \leq i \leq N-1.$

$\{v_i\}_{i=0}^{N}$ *is called a sequence of remaining (sampled) quadratic variations.*

The second condition shows that every price move uses up the fraction $(\ln S_{i+1} - \ln S_i)^2$ of the total quadratic variation, until finally at option expiry T, the given value v is all exactly depleted.

Note that by using items 1. and 2. from Definition 3, we have

$$v_i = v_0 - \sum_{k=0}^{i-1} (\ln S_{k+1} - \ln S_k)^2. \tag{2.1}$$

Hence, the v_i gives information about how much of the quadratic variation is available to consume in each trading occasion i.

In Figures 2.1 and 2.2, one can see a time series path of prices for two different trading strategies. In both cases, the trader will trade to re-balance her/his hedging portfolio; note that there is no presumption that trading takes place at constant time intervals and the number of trades can be as small or large as the trader requires.

Example A: In the case A, assume that maximum jump size is d, the quadratic variation is equal to v and in addition assume the trader is very lazy to trade more often or he/she expects larger gains. As a result he/she trades whenever:

$$|\ln S_{i+1} - \ln S_i| = d; \qquad 0 \leq i \leq N-1.$$

As displayed in Figure (2.1), the trader has traded $N = 24$ times and has used up all of the quadratic variation v.

Example B: In this case, Figure 2.2, for the same fixed maximum jump size d, the quadratic variation v, the trader re-balances his/her portfolio more often and at any time he/she wishes as far as the log of price changes is less or equal max jump d. Therefore at expiration T, he ends up with a permissible path $\{S_i\}_{i=0}^{45}$.

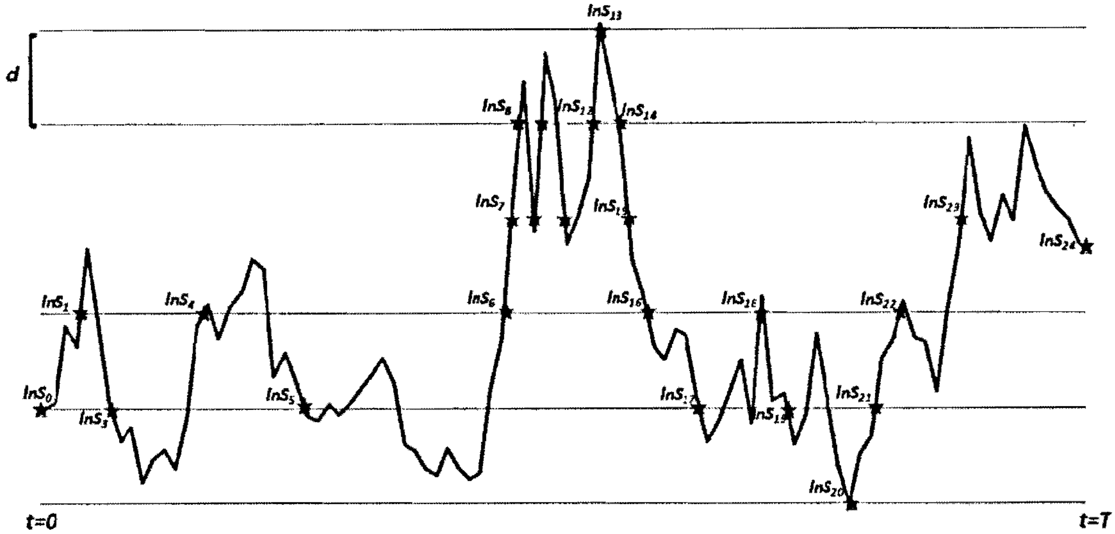


Figure 2.1: Case A: A lazy permissible path for the given maximum jump size d and the quadratic variation v.

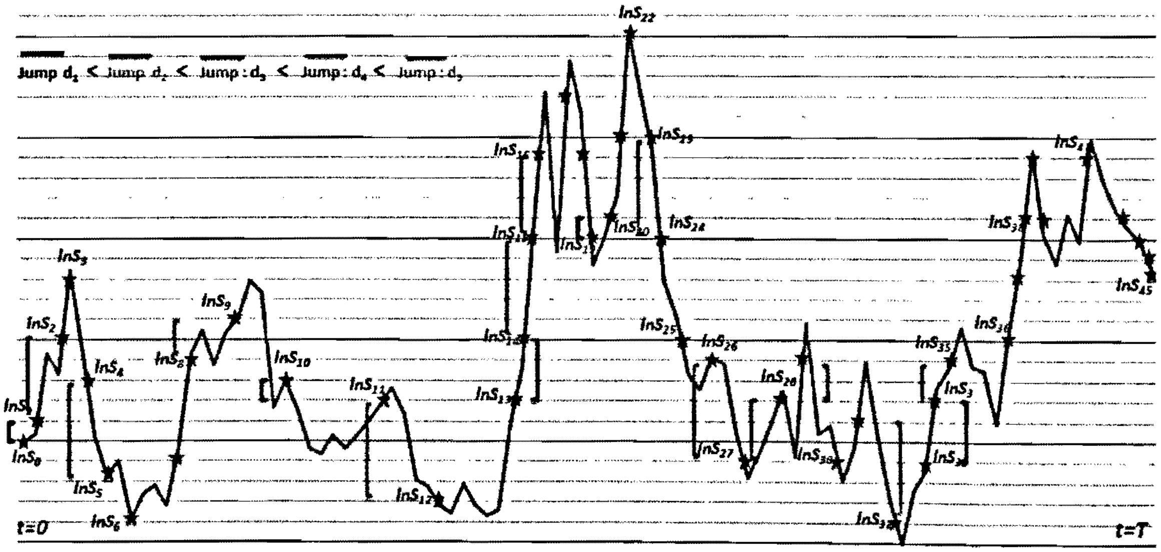


Figure 2.2: Case B: A possible permissible path which trader can trade any time as far as log return is less or equal maximum jump size d

2.1.2 Trading Strategy

The most important tool that a trader uses to make money is undoubtedly her/his trading strategies. A trading strategy, may mean different things to different people. At any given point in time, for any given security, a trading strategy should first tell a trader whether to buy, sell or hold; it then should provide the optimal trading size under the given market price. In finance, a trading strategy is a predefined set of rules for making trading decisions.

An investment strategy on the stock is given by a function $H(.,.)$ that depends on the spot S_i and the remaining volatility v_i and it represents the number of shares in the holding portfolio when $S(t_i) = S_i$ and the the remaining quadratic variation is equal to v_i.

We will restrict the investment strategies $H(.,.)$ to belong to a given set $\mathbb{H}$, whenever required this set $\mathbb{H}$ will satisfy the following definition.

Definition 4. *$\mathbb{H}$ is called an admissible class if it satisfies*

$$\inf_{H\in\mathbb{H}}\left\{\sup_{\pi\in P(S_0,v_0,d)}\left\{-\sum_{i=0}^{N-1}H(S_i,v_i)(S_{i+1}-S_i)\right\}\right\}=0. \tag{2.2}$$

The holdings in the bank account will be denoted by B_{S_i,v_i}, hence, a portfolio can be define as a vector valued function $\Gamma(S_i,v_i) = (H(S_i,v_i), B_{S_i,v_i})$. Sometimes for simplicity we set $h_i \equiv H(S_i,v_i)$ and $\mathcal{B}_i = B_{S_i,v_i}$.

2.1.3 Profits/Losses of Investor

We assume that, at time zero, the investor has V_0 dollars and invests this money in a portfolio containing $H(S_0,v_0)$ shares and deposits $\mathcal{B}_0 = V_0 - H(S_0,v_0)S_0$ in a bank account with zero interest rate. The initial value V_0 of this portfolio is given by:

$$V_0 = H(S_0,v_0)S_0 + \mathcal{B}_0. \tag{2.3}$$

In general, the value of the portfolio V_n at n-th trade, after adjustment, may be written as;

$$V_n = H(S_n,v_n)S_n + \mathcal{B}_n, \tag{2.4}$$

where $H(S_n,v_n)$ is the number of shares of the underlying asset and $\mathcal{B}_n$ is the amount of cash in the account. The investor holds this portfolio until $n+1$.

To proceed further, we need an equation governing the dynamics of the self-financing hedging portfolio. At stage $n+1$ and before adjusting the portfolio under the self-financing constraint, the portfolio's value equals (recall that we assume interest rates $r = 0$):

$$V_{n+1} = H(S_n, v_n)S_{n+1} + \mathcal{B}_n.$$

Then, using the self-financing trading strategy condition, the portfolio becomes

$$H(S_n, v_n)S_{n+1} + \mathcal{B}_n = H(S_{n+1}, v_{n+1})S_{n+1} + \mathcal{B}_{n+1}. \tag{2.5}$$

So at $(n+1)^{th}$ trading position after re-balancing, the value of the investor's portfolio is:

$$V_{n+1} = H(S_{n+1}, v_{n+1})S_{n+1} + \mathcal{B}_{n+1}. \tag{2.6}$$

By using (2.4) and (2.6) and then applying the self-financing equation (2.5) we can find the variation of the portfolio at two successive moments of trading in the form;

$$V_n = V_{n+1} - H(S_n, v_n)(S_{n+1} - S_n). \tag{2.7}$$

By applying (2.7) recursively; at maturity T, we have

$$V_{N-1} = V_N - H(S_{N-1}, v_{N-1})(S_N - S_{N-1}).$$

Hence; by the self financing property of the constructed portfolio:

$$H(S_{N-1}, v_{N-1})S_T + \mathcal{B}_T \equiv V_\Gamma(S_0, v_0) + \sum_{i=0}^{N-1} H(S_i, v_i)(S_{i+1} - S_i),$$

where the initial value $V_\Gamma(S_0, v_0) = H(S_0, v_0)S_0 + \mathcal{B}_0$ will also be denoted V_0. An important remark, to be used below, is that once $H(.,.)$ is fixed, the self financing constraint on the portfolio Γ determines $\mathcal{B}_k$ for $k = 1, \ldots, N-1$ and only, either V_0 *or* $\mathcal{B}_0$, can be set arbitrarily.

Definition 5. *Assume a derivative with payoff $V(S_T)$, for a given hedging strategy $H(.,.)$ and price path $\pi \in P(S_0, v_0, d)$, we denote the total balance of hedging (profit or loss) as follows:*

$$\Pi_H(\pi) = V_0 - V(S_T) + \sum_{i=0}^{N-1} H(S_i, v_i)(S_{i+1} - S_i). \tag{2.8}$$

2.2 Structure of an Optimal Strategy

We show next how the above introduced framework can be used to provide the upper and the lower bounds for the value of an option.

2.2.1 Why Minimax Approach?

Mathematical finance theory is devoted to the modeling of stock prices and devising investment strategies that maximize wealth gain, minimize risk while doing so, and so on. Typically, this is done by estimating the parameters in a probabilistic model of stock prices returns. One of the most popular probabilistic models is known as the Geometric Brownian Motion.

Even though, empirically the Geometric Brownian Motion [Osborne, 1959] has enjoyed great predictive success and every year trillions of dollars are traded assuming this model [2]; in reality, several decisions fall into the non-probabilistic category. While there maybe insufficient quantitative information to establish probabilistic criteria, the decision-maker can still apply qualitative judgment and experience to the situation. It would be a very unusual problem indeed that would leave a decision-maker bereft of ideas! For cases where probabilities cannot be specified, the main decision criteria are maximin, maximax, and minimax.

- The *maximax rule*, A risk-seeking decision-maker will choose the adventurous maximax decision rule. He takes an optimistic view that everything will turn out right regardless of what decision is taken. It looks only at the highest possible gain and ignores the opportunities that other alternatives might present. In monetary terms, a maximax decision looks for the best of the best by evaluating the largest possible profit that each alternative can produce and then choosing the alternative with the highest profit. An organization that adopts a maximax approach perceives a business opportunity and takes a gamble in trying to achieve its goal. It may become very profitable or it may go broke.

- The *maximin rule*, A risk-averse decision-maker will use the conservative maximin criterion which often leads to a decision to do nothing. An organization that adopts a maximin attitude is non-competitive and will soon be overtaken by more innovative risk-taking competitors. The maximin criterion essentially takes a pessimistic view and considers the results of taking the wrong alternative. It evaluates the worst outcome for each alternative and then chooses the alternative which leads to the best of these worst outcomes. In monetary terms, a maximin decision looks for the best of the worst by choosing the alternative that yields the maximum profit from all minimum possible returns hence the name maximin.

- The *minimax rule* can be considered as another conservative or pessimistic decision rule. In this case, a minimax decision looks for the minimum of the maximums by evaluating the largest possible opportunity loss that each alternative would produce and then choosing the alternative with the smallest loss. Thus, the minimax rule chooses the alternative with the smallest (or minimum) maximum regret hence the name minimax.

[2] Black-Scholes used this same model in their work on pricing options on stocks.

2.2.2 Optimal Strategy

Consider an investor who sold a derivative [3] with payoff $V(S_T)$ and option's price of V_0, and also who uses the trading strategy $H(.,.)$ to hedge the derivative.

Remark 1. *Note that we use the amount V_0 to set up the portfolio as an initial value in order to find a suitable upper bound for this amount.*

Let us to fix a hedging strategy $\tilde{H}(.,.)$ and according to (2.8), the investor's profit which depends on permissible path π is

$$\Pi_{\tilde{H}}(\pi) = V_0 - V(S_T) + \sum_{i=0}^{N-1} \tilde{H}(S_i, v_i)(S_{i+1} - S_i).$$

The second term (appearing with a minus sign) represents the payoff of the derivative at time of expiry T and the last term represents the gains/losses from trading.

Definition 6. *A strategy $H(.,.)$ is an arbitrage strategy if*

- $\forall \pi \in p(S_0, v_0, d), \quad \Pi_H(\pi) \geq 0$ *&* $V_0 = H(S_0, v_0)S_0 + B_0 = 0,$
- $\exists \pi^* \in p(S_0, v_0, d)$ *such that* $\Pi_H(\pi^*) > 0.$

Therefore a given trading strategy $\tilde{H}(.,.)$ is not an arbitrage if there exists $\pi \in P(S_0, v_0, d)$

$$\Pi_{\tilde{H}}(\pi) = V_0 - V(S_T) + \sum_{i=0}^{N-1} \tilde{H}(S_i, v_i)(S_{i+1} - S_i) < 0.$$

or

$$V_0 < V(S_T) - \sum_{i=0}^{N-1} \tilde{H}(S_i, v_i)(S_{i+1} - S_i).$$

Therefore,

$$V_0 < \sup_{\pi \in P(S_0, v_0, d)} \left\{ V(S_T) - \sum_{i=0}^{N-1} \tilde{H}(S_i, v_i)(S_{i+1} - S_i) \right\}.$$

The trading strategy $\tilde{H}(.,.)$ being arbitrary, we obtain:

$$V_0 \leq \inf_H \left\{ \sup_{\pi \in P(S_0, v_0, d)} \left\{ V(S_T) - \sum_{i=0}^{N-1} H(S_i, v_i)(S_{i+1} - S_i) \right\} \right\}. \tag{2.9}$$

[3] The derivative here is an European path-independent derivative.

Let us denote the minimum upper bound (2.9) as $\overline{V}(S_0, v_0)$

$$\overline{V}(S_0, v_0) \equiv \inf_H \left\{ \sup_{\pi \in P(S_0, v_0, d)} \left\{ V(S_T) - \sum_{i=0}^{N-1} H(S_i, v_i)(S_{i+1} - S_i) \right\} \right\}. \tag{2.10}$$

In Proposition 3, we show that $\overline{V}(S_0, v_0)$ can be interpreted as the maximum price a trader can charge by selling the derivative asset: nobody (who acknowledges only future paths with the quadratic variation v and maximum jumps d) will buy the option for more than $\overline{V}(S_0, v_0)$.

According to Proposition 7, the maximum lower bound can be calculated as the negative value of the minimum upper bound $\overline{V}(S_0, v_0)$ with the negative payoff (short position). So from a financial point of view, in the short position, the upper bound on this position can be interpreted as the lower bound on the long position.

$$\begin{aligned} \underline{V}(S_0, v_0) &= \sup_H \left\{ \inf_{\pi \in P(S_0, v_0, d)} \left\{ V(S_T) + \sum_{i=0}^{N-1} H(S_i, v_i)(S_{i+1} - S_i) \right\} \right\} \qquad (2.11) \\ &= -\inf_H \left\{ \sup_{\pi \in P(S_0, v_0, d)} \left\{ -V(S_T) - \sum_{i=0}^{N-1} H(S_i, v_i)(S_{i+1} - S_i) \right\} \right\} \end{aligned}$$

$\underline{V}(S_0, v_0)$ can be interpreted as the minimum price a trader will have to pay for the derivative. Nobody will sell it for less than $\underline{V}(S_0, v_0)$.

Before proceeding any further, let us introduce some basic properties of the minimum upper bound $\overline{V}(S_0, v_0)$.

Remark 2. *For given a contingent claim with payoff $V(S_T)$, the minimum upper bound*

$$\overline{V}(S_0, v_0) = \inf_H \left\{ \sup_{\pi \in P(S_0, v_0, d)} \left\{ V(S_T) - \sum_{i=0}^{N-1} H(S_i, v_i)(S_{i+1} - S_i) \right\} \right\}$$

from a mathematical point of view, means that for all $\varepsilon > 0$, there exist a trading strategy H_ε such that

$$0 \leq V(S_T) - \sum_{i=0}^{N-1} H_\epsilon(S_i, v_i)\ (S_{i+1} - S_i) - \overline{V}(S_0, v_0) \leq \epsilon. \tag{2.12}$$

The following propositions show that by means of $\underline{V}(S_0, v_0)$ and $\overline{V}(S_0, v_0)$ one has an upper-hedging and under-hedging portfolio for a derivative with payoff $V(S_T)$.

Proposition 1. *Fix a payoff $V(S_T)$, then for all $\epsilon > 0$ there exists a portfolio $\Gamma(.,.) = (H(.,.), B_{.,.})$, depending on ϵ, with $V_\Gamma(S_0, v_0) \geq \overline{V}(S_0, v_0)$ such that the following bound holds for all permissible paths*

$\{S_0, ..., S_N\}$:

$$V(S_T) \leq V_\Gamma(S_0, v_0) + \sum_{i=0}^{N-1} H(S_i, v_i)\ (S_{i+1} - S_i) + \epsilon. \tag{2.13}$$

Proof. For an arbitrary $\epsilon > 0$ select $H_\epsilon(.,.)$ satisfying:

$$0 \leq V(S_T) - \sum_{i=0}^{N-1} H_\epsilon(S_i, v_i)\ (S_{i+1} - S_i) - \overline{V}(S_0, v_0) \leq \epsilon, \tag{2.14}$$

for all permissible paths $\{S_0, ..., S_N\}$. Then, by defining $\mathcal{B}_0 \equiv \overline{V}(S_0, v_0) - H_\epsilon(S_0, v_0)S_0$, the portfolio defined by $\Gamma(.,.) = (H_\epsilon(.,), \mathcal{B}_{.,.})$ satisfies $V_\Gamma(S_0, v_0) = \overline{V}(S_0, v_0)$. A re-writing of (2.14) gives then (2.13)

$$V(S_T) \leq V_\Gamma(S_0, v_0) + \sum_{i=0}^{N-1} H(S_i, v_i)\ (S_{i+1} - S_i) + \epsilon.$$

□

Proposition 2. *Fix a payoff $V(S_T)$, assume $\mathbb{H}$ is closed under multiplication by -1, then for all $\epsilon > 0$, there exists a portfolio $\Gamma(.,.) = (H(.,.), \mathcal{B}_{.,.})$, depending on ϵ, with $V_\Gamma(S_0, v_0) \leq \underline{V}(S_0, v_0)$ such that the following bound holds for all permissible paths $\{S_0, ..., S_N\}$:*

$$V(S_T) \geq V_\Gamma(S_0, v_0) + \sum_{i=0}^{N-1} H(S_i, v_i)\ (S_{i+1} - S_i) - \epsilon. \tag{2.15}$$

Proof. For an arbitrary $\epsilon > 0$ select $H'_\epsilon(.,.)$ satisfying:

$$0 \leq \underline{V}(S_0, v_0) - \left(V(S_T) + \sum_{i=0}^{N-1} H'_\epsilon(S_i, v_i)\ (S_{i+1} - S_i) \right) \leq \epsilon, \tag{2.16}$$

for all permissible paths $\{S_0, ..., S_N\}$. Then, by defining $\mathcal{B}_0 \equiv \underline{V}(S_0, v_0) - H'_\epsilon(S_0, v_0)S_0$, the portfolio defined by $\Gamma(.,.) = (H'_\epsilon(.,.), \mathcal{B}_{.,.})$ satisfies $V_\Gamma(S_0, v_0) = \underline{V}(S_0, v_0)$. So by taking $H_\epsilon(.,.) \equiv -H'_\epsilon(.,.)$ we obtain:

$$V(S_T) \geq V_\Gamma(S_0, v_0) + \sum_{i=0}^{N-1} H_\epsilon(S_i, v_i)\ (S_{i+1} - S_i) - \epsilon.$$

□

The following shows that superhedge for the minimum upper bound and underhedge for the maximum lower bound are tight.

Proposition 3. *Consider contingent claim with a payoff $V(S_T)$ and an arbitrary $H(.,.) \in \mathbb{H}$.*

- *If $V_0 = H(S_0, v_0)S_0 + \mathcal{B}_0 < \overline{V}(S_0, v_0)$ then,*

$$V_0 + \sum_{i=0}^{N-1} H(S_i, v_i)(\hat{S}_{i+1} - \hat{S}_i) < V(\hat{S}_T) \text{ for some permissible price sequence } \{\hat{S}_i\}. \tag{2.17}$$

- *If* $V_0 = H(S_0, v_0)S_0 + \mathcal{B}_0 > \underline{V}(S_0, v_0)$ *then,*

$$V_0 + \sum_{i=0}^{N-1} H(S_i, v_i)(\hat{S}_{i+1} - \hat{S}_i) > V(\hat{S}_T) \textit{ for some permissible price sequence } \{\hat{S}_i\}. \tag{2.18}$$

Proof. We prove only (2.17); assume there exists $H(.,.) \in \mathbb{H}$ satisfying $V_0 = H(S_0, v_0)S_0 + \mathcal{B}_0 < \overline{V}(S_0, v_0)$, also assume that (2.17) does not hold. Therefore,

$$V_0 \geq \sup_{\pi \in P(S_0, v_0, d)} [V(S_T) - \sum_{i=0}^{N-1} H(S_i, v_i)(S_{i+1} - S_i)].$$

This implies $V_0 \geq \overline{V}(S_0, v_0)$. For the lower bound, the process is the same just one can use inf rather than sup. □

Proposition 4. *Consider a contingent claim with payoff* $V_3(S_T)$ *that is a convex combination of the payoffs on two other claims, i.e.* $V_3(S_T) = \lambda V_1(S_T) + (1 - \lambda)V_2(S_T)$, *where* $0 \leq \lambda \leq 1$. *Also assume* $\mathbb{H}$ *is closed under convex combinations. Then the minimum upper bound* $\overline{V}_3(S_0, v_0)$ *satisfies*

$$\overline{V}_3(S_0, v_0) \leq \lambda \overline{V}_1(S_0, v_0) + (1 - \lambda)\overline{V}_2(S_0, v_0)$$

.

Proof. For any $\lambda \in (0, 1)$, the payoff $V_3(S_T)$ is defined by:

$$V_3(S_T) = \lambda V_1(S_T) + (1 - \lambda)V_2(S_T). \tag{2.19}$$

Assume $\overline{V}_1(S_0, v_0)$ be the minimum upper bound which has the payoff $V_1(S_T)$. Then for any arbitrary $\varepsilon > 0$ and all $\pi \in P(S_0, v_0, d)$, there is a $H(.,.) \in \mathbb{H}$ such that

$$\lambda \overline{V}_1(S_0, v_0) \geq \sup_{\pi} \left\{ \lambda V_1(S_T) - \lambda \sum_{i=0}^{N-1} H(S_i, v_i)(S_{i+1} - S_i) \right\} - \frac{\varepsilon}{2}. \tag{2.20}$$

Similarly there exist $L(.,.) \in \mathbb{H}$

$$(1 - \lambda)\overline{V}_2(S_0, v_0) \geq \sup_{\pi} \left\{ (1 - \lambda)V_2(S_T) - (1 - \lambda) \sum_{i=0}^{N-1} L(S_i, v_i)(S_{i+1} - S_i) \right\} - \frac{\varepsilon}{2}. \tag{2.21}$$

By adding (2.20) and (2.21), we obtain

$$\begin{aligned} \lambda \overline{V}_1(S_0, v_0) + (1 - \lambda)\overline{V}_2(S_0, v_0) \geq \sup_{\pi} & \left\{ \lambda V_1(S_T) - \lambda \sum_{i=0}^{N-1} H(S_i, v_i)(S_{i+1} - S_i) \right\} - \frac{\varepsilon}{2} \\ & + \sup_{\pi} \left\{ (1 - \lambda)V_2(S_T) - (1 - \lambda) \sum_{i=0}^{N-1} L(S_i, v_i)(S_{i+1} - S_i) \right\} - \frac{\varepsilon}{2} \end{aligned}$$

So, by using (2.19)

$$\lambda\overline{V}_1(S_0,v_0)+(1-\lambda)\overline{V}_2(S_0,v_0)\geq\sup_{\pi}\left\{V_3(S_T)-\sum_{i=0}^{N-1}(\lambda H(S_i,v_i)-(1-\lambda)L(S_i,v_i))(S_{i+1}-S_i)\right\}-\varepsilon,$$

Set $Y(S_i,v_i)\equiv\lambda H(S_i,v_i)-(1-\lambda)L(S_i,v_i)$ [4], so the trading strategy $Y(.,.)\in\mathbb{H}$. Hence;

$$\lambda\overline{V}_1(S_0,v_0)+(1-\lambda)\overline{V}_2(S_0,v_0)\geq\sup_{\pi\in P(S_0,v_0,d)}\left\{V_3(S_T)-\sum_{i=0}^{N-1}Y(S_i,v_i)(S_{i+1}-S_i)\right\}-\varepsilon$$

Since $\epsilon>0$ is arbitrary, we can choose it small enough such that

$$\lambda\overline{V}_1(S_0,v_0)+(1-\lambda)\overline{V}_2(S_0,v_0)\geq\overline{V}_3(S_0,v_0).$$

□

Proposition 5. *Given two contingent claims with payoff $V_1(S_T)$ and $V_2(S_T)$ such that*

$$V_2(S_T)=aV_1(S_T)+b.$$

where a,b are arbitrary real numbers, and also assume $\mathbb{H}$ is close under multiplication, then

$$\overline{V}_2(S_0,v_0)=|a|\overline{V}_1(S_0,v_0)+b,$$

Proof. assume $a>0$ (when $a=0$ the result holds trivially) by definition:

$$\overline{V}_2(S_0,v_0)=\inf_H\left\{\sup_{\pi\in P(S_0,v_0,d)}\left\{a\ V_1(S_T)-\sum_{i=0}^{N-1}H(S_i,v_i)(S_{i+1}-S_i)\right\}\right\}+b.$$

Now, it is enough to pull out a as common factor so,

$$\overline{V}_2(S_0,v_0)=a\inf_H\left\{\sup_{\pi\in P(S_0,v_0,d)}\left\{V_1(S_T)-\sum_{i=0}^{N-1}\frac{1}{a}H(S_i,v_i)(S_{i+1}-S_i)\right\}\right\}+b.$$

So let us call $\frac{1}{a}H(S_i,v_i)$ as a new strategy $H'(S_i,v_i)$. Hence

$$\overline{V}_2(S_0,v_0)=a\ \overline{V}_1(S_0,v_0)+b. \tag{2.22}$$

When $a<0$, then $-a>0$. To use the same argument, equation (2.22) can be written as

$$\overline{V}_2(S_0,v_0)=-a\ \overline{V}_1(S_0,v_0)+b.$$

[4] Recall $H(.,.),L(.,.)\in\mathbb{H}$, and $\mathbb{H}$ is assumed to be closed under convex combinations

Therefore

$$\overline{V}_2(S_0, v_0) = |a|\overline{V}_1(S_0, v_0) + b.$$

□

Proposition 6. *Assume a derivative with payoff $V(S_T)$, and also $\mathbb{H}$ is an admissible class, then the minimum upper bound*

$$\overline{V}(S_0, v_0) = \inf_H \left\{ \sup_{\pi \in P(S_0, v_0, d)} \left\{ V(S_T) - \sum_{i=0}^{N-1} H(S_i, v_i)(S_{i+1} - S_i) \right\} \right\}. \tag{2.23}$$

satisfies the following properties:

1. *If $V(S_T) = 0$ then $\overline{V}(S_0, v_0) = 0$.*

2. *If $V(S_T) = k$ then $\overline{V}(S_0, v_0) = k$.*

3. *If $V(S_T) = kS_T$ and $\mathbb{H}$ is closed under multiplication by (-1) and shifts by a constant k then $\overline{V}(S_0, v_0) = kS(0)$.*

Proof. 1. Suppose that $V(S_T) = 0$ because $\mathbb{H}$ is an admissible class, then according to equation (2.2),

$$\overline{V}(S_0, v_0) = \inf_H \left\{ \sup_{\pi \in P(S_0, v_0, d)} \left\{ \underbrace{V(S_T)}_{0} - \sum_{i=0}^{N-1} H(S_i, v_i)(S_{i+1} - S_i) \right\} \right\} = 0$$

Therefore, $\overline{V}(S_0, v_0) = 0$.

2. Assume that $\overline{V}(S_T) = k$, and let $\tilde{V}(S_T) \equiv V(S_T) - k = 0$, so from Property 1 we obtain $\overline{\tilde{V}}(S_0, v_0) = 0$, on the other hand

$$\begin{aligned}
\overline{\tilde{V}}(S_0, v_0) &= \inf_H \left\{ \sup_{\pi \in P(S_0, v_0, d)} \left\{ \tilde{V}(S_T) - \sum_{i=0}^{N-1} H(S_i, v_i)(S_{i+1} - S_i) \right\} \right\} \\
&= \inf_H \left\{ \sup_{\pi \in P(S_0, v_0, d)} \left\{ V(S_T) - \sum_{i=0}^{N-1} H(S_i, v_i)(S_{i+1} - S_i) \right\} \right\} - k \\
&= \overline{V}(S_0, v_0) - k = 0.
\end{aligned}$$

Hence, $\overline{V}(S_0, v_0) = k$.

3. Consider the payoff $V(S_T) = kS_T$ where k is a fixed real number, recall that $S_T = S_N$ and $\pi \in$

$P(S_0, v_0, d)$ then:

$$\begin{aligned}\overline{V}(S_0, v_0) &= \inf_H \left\{ \sup_\pi \left\{ kS_T - \sum_{i=0}^{N-1} H(S_i, v_i)(S_{i+1} - S_i) \right\} \right\} \\ &= \inf_H \left\{ \sup_\pi \left\{ k\ S_N - H(S_{N-1}, v_{N-1})S_N + H(S_{N-1}, v_{N-1})S_{N-1} - \sum_{i=0}^{N-2} H(S_i, v_i)(S_{i+1} - S_i) \right\} \right\}\end{aligned}$$

$$\begin{aligned}&= \inf_H \left\{ \sup_\pi \left\{ (k - H(S_{N-1}, v_{N-1})S_N + (H(S_{N-1}, v_{N-1}) + k - k)S_{N-1} - \sum_{i=0}^{N-2} H(S_i, v_i)(S_{i+1} - S_i) \right\} \right\} \\ &= \inf_H \left\{ \sup_\pi \left\{ -H(S_{N-1}, v_{N-1}) - k)(S_N - S_{N-1}) + kS_{N-1} - \sum_{i=0}^{N-2} H(S_i, v_i)(S_{i+1} - S_i) \right\} \right\} = ... \\ &= \inf_H \left\{ \sup_\pi \left\{ kS_0 - \sum_{i=0}^{N-2} (k - H(S_{N-1}, v_{N-1}))(S_{i+1} - S_i) \right\} \right\},\end{aligned}$$

since kS_0 is constant and $\mathbb{H}$ is an admissible class, so by equation (2.2) we obtain:

$$\overline{V}(S_0, v_0) = \inf_H \left\{ \overbrace{\sup_\pi \left\{ - \sum_{i=0}^{N-2} (k - H(S_i, v_i))(S_{i+1} - S_i) \right\}}^{0} \right\} + kS_0 = kS_0.$$

□

The following proposition gives the relationship between the bound on the long position and the bound on a short position on the same claim.

Proposition 7. *Given two contingent claims with payoff $V_1(S_T)$ and $V_2(S_T)$ such that $V_2(S_T) = -V_1(S_T)$, assume $\mathbb{H}$ is closed under convex combination, then*

1. $\overline{V}_1(S_0, v_0) + \overline{V}_2(S_0, v_0) \geq 2\ \overline{V}_3(0)$, *where, allowing for some abuse of notation, $\overline{V}_3(0)$ denotes the minimum upper bound for a continent claim with payoff $V_3(S_T) = 0$.*

2. *If $\mathbb{H}$ is an admissible class, then $\overline{V}_1(S_0, v_0) + \overline{V}_2(S_0, v_0) \geq 0$.*

Proof. 1. By assumption, we have two contingent claims with payoff $V_1(S_T)$, and $V_2(S_T)$ such that

$$V_1(S_T) + V_2(S_T) = 0,$$

or equally

$$\frac{1}{2}V_1(S_T) + \frac{1}{2}V_2(S_T) = V_3(S_T) = 0.$$

Then from Proposition 4

$$\overline{V}_1(S_0, v_0) + \overline{V}_2(S_0, v_0) \geq 2\ \overline{V}_3(S_0, v_0) = 2\ \overline{V}_3(0).$$

2.If $\mathbb{H}$ is an admissible class, then by Proposition 6, since $\overline{V}_3(S_T) = 0$ when $\overline{V}_3(S_0, v_0) = 0$. Hence

$$\overline{V}_1(S_0, v_0) + \overline{V}_2(S_0, v_0) \geq 0.$$

□

As a result of Proposition 7, since

$$V(S_T) + (-V(S_T)) = \frac{1}{2}V(S_T) - \frac{1}{2}V(S_T) = 0.$$

If $\mathbb{H}$ is an admissible class, it follows from Proposition 7, item 2, that:

$$\overline{V}(S_0, v_0) \geq \underline{V}(S_0, v_0). \tag{2.24}$$

These arbitrage bounds mean that investors would not pay more than $\underline{V}(S_0, v_0)$ and would not sell it for less than $\overline{V}(S_0, v_0)$.

2.3 Dynamic Programming Formulation

The solution to the problem (2.10) is a strategy $H(.,.)$ such that minimizes total payout in the worst-case scenario. Therefore, potentially, this is a hard problem because involves a search over all permissible paths and over all hedging strategies. The beauty of dynamic programming (see Appendix C for more details about dynamic programming) is to convert the sequential problem into a collection of two-period problems which is easy to solve.

2.3.1 The General Problem

Let us clarify up the assumptions which are necessary to convert the sequential problem (2.10) into a dynamic programming problem, first we need to detect state variables and control variables (Appendix C).

Assume S_i and v_i, where $0 \leq i \leq N$, be two of state variables of the system which summarize the past information that is relevant for future optimization and c_i is a control variables which can be chosen in every period of trading by the decision-maker. We define c_i as control variable which is equal the continuously compounded growth (rate of return) between the adjusted prices:

$$\ln \frac{S_{i+1}}{S_i} = c_i.$$

So

$$S_{i+1} = S_i e^{c_i}.$$

The other state variable is the volatility v_i which evolves as;

$$\begin{aligned} v_{i+1} &= v_i - (\ln S_{i+1} - \ln S_i)^2 \\ &= v_i - {c_i}^2. \end{aligned}$$

Now, we have following optimization problem:

$$\overline{V}(S_0, v_0) = \inf_{\{h_i\}_{i=0}^{N-1}} \left\{ \sup_{\{S_i\}_{i=0}^{N-1}} \left\{ V(S_T) - \sum_{i=0}^{N-1} h_i(S_{i+1} - S_i) \right\} \right\}, \tag{2.25}$$

where $h_i = H(S_i, v_i)$.

Subject to

- $S_{i+1} = S_i e^{c_i}$ & $v_{i+1} = v_i - {c_i}^2 \geq 0$.
- $c_i \in \mathbb{R}$ & $c_i^2 \leq d^2$ where $i = 0, ..., N-1$.
- $S_0 = S(0)$ and $v_0 = v$ are given.
- $S_N = S_T \geq 0$ and $v_N = 0$.

$V(S_T)$ is called a "scrap" value function at the end of the program where no further decisions are made. $\sum_{i=0}^{N-1} h_i(S_{i+1} - S_i)$ is separable after the start trading and $S_{i+1} = S_i e^{c_i}$ is separable by following structure:

$$\begin{aligned} S_1 &= S_0 e^{c_0}, \\ S_2 &= S_1 e^{c_1}, \\ &\vdots \\ S_N &= S_{N-1} e^{c}_{N-1}. \end{aligned}$$

These equations are called *transition equations.*

This problem can be solved by using the standard constrained optimization technique of *Lagrange multipliers.* This is a perfectly good approach, but dynamic programming is an alternative that is sometimes more convenient. As we consider in Appendix C, as more convenient method to use is Bellman's Principle of Optimality which guarantees that if we convert the original sequential problem into a collection of small problems, the optimal choice in each of the two-period problem must be globally optimal to justify the transformation.

Bellman's Method:

Consider the problem at time zero:

Problem I:

$$\overline{V}(S_0, v_0) = \inf_{\{h_i\}_{i=0}^{N-1}} \left\{ \sup_{\{c_i\}_{i=0}^{N-1}} \left\{ V(S_T) - \sum_{i=0}^{N-1} h_i(S_{i+1} - S_i) \right\} \right\}.$$

Subject to

- $S_{i+1} = S_i e^{c_i}$, & $v_{i+1} = v_i - {c_i}^2 \geq 0$.
- $c_i \in \mathbb{R}$ & $c_i^2 \leq d^2$ where $i = 0, ..., N-1$.
- $S_0 = S(0)$ and $v_0 = v$ are given.

Now consider the same problem, starting at some $n_0 > 0$:

Problem II

$$\overline{V}(S_{n_0}, v_{n_0}) = \inf_{\{h_i\}_{i=n_0}^{N-1}} \left\{ \sup_{\{c_i\}_{i=n_0}^{N-1}} \left\{ V(S_T) - \sum_{i=0}^{N-1} h_i(S_{i+1} - S_i) \right\} \right\}.$$

Subject to

- $S_{i+1} = S_i e^{c_i}$.
- $c_i \in \mathbb{R}$ & $c_i^2 \leq d^2$ where $i = n_0, ..., N-1$.
- S_{n_0} is given.

According to Bellman's *Principle of Optimality asserts* [Bellman, 1957], any solution to Problem I (i.e. on the range $i = 0, ..., T$) which yields to $S_0 = S(0)$ and an optimal strategy also must solve Problem II (i.e.: on the range $i = n_0, ..., T$).

Remark 3. *This result depends on additive time separability, since otherwise we could not "break" the solution at n_0. Additive separability is sufficient for Bellman's principle of optimality.*

Bellman's principle of optimality allows us to use the trick of solving large Problem I by solving the smaller Problem II, sequentially. Also, since n_0 is arbitrary, we can choose to solve the problem $n_0 = N-1$ first, which is a simple 2-period problem, and then work backwards as follows.

Step 1:
Set $n_0 = N-1$, so that Problem II is simply:

$$\overline{V}(S_{N-1}, v_{N-1}) = \inf_{\{h_{N-1}\}} \left\{ \sup_{\{c_{N-1}\}} \{-h_{N-1}(S_N - S_{N-1}) + V(S_T)\} \right\}.$$

Subject to

- $S_N = S_{N-1}e^{c_{N-1}}$, $v_N = 0$.
- S_{N-1} is given.

Step 2: Set $n_0 = N - 2$, so that Problem II is:

$$\overline{V}(S_{N-2}, v_{N-2}) = \inf_{\{h_i\}_{i=N-2}^{N-1}} \left\{ \sup_{\{c_i\}_{i=N-2}^{N-1}} \{-h_{N-2}(S_{N-1} - S_{N-2}) - h_{N-1}(S_N - S_{N-1}) + V(S_T)\} \right\} \tag{2.26}$$

Subject to

- $S_N = S_{N-1}e^{c_{N-1}}$, $v_N = v_{N-1} - c_{N-1}^2 \geq 0$
- $S_{N-1} = S_{N-2}e^{c_{N-2}}$, $v_{N-1} = v_{N-2} - c_{N-2}^2 \geq 0$
- S_{N-2} , S_{N-1} are given.

According to [Bertsekas, 1976], we can rewrite this as :

$$\inf_{\{h_{N-2}, h_{N-1}\}} \left\{ \sup_{\{c_{N-1}\}} - h_{N-2}(S_{N-1} - S_{N-2}) + \sup_{\{c_{N-1}\}} \{-h_{N-1}(S_N - S_{N-1}) + V(S_T)\} \right\} =$$

$$\inf_{\{h_{N-2}\}} \left\{ \sup_{\{c_{N-2}\}} - h_{N-2}(S_{N-1} - S_{N-2}) + \inf_{\{h_{N-1}\}} \sup_{\{c_{N-1}\}} \{-h_{N-1}(S_N - S_{N-1}) + V(S_T)\} \right\}$$

(2.26) has already given us the solution to the inside maximization problem, so that we can re-write Step 2 as:

$$\inf_{\{h_{N-2}\}} \left\{ \sup_{\{c_{N-2}\}} \{-h_{N-2}(S_{N-1} - S_{N-2}) + \overline{V}(S_{N-1}, v_{N-1})\} \right\}$$

Subject to

- $S_{N-1} = S_{N-2}e^{c_{N-2}}$, $v_{N-1} = v_{N-2} - c_{N-2}^2 \geq 0$
- S_{N-2} is given and $c_{N-2}^2 \leq d^2$.

Step 3:
Using an argument analogous Step 2, in general, the problem in period i can be written as:

$$V(S_i, v_i) = \inf_{\{h_i\}} \left\{ \sup_{\{c_i\}} \{-h_i(S_i - S_{i+1}) + \overline{V}(S_{i+1}, v_{i+1})\} \right\}.$$

Subject to:

- $S_{i+1} = S_i e^{c_i}$, $v_{i+1} = v_i - c_i^2 \geq 0$.

- S_i is given.

Step 4:
After going through the successive rounds of single period optimization problems, eventually one reaches the problem in time zero:

$$V_0(S,v) = \inf_{\{h_0\}} \left\{ \sup_{\{c_0\}} \{\overline{V}(S_1, v_1) - h_0(S_0 - S_1)\} \right\}.$$

- $S_1 = S_0 e^{c_0}$ and $c_0^2 \leq d^2$.
- $S_0 = S, \ v_0 = v$ is given.

2.3.2 Bellman Equation

Therefore, the Bellman Equation can be written as

$$\overline{V}(S_i, v_i) = \inf_{h_i} \left\{ \sup_{c_i} \{\overline{V}(S_{i+1}, v_{i+1}) - h_i(S_{i+1} - S_i)\} \right\} \tag{2.27}$$

Subject to the conditions

- $(\ln S_{i+1} - \ln S_i)^2 = c_i^2 \leq d^2$.
- $v_{i+1} = v_i - c_i^2 \geq 0$.

The initial boundary conditions are given by the initial values of the state variables, S_0, v_0. Obviously, at the boundary $v = 0$, we have $V(S_T, 0) = V(S_T)$. Note that the terminal price S_T is free, as is the number of price moves N.

In chapter 4 we will developed a numerical solution for above optimization based on the Britten-Neuberger model.

Chapter 3

Transaction Costs Models

In the analysis of the Britten and Neuberger's model, described in the previous chapter, we have assumed that trading, by buying and selling the underlying, is costless in the sense that we can buy or sell any amount at the given market price without paying an intermediary or affecting the price.

In order to have a more realist model, we extend the previous setup to include models with transaction costs. This will allow us to study how several of the model's features are affected by the introduction of transaction costs. One way of introducing transaction costs in the basic model is to interpret the sequence of prices in a permissible price path as bid or ask prices; this point of view is suggested in [Britten and Neuberger, 1996] and [Rebonato, 2004]. In this work, we propose more direct ways to approach the pricing and hedging of contingent claims under transaction costs with the same general assumptions on price paths and the quadratic variation as in the previous chapter.

After a short review of how transaction costs appear in financial practice, we proceed to describe two reasonable models of them which fit with our basic model. More precisely, we present an original extension of the basic framework for pricing derivatives in [Britten and Neuberger, 1996] by introducung two models of transaction costs and we show how we can find the option's price bounds in these cases.

3.1 What are Transaction Costs?

Over the past decades, trading has increasingly been regarded as an area for cutting costs. Several alternatives have evolved for placing orders, and models for transaction costs have become increasingly important as a means for evaluating them. Also, the option pricing in the presence of transaction costs has recently become a very popular subject for research; in particular, it represents one way of removing arbitrages in real markets as well as in some models.

In financial economics, transaction costs are generally understood as all costs associated with trading and are categorized according to three groups [Harris, 2003]; *explicit costs*, *implicit costs*, and *missed trade opportunity costs*.

- **Explicit transaction costs :** are costs that are observable and known upfront and directly related to trading such as commissions paid to brokers, fees paid to exchanges, and other transaction based taxes, and also, as in the case of larger trading organizations, the internal costs such as salaries of traders, software, and accounting.
- **Implicit transaction costs:** are also referred to as unobservable transaction costs and include such elements as price impact [1] and the bid/ask spread[2].
- **Missed trade opportunity costs:** are costs that arise when a trader acts in a suboptimal manner and as a consequence forgoes an opportunity to execute a trade, for example by waiting for the market to move in a favorable direction before sending an order to the market, only to see the market to move in the opposite direction.

3.1.1 More Realistic Investment Trading Model

To render our model more realistic let us assume that moving our wealth from one position to another incurs into some cost. The problem can be formulated in terms of an agent that buys and sells options on the stock of a company. At some point in time, he/she decides to hedge the book or options portfolio against future price fluctuations. The agent would like to determine the least costly strategy taking into account the projected transaction costs due to dynamic hedging. The initial cost of such strategy can be interpreted as the minimal capital reserve needed to protect the portfolio against future market moves.

For the sake of simplicity, we will assume that the investor pays the same transaction costs either if he/she buys or sells a given volume of the risky asset.

As we have assumed in the previous chapter, we consider a market in which there are two assets available for investment: one risky asset $\mathcal{S}$ which pays no dividends and one risk-free asset (bank account) $\mathcal{B}$ with zero interest rate. We consider hedging a simple portfolio composed of a path independent European option with maturity T. In addition, we assume the same model as in Chapter 2 which, in particular, allows for a finite, but arbitrarily large, number of trades to occur which are not necessarily equally spaced in time. We then use a no-arbitrage argument to establish an upper and lower bound by means of approximate dynamic hedging strategies in the presence of transaction costs.

Two types of transaction costs are considered:

- Fixed cost per-trading.

[1] Price impact is the effect the execution of a sizable order has on the market. While a trader may be able to transact a small order without materially affecting the market price, the larger the order the greater the impact is on the market. This effect is variable in terms of the liquidity conditions of the specific market and depends on factors such as the time-of-day.

[2] The bid/ask spread is the difference between the bid price, the price where market participants are willing to buy a financial instrument, and the ask price, the price where market participants are willing to sell a financial instrument.

- Proportional to the volume of the trading.

Both types of transactions costs are very common in markets and, hence, the results we report will be directly applicable to common investors.

3.2 Fixed Transaction Costs

To motivate the subsequent definition of the cost of a strategy under transaction costs, it is helpful to first explain how it is implemented in our model.

3.2.1 The Portfolio of the Hedger

In this section, we are going to find the optimal investment strategy when only fixed transaction costs are present. The main method is similar as the one we introduced in Chapter 2.

Remark 4. *The positive fixed transaction cost paid if trading has occurred in the risky security. We assume that re-balancing of the bank account is costless (besides assuming* $r = 0$*). If agents do not trade, then we assume that they do not incur in any transaction cost.*

First, in order to construct the portfolio, assume that at time $t = 0$, the investor starts his business by having $H(S_0, v_0)$ units of stock and $\mathcal{B}_0$ in the bank account, so the value portfolio including the effect of the transaction cost $\mathcal{K}$ can be written as

$$V_0 = \mathcal{B}_0 + H(S_0, v_0)S_0 + \mathcal{K}, \tag{3.1}$$

where $\mathcal{K}$ is the transaction cost and is a fixed positive number and V_0 is a given constant, representing the initial wealth.

At time n, the agent's portfolio value is:

$$V_n = \mathcal{B}_{n-1} + H(S_{n-1}, v_{n-1})S_n, \qquad 0 < n \leq N.$$

where $\mathcal{B}_{n-1}$ is the dollar amounts in the bank account and $H(S_{n-1}, v_{n-1})$ is the number of shares held in the portfolio at time n before trading.

Then the investor needs to balance his portfolio by using a self-financing strategy, so

$$\mathcal{B}_{n-1} + H(S_{n-1}, v_{n-1})S_n = \mathcal{B}_n + H(S_n, v_n)S_n + \mathcal{K}, \qquad 0 < n \leq N. \tag{3.2}$$

By using (3.1) and applying (3.2), recursively, we obtain the total cost of a strategy.

$$\begin{aligned}V_0 &= \mathcal{B}_0 + H(S_0,v_0)S_0 + \mathcal{K}\\ &= \mathcal{B}_1 + H(S_1,v_1)S_1 - H(S_0,v_0)(S_1 - S_0) + 2\mathcal{K}\\ &= \mathcal{B}_2 + H(S_2,v_2)S_2 - \sum_{i=0}^{2} H(S_i,v_i)(S_{i+1} - S_i) + \sum_{i=0}^{1} \mathcal{K} = ...\\ &= \mathcal{B}_{N-1} + H(S_{N-1},v_{N-1})S_{N-1} - \sum_{i=0}^{N-1} H(S_i,v_i)(S_{i+1} - S_i) + \sum_{i=0}^{N-1} \mathcal{K},\end{aligned}$$

For simplicity assume the investor's payoff at T is $V(S_T) = H(S_{N-1},v_{N-1})S_N + \mathcal{B}_N$. This assumes no liquidation costs (see Section 3.2.2 for more dissuasion).

Then the investor's profit/loss for a fixed trading strategy $H(.,.)$ becomes:

$$\begin{aligned}\Pi_H(\pi,\mathcal{K}) &= V_0 - V(S_T) + \sum_{i=0}^{N-1} H(S_i,v_i)(S_{i+1} - S_i) - \sum_{i=0}^{N-1} \mathcal{K}\\ &= V_0 - V(S_T) + \sum_{i=0}^{N-1} H(S_i,v_i)(S_{i+1} - S_i) - N\mathcal{K}. \qquad (3.3)\end{aligned}$$

Remark 5. *For simplicity, in this thesis, we assume for all* $0 < n \leq N$, $H(S_n,v_n) \neq H(S_{n-1},v_{n-1})$. *In the general case, one should work with* $\mathcal{K}\ \mathbf{1}_{H(S_n,v_n)\neq H(S_{n-1},v_{n-1})}$. *This could substantially change later optimization developments.*

Equation (3.3 helps us to build an inequality which by solving it, one can obtain the optimal strategy. In order to find the no arbitrage interval, we use Definition 6 and apply the same approach as in the previous chapter.

Therefore, in the model with fixed trading costs if there is no arbitrage, we have;

$$V_0 \leq \inf_H\{ \sup_{\pi\in P(S_0,v_0,d)} \{V(S_T) - \sum_{i=0}^{N-1} H(S_i,v_i)(S_{i+1} - S_i) + N\mathcal{K}\}\}.$$

Definition 7. *For a given fixed transaction* $\mathcal{K}$, *then*

$$\overline{V}(S_0,v_0,\mathcal{K}) \equiv \inf_H\{ \sup_{\pi\in P(S_0,v_0,d)} \{V(S_T) - \sum_{i=0}^{N-1} H(S_i,v_i)(S_{i+1} - S_i) + N\mathcal{K}\}\}. \qquad (3.4)$$

is called the minimum upper bound for the fixed transaction cost $\mathcal{K}$.

Similarly,

$$\underline{V}(S_0, v_0, \mathcal{K}) \equiv \sup_H \{ \inf_{\pi \in P(S_0, v_0, d)} \{ V(S_T) + \sum_{i=0}^{N-1} H(S_i, v_i)(S_{i+1} - S_i) + N\mathcal{K} \} \}. \tag{3.5}$$

is called the maximum lower bound for the fixed transaction cost.

Remark 6. *The minimum upper bound has the following simple property, which is straightforward to prove.*

- *It is clear that by increasing transaction cost, the upper bound $\overline{V}(S_0, v_0, \mathcal{K})$ is increasing i.e. if $\mathcal{K}_1 \leq \mathcal{K}_2$ then* [3]

$$\overline{V}(S_0, v_0, \mathcal{K}_1) \leq \overline{V}(S_0, v_0, \mathcal{K}_2).$$

3.2.2 Bellman Equation and Optimal Hedging

Even though, the minimum upper bound $\overline{V}(S_0, v_0, \mathcal{K})$ could be found by a search over paths and strategies, the formulation in equation (3.4) is impracticable. By means of similar arguments as in Chapter 2, the portfolio optimization problem (3.4) can be solved by applying a dynamic programming technique. Towards this end, we define the value function in a recursive way, i.e we can write the problem as the following Bellman equation:

$$\overline{V}(S_i, v_i, \mathcal{K}) = \inf_{h_i} \{ \sup_{c_i} \{ \overline{V}(S_{i+1}, v_{i+1}, \mathcal{K}) - h_i \ (S_{i+1} - S_i) + \mathcal{K} \} \}, \tag{3.6}$$

where $h_i = H(S_i, v_i)$.
With two state variables

$$S_{i+1} = S_i e^{c_i},$$
$$v_{i+1} = v_i - {c_i}^2,$$

where c_i is the control variable, and we set $V(S_N, 0, \mathcal{K}) = V(S_T)$.

How the computation of $\overline{V}(S_0, v_0, \mathcal{K})$ is accomplished in practice is shown in the next chapter.

3.3 Proportional Transaction Costs

In this section, we study optimal hedging strategies and bounds on option prices when there are proportional transactions costs on the stock. Namely, if we want to invest 1 dollar, we will have to pay $\alpha \times 1$ dollars as transaction costs. That is, when at the end of the day we want to re-balance our portfolio,

[3] Note that everything changes when fixed transaction costs enter the scene. Paying the same fixed amount of money for every transaction, the investor clearly goes to bankruptcy if he/she chooses a continuum of trades. Now he/she has to trade at a carefully chosen discrete sequence of instances.

the amount of money obtained from liquidating one position (in order to enter into a new position) from the stock can not all be invested again but part of it has to be used to cover the transaction costs. We look at the re-balancing process in detail in the following.

3.3.1 Hedging Strategies

In this section, we assume that the proportional transaction costs are incurred when shares of the risky asset are traded and on the other hand, it is admitted that trading in the riskless asset is cost-free. Also for simplicity assume that no transaction costs are incurred when a portfolio is established at time 0. But the liquidation costs [4] are incurred when a portfolio is liquidated at the terminal date T.

Remark 7. *A trading strategy is defined, as in the previous chapter, as a function $H(.,.)$ that determines the number of shares that are held as a function of the price S_i and the remaining volatility v_i during each position $i = 0, ..., N-1$ and in addition to our portfolio, we have $\mathcal{B}_i = B_{S_i,v_i}$ as a balance of a money market in order to have a self-financing strategy.*

First, assume V_0 is a given constant, representing the initial value of the portfolio which is the investor's initial capital spent at time 0 to have $H(S_0, v_0)$ units of stock and $\mathcal{B}_0$ in the bank account, so that the amount of money spent for this portfolio including the effect of the transaction costs can be written as

$$\begin{aligned} V_0 &= \mathcal{B}_0 + H(S_0, v_0)S_0 + \alpha|H(S_0, v_0)|S_0 \\ &= \mathcal{B}_0 + H(S_0, v_0)S_0, \end{aligned} \tag{3.7}$$

where $\alpha|H(S_0, v_0)|S_0$ is the transaction cost is at time 0 which is zero by our first assumption, and $0 \leq \alpha < 1$ is called a *round-trip transaction.*

At n^{th} trading position, but before readjusting, the portfolio is composed of $H(S_{n-1}, v_{n-1})$ shares and $\mathcal{B}_{n-1}$ in cash. We denote the value of the portfolio before revision at n by

$$V_n^- := \mathcal{B}_{n-1} + H(S_{n-1}, v_{n-1})S_n, \qquad 0 < n \leq N. \tag{3.8}$$

For re-balancing purposes, the investor requires $H(S_n, v_n)$ number of shares; therefore, a number of $H(S_n, v_n) - H(S_{n-1}, v_{n-1})$ shares need to be traded. In the presence of the proportional transaction costs, the net total dollars exchanged of this operation (the volume of transactions plus the cost of that transaction) is:

$$C(H(S_n, v_n) - H(S_{n-1}, v_{n-1}))\ S_n \equiv (H(S_n, v_n) - H(S_{n-1}, v_{n-1}))S_n + \alpha\,|H(S_n, v_n) - H(S_{n-1}, v_{n-1})|\,S_n. \tag{3.9}$$

An additional term $\alpha|H(S_n, v_n) - H(S_{n-1}, v_{n-1})|S_n$ where $0 \leq \alpha < 1$, on the right-hand side represents

[4] Liquidation is the process of taking a business' real assets and turning them into cash.

the cost incurred when $H(S_n, v_n) - H(S_{n-1}, v_{n-1})$ shares are transacted at n by the purchase (or sell)[5] of the asset S_n. Clearly, it reduces the amount of money intended for the riskless position. The case without transaction costs is recovered by letting $\alpha = 0$.

Remark 8. *For a given trading strategy $H(.,.)$, the total cost of the investor for one success trading at stage n, using equation (3.9), is given by:*

$$\mathcal{C}(\Delta H(S_n, v_n))S_n = \begin{cases} (1+\alpha)\Delta H(S_n, v_n)S_n, & \text{if} \quad \Delta H(S_n, v_n) \geq 0, \\ (1-\alpha)\Delta H(S_n, v_n)S_n, & \text{if} \quad \Delta H(S_n, v_n) < 0. \end{cases}$$

where $\Delta H(S_n, v_n)$ is the number of shares bought/sold, with $0 \leq \alpha < 1$. The above expression satisfies the following properties:

- $\mathcal{C}(\lambda \Delta H(S_n, v_n))S_n = \lambda \mathcal{C}(\Delta H(S_n, v_n))S_n$.
- $\mathcal{C}(\Delta H(S_n, v_n) + \Delta L(S_n, v_n)S_n \leq \mathcal{C}(\Delta H(S_n, v_n))S_n + \mathcal{C}(\Delta L(S_n, v_n))S_n$, *where* $\Delta H(S_n, v_n)$ *and* $\Delta L(S_n, v_n)$ *are the number of shares bought/sold.*
- $\mathcal{C}(\Delta H(S_n, v_n))S_n$ *is non-decreasing as a function of* $\Delta H(S_n, v_n)$.
- $\mathcal{C}(\Delta H(S_n, v_n))S_n \geq \Delta H(S_n, v_n)S_n$ *and* $\mathcal{C}(0) = 0$ [6].

Then, right after the n^{th} trading occasion, the investor balances the portfolio by using a self-financing strategy, and the balance of the operation including transaction costs is added to the money-market account and according to the preceding notation for total cost (3.9). The bank account investment at stage n can be written as

$$\mathcal{B}_n = \mathcal{B}_{n-1} - \mathcal{C}_n(H(S_n, v_n) - H(S_{n-1}, v_{n-1}))\ S_n, \qquad 0 < n \leq N.$$

or

$$\mathcal{B}_{n-1} + H(S_{n-1}, v_{n-1})S_n = \mathcal{B}_n + H(S_n, v_n)S_n + \alpha|H(S_n, v_n) - H(S_{n-1}, v_{n-1})|S_n, \qquad 0 < n \leq N. \tag{3.10}$$

By using (3.8) and (3.10), we have

$$\begin{aligned} V_n^- &= \mathcal{B}_{n-1} + H(S_{n-1}, v_{n-1})S_n \\ &= \mathcal{B}_n + H(S_n, v_n)S_n + \alpha|H(S_n, v_n) - H(S_{n-1}, v_{n-1})|S_n \\ &= V_n^+ + \alpha|H(S_n, v_n) - H(S_{n-1}, v_{n-1})|S_n. \end{aligned}$$

[5] $H(S_n, v_n) - H(S_{n-1}, v_{n-1})$ is the number of shares bought when $H(S_n, v_n) - H(S_{n-1}, v_{n-1}) > 0$ or sold when $H(S_n, v_n) - H(S_{n-1}, v_{n-1}) < 0$.

[6] $0 \leq \mathcal{C}(0) = \mathcal{C}(\Delta H(S_n, v_n) - \Delta H(S_n, v_n))S_n \leq \mathcal{C}(\Delta H(S_n, v_n))S_n - \mathcal{C}(\Delta H(S_n, v_n))S_n = 0$ so $\mathcal{C}(0) = 0$.

Note that, V_n^+ is the value of the portfolio after rebalancing. Clearly if $H(S_n, v_n) = H(S_{n-1}, v_{n-1})$, then $V_n^+ = V_n^-$.

By applying (3.10) recursively, we obtain the total cost of a strategy.

$$\begin{aligned}
V_0 &= \mathcal{B}_0 + H(S_0, v_0)S_0 \\
&= \mathcal{B}_1 + H(S_1, v_1)S_1 - H(S_0, v_0)(S_1 - S_0) + \alpha|H(S_1, v_1) - H(S_0, v_0)|S_1 \\
&= \mathcal{B}_2 + H(S_2, v_2)S_2 - \sum_{i=0}^{1} H(S_i, v_i)(S_{i+1} - S_i) + \sum_{i=0}^{1} \alpha|H(S_i, v_i) - H(S_{i+1}, v_{i+1})|S_{i+1} = ... \\
&= \mathcal{B}_{N-1} + H(S_{N-1}, v_{N-1})S_{N-1} - \sum_{i=0}^{N-2} H(S_i, v_i)(S_{i+1} - S_i) + \sum_{i=0}^{N-2} \alpha|H(S_i, v_i) - H(S_{i+1}, v_{i+1})|S_{i+1}.
\end{aligned}$$

Now assume a European path independent option with cash settlement (liquidation), so the writer who delivers the payoff with liquidation receives $V(S_T)$ including liquidation costs in cash at maturity $T = N$

$$\mathcal{B}_{N-1} + H(S_{N-1}, v_{N-1})S_N = V_N + \alpha|H(S_{N-1}, v_{N-1})|S_N. \tag{3.11}$$

So

$$\begin{aligned}
V_0 &= \mathcal{B}_{N-1} + H(S_{N-1}, v_{N-1})S_{N-1} - \sum_{i=0}^{N-2} H(S_i, v_i)(S_{i+1} - S_i) + \sum_{i=0}^{N-2} \alpha|H(S_i, v_i) - H(S_{i+1}, v_{i+1})|S_{i+1} \\
&= V_N - H(S_{N-1}, v_{N-1})S_N + H(S_{N-1}, v_{N-1})S_{N-1} + \alpha|H(S_{N-1}, v_{N-1})|S_N \\
&- \sum_{i=0}^{N-2} H(S_i, v_i)(S_{i+1} - S_i) + \sum_{i=0}^{N-2} \alpha|H(S_i, v_i) - H(S_{i+1}, v_{i+1})|S_{i+1} \\
&= V(S_T) - \sum_{i=0}^{N-1} H(S_i, v_i)(S_{i+1} - S_i) + \sum_{i=0}^{N-1} \alpha|H(S_i, v_i) - H(S_{i+1}, v_{i+1})|S_{i+1}.
\end{aligned}$$

Note that at maturity $V_N^- = V_N^+ = V(S_T)$ and $H(S_N, v_N) = 0$.

Definition 8. *For a given trading strategy $H(.,.)$, the profit/loss of trading in the presence of the proportional transaction costs is*

$$\Pi_H(\pi, \alpha) = V_0 - V(S_T) + \sum_{i=0}^{N-1} H(S_i, v_i)(S_{i+1} - S_i) - \alpha \sum_{i=0}^{N-1} |H(S_{i+1}, v_{i+1}) - H(S_i, v_i)|S_{i+1}. \tag{3.12}$$

3.3.2 No Arbitrage Condition

In this section, again we consider an investor who uses a trading strategy $H(.,.)$ to hedge the derivative. Assume that some strategy $H(.,.)$ has been fixed, according to (3.12), the investor's profit/loss which depends on permissible path π is

$$\Pi_H(\pi,\alpha) = V_0 - V(S_T) + \sum_{i=0}^{N-1} H(S_i,v_i)(S_{i+1}-S_i) - \sum_{i=0}^{N-1} \alpha|H(S_{i+1},v_{i+1}) - H(S_i,v_i)|S_{i+1},$$

where V_0 is the price received for issuing the contingent claim (a positive number) and $V(S_T)$ is the payout of the derivative which appears with a minus sign as the above expression reflects the balances of the seller of the option. The third term is the hedging profit/loss and it could be negative or positive and the last term is the cost of transactions.

As we defined (Definition 6), if there is an arbitrage then

$$\Pi_H(\pi,\alpha) = V_0 - V(S_T) + \sum_{i=0}^{N-1} H(S_i,v_i)(S_{i+1}-S_i) - \sum_{i=0}^{N-1} \alpha|H(S_{i+1},v_{i+1}) - H(S_i,v_i)|S_{i+1} \geq 0. \tag{3.13}$$

where the inequality is strict for at least one permissible path.

Remark 9. *If the condition in equation (3.13) is satisfied, the seller of the option would certainly receive at least as much, and sometimes more than what he has to pay at the end,* $V(S_T) + \sum_{i=0}^{N-1} \alpha|H(S_{i+1},v_{i+1}) - H(S_i,v_i)|$.

Now we set up the arbitrage bound by assuming a fixed trading strategy, say $H^*(.,.)$. This trading will be a deterministic function of all the possible values of the remaining volatility and stock price.

Since we do not want to allow arbitrage strategies, it follows that the price that we obtain today, V_0, cannot be larger than the maximum cost incurred in paying the final payoff of the claim, and accumulating the hedging profits/losses for the chosen strategy over all the permissible paths, π:

$$V_0 < \sup_{\pi\in P(S_0,v_0,d)} \{V(S_T) - \sum_{i=0}^{N-1} H^*(S_i,v_i)(S_{i+1}-S_i) + \sum_{i=0}^{N-1} \alpha|H^*(S_{i+1},v_{i+1}) - H^*(S_i,v_i)|S_{i+1}\}. \tag{3.14}$$

For the next step, we want equation (3.14) to hold for all trading strategies. Therefore, the price we receive from the sale of the option today must be no greater than even the lowest possible value of the quantity over all the possible trading strategies:

$$V_0 \leq \inf_H\{ \sup_{\pi\in P(S_0,v_0,d)} \{V(S_T) - \sum_{i=0}^{N-1} H(S_i,v_i)(S_{i+1}-S_i) + \sum_{i=0}^{N-1} \alpha|H(S_{i+1},v_{i+1}) - H(S_i,v_i)|S_{i+1}\}\},$$

where the right hand side of the equation is the minimum upper bound for the option price V_0 and we denote it as $\overline{V}(S_0, v_0, \alpha)$.

$$\overline{V}(S_0, v_0, \alpha) \equiv \inf_{H}\{\sup_{\pi \in P(S_0,v_0,d)}\{V(S_T) - \sum_{i=0}^{N-1} H(S_i, v_i)(S_{i+1} - S_i) + \sum_{i=0}^{N-1} \alpha|H(S_{i+1}, v_{i+1}) - H(S_i, v_i)|S_{i+1}\}\}. \tag{3.15}$$

3.3.3 Dynamic Programming and Bellman Equation

The minimum upper bound in Equation (3.15) could in principle be found by a search over paths and strategies. While conceptually correct, this approach is computationally expensive and impracticable, because it requires a search over all possible paths and hedging strategies. In order to make the problem more tractable, as we mentioned in Chapter 2, we can apply dynamic programming. Note that, since the current price of the asset, $S(0)$, and the quadratic variation v_0, determine both the permissible price sequences, $P(S_0, v_0, d)$, and the trading strategies, $H(.,.)$, the function $\overline{V}(S_0, v_0, \alpha)$ defined in (3.15) provides the minimum upper bound as a function S_0 and v_0. More precisely, $\overline{V}(S_0, v_0, \alpha)$ is a function with domain given by all the possible values of $S(0)$ and v_0. By following the same processes as in the previous chapter, we can transform equation (3.15) into:

$$\overline{V}(S_i, v_i, \alpha) = \inf_{h_i}\{\sup_{c_i}\{\overline{V}(S_{i+1}, v_{i+1}, \alpha) - h_i(S_{i+1} - S_i) + \alpha|h_{i+1} - h_i|S_{i+1}\}\}, \tag{3.16}$$

where $h_{i+1} = H(S_{i+1}, v_{i+1})$ is known, and $\overline{V}(S_N, 0, \alpha) = V(S_T)$.

Remark 10. *To move from Equation (3.15) to the Bellman Equation (3.16) several small but important changes have taken place.*

a) *The sup is no longer taken over paths, it is over at the next step, i.e. S_{i+1} and v_{i+1} .*

b) *For a given (S_i, v_i) we have a finite number of possible terminal destinations (S_{i+1}, v_{i+1}) to keep track of, instead of a multitude of connecting paths.*

c) *The sum over time-steps of the transaction costs terms and the cost of the hedging have disappeared, since we are now dealing with a single time-step.*

d) *The term $\overline{V}(S_{i+1}, v_{i+1}, \alpha)$ no longer indicates the terminal pay-out, but simply the value of the minimum upper bound itself at the next time-step.*

Let us now implicitly define a variable c_i by the relationship:

$$S_{i+1} = S_i e^{c_i},$$
$$v_{i+1} = v_i - c_i^2.$$

Therefore Equation (3.16) can be rewritten as

$$\overline{V}(S_i, v_i, \alpha) = \inf_{h_i}\{\sup_{c_i}\{\overline{V}(S_i e^{c_i}, v_i - c_i^2, \alpha) - h_i S_i(e^{c_i} - 1) + \alpha|h_{i+1} - h_i|S_i e^{c_i}\}\}$$

Subject to the conditions:

$$c_i^2 \leq v_i, \qquad c_i^2 \leq d^2.$$

The initial boundary condition are given by the initial values of the state variable, S_0 and v_0. The terminal boundary condition is $V(S_i, 0, \alpha) = V(S_T)$ without including the liquidations costs. Note that the terminal price S_T is free, as is the number of price moves N. The above implementation can be developed as a numerical approach for valuing European options, in the following chapter we will describe an interesting algorithm for this purpose introduced by Britten-Neuberger.

Chapter 4

Numerical Implementation

In Chapter 2, we described how the price of the option can be found by solving recursively the following dynamic optimization problem:

$$\overline{V}(S_i, v_i) = \inf_{h_i}\{\sup_{c_i}\{\overline{V}(S_i e^{c_i}, v_i - c_i^2) - h_i S_i(e^{c_i} - 1)\}\}.$$

As we mentioned, even though we cannot solve for $\overline{V}(S_i, v_i)$ analytically, we can compute this quantity numerically.

In this chapter, we describe a numerical implementation to evaluate the bounds described in previous chapters. The numerical approach is based on the method described in [Britten and Neuberger, 1996]. We investigate the upper bound and the lower bound by using a kind of multinomial tree that allows us to introduce the necessary constraints of the price path. We also briefly describe the optimization methods that we use in this thesis.

4.1 Jones-Neuberger Algorithm

We will refer to the numerical algorithm to compute the minimum upper bound $\overline{V}(S_0, v_0)$[1], which is proposed by Britten-Jones and Neuberger, as the J-N Algorithm (Jones-Neuberger Algorithm). In the present chapter, we provide implementation details as well as a conceptual analysis of the computations being performed.

The J-N algorithm that computes $\overline{V}(S_0, v_0)$ in both instances, with and without transaction costs, makes use of the Bellman's equation. Let us first consider the problem in the case of no transaction costs. In the general case, namely the case with transaction costs, the algorithm remains the same with only small

[1] In the case of $\underline{V}(S_0, v_0)$, the algorithm remains the same, we can evaluate the maximum lower bound by using $\underline{V}(Z) = -\overline{V}(-Z)$ where Z is the given payoff.

changes in the optimization constraint.

To see how we can implement the bound $\overline{V}(S_i, v_i)$, we need to recall Bellman's equation,

$$\overline{V}(S_i, v_i) = \min_{h_i}\{\max_{c_i}\{\overline{V}(S_i e^{c_i}, v_i - c_i^2) - h_i S_i(e^{c_i} - 1)\}\}.$$

We use backward induction as one method of solving the above dynamic optimization (Appendix D). If we are working from $i = T$ to today's $i = 0$, $\overline{V}(S_i, v_i)$ is the new value of the option that we have to determine. $V(S_i e^{c_i}, v_i - c_i^2)$ are option's values that have been already calculated, note that we use the fact that $V(S_T)$ at T is known[2]. Since at each stage, we are holding a position h_i; therefore, the change in the value of the portfolio,

$$\overline{V}(S_i e^{c_i}, v_j - c_i^2) - h_i S_i(e^{c_i} - 1), \tag{4.1}$$

is a linear combination of the stock price S_i and $\overline{V}(S_i e^{c_i}, v_j - c_i^2)$. So it seems that we have a piecewise linear optimization where decision variable is h_i.

What we do not know is the value of h_i in (4.1), i.e. the amount of stock to be held at i. But we do know that $\overline{V}(S_i, v_i)$ must be an upper bound. So, an optimization possibility is to find the slope of the line which is h_i, such that $\overline{V}(S_i, v_i)$ attains a maximum.

Before providing the method of optimization, we have to construct the discretization of the permissible path $\{S_i\}_{i=0}^N$ and the remaining quadratic variation $\{v_i\}_{i=0}^N$. In sections (4.1.1)-(4.1.4), we provide the necessary details to build the multinomial tree which is going to be used for our optimization problem.

4.1.1 The Mesh of the J-N Algorithm

As a first introduction to the methodology, we will build a tree which is slightly different than the traditional binomial tree. Recall that in the traditional binomial tree (Appendix B), one normally builds the tree by specifying a time-step, Δt, and volatility σ for a given S_0. There are two possibilities for the next stock price,

$$S_{up} = S_0 \exp \sigma\sqrt{\Delta t},$$

and

$$S_{down} = S_0 \exp - \sigma\sqrt{\Delta t}.$$

According to the J-N algorithm, the remaining quadratic variation v_j, where j is a positive integer, is mapped onto the x-axis. This should be contrasted with the usual approach where the time variable is the one mapped onto the x-axis. Then, we set up the grid similarly to the traditional binomial in the following way.

[2] The set up is quite similar to the usual Block-Scholes framework where the initial and boundary conditions allow to solve the second order partial differential equation, the J-N numerical method is very similar to a binomial tree.

We consider a grid of points of the form (i,j), j represents the label of the remaining volatility v_j and is non-negative and i is the label of the price path S_i and can be positive or negative. If we use N steps in the tree, the domain of j is $[0,N]$ while the domain of i is $[-N,N]$. In figure (4.1), a 11-step tree is displayed.

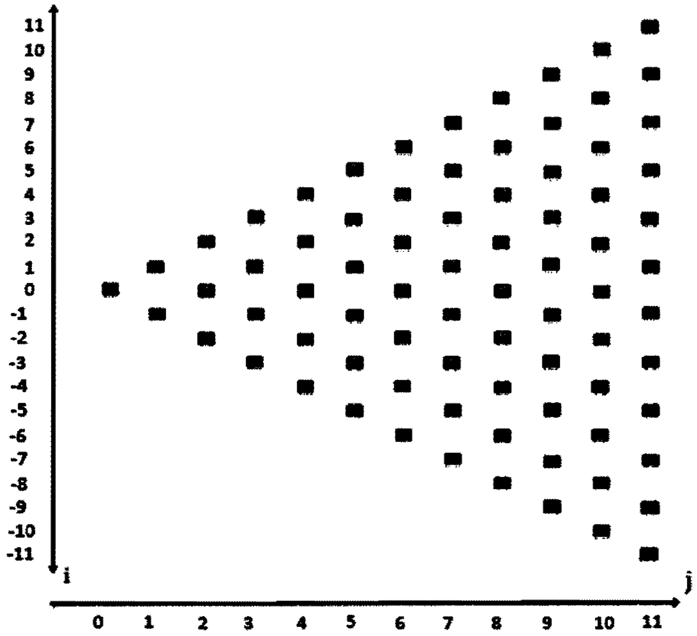


Figure 4.1: The sample grid with $N = 11$ steps in the tree construction, in this case for any pair (i,j), $i \in [-11,11]$ and $j \in [0,11]$. Starting from $(0,0)$ and end up to $(i,11)$ where $i = -11, ..., 11$.

The following shows a Matlab function to create the grid [3].

The Matlab Code to Create the Grid

```
function  [i] = matrixI(N)
 i = (2*N+1)ones(N+1,N+1);
 i(:,1) = (-N:2:N)';
   for c = 2:N+1
    a = 1;
    I(c) = (N-(c-1));
       for e = -I(c):2:I(c)
         i(a,c)= e;
         a = a+1;
       end
   end
```

[3] In this thesis, we make use of Matlab for the computer implementation.

4.1.2 J-N Discretization for a Special Case

After building the grid, we construct the basic building block for our tree. Assume we are working in the $(i-1)^{th}$ step where the price stock is denoted by S_{i-1}; what are the possible successors of S_i?

In Chapter 2, we introduced the control variable c_i such that

$$S_i = S_{i-1}\exp(c_{i-1}).$$

To discretize $\{S_i\}_{i=0}^{N}$, we assume all control variables $\{c_i\}_{i=0}^{N-1}$ are equal to some arbitrary step size δ. The discrete grid of stock values is given by:

$$S_i = S_0 \exp(i\delta), \tag{4.2}$$

where i can be positive or negative.

In order to define the discrete set of values for the variables v_j , we use the definition of the remaining quadratic variation (Definition 3). First, we set $v_{N_2} = 0$ and use backward induction, the discretization for the quadratic variation is given by

$$v_j = (N_2 - j)\delta^2 \quad \& \quad v_0 = N_2\, \delta^2, \tag{4.3}$$

where $j \geq 0$ and at maturity T, $j = N_2$.

4.1.3 J-N Discretization for the General Case

In a binomial tree, starting from any point (i,j), the next point reached must be $(i+1, j-1)$ or $(i-1, j-1)$. The value of the claim at (i,j) is then computed from the value at $(., j-1)$. In the J-N tree, the same procedure applies but with one difference that is related to the following question: how far the investor can wait until the next trading?

To answer this question, recall that from Chapter 2, the investor can trade whenever the following condition is satisfied;

$$|\ln S_{i+1} - \ln S_i| \leq d.$$

If the trader is in the lattice point (i,j), i' is the next point for the price path π such that

$$\begin{aligned} |\ln S_{i'} - \ln S_i| &= \left|\ln S_0 e^{(i'\delta)} - \ln S_0 e^{i\delta}\right| \\ &= |(i'-i)\delta| \leq d. \end{aligned}$$

It then follows that $-d \leq n\delta \leq d$ where $i' - i = n$ and $n = \pm 1, .., \pm\frac{d}{\delta}$. Therefore we have the first constraint given by;

$$n^2 \leq \frac{d^2}{\delta^2}. \tag{4.4}$$

In addition to the condition (4.4), by making use of $S_{i'} = S_i\ e^{n\ \delta}$ and $v_j = (N-j)\delta^2$ we have

$$\begin{aligned} v_{j'} &= v_j - (\ln S_{i'} - \ln S_i)^2 \\ v_{j'} &= v_j - (n\delta)^2 \\ (N_2 - j')\delta^2 &= (N_2 - j - n^2)\delta^2 \\ &\geq 0. \end{aligned}$$

So, we arrive at a second constraint, which is imposed to maintain the non-negative character of the remaining quadratic variation,

$$n^2 \leq N_2 - j. \tag{4.5}$$

The above results (4.5) and (4.4) correspond to

$$n^2 \leq \min(N_2 - j, \frac{d^2}{\delta^2}), \tag{4.6}$$

where n is a non-zero integer.

The length N of the discretized trajectories satisfies $N_1 \leq N \leq N_2$, where N_1 is the largest integer such that $N_1 d^2 \leq v_0$.

Therefore, in order to account for the possibility of jumps d, from a given parent node (i,j), we modify the possible children nodes as $(i+n, j+n^2)$, with n a positive or negative (but not zero) integer which satisfies (4.6).

Figure 4.2 illustrates all possible children nodes for parent node (S_i, v_j) when $n=3$.

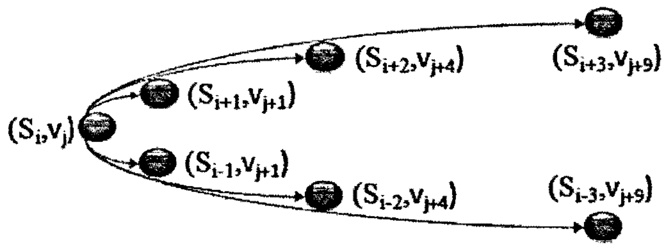


Figure 4.2: This figure displays a very simple example for all destination points form node (i,j) when $n=3$.

4.1.4 Option Price at Each Stage in J-N Algorithm

In this section, we explain how one can obtain the option price at each stage j.

Option Price At Maturity $j = N_2$:

According to Definition 3, at expiration, there is no quadratic variation left. Therefore, because of the construction of the price path, we will not be able to move up any more. So, we must simply have the terminal payoff condition and it is not dependent on which node we might have arrived from;

$$V(S_T, 0) = V(S_T) = \max(S_T - K, 0). \tag{4.7}$$

Option Price At $j = N_2 - 1$:

In the second to last step which is labeled by $j = N_2 - 1$ in the grid, everything must also look exactly the same as in the simple binomial tree; because on any node $(., N_2 - 1)$, we only have one unit of quadratic variation δ^2 left to play before consuming the total quadratic variation v (Figure 4.3). This allows us to only move either to node $(i+1, N_2)$ or $(i-1, N_2)$. In this stage, the option price of the node $(., N_2 - 1)$ can be found as a linear combination of its value (the same as in the usual binomial tree) at nodes $V(S_{i+1})$ and $V(S_{i-1})$.

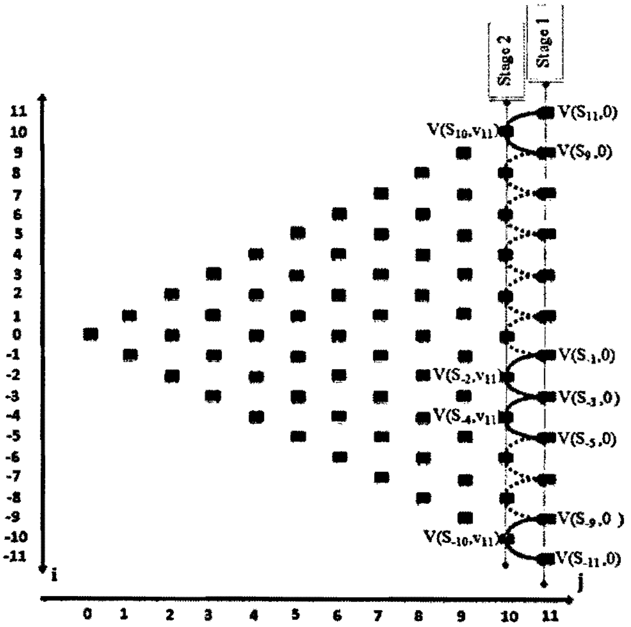


Figure 4.3: In stage 1, we can find the value of payoff $V(S_T, 0)$ simply as $\max(S_T - K, 0)$, then, in stage 2, we can find the value by using the information from stage 1.

Remark 11. *Note that, we might have arrived at a node $(i, N_2 - 1)$ directly either from the root $(i - n, (N_2 - 1) - n^2)$, if such big jumps had been allowed by the d constraint, or from third stage to the last $(i - 1, N_2 - 2)$.*

Option Price At $j > N_2 - 3$:

In the last three steps, the possible values of the option are acting exactly the same as in the simple binomial case because the unit of quadratic variation that is left does not allow jumps. According to (4.5), we need at least $n^2 = 4$ to have a big jump $(i + 4, 0)$ (Figure 4.3).

The procedure of the optimization truly becomes different when we move to the fourth to last step and beyond until we reach the root. Because for any given node (i, j), there are a multiplicity of possible arrival points $(i + n, j + n^2)$ (Figure 4.4).

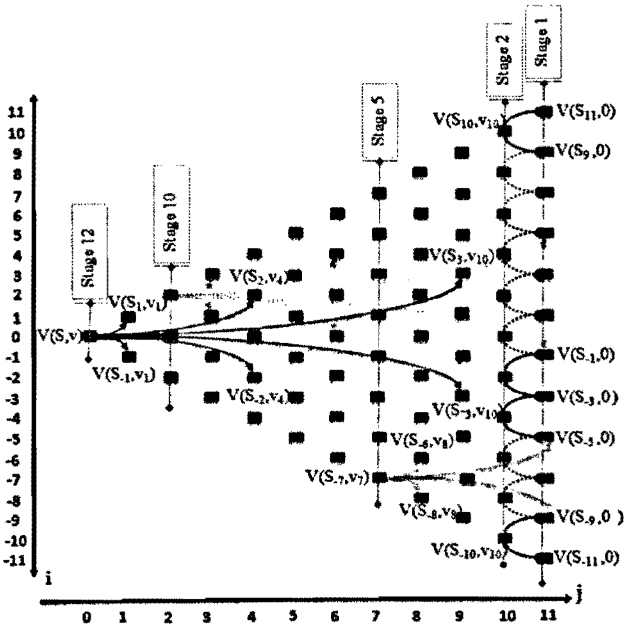


Figure 4.4: This figure shows, when $j \geq (N_2 - 4)$, there are multiple destination points.

In general, one can use the following Matlab code in order to find all destination nodes from a certain node (i, j).

The Matlab Function to Find all Destination Nodes

```
function [Value1, Stock1] = destpoint(n1,time,h,Stock0,Value,delta,i)
 for  n=-n1:n1
    if  n~=0
       if  time-1>=n^ 2
         [tf, index] = ismember(i(h,time)+n,i(:,time-n^ 2));
         Value1(p) = Value(index,time-n^ 2);
         Stock1(p) = Stock0*exp((i(h,time)+n)delta);
         p = p+1;
       end
    end
 end
```

4.1.5 Optimal Hedging and Convex Hull

With a multiplicity of possible destination nodes, we can not value $\overline{V}(S_i, v_j, .)$ as a linear combination of its values at the destination nodes. We need to find a hedge ratio which minimizes the maximum possible loss. At this point we need to perform a minimax optimization.

$$\overline{V}(S_i, v_i) = \min_h\{\max_n\{\overline{V}(S_i e^{n\delta}, v_i - (n\delta)^2) - hS_i(e^{n\delta} - 1)\}\}, \quad n = \pm 1, .., \pm\frac{d}{\delta},$$

where $h = h_i$.

$V(S_i e^{n\delta}, v_i - (n\delta)^2)$ are known option values (that have already been calculated), so the optimization is a linear optimization. We use two methods of optimization in this thesis, Convex Hull and Piecewise linear optimization.

Convex Hull

In order to solve the above optimization numerically, Britten-Neuberger used a method which they called *Convex Hull*:

If $\tilde{h}$ is the optimal hedge, then

$$\overline{V}_k(S_i, v_i) = \overline{V}(S_i e^{k\delta}, v_i - (k\delta)^2) - \tilde{h}S_i(e^{k\delta} - 1), \qquad k = \pm 1, \pm 2, ... \pm n. \tag{4.8}$$

We do know $\overline{V}(S_i, v_i)$ must be the upper bound, so there is a k' such that

$$\overline{V}_{k'}(S_i, v_i) \geq \overline{V}_k(S_i, v_i), \qquad k = \pm 1, \pm 2, ... \pm n.$$

This implies that

$$\begin{aligned}\tilde{h} &\leq \frac{\overline{V}(S_ie^{k'\delta}, v_i - (k\delta)^2) - \overline{V}(S_ie^{k\delta}, v_i - (k\delta)^2)}{S_ie^{k'\delta} - S_ie^{k'\delta}} \\ &= \frac{\overline{V}(S_{i+k'}, v_{i-k'^2}) - \overline{V}(S_{i+k}, v_{i-k^2})}{S_{i+k'} - S_{i+k}}.\end{aligned}$$

If we draw all the destination asset prices S_{i+k} with respect to their values $\overline{V}(S_{i+k}, .)$ and find a line which passes through two distinct (destination) points such that all (destination) points lie at this line or below of this line (this line is called the Hull line in [Britten and Neuberger, 1996]), then the slope Δ of the Hull line is the desired optimal hedge (see the dash line in Figure 4.5). Assume that this line is the line that passes through the points $(\overline{V}_m, S_m)$ and $(\overline{V}_{m'}, S_{m'})$, (Figure 4.5). So,

$$\Delta = \frac{\overline{V}(S_m) - \overline{V}(S_{m'})}{S_m - S_{m'}} \leq \frac{\overline{V}(S_{i+k'}, v_{i-k'^2}) - \overline{V}(S_{i+k}, v_{i-k^2})}{S_{i+k'} - S_{i+k}}. \tag{4.9}$$

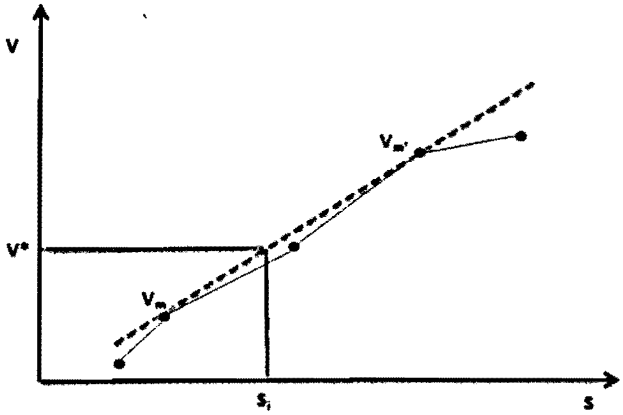


Figure 4.5: The slope of the dashed line is the optimal hedge h_i and $V^*(S_i)$ is the optimal value of the option at S_i at node (i,.).

The following lemma guarantees that the optimal value of the function $\overline{V}(i,j)$ is the value of V^* (Figure 4.5), where the Hull line crosses the vertical line through the current price S_i. Then as indicated above, the hedging ration Δ, which is the value of the slope of the hull line at that point, is the optimal hedging strategy according to (4.9). So in order to find the optimal hedging, it is enough to find the hull line, the slope of the line at that point $(S_i, \overline{V}(S_i, v_i))$ is the hedging h_i. This method is called convex hull optimization [4].

[4] The computer code is provided in an Appendix.

Lemma 1. *Let $g(.)$ be a convex function then*

$$g(x) = \min_{h\ \in\mathbb{R}}\{\max_{y}\{g(y) - h\ (y-x)\}\}, \tag{4.10}$$

where $y \in [xe^{-d}, xe^{d}]$.

Proof. We obviously have

$$\min_{h\ \in\mathbb{R}} \max_{xe^{-d}\leq y\leq xe^{d}}\{g(y) - h\ (y-x)\} \leq g(x). \tag{4.11}$$

for all h, since the value at the right hand side is achieved at the left hand side for $y = x$. So to complete the proof it suffices to show that there exists h such that

$$g(y) - h(y-x) \geq g(x), \tag{4.12}$$

for all y. Clearly, any sub gradient $\frac{g(y)-g(x)}{y-x}$ of g at x can serve as such. □

Piecewise Linear

Although the convex hull method is a very fast and efficient way to solve the previous optimization we have not used it for the case when we include transaction costs. The reason for this is mainly due that some of the numerical results were not quite as good as we expected; moreover, we were not able to extend the technique to the case of transaction costs in a nice and convenient way. All in all, we ended up using the piecewise linear optimization method.

In general we have the following problem to be optimized [Tsitsiklis, 1997]:

$$\overline{V}(S_i, v_i) = \min_{h}\{\max_{k}\{\overline{V}(S_ie^{k\delta}, v_i - (k\delta)^2) - hS_i(e^{k\delta} - 1)\}\}, \quad k = \pm 1, \pm 2, \ldots \pm n \tag{4.13}$$

Note that $\max_{k}\{\overline{V}(S_ie^{k\delta}, v_i - (k\delta)^2) - \tilde{h}S_i(e^{k\delta} - 1)\}$ is equal to the smallest number z that satisfies:

$$z \geq \overline{V}(S_ie^{k\delta}, v_i - (k\delta)^2) - hS_i(e^{k\delta} - 1), \text{ for all } \quad k = \pm 1, \pm 2, \ldots \pm n.$$

For this reason, the optimization problem is equivalent to the linear programming problem

$$minimize\ \ z$$

Subject to

$$z \geq \overline{V}(S_ie^{k\delta}, v_i - (k\delta)^2) - hS_i(e^{k\delta} - 1), \text{ for all } \quad k = \pm 1, \pm 2, \ldots \pm n, \tag{4.14}$$

where the decision variables are z and h.

The following Matlab function shows the code of above constraint.

The Matlab function for optimization Equation 4.13

```
function [V,Vh] = opthedge(Value1,Stock1,Stockhtime)
      f1 = [1; 0];                                  %f1 is z
      b=cat(2,-Value1);                            % response for Ax<b as the constraint
      col1 =cat(2,-ones(1,l(Value1)));
      col2 = cat(2,-(Stock1-Stockhtime));
      A=cat(2,col1',col2');                         % response for Ax<b as the constraint
      lb = [-10^4;-10^4];                            % lb and ub are upper and lower for outputs
      ub =[10^4;10^4];
      [x,fval] = linprog(f1,A,b,[],[],lb,ub);       % optimization
      Vh = x(2);                                     % optimal hedging
      V = x(1);                                     % optimal value
```

In the presence of transaction costs the approach is the same as above (4.13)

$$V(S_i, v_i, \alpha) = \min_h\{\max_k\{\overline{V}(S_i e^{k\delta}, v_i - (k\delta)^2) - hS_i(e^{k\delta} - 1) + \alpha|h - g_k|S_{i+1}\}\}. \qquad (4.15)$$

g_k is known (it is calculated in the previous stage, $i-1$). Let $|h - g_k| = y_k$, our first reformulation yields

$$\overline{V}(S_i, v_j, \alpha) = \min_h\{\max_k\{\overline{V}(S_i e^{k\delta}, v_j - (k\delta)^2) - hS_i(e^{k\delta} - 1) + \alpha\ .y_k.S_{i+1}\}\}.$$

Subject to

$$h - g_k \leq y_k,$$
$$-h + g_k \leq y_k.$$

And, by the same process we applied in (4.13),

$$minimum\ \ z$$

Subject to

$$\overline{V}(S_i e^{k\delta}, v_j - (k\delta)^2) - hS_i(e^{k\delta} - 1) + \alpha y_k S_{i+1} \leq z,$$
$$h - g_k \leq y_k,$$
$$-h + g_k \leq y_k.$$

The Matlab Codes to Create Necessary Condition for Optimization

```
function[V,Vh] = opthedgetrans(Value1,Stock1,xi1,alpha,Stockhtime)
K1 = length(Value1);
f1 = cat(2,1 ,0,zeros(1,K1));                 %f1 is z
b = cat(2,-Value1,xi1,-xi1);                   % b for Ax<b, included the absolute condition
A1 = cat(2,-ones(K1,1),(-(Stock1-Stockhtime))',alpha.*diag(Stock1));
A2 = cat(2,zeros(K1,1),ones(K1,1),-eye(K1));
A3 = cat(2,zeros(K1,1),-ones(K1,1),-eye(K1));
A = cat(1,A1,A2,A3);                          % A for Ax<b, included the absolute condition
lb = cat(2,-10^4,-10^4,zeros(1,K1));
ub = cat(2,10^4,10^4,10Λ4ones(1,K1));
[x,fval] = linprog(f1,A,b,[],[],lb,ub);      % optimization
Vh = x(2);                                    % optimal hedging
V = fval;                                     % optimal value
```

We used a Matlab optimization from the optimization toolbox by the name of linprog. There are three types of algorithms that are being implemented in the linprog.m; a simplex algorithm, an active-set algorithm, a primal-dual interior point method [Geletu, 2007].

We make use of linprog along the destination points $V(i+n, j+n^2)$, we can then move backwards the same procedure all the way to the root. In the next chapter, we present some numerical results obtained by using the algorithms described in the present chapter.

Chapter 5

Numerical Analysis

In this chapter, we implement two numerical tests of the J-N algorithm to show some practical aspects of it. First, we compute the arbitrage bounds of the option price on the stock in some artificial permissible paths with the J-N algorithm and then we study the effects of transaction costs in both cases with fixed and proportional transaction costs. In the second part, by means of the J-N algorithm, we find the option's price for a real data set by somehow forecasting the quadratic variation from historical data and then comparing the results with actual market prices.

5.1 Artificial Paths and General Results

As a first step, we compute numerical arbitrage bounds for a two-month European call option with strike of \$1 on a stock that pays no dividends, with current price \$1 and the volatility of the stock is taken to be equal to $\sigma = 20\%$ per year, so the quadratic variation is

$$v_0 = \sigma^2 T = 0.04 * 2/12.$$

We apply the numerical algorithm described in the previous chapter to our base case model when no transaction costs exist. Recall that for a given N_2 which is the maximum number of steps taken in the J-N tree, the minimum return $\delta = \min_{0 \leq i \leq N} |\ln(S_{i+1}) - \ln(S_i)|$ can be obtained by [1]

$$\delta = \sqrt{\frac{v_0}{N_2}} = \sqrt{\frac{0.0067}{N_2}}.$$

Remark 12. *The values, which are taken in this section, such as the current price and the strike price, are for illustrative purposes only and they do not reflect actual values in the real markets.*

Remark 13. *Option value in figures means the minimum upper bound price $\overline{V}$ or the maximum lower bound price $\underline{V}$ of an option.*

[1] According to Section 4.1.2 Chapter 4, $v_0 = N_2\delta^2$ where δ is the minimum step size that one can take.

5.1.1 Comparison to Black-Scholes

If no jumps are allowed, i.e. $\frac{d}{\delta} = 1$ (we call this ratio: the jump units) then the Black-Scholes price [2] is \$0.033 when $S_0 = K = 1$ with $\sigma = 0.2$. Figure 5.1 displays the minimum upper bound price of the option as a function of N_2. As we see, the option price converges to the Black-Scholes price as N_2 increases. According to equation (4.3), by increasing N_2, we are reducing δ. So as a clear result, when δ decreases, the value of the option increases and tends to the Black-Scholes price.

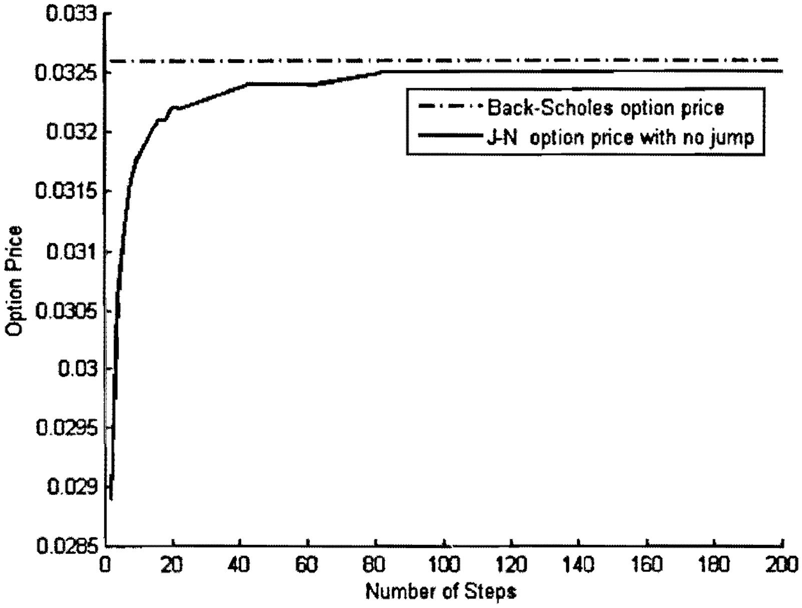


Figure 5.1: The figure shows the model's price, when the jump unit $(\frac{d}{\delta}) = 1$, approaches to the Black-Scholes price as the number of steps N_2 increases.

One may wonder what happens if $\frac{d}{\delta} > 1$. Figure 5.2 shows the convergence behavior of the option price when $\frac{d}{\delta} = 3, 5$ with N_2 ranging from 10 to 200. When the jump units is greater than one, it seems that we have convergence if N_2 increases but the value of the option does not converge to Black-Scholes price due to jumps. Also, for a given v_0 and a fixed number of steps N_2 in the J-N tree, by increasing $\frac{d}{\delta}$, due to the fact that δ is fixed, we are increasing the maximum jump size d, and we can see that the value of the option increases (Figure 5.2).

As we can see, when $N_2 = 10$, in the case of the jump units are 3 and 5, the minimum upper bounds are equal, because according to equation (4.5), the maximum jump that the algorithm can take is $\sqrt{N_2} \approx 3$. Therefore, although we can run the program for the jump unit 5, this jump does not really take into

[2] Formally the Black-Scholes model is not nested in our model. This is due to the fact that Black and Scholes is a continuous time model while we consider a discrete time model. Nevertheless one can overcome this technical difference using the fact the Black-Scholes model is a limit of discrete time models.

account and thus does not affect the price of the option in the J-N algorithm.

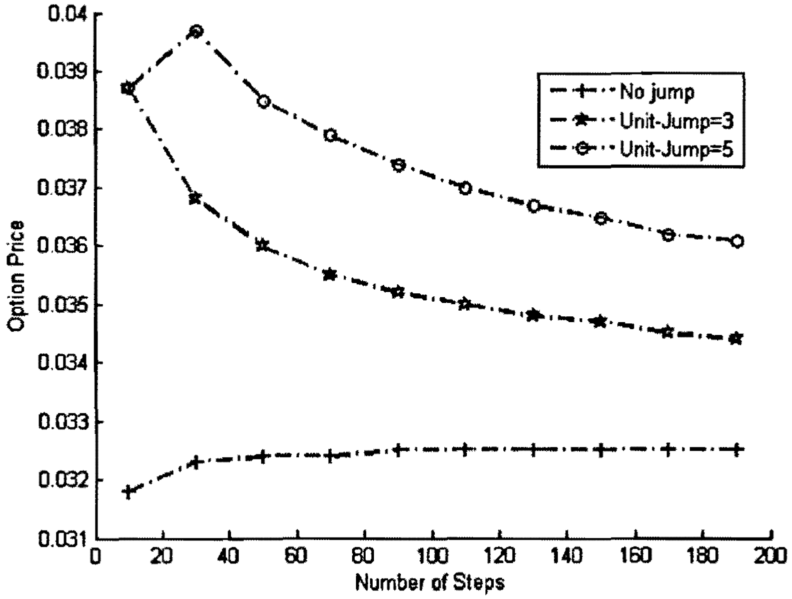


Figure 5.2: Convergence of $\overline{V}$ as a function N_2 for different values of $\frac{d}{\delta}$.

5.1.2 Arbitrage Bounds: Basic Model

In our empirical work, we shall use the J-N algorithm to find numerically the arbitrage bounds of the option price. First, we derive arbitrage bounds in the basic model (Chapter 2) for different jump units with the same set up as above i.e. $K = 1$, $\sigma = 20\%$ per year. We calculate the option values for different starting levels of the stock price S_0, with $N_2 = 100$ steps in the tree (trading around twice a day).

Figure 5.3 displays the arbitrage bounds for different jump units [3] $\frac{d}{\delta} = 3, 7$ and apparently as the maximum jump is decreasing, we can see that the upper and lower bounds are shrinking to the case without jumps. Also we notice that the arbitrage bounds are very narrow for the higher starting level S_0. Therefore it seems that the jumps have less of an effect on the bound of the option's prices for the higher stock values. Again, for a fixed S_0, $\overline{V}(S_0, v_0)$ increases as $\frac{d}{\delta}$ increases (Figure 5.3).

Table 5.1 illustrates no arbitrage interval with different jump units as the level of starting S_0 changes.

[3] Recall that $(\frac{d}{\delta})$ is representing the jump units.

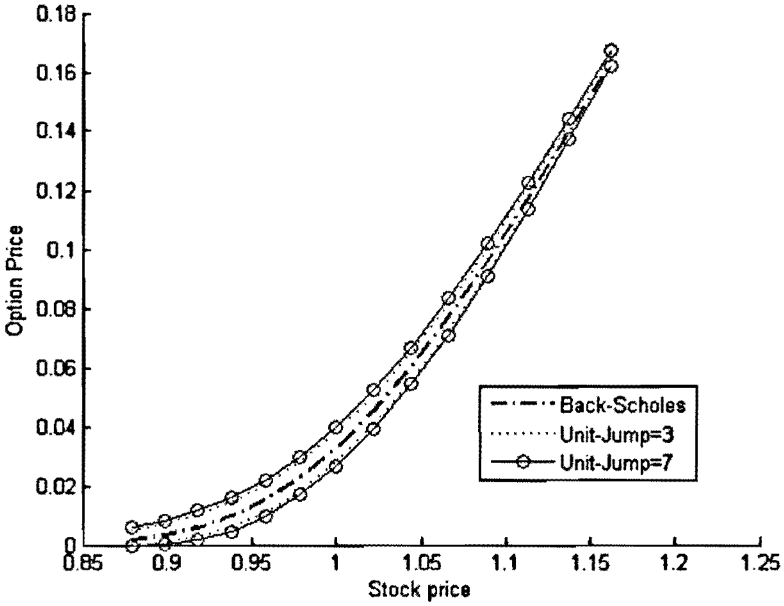


Figure 5.3: The upper and lower bounds for the basic model: the plot shows that by reducing the jump units, for example, from 7 to 1, the option price bounds are shrinking to the Black-Schoes price.

Although, most of the analysis in this thesis was done for a European payoff $\max(S_T - K, 0)$, the general approach clearly applies to derivatives with different payoffs, Figure 5.5 shows the results for some known combinations of calls and puts.

First, we use a call *Butterfly payoff* [Hull, 1997], which is a combination of long call with strike $K_1 = 0.958$ and one long put with $K_2 = 1.066$. Its payoff is:

$$V_b = \begin{cases} \max(S_T - K_1, 0), & \text{if } S_T \leq \frac{K_2+K_1}{2}, \\ \max(K_2 - S_T, 0), & \text{if } S_T > \frac{K_2+K_1}{2}. \end{cases} \tag{5.1}$$

Also we consider a *Long Strangle* [Hull, 1997], which is long one put option with a lower strike $K_1 = 0.958$ price and long one call option at a higher strike price $K_2 = 1.066$ to calculate the minimum upper bound.

$$V_l = \begin{cases} \max(K_1 - S_T, 0), & \text{if } S_T \leq \frac{K_2+K_1}{2}, \\ \max(S_T - K_2, 0), & \text{if } S_T > \frac{K_2+K_1}{2}. \end{cases} \tag{5.2}$$

As it can be observed in Figure 5.5, the shape of bounds are very similar to their payoff at maturity T (Figure 5.4) in both cases. The area between two strikes $K_1 = 0.958$ and $K_2 = 1.066$ becomes bigger for an option with payoff V_l and V_b. And again by increasing the jump units, the arbitrage bounds of

$\frac{d}{\delta}$	1	3	5	7	9
S_0	$[\underline{V},\overline{V}]$	$[\underline{V},\overline{V}]$	$[\underline{V},\overline{V}]$	$[\underline{V},\overline{V}]$	$[\underline{V},\overline{V}]$
0.879	0.0019	[0.0008,0.0032]	[0.0002,0.0044]	[0,0.0055]	[0,0.0067]
0.958	0.0151	[0.0129,0.0178]	[0.0110,0.0199]	[0.0100,0.0215]	[0.0082,0.0232]
1	0.0327	[0.0305,0.0354]	[0.0272,0.0376]	[0.0277,0.0392]	[0.0268,0.0407]
1.044	0.0598	[0.0574,0.0683]	[0.0554,0.0647]	[0.0544,0.0664]	[0.0525,0.0681]
1.066	0.0764	[0.0492,0.0790]	[0.0723,0.0811]	[0.0713,0.0827]	[0.0690,0.0844]
1.162	0.1631	[0.1623,0.1642]	[0.1620,0.1654]	[0.1619,0.1666]	[0.1618,0.1677]

Table 5.1: This table shows no arbitrage intervals for different jump units when the staring level S_0 increases.

the option price become wider (Figure 5.5)).

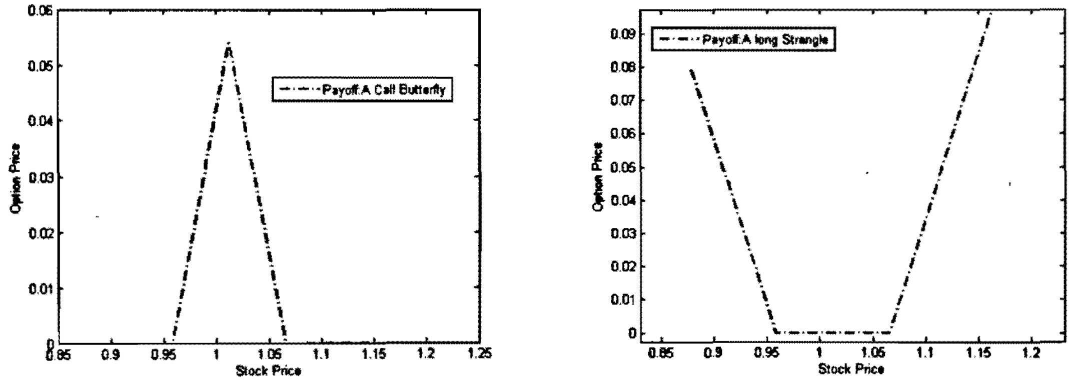


Figure 5.4: Left-hand side figure shows the payoff for a call butterfly and right-hand side figure shows the payoff for a long strangle with a lower strike $K_1 = 0.958$ and a higher strike price $K_2 = 1.066$.

Table 5.2 and 5.3 show no arbitrage intervals (upper and lower bounds) with jump units 1,3,5 and 7 for a derivative with payoff 5.1 and 5.2 respectively.

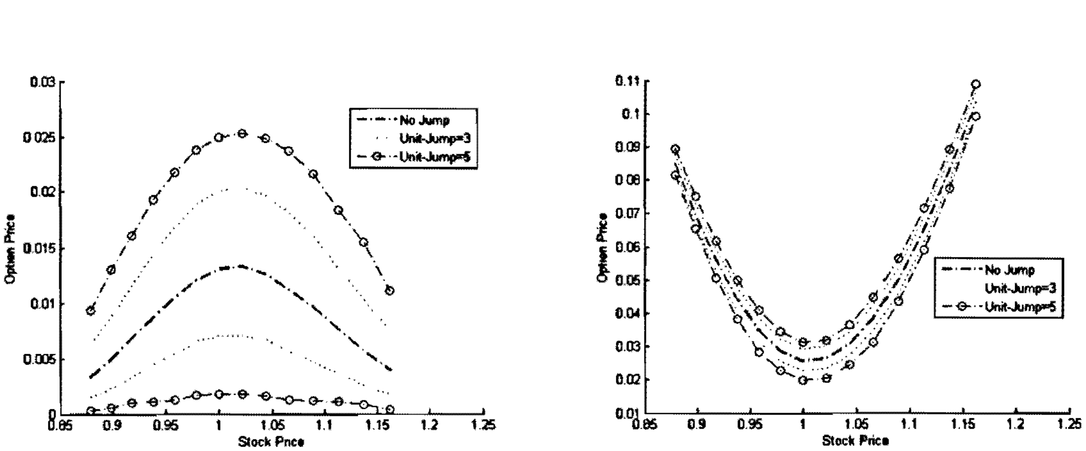


Figure 5.5: Left-hand side graph shows the arbitrage bounds for the basic model with a call butterfly payoff (equation (5.1)) with a lower strike $K_1 = 0.958$ and higher strike $K_2 = 1.066$. Right-hand side plot shows the arbitrage bounds for a long strangle (equation (5.2)) with a long position in strike $K_1 = 0.958$ and one long put at strike $K_2 = 1.066$.

$\frac{d}{\delta}$	1	3	5	7	9
S_0	$[\underline{V},\overline{V}]$	$[\underline{V},\overline{V}]$	$[\underline{V},\overline{V}]$	$[\underline{V},\overline{V}]$	$[\underline{V},\overline{V}]$
0.879	0.0848	[0.0829,0.0870]	[0.0815,0.0889]	[0.0805,0.0961]	[0.0462,0.0998]
0.958	0.0348	[0.0320,0.0378]	[0.0295,0.0400]	[0.0273,0.0416]	[0.0264,0.0424]
1	0.0257	[0.0233,0.1677]	[0.0209,0.0307]	[0.0196,0.0314]	[0.0192,0.0315]
1.044	0.0306	[0.0278,0.0336]	[0.0254,0.0365]	[0.0157,0.0370]	[0.0227,0.0373]
1.066	0.0383	[0.0354,0.0414]	[0.0327,0.0441]	[0.0304,0.0452]	[0.0293,0.0459]
1.162	0.1030	[0.1007,0.1056]	[0.0991,0.1285]	[0.0978,0.1092]	[0.0969,0.1114]

Table 5.2: This table shows no arbitrage interval for different jump units when the staring level S_0 increases in the case of a butterfly payoff (5.1)

$\frac{d}{\delta}$	1	3	5	7	9
S_0	$[\underline{V},\overline{V}]$	$[\underline{V},\overline{V}]$	$[\underline{V},\overline{V}]$	$[\underline{V},\overline{V}]$	$[\underline{V},\overline{V}]$
0.879	0.0032	[0.0015,0.0054]	[0,0.0076]	[0,0.0098]	[0,00.0115]
0.958	0.0107	[0.0059,0.0296]	[0.0024,0.0203]	[0,0.0235]	[0,0.0271]
1	0.0134	[0.0079,0.0189]	[0.0033,0.0235]	[0,0.0267]	[0,0.0280]
1.044	0.0128	[0.0072,0.0185]	[0.0030,0.0233]	[0,0.0262]	[0,0.0288]
1.066	0.0115	[0.0062,0.0173]	[0.0026,0.0221]	[0,0.0258]	[0,0.0297]
1.162	0.0038	[0.0017,0.0065]	[0.0006,0.0149]	[0,0.0119]	[0,0.0139]

Table 5.3: This table shows no arbitrage interval for different jump units when the staring level S_0 increases in the case of of a strangle payoff (5.2).

5.1.3 Arbitrage Bounds: Transaction Costs Model

In the transaction costs framework, we study the effect of a fixed transaction cost rate and a proportional transaction costs on the arbitrage bounds and compare the results with the ones without transaction costs.

Part I: Fixed Transaction Costs

In the simple case, fixed transaction costs, we test the effect of this transaction cost in the arbitrage bounds of the option price. As we expected (see equation (3.4) compared with (2.27)), the minimum upper bound in the case of fixed transaction costs is much higher than the minimum upper bound in the J-N framework without transaction costs.

In Figure 5.6, we implement the bounds for a fixed unit $\frac{d}{\delta} = 3$ and with fixed transaction costs $\mathcal{K} = 0.0001, 0.001$ (as we mentioned in Remark 12, we are just working with some artificial paths, so the fixed transaction costs $\mathcal{K}$ are not realist values) with different starting level S_0 and $N_2 = 70$. So we allow to have a maximum of 70 hedging times, as a result one could pay a maximum $N_2 * \mathcal{K} = 0.07$ more in the case of $\mathcal{K} = 0.001$.

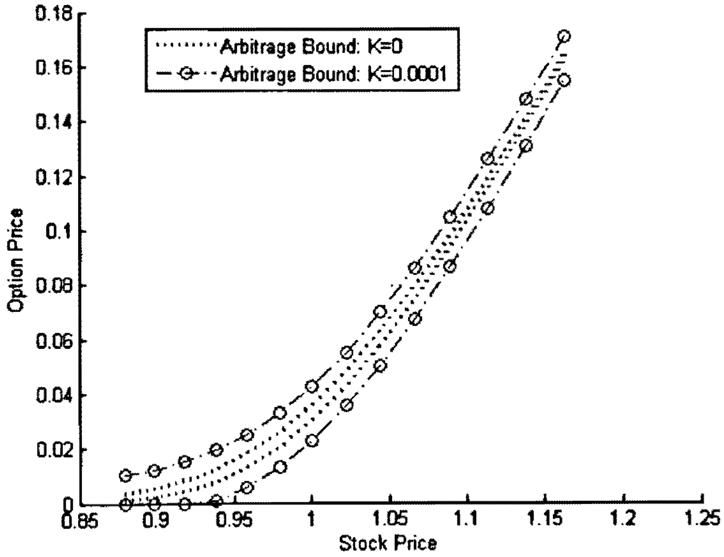


Figure 5.6: Upper and lower bounds for the fixed transaction costs $\mathcal{K}$: the minimum upper bound for the option value is increasing for a fixed S_0 and fixed unit-jump=3 when $\mathcal{K}$ takes the values 0 and 0.0001.

With respect to the arbitrage bounds in the basic model, the lower bound decreases while the upper bounds increases due to the effect of the fix transaction costs. Also, as we increase the rate $\mathcal{K}$, the region between the lower and upper curves becomes large (Figure 5.7).

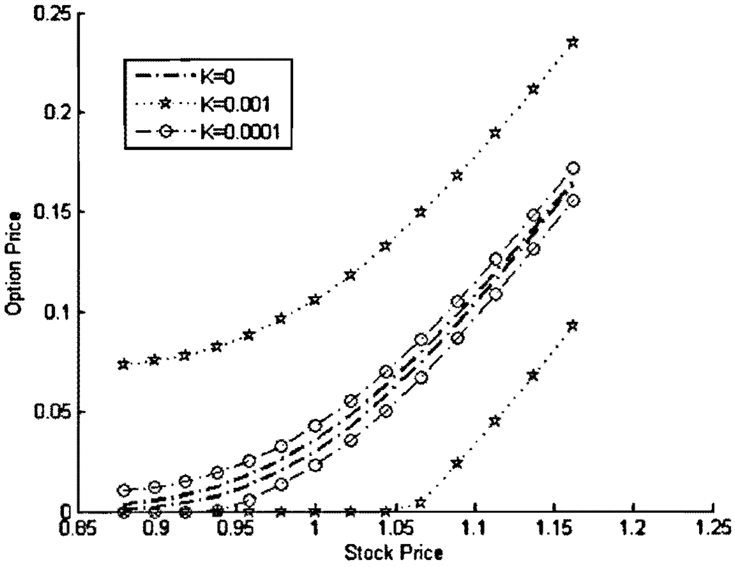


Figure 5.7: Upper and lower bounds become wider by increasing the rate $\mathcal{K}$ with the fixed jump units $\frac{d}{\delta} = 3$.

Note that in our implementation, we use a single call option to find the arbitrage interval, therefore; the bounds that obtained are very small. Then, we could not set more realistic fixed transaction costs for the model. Just assume that fixed transaction costs $\mathcal{K} = \$10$ per transaction, so at the end of the day, the investor may have to pay a maximum of \$700 for the option that has the payoff $\max(S_T - 1, 0)$. In order to have a realistic framework, assume one can buy a package of 1000 call options, then the bounds for the fixed transaction $\mathcal{K} = \$10$ can be obtained as in Figure 5.8.

Remark 14. *If one buys a contract (package) which includes c of call options so the payoff of this contract at maturity T will be c $V(S_T)$, so the minimum upper bound is*

$$\begin{aligned}
\overline{V}(S_0, v_0, \mathcal{K}_R) &= \inf_{H_c}\left\{\sup_{\pi\in P(S_0,v_0,d)}\left\{c\ V(S_T) - \sum_{i=0}^{N-1} H_c(S_i,v_i)(S_{i+1}-S_i) + N\mathcal{K}_R\right\}\right\} \\
&= c\inf_{H_c}\left\{\sup_{\pi\in P(S_0,v_0,d)}\left\{V(S_T) - \sum_{i=0}^{N-1}\frac{1}{c}H_c(S_i,v_i)(S_{i+1}-S_i) + N\frac{\mathcal{K}_R}{c}\right\}\right\} \\
&= c\inf_{H}\{\sup_{\pi\in P(S_0,v_0,d)}\{V(S_T) - \sum_{i=0}^{N-1} H(S_i,v_i)(S_{i+1}-S_i) + N\frac{\mathcal{K}_R}{c}\}\} \\
&= c\ \overline{V}(S_0, v_0, \frac{1}{c}\mathcal{K}_R) = c\overline{V}(S_0, v_0, \mathcal{K})
\end{aligned}$$

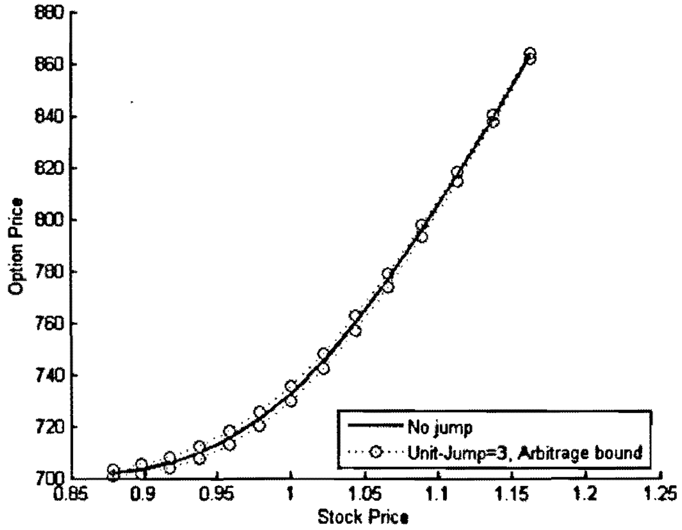


Figure 5.8: This plot shows the option bounds when the fixed transaction cost is \$10 by assuming 1000 call options.

where $\mathcal{K}_R$ is the value of a realistic fixed transaction cost (per stock transaction) and $\mathcal{K} \equiv \frac{\mathcal{K}_R}{c}$ represents the fixed cost per stock transaction used in the thesis. This computation justifies our use of small values of $\mathcal{K}$ when performing numerical experiments corresponding to bounds for a single call option. As a example let assume that $\mathcal{K}_R = 10\$$ per transaction for a package with $c = 100$, this can be transfered to $\mathcal{K} = 0.1\$$ in the case of one call option.

Part II: Proportional Transaction Costs

Let us start the analysis of this part with a question. What if we had followed the basic framework with the same assumptions, but paid the proportional transaction costs? Of course, one would expect the higher minimum upper bound as a results of the transaction costs according to equation (3.16).

A. Convegence of the Model:

First of all, we verify numerically our results by examining the behavior of the minimum upper bound as a function of the quadratic variation in the basic model and then with the proportional transaction costs. The results indicate that the minimum upper bound for the option price increases as the quadratic varication growths in both cases, with and without transaction (Figure 5.9).

Then, we test the convergence of the proportional transaction cost model as the number of trades increases. Table 5.4 illustrates the relation between the maximum number of steps N_2 and the minimum upper bound $\overline{V}(S_0, v_0, \alpha)$ (see equation (3.16)), where α is the round-trip and $0 \leq \alpha < 1$, for different

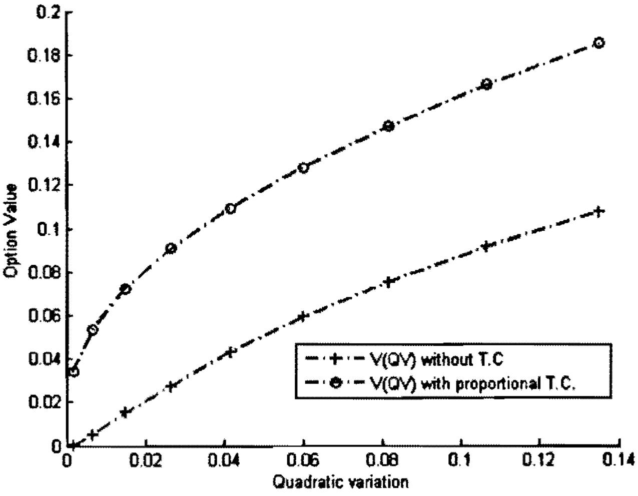


Figure 5.9: In this figure, we plot $\overline{V}$ as a function of the quadratic variation in both cases, without transaction costs and with proportional transaction costs with $\alpha = 0.01$ with $N_2 = 70$.

values $\alpha = 0, 2\%, 3\%, 4\%$ with $S_0 = 1$ and $v_0 = 0.0067$.

As it can be observed from Table 5.4, the values of the minimum upper bound $\overline{V}(1, 0.0067, \alpha)$ where $\alpha = 3\%, 4\%$ are increasing very fast at some value of N_2. For example, when the round trip $\alpha = 0.04$ (see the third column in the table), after $N_2 = 4$ (compare it with the case of no transaction costs $\overline{V}(1, 0.0067)$, which is located in the last column of the table), the option values are extremely larger. These points become less surprising when our result is compared against the analysis of Leland [4].

From Table 5.4 and Figure 5.9, we notice that the steps N_2, consequently the quadratic variation has a big impact on the minimum upper bound prices for a European option and specially when the proportional transaction costs exist in the J-N algorithm. We strongly believe that there should be a relationship between the quadratic variation more precisely $\delta = \sqrt{\frac{v_0}{N_2}}$ (see the second column) and the round-trip α. Although we can not prove this claim theoretically, our numerical result shows apparently $\frac{\alpha}{\delta}$ should be less than 1[5] in the J-N algorithm when the proportional transaction costs exit.

B. Arbitrage Bounds:

Additionally, we also implement arbitrage bounds for the proportional transactions cost. Recall that we

[4] Under the proportional transaction costs with hedging portfolios rebalanced only at discrete time instances, with a fixed finite time interval h between them. Leland type approaches are limited by the condition $\sqrt{\frac{2}{\pi}}\frac{\alpha}{\sigma\sqrt{h}} < 1$ the expression on the left known as the Leland number, where α is the round-trip transaction cost rate (expressed as a percentage of the stock price) and σ is the volatility of the underlying asset

[5] See the second column of Table 5.4 and compare the dramatic value $\overline{V}(1, 0.0067, \alpha)$ where $\alpha = 4\%, 3\%, 2\%, 0$.

N_2	$\delta = \sqrt{\frac{v_0}{N_2}}$	$\overline{V}(1,0.0067,0.04)$	$\overline{V}(1,0.0067,0.03)$	$\overline{V}(1,0.0067,0.02)$	$\overline{V}(1,0.0067)$
1	0.0816	0.0625	0.0569	0.0514	0.0408
2	0.0577	0.0709	0.0601	0.0495	0.0289
3	0.0471	0.0774	0.0666	0.0560	0.0353
4	0.0408	0.0793	0.0681	0.0569	0.0408
5	0.0365	0.2003	0.0738	0.0616	0.0365
6	0.0333	0.2214	0.0749	0.0618	0.0389
7	0.0309	0.2411	0.0763	0.0634	0.0360
8	0.0289	0.2598	0.2598	0.0637	0.0377
9	0.0272	0.2776	0.2776	0.0649	0.0408
10	0.0258	0.2946	0.2946	0.0653	0.0387
11	0.0246	0.3110	0.3110	0.0666	0.0400
12	0.0236	0.3269	0.3269	0.0675	0.0383
13	0.0226	0.3423	0.3423	0.0682	0.0393
14	0.0218	0.3573	0.3573	0.0687	0.0382
15	0.0211	0.3719	0.3719	0.0693	0.0389

Table 5.4: Relation between δ and round-trip α: This table shows for fixed quadratic variation $v_0 = 0.0067$, how δ and $\overline{V}(S_0, v_0, \alpha)$ (see equation (3.16)) can change. Also this table shows that if N_2 increases, as a result δ decreases, and we have to reduce round-trip α in order to be satisfied in inequality $\alpha < \delta$.

are not able to check the bounds for any round-trip α as we wish. We set our program for $N_2 = 70$, $S_0 = 1$ and $K = 1$ and the annual volatility is $\sigma = 0.2$, so

$$\delta = \sqrt{\frac{v_0}{N_2}} \simeq 0.009.$$

it means, we have to run the program for $\alpha < 0.009$ [6].

Figure 5.10 displays the minimum upper and the maximum lower pricing bounds for an option in the proportional transaction costs model for the fix jump size $\frac{d}{\delta} = 3$, with different round trips $\alpha = 0, 0.001, 0.004$. Clearly by increasing the level of α, the minimum upper bound price increases and the shape has less curvature by changing of the amount of round-trip $\alpha = 0$, 0.1%, 0.4%, (Figure 5.10). Note that for the round-trip, for example 0.1%, the upper bound is very close to the case of no transaction costs ($\alpha = 0$) while the lower bound is not as tight.

5.2 Market Data and No-Arbitrage Bounds

From the point of view of the J-N algorithm, it is clear that a key issue is to forecast the future quadratic variation v_0 over the life of the option. To reach this goal, we attempt to calculate the historical quadratic variation from the market data. To this end, we use daily closing prices of Yahoo! Inc.[7] for the period May 2009 - May 2011(closing prices are shown in Figure 5.11).

[6] Admittedly this α is too small for practical matters but as we mention in Remark 12, we use this number for the

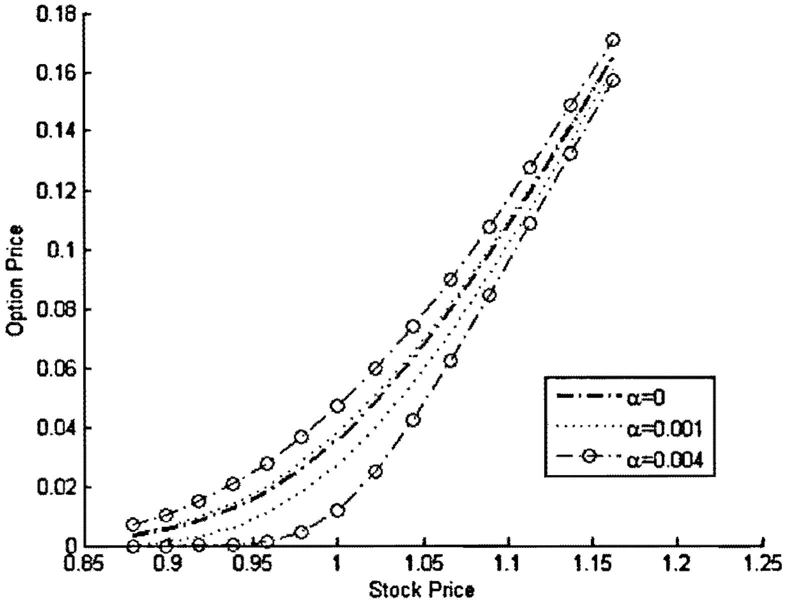


Figure 5.10: Upper and lower bounds for the proportional transaction costs: the plot shows that by reducing α, the bounds become smaller when the jump units is fixed.

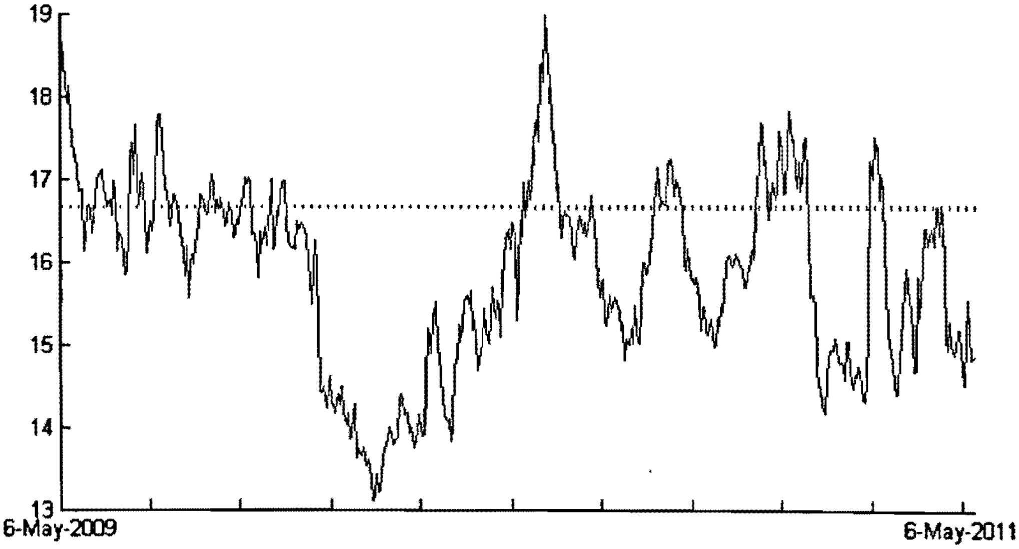


Figure 5.11: Time Series: The sequence of closing price between May 2009-May 2011 for the chosen stock is shown in this figure

5.2.1 Forecasting the Quadratic Variation

We set the initial stock price on April 14, 2011. At that moment, Yahoo! Inc. closing price was at \$16.69. In order to find the (sampled) quadratic variations from historical data, we followed the following process.

Let $[T_0, T_1]$ be closing daily business days where $T_0 =$ May 6, 2009 and $T_1 =$ May 6, 2011. We consider $\Delta t = 43$ which are (almost) two-months of business days. Then we look for $t_s \in [T_0, T_1 - \Delta t]$ such that $S(t_s) = S(0) = 16.69$ (See Figure 5.11). For any t_s, the quadratic variation Qv_s can be found by applying

$$Qv_s = \sum_{i=0}^{42} (\ln(S_{i+1}) - \ln(S_i))^2, \tag{5.3}$$

where $S_i \in [S(t_s), S(t_s + \Delta t)]$.

By applying (5.3) for all $t_s \in [T_0, T_1 - \Delta t]$, we found $Qv = \{Qv_1, ..., Qv_{16}\}$ (Figure 5.12).

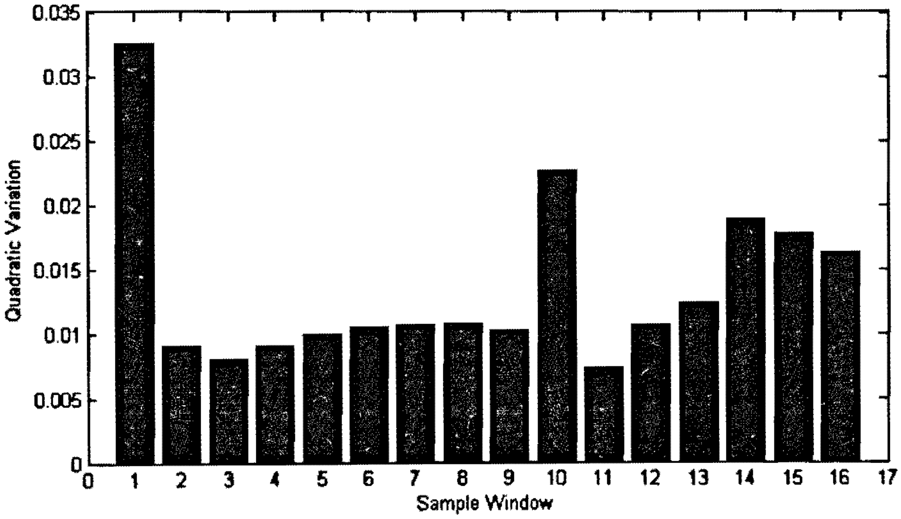


Figure 5.12: This figure shows the quadratic variation for different windows 1, .., 16 which start at times t_s such that $S_0 = S(t_s) \approx 16.69$ and all window's sizes are $N_2 = 43$

Let us assume that the future quadratic variation is neither higher than the maximum value of the set of the quadratic variation Qv_s nor less that the minimum of the set Qv_s. So we find the minimum and the maximum of the set Qv

$$Qv_{max} = \max_s Qv_s = 0.0326 \qquad Qv_{min} = \min_s Qv_s = 0.0073 \tag{5.4}$$

illustration propose only.

[7] This Data can be obtained from $http://ca.finance.yahoo.com/q?s = YHOO\&ql = 0$

where $s = 1, \ldots 16$.

A. Bounds with No Jumps vs. Option Price of the Market:

We are interested in finding a no arbitrage interval which denoted now as $[\underline{V}(Qv_{min}), \overline{V}(Qv_{max})]$. Let assume the future quadratic variation $Qv_f \in [0.0073, 0.0326]$, then we consider call options that expire within two months with strike prices ranging from \$13 – \$28. Table illustrates 5.5 the ask price of the option with different strike prices.

Strike price	13	14	15	16	17	18	19	20	21	22
Option price (Ask price)	5.7	4.75	3.75	2.81	1.97	1.28	0.77	0.45	0.26	0.15

Table 5.5: This table shows the option price with different strike prices.

First of all, we find the minimum upper bound [8] of the option values with no jumps and then we compare the results with the actual option prices (ask price is used for the minimum upper bound and bid price is used for the maximum lower bound) for each available strike price.

Note that for each strike price, the initial value is $S(0) = 16.69$ and $N_2 = 43$; therefore,

$$\delta = \sqrt{\frac{0.0326}{43}} \simeq 0.027.$$

We notice that $\delta_{empirical} = \min_{0 \leq i \leq 43} |\ln(S_{i+1}) - \ln(S_i)| = 0$, which can be an issue when one wants to use this model in practice.

By an inspection of the minimum upper bound in Figure 5.13, it seems that there is an arbitrage opportunity if the future quadratic varication be at most 0.0326 because $\overline{V}(Qv_{max}) < V_{ask}$.

Remark 15. *Recall that our model is an incomplete model, so if the option value $V_0 \in [\overline{V}, \underline{V}]$, then there is no arbitrage. But in this example V_{ask}, which is the option price, is out of the interval, so we have an arbitrage opportunity.*

B. Bounds with Jumps vs. Market's Option Prices:

Will the above arbitrage disappear by including jumps? We test the algorithm with max jump-unit 6 ($\sqrt{N_2} \simeq 6$). Figure 5.14 illustrates the effect of jumps to remove the arbitrage. It seems that by increasing the jump size, the arbitrage for the strike price \$20-\$22 can be made to disappear. But still there is an arbitrage opportunity when the strike price is less than \$20.

[8] In this thesis, we consider to find the minimum upper bound. For the maximum lower bound, the analysis is done similarly by using minimum quadratic variation 0.0073 and comparing the result with the bid price.

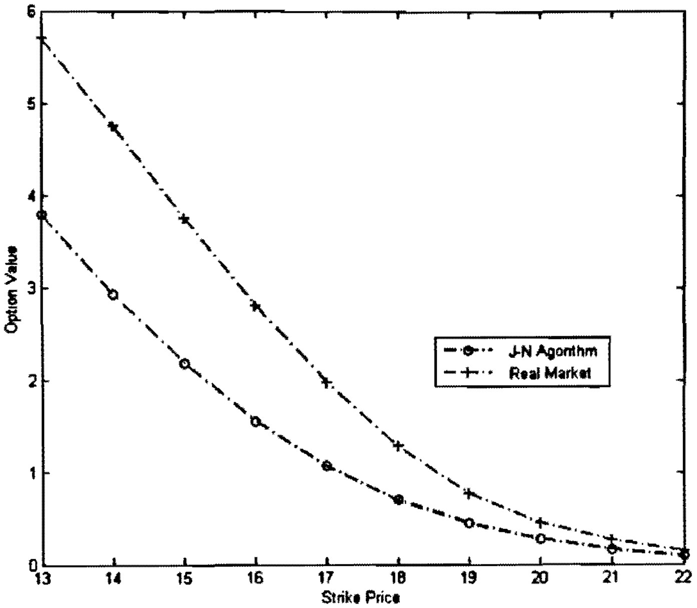


Figure 5.13: This figure shows the ask option price and the minimum upper bound with the maximum quadratic variation 0.0326, in the case of no jump occurs

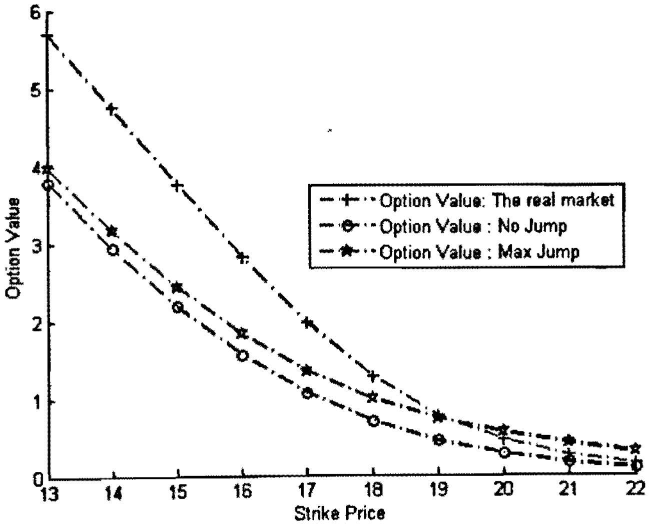


Figure 5.14: The minimum upper bound with no jumps and maximum jump units equal to 6 compared with the real option value show in this figure.

C. Bounds with Fixed Transaction Costs vs. Market's Option Prices:

For the next step, we wish to consider whether transaction costs can reduce/remove the chance of having an arbitrage opportunity.

To reach this goal, first we add fixed transaction costs, without jumps, to our model in order to make it more realistic. We expected that any arbitrage profits should disappear. In the case of fixed transaction costs, we found out by adding at least $\mathcal{K} = 0.045$ as a cost of trading [9](according to Remark 14, if we buy have a contract with $c = 222$, we will have $\mathcal{K}_R = 10\$$ which is more realistic) to the previous set up, i.e $S_0 = \$16.69$ and $N_2 = 43$ with $v_0 = 0.0326$, the arbitrage is vanished for all strike price.

Note that the minimum upper bound of the option becomes very expensive by enlarging of the strike price form \$13 – \$22 (Figure 5.15).

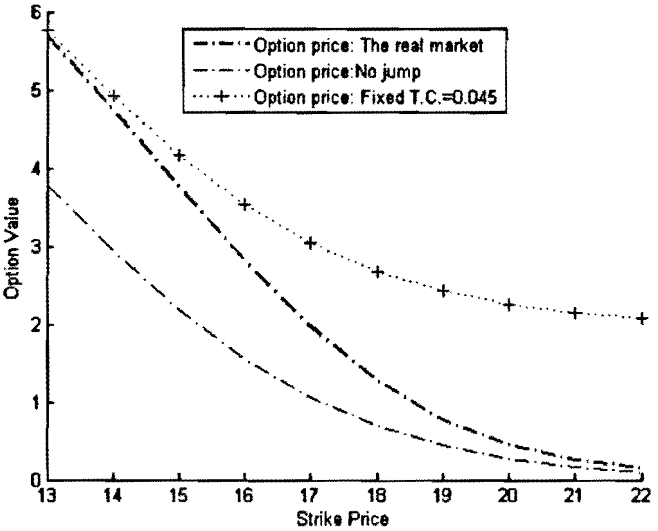


Figure 5.15: This plot shows the upper bound with the minimum fixed transaction $\mathcal{K} = 0.045$ which can remove the arbitrage in this market for all strike price.

D. Bounds with Proportional Transaction costs vs. Market's Option Prices:

Finally, we check if we can remove the arbitrage by adding the proportional transaction costs. Recall that, $\delta = 0.027$; Therefore, the maximum value of the round-trip α that we can run our program with $\alpha = 0.026 < \delta$.

[9] As we mention before, in this thesis, we work with a single option. To have more realistic fixed transaction costs, we need to consider trading a large number of options (as is the case in actual markets)

We plot the minimum upper bound with $S_0 = \$16.69$ for a maximum round-trip $\alpha = 0.026$, first with no jumps and then for a maximum jump units 6. So, as it can be observed in Figure 5.16, in the case of proportional transaction costs, we are not able to remove the profit for $K = \$13 - \$17$, because of restriction that we have on the round-trip (but it is already realistically high).

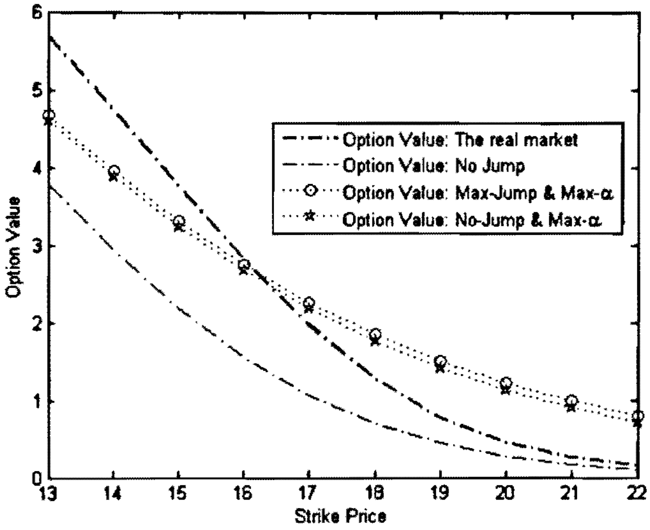


Figure 5.16: In the case of proportional transaction costs, we are not able to remove the gap, because of limitation on $\alpha < 0.027$.

5.2.2 Investment Strategy

In the previous section, we found an example that the call is overpriced, so there is an arbitrage opportunely in this case. In this section, we implement the actual hedging by using the J-N algorithm to see whether we are really superhedging according to proposition 1 or not.

As we know a most important parameter needs as input in the J-N algorithm is v_0. To reconstruct the hedging approximation, first we use $v_0 = Qv_{max}$ and then the actual quadratic variation of a window Qv_w, this will allow us to see what is the effect of using a larger quadratic variation which is Qv_{max} than the actual realized quadratic variation Qv_w.

To reach this goal, we run the J-N algorithm when $Qv_{max} = 0.0326$ and use the results such as price and hedging amounts at each node to actually hedge and then we replace Qv_{max} by Qv_w to compare the results.

Let us use $S(t_k)$ to denote the market data (daily closing prices for Yahoo! Inc.) in one of the windows where $k = 0, 1, 2, ..., R$ ($R = 43$ is length of data window) clearly $|\ln S(t_k) - \ln S(t_{k-1})|$ will take a variety of values which are not necessary equal δ, where δ is the minimum log return from the algorithm.

In order to implement the hedging investment, let $S_{i_k} \approx S(t_k)$ where S_{i_k} denotes the stock discretization value on the grid of the J-N algorithm closer to $S(t_k)$, therefore;

$$S_{i_k} = S_0 e^{i_k \delta} \approx S(t_k). \tag{5.5}$$

On the other side, $v_{j_k} = (N_2 - j_k)\delta^2 \approx v(t_k)$ where

$$v(t_k) \equiv [QV_{max} - \sum_{p=0}^{k} (\ln S(t_p) - \ln S(t_{p-1}))^2]. \tag{5.6}$$

So now we have a sequence (S_{i_k}, v_{j_k}) in the grid. By knowing $S_{i_k} = S_0 e^{i_k \delta}$ and $v_{j_k} = (N_2 - j_k)\delta^2$, we will need to decide how much to invest in the stock. Thus we modify the J-N program and create a matrix $H[i, j]$ such that $H[i, j] = h_{i,j}$ and choose the necessary number of shares by using indexes i_k and j_k as follows:

$$i_k = \frac{1}{\delta} ln[S(t_k)/S_0], \tag{5.7}$$

And

$$j_k = N_2 - \frac{v(t_k)}{\delta^2}. \tag{5.8}$$

Now, a problem is that the node (i_k, j_k) may not have been visited by the J-N algorithm i.e. there is no h_k value available at this node. Then one needs to look for the closest node that actually has h_k available. A problem we have found dealing with the current data is that the J-N grid does not capture well the small changes, we solved this problem by using the following process:

- It seems the problem should disappear if N_2 is large enough.
- The problem can be fixed if we can stay in node (S_i, v_j), namely doing nothing.

The hedging difference should then be:

$$-V(S_{t_R}) + \overline{V}(QV_{max}, d) + \sum_{i=0}^{N} h_i(S(t_{i+1}) - S(t_i))$$

The final balance (which will be an actual profit if the above hedging difference is nonnegative and there is an arbitrage, i.e. [Price of option $- \overline{V}(QV_{max}, d)] > 0$) will actually be:

$$\text{Price of option} - \overline{V}(QV_{max}, d) - V(S_{t_R}) + \overline{V}(QV_{max}, d) + \sum_{i=0}^{N} h_i(S(t_{i+1}) - S(t_i))$$

where N is the number trading and h_k is the amount of shares that the J-N algorithm gives at node

(S_{i_k}, v_{j_k}) and note that $S_{k_N} \approx S_{t_R}$.

In following we implement the hedging strategy with the different approaches.

I Investment Strategy with Qv_{max} when $N_2 = 43$.

First of all, we find the closest path, to the actual empirical path, among all passible paths that the J-N algorithm creates when $N_2 = 43$ with $v_0 = Qv_{max}$ in the case that no transaction costs occur. We do this by staying in some nodes whenever the movement of stock price is very small, namely the log return is not greater than δ which is $\delta = \min_{0 \leq i \leq N} |\ln(S_{i+1}) - \ln(S_i)|$.

Figure 5.17 displays the price path of Yahoo! Inc. for one of the windows and the simulated price path from the J-N algorithm, as it can be seen, because some of the movements of the market prices are small, we have to stay in a node and wait for another day in several instances. As a result, the chosen price path from the J-N algorithm is not very close to the market movements.

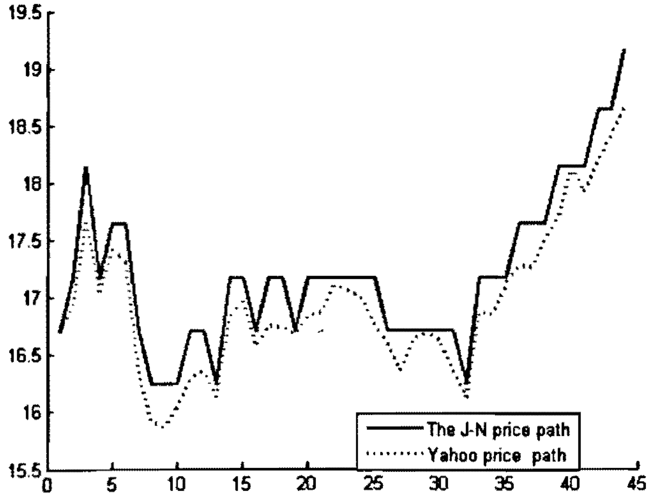


Figure 5.17: The figure shows the Yahoo! Inc. price path and the closest J-N price path for one of the windows but using the future quadratic variation QV_{max} when $N_2 = 43$.

II Investment Strategy with Qv_w when $N_2 = 43$.

As a second step, we implemented the hedging strategy and obtained the closer price path from the J-N algorithm by using the same number N_2 but now we took $v_0 = Qv_w$. Figure 5.18 shows the Yahoo! Inc. price path and the J-N price path closer to the data for the same window but using the exact quadratic variation, namely Qv_w. Apparently, the artificial price path form the J-N algorithm is now much closer to the actual price path for this window, this can be explained given that $v_0 = Qv_w < QV_{max}$ implies that we decrease δ so we will have a better approximation to S_k.

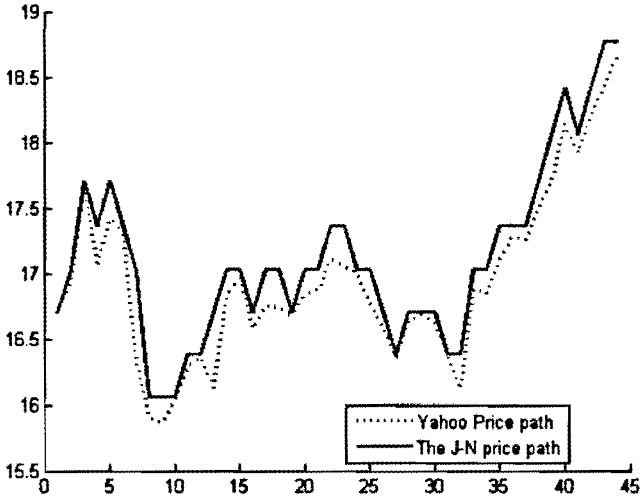


Figure 5.18: The figure shows the Yahoo! Inc. price path and the closest J-N price path for one of the windows with the actual quadratic variation of the window QV_w when $N_2 = 43$.

The next solution to the problem that we have tried increases the value of N_2 in the J-N algorithm, this provides a better approximation by decreasing δ, we run the program when $N_2 = 80$ and followed the same process as before.

III Investment Strategy with Qv_{max} when $N_2 = 80$.

In this case, we use Qv_{max} with larger $N_2 = 80$, clearly we decrease δ, so we have a better approximation of S_{t_k} and there is no need to stay in the same node many times. Figure 5.19 shows the result in this case.

IV Investment Strategy with Qv_w when $N_2 = 80$.

Finally, if the exact quadratic variation of the window is used for the implementation, the result is very close to the actual path for this window(Figure 5.20). So it seems that if we want to have a very good approximation of a given price path we need to have δ as close as possible as to the minimum return of the window price path.

Now we would like to investigate the profits/losses described above. In particular we are interested in

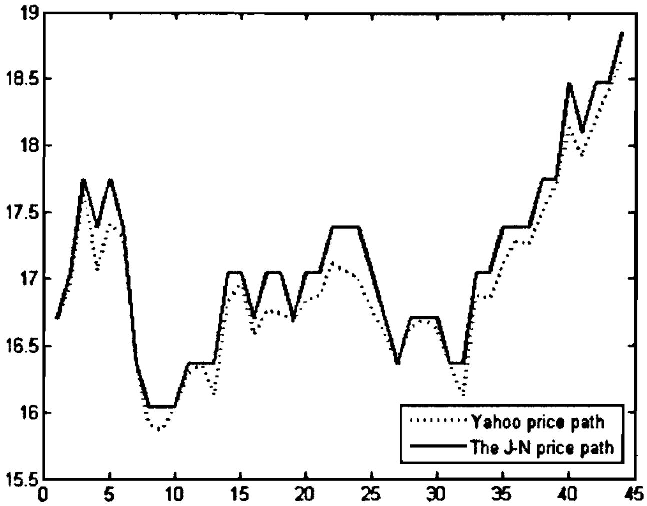


Figure 5.19: The figure shows the Yahoo! Inc. price path and the closest J-N price path for one of the windows with the future quadratic variation QV_{max} when $N_2 = 80$.

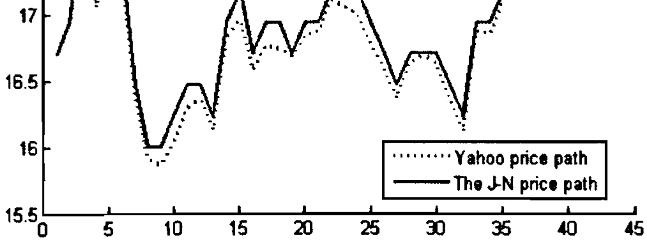


Figure 5.20: The figure shows the Yahoo! Inc. price path and the closest J-N price path for one of the windows with the actual quadratic variation of the window QV_w when $N_2 = 80$.

observing if whether we have superhedged or not, especially in the case when we use $v_0 = QV_{max}$ as the superhedge is only guaranteed theoretically for the case $v_0 = QV_w$. Moreover, we are interested in seeing the effect of transaction costs in the profit/loss as well. Table 5.6 illustrates the profits/losses for all cases.

Models	N_2	N	Qv	delta	profits/losses
Model with no Transactio Costs	43	29	0.0163	0.0195	2.2666
Model with no Transactio Costs	43	21	0.0326	0.0202	1.9229
Fixed Transaction cost $\mathcal{K} = 0.045$	43	21	0.0326	0.0202	1.0230
Fixed Transaction cost $\mathcal{K} = 0.097$	43	21	0.0326	0.0202	-0.018
Proportional Transaction Costs with $\alpha = 0.02$	43	21	0.0326	0.0202	1.0271
Model with no Transactio Costs	80	34	0.0163	0.0143	2.7220
Model with no Transactio Costs	80	28	0.0326	0.0202	1.8888
Fixed Transaction cost $\mathcal{K} = 0.045$	80	28	0.0326	0.0202	0.6839
Fixed Transaction cost $\mathcal{K} = 0.071$	80	28	0.0326	0.0202	-0.0181
Proportional Transaction Costs with $\alpha = 0.02$	80	28	0.0326	0.0202	-0.0692

Table 5.6: This table shows the profits/losses of Yahoo! inc. price path and also required number of trading when $v_0 = Qv_{max}$ and $v_0 = Qv_w$. Note that N is the actual number of tradings, N_2 is the number that we run the J-N algorithm with and Qv is the quadratic variation that could be the future quadratic variation or the actual quadratic variation.

As it can be seen in Table 5.6, we have profits almost in all cases except in the case of a large fixed transaction cost. Also when we hedge along the path with its actual quadratic variation, the profit seems that is higher than when we use the Qv_{max} because we use the smaller δ so we trade more. As we mentioned in figure 5.20, the J-N price path is very close to the actual price path and this result can be see in Table 5.6 as well, we trade around 34 times and we have higher profits there.

5.2.3 Smile Quadratic Variation

For our last empirical work, it is interesting to see that the minimum upper bound on the option value that we derived has a smile feature in the quadratic variation.

We have seen that for the future quadratic variation of $Qv_{max} = 0.0326$, the option price that we can obtain from the J-N algorithm is lower than the actual market price if no transaction costs are used in the J-N algorithm. So, one can ask, what is the real sampled quadratic variation to be used as input to the J-N algorithm in order to have the actual market's price?

To test the results, we set up the program with $S_0 = 16.69$ and $N_2 = 43$. Then for each $K = 13, 14, ..., 22$, we change the quadratic variation until we find the (*implied*) quadratic variation that gives us the option market's value [10].

[10] We use the fact that the minimum upper bound as a function of the quadratic variation is a monotonic function.

After this, we attempt to find the implied quadratic variation's market in the presence of fixed and proportional transaction costs to see how transaction costs can affect the implied quadratic variation.

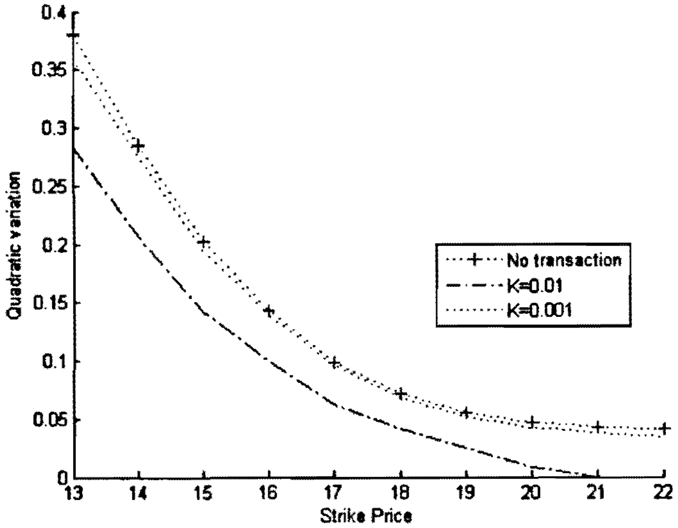


Figure 5.21: This figure displays the behavior of the quadratic variation as the strike price increases with fixed transaction costs. We assume the option price is the option market and we find the quadratic variation such that gives us the option price market.

Figures 5.21 and 5.22 display the behavior of the implied quadratic variation as a function of the strike price. The smile (i.e. non flat) feature of the graphs displayed does not disappear with the inclusion of fixed or proportional transaction costs.

So, as a result of the above findings, it seems that the basic model (Chapter 2) should be extended to consider the case of v_0 to be path dependent, i.e. the quadratic variation should not be constant across the modeling paths. This extension is known to capture the smile phenomena in probabilistic models.

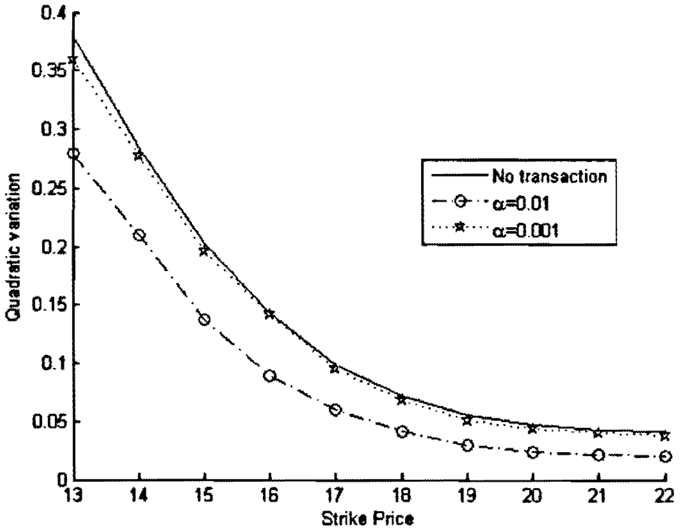


Figure 5.22: In this figure, we show if the proportional transaction costs exist, the actual quadratic variation should be less than in the case of no transaction costs.

Conclusions

The Black-Scholes assumptions on the stock price process is an important limitation of the Black-Scholes model, because in practice trading is discrete and the price paths are discontinuous and include price jumps and portfolio re-arrangements are subject to transaction costs.

Britten-Jones and Neuberger [Britten and Neuberger, 1996] show how one can relax the assumptions of continuity and infinite trading possibilities by adding some restrictions to the possible price paths in order to have a more realist framework. The most important assumption in their approach is that the model requires a known (sampled log) quadratic variation of price changes over a finite time interval and also they assume price jumps not to be greater that some specified jump size. By making use of those assumptions, they provide an interesting financial description of no arbitrage bounds of an option price by using a non probabilistic approach.

This thesis develops further the Britten-Jones and Neuberger framework, by introducing two kinds of transactions costs, fixed and proportional transaction costs, and analyzing the impact that adding these transaction costs have on the pricing interval (no arbitrage bounds).

The extension has some important practical implications; in particular, allowed us to conduct a preliminary analysis in order to assess the empirical relevance of the extended framework in real market data. The preliminary conclusions drawn from our studies are that the realistic features of the framework, in the sense that it requires few assumptions and observable inputs, make it a useful instrument that can be used to look for miss-pricing opportunities in option's prices. In particular, we provide some preliminary evidence that arbitrage opportunities, once present, may remain after transactions costs are taken into account.

Appendix A

Some Basic Concepts in Finance

In this appendix, we briefly introduce some fundamental concepts in finance.

A derivative (also referred to as contingent claims) is a general term for any financial contract whose value depends on underlying asset. *An option* is a derivative security that gives the right to buy or sell the underlying asset. Options can be *European style*, which can only be exercised at the expiry date, or *American style*, where exercise is at the discretion of the holder, at any time before or at the expire date.

The underlying variable can be a traded asset, such as a stock; an index portfolio; a futures price; a currency; or some measurable state variable, such as the temperature at some location or the volatility of an index. The payoff can involve various patterns of cash flows. Payments can be spread evenly through time, occur at specific dates, or a combination of the two. Our work on derivative pricing will focus on two specific types of contracts, a *forward contract* and *call option*.

Definition 9. *A call option is a contract that gives the holder the right, but not the obligation, to purchase the underlying for a pre-specified price K, called the strike price at the expiry date T with the exercise payoff* $\max(S-K,0)$ *where S denotes the price of the underlying asset.*

Definition 10. *A forward contract is an agreement between two parties to exchange an amount of the underlying commodity, currency or financial asset for cash at some specific date (the expiry date) in the future. The amount of cash to be paid (called the forward price) at expiry is determined when the contract is formed and is set so that there is no cost to either party to enter into the agreement.*

The Forward Price can be determined by the following formula:

$$F_0 = S_0 e^{rT}$$

where r is interest rate.

In general, forward price is $F_{t,T} = S_t e^{r(T-t)}$ for $0 \leq t \leq T$. Because in this thesis, we assume that the interest rate is zero i.e $F_{t,T} = S_t$, therefore instead of forward price, we work with the stock price.

Appendix B

Binomial Tree Model

Binomial models were first suggested by [Cox and Rubinstein, 1979], CRR, and then became widely used because of its intuition and easy implementation. Binomial trees are constructed on a discrete-time lattice. With the time between two trading events shrinking to zero, the evolution of the price converges weakly to a lognormal diffusion. Within this mode the European options value converges to the value given by the Black-Scholes formula.

STEP 1: Create the binomial price tree
Suppose that we have an underlying instrument (a stock price) at time $t = 0$ with initial value S_0. At any node on time layer t_i, the asset price S, can branch up to the value $S_{up} = S.u$ at time t_{i+1}, or down to the value $S_{down} = S.d$, where, by definition, $u \geq 1$ and $0 < d \leq 1$ are called up and down factor respectively. The probability of an "up move" is p and that of a "down move" is 1-p. Over one time period, Δt, the up and down factors are calculated using the underlying volatility, σ and the time duration of a step, Δt, measured in years. From the condition that the variance of the log of the price is $\sigma^2 \Delta t$, we have:

$$u = e^{\sigma\sqrt{\Delta t}}, \qquad d = e^{-\sigma\sqrt{\Delta t}} = \frac{1}{u}.$$

The CRR method ensures that the tree is recombinant, i.e. if the underlying asset moves up and then down (u,d), the price will be the same as if it had moved down and then up (d,u) here the two paths merge or recombine. This property reduces the number of tree nodes, and thus accelerates the computation of the option price. This property also allows that the value of the underlying asset at each node can be calculated directly via formula, and does not require that the tree be built first.

STEP 2: Computed the asset prices at all the nodes
The asset price tree is built by setting the value of the asset on the $(i+1)^{th}$ layer (that is at time t_i, since we start at $t_0 = 0$):

$$S(j,i) = S_j^i \qquad j = 0, \ldots, N,$$

where j corresponds to the $(j+1)^{th}$ node up the layer (i.e. j increases upwards) and,

$$S_i^j \equiv u^j d^{i-j} S_0.$$

STEP 3: Find Option value at each final node (Terminal)
At each final node of the tree i.e. at expiration of the option T, the option value is simply its intrinsic, or exercise, value.

- A call option: $\max(S_T - K, 0)$,
- A put option: $\max(K - S_T, 0)$,

where K is the Strike price and S_T is the price of the underlying asset.

STEP 4: Find the option value at earlier nodes
Once the above step is complete, the option value is then found for each node, starting at matutity time, and working back to the first node of the tree where the calculated result is the value of the option. The binomial value is found at each node, using the risk neutrality assumption, is:

$$V_{t-\Delta t,i} = e^{-r\Delta t}(pV_{t,i+1} + (1-p)V_{t,i-1}) \tag{B.1}$$

Where $V_{t,i}$ is the option's value for t^{th} node at time t and r is the risk free rate corresponding to the life of the option with probability p is:

$$p = \frac{e^{r\Delta t} - d}{u - d}.$$

Implementing the Tree:

We start with computing the current price V_0 of a European call; following algorithm gives the procedure for given values of the asset price S, the strike K, the volatility σ, the riskfree rate r and time to maturity T.

`Algorithem European Call Option:`

1. Input Strik K , S_0, r, σ, T and Time Steps M.
2. Compute $\Delta t = \frac{T}{M}$ where n is steps on tree .
3. Compute $ud = 1$ where $u = \sigma\sqrt{\Delta t}$.
4. Compute $p = \frac{e^{r\Delta t} - d}{u-d}$.
5. Evaluate $S^i - j$.
6. Calculate Price for Calls at T.

7. Binomial Value = [$p\times$ Option up + $(1-p)\times$ Option down] $\exp(-r\times\Delta t)$,

The above is the original Cox, Ross, Rubinstein (CRR) method; there are other techniques for generating the lattice, such as "the equal probabilities" tree. The Trinomial tree is a similar model, allowing for an up, down or stable path. The CRR method ensures that the tree is recombinant, i.e. if the underlying asset moves up and then down (u,d), the price will be the same as if it had moved down and then up (d,u) here the two paths merge or recombine. This property reduces the number of tree nodes, and thus accelerates the computation of the option price. This property also allows that the value of the underlying asset at each node can be calculated directly via formula, and does not require that the tree be built first.

Appendix C

Dynamic Programming

C.1 What Is Dynamic Programming?

The optimization of problems over time arises in many settings, from the control of heating systems to managing entire economies and finances. In between are examples including landing aircraft, managing blood inventories, selling assets, investing money in portfolios, or just playing a game of tic-tac-toe or backgammon. These problems involve making decisions, then observing information, after which we make more decisions, and then more information, and so on.

Dynamic programming is an optimization procedure that is particularly applicable to problems requiring a sequence of interrelated decisions. Each decision transforms the current situation into a new situation. A sequence of decisions, which in turn yields a sequence of situations. The value of a sequence of decisions is generally equal to the sum of the values of the individual decisions and situations in the sequence. The main concept of dynamic programmingis that choosing the best state in each step leads to the optimum final state.

The field of Dynamic programming is young, the original developments of the material are still accessible and relevant. Richard Bellman first introduced the title of dynamic programming to the study of these methods in his 1957 monograph [Bellman, 1957].

C.2 Analytical Concepts in DP

In this section, we briefly present the fundamental concept of *Dynamic Programming*[Bertsekas, 1976], [King, 2002]. For our approach, we fucus on a finite horizon discrete time dynamic programming model.

Suppose we wish to optimize a separable cost function of the form

$$V = \sum_{i=0}^{N-1} U_i(s_i, c_i) + V_T(s_T). \tag{C.1}$$

Note that the initial and terminal condition on the function are necessarily.

Even though we can use the Lagrange multiplier method for solving simple dynamic optimization problems, we are simply looking for a more tractable, and more powerful way. All dynamic programs can be written in terms of a recursion that relates the value of being in a particular state at one point in time to the value of the states that we are carried into at the next point in time is known as Bellmans equation, or the Hamilton-Jacobi equation, or increasingly, the Hamilton-Jacobi-Bellman equation (usually, in continuous-time optimization problems. Some textbooks refer to them as the functional equation of dynamic programming. We use the term Bellman Equation in our work, which is so widely used in the dynamic programming community.

C.2.1 Utility or payoff Function

To understand the Bellman equation, several concepts must be understood and it will be useful to review the elements of a dynamic program. First, any optimization problem has some objective such as minimizing travel time, minimizing cost, maximizing profits, maximizing utility, et cetera. The mathematical function that describes this objective is called the *objective function.* It means the agent has a utility or payoff (Cost) function $U(s_1, c_1, ..., s_T, c_T)$ that depends on the realized states and decisions from period $t = 0$ to the horizon T_1. U has the form

$$U(s_1, c_1, ..., s_T, c_T) = \sum_{i=0}^{T_1} U_i(s_i, c_i). \tag{C.2}$$

where $U_i(s_i, c_i)$ is the agent's period i utility (payoff) function.

C.2.2 Control and State Variable

There are two key variables in any dynamic programming problem: a state variable s_t, and a decision variable c_t (the decision is often called a control variable in the engineering literature). These variables can be vectors in $\mathbb{R}^n$, but in some cases they might be infinite dimensional objects.

Definition 11. ***The state variable:*** *this captures all the information of the current situation which is needed to make a correct decision, as well as the information that we need to describe how the system evolves over time.*

For example, to decide how much to consume and spend at each point in time, people would need to know (among other things) their initial wealth. Therefore, wealth would be one of their state variables, but there would probably be others.

Definition 12. ***The decision (control) variable:*** *Decisions represent how we control the process and chosen at any given point in time.*

For example, given their current wealth, people might decide how much to consume now. Choosing the control variables now may be equivalent to choosing the next state; more generally, the next state is affected by other factors in addition to the current control. Therfore, in the simplest case, today's wealth (the state) and consumption (the control) might exactly determine tomorrow's wealth (the new state), though typically other factors will affect tomorrow's wealth too.

C.2.3 policy

The dynamic programming approach describes the optimal plan by finding a rule that tells what the controls should be, given any possible value of the state. For example, if consumption (c) depends only on wealth (W), we would seek a rule $c(W)$ that gives consumption as a function of wealth. Such a rule, determining the controls as a function of the states, is called apolicy function

Definition 13. *Assume that the state s_i is the state variables at the end of time interval i (an element of a space Φ_i), the control variable c_i is the control variables applied during time interval i (is an element of the a space $\mathbb{R}^m$). A policy function or control law is a sequence of N functions $(\mu_0, ..., \mu_{N-1})$:*

$$\mu_i : \Phi_i \to \mathbb{R}^m, \tag{C.3}$$

where μ_i maps states s_i into control $c_i = \mu_i(s_i)$

C.2.4 Basic Model

A dynamic optimization problem (pay off function) in discrete time is given by the form

$$opt \sum_{i=0}^{N-1} U_i(s_i, c_i) + V_N(s_N). \tag{C.4}$$

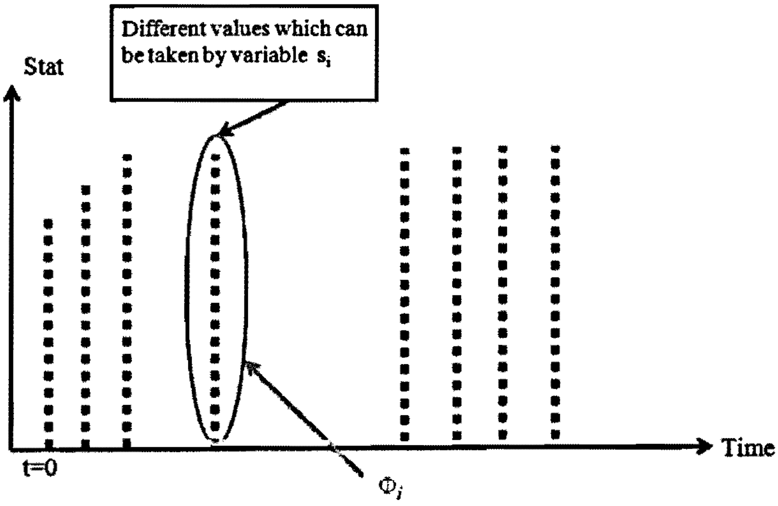
subject to:

$$s_{i+1} = G_i(s_i, c_i) \tag{C.5}$$

with the initial condition s_0 given and the terminal value is given.

G_i is a function that describes the system and in particular the mechanism by which the state is updated. And express the evolution of the system of the state under the influence of the control variable.

We now formulate a general problem of decision over a finite number of stage. This basic problem is very general. We start by considering problems that have the following representation:



Consider a discrete-time dynamic system modeled by the state equation

$$s_{i+1} = G_i(s_i, c_i) \qquad i = 0, 1, ..., N-1, \tag{C.6}$$

where the state s_i is the state variables at the end of time interval, the control variable $c_i \in \mathbb{R}^m$ is the vector of the control variables applied during time interval i.

The initial state s_0 is given; the final state s_T may be constrained or not. No uncertainty is considered here: given the current value of the state variable s_i, after selecting the control variable c_i we know exactly what the future state will be, according to the time-varying dynamics described by the G_i functions.

Given an initial state s_0 and a fixed admissible policy function $\mu = (\mu_0, ..., \mu_{N-1})$, we start with observations which are already exist and dynamic system C.6 becomes

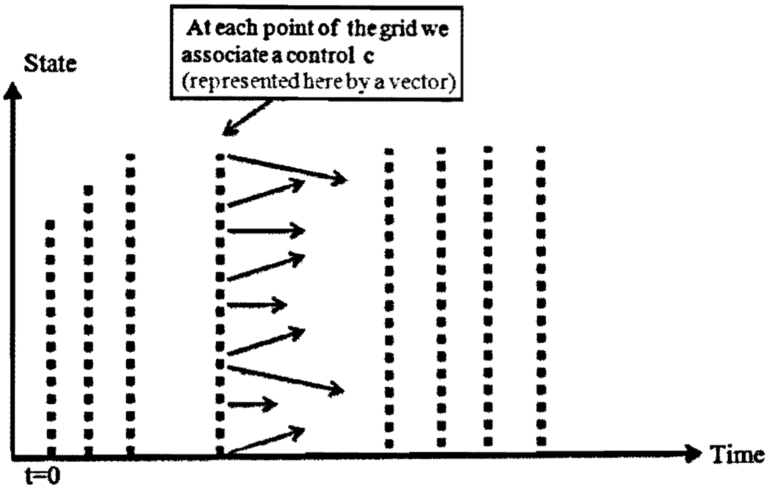
$$s_{i+1} = G_i(s_i, \mu_i(s_i)), \qquad i = 0, ..., N-1 \tag{C.7}$$

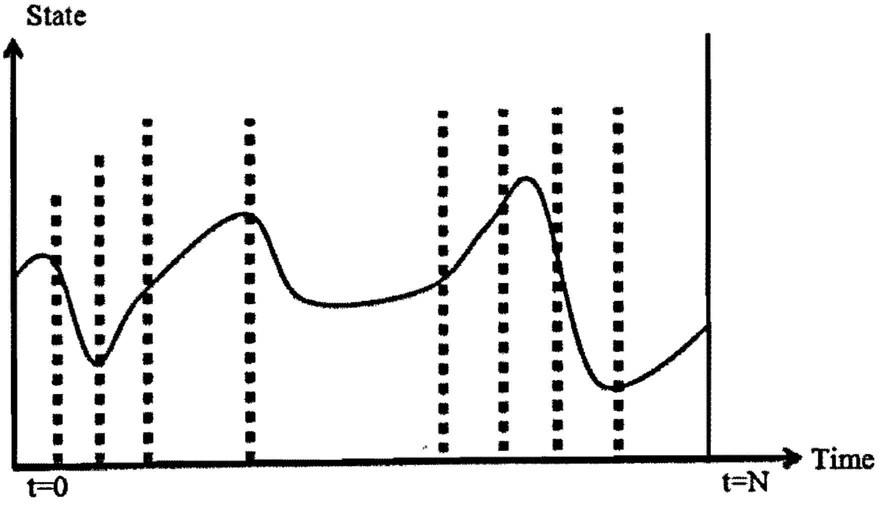
Thus, for given function U_i, $i = 0, ..., N-1$,

$$V_\mu(s_0) = \sum_{i=0}^{N-1} U_i(s_i, \mu_i(s_i)) + V_N(s_N) \tag{C.8}$$

Optimal policy μ^* is one that optimize the function C.8,

$$V_{\mu^*}(s_0) = opt_{\mu \in \mathbb{R}^m} \sum_{i=0}^{N-1} U_i(s_i, \mu(s_i)) + V_N(s_N) \tag{C.9}$$


At each point of the grid we associate a control c (represented here by a vector)
State
Time
t=0



where $\mathbb{R}$ is the set of all admissible policies.

The Principle of Optimality

Richard Bellman showed that a dynamic optimization problem in discrete time can be stated in a recursive, step-by-step form by writing down the relationship between the value function in one period and the value function in the next period. The relationship between these two value functions is called the Bellman equation. In general there are following method to solve this equation:

1. The method of undetermined coefficients, also known as 'guess and verify', can be used to solve some infinite-horizon.

2. The Bellman equation can be solved by backwards induction.

3. By calculating the first-order conditions associated with the Bellman equation [1].

We use the second method, which is backward induction, to solve the Bellman equation.

The key idea underlying why backward induction produces an optimal decision rule is called The Principle of Optimality. Roughly, the principle of optimality states the following:

Claim 1. *Let $\mu^* = \{\mu_0^*, ..., \mu_{N-1}^*\}$ be an optimal policy for basic problem and assume that using μ^*, a given state s_i occurs at time i. Consider the subproblem whereby we are at s_i at time i and wish to minimize the cost-to-go from time i to time N Then the truncated policy $\{\mu_i^*, ..., \mu_{N-1}^*\}$ is optimal for this subproblem.*

Proof: [Bertsekas, 1976]

So an optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.

[1] It is possible to obtain a system of difference equations or differential equations called the 'Euler equations'. Standard techniques for the solution of difference or differential equations can then be used to calculate the dynamics of the state variables and the control variables of the optimization problem.

Appendix D

The Optimal Hedging

D.1 Basic Model

As we mention in Chapter 2, the Bellman Equation can be written as

$$\overline{V}_i(S_i, v_i) = \inf_{\{h_i\}} \left\{ \sup_{\{c_i\}} \{ \overline{V}(S_{i+1}, v_{i+1}) - h_i(S_{i+1} - S_i) \} \right\}. \tag{D.1}$$

Subject to the conditions

- $(\ln S_{i+1} - \ln S_i)^2 = c_i^2 \leq d^2$.

- $v_{i+1} = v_i - c_i^2 \geq 0$.

Note that, the Bellman equation (3.6) is a non standard dynamic programming problem due to being a minimax optimization problem. However, the optimal hedging strategy can be easily found and with this out of the way the problem becomes a standard dynamic programming problem. The infimum over the trading strategies that makes the problem non-standard dynamic programming.

We are going to use some really neat tricks. Suppose that we have found optimal hedging, $\tilde{h}$, for the problem (2.27), therefore we have

$$\overline{V}(S_i, v_i) = \sup_{\{c_i\}} \{ \overline{V}(S_i e^{c_i}, v_i - c_i^2) - \tilde{h} S_i (e^{c_i} - 1) \}.$$

Since we have defined $\overline{V}(S_i, v_i)$ as an upper bound, one can rest assured that

$$\overline{V}(S_i, v_i) \geq \overline{V}(S_i e^{c_i}, v_i - c_i^2) - \tilde{h} S_i (e^{c_i} - 1).$$

The amount by which the upper bound $\overline{V}(S_i, v_i)$ is greater than the right-hand-side is a function, F, of the permissible value for c_i. We can therefore write as

$$F(c_i) \equiv V(S_i, v_i) - V(S_ie^{c_i}, v_i - c_i^2) + \tilde{h}S_i(e^{c_i} - 1) \geq 0. \tag{D.2}$$

Obviously from the equation (D.2), $F(c_i) \geq 0$ and by substitution $c_i = 0$ in definition of function F, we have $F(0) = 0$.

Remark 16. *If the function F is equal to zero at the origin, and greater or equal to zero for positive or negative values of c, as long as arbitrarily small in magnitude, then the derivative of F with respect to c_i in the neighborhood of $c_i = 0$ must be equal to 0.*

Above grantee that

$$\frac{dF}{dc} |_{c=0} = F'(0) = 0.$$

Now by applying the derivative in (D.2),

$$\begin{aligned}
\frac{\partial F}{\partial c_i} &= 0 - \frac{\partial \overline{V}}{\partial (S_ie^{c_i})} \cdot \frac{\partial (S_ie^{c_i})}{\partial c_i} + \frac{\partial \overline{V}}{\partial (v_i - c_i^2)} \frac{\partial (v_i - c_i^2)}{\partial c_i} + \tilde{h}S_ie^{c_i} \\
&= -\frac{\partial \overline{V}}{\partial S_ie^{c_i}} S_ie^{c_i} + 2c_i \frac{\partial \overline{V}}{\partial (v - c_i^2)} + \tilde{h}S_ie^{c_i}.
\end{aligned}$$

Evaluating this quantity at $c_i = 0$, and equating it to zero, gives

$$\frac{\partial F}{\partial c_i} |_{c_i=0} = -\frac{\partial \overline{V}}{\partial S_i} S_i + \tilde{h}S_i = 0. \tag{D.3}$$

Therefore follows that the optimal hedge ratio is given by

$$\frac{\partial V}{\partial S_i} = \tilde{h}, \qquad c_i \leq min(c_i, d^2). \tag{D.4}$$

Equation (D.4) determines the optimal strategy. Note that, if all the Black-Scholes conditions are met, the usual expression for the Black delta holding of shares is recovered. The optimal strategy, however, holds also in all the more general cases where the J-N method is applicable.

As a last step we can substitute the optimal strategy we have just determined in the expression for the minimum upper bound. When we do so we obtain:

$$V_i(S_i, v_i) = \inf_{\{h_i\}} \left\{ \sup_{\{c_i\}} \left\{ V(S_{i+1}, v_{i+1}) - \frac{\partial V}{\partial S_i}(S_{i+1} - S_i) \right\} \right\}. \tag{D.5}$$

Subject to the inequality constraint:

$$c_i \leq min(v_i, d^2).$$

and the terminal condition

$$V(S_i, 0) = V(S_i).$$

D.2 Proportional Transaction Costs

Let us assume that the optimal hedge, $\tilde{h}$, has somehow been found, so

$$\overline{V}(S_i, v_i, \alpha) = \sup_{\{c_i\}}\{\overline{V}(S_i e^{c_i}, v_i - c_i^2, \alpha) - \tilde{h}S_i(e^{c_i} - 1) + \alpha|\tilde{h} - g|S_i e^{c_i}\}. \tag{D.6}$$

where $g = H(S_{i+1}, v_{i+1})$ and is known.

Therefore we can define the difference between the left-hand side and the right-hand side as a function

$$F_\alpha(c_i) = \overline{V}(S_i, v_i, \alpha) - \overline{V}(S_i e^{c_i}, v_i - c_i^2, \alpha) + \tilde{h}S_i(e^{c_i} - 1) - \alpha|\tilde{h} - g|S_i e^{c_i} \geq 0. \tag{D.7}$$

By using induction, we can show $F_\alpha(0) = 0$.

Assume $i = N - 1$, so

$$F_\alpha(c_{N-1}) = \overline{V}(S_{N-1}, v_{N-1}, \alpha) - V(S_N) + \tilde{h}S_{N-1}(e^{c_{N-1}} - 1) - \alpha|\tilde{h}|S_{N-1}e^{c_{N-1}}. \tag{D.8}$$

According to our set up

$$\mathcal{B}_{N-1} + H(S_{N-1}, v_{N-1})S_N = V_N + \alpha|H(S_{N-1}, v_{N-1})|S_N.$$

where $\mathcal{B}_{N-1} + H(S_{N-1}, v_{N-1})S_N = \overline{V}(S_{N-1}, v_{N-1}, \alpha)$, So

$$\overline{V}(S_{N-1}, v_{N-1}, \alpha) = V(S_N) + \alpha|\tilde{h}|S_{N-1}e^{c_{N-1}}$$

So $F_\alpha(c_{N-1}) = 0$ when $c_{N-1} = 0$.

When $i = n$

$$F_\alpha(c_n) = \overline{V}(S_n, v_n, \alpha) - \overline{V}(S_{n+1}, v_{n+1}, \alpha) + \tilde{h}S_n(e^{c_n} - 1) - \alpha|\tilde{h} - g|S_n e^{c_n}. \tag{D.9}$$

where

$$\overline{V}(S_n, v_n, \alpha) = \overline{V}(S_{n+1}, v_{n+1}, \alpha) + \alpha|H(S_n, v_n) - H(S_{n+1}, v_{n+1})|S_{n+1}.$$

By doing the same process we can show $F_\alpha(0) = 0$, hence according to remark 16, $F'_\alpha(0) = 0$. Therefore follows that the optimal hedge ratio is given by

$$\frac{\partial F_\alpha}{\partial c_i} = -\frac{\partial \overline{V_\alpha}}{\partial S_i e^{c_i}} S_i e^{c_i} + 2c_i \frac{\partial \overline{V_\alpha}}{\partial (v - c_i^2)} + \tilde{h} S_i e^{c_i} - \alpha |\tilde{h} - g| S_i e^{c_i}.$$

Evaluating this quantity at $c_i = 0$, and equating it to zero, gives

$$\frac{\partial F_\alpha}{\partial c_i} |_{c_i=0} = -\frac{\partial \overline{V_\alpha}}{\partial S_i} S_i + \tilde{h} S_i - \alpha |\tilde{h} - g| S_i = 0. \tag{D.10}$$

Finally $\tilde{h}$ the optimal hedge ratio is given by

$$\frac{\partial \overline{V}_\alpha}{\partial S_i} = \tilde{h} - \alpha |\tilde{h} - g|, \quad \text{where} \quad c_i \leq min(c_i, d^2). \tag{D.11}$$

Hence, we look for the optimal trading strategies such that

$$\frac{\partial \overline{V}_\alpha}{\partial S_i} = \begin{cases} (1-\alpha)\tilde{h} + \alpha g, & \text{if } \tilde{h} \geq g, \\ (1+\alpha)\tilde{h} - \alpha g, & \text{if } \tilde{h} < g. \end{cases}$$

Therefore

- If $\tilde{h} \geq g$, then $\tilde{h} = \frac{1}{1-\alpha}(\frac{\partial \overline{V}_\alpha}{\partial S_i} - \alpha g)$.
- If $\tilde{h} < g$, then $\tilde{h} = \frac{1}{1+\alpha}(\frac{\partial \overline{V}_\alpha}{\partial S_i} + \alpha g)$.

Appendix E

MATLAB Code

Optimization without Transaction costs

```
%By this program, we find the minimum upper bound without transaction cost.

%We set the program for S_0=1 and K=1, annual volatility=0.2 for 2—month
%observation

%%% Inputs
   N=100;                             %number of setps in the tree
   volannual = 0.2;                   %annual volatility
   T = 2/12;                          %(number of month)/12
   remainvol = T*(volannual^2); %remainding volatility v_0
   delta = sqrt(remainvol/N);         %N*\delta = v_0
   ddelta = 3                         %d/delta
   Strike = 1;                        % Strike Price

   i = matrixI(N);                   % is a function
   C = length(i(:,1));               % length of i in maturity
   Value = (-100)*ones(N+1,N+1);
   Stock = (-100)*ones(N+1,N+1);
   hstar = zeros(N,N);               %matrix of number of shares

%%% Payoff at T
   time = 1;
   Stock0 = 1;
   Stock(:,time) = Stock0*exp(i(:,time)*delta);
    for  A = 1:C
```

```
          Value(A,time) = max(Stock(A,time)-Strike ,0);
     end

     for time = 2:N+1
        Stock(1:C-(time-1),time) = Stock0*exp(i(1:C-(time-1),time)*delta);
        n1 = fix(sqrt(min((ddelta)^2,time-1))); %condintion n<=min(ddelta ,N-j)
        for h=1:C-(time-1)
        [Value1 ,Stock1] = destpoint(n1,time ,h,Stock0 ,Value ,delta ,i); %function

     %%% optimization %%%
      [V,Vh] = opthedge(Value1 ,Stock1 ,Stock(h,time)); %function
      hstar(h,time) = Vh; %optimal hedge
      Value(h,time) = V;  %optimal value
      clear Value1 Stock1
        end
    end
   V10 = Value(1,time); %the minimum upper bound

function [i] = matrixI(N)
%To make matrix i which create the gide for index i, where N is steps in
%the tree
i = (2*N+1)*ones(N+1,N+1);
i(:,1) = (-N:2:N)';
    for c = 2:N+1
        a = 1;
        I(c) = (N-(c-1));
        for e = -I(c):2:I(c)
            i(a,c)= e;
            a = a+1;
        end

    end

function [Value1, Stock1] = destpoint(n1,time ,h,Stock0 ,Value ,delta ,i)
% With this function, we find the all destination points from a certain
% node. Recall that (i,j) goes to (i+n,j+n^2) where  n<= min(j,d^2/delta^2).
p = 1;
for n=-n1:n1
    if n~=0
        if  time-1>=n^2
```

```
            [tf, index] = ismember(i(h,time)+n,i(:,time-n^2));
            Value1(p) = Value(index,time-n^2);
            Stock1(p) = Stock0*exp((i(h,time)+n)*delta);
            p = p+1;
        end
    end
end
Value1;
Stock1;

function [V,Vh] = opthedge(Value1,Stock1,Stockhtime)
%With this function, we can find the optimal value (V*) and the optimal
%hedge (h*).
            f1 = [1; 0];
            b=cat(2,-Value1);
            col1 =cat(2,-ones(1,length(Value1)));
            col2 = cat(2,-(Stock1-Stockhtime));
            A=cat(2,col1',col2');

            lb = [-10^4;-10^4];
            ub =[10^4;10^4];
            [x,fval] = linprog(f1,A,b,[],[],lb,ub)
            Vh = x(2);
            V = x(1);
```

Optimization without Transaction costs

```
%In this code, we find the minimum upper bound with proportional
%transaction cost.

%We set the program for S_0=1 and K=1 annual volatility=0.2 for 2-month
%observation and alpha =0.001.

%%%% Inputs
    N=100;                           %number of setps in the tree
    volannual = 0.2;                 %annual volatility
    T = 2/12;                        %(number of month)/12
    remainvol = T*(volannual^2);     %remainding volatility v_0
    delta = sqrt(remainvol/N);       %N*\delta = v_0
    ddelta = 3                       %d/delta
    Strike = 1;                      % Strike Price
    i = matrixI(N);               % is a function
    C = length(i(:,1));           % length of i in maturity
    Value = (-100)*ones(N+1,N+1);
    Stock = (-100)*ones(N+1,N+1);
    hstar = zeros(N,N);           %matrix of number of shares
    xi = zeros(N+1,N+1);

    alpha =0.001                  %transaction cost rate
 %%% Payoff at T
    time = 1;
    Stock0 = 1;
    Stock(:,time) = Stock0*exp(i(:,time)*delta);
 for  A = 1:C
      Value(A,time) = max(Stock(A,time)-Strike,0);
 end
%%%Destination nodes
 for time = 2:N+1
    Stock(1:C-(time-1),time) = Stock0*exp(i(1:C-(time-1),time)*delta);
    n1 =fix(sqrt(min((ddelta)^2,time-1))); %condintion n<=min(ddelta,N-j)
   for h=1:C-(time-1)
    [Value1,Stock1,xi1]=destpointtrans(n1,time,h,Stock0,Value,xi,delta,i);
%%%%%Find optimal value and share
    [V,Vh] = opthedgetrans(Value1,Stock1,xi1,alpha,Stock(h,time)); %function
    hstar(h,time-1)=Vh;                            %optimal hedge
    Value(h,time)=V;                               %optimal value
    clear Value1 Stock1 xi1 V Vh
    xi(h,time) = hstar(h,time-1); % make it as a number of shares for future
   end
 end
   V10 = Value(1,time); %the minimum upper bound

 function[Value1,Stock1,xi1]=destpointtrans(n1,time,h,Stock0,Value,xi,delta,i)

%From given node(i,j),the children nodes are (i+n,j+n^2), also in this case we need
%numbers of shares for the children nodes as well.
p = 1;
for n=-n1:n1
    if n~=0
        if  time-1>=n^2
```

```
            [tf, index] = ismember(i(h,time)+n,i(:,time-n^2));
            Value1(p) = Value(index,time-n^2); %j+n^2
            Stock1(p) = Stock0*exp((i(h,time)+n)*delta); %i+n
            xi1(p) = xi(index,time-n^2); %number of shares at (i+n,j+n^2)
            p = p+1;
        end
    end
end

function [ V, Vh] = opthedgetrans(Value1,Stock1,xi1,alpha,Stockhtime)
%This function find the optimal value with transaction cost
   K1 = length(Value1);
   f1 = cat(2,1  ,0,zeros(1,K1));
   b=cat(2,-Value1,xi1,-xi1);
%make A and b,
   A1 = cat(2,-ones(K1,1),(-(Stock1-Stockhtime))',alpha.*diag(Stock1));
   A2 = cat(2,zeros(K1,1),ones(K1,1),-eye(K1));
   A3 = cat(2,zeros(K1,1),-ones(K1,1),-eye(K1));
   A  = cat(1,A1,A2,A3);

%upper and lower bounded
   lb = cat(2,-10^4,-10^4,zeros(1,K1));
   ub = cat(2,10^4,10^4,10^4*ones(1,K1));
   [x,fval] = linprog(f1,A,b,[],[],lb,ub);
   Vh = x(2); %optimal hediging
   V = fval;  %optimal value
```

Convex Hull Optimization-Without Transaction Costs

```
%This code represent the same optimization but with Convex Hull method
   %%%%% Inputs
    N=100;                            %number of setps in the tree
    volannual = 0.2;                  %annual volatility
    T = 2/12;                         %(number of month)/12
    remainvol = T*(volannual^2);      %remainding volatility v_0
    delta = sqrt(remainvol/N);        %N*\delta = v_0
    ddelta = 3                        %d/delta
    Strike = 1;                       % Strike Price

    i = matrixI(N);                  % is a function
    C = length(i(:,1));              % length of i in maturity
    Value = (-100)*ones(N+1,N+1);
    Stock = (-100)*ones(N+1,N+1);
    hstar = zeros(N,N);          %matrix of number of shares

    time = 1; Stock0 = 1;
    Stock(:,time) = Stock0*exp(i(:,time)*delta);
    %%%% Payoff at T
     for  A = 1:C
          Value(A,time) = max(Stock(A,time)-Strike,0);
     end

    for time = 2:N+1
      Stock(1:C-(time-1),time) = Stock0*exp(i(1:C-(time-1),time)*delta);
      n1 = fix(sqrt(min((ddelta)^2,time-1))); %condintion n<=min(ddelta,N-j)
      for h=1:C-(time-1)
      [Value1,Stock1] = destpoint(n1,time,h,Stock0,Value,delta,i); %function
%Convex Hull Method
       if find(Value1)~=0
       %If all points are in the same line
       X =[Stock1(:) Value1(:)];
       dt = DelaunayTri(X);
       scopedWarnOff = warning('off', 'MATLAB:TriRep:EmptyTri2DWarnId');
       restoreWarnOff = onCleanup(@()warning(scopedWarnOff));
       tri = dt.Triangulation;
       if isempty(tri)
          maxx1 = Stock1(1); maxx2 = Stock1(2);
          maxy1 = Value1(1); maxy2 = Value1(2);
       else
       %otherwise, find the hull line
          k = convhull(Stock1,Value1);
          maxa = find(k==max(k)); mina = find(k==min(k));
          k(1:maxa-1) = [];
          localx = Stock1(k(:)); localy = Value1(k(:));
          v = [localy(1)-localy(length(k)); -(localx(1)-localx(length(k)))];
          r = [localx(length(k))-localx; localy(length(k))-localy];
          d = [];
             for ii=1:length(k)
              d(ii) = abs(dot(v,r(:,ii)))/norm(v);
             end
```

```
        dsort = sort(d,'descend');
           if isempty(find(d))
              maxy2 = localy(1); maxy1 = localy(length(k));
              maxx2 = localx(1); maxx1 = localx(length(k));
           else
              [s1,index1] = ismember(dsort(1),d);
               maxx2 = localx(index1);
               maxy2 = localy(index1);
                 if length(find(dsort))==1
                    if Stock(h,time)<=maxx2
                      maxx1 = localx(length(k));
                      maxy1 = localy(length(k));
                    else
                      maxx1 = localx(length(1));
                      maxy1 = localy(length(1));
                    end
                 else
                    [s2,index2] = ismember(dsort(2),d);
                     maxx1 = localx(index2);
                     maxy1 = localy(index2);
                 end
            end
        end
        %V* and h*
           hstar(h,time) = (maxy1-maxy2)/(maxx1-maxx2); % find h star
           Value(h,time) = hstar(h,time)*(Stock(h,time)-maxx1)+maxy1;
          else
              Value(h,time) = 0;
          end
         clear  Value1 Stock1
     end
 end
V10=Value(1,time); %the minimum upper bound
```

Bibliography

[Avellaneda and Parás, 1995] Avellaneda, M. and Parás, A. (1995). Dynamic hedging with transaction costs: From lattice models to nonlinear volatility and free boundary problems. It is available at www.math.nyu.edu/faculty/avellane/gordo.ps.

[Bellman, 1957] Bellman, R. (1957). *Dynamic Programming*. Princeton, NJ: Princeton University Press,.

[Bensaid and Scheinkman, 1992] Bensaid, B., J. L. H. P. and Scheinkman, J. (1992). Derivative asset pricing with transaction costs. *Mathematical Finance*, 2(2):63–86.

[Bertsekas, 1976] Bertsekas, D. P. (1976). *Dynamic Programming and Stochastic Control*. New York, NY: Academic Press,.

[Black and Scholes, 1973] Black, F. and Scholes, M. (1973). The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):637–654.

[Boyle and Lin, 1997] Boyle, P. and Lin, X. (1997). Optimal portfolio selection with transaction costs. *North American Actuarial Journal*, 1:27–39.

[Britten and Neuberger, 1996] Britten, J. M. and Neuberger, A. (1996). Arbitrage pricing with incomplete markets. *Applied Mathematical Finance*, 3:347–363.

[Britten and Neuberger, 1998] Britten, J. M. and Neuberger, A. (1998). Welcome to the real world. *Risk, Black-Scholes-Merton Supplement,*, 11(9):11–13.

[Cox and Rubinstein, 1979] Cox, J., R. S. and Rubinstein, M. (1979). Option pricing: A simplified approach. *Journal of Financial Economics*, 7:229–263.

[DeMarzo and Mansour, 2005a] DeMarzo, P. M., K. I. and Mansour, Y. (2005a). On regret and options: A game theoretic approach for option pricing. It is available at levine.sscnet.ucla.edu/archive/refs4782713000000000012.pdf.

[DeMarzo and Mansour, 2005b] DeMarzo, P. M., K. I. and Mansour, Y. (2005b). Online trading algorithms and robust option pricing. In Proceedings of the thirty-eighth annual AMC Symposium on Theory of Computing, pages 477486, New York, NY, USA, 2006. ACM.

[Fedotov and Mikhailov, 2001] Fedotov, S. and Mikhailov, S. (2001). Option pricing for incomplete markets via stochastic optimization: Transaction costs, adaptive control, and forecast. *Int. J. of Theoretical and Applied Finance*, 4(1):179–195.

[Geletu, 2007] Geletu, A. (2007). Solving optimization problems using the matlab optimization toolbox. A Tutorial, available at www.faculty.jacobs-university.de/poswald/teaching/OptimMatlab.pdf.

[Hao, 2008] Hao, T. (2008). Option pricing and hedging bounds in incomplete markets. *Journal of Derivatives & Hedge Funds*, 14(2):78–89.

[Harris, 2003] Harris, L. (2003). *Trading and exchanges: market microstructure for practitioners the main book rebonato idea.* Oxford University Press US.

[Hull, 1997] Hull, J. (1997). *Option, Futures and Other Derivatives.* Prentic Hall.

[King, 2002] King, I. (2002). A simple introduction to dynamic programming in macroeconomic models. It is available at researchspace.auckland.ac.nz/bitstream/handle/2292/190/230.pdf.

[Leland, 1985] Leland, H. (1985). Option pricing and replication with transaction costs. *Journal of Finance*, 40:1283–1301.

[Merton, 1973] Merton, R. (1973). Theory of rational option pricing. *Bell Journal of Economics and Management Science*, 4(1):141–183.

[Osborne, 1959] Osborne, M. F. M. (1959). Brownian motion in the stock market. *Operations Research*, 2:145–173.

[Rebonato, 2004] Rebonato, R. (2004). *Volatility and Correlation: The Perfect Hedger and the Fox.* Wiley Finance.

[Roorda and Schumacher, 2005] Roorda, B., E. J. and Schumacher, J. (2005). Performance of hedging strategies in interval models. *North American Actuarial Journal*, 41:575–592.

[Sargent, 1987] Sargent, T. J. (1987). *Dynamic Macroeconomic Theory.* Harvard University Press.

[Stokey and Prescott, 1989] Stokey, N., L. R. and Prescott, E. (1989). *Recursive Methods in Economic Dynamics.* Cambridge, MA: Harvard University Press.

[Tsitsiklis, 1997] Tsitsiklis, N., B. D. (1997). *Introduction to Linear Optimization.* Athena Scientific.