AN OPTIMAL APPROXIMATION FOR THE PAYOFFS OF VARIANCE SWAPS IN STATIC REPLICATION

by

QIANG CHEN

JAMES WANG, COMMITTEE CHAIR
ZHIJIAN WU, CO-CHAIR
BRUCE TRACE
TAVAN TRENT
YANG XIAO

A DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Mathematics
in the Graduate School of
The University of Alabama

TUSCALOOSA, ALABAMA

2014

ABSTRACT

In this dissertation, we create a portfolio of simple vanilla put and call options as an optimal approximation of nonlinear payoffs by using static replication (1995, 1998) [**1, 2**] under certain measure which is called $E(a, b, N, f)$. More specifically, we focus on the static replication of variance swaps payoffs because of their popularity in current financial market [**3**].

The analysis is motivated by the following reasons. Due to the limited availability of strike prices with traded vanilla options, static replication is only an approximation [**1**]. Bradie and Jain (2008) [**4**] used Black-Scholes and Heston stochastic volatility model to find the optimal approximation. Liu (2010) [**5**] created three approximation methods.

In order to improve the approximation, we use a new measure for the static replication to construct the replicating portfolio with lower cost compared with the current methods.

DEDICATION

This dissertation is dedicated to everyone who helped me and guided me through the trials and tribulations of creating this manuscript. In particular, my family and close friends who stood by me throughout the time taken to complete this work.

ACKNOWLEDGMENTS

It has been a long way for me from a college student majoring in chemical engineering to a Ph.D student doing research in financial mathematics. I spent more than four years to find out my real interest and used almost eight years to make my dream come true. I got a lot of help from my family. In these years, my family always support me to move on whenever I face difficulities. I would like to gratefully and sincerely thank my father, my mother and my sister. Without their encouragment, it is impossible for me to achieve my goal.

I would like to thank Dr. Zhijian Wu for his guidance, support and concern. As an advisor, he can always find the best way to explain things to me. I learnt so much from him not only in mathematics but also in other aspects. I believe all of these will be benefical for me in my whole life.

I would also like to thank my other committee members, Dr. James Wang, Dr. Tavan Trent, Dr. Bruce Trace and Dr. Yang Xiao for their help of my dissertation and academic progress.

Finally, I would like to thank all my friends in Mathematics Department for their encouragement and help. All of them are always happy to help me whenever it is needed.

I am so appreciated for the suppports of my family, my teachers and my friends.

Contents

List of Tables

CHAPTER 1

INTRODUCTION

1.1. Background

Variance swaps [**6**], as financial derivatives, are widely traded in the financial market. People in the market like variance swaps because of their purity. It is the advantage of variance swaps, since they provide pure exposure to the volatility of underlying asset price. Another way to invest in volatility is to use options, but it requires constant delta hedging to remove the directional risk of the underlying security [**7**]. Variance swaps do not have this restriction because the profit and loss only depends directly on the gap between actual price volatility and expected volatility. On the other hand, many traders are interested in variance swaps because that expected volatility in the financial derivative market effects directly on the quoted strike price [**8**]. However, the final payoff will be determined by actual price variance. Based on the historical data, the actual price variance has been higher than the expected variance, a fact called as the variance risk premium, bringing a chance for traders to earn arbitrage profit [**9**]. For the same reason, people can hedge options on actual price variance by using variance swaps. Many authors believe that it is meaningful to do some research related to variance swaps.

One of the most popular research topics in recent years is to price variance swaps with given payoffs. In view of financial mathematics, the job can be done by using replication, which is a common financial tool derived from no-arbitrage principle. In financial markets, a replicating portfolio for a given asset or series of cash flows is a portfolio of assets with the same properties. This strategy has been developing for many years. In 1976, Stephen A. Ross introduced an idea that people can build up a contract with complex payoffs by constructing a portfolio of simple options [**10**]. Furthermore, he pointed out that arbitrary simple options are equivalent to a portfolio of call options and complex multiple options are equivalent to simple options written on a portfolio of original marketed assets [**11**].

In 1978, Douglas T. Breeden from Univeristy of Chicago and Robert H. Litzenberger from Stanford Univerisity indicated that the price of original securities would be derived from the total consumption of constructing a portfolio of call options. And the value of the original securites with certain payoffs are equivalent with the value of the constructed portfolio at many future times. They attempted to make the state preference model operational in a multiperiod economy by deriving the prices of original securities from the prices of European call options on aggregate-consumption expenditures at each date [**12**].

In 1995, Emanuel Derman, Deniz Ergener and Iraj Kani created a replicating method which is called static replication. This method uses a portfolio of plane vanilla options to value a target contract with a nonlinear payoff [**2**]. It emplys plane vanilla opions with different strike prices and fixed weights to construct the portfolio. By adapting static replicaton, a target contract with complicated payoff can be decomposed into a portfolio with some vanilla options. The fair value of the target contract is the cost of the replicating portfolio [**13**]. Once constructed, the portfolio will exactly replicate the value of the target contract without any further adjustment. This is one of the most important features of static replication compared with dynamic replication which requires continual adjustments since the behavor of portfolios are assumed to be similar at an isolate point. Static replication is developing very fast in recent years.

Except pricing a target contract, static replicaton is useful in two areas. First, we can create a static replication portfolio with the same payoff as an exotic option, perhaps at a reduced premium. We can even replicate target options of our own design that are not offered in the market. Second, static replication is especially suited to hedge exotic high-gamma options with the assumption of no-arbitrage [**2**].

Static replication is attractive for hedging an option position, because of the following reason. According to the Black-Scholes theory, a stock option, as a contract, at any instant behaves like a weighted portfolio of risky stock and riskless zero-coupon bonds [**2**]. Instead of owning an option, we can, in principle, own a portfolio of stock and riskless bonds, and achieve exactly the same returns. To do so, we must continuously adjust the weights in our portfolio according to the formula as time passes the stock price moves. However, continuous weight adjustment is impossible, and so traders adjust at discrete intervals. This causes small errors that compound over the life of option and results in replication whose accuracy increases with the frequency of hedging [**14**]. Moreover, there are transaction costs associated with adjusting the portfolio weights which grows with the frequency of adjustment and can overwhelm the profit margin of the opton. Traders have to compromise between the accuracy and cost. In contrast, static replication does not have this issue. After setup, a static hedge does not need to be adjusted dynamically over the life of a derivative, and should be the preferred hedging approach whenever it is possible [**15**].

1.2. Volatility Swaps and Variance Swaps

A reference for this section, for example, is John C. Hull's *Options, Futures, and Other Derivatives* [**16**].

1.2.1. Volatility Swaps. In finance, a volatility swap is a forward contract on the future realized volatility of a given underlying asset. Volatility swaps allow investors to trade the volatility of an asset directly, much as they would trade a price index.

Compared with an option, whose volatility exposure is contingent on its stock price, these swaps provide pure exposure to volatility. As a result, the payoffs of swaps depend directly on the volatility. It is true only for forward starting volatility swaps. However, once the swap has its asset fixed, its mark-to-market value also depends on the current asset price. Traders can use these instruments to speculate on future volatility levels, to trade the spread between realized and implied volatility, or to hedge the volatility exposure of other positions or businesses.

In particular, a stock volatility swap is an annualized volatility forward contract. At the expiration date, the payoff is expressed as,

$$N(\sigma_R - K_{vol}),$$

where

σ_R: realized stock volatility over the life of the contract,

K_{vol}: annualized volatility price at delivery,

N: notional amount of the swaps in dollars per annualized volatility point.

At expiration date, traders who have the volatility swaps earn N dollars for each point which σ_R has exceeded K_{vol}. They are swapping the fixed volatility K_{vol} for the actual future volatility σ_R.

1.2.2. Variance Swaps. A variance swap is a forward contract on annualized variance, the square of the realized volatility. At expiration date, its payoff is expressed as

$$N(\sigma_R^2 - K_{var}),$$

where

σ_R^2: realized stock variance over the life of the contract,

K_{var}: annualized variance price at delivery,

N: notional amount of the swap in dollars per annualized volatility point squared.

At expiration date, traders who have the variance swaps earn N dollars for each point which σ_R^2 has exceeded K_{var}. They are swapping the fixed variance K_{var} for the actual future variance σ_R^2.

1.3. Static Replication by Using Options

References for this section are John C. Hull's *Options, Futures, and Other Derivatives* [**16**], Christian Ekstrand's *Financial Derivatives Modeling* [**17**] and Marek Capinski and Tomasz Zastawniak's *Mathematics for finance: an introduction to financial engineering* [**18**].

1.3.1. Options. In finance, an option, also called vanilla option, is a contract which gives the buyer (the owner) the right, but not the obligation, to buy or sell an underlying asset or instrument at a specified strike price on or before a specified date. The seller incurs a corresponding obligation to fulfill the transaction that is to sell or buy if the owner elects to exercise the option prior to expiration. The buyer pays a premium to the seller for this right. An option which conveys to the owner the right to buy something at a specific price is referred to as a call; an option which conveys the right of the owner to sell something at a specific price is referred to as a put. The style of an option is usually defined by the dates on which the option may be exercised. The vast majority of options are either European or American options. These options, as well as others, where the payoff is calculated similarly which are referred to as vanilla options.

In FIGURE 1.1, the x-axis represents asset prices from 0 to ∞ and the y-axis represents cash amount. X is the strike price of options. The broken lines are gains of simple call and put options. And the solid lines are payoffs of these options. For a call option, it is clear that if the stock price S is less than the strike price X, then the gain will be a constant negative number, $C^E e^{rT}$, which is the present value of the money paid for the option. In this situation, the owner of call options will not

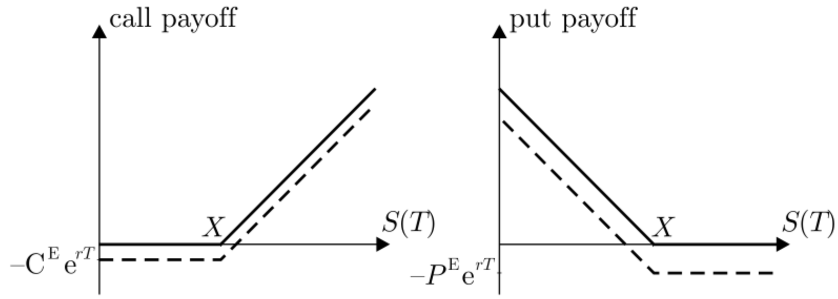

FIGURE 1.1. Payoffs(solid lines) and gains(break lines) for a buyer of European calls and puts

exercise the option and the payoff is 0. If the stock price S is greater or equal to the strike price X, the owner will exercise the option. As a result, the payoff goes up with the increase of the asset price. At one point, the payoff from the option will be equal to the present value of the initial cost, so the gain will be equal to zero. After that, the payoff and gain will both keep increasing with respect to the increase of stock price. The gap between payoff and gain is always the present value of initial cost. For a put option, since the owner has the right to sell the stock at strike price X, so, with the increase of the asset price, the payoff and gain decrease at same time. The gain will touch the bottom first since the payoff earned will be cancelled out by the present value of initial cost. If the asset price is greater than the strike price and the owner of the option will not exercise the option any more, then the gain is a constant negative number, $P^E e^{rT}$, which is the present value of the money paid for the option and the payoff is 0.

1.3.1.1. *European Options.* European options are a type of options. It has all properties of vanilla options with only one restriction. It can not be exercised before the maturity date T, which is a specified future time. The owner of a European call option has the right to exercise the option, with strike price X, or not. If the asset price at maturity is S_T, the European call option value is therefore

$$(S_T - X)_+ = \max(S_T - X, 0).$$

Similarly, a European put option provides the owner the option to sell the underlying asset with the strike price X at maturity date T or not. The option value at maturity is therefore

$$(X - S_T)_+ = \max(X - S_T, 0).$$

Suppose there is a digital European call option which pays \$1 if $S_T > X$ and 0 otherwise. Thus, the value of the digital European option at maturity is $\theta(S_T - X)$, where θ is the Heaviside function. Similarly, a digital European put option is worth $\theta(S_T - X)$, at T .

1.3.1.2. *ITM, OTM and ATM.* ITM is short for in the money, OTM is short for out of the money and ATM is short for at the money. We will use the following notation,

t: The time before the option is exercised,

S_t: The underlying asset price at time t,

X: The strike price of given option.

For a call option, it is ITM if $S_t > X$, OTM if $S_t < X$ and ATM if $S_t = X$. For a put option, it is ITM if $S_t < X$, OTM if $S_t > X$ and ATM if $S_t = X$.

1.3.2. No-Arbitrage Principle. In financial markets, no-arbitrage principle is the most important assumption. The principle declares that, without an initial investment, no one can make risk-free profits. It is also the foundation for the pricing of financial derivatives. Suppose V and U are two financial portfolios which means they are constructed by some weighted financial assets. Let $P(\cdot)$ be a probability measure. The portfolio V is arbitrage if the initial value $V_0 = 0$ and $P(V_T \geq 0) = 1$, where V_T

is the value of V at the maturity date T which is a future date. The no-arbitrage principle assumes that this will never happen. Let t be any time before T, then the application for no-arbitrage principle in derivative pricing is that

$$P(V_T = U_T) = 1 \Rightarrow P(V_t = U_t) = 1.$$

In particular, when t is todays date we obtain $V_0 = U_0$.

1.3.3. Static Replication of Nonlinear Payoffs. Suppose we have a portfolio V that pays $h(S_t)$ at time t, for all $t \leq T$, we want to determine the present value of V if, for arbitrary $h(S_t)$, it is a fixed function with a well-defined second derivative. If $h(S_t) = \alpha S_t - X$, we can use the present values of a portfolio consisting of underlying assets with price S_t at time t for all $t \leq T$, and the zero-coupon bond maturing at T to price the portfolio V. When $h(S_t)$ is nonlinear, the present values of European call options maturing at T provide sufficient information to determine the present value of the portfolio V. This statement is made clear by the following computation (Ekstrand, 2011) [page 10],

$$\begin{aligned}
h(S_t) &= \int_0^\infty h(X) d[\theta(S_t - X)] \\
&= \int_0^\infty \left(-\frac{d}{dX}(h(X)\theta(S_t - X)) + h'(X)\theta(S_t - X) \right) dX \\
&= h(0) + \int_0^\infty h'(X)\theta(S_t - X) dX \\
&= h(0) + \int_0^\infty \left(-\frac{d}{dX}(h'(X)(S_t - X)_+) + h''(X)(S_t - X)_+ \right) dX \\
&= h(0) + h'(0)S_t + \int_0^\infty h''(X)(S_t - X)_+ dX
\end{aligned}$$

Thus, based on the no-arbitrage principle, we may take the following actions on the financial market, at time $t = 0$, to get the present value of the portfolio V,

- Buy $h(0)$ number of bonds maturing at T,

- Buy $h'(0)$ number of underlying assets S,
- Buy $h''(X)dX$ number of options with strike X.

Then, the portfolio is worth $h(S_T)$ at time T. By the no-arbitrage principle, we have the following formula,

$$V = h(0)P_{0T} + h'(0)S_0 + \int_0^\infty h''(X)C_0(S_0, X)dX.$$

P_{0T}: The discount factor from T to present,

$C_0(S_0, X)$: The present value of a call option with strike X and current underlying asset price S_0.

1.3.4. Put-Call Parity. As indicated in Christian Ekstrand's *Financial Derivatives Modeling* [**17**], recall the previous formula,

$$h(S_t) = h(0) + h'(0)S_t + \int_0^\infty h''(X)(S_t - X)_+ dX.$$

Let a put option pay $h(S_T) = (X - S_T)_+$ at time T, then we have,

$$P_0(S_0, X) = XP_{0T} - S_0 + C_0(S_0, X) \tag{1.3.1}$$

$P_0(S_0, X)$: The present value of a put option with strike X and current underlying asset price S_0. The formula 1.3.1 is called put-call parity. It shows that a call and a put option only differ by a linear payoff.

The parity is obvious as the difference in payoff at maturity T,

$$(S_T - X)_+ - (X - S_T)_+ = S_T - X$$

is equal to the payoff of a forward contract. Since put options are cheaper than call options when the strike is low, we have the motivation to use as many put options to substitute call options to decrease the cost. The details can be understood from the computation,

$$
\begin{aligned}
&\int_0^K h''(X)(X-S_t)_+ dX + \int_K^\infty h''(X)(S_t-X)_+ dX \\
=&h'(K)(K-S_t)_+ - \int_0^K h'(X)\theta(X-S_t)dX \\
&- h'(K)(S_t-K)_+ + \int_K^\infty h'(X)\theta(X-S_t)_+ dX \\
=&h'(K)(K-S_t) - h(K) + h(S_t)
\end{aligned}
$$

where K is, arbitrarily, a positive number. The no-arbitrage principle implies that

$$
\begin{aligned}
&\int_0^K h''(X)P_0(S_0,X)dX + \int_K^\infty h''(X)C_0(S_0,X)dX \\
=&(h'(K)K - h(K))P_{0T} - h'(K)S_0 + V
\end{aligned}
\tag{1.3.2}
$$

which shows how a fixed-time payoff can be replicated with low strike puts and high strike calls. Denoting the right-hand side with $g(K)$, we obtain

$$
\begin{aligned}
g'(K) &= h''(K)(KP_{0T} - S_0) \\
g''(K) &= h'''(K)(KP_{0T} - S_0) + h''(K)
\end{aligned}
\tag{1.3.3}
$$

Assume for a moment that the second derivative of h is positive. The only extreme point of $g(K)$ is then a minimum located at the forward $K = P_{0T}^{-1}S_0 = F$. We conclude that (Ekstrand, 2011),

$$
V = (h(F) - h'(F)F)P_{0T} + h'(F)S_0 + \int_0^F h''(X)P_0(S_0,X)dX + \int_F^\infty h''(X)C_0(S_0,X)dX
\tag{1.3.4}
$$

is the static replication of the target portfolio that has the cheapest option content. The same result is obtained if h has a negative second derivative.

The formula (1.3.4) is very attractive in view of finance, it provides us a tool to replicate a nonlinear payoff by using some simple options. However, it has its own shortcoming in practice. We will discuss the problem in the following chapters.

1.4. The Black-Scholes-Merton Model

References for this section are Steven E. Shreve' *Stochastic Calculus for Finance II, Continuous Time Models* [**19**] and Bernt Øksendal's *Stochastic Differential Equations–An Introduction with Applications* [**20**].

1.4.1. Stochastic Process. In probability theory, a stochastic process, as a collection of random variables is often used to represent the evolution of some random value, or system, over time. It is the probabilistic counterpart to a deterministic process. Instead of describing a process which can only evolve in one way, in a stochastic process there is some indeterminacy, even if the initial condition is known, there are several directions in which the process may evolve.

Definition 1.4.1. *Suppose $(\Omega, \mathcal{F}, P)$ is a probability space, then a stochastic process X is a collection of random variables*

$$\{X_t, t \in T\}$$

defined on the given probability space.

1.4.2. Brownian Motion. Brownian Motion, as a classical stochastic process, is the random motion of particles suspended in a fluid resulting from their collision with the quick atoms or molecules in the gas or liquid. In mathematics, Brownian motion is described as a continuous-time stochastic process

$$\{W_t, t \geq 0\}.$$

W_t has the following preperties,

- $W_0 = 0$;
- the function $t \to W_t$ is almost surely continuous;
- $\{W_t\}_{t \geq 0}$ has stationary independent increments with $W_t - W_s$ is normally distributed with mean 0 and variance $t - s$ (for $0 \leq s < t$).

In particular, a geometric Brownian motion (GBM) is a continuous-time stochastic process in which the logarithm of the randomly varying quantity follows a Brownian motion with drift [**21**]. It is an important example of stochastic processes satisfying a stochastic differential equation. In financial mathematics, the famous Black-Scholes model for stock prices is dervied from geometric Brownian Motion.

DEFINITION 1.4.2. *A stochastic process S_t is said to follow a GBM if it satisfies the following stochastic differential equation,*

$$dS_t = \mu S_t dt + \sigma S_t dW_t \tag{1.4.1}$$

where W_t is a Brownian Motion and μ('the percentage drift') and σ('the percentage volatility') are constants.

1.4.3. The Black-Scholes-Merton Equation. It is well known that, for any given S_0, the stochastic differential equation (1.4.1) has a solution,

$$S_t = S_0 \text{exp}\left(\left(\mu - \frac{1}{2}\sigma^2\right) t + \sigma W_t\right).$$

This is the usual geometric Brownian motion model, and the distribution of $S(t)$ is log-normal with the expectation and variance,

$$E(S_t) = S_0 e^{\mu t}, \quad Var(S_t) = e^{2\mu t}(e^{\sigma^2 t} - 1).$$

In particular, if S_t is an asset price process, then we know that the distribution of the asset price is log-normal. μ and σ^2 represent the mean of the instantaneous rate of return and volatility respectively. For any given future time t, the rate of return on the asset has mean μt and volatility $\sigma^2 t$. As a result, the expectation of the asset price at t is $\exp\left(\ln S_0 + \mu t + \frac{1}{2}\sigma^2\right)$.

However, based on risk-neutrality, the expectation of future asset price should be $\exp\left(\ln S_0 + rt\right)$, where r is th risk-free interest rate. If we replace $\mu + \frac{1}{2}\sigma^2$ with r, the following formula will be obtained,

$$S_t = S_0 \mathrm{exp}\left(\left(r - \sigma^2\right) t + \sigma W_t\right).$$

The distribution of $S(t)$ is log-normal with the expectation and variance,

$$E(S_t) = S_0 e^{rt}, \quad Var(S_t) = e^{\sigma^2 t}.$$

The Black-Scholes-Merton model is a widely used pricing tool in the finanical market currently. It contains some derivative instruments. The Black-Scholes-Merton formula, derived from the model, provides us a theoretical estimate of the price of European options. Based on the historcial data, the Black-Scholes-Merton price is "fairly close" to the observed price, although some necessarily adjustments are always required.

In the Black-Scholes-Merton model, the Black-Scholes-Merton equation is a partial differential equation, which describes the price of the option over time. The equation is,

$$\frac{\partial V(S,t)}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V(S,t)}{\partial^2 S^2} + rS\frac{\partial V}{\partial S} - rV(S,t) = 0,$$

where

S: The price of the stock;

t: A time in years;

r: The annualized risk-free interest rate, continuously compounded;

σ: The volatility of the stock's returns;

$V(S,t)$: The price of a derivative as a function of time t and stock price S.

The solution to the Black-Scholes-Merton equation is given as a theorem below, the derivation can be found in Steven E. Shreve' *Stochastic Calculus for Finance II, Continuous Time Models* (page 218, 2004).

THEOREM 1.4.1. *(Black-Scholes-Merton Formula for Options) Suppose a European call option has strike price X and maturity T, σ is the volatility of the stock's returns and r is the annualized risk-free interest rate, then the price of the option at time t, where $t \leq T$ is given by the following formula,*

$$C_t(S_t, X) = S_t N(d_1) - X e^{-r(T-t)} N(d_2),$$

for a European put option which has strike price X and maturity T, σ is the volatility of the stock's returns and r is the annualized risk-free interest rate, then the price of the option at time t, where $t \leq T$ is given by the following formula,

$$P_t(S_t, X) = X e^{-r(T-t)} N(-d_2) - S_t N(-d_1),$$

with

$$d_1 = \frac{\ln \frac{S_t}{X} + (r + \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}}, \ d_2 = \frac{\ln \frac{S_t}{X} - (r + \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}}.$$

$N(\cdot)$ is the cumulative distribution function of standard normal distribution.

Note that these two formulas can be derived from each other by using put-call parity.

CHAPTER 2

A CLASS OF STATIC REPLICATION

It is well known (also see subsection 1.3.3), the formula

$$\begin{aligned} V =& (h(F) - h'(F)F)P_{0T} + h'(F)S_0 + \int_0^F h''(X)P_0(S_0, X)dX \\ &+ \int_F^\infty h''(X)C_0(S_0, X)dX \end{aligned}$$

has its practical shortcoming. According to the formula, for any target derivative, it can be replicated exactly by static replication only if options with strikes from zero to infinity are all available. Unfortunately, it is impossible in practice. By increasing the number of options in the static replication portfolio, we can achieve a better replication. How to find the optimal static replication is the question that needs to be answered. In this aspect, some researchers, for example Broadie and Jain (2008) [**4**] and Qiang Liu (2010) [**5**], have done some meaningful work.

2.1. Current Appromxiation Methods

References for this section are Mark Broadie and Ashish Jain' *Pricing and hedging volatility derivatives* [**4**] and Qiang Liu' *Optimal approximations of nonlinear payoffs in static replication* [**5**].

2.1.1. Mark Broadie and Ashish Jain' method. Broadie and Jain (2008) [**4**] proposed an approach to obtain an optimal approximation of the static replication. Given a fixed number of options with known equal-spaced strike prices, they processed an optimization problem by forming the lagrangian and solving the resulting system of linear equations to obtain the quantities of options that minimize the sum of squared differences between the payoff curve of a variance swap (Portfolio B) and that of the replicating bucket of options (Portfolio A), while requiring the initial values of portfolios A and B to be equal.

The optimization problem is described as following,

$$\begin{aligned} &\min_{w^p, w^c} \sum_{j=1}^{n} \left(V_A(S_T^j) - V_B(S_T^j) \right)^2 \\ &s.t. \qquad \sum_{i=1}^{n_p} w_i^p P_0(S_0, X_i^p) + \sum_{i=1}^{n_c} w_i^c C_0(S_0, X_i^c) = P_B(S_0). \end{aligned} \tag{2.1.1}$$

In the optimization problem, the decision variables are vectors w_p and w_c of sizes n_p and n_c, respectively, which represent the quantities of call and put options in the portfolio. The value $V_B(S_T^j)$ is the payoff of the portfolio of log contract and forward contract when the terminal stock price is S_T^j. The value $V_A(S_T^j)$ is the payoff of the portfolio of call and put options when the terminal stock price is S_T^j. The value $P_B(S_0)$ represents the initial value of the portfolio of the log contract and forward contract. The value $P_0(S_0, X_i^p)$ represents the initial value of the put option with strike X_i^p and $C_0(S_0, X_i^c)$ represents the initial value of the call option with strike X_i^c. Bradie and Jain (2008) also introduced some measures of error to evaluate the performance of the replicating portfolio of call and put options,

$$\begin{aligned} e_1 : &\qquad \frac{E\Big|V_A(S_T) - V_B(S_T)\Big|}{P_B(S_0)}, \\ e_2 : &\qquad \frac{\sqrt{E\left(V_A(S_T) - V_B(S_T)\right)^2}}{P_B(S_0)}, \\ e_\infty : &\qquad \frac{\max\Big|V_A(S_T) - V_B(S_T)\Big|}{P_B(S_0)}, \end{aligned} \tag{2.1.2}$$

where $P_B(S_0)$ represents the value of portfolio B, at $t = 0$ when the stock price is S_0. The expectation is under the real-world probability measure. The error measures e_1

and e_2 weigh the scenarios by their real-world probabilities, so that extreme outcomes have less effect on the results. The error measure e_∞ will be determined by the single scenario with the most extreme outcome. It is obvious that the e_∞ is the most rigorous one. This statement is conformed by their simulations.

2.1.2. Qiang Liu's methods. Without resorting to the method used by Bradie and Jain (2008), Liu (2010) introduced three optimal approximations of nonlinear payoffs. They are naive minimum area method, minimum expected area method and least expected squares method. In the first two methods, the author assumed that the strike prices are all adjustable. In the last method, the author created a method to replicate the payoff given a set of fixed strike prices.

In naive minimum area method, it was assumed that the best approximation would be obtained if the area given by the following formula is minimized.

$$A_i = \int_{X_i}^{X_{i+1}} \big[I_i(S_t) - f(S_t)\big]\, dS_t,$$

and the total area is

$$A = \sum_{i=0}^{n-1} A_i = A(X_0, X_1, \cdots, X_n),$$

which is a function of $X_i, i = 0, 1, \cdots, n$.

Then, it is possible to obtain a solution that minimizes the total area A, by solving the following system of equations,

$$\frac{\partial A(X_0, X_1, \cdots, X_n)}{\partial X_i} = 0 \qquad\qquad i = 0, 1, \cdots, n.$$

However, the system of equations may be not easy to solve since all equations are nonlinear. Fortunately, the problem is actually much easier than we thought. Note that X_{i+1} appears only in A_i and A_{i+1}. As a result

$$\frac{\partial A}{\partial X_{i+1}} = \frac{\partial A_i}{\partial X_{i+1}} + \frac{\partial A_{i+1}}{\partial X_{i+1}} = 0. \tag{2.1.3}$$

In the minimum expected area method, it is similar with minimum area method, but taking the distribution of the underlying asset price into consideration. In the minimum area method, it was assumed that the underlying asset price is distributed uniformly at maturity. However, this is not true in the financial market. To achieve a better replication, the differences between the replication and the actual payoff should be minimized on average for the majority of the expiry prices. Therefore, th expected total area shall instead be minimized for a better approximation.

Denote the disribution of the underlying asset price at maturity conditional on current price as $g(S_t)$, so the area A_i between X_i and X_{i+1} can be expressed as,

$$A_i = \int_{X_i}^{X_{i+1}} \big[I_i(S_t) - f(S_t)\big] g(S_t) dS_t.$$

It is easy to obtain the following result from equation (2.1.3),

$$\int_{X_i}^{X_{i+1}} (B_{i+1}S_t + C_{i+1}) g(S_t) dS_t + \int_{X_{i+1}}^{X_{i+2}} (D_{i+1}S_t + E_{i+1}) g(S_t) dS_t = 0, \qquad (2.1.4)$$

where B, C, D, and E are defined as in the following,

$$\begin{aligned}
B_{i+1} &= \frac{f'_{i+1}}{X_{i+1} - X_i} - \frac{y_{i+1} - y_i}{(X_{i+1} - X_i)^2}, \\
C_{i+1} &= \frac{y_{i+1} - X_i f'_{i+1}}{X_{i+1} - X_i} - \frac{y_i X_{i+1} - y_{i+1} X_i}{(X_{i+1} - X_i)^2}, \\
D_{i+1} &= -\frac{f'_{i+1}}{X_{i+2} - X_{i+1}} - \frac{y_{i+2} - y_{i+1}}{(X_{i+1} - X_i)^2}, \\
E_{i+1} &= -\frac{y_{i+2} - X_{i+2} f'_{i+1}}{X_{i+2} - X_{i+1}} + \frac{y_{i+1} X_{i+2} - y_{i+2} X_{i+1}}{(X_{i+2} - X_{i+1})^2}.
\end{aligned}$$

For a given $g(S_t)$, equation (2.1.4) is theoretically solvable. In financial markets, we assume that the price of underlying assets S_t, obeys a lognormal normal distribution with a mean of $\ln S_0 + (r - \frac{1}{2}\sigma^2)T$ and a variance of $\sigma^2 T$, where r is the constant risk-free rate, σ the constant volatility, and T the maturity. This model is derived from Geometric brownian motion. Based on the assumption of the distribution of stock price, the following equation can be derived from equation (2.1.4),

$$\big[B_{i+1}N(d_{1,i}) + (D_{i+1} - B_{i+1})N(d_{1,i+1}) - D_{i+1}N(d_{1,i+2})\big]S_0e^{rT}$$
$$+ C_{i+1}N(d_{2,i}) + (E_{i+1} - C_{i+1})N(d_{2,i+1}) - E_{i+1}N(d_{2,i+1}) = 0,$$

where $N(d)$ is the cumulative normal density function, and

$$d_{1,i} = \frac{\ln\frac{S_0}{X_i} + (r + \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}},\ d_{2,i} = d_{1,i} - \sigma\sqrt{T}.$$

Futhermore, the author also created a method to construct the replicating porfolio if the set of strike prices, X_i, $i = 0, 1, \cdots, n$, is fixed. A portfolio of call options at maturity can be expressed as,

$$\Pi = \sum_{j=1}^{n} w_j h_j(S_t),$$

where $h_j(S_t) = \max[S_t - X_j, 0], j = 1, 2, \cdots n$, and w_j is the weight of $h_j(S_t)$ that is to be determined by minimization.

With such a portfolio, an objective function for approximation can be defined as,

$$V = \int_0^\infty [f(S_t) - \Pi]^2 dS_t,$$

where $f(S_t)$ is a nonlinear payoff function.

Then, the approximation is given by the following expression,

$$\frac{\partial V}{\partial w_i} = -2\int_{X_i}^\infty [f(S_t) - \Pi](S_t - X_i)dS_t = 0.$$

Most of the current approximation methods, such as Qiang Liu's minimum area and minimum expected area, employ a mathematical tool which is called spline interpolation in numerical analysis. We will also adapt the idea in our study, some useful knowledge about spline interpolation are reviewed in the following section.

2.2. Spline Interpolation

References for this section are David Kincaid and Ward Cheney's *Numerical analysis: mathematics of scientific computing* [**22**] and Qiang Liu's *Optimal approximations of nonlinear payoffs in static replication* [**5**].

A spline function consists of polynomial pieces on subintervals joined together with certain continuity conditions. Formally, suppose that $n+1$ points $X_0, X_1, \cdots, X_n$ have been specified and satisfy $X_0 < X_1 < \cdots < X_n$. A spline function of degree 1 having partition pionts $X_0, X_1, \cdots, X_n$ is a continuous function I such that: On each interval $[X_{i-1}, X_i)$, I is a polynomial of degree ≤ 1.

The intervals $[X_{i-1}, X_i)$ do not intersect each other, and so no ambiguity arises in defining such a function at the partition points. A function such as this can be defined explicitly by

$$I(x) = \begin{cases} I_0(x) = a_0 x + b_0 & x \in [X_0, X_1), \\ I_1(x) = a_1 x + b_1 & x \in [X_1, X_2), \\ \quad \vdots & \quad \vdots \\ I_{n-1}(x) = a_{n-1} x + b_{n-1} & x \in [X_{n-1}, X_n). \end{cases} \tag{2.2.1}$$

If the partition points X_i and the coefficients a_i, b_i are all prescribed, then the value of I at S is obtained by first identifying the subinterval $[X_{i-1}, X_i)$ that contains S. For convenience, we can use the expression $a_0 x + b_0$ on the interval $(-\infty, X_1)$ and the expression $a_{n-1}x + b_{n-1}$ on the interval $(X_{n-1}, +\infty)$. The function I is continuous, and so the piecewise polynomials match up at the partition points; that is $I_i(X_{i+1}) = I_{i+1}(X_{i+1}) = y_i$, where $y = f(X)$ is the interpolated function and $y_i = f(X_i)$. So,

$$a_i = \frac{y_{i+1} - y_i}{X_{i+1} - X_i}, \qquad b_i = \frac{y_i X_{i+1} - y_{i+1} X_i}{X_{i+1} - X_i}.$$

The following results can be found in David Kincaid and Ward Cheney's *Numerical analysis: mathematics of scientific computing* [**22**]. For readers' convenience, the proof is provided.

THEOREM 2.2.1. *A spline function of degree 1 having partition points* $X_0 < X_1 < \cdots < X_n$ *can be expressed in the form,*

$$I(S_t) = pS_t + q + \sum_{i=1}^{n-1} r_i (S_t - X_i)^+,$$

where $p = \dfrac{y_1 - y_0}{X_1 - X_0}$, $q = \dfrac{X_1 y_0 - X_0 y_1}{X_1 - X_0}$, $r_i = \dfrac{y_{i+1} - y_i}{X_{i+1} - X_i} - \sum_{j=1}^{i-1} r_j$ $(i{=}1,\ 2,\cdots,\ n-1)$. *and* u^+ *is the positive part* u *(i.e.* $u^+ = u$, *if* $u \geq 0$, *and* $u^+ = 0$, *if* $u < 0$*).*

PROOF. To verify the expression, we just need to show that it can interpolate arbitary data $y_0, y_1, \cdots, y_n$ at the partition points $X_0, X_1, \cdots, X_n$ with $X_0 < X_1 < \cdots < X_n$ by using mathematical induction.

For $n = 0$: $S_t = X_0$, we want $pX_0 + q = y_0$. For $n = 1$: $S_t = X_1$, we want $pX_1 + q = y_1$. With $X_0 \neq X_1$, the system

$$\begin{cases} pX_0 + q = y_0 \\ pX_1 + q = y_1 \end{cases}$$

is uniquely solvable for (p, q).

By induction, suppose $n = k$, so $I(S_t)$ interpolates data $\begin{array}{c|cccc} t & X_0 & X_1 & \cdots & X_k \\ \hline y & y_0 & y_1 & \cdots & y_k \end{array}$,

we will show that it also interpolates $\begin{array}{c|c} t & X_{k+1} \\ \hline y & y_{k+1} \end{array}$.

For $S_t \leq K$, we have

$$I(S_t) = pS_t + q + \sum_{i=1}^{k-1} r_i (S_t - X_i)^+,$$

since, if $S_t \leq K$, all terms $(S_t - X_i)^+$, $j \geq k$, are equal to 0.

By the induction hypothesis, p, q, $\{r_i : 1 \leq i \leq k-1\}$ have been determined so that $I(X_i) = y_i$, $0 \leq i \leq k$.

For $X_0 \leq S_t \leq X_{k+1}$, similarly we have

$$I(S_t) = pS_t + q + \sum_{i=1}^{k} r_i (S_t - X_i)^+$$

and we only need to find r_k so that $I(X_{k+1}) = y_{k+1}$.

Now, we have

$$I(X_{k+1}) = pX_{k+1} + q + \sum_{i=1}^{k-1} r_i (S_t - X_i)^+ + r_k (X_{k+1} - K).$$

Since p, q, r_1, r_2, $\cdots$, r_{k-1} are known, by the induction hypothesis, and $X_{k+1} - K = 0$, the equation

$$I(X_{k+1}) = y_{k+1}$$

is uniquely solvable for r_k. □

As a method to express the spline function, the following corollary is useful in chapter three.

Corollary 2.2.1. *A spline function of degree 1 having partition points* $X_0 < X_1 < \cdots < X_n$ *can be expressed in the form,*

$$I(S_t) = a_k X_k + b_k + \sum_{i=1}^{k} u_i (X_i - S_t)^+ + \sum_{j=k}^{n-1} v_j (S_t - X_i)^+. \tag{2.2.2}$$

where

$$a_k = \frac{y_{k+1} - y_k}{X_{k+1} - X_k},\ b_k = \frac{y_k X_{k+1} - y_{k+1} X_k}{X_{k+1} - X_k},$$
$$u_i = \frac{y_{i+1} - y_i}{X_{i+1} - X_i} - \frac{y_i - y_{i-1}}{X_i - X_{i-1}},\ \textit{if}\ 1 \leq i \leq k-1,\ u_k = -\frac{y_k - y_{k-1}}{X_k - X_{k-1}},$$
$$v_i = \frac{y_{i+1} - y_i}{X_{i+1} - X_i} - \frac{y_i - y_{i-1}}{X_i - X_{i-1}},\ \textit{if}\ k+1 \leq i \leq n-1,\ v_k = \frac{y_{k+1} - y_k}{X_{k+1} - X_k}.$$

Proof. Since we know the following identities,

$$(S_t - X_i)^+ = \frac{1}{2}\left(S_t - X_i + |S_t - X_i|\right), \ (X_i - S_t)^+ = \frac{1}{2}\left(X_i - S_t + |X_i - S_t|\right).$$

we have $(S_t - X_i)^+ - (X_i - S_t)^+ = S_t - X_i$.

Then the proof can be done by the following computation.

$$\begin{aligned}
I(S_t) =& pS_t + q + \sum_{i=1}^{n-1} r_i (S_t - X_i)^+ \\
=& pS_t + q + \sum_{i=1}^{k} r_i (S_t - X_i)^+ + \sum_{j=k+1}^{n-1} r_j (S_t - X_j)^+ \\
=& pS_t + q + \sum_{i=1}^{k} r_i \left((S_t - X_i) + (X_i - S_t)^+\right) + \sum_{j=k+1}^{n-1} r_j (S_t - X_j)^+ \\
=& pS_t + q + \sum_{i=1}^{k} r_i S_t - \sum_{i=1}^{k} r_i X_i + \sum_{i=1}^{k} r_i (X_i - S_t)^+ + \sum_{j=k+1}^{n-1} r_j (S_t - X_j)^+ \\
=& \sum_{i=0}^{k} r_i S_t + \left(q - \sum_{i=1}^{k} r_i X_i\right) + \sum_{i=1}^{k} r_i (X_i - S_t)^+ + \sum_{j=k+1}^{n-1} r_j (S_t - X_j)^+ \\
=& \left(\sum_{i=0}^{k} r_i S_t - \sum_{i=0}^{k} r_i S_k\right) + \sum_{i=0}^{k} r_i S_k + \left(q - \sum_{i=1}^{k} r_i X_i\right) \\
& + \sum_{i=1}^{k} r_i (X_i - S_t)^+ + \sum_{j=k+1}^{n-1} r_j (S_t - X_j)^+ \\
=& \sum_{i=0}^{k} r_i \left((S_t - X_k)^+ - (X_i - S_t)^+\right) + \sum_{i=0}^{k} r_i S_k + \left(q - \sum_{i=1}^{k} r_i X_i\right) \\
& + \sum_{i=1}^{k} r_i (X_i - S_t)^+ + \sum_{j=k+1}^{n-1} r_j (S_t - X_j)^+ \\
=& \sum_{i=0}^{k} r_i (S_t - X_k)^+ - \sum_{i=0}^{k} r_i (X_k - S_t)^+ + \sum_{i=0}^{k} r_i S_k + \left(y_0 - \sum_{i=0}^{k} r_i X_i\right) \\
& + \sum_{i=1}^{k} r_i (X_i - S_t)^+ + \sum_{j=k+1}^{n-1} r_j (S_t - X_j)^+
\end{aligned}$$

$$= \sum_{i=0}^{k} r_i S_k + \left(y_0 - \sum_{i=0}^{k} r_i X_i \right) + \sum_{i=1}^{k-1} r_i (X_i - S_t)^+ + (r_k - \sum_{i=0}^{k} r_i)(X_k - S_t)^+$$
$$+ \sum_{i=0}^{k} r_i (S_t - X_k)^+ + \sum_{j=k+1}^{n-1} r_j (S_t - X_j)^+$$

So

$$I(S_t) = aS_k + b + \sum_{i=1}^{k} u_i (X_i - S_t)^+ + \sum_{j=k}^{n-1} v_j (S_t - X_i)^+$$

where

$$a_k = \textstyle\sum_{i=0}^{k} r_i S_k = \dfrac{y_{k+1} - y_k}{X_{k+1} - X_k},$$
$$b_k = y_0 - \sum_{i=0}^{k} r_i X_i = \frac{y_k X_{k+1} - y_{k+1} X_k}{X_{k+1} - X_k},$$
$$u_i = r_i = \frac{y_{i+1} - y_i}{X_{i+1} - X_i} - \frac{y_i - y_{i-1}}{X_i - X_{i-1}}, \text{ if } 1 \le i \le k-1,$$
$$u_k = \sum_{i=0}^{k} r_i = -\frac{y_k - y_{k-1}}{X_k - X_{k-1}},$$
$$v_i = r_i = \frac{y_{i+1} - y_i}{X_{i+1} - X_i} - \frac{y_i - y_{i-1}}{X_i - X_{i-1}}, \text{ if } k+1 \le i \le n-1,$$
$$v_k = \sum_{i=0}^{k} r_i = \frac{y_{k+1} - y_k}{X_{k+1} - X_k}. \qquad \square$$

In view of finance, all terms in the equation (2.2.2) have their specific meaning, the first two terms are a cash amount, the next k terms are nothing but a portfolio of put options with strike price X_i, $i = 1, 2, \cdots k$, and then, there is another portfolio of call options with strike price X_j, $j = k, k+1, \cdots, n-1$. Demeterfi mentioned that K should be close to the current underlying asset price such. By choosing such a K, not only put options but also call options tend to be OTM, which is prefered in the financial market [**13**].

In general, any nonlinear payoffs of an asset can be written as $y = f(S_t)$, where S_t is the underlying asset price at time t. Then the payoffs of the asset is a non-negative function of underlying asset price. It can be approximated by spline interpolation the equation (2.2.2) provides enough information for us to achieve this goal.

2.3. Evaluation of Current Methods

Bradie and Jain (2008) [**4**] used Black-Scholes-Merton and Heston stochastic volatility model to solve for the approximation problem (2.1.1) under the three different measures (2.1.2). However, all of these methods are computationally expensive. Liu (2010) [**5**] created three approximation methods which solved the computational issue. Furthermore, in the minimum expected area method, it is attractive that the author took the distribution of the underlying asset price into consideration. But, the measures used in these methods could be better. Becasue the minimized area, for example, does not necessarily mean the optimal approximation.

In chapter three, we will use a new measure to achieve the approximation target without increasing the cost of replication.

CHAPTER 3

STRATEGIES OF STATIC REPLICATION

In this chapter, we will define a new measure for evaluating the approximation of nonlinear payoffs in static replication. Then, the limit cost of interpolation methods will be introduced and proved. After that, we will show a new optimal strategy for static replication. Finally, some numercial examples will be exhibited.

Observation 1. *Suppose X and Y are two strike prices and $X < Y$. If a put option with strike price Y is preferred, then a put option with strike price X should be also preferred. If a call option with strike price X is preferred, then a call option with strike price Y should be also preferred.*

Based on this observation, a portfolio Π of put and call options can be divided into two sub-portfolios Π_p and Π_c, where Π_p and Π_c represent the portfolios of put and call options in Π respectively. Moreover, we have $\Pi_p \cap \Pi_c = \varnothing$ and $\Pi_p \cup \Pi_c = \Pi$. Let $K_p(\Pi) = \sup\{X : X \text{ is a strike price of a put option in } \Pi\}$ and $K_c(\Pi) = \inf\{Y : Y \text{ is a strike price of a call option in } \Pi\}$. Then, $K_p(\Pi) \leq K_c(\Pi)$.

Any number $K \in [K_p(\Pi), K_c(\Pi)]$ is referred as a separation point of the portfolio Π. If we assume that Π is the collection of all options in the market and all strike prices are possible, then $K_p = K_c$ and the separation point is unique, called put-call separation.

In the rest of this chapter we will use the following notations,

$P_0(S_0, X)$: The value of put options when the strike price is X;

$C_0(S_0, X)$: The value of call options when the strike price is X;

Π: A partition of $[0, \infty)$;

m: The first partition point of the partition;

K: The put-call separation;

M: The last partition point of the partition;

r: Risk-free rate;

T: The maturity of put and call options;

$V(\Pi, K, f)$: The cost to construct the corresponding portfolio by using piecewise interpolation for a payoff f with the put-call separation K.

3.1. Limit Cost of Interpolation Methods

Let $\Pi_{[m,M]}$ be any partition $[m = X_0 < X_1 < \cdots < X_n = M]$. Then we have the following theorem.

THEOREM 3.1.1. *Suppose $f \in C^1$ is a nonlinear payoff, and K is the put-call separation. If the norm of the partition $\Pi_{[m,M]}$ goes to zero, then the cost $V\left(\Pi_{[m,M]}, K, f\right)$ converges to the following value,*

$$\begin{aligned} V(m, M, K, f) = &- f'(m)P_0(S_0, m) - \int_m^K f'(X)dP_0(S_0, X) + f'(M)C_0(S_0, M) \\ &- \int_K^M f'(X)dC_0(S_0, X). \end{aligned} \tag{3.1.1}$$

Furthermore, if $f \in C^2$, then the above formula can be further simplified as

$$\begin{aligned} V(m, M, K, f) = &f'(K)\left[S_0 - Ke^{-rT}\right] + \int_m^K P_0(S_0, X)f''(X)dX \\ &+ \int_K^M C_0(S_0, X)f''(X)dX. \end{aligned} \tag{3.1.2}$$

It is worth noting that the theorem is very attractive since it provides us with a limit cost of all interpolation methods, such as, naive piecewise chord or minimum area. We will know the best that we can do given the smallest and largest strike prices. It is meaningful in the real financial market because the more options that are used, the higher the cost will be. Based on the theorem, once the limit of the cost is achieved, we should not keep increasing the number of options to pursue a better approximation since the action will be very inefficient and expensive.

As an example, suppose we have a nonlinear payoff function [**13**] given by the following expression,

$$f(S) = \frac{2}{T}\left(\frac{S - S_0}{S_0} - \ln\frac{S}{S_0}\right).$$

Assume the payoff gives a \$1 exposure for one volatility point squared, the current price of underlying asset S_0 is 100, maturity T is 0.25 years, the risk-free rate r is 5%, and the volatility σ is 20%.

Now, we can compute the limit cost $V\left(45, 140, 100, f\right)$ by using theorem 3.1.1. Since $f \in C^2$, so the limit cost can be computed by formula (3.1.2). We have

$$\begin{aligned} V\left(45, 140, 100, f\right) = &f'(100) \cdot \left[100 - 100e^{-0.05 \cdot 0.25}\right] + \int_{45}^{100} P_0(100, X) f''(X) dX \\ &+ \int_{100}^{140} C_0(100, X) f''(X) dX = 4.012025. \end{aligned}$$

Based on the same assumption, if we choose 18 equal spaced strikes between 45 and 140. The cost is equal to 4.177682.

It is obvious that we should not use hundreds of strikes to fill in the gap between 4.012025 and 4.177682. At this point, we can say the the approximation with cost 4.177682 is decent.

Proof of Theorem 3.1.1. Given a partition $\Pi_{[m,M]} = [m = X_0 < X_1 < \cdots < X_n = M]$, WLOG, let $X_k = K$. Based on equation (2.2.2), we know that the cost to construct the corresponding portfolio is

$$\sum_{i=1}^{k} u_i P_0\left(S_0, X_i\right) + \sum_{j=k}^{n-1} v_j C_0\left(S_0, X_i\right)$$

$$
\begin{aligned}
=&\sum_{i=1}^{k-1} u_i P_0(S_0, X_i) + u_k P_0(S_0, X_k) + v_k C_0(S_0, X_k) + \sum_{j=k+1}^{n-1} v_j C_0(S_0, X_i) \\
=&\sum_{i=1}^{k-1} \left(\frac{y_{i+1} - y_i}{X_{i+1} - X_i} - \frac{y_i - y_{i-1}}{X_i - X_{i-1}} \right) P_0(S_0, X_i) - \frac{y_k - y_{k-1}}{X_k - X_{k-1}} P_0(S_0, X_k) \\
&+ \frac{y_{k+1} - y_k}{X_{k+1} - X_k} C_0(S_0, X_k) + \sum_{j=k+1}^{n-1} \frac{y_{i+1} - y_i}{X_{i+1} - X_i} - \frac{y_i - y_{i-1}}{X_i - X_{i-1}} C_0(S_0, X_i) \\
=&- \frac{y_1 - y_0}{X_1 - X_0} P_0(S_0, X_0) - \sum_{i=0}^{k-1} \frac{y_{i+1} - y_i}{X_{i+1} - X_i} \left(P_0(S_0, X_{i+1}) - P_0(S_0, X_i) \right) \\
&+ \frac{y_n - y_{n-1}}{X_n - X_{n-1}} C_0(S_0, X_n) - \sum_{j=k}^{n-1} \frac{y_{j+1} - y_j}{X_{j+1} - X_j} \left(C_0(S_0, X_{j+1}) - C_0(S_0, X_j) \right),
\end{aligned}
$$

where $y_i = f(X_i)$, for $i = 0, 1, \cdots, n$.

Since $f \in C^1$, there exists $\xi_i \in (X_i, X_{i+1})$, $i = 1, 2, \cdots, n-1$, such that

$$
\frac{y_{i+1} - y_i}{X_{i+1} - X_i} = \frac{f(X_{i+1}) - f(X_i)}{X_{i+1} - X_i} = f'(\xi_i).
$$

Therefore the above computation can be continued with

$$
\begin{aligned}
=&f'(\xi_0) P_0(S_0, X_0) - \sum_{i=0}^{k-1} f'(\xi_i) \left(P_0(S_0, X_{i+1}) - P_0(S_0, X_i) \right) \\
&+ f'(\xi_{n-1}) C_0(S_0, X_n) - \sum_{j=k}^{n-1} f'(\xi_j) \left(C_0(S_0, X_{j+1}) - C_0(S_0, X_j) \right)
\end{aligned}
$$

Since $m = X_0, K = X_k,$ and $M = X_n$, we know that if the norm of the partition goes to zero, then ξ_0 converges to m, ξ_{n-1} converges to M, and therefore the cost $V(\Pi_{[m,M]}, K, f)$ converges to

$$
-f'(m)P_0(S_0, m) - \int_m^K f'(X) dP_0(S_0, X) + f'(M) C_0(S_0, M) - \int_K^M f'(X) dC_0(S_0, X).
$$

If $f \in C^2$, then the above formula can be further simplified by applying intergration by parts

$$\int_m^K P_0(S_0, X) f''(X) dX - f'(K)P(K) + f'(K)C(K)$$
$$+ \int_K^M C_0(S_0, X) f''(X) dX$$
$$= f'(K)\left[S_0 - Ke^{-rT}\right] + \int_m^K P_0(S_0, X) f''(X) dX$$
$$+ \int_K^M C_0(S_0, X) f''(X) dX.$$

The proof is complete. □

3.2. An Optimal Strategy of Static Replication

As we mentioned in chapter two, many measures used in current methods are not optimal. For example, in minimum area method and minimum expected area method [**5**], Liu (2010) used minimized area as the measure of the approximation. And he used least expected squares as the measure in another method [**5**]. However, neither minimized area nor least expected squares necessarily means the optimal approximation. Moreover, Kemeterfi, Derman, Kamal, and Zou (2010) used the minimized cost [**13**] as the measure. The existence of limit cost makes the measure not as good as we thought, since the construct cost of the replicating portfolio should be considered final.

In this section, we define a new measure, it will provide us with an optimal static replication of nonlinear payoffs.

Definition 3.2.1. *For any N-partition of $[a, b]$, let $\Pi_N = [a = a_0 < a_1 < a_2 < ... < a_n = b]$. Let $L(\Pi_N)$ be the class of all continuous piecewise linear function g on $[a, b]$ such that g is linear at every point on $[a, b]$ except possibly points of $\{a_i\}_{i=0}^n$.*

Suppose f is a continuous function on $[a, b]$. Then we define,

$$m(f, \Pi_N) = \inf_{g \in L(\Pi_N)} \max_{x \in [a,b]} |f(x) - g(x)|,$$

$$E(a, b, N, f) = \inf_{\Pi_N} \inf_{g \in L(\Pi_N)} \max_{x \in [a,b]} |f(x) - g(x)|.$$

LEMMA 3.2.1.

$$m(f, \Pi_N) = \min_{g \in L(\Pi_N)} \max_{x \in [a,b]} |f(x) - g(x)| \tag{3.2.1}$$

PROOF. By definition, we know that there exists $g_k \in L(\Pi_N)$, $k = 1, 2, \cdots$, such that

$$\lim_{k \to \infty} \max_{x \in [a,b]} |f(x) - g_k(x)| = m(f, \Pi_N)$$

We need to show that there is a subsequence $\{g_{k_j}\}$ of $\{g_k\}$ such that

$$\lim_{j \to \infty} g_{k_j} = \tilde{g} \text{ and } \tilde{g} \in L(\Pi_N).$$

Consider the sequence of vectors $\{\langle g_k(a_0), g_k(a_1), \cdots, g_k(a_n)\rangle\}$. Since this sequence is bounded, there is a convergent subsequence $\{\langle g_{k_j}(a_0), g_{k_j}(a_1), \cdots, g_{k_j}(a_n)\rangle\}$ which coverges to $\langle A_0, A_1, \cdots, A_n\rangle$.

Let $\tilde{g} \in L(\Pi_N)$ with $\tilde{g}(a_i) = A_i$, $i = 0, 1, \cdots, n$. Clearly

$$\lim_{j \to \infty} g_{k_j}(x) = \tilde{g}(x)$$

uniformly over $[a, b]$. We know

$$\lim_{j \to \infty} \max_{x \in [a,b]} |f(x) - g_{k_j}(x)| = \max_{x \in [a,b]} |f(x) - \tilde{g}(x)|$$

This implies

$$m(f, \Pi_N) = \max_{x \in [a,b]} |f(x) - \tilde{g}(x)|,$$

and therefore

$$m(f, \Pi_N) = \min_{g \in L(\Pi_N)} \max_{x \in [a,b]} |f(x) - g(x)|.$$

The proof is complete. □

For naive piecewise method, consider the partition

$$\Pi_N = [a = a_0 < a_1 < a_2 < \cdots < a_n = b]. \tag{3.2.2}$$

Let f be a payoff function and L be the naive piecewise function interpolating f based on Π_N. We know that

$$L(a_j) = f(a_j), \ \ j = 0, 1, 2, \cdots, n,$$

and

$$L(S) = \frac{S - a_j}{h} f(a_{j+1}) + \frac{a_{j+1} - S}{h} f(a_j), \quad S \in [a_j, a_{j+1}], \quad j = 0, 1, 2, \cdots, n.$$

If f is convex (i.e. $f'' > 0$), we know further that

$$\max_{S \in [a_j, a_{j+1}]} |f(S) - L(S)| = |f(S_j) - L(S_j)| = f(S_j) - L(S_j),$$

where $S_j \in (a_j, a_{j+1})$, satisfies

$$f'(S_j) = \frac{f(a_{j+1}) - f(a_j)}{h}.$$

Therefore the error in value can be formulated by

$$error = \max_{S \in [a,b]} |f(S) - L(S)| = \max_{0 \le j \le n-1} [L(S_j) - f(S_j)].$$

The error defined above can be used to measure the closeness of a replicating portfolio. Logically, the question we can ask is: what is the smallest possible error under such a measure? Before we investigate this quesiton, we may consider an example first.

Example 3.1. *Some authors* [**5, 13**] *consider the equal length partition, that is for partition* (3.2.2)*, let* $a_j = a + jh,\ j = 0, 1, 2, \cdots, n,$ *and* $h = \dfrac{b-a}{n}$*. Then, for the payoff function* $f(S) = \dfrac{2}{T}\left(\dfrac{S - S_0}{S_0} - \ln\dfrac{S}{S_0}\right),\ S \in [a, b],$ *we have*

$$S_j = \frac{h}{\ln(1+\frac{h}{a_j})}, \quad j = 0, 1, 2, \cdots, n$$

and since $\{L(S_j) - f(S_j)\}$ *is a decreasing sequence,*

$$\begin{aligned} error &= \max_{0\le j\le n-1}[L(S_j) - f(S_j)] = L(S_0) - f(S_0) \\ &= \frac{2}{T}\left(\ln\frac{S_0}{a_0} - 1 + \frac{a_0}{S_0}\right) \\ &= \frac{2}{T}\left(\ln\frac{\frac{h}{\ln(1+\frac{h}{a})}}{a} - 1 + \frac{a}{\frac{h}{\ln(1+\frac{h}{a})}}\right). \end{aligned}$$

If a=50, h=5, T=0.25*, One can see that*

$$error = 8\left(\ln\frac{\frac{5}{\ln(1+\frac{5}{50})}}{50} - 1 + \frac{50}{\frac{5}{\ln(1+\frac{5}{50})}}\right) = 0.0090288.$$

This example shows that the first sub-interval, not other sub-intervals, contributes the *error*. It suggests that an equal length partition may not be the best in view of reducing replication error.

Lemma 3.2.2. *Suppose* f *is continuous on a finite interval* $[a, b]$*,* $f > 0$ *on* (a, b) *and* $f(a) = f(b) = 0$*. Then*

$$\inf_L \max_{a\le x\le b} |f(x) - L(x)| = \frac{1}{2}\max_{a\le x\le b} f(x). \tag{3.2.3}$$

Here L *denotes a linear function on* $[a, b]$*. Moreover, the solution to the above optimization is unique, and the unique solution is*

$$L(x) = \frac{1}{2}\max_{a\le x\le b} f(x).$$

Proof. Denote $M = f(x_0) = \max_{a\le x\le b} f(x)$ and let $L_0(x) = \dfrac{M}{2}$. Then

$$\max_{a\le x\le b} |f(x) - L_0(x)| = \frac{M}{2}.$$

This implies

$$\inf_L \max_{a\le x\le b} |f(x) - L(x)| \le \frac{M}{2}.$$

Suppose there is a linear function L_1, such that

$$\max_{a\le x\le b} |f(x) - L_1(x)| < \frac{M}{2}.$$

Then we have

$$|L_1(a)| = |f(a) - L_1(a)| < \frac{M}{2},\ |L_1(b)| = |f(b) - L_1(b)| < \frac{M}{2},\ |f(x_0) - L_1(x_0)| < \frac{M}{2}.$$

Since L_1 is a linear function, we have

$$L_1(x_0) = \frac{b - x_0}{b - a} L_1(a) + \frac{x_0 - a}{b - a} L_1(b).$$

Therefore

$$\begin{aligned} |f(x_0) - L_1(x_0)| = |M - L_1(x_0)| &= \left| \frac{b - x_0}{b - a}(M - L_1(a)) + \frac{x_0 - a}{b - a}(M - L_1(b)) \right| \\ &\ge \frac{b - x_0}{b - a}(M - |L_1(a)|) + \frac{x_0 - a}{b - a} M - |L_1(b)|) \\ &> \frac{b - x_0}{b - a}\frac{1}{2}M + \frac{x_0 - a}{b - a}\frac{1}{2}M = \frac{M}{2}. \end{aligned}$$

The proof is complete. □

Corollary 3.2.1. *Suppose $f \in C^2[a,b], f'' < 0$ on (a,b). Then*

$$\inf_L \max_{a\le x\le b} |f(x) - L(x)| = \frac{1}{2}\left[f(x_0) - \frac{b - x_0}{b - a} f(a) - \frac{x_0 - a}{b - a} f(b) \right], \qquad (3.2.4)$$

where $x_0 \in (a,b)$ *is the point (unique) which satisfies* $f'(x_0) = \dfrac{f(b)-f(a)}{b-a}$. *The unique solution to the optimization is*

$$L(x) = \frac{1}{2}\left[f(x_0) + \frac{b+x_0-2x}{b-a}f(a) + \frac{2x-a-x_0}{b-a}f(b)\right]. \qquad (3.2.5)$$

REMARK 1. *Suppose* $f \in C^2(0,\infty)$, $f'' < 0$, *and* $[a,b] \subset (0,\infty)$. x_0 *in corollary* 3.2.1 *can be viewed as a function of* a *and* b. *It is not hard to show that* $\dfrac{\partial x_0}{\partial a} > 0$ *and* $\dfrac{\partial x_0}{\partial b} > 0$. *Indeed, taking the partial derivative with respect to* a *on both sides of*

$$f'(x_0) = \frac{f(b)-f(a)}{b-a}.$$

We have

$$f''(x_0)\frac{\partial x_0}{\partial a} = -\frac{f'(a)}{b-a} + \frac{f(b)-f(a)}{(b-a)^2} = \frac{f'(x_0)-f'(a)}{b-a} = \frac{f''(y_0)(x_0-a)}{b-a},$$

where $y_0 \in (a,x_0)$ *satisfies* $f''(y_0) = \dfrac{f'(x_0)-f'(a)}{b-a}$. *This is enough to conclude* $\dfrac{\partial x_0}{\partial a} > 0$. *Similarly, we can show* $\dfrac{\partial x_0}{\partial b} > 0$.

REMARK 2. *The quantity* $m = \inf_L \max_{a\le x\le b} |f(x) - L(x)|$ *can be also viewed as a function of* a *and* b. *It can be shown that* $\dfrac{\partial m}{\partial a} < 0$ *and* $\dfrac{\partial m}{\partial b} > 0$. *For example*

$$\begin{aligned}\frac{\partial m}{\partial b} =& \frac{1}{2}\left[f'(x_0)\frac{\partial x_0}{\partial b} + \frac{(b-x_0)-(b-a)(1-\frac{\partial x_0}{\partial b})}{(b-a)^2}f(a)\right.\\ &\left. + \frac{(x_0-a)-(b-a)\frac{\partial x_0}{\partial b}}{(b-a)^2}f(b) - \frac{x_0-a}{b-a}f'(b)\right]\\ =& -\frac{1}{2}\frac{(x_0-a)(b-x_0)}{b-a}f''(z_0)\end{aligned}$$

where $z_0 \in (x_0,b)$ *satisfies* $f''(z_0) = \dfrac{f'(b)-f'(x_0)}{b-x_0}$.

Proof of corollary 3.2.1. Denote $l(x) = \dfrac{f(b)-f(a)}{b-a}(x-a)+f(a)$. Let $h(x) = f(x) - l(x)$. Then, $h(a) = h(b) = 0$. Since $h'' = f'' < 0$, we know $h(x) > 0$. We have

$$\inf_L \max_{x\in[a,b]} |f(x) - L(x)| = \inf_L \max_{x\in[a,b]} |(f(x) - l(x)) - (L(x) - l(x))| = \inf_T \max_{x\in[a,b]} |h(x) - T(x)|.$$

Suppose x_0 satisfies $h'(x_0) = 0$, or equivalently, $f'(x_0) = \dfrac{f(b)-f(a)}{b-a}$. By Lemma 3.2.2, we have

$$\begin{aligned}\inf_L \max_{a\le x\le b} |f(x) - L(x)| =& \frac{1}{2}\max_{a\le x\le b} h(x) = \frac{1}{2}h(x_0) = \frac{1}{2}\left[f(x_0) - l(x_0)\right]\\ =&\frac{1}{2}\left[f(x_0) - \frac{f(b)-f(a)}{b-a}(x_0-a) - f(a)\right]\\ =&\frac{1}{2}\left[f(x_0) - \frac{b-x_0}{b-a}f(a) - \frac{x_0-a}{b-a}f(b)\right].\end{aligned}$$

and the unique solution to the optimization is

$$T(x) = \frac{1}{2}\left[f(x_0) - \frac{b-x_0}{b-a}f(a) - \frac{x_0-a}{b-a}f(b)\right].$$

Since $T(x) = L(x) - l(x)$, we have

$$\begin{aligned}L(x) = T(x) + l(x) &= \frac{1}{2}\left[f(x_0) - \frac{b-x_0}{b-a}f(a) - \frac{x_0-a}{b-a}f(b)\right] + \frac{f(b)-f(a)}{b-a}(x-a) + f(a)\\ &= \frac{1}{2}\left[f(x_0) - \frac{b-x_0}{b-a}f(a) - \frac{x_0-a}{b-a}f(b)\right] + \frac{b-x}{b-a}f(a) + \frac{x-a}{b-a}f(b)\\ &= \frac{1}{2}\left[f(x_0) + \frac{b+x_0-2x}{b-a}f(a) + \frac{2x-a-x_0}{b-a}f(b)\right].\end{aligned}$$

The proof is complete. □

THEOREM 3.2.1. *Suppose* $f \in C^2[a,b]$, $f'' < 0$ *on* (a,b). *Fix* $N > 0$, *then there exits a* $\widetilde{\Pi}_N = [a = a_0 < a_1 < a_2 < \cdots < a_N = b]$ *and* $\widetilde{g} \in L(\widetilde{\Pi}_n)$, *such that*

$$E(a,b,N,f) = m(f, \widetilde{\Pi}_N) \tag{3.2.6}$$

and for all $i = 0, 1, \cdots, N-1$

$$\max_{x \in [a_i, a_{i+1}]} |f(x) - g(x)| = E(a,b,N,f). \tag{3.2.7}$$

Here, we want to point out that,

- as long as the function f is concave down, the theorem applies;
- If $f'' > 0$, we just need to consider $-f$.

PROOF OF THEOREM 3.2.1. By definition, there exists

$$\Pi_k = [\ a = a_0^{(k)} < a_1^{(k)} < a_2^{(k)} < \cdots < a_N^{(k)} = b\ ],\ \ k = 1, 2, \cdots$$

such that

$$\lim_{k \to \infty} m(f, \Pi_k) = E(a,b,N,f).$$

By Lemma 3.2.1, for each k, there exists $g_k \in \Pi_k$, such that

$$\max_{x \in [a,b]} |f(x) - g_k(x)| = m(f, \Pi_k)$$

Claim. There exists a sub-indexes $\{k_j\}$, $j = 1, 2, \cdots$, such that

$$\lim_{j \to \infty} a_i^{(k_j)} = a_i,\ \ i = 0, 1, 2, \cdots, N. \text{ and } a = a_0 \le a_1 \le a_2 \le \cdots \le a_N = b.$$

In fact, since $\{a_i^{(k)}\}_{k=1}^{\infty}$ is a bounded sequence, we know that a convergent subsequence exists. If such a sub-indexes $\{k_j\}$ is identified, we can use the argument in the proof of Lemma 3.2.1 to conclude that there is a covergent subsequence of g_{k_j}, which converges to $\widetilde{g} \in L(\widetilde{\Pi}_N)$, and

$$\max_{a \le x \le b} |f(x) - \widetilde{g}(x)| = m(f, \widetilde{\Pi}_N).$$

This is enough to conclude that

$$E(a, b, N, f) = m(f, \widetilde{\Pi}_N).$$

It remains to show that

$$\max_{x \in [a_i, a_{i+1}]} |f(x) - \widetilde{g}(x)| = E(a, b, N, f).$$

Recall that $\widetilde{\Pi}_N = [a = a_0 < a_1 < a_2 < \cdots < a_N = b]$, and let $h(x) = f(x) - \widetilde{g}(x) + E(a, b, N, f)$. Clearly $h(x) \ge 0$ for all $x \in [a, b]$ and

$$\max_{a \le x \le b} h(x) = 2E(a, b, N, f)$$

but

$$\max_{a_i \le x \le a_{i+1}} h(x) = 2E(a, b, N, f).$$

WLOG, we assume $h(a_j) = 0$, $j = 0, 1, 2, \cdots, N$. In fact, if this is not the case, we can construct a non-negative continuous linear function $l \in L(\widetilde{\Pi}_N)$, such that

$$0 \le h(x) - l(x) \le h(x), \text{ for all } x \in [a, b].$$

Therefore $\widetilde{g}$ can be replaced by $\widetilde{\widetilde{g}} = \widetilde{g} + l$, and

$$\max_{a \le x \le b} \left| f(x) - \widetilde{\widetilde{g}}(x) \right| = E(a, b, N, f).$$

Indeed, such a function l is just the piecewise continuous linear function connecting the following points:

$$(a_0, h(a_0)),\ (a_1, h(a_1)), \cdots, (a_N, h(a_N))$$

Denote $\widetilde{h}(x) = h(x) - l(x)$. $\widetilde{h}$ can be viewed as a function of $a_0, a_1, a_2, \cdots, a_N$. It is also easy to see that $\widetilde{h}(x)$ is uniquely determined by $a_0, a_1, a_2, \cdots, a_N$. Indeed $\widetilde{\widetilde{g}}$ is the spline interpolation of f at points $a_0, a_1, a_2, \cdots, a_N$, and $\widetilde{h} = f - \widetilde{\widetilde{g}}$ is just the error function of f estimated by its spline interpolation. Since

$$\max_{a_i \le x \le a_{i+1}} \widetilde{h}(x) < 2E(a, b, N, f),$$

we can verify $\{a_j\}$ to obtain $\{\widetilde{a}_j\}$, such that the corresponding spline interpolation g_0 of f satisfies

$$\max_{\widetilde{a}_j \le x \le \widetilde{a}_{j+1}} (f(x) - g_0(x)) < 2E(a, b, N, f), \ j = 0, 1, \cdots, N-1.$$

A typical choice $\{\widetilde{a}_j\}$ can be

$$\widetilde{a}_0 = a_0, \ \widetilde{a}_1 = a_1 - \varepsilon_1, \cdots, \widetilde{a}_i = a_i - \varepsilon_i, \ \widetilde{a}_{i+1} = a_{i+1} - \varepsilon_{i+1}, \cdots, \ \widetilde{a}_{N-1} = a_{N-1} - \varepsilon_{N-1}, \ \widetilde{a}_N = a_N,$$

for sufficient small $\varepsilon_j > 0, \ j = 1, 2, \cdots, N-1$. The existence of such a g_0 is a contradiction of

$$\inf_{\Pi_N} m(f, \Pi_n) = E(a, b, N, f).$$

□

Example 3.2. *We still consider the payoff function of variance swaps,*

$$f(S) = \frac{2}{T}\left(\frac{S - S_0}{S_0} - \ln \frac{S}{S_0}\right).$$

Let $\Pi_N = [\ a = a_0 < a_1 < \cdots < a_N = b\ ]$ *be the optimal partition for* f *in terms of Theorem* 3.2.1. *We want to determine* $a_i, \ i = 1, 2, \cdots, N-1$. *Apply Corollary* 3.2.1 *and Theorem* 3.2.1 *to* $-f(S)$, *we have*

$$2E(a, b, N, f) = -f(S_i) + \frac{a_{j+1} - S_j}{a_{j+1} - a_j} f(a_j) + \frac{S_j - a_j}{a_{j+1} - a_j} f(a_{j+1}). \ j = 1, 2, \cdots, N-1.$$

Where $S_j \in (a_j, a_{j+1})$ *satisfies*

$$f'(S_j) = \frac{f(a_{j+1}) - f(a_j)}{a_{j+1} - a_j} = \frac{2}{T}\left(\frac{1}{S_0} - \frac{1}{a_{j+1} - a_j} \ln \frac{a_{j+1}}{a_j}\right).$$

Since $f'(S) = \frac{2}{T}\left(\frac{1}{S_0} - \frac{1}{S}\right)$, *we have*

$$S_j = \frac{a_{j+1} - a_j}{\ln \frac{a_{j+1}}{a_j}}, \ j = 0, 1, 2, \cdots, N-1.$$

It is now easy to obtain

$$\begin{aligned} E(a, b, N, f) &= \frac{1}{2}\left[\frac{a_{j+1} - S_j}{a_{j+1} - a_j} f(a_j) + \frac{S_j - a_j}{a_{j+1} - a_j} f(a_{j+1}) - f(S_j)\right] \\ &= \frac{1}{T}\left[\ln S_j - \frac{a_{j+1} - S_j}{a_{j+1} - a_j} \ln a_j - \frac{S_j - a_j}{a_{j+1} - a_j} \ln a_{j+1}\right] \\ &= \frac{1}{T}\left[\ln \frac{S_j}{a_j} - \frac{S_j - a_j}{a_{j+1} - a_j} \ln \frac{a_{j+1}}{a_j}\right] \\ &= \frac{1}{T}\left[\ln \frac{S_j}{a_j} - \frac{S_j - a_j}{S_j}\right] = \frac{1}{T}\left[\ln \frac{S_j}{a_j} - \frac{\frac{S_j}{a_j} - 1}{\frac{S_j}{a_j}}\right]. \end{aligned}$$

Let $H_j = \frac{S_j}{a_j}$, $j = 0, 1, 2, \cdots, N-1$. *We have*

$$\ln H_j - \frac{H_j - 1}{H_j} = TE(a, b, N, f).$$

Since the function

$$\ln x - \frac{x-1}{x}, \ x \in [1, \infty)$$

is an increasing function, we know that the solution to the equation

$$\ln x - \frac{x-1}{x} = TE(a, b, N, f)$$

is unique. Denote the solution by H. We have

$$\frac{S_j}{a_j} = H_j = H, \ j = 0, 1, 2, \cdots, N-1.$$

Note that

$$\frac{S_j}{a_j} = \frac{\frac{a_{j+1}}{a_j} - 1}{\ln \frac{a_{j+1}}{a_j}}, \ j = 0, 1, 2, \cdots, N-1,$$

and clearly the equation

$$H = \frac{y-1}{\ln y}, \ y > 1$$

has a unique solution for $H > 1$. Denote the solution by h, we have

$$\frac{a_{j+1}}{a_j} = h, \ j = 0, 1, 2, \cdots, N-1.$$

Thus $b = a_N = a_0 h^N = a h^N$. This implies

$$h = \left(\frac{b}{a}\right)^{\frac{1}{N}}, \ \text{and } a_j = a\left(\frac{b}{a}\right)^{\frac{j}{N}}.$$

We now back to the formulation of error $E(a, b, N, f)$. The above discussion shows

$$\begin{aligned} E(a, b, N, f) &= \frac{1}{T}\left(\ln H - \frac{H-1}{H}\right) \\ &= \frac{1}{T}\left(1 - \frac{\ln h}{h-1} + \ln \frac{\ln h}{h-1}\right) \\ &= \frac{1}{T}\left(\frac{\frac{1}{N}\ln\frac{b}{a}}{(\frac{b}{a})^{\frac{1}{N}} - 1} \ln \frac{\frac{1}{N}\ln\frac{b}{a}}{(\frac{b}{a})^{\frac{1}{N}} - 1} - 1\right). \end{aligned}$$

Thus, for the payoff $f(S) = \frac{2}{T}\left(\frac{S - S_0}{S_0} - \ln \frac{S}{S_0}\right)$, the optimal error with N-partition on $[a, b]$ is

$$E(a, b, N, f) = \frac{1}{T}\left(\frac{\frac{1}{N}\ln\frac{b}{a}}{(\frac{b}{a})^{\frac{1}{N}} - 1} - \ln \frac{\frac{1}{N}\ln\frac{b}{a}}{(\frac{b}{a})^{\frac{1}{N}} - 1} - 1\right).$$

It is easy to calculate that $E(50, 135, 18, f) = 1.706751e - 03$. *In contrast, if we use naive piecewise chords method, we have* $m(f, \Pi_N) = 9.082884e - 03$. *It clear that the optimal error in our method is much smaller than naive piecewise chords method.*

3.3. Numerical Examples

In this section, we will use the optimal approximation method to approximate the payoff of a variance swap. As a comparison, we will present the results of some current methods first, and then our new method will be exhibited.

We utilize the nonliner payoff function

$$f(S) = \frac{2}{T}\left(\frac{S - S_0}{S_0} - \ln\frac{S}{S_0}\right).$$

This function has been used many times in the previous chapters because some related research on variance swaps are based on this nonlinear payoff. For convenience, we also choose it as the payoff model. Assume the payoff gives a \$1 exposure for one volatility point squared, the current price of underlying asset S_0 is 100, maturity T is 0.25 years, the risk-free rate r is 5%, and the volatility σ is 20%. Eighteen strikes between 45 and 140 are chosen to approximate the payoff.

At first, we conduct the naive piecewise chords method, the job can be done by using Corollary 2.2.1. All values are computed by equation (2.2.2). It is worth noting that X_k should be chosen as the put-call separation based on the observation introducted in the begining of this chapter. Table 1 shows the results for the method. The construction cost represents the cost to take relevant action listed on the table in the financial market. The total cost is the construction cost plus the cash amount needed in the replicating portfolio. The error term, $m(f, \Pi(N))$ is defined in the Definition 3.2.1. It is clear that 100 is the call-put separation since it is the closest value available to the current asset price. As a result, all options tend to be OTM which is suggested by Demeterfi (1999) [**13**] and Liu (2010) [**5**].

Table 1. Naive Piecewise Chords Method for Static Replication

	Naive Piecewise Chords Method			
	Strike	*Weight*	*Value per Option*	*Cost*
Put	50.00	1.608054	0.000000	0.000000
	55.00	1.327808	0.000000	0.000000
	60.00	1.114987	0.000000	0.000000
	65.00	0.949558	0.000008	0.000007
	70.00	0.818416	0.000223	0.000182
	75.00	0.712696	0.003264	0.002326
	80.00	0.626224	0.027522	0.017235
	85.00	0.554593	0.147976	0.082067
	90.00	0.494591	0.552089	0.273058
	95.00	0.443828	1.534260	0.680948
	100.00	0.206927	3.372777	0.697919
Call	100.00	0.193574	4.614997	0.893342
	105.00	0.363224	2.477902	0.900033
	110.00	0.330920	1.191132	0.394170
	115.00	0.302744	0.513689	0.155516
	120.00	0.278019	0.199764	0.055538
	125.00	0.256205	0.070530	0.018070
	130.00	0.236862	0.022780	0.005396
	135.00	0.219629	0.006783	0.001490
Construction Cost				4.177298
Total Cost				4.177298
$m(f, \Pi(N))$				9.082884e-03

The minimum area method [**5**] is also presented. Table 2 shows the results for the method. The total cost is used as the criteria to measure the approximation. As we can see, the total cost in naive piecewise chords method is 4.177298 and it is 4.214943 in minimum area. If we assume that the lower total cost indicates the better replication, surprisingly, we will find that the result is worse after the optimization process. It is enough to conclude that, the total cost is not a good measure here. Actually, we have proved that there is a limit cost for interpolation methods in static replication. For this example, as we did in section 3.1, the limit cost is 4.012025. Both 4.177298 and 4.214943 are quite close to the limit, in this situation, it is not

Table 2. Minimum Area Method for Static Replication

	Minimum Area Method			
	Strike	*Weight*	*Value per Option*	*Cost*
Put	48.35	1.173564	0.000000	0.000000
	51.86	1.068874	0.000000	0.000000
	55.53	0.975602	0.000000	0.000000
	59.38	0.892285	0.000000	0.000000
	63.39	0.817676	0.000002	0.000002
	67.59	0.750705	0.000048	0.000036
	71.96	0.690452	0.000688	0.000475
	76.53	0.636125	0.006599	0.004198
	81.28	0.587037	0.043956	0.025804
	86.22	0.542595	0.210303	0.114110
	91.36	0.502280	0.748103	0.375757
	96.70	0.465640	2.052000	0.955494
	102.24	0.044872	4.510530	0.202395
Call	102.24	0.387410	3.540909	1.371784
	107.99	0.401858	1.620086	0.651044
	113.95	0.374064	0.617902	0.231135
	120.13	0.348632	0.194563	0.067831
	126.53	0.325325	0.050360	0.016383
	133.15	0.303934	0.010711	0.003255
Construction Cost				4.019702
Total Cost				4.214943
$m(f, \Pi(N))$				4.908813e-03

reasonable to announce that the naive piecewise method is better since the cost is lower. However, this does not mean minimized area is a good measure. One simple example will speak for itself. Suppose we use minimized area as the criteria to judge a replication method. If the difference between the payoffs of the target derivative and the replicating portfolio is immense in a interval, it is still possible that the area is almost 0 due to the tininess of the interval. Unfortunately, under the minimized area criterica, the issue can not be noticed. Furthermore, if the payoff is not a concave function, it is possible that the total area for a bad approximation is small because some postive values are cancelled by negative ones.

Table 3. An Optimal Approximation Method for Static Replication

	An Optimal Approximation Method			
	Strike	*Weight*	*Value per Option*	*Cost*
Put	47.77	1.000390	0.000000	0.000000
	50.71	0.942380	0.000000	0.000000
	53.83	0.887735	0.000000	0.000000
	57.15	0.836258	0.000000	0.000000
	60.66	0.787766	0.000000	0.000000
	64.40	0.742086	0.000005	0.000004
	68.36	0.699055	0.000080	0.000056
	72.57	0.658520	0.000955	0.000629
	77.04	0.620334	0.008265	0.005127
	81.78	0.584363	0.052474	0.030664
	86.81	0.550478	0.247823	0.136421
	92.16	0.518558	0.886246	0.459570
	97.83	0.424156	2.457487	1.042357
Call	97.83	0.064333	5.843096	0.375902
	103.85	0.460162	2.886434	1.328228
	110.24	0.433479	1.146017	0.496774
	117.03	0.408343	0.354309	0.144680
	124.23	0.384665	0.083212	0.032009
	131.88	0.362359	0.014572	0.005280
Construction Cost				4.057701
Total Cost				4.070321
$E(a, b, N, f)$				1.853729e-03

Now, we will exhibit our optimal strategy of the replication under the measure $E(a, b, N, f)$ based on the same assumption used in the previous two methods. Table 3 shows the results for the optimal method. The error term is expressed as $E(a, b, N, f)$ since it is the smallest $m(f, \Pi(N))$ with respect to all possible $\Pi(N)$. Under our measure, the error for the optimal method is 1.853729e-03, it is much smaller than 4.908813e-03 in minimum area method and 9.082884e-03 in naive piecewise chords method. Furthermore, we can say that minimum area method is better than naive piecewise chords method.

3.4. Conclusion and Future Research

We discovered and proved the existence of limit cost of interpolation methods in static replication. We also created the new measure $E(a, b, N, f)$ to evaluate different approximation methods. Moreover, we found an optimal strategy for the static replication.

In future research, it is desirable to take the distribution of underlying assets price S into consideration in the measure $E(a, b, N, f)$ and the optimal approximation method. The idea is that we should pay more attention to the area where S is most likely distributed.

We will attempt to evaluate different methods of static replication by utilizing more tools in view of statistics. Consider a nonlinear payoff $f(S)$. Our goal is to construct a portfolio, denoted by V, so that the payoff, $L_V(S)$, of the portfolio V is close to the nonlinear payoff $f(S)$. Denote the cost of construction of the portfolio V by C_V. As discussed before, $L_V(S)$ is a continuous piecewise linear function with the turning points being strike prices of options.

Let $\Delta_V(S) : f(S) - L_V(S)$ (The spot error of the replicating portfolio V).

We are interested in the situation that an unbiased static replication, V is used, that is

$$E(\Delta_V(S)) = 0. \tag{3.4.1}$$

We also can try to find a minimum variance replication if it exists, that is to find V_0 such that

$$Var\big(\Delta_{V_0}(S)\big) = \inf_V Var\big(\Delta_V(S)\big) \tag{3.4.2}$$

Most of these methods introduced in the previous sections are derived from spline interpolation. Our optimal method also draws on the idea of spline interpolation. These methods are solutions for the approximation of static replication in the sense of mathematics. To make our optimal strategy more meaningful in view of finance, we need to consider some basic desire in the financial market. For example, the equation 3.4.1 guarantees no-arbitrage principle in the market. If it fails, someone could earn a risk-free profit without any initial investment. We have set up these two criteria to measure a model in the sense of finance. In future research, we can think about these two problems respectively or consider both together.

On the other hand, it is assumed that no transaction cost exists in the current research. However, this is not true in the real world. The transaction cost, actually, plays a big role in the financial market. It is necessary to think about the effect of transaction cost on the optimal static replication.

Bibliography

[1] Peter Carr, Katrina Ellis, and Vishal Gupta. Static hedging of exotic options. *The Journal of Finance*, 53(3):1165–1190, 1998.

[2] Emanuel Derman, Deniz Ergener, and Iraj Kani. Static options replication. *The Journal of Derivatives*, 2(4):78–95, 1995.

[3] Baldeaux Jan and Rutkowski Marek. Static replication of forward-start claims and realized variance swaps. 2010.

[4] Mark Broadie and Ashish Jain. Pricing and hedging volatility derivatives. *The Journal of Derivatives*, 15(3):7–24, 2008.

[5] Qiang Liu. Optimal approximations of nonlinear payoffs in static replication. *Journal of Futures Markets*, 30(11):1082–1099, 2010.

[6] Emil SF Stamp and Thomas F Thorsen. Variance and volatility swaps. 2011.

[7] Marco Avellaneda. An introduction to option pricing and the mathematical theory of risk. *Milan Journal of Mathematics*, 67(1):225–250, 1997.

[8] Robert E Whaley. Derivatives on market volatility: Hedging tools long overdue. *The journal of Derivatives*, 1(1):71–84, 1993.

[9] Sam Howison, Avraam Rafailidis, and Henrik Rasmussen. On the pricing and hedging of volatility derivatives. *Applied Mathematical Finance*, 11(4):317–346, 2004.

[10] Stephen A Ross. *Neoclassical finance*. Princeton University Press, 2009.

[11] Stephen A Ross. Options and efficiency. *The Quarterly Journal of Economics*, 90(1):75–89, 1976.

[12] Douglas T Breeden and Robert H Litzenberger. Prices of state-contingent claims implicit in option prices. *Journal of business*, pages 621–651, 1978.

[13] Kresimir Demeterfi, Emanuel Derman, Michael Kamal, and Joseph Zou. A guide to volatility and variance swaps. *The Journal of Derivatives*, 6(4):9–32, 1999.

[14] Yunbi An and Wulin Suo. An empirical comparison of option-pricing models in hedging exotic options. *Financial Management*, 38(4):889–914, 2009.

[15] Robert G Tompkins. Static versus dynamic hedging of exotic options: an evaluation of hedge performance via simulation. *Journal of Risk Finance, The*, 3(4):6–34, 2002.

[16] John C Hull and Futures Options. other derivatives. *Prentice Hall, Englewood Cliffs, NJ*, 4:698, 2000.

[17] Christian Ekstrand. *Financial Derivatives Modeling*. Springer, 2011.

[18] Marek Capinski and Tomasz Zastawniak. *Mathematics for finance: an introduction to financial engineering*, volume 1. Springer London, 2003.

[19] Steven E Shreve. Stochastic calculus for finance ii, continuous time models, 2004.

[20] B Øksendal. Stochastic differential equations: An introduction with applications, 2003.

[21] Sheldon M Ross. *Introduction to probability models*. Academic press, 2006.

[22] David Ronald Kincaid and Elliott Ward Cheney. *Numerical analysis: mathematics of scientific computing*, volume 2. American Mathematical Soc., 2002.

APPENDIX A

MATLAB CODES FOR RELATED METHODS

1.1. The Matlab Code for Naive Piecewise Chords Method

```
fprintf(1,'This is naive piecewise chords method for static replication
    .\n');
s=input('Please input your strike prices:');
N=length(s);
x=input('Please input the put-call separation:');
k=find(abs(s(1,:)-x)<1e-10);
XK=s(k); %%%%%% the seperation of strikes

%Basic Conditions
S0=100;
T=0.25;
R=0.05;
SIGMA=0.2;
h=@(x)((2/T)*((x-S0)/S0-log(x/S0))); %payoff function

%Weight of Put options
if 1<k
    W(1)=(h(s(1))-h(s(2)))/(s(2)-s(1));
    V(1) = Putvalue(s(2),S0,T,R,SIGMA);



for i=2:k-1
```

```
    V(i) = Putvalue(s(i+1),S0,T,R,SIGMA);
    W(i) =(h(s(i))-h(s(i+1)))/(s(i+1)-s(i));
end


for i=1:k-2
   w(i)=(W(i)-W(i+1))*100;
end
w(k-1)=W(k-1)*100;

else

     W(1)=(h(s(1))-h(s(2)))/(s(2)-s(1));
     V(1) = Putvalue(s(2),S0,T,R,SIGMA);
     w(1)=W(1)*100;
end

%Weight of Call options
for i=k:N-1
   V(i) = Callvalue(s(i),S0,T,R,SIGMA);
   W(i) =(h(s(i+1))-h(s(i)))/(s(i+1)-s(i));
end

w(k)=W(k)*100;

for i=k+1:N-1
    w(i)=(W(i)-W(i-1))*100;
end
```

```
%Cost
    C=w.*V;
    TC=sum(C)+h(s(k))*exp(-R*T)*100;

%Max Error
z1=s(2):0.001:s(N-1);
y=interp1(s,h(s),z1);
m=y-h(z1);
Max = max(m);

%Graph
plot(s,h(s),'r',z1,h(z1),'b') %% red is interpolation, blue is the real
    payoffs

fprintf( '                           Naive Piecewise Chords\n');
fprintf( '                 -----------------------------------------------\
    n');
fprintf( '\n');
fprintf( '          Strike      Weight         Value per option
    Cost\n');
fprintf( '
    ----------------------------------------------------------------
    \n');
fprintf('Put %10.2f %13.6f %17.6f %17.6f\n',s(2),w(1),V(1),C(1));
for i=2:k-1
fprintf('    %10.2f %13.6f %17.6f %17.6f\n',s(i+1),w(i),V(i),C(i));
end
fprintf( '\n');
fprintf('Call%10.2f %13.6f %17.6f %17.6f\n',s(k),w(k),V(k),C(k));
```

```
for i=k+1:N-1
fprintf('    %10.2f %13.6f %17.6f %17.6f\n',s(i),w(i),V(i),C(i));
end
fprintf('\n');
fprintf(' Construction Cost %45.6f\n',sum(C));
fprintf(' Total Cost %52.6f\n',TC);
fprintf(' Max error %53.6e\n',Max);
```

1.2. The Matlab Code for Minimum Area Method

```
fprintf(1,'This is minimum area method for static replication.\n');
a=input('Please input your lower bound of strike prices:');
b=input('Please input your upper bound of strike prices:');
N=input('Please input the size of partition:');
k=input('Please input the position of put-call separation:');

%Basic Conditions
S0=100;
T=0.25;
R=0.05;
SIGMA=0.2;
h=@(x)((2/T)*((x-S0)/S0-log(x/S0))); %payoff function
%step 1
V = zeros(N,1);
tol =1e-7;
V(1)=a;
V(N)=b;
V(2)=50;
%step 2
for i=3:N
```

```
        aa=V(i-2);
        bb=V(i-1);
V(i) = Newton(aa,bb);
end
%step 3
VN1=V(N);
V21=V(2);
if V(N)>b
        V(2)=V(2)-1;
else
        V(2)=V(2)+1;
end
%step 4
VN2=VN1;
while (VN1-b)*(VN2-b)>0
for i=3:N
        aa=V(i-2);
        bb=V(i-1);
V(i) = Newton(aa,bb);
end
V22=V(2);
if V(N)>b
        V(2)=V(2)-1;
else V(2)=V(2)+1;
end
VN2=V(N);
end
while abs(V(N)-b) >tol
%step 5
```

```
V(2)=(V21+V22)/2;
for i=3:N
    aa=V(i-2);
    bb=V(i-1);
V(i) = Newton(aa,bb);
end
%step 6
if V(N) > b
    if V22>V21
        V22=V(2);
    else
        V21=V(2);
    end
else
    if V22 > V21
        V21=V(2);
    else
        V22 = V(2);
    end
end


end


% the seperation of strikes
XK=V(k);


%Weight of Put options
if 1<k
   W(1)=(h(V(1))-h(V(2)))/(V(2)-V(1));
```

```
      VV(1) = Putvalue(V(2),S0,T,R,SIGMA);

for i=2:k-1
      VV(i) = Putvalue(V(i+1),S0,T,R,SIGMA);
      W(i) =(h(V(i))-h(V(i+1)))/(V(i+1)-V(i));
end

for i=1:k-2
    w(i)=(W(i)-W(i+1))*100;
end
w(k-1)=W(k-1)*100;
else
       W(1)=(h(V(1))-h(V(2)))/(V(2)-V(1));
       VV(1) = Putvalue(V(2),S0,T,R,SIGMA);
       w(1)=W(1)*100;
end

%Weight of Call options
for i=k:N-1
    VV(i) = Callvalue(V(i),S0,T,R,SIGMA);
    W(i) =(h(V(i+1))-h(V(i)))/(V(i+1)-V(i));
end

w(k)=W(k)*100;
for i=k+1:N-1
      w(i)=(W(i)-W(i-1))*100;
end

%Cost
```

```
    C=w.*VV;
    TC=sum(C)+h(V(k))*exp(-R*T)*100;

%Max Error
z1=V(2):0.001:V(N-1);
y=interp1(V,h(V),z1);
m=y-h(z1);
Max = max(m);

%Graph
plot(V,h(V),'r',z1,h(z1),'b')

fprintf( '                            Minimum Area Method\n');
fprintf( '                -------------------------------------------\n');
fprintf( '\n');
fprintf( '         Strike      Weight        Value per option         Cost\n');
fprintf( '------------------------------------------------------------------\n');
fprintf('Put %10.2f %13.6f %17.6f %17.6f\n',V(2),w(1),VV(1),C(1));
for i=2:k-1
fprintf('    %10.2f %13.6f %17.6f %17.6f\n',V(i+1),w(i),VV(i),C(i));
end
fprintf( '\n');
fprintf('Call%10.2f %13.6f %17.6f %17.6f\n',V(k),w(k),VV(k),C(k));
for i=k+1:N-1
fprintf('    %10.2f %13.6f %17.6f %17.6f\n',V(i),w(i),VV(i),C(i));
```

```
end
fprintf('\n');
fprintf(' Construction Cost %45.6f\n',sum(C));
fprintf(' Total Cost %52.6f\n',TC);
fprintf(' Max error %53.6e\n',Max);
```

1.3. The Matlab Code for An Optimal Approximation Method

```
fprintf(1,'This is an optimal apprximation method.\n');
A=input('Please input your lower bound of strike price:');
B=input('Please input your upper bound of strike price:');
N=input('Please input the size of partition:');
k=input('Please input the position of put-call separation:');

%Basic Conditions
S0=100;
T=0.25;
R=0.05;
SIGMA=0.2;
h=@(x)((2/T)*((x-S0)/S0-log(x/S0)));          %payoff function
t=(B/A)^(1/(N-1));                              %the step length of strikes
s(1)=A;
for i=2:N
    s(i)=A*t^(i-1);
end
XK=s(k);                                        % Put-Call separation

for j=1:N-1
    yy(j)=optimal(s(j),s(j),s(j)*t);
end
```

```
yy(N)=optimal(s(N),s(N-1),s(N));

%Weight of Put options
if 1<k
    W(1)=(yy(1)-yy(2))/(s(2)-s(1));
    V(1) = Putvalue(s(2),S0,T,R,SIGMA);

for i=2:k-1
    V(i) = Putvalue(s(i+1),S0,T,R,SIGMA);
    W(i) =(yy(i)-yy(i+1))/(s(i+1)-s(i));
end

for i=1:k-2
   w(i)=(W(i)-W(i+1))*100;
end
w(k-1)=W(k-1)*100;
else

      W(1)=(yy(1)-yy(2))/(s(2)-s(1));
      V(1) = Putvalue(s(2),S0,T,R,SIGMA);
      w(1)=W(1)*100;
end

%Weight of Call options
for i=k:N-1
   V(i) = Callvalue(s(i),S0,T,R,SIGMA);
   W(i) =(yy(i+1)-yy(i))/(s(i+1)-s(i));
end
```

```
w(k)=W(k)*100;

for i=k+1:N-1
    w(i)=(W(i)-W(i-1))*100;
end

%Cost
    C=w.*V;
    TC=sum(C)+yy(k)*exp(-R*T)*100;

%Max Error
M=floor((s(N-1)-s(2))/0.01+1);
for i=1:M
z(i)=s(2)+(i-1)*0.01;
[m,index]=min(abs(z(i)-s));
aa(i)=s(index);
if z(i)>=aa(i);
    a(i)=aa(i);
    b(i)=a(i)*t;
else
    b(i)=aa(i);
    a(i)=b(i)/t;
end
    y(i)=optimal(z(i),a(i),b(i));
    m(i)=y(i)-h(z(i));
end
Max = max(abs(m))

%Graph
```

```
plot(s, yy,'r',z,h(z),'b')

fprintf( '                         An Optimal Apprximation Method\n');
fprintf( '                ----------------------------------------------------------\
    n');
fprintf( '\n');
fprintf( '         Strike      Weight        Value per option         
    Cost\n');
fprintf( '
    -----------------------------------------------------------------------
    \n');
fprintf('Put %10.2f %13.6f %17.6f %17.6f\n',s(2),w(1),V(1),C(1));
for i=2:k-1
fprintf('    %10.2f %13.6f %17.6f %17.6f\n',s(i+1),w(i),V(i),C(i));
end
fprintf( '\n');
fprintf('Call%10.2f %13.6f %17.6f %17.6f\n',s(k),w(k),V(k),C(k));
for i=k+1:N-1
fprintf('    %10.2f %13.6f %17.6f %17.6f\n',s(i),w(i),V(i),C(i));
end
fprintf('\n');
fprintf(' Construction Cost %45.6f\n',sum(C));
fprintf(' Total Cost %52.6f\n',TC);
fprintf(' Max error %53.6e\n',Max);
```

1.4. The Matlab Code for the Value of Put Options in Black-Scholes

```
function [PV] = Putvalue(PP,S0,T,R,Sigma)
d1 =1./(Sigma.*((T)^0.5)).*[log(S0./PP)+(R+Sigma^2./2).*T];
d2 =1./(Sigma.*((T)^0.5)).*[log(S0./PP)+(R-Sigma^2./2).*T];
```

```
PV = normcdf(-d2).*PP.*exp(-R.*T)-normcdf(-d1).*S0;
end
```

1.5. The Matlab Code for the Value of Call Options in Black-Scholes

```
function [CV] = Callvalue(CP,S0,T,R,SIGMA)
d1 =1/(SIGMA.*((T).^0.5)).*[log(S0./CP)+(R+SIGMA.^2./2).*T];
d2 =1/(SIGMA.*((T).^0.5)).*[log(S0./CP)+(R-SIGMA.^2./2).*T];
CV = normcdf(d1).*S0-normcdf(d2).*CP.*exp(-R.*T);
end
```

1.6. The Matlab Code for the Optimal Approximation Line

```
function [L] = optimal(x,a,b)
h=@(x)((2/T)*((x-S0)/S0-log(x/S0)));
x0=fzero(@(y)((2/25 - 8/y)-(h(b)-h(a))/(b-a)),a);
L =(1/2)*(h(x0)+(b+x0-2*x)/(b-a)*h(a)+(2*x-a-x0)/(b-a)*h(b));
end
```

1.7. The Matlab Code for the Newton Method

```
function x=Newton(a,b)
x=b+1;
xx = x - (x-a-b*(log(x)-log(a)))/(1-b/x);
Tol = 10^(-10);

while abs(xx-x)>Tol;
    x = xx;
    xx = x - (x-a-b*(log(x)-log(a)))/(1-b/x);
end
end
```