SIMULATION STUDY ON OPTION PRICING UNDER JUMP DIFFUSION MODELS

by

Justin Rodrigues

A Thesis Submitted to the Faculty of

The Charles E. Schmidt College of Science

in Partial Fulfillment of the Requirements for the Degree of

Master of Science

Florida Atlantic University

Boca Raton, FL

December 2013

SIMULATION STUDY ON OPTION PRICING UNDER JUMP DIFFUSION MODELS

by

Justin Rodrigues

This thesis was prepared under the direction of the candidate's thesis advisor, Dr. Hongwei Long, Department of Mathematical Sciences, and has been approved by the members of his supervisory committee. It was submitted to the faculty of the Charles E. Schmidt College of Science and was accepted in partial fulfillment of the requirements for the degree of Master of Science.

SUPERVISORY COMMITTEE:

Hongwei Long, Ph.D.
Thesis Advisor

Paul Yiu, Ph.D.

Heinrich Niederhausen, Ph.D.

Lee Klingler, Ph.D.
Chair, Department of Mathematical Sciences

Russell Ivy, Ph.D.
Interim Dean, The Charles E. Schmidt College of Science

Barry T. Rosson, Ph.D.
Dean, Graduate College

November 20, 2013
Date

ACKNOWLEDGEMENTS

The completion of this thesis would not have been possible without the help of several individuals who, in one way or another, contributed and extended their valuable assistance.I would like to express my utmost gratitude to my advisor, Dr. Hongwei Long, for his guidance and encouragement, and my committee members.

ABSTRACT

Author:	Justin Rodrigues
Title:	Simulation Study on Option Pricing Under Jump Diffusion Models
Institution:	Florida Atlantic University
Thesis Advisor:	Dr. Hongwei Long
Degree:	Master of Science
Year:	2013

The main objective of this thesis is to simulate, evaluate and discuss several methods for pricing European-style options. The Black-Scholes model has long been considered the standard method for pricing options. One of the downfalls of the Black-Scholes model is that it is strictly continuous and does not incorporate discrete jumps. This thesis will consider two alternate Lévy models that include discretized jumps; The Merton Jump Diffusion and Kou's Double Exponential Jump Diffusion. We will use each of the three models to price real world stock data through software simulations and explore the results.

Keywords: Lévy Processes, Brownian motion, Option pricing, Simulation, Black-Scholes, Merton Jump Diffusion, Kou, Kou's Double Exponential Jump Diffusion.

DEDICATION

To my friends and family who've always shown me love.

SIMULATION STUDY ON OPTION PRICING UNDER JUMP DIFFUSION MODELS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Brownian motion was used by Black and Scholes in 1973 to determine financial returns over a continuous time period (Black, F. and M. Scholes 1973). They discovered a stochastic partial differential equation that determines an option's price over time. This has been a popular method for describing asset returns over a continuous time period for the past 30 years.

One of the benefits of the Black-Scholes model is its tractability (Kou, S.G. 2002) The model has a straightforward pricing formula that can easily be used by hand or via software to price options. The computations are not difficult, and computer software can quickly price hundreds of thousands of options using the Black-Scholes model. This has led to its popularity and widespread use in the financial industry.

There are several downfalls of the Black-Scholes model however, making it a bit outdated and unreliable. The Black-Scholes model is derived under the assumption that the underlying asset's price process is purely continuous. If this assumption is violated, the model will underestimate options that are out of the money, call options with a strike price that is higher than the market price of the underlying asset. In the marketplace it is common for an option's underlying asset price process to not be purely continuous. Therefore, there is a need for improved modeling of option prices.

Lévy processes have emerged as an improvement over the Black-Scholes model for financial modeling. A Lévy process contains a Brownian motion that generates a normal distribution, while non-normal distributions can be generated by specifying a particular Lévy density for a Lévy jump process which determines the arrival of jumps of all sizes (Wu, Liuren 2008).

Returns are driven by many economic forces. Each of these forces can vary stochastically over time. We can use multiple Lévy processes to model the return from each economic force. Applying time change to each Lévy component will randomize the time for each process and can simulate the stochastically varying impacts from the different economic forces. By choosing appropriate Lévy processes and time changes, we can determine returns of almost any financial product.

1.1 BACKGROUND

1.2 OPTIONS BACKGROUND

Options are a special type of financial instrument. There are two kinds of options. Call options give the owner the right to buy the underlying asset by a certain date for a predetermined price. Put options give the owner the right to sell the underlying asset by a certain date for a predetermined price. The predetermined price is known as the strike price, and the date in the option contract is called the option's expiration date. There are many variations of options as well. American options can be exercised at any time up until the expiration date. European style options can only be exercised on the expiration date itself. One option contract is generally an agreement to buy or sell 100 shares of a stock. It is important to note that options give their owner the "right" to exercise them. They are not obligated to exercise an option. This is the main difference between options and a futures contract or a forward contract. While those contracts require no money up front, there is a cost to enter into an option contract. One aspect of options that is very attractive to investors is that it gives the holder the ability to leverage their money to a great extent.

For instance, suppose Google stock is trading at $500, it would take $500,000 to buy 1000 shares of the stock. Instead of purchasing the stock, you could purchase a GOOG call option with a strike price of $500 and an expiration 1 month in the future. Suppose in October you buy 10 GOOG NOV 30 call options for $10. This transaction will allow you to capitalize on an upward movement in stock price while minimizing the risk of the stock price falling.

Since each contract is for 100 shares, in this example you purchased the right to buy 1000 shares of Google Stock for $500 per share. The price, $10, is quoted on a per share basis. Therefore the total cost of this option contract is $10,000. If Google's stock price stays at or below $500, $10,000 is the most that you can lose. However, suppose the stock price rises to $600 at expiration, then your $10,000 investment will be worth around $100,000.

	Purchase Price	**Sale Price**	**Profit (L)**	**% Gain (L)**
Stock Price	$500	$600	$100	20%
1,000 Shares	$500,000	$600,000	$100,000	20%
10 NOV 30 Calls	$10,000	$100,000	$90,000	900%

Now let us examine this same scenario when Google's stock price drops from $500 to $400.

	Purchase Price	**Sale Price**	**Profit (L)**	**% Gain (L)**
Stock Price	$500	$400	($100)	(20%)
1,000 Shares	$500,000	$400,000	($100,000)	(20%)
10 NOV 30 Calls	$10,000	$0	($10,000)	(100%)

It is clear that using a call option in this case provides safety net by limiting the amount of money you can lose in the worst case scenario. Another benefit is that options allow you to leverage your money. For instance, in the first case where Google stock price rises to \$600 we see a 900% return on our investment. Compare this to the 20% return realized in the traditional stock investment. This ability to leverage capital is the other attractive feature of options.

On the other hand, if we expect Google stock price to drop, we can purchase put options. This will allow us to sell Google stock for a set price. If Google's stock price drops, we earn money through executing the put options. Once again, our money is leveraged so that we will realize a greater return on our option than if we were to have invested the same amount in shares of Google stock. It is important to determine and establish a fair market price for options. If fair prices are not set, then arbitrage opportunities can present themselves. In an arbitrage, there is the opportunity to earn a risk-less profit on an investment. In option markets it is imperative that arbitrage opportunities do not exist, and that the market is fair.

An important relationship between puts and calls is put-call parity. Let c be a call options and p be a put option. Let K be the strike price, r the risk-free rate, T the time in years of the option and S the underlying asset value. Put-call parity states the following:

$$c + Ke^{-rT} = p + S_0$$

This means that the value of an option with a certain strike price and length can be extrapolated from the value of a put option with the same strike price and length. If this parity did not exist, then arbitrage opportunities would occur (Hull, J. C. 2009).

1.3 LÉVY PROCESS

It is difficult to talk about jump diffusion models without a bit of background on Lévy processes first. A Lévy process is a stochastic process with independent, stationary increments. The two most popular Lévy processes are Brownian motion and Poisson processes. Brownian motion is the only purely continuous Lévy process, all others have discontinuous paths. These discontinuous Lévy processes can have infinitely many such jumps. We will explore two such Lévy processes with discretized jumps; Merton's jump diffusion (Merton, R. 1976) and Kou's double exponential jump diffusion (Kou, S.G. 2002).

1.4 MATHEMATICAL DEFINITION

A Lévy process is a stochastic process $\{X_t, t \geq 0\}$ defined on a probability space $\{\Omega, \mathcal{F}, \mathbb{P}\}$ with the following three conditions satisfied.

1. $X_0 = 0$ almost surely.

2. Independent increments: For any $0 \leq t_1 < t_2 ... < t_n < \infty$, $X_{t_2} - X_{t_1}, X_{t_3} - X_{t_2}, ..., X_{t_n} - X_{t_{n-1}}$ are independent.

3. Stationary increments: For any $s < t$, $X_t - X_s$ is equal in distribution to X_{t-s}.

Having independent increments means that increments $X_s - X_t$ and $X_u - X_v$ are independent whenever the two time intervals; (t, s) and (v, u) do not overlap. We can extend this property so that any finite number of pairwise non-overlapping intervals are mutually (not only pairwise) independent.

Having stationary increments means that the distribution of any increment $X_t - X_s$ depends only on the length of the time interval $t - s$. Increments that are on time intervals of equal length are identically distributed.

The distribution of a Lévy process is given by the Lévy-Khinchine formula. If $(X_t)_{t\geq 0}$ is a Lévy process, then its characteristic function is given by

$$\mathbb{E}[e^{i\theta X_t}] = \exp(ait\theta - \frac{1}{2}\sigma^2 t\theta^2 + t\int_{\mathbb{R}\setminus\{0\}} (e^{i\theta x} - 1 - i\theta x \mathbb{1}_{|x|<1})\nu(dx))$$

with $a \in \mathbb{R}$, $\sigma \leq 0$, $\mathbb{1}$ is the indicator function and ν is the Lévy measure of X such that

$$\int_{\mathbb{R}\backslash 0} \min\{x^2, 1\}\nu(dx) < \infty.$$

A Lévy process can be broken down into the sum of the following three processes

1. Brownian motion
2. Linear drift
3. Pure jump process

This decomposition of a Lévy process is known as the Lévy-Ito decomposition. For any Lévy triplet (a, σ^2, ν) there are three independent Lévy process in the same probability space, $X^{(1)}, X^{(2)}, X^{(3)}$ so that $X^{(1)}$ is Brownian motion with drift a and diffusion σ^2, $X^{(2)}$ is a compound Poisson process, that corresponds to the pure point part of the measure ν and $X^{(3)}$ is a pure jump martingale that corresponds to the continuous part of the singular measure ν. Therefore, a Lévy process X is decomposed as follows:

$X = X^{(1)} + X^{(2)} + X^{(3)}$ is a Lévy process with Lévy triplet (a, σ^2, ν).

1.5 OVERVIEW

In this chapter, we have defined Lévy processes and discussed the background and development of option pricing methods.

In Chapter 2, we will look at three standard option pricing models: The Black-Scholes model, the Merton jump diffusion model and Kou's double exponential jump diffusion model. The Black-Scholes model has been the most popular since its inception. Over time it has been shown that the Black-Scholes model can be improved upon by using a Lévy process to model option prices. Both the Merton jump diffusion and Kou's double exponential jump diffusion model improve upon the Black-Scholes model by adding discretized jumps to the underlying Brownian motion.

In Chapter 3, we will price real world Google stock options using each of the three models. The option data was purchased online via DeltaNeutral.com; it includes Google option data from January 7, 2011 to December 31, 2012. For each day of open trading there are numerous different options with their own strike price, expiration...etc. In total we will have 650,000 different Google options over this two years period to apply the models to. Software that will be used for the calculations will include Microsoft Excel, MATLAB and Mathematica.

In Chapter 4, we will compare and discuss the results. We will look at some standard global measures of fit in order to determine which of our three models is the most accurate.

In Chapter 5, we will wrap up with a conclusion.

CHAPTER 2
METHODOLOGY

There are many ways to price options; binomial tree pricing, Monte Carlo simulations and risk neutral valuation models. The mathematics behind calculating option prices can range from simple to quite complex. There are many different methods and models that can be used to price options. In this thesis, we will focus on three standard risk neutral pricing models: Black-Scholes, Merton Jump Diffusion and Kou's Double Exponential Jump model. Each of the three methods is increasingly complex and will require more processing power and time. We will explore whether the added complexity and time needed to price the options results in a more accurate model.

2.1 BLACK-SCHOLES MODEL

The Black-Scholes model gives investors a method to quickly estimate the price of European options. It was first published in 1973 by Fischer Black and Myron Scholes. The model is still widely used today, due to its ease of implementation and tractability.

The Black-Scholes model has been popular amongst investors looking to calculate the price of European put and call options. The Black-Scholes model requires a number of assumptions;

1. The underlying asset price is strictly continuous.
2. There are no arbitrage opportunities.
3. The risk-free rate of interest, r , is constant for all maturities.

4. There are no transactions costs or taxes.
5. There are no dividends during the life of the derivative.
6. The stock price distribution at maturity of the option is lognormal.

The stock price process we are utilizing is $dS_t = \mu S_t dt + \sigma S_t dW_t$, where μ is the expected rate of return, and σ is the volatility of the stock price, and W_t is a standard Brownian motion. If we define a function f to be the price of an option dependent on S and t, we can then apply Itô's lemma to obtain the following equation:

$$df = \left(\frac{\partial f}{\partial S}\mu S + \frac{\partial f}{\partial t} + \frac{1}{2}\frac{\partial^2 f}{\partial S^2}\sigma^2 S^2\right) dt + \frac{\partial f}{\partial S}\sigma S dW$$

Black and Scholes showed that a portfolio, Π composed of an option and its underlying stock only depends on t. In other words, the portfolio is riskless during time dt. Therefore Π must earn the same rate of return as the risk-free interest rate. This led to the Black-Scholes differential equation:

$$\frac{\partial f}{\partial t} + rS\frac{\partial f}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 f}{\partial S^2} = rf$$

The Black-Scholes differential equation can provide pricing for many different derivatives, depending on what the underlying asset S is. The derivative that is gotten when the equation is solved relies on what boundary conditions are used. For our purposes, we will be looking at European style call and put options with the following boundary values. Let S_0 and S_T be the stock prices at time $t = 0$ and $t = T$, respectively, where T is the time to maturity. The boundary for a European call option is $f = \max(S_T - K, 0)$, when $t = T$, while the boundary value for a European put option is $f = \max(K - S_T, 0)$ when $t = T$ (Hull, J.C. 2009).

One of the key aspects of the Black-Scholes differential equation is that it is risk-neutral. This means that it does not include any variables that are affected by the preferences of investors. If the equation included μ, the expected rate of return, then it would no longer be risk-neutral. This is because μ depends on the preferences of investors. Since risk preferences do not enter the equation, they cannot affect its solution. This simplifies matters greatly, and the assumption that all investors are risk neutral can be made. Using risk-neutral valuation will be a three step process;

1. Let the expected return be equal to the risk free rate, $\mu = r$.
2. Calculate the payoff from the option.
3. Discount the expected payoff using the risk free rate.

Applying this process, we can obtain the Black-Scholes call pricing formula;

$$C = S\Phi(d_1) - \Phi(d_2)Ke^{-rT}$$

, with $d_1 = \frac{ln(S_0/K)+(r+\sigma^2/2)T}{\sigma\sqrt{T}}$, and $d_2 = d_1 - \sigma\sqrt{T}$, where C is the call option's value, S_0 is the underlying asset's price, K is the strike price, r is the risk-free rate, T is the time to maturity (in years) and σ is the volatility of returns of the underlying asset.

Below is the pricing formula for a corresponding put option based on put-call parity;

$$P = N\Phi(-d_2)Ke^{(-rT)} - \Phi(-d_1)S_0.$$

The first part of the Black-Scholes formula for call options, $S\Phi(d_1)$ determines the expected gain from purchasing a stock outright. The second part of the formula, $Ke^{-rT}\Phi(d_2)$ calculates the present value of paying the exercise price on the expiration date. The value of the call option is calculated by taking the difference between these two parts.

2.2 MERTON JUMP DIFFUSION

The first Lévy process we will explore is the Merton jump diffusion (Merton, 1976). Merton built upon the Black-Scholes model by adding jumps, which occur according to a Poisson process. This would prove to be useful for pricing options whose underlying asset prices are not purely continuous. By adding this Poisson jump process to the underlying Brownian motion in the Black-Scholes model, Merton showed that his model was more accurate.

The Merton jump diffusion model is a particular Lévy model in which there are a finite number of jumps. The stochastic differential equation for the stock price is given as

$$dS_t = \mu S_t dt + \sigma S_t dW_t + S_t dJ_t,$$

where W_t is Brownian motion and $J_t = \sum_{i-1}^{N_t} Y_i$ is a Poisson process with jumps of size Y_i being identically distributed. Therefore the asset price follows Brownian motion between jumps. The stochastic differential equation has the solution

$$S_t = S_0 \exp(\mu t + \sigma W_t - \sigma^2 t/2 + J_t).$$

Merton explored the case where the jump sizes Y_i follow the normal distribution. If the forward price $F = S_0 e^{rT}$, and if $x_t = log(S_t/F)$ is a Lévy process, then it has a characteristic function $\phi_t(u) = \mathbb{E}[e^{iux_t}]$ with the following Lévy-Khintchine form

$$\phi_t(u) = \exp[iu(\mu - \sigma^2/2)t - \frac{u^2\sigma^2}{2}t + t\int [e^{iu\xi} - 1]\nu(\xi)d\xi].$$

Merton's model makes the assumption that the distribution of jump sizes is normal, and the Lévy density $\nu(\cdot)$ is $\nu(\xi) = \frac{\lambda}{\sqrt{2\pi}\delta}\exp[-\frac{(\xi-\alpha)^2}{2\delta^2}]$ with α being the mean of the log-jump size, and δ the standard deviation of the jumps.

This leads to the explicit characteristic function

$$\phi_t(u) = \exp[iu\omega t - \frac{u^2\sigma^2 t}{2} + \lambda t(e^{iu\alpha - \frac{u^2\delta^2}{2}} - 1)].$$

Using this, Merton determined the value $V(S,t)$ of a European style option satisfies the following:

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rV + \lambda \int F(d\xi)[V(Se^{\xi}, t) - V(S,t) - (e^{\xi} - 1)S\frac{\partial V}{\partial S}] = 0$$

Merton discovered a solution for the previous equation, for a European call option with strike K and length T, which is an infinite sum of the following form

$$C(S,K,T) = \sum_{n=0}^{\infty} \frac{e^{-\lambda T}(\lambda T)^n}{n!} \int F_n(d\xi) \times C_{BS}(Se^{\xi}e^{\mu_J T}, K, r, \sigma, T)$$

with $C_{BS}(\cdot)$ being the Black-Scholes call price.

If the jumps are normally distributed so that $Y_i \sim N(\alpha, \delta^2)$, then the formula becomes simpler

$$C(S,K,T) = \sum_{n=0}^{\infty} \frac{e^{-\lambda' T}(\lambda' T)^n}{n!} C_{BS}(S, K, r_n, \sigma_n, T),$$

where

$\lambda' = \lambda e^{\alpha + \delta^2/2}$,

$\sigma_n^2 T = \sigma^2 T + n\delta^2$,

$r_n T = (r + \mu)T + n(\alpha + \delta^2/2)$.

Each of the $C_{BS}(S, K, r_n, \sigma_n, T)$ terms in the normally distributed jump case is the conditional value of the call option given that there are n jumps during the option's life.

2.3 KOU'S DOUBLE EXPONENTIAL JUMP DIFFUSION

The second Lévy process we will discuss is Kou's double exponential jump diffusion (Kou, 2002). This model is considerably more complex than the previous two. The asset price S_t, under risk-neutral probability is modeled as follows:

$$\frac{dS_t}{S_{t-}} = \mu\, dt + \sigma\, dW_t + \, d(\sum_{i=1}^{N_t} e^{Y_i} - 1),$$

where W is a Brownian motion, N is a Poisson process with rate λ and μ and σ are drift and volatility of the diffusion. The jump sizes $(Y_1, Y_2, ...)$ are iid random variables with the same double exponential distribution with density:

$$f_Y(y) = p\eta_1 e^{-\eta_1 y} \mathbb{1}_{(y \geq 0)} + q\eta_2 e^{\eta_2 y} \mathbb{1}_{(y<0)}$$

with p and q non negative such that $p + q = 1$, $\eta_1 > 1$ and $\eta_2 > 0$. The processes W_t, N_t and the random variables $(Y_1, Y_2, ...)$ are independent.

In order to determine the European call and put prices under Kou's model we first need to define some unique functions:

$$Hh_{-1}(x) = e^{-\frac{x^2}{2}},$$

$$Hh_0(x) = \sqrt{2\pi}\Phi(-x),$$

$$Hh_n(x) = \int_x^{\infty} Hh_{n-1}(y)\, dy = \frac{1}{n!}\int_x^{\infty} (t-x)^n e^{-\frac{t^2}{2}}\, dt, \quad \forall n \geq 0.$$

$$I_n(c; \alpha, \beta, \gamma) = \int_c^{\infty} e^{\alpha x} Hh_n(\beta c - \gamma)\, dx \quad \forall n \geq -1.$$

with Φ being the standard normal cumulative distribution. Therefore we have the following:

$$nHh_n(x) = Hh_{n-2}(x) - xHh_{n-1}(x), \quad \forall n \geq 1$$

Then $\forall n \geq -1$:

$$I_n(c; \alpha, \beta, \gamma) = -\frac{e^{\alpha c}}{\alpha}\sum_{i=0}^{n}\left(\frac{\beta}{\alpha}\right)^{n-i} Hh_i(\beta c - \delta) + \left(\frac{\beta}{\alpha}\right)^{n+1}\frac{\sqrt{2\pi}}{\beta} e^{\frac{\alpha\delta}{\beta} + \frac{\alpha^2}{2\beta^2}}\Phi(-\beta c + \delta + \frac{\alpha}{\beta}),$$

when $\beta > 0$ and $\alpha \neq 0$. When $\beta < 0$ and $\alpha < 0$, we have the following form:

$$I_n(c;\alpha,\beta,\gamma) = -\frac{e^{\alpha c}}{\alpha}\sum_{i=0}^{n}\left(\frac{\beta}{\alpha}\right)^{n-i} Hh_i(\beta c-\delta) + \left(\frac{\beta}{\alpha}\right)^{n+1}\frac{\sqrt{2\pi}}{\beta}e^{\frac{\alpha\delta}{\beta}+\frac{\alpha^2}{2\beta^2}}\Phi(\beta c-\delta-\frac{\alpha}{\beta}).$$

Kou goes on to define the following for any probability P:

$$\Psi(\mu,\sigma,\lambda,p,\eta_1,\eta_2;a,T) = \mathbb{P}[Z_T \geq a]$$

where $Z_T = \mu t + \sigma W_t + \sum_{i=1}^{N_t} Y_i$ and Y has a double exponential distribution and N is a Poisson process with rate λ.

Kou shows that

$$\Psi(\mu,\sigma,\lambda,p,\eta_1,\eta_2;a,T) =$$

$$\frac{e^{(T\sigma\eta_1)}}{\sigma\sqrt{2\pi T}}\sum_{n=1}^{\infty}\pi_n\sum_{k=1}^{n}P_{n,k}(\sigma\sqrt{T}\eta_1)^k I_{k-1}(a-\mu T;-\eta_1,-\frac{1}{\sigma\sqrt{T}},-\sigma\sqrt{T}\eta_1)$$
$$+\frac{e^{(T\sigma\eta_2)}}{\sigma\sqrt{2\pi T}}\sum_{n=1}^{\infty}\pi_n\sum_{k=1}^{n}Q_{n,k}(\sigma\sqrt{T}\eta_2)^k I_{k-1}(a-\mu T;\eta_2,\frac{1}{\sigma\sqrt{T}},-\sigma\sqrt{T}\eta_2)$$
$$+\pi_0\Phi(-\frac{a-\mu T}{\sigma\sqrt{T}}),$$

where

$$P_{n,i} = \sum_{j=i}^{n-1} p^j q^{n-j}\binom{n-i-1}{j-i}\left(\frac{\eta_1}{\eta_1+\eta_2}\right)^{j-i}\left(\frac{\eta_2}{\eta_1+\eta_2}\right)^{n-j}, \quad 1 \leq i \leq n-1$$

$$Q_{n,i} = \sum_{j=i}^{n-1} q^j p^{n-j}\binom{n-i-1}{j-i}\left(\frac{\eta_2}{\eta_1+\eta_2}\right)^{j-i}\left(\frac{\eta_1}{\eta_1+\eta_2}\right)^{n-j}, \quad 1 \leq i \leq n-1$$

$$P_{n,n} = p^n, \quad Q_{n,n} = q^n, \quad \pi_n = \frac{e^{-\lambda T}\lambda^n}{n!}.$$

Therefore the price of a european call option is given by

$$S_0\Psi(r+\frac{1}{2}\sigma^2-\lambda\xi,\sigma,\tilde{\lambda},\tilde{p},\tilde{\eta_1},\tilde{\eta_2};\log\left(\frac{K}{S_0}\right),T)$$

$$-Ke^{-rT}\Psi(r-\frac{1}{2}\sigma^2-\lambda\xi,\sigma,\lambda,p,\eta_1,\eta_2;\log\left(\frac{K}{S_0}\right),T)$$

with

$$\tilde{p} = \frac{p}{1+\xi}\frac{\eta_1}{\eta_1 - 1}, \qquad \tilde{\lambda} = \lambda(1+\xi), \qquad \tilde{\eta_1} = \eta_1 - 1, \qquad \tilde{\eta_2} = \eta_2 + 1.$$

The corresponding European option's put price can then be easily calculated using put-call parity.

CHAPTER 3
APPLICATION:
PRICING GOOGLE STOCK OPTIONS

In the previous chapter, we have discussed three increasingly complex option pricing methods. In this chapter, we will valuate real world options. We will run three different simulations, one for the Black-Scholes model, and one each for our two Lévy models; Merton's Jump Diffusion and Kou's Double Exponential Jump Diffusion.

3.1 DATA PREPARATION

In this section, we plan to look at 2 years' worth of Google stock option data. The data was purchased online from www.deltaneutral.com, and includes data from the beginning of 2011 through the end of 2012. The data includes everything needed for simulation; underlying asset price, strike price, expiration date, quote date, as well as implied volatility and the risk free rate, and if the option is a call or a put. In order to prepare the data for use, we needed the length in years of each option, so a new column was created that computed length of each option by subtracting the expiration date from the quote date and converting into years. Each row of the spreadsheet represents an individual option with its own type, length and strike price for each day when the market was open. For any given day, there are many different strike prices and expiration lengths offered for options. In total, over our 2 year window we will be looking at just over 650,000 Google stock options and determining which model is most appropriate and accurate for evaluating option prices.

3.2 BLACK-SCHOLES MODEL

The first model to be examined is the Black-Scholes model. For this model, we need to input 5 parameters to determine the price of each option. From the raw data, we will use the following columns; length in years, whether the option is a call or a put, the underlying asset price, option strike price, and volatility. Below we have a snapshot of what our prepared data looks like in a spreadsheet.

Quote Date	**Length Years**	**Call/Put**	**Underlying Price**	**Strike Price**	**Volatility**
1/3/11	.010951	call	604.35	540	.01
1/3/11	.010951	call	604.35	550	.01
⋮	⋮	⋮	⋮	⋮	⋮
12/31/12	2.045171	put	707.38	990	.2556
12/31/12	2.045171	put	707.38	1000	.2556

Table 3.1: Snapshot of data for Black-Scholes model

The Black-Scholes option pricing model is evaluated in Microsoft Excel. First all options are priced as if they are all calls. Then, using put-call parity, it is easy to determine the put value of each option. When finished, it is then simple to split the options into calls and puts and compare the accuracy of the model. Since the data is in a spreadsheet with each row representing one particular option, it was straightforward to utilize Excel to compute all options at once by calling on the necessary column vectors in the Excel formula. Below is the code used to compute the call value of the Google options.

```
=(B2*(NORM.S.DIST(((LN(B2/K2))+
((0.03+(Q2*Q2/2))*J2))/(Q2*(SQRT(J2))),TRUE)))-
(K2*(EXP(0.03*J2*(-1)))*(NORM.S.DIST(((LN(B2/K2))+
((0.03+(Q2*Q2/2))*J2))/(Q2*(SQRT(J2)))-(Q2*(SQRT(J2))),TRUE)))
```

With the variables as follows; column B is the underlying asset price, column K is the strike price, column Q is the volatility, column J is the length of the option in years and 3% or .03 is the risk free rate for the time period in question. The standard method to obtain the value of the risk free rate is to use the 3-month Treasury Bill rate, which is right around 3%. The next step is to compute the corresponding put values of all the options, which is much simpler. The code used is below.

```
=K2*(EXP(-0.03*J2))-B2+Y2
```

With the variables as follows: column K is the strike price, column J is the length of the option in years, column B is the underlying asset price and column Y is the corresponding call value. It is simpler to compute all of the options as if they are calls and then apply put-call parity to price all of the options to determine the corresponding put values . After pricing the options as both, they can be split into calls and puts and evaluated. The Black-Scholes call value is fairly accurate when comparing it to the actual ask price of each call option. Please refer to figure A.4 and table A.1 in the appendix for more details.

3.3 MERTON JUMP DIFFUSION MODEL

The second model to be simulated is Merton's Jump Diffusion (Merton, R. 1976). For this simulation, MATLAB was used. First, all relevant data was imported as column vectors into MATLAB. A suitable MATLAB function was found called "calcMJDOptionPrice", that would return a 3-dimensional matrix, or option surface. Changes were made to the function so that the diagonal of this resulting option surface would be the MJD model's option price for each option. Due to the complexity of the model, and the fact the output of the function is a 2-dimensional matrix, calculating the MJD price of n options at a time would result in a matrix of $n \times n$ cells. Due to limits on processing power and RAM, roughly 3,000 Google options were priced at a time using this model until all had been priced. Below is the code used for this function. First the function "Norm Inverse" is defined so it can be called upon in the main function.

```
%% NORM INVERSE

function p = Nfcn(x)

p=0.5*(1.+erf(x./sqrt(2)));

end
```

Next is the main function "calcMJDOptionPrice";

```
function P = calcMJDOptionPrice(cp,S,K,T,sigma,r,q,lambda,a,b,n)

[K,T] = meshgrid(K,T);
[u,v] = size(K);
K = K(:,:,ones(1,1,n)); T = T(:,:,ones(1,1,n));
sigma = sigma(:,ones(1,v,1),ones(1,1,n)); % New Line
S = S(:,ones(1,v,1),ones(1,1,n)); % New Line
n = ones(1,1,n); n(:)=0:size(n,3)-1; factn = factorial(n);
n = n(ones(u,1),ones(1,v),:); factn = factn(ones(u,1),ones(1,v),:);

m = a+0.5*b.^2;
lambda_prime = lambda.*exp(m);
r_n = r - lambda*(exp(m)-1) + n.*(m)./T;
```

```
sigma_n = sqrt(sigma.^2 + (n*b^2)./T);


dfcn = @(z,sigma,r)((1./(sigma.*(T.^(0.5)))).*(log(S./K) +
   (r - q + z.*0.5*(sigma.^2)).*T));
callfcn = @(sigma,r)(((1./factn).*exp(-lambda_prime.*T).
   *(lambda_prime.*T).^n).*(exp(-q.*T).*S.*Nfcn(dfcn(1,sigma,r))
    - K.*exp(-r.*T).*Nfcn(dfcn(-1,sigma,r)))); % Call


P = sum(callfcn(sigma_n,r_n),3); % Eqn 19 of Merton, 1976


if cp==-1
    P = P - S.*exp(-q.*T(:,:,1)) +
    K(:,:,1).*exp(-r.*T(:,:,1)); % Convert Call to Put by Parity
end
end}
```

The function calls several input vectors, which are as follows;

cp is [1,-1] for call or put, S is the current asset price, K is the strike price, T is the time to maturity in years, *sigma* is the volatility of diffusion, r is the risk free rate, q is the dividend yield (0 in this case), *lambda* is the Poisson rate, a is the jump mean, b is he jump standard deviation and n is the event count.

The option data is imported to MATLAB and each column vector represents one of the variables; strike price, time to maturity etc. Since it is only possible to calculate 3,000 options at a time, a bit of data analysis is needed. Looking at the 3,000 options in question, we can determine the Poisson rate, the jump mean and the jump standard deviation from the underlying Google stock price over the same period. Historical data is available to determine the risk free rate at the issue date as well. The next step is to run the function to calculate the first 3,000 option values.

```
P = calcMJDOptionPrice(cp,S,K,T,sigma,r,q,lambda,a,b,n)
```

```
A=Diag(P)
```

This will create a 3,000 by 3,000 option surface called P. The model's best estimate of each individual stock price is the diagonal of this matrix, which is called A. These values are then added to the spreadsheet as estimates for Merton Jump Diffusion model. This process is then repeated until all of the options have been priced using Merton's model. The fixed parameters used for this model are as follows; $r = 0.03$, $q = 0$, $\lambda = 0.01$, $a = -0.2$, $b = 0.5$, and $n = 50$. The other parameters are taken directly from Google's option data. Please refer to figure A.5 and table A.2 in the appendix for more details.

3.4 KOU'S DOUBLE EXPONENTIAL JUMP MODEL

Kou's Double Exponential Jump Model is the final model that we will use to price the Google stock options (Kou, S.G. 2002). When Kou published this model in his 2002 work, "A Jump Diffusion Model for Option Pricing", he was kind enough to include mathematica code to apply the model to real world data. Therefore, we will use mathematica as well to price our Google option data. The model itself is quite complex, and the code is equally involved.

The main function of the code is named `callOR`. It will be the function used to return the call price of each option according to Kou's model. `callOR` calls on several other functions; phi, Hh, I function, Pni and Qni. First we will define them.

```
phi[x_] = (1 + Erf[x/Sqrt[2]])/2


Hh[n_, x_] := If[x >= -6 , If [x <10, 1/n!*NIntegrate[(t - x)^n*
      Exp[-t^2/2], {t, x, Infinity}], 0],
      (temp = (x + Sqrt[x*x +4*n])*0.5;
  ( NIntegrate[(t - x)^n*Exp[-t^2/2], {t, x, temp-3 }]+
  NIntegrate[(t - x)^n*Exp[-t^2/2], {t, temp-3 , temp-1 }]+
  NIntegrate[(t - x)^n*Exp[-t^2/2], {t, temp-1 , temp }]+
  NIntegrate[(t - x)^n*Exp[-t^2/2], {t, temp, temp+1 }]+
  NIntegrate[(t - x)^n*Exp[-t^2/2], {t, temp+1 , temp+3 }]+
  NIntegrate[(t - x)^n*Exp[-t^2/2], {t, temp+3, Infinity }]  )
   /(n!)  ) ]


II[jj_, ll_, aa_, bb_, dd_] :=
Which[ (bb >0 && aa !=0 ),
```

```
 - (Exp[aa*ll]/aa) * (Table[(bb/aa)^(jj-i), {i, 0, jj}] .
 Table[Hh[i, bb*ll-dd], {i, 0, jj}]) +
  ((bb /aa)^(jj+1) )*(Sqrt[2*Pi]/bb)
  *Exp[aa*dd/bb + (1/2) * (aa/bb)^2 ] *
  phi[-bb*ll + dd + aa/bb],
 (bb<0 && aa <0),
 - (Exp[aa*ll]/aa) *(Table[(bb/aa)^(jj-i), {i, 0, jj}] .
 Table[Hh[i, bb*ll-dd], {i, 0, jj}]) -
  ((bb /aa)^(jj+1) )*(Sqrt[2*Pi]/bb)
  *Exp[aa*dd/bb + (1/2) * (aa/bb)^2 ] *
  phi[bb*ll - dd - aa/bb],
 (bb >0 && aa ==0),
  Hh[n+1, bb*ll -dd]/bb
  ]

Pni[n_, i_, p_, eta1_, eta2_] := Sum[ Binomial[n, j]
       *( p^j )* ((1-p)^(n-j) )*Binomial[n-i-1, j-i]*
       ( (eta1/(eta1 +eta2))^(j-i) ) *
        ( (eta2/(eta1 + eta2))^(n-j)),
      {j, i, n-1}] /; i<n

Qni[n_, i_, p_, eta1_, eta2_] := Sum[ Binomial[n, j]
*( (1-p)^j )* ( p^(n-j) )*Binomial[n-i-1, j-i]*
       ( (eta2 /(eta1 + eta2) )^(j-i) ) *
       ( (eta1 /(eta1 + eta2))^(n-j)),
       {j, i, n-1}] /; i<n
```

The following function calculates the call price of a non-stock option, and will be used in determining the price for stock options later.

```
cprob[mu_, eta1_, eta2_, la_, p_, sig_, aa_,  bigT_, nStep_] :=
 ( IITwo = Table[  II[k-1, aa - mu * bigT, -eta1, -1/(sig*Sqrt[bigT]),
  -(sig*Sqrt[bigT])*eta1],
                  {k, 1, nStep}];


  IIFour = Table[ II[k-1, aa - mu * bigT, eta2, 1/(sig*Sqrt[bigT]),
   -(sig*Sqrt[bigT])*eta2],
                    {k, 1, nStep}];


   PiN[n_] = Exp[-la*bigT]*((la*bigT)^n) /(n!);


  PiNPni = Table [PiN[n] * Pni[n, k, p, eta1, eta2] *
   ((sig*Sqrt[bigT]*eta1)^k)
                  , {n, 1, nStep}, {k, 1, n}];
  PiNQni = Table [PiN[n] * Qni[n, k, p, eta1, eta2] *
   ((sig*Sqrt[bigT]*eta2)^k)
                   , {n, 1, nStep}, {k, 1, n}];



   sec = Sum[PiNPni[[n, k]] * IITwo[[k]],
   {n, 1, nStep}, {k, 1, n}];


  fourth = Sum[PiNQni [[n, k]] * IIFour[[k]],
```

```
  {n, 1, nStep}, {k, 1, n}];


   (sec * Exp[((sig*eta1)^2)*bigT/2] + fourth *
   Exp[((sig*eta2)^2)*bigT/2]) /( Sqrt[2*Pi] * sig * Sqrt[bigT] ) +
    Exp[- la* bigT] * phi [-(aa- mu*bigT)/(sig*Sqrt[bigT])]
 )
```

Finally we can define `callOR`, which will use Kou's model to return the call price of each option.

```
callOR[eta1_, eta2_, la_, p_, sig_, rr_, bigS_,
    bigK_,  bigT_, nStep_] :=
(
      zetaaOR = p*eta1 /(eta1 -1) + (1-p)*eta2/(eta2+1) -1;
      tempaa1OR =  rr + sig*sig/2 - la*zetaaOR;
      tempaa2OR = tempaa1OR - sig*sig;


    bigS *
 cprob[tempaa1OR, eta1 - 1, eta2 + 1, la *(1+zetaaOR),
  p*eta1/((1+zetaaOR)*(eta1-1)), sig, Log[bigK/bigS],  bigT, nStep] -
  bigK * Exp[- rr*bigT] *
  cprob[tempaa2OR, eta1, eta2, la, p, sig, Log[bigK/bigS],
    bigT, nStep]


)
```

The function "callOR" has 10 parameters which need to be determined from our option data. $Eta1$ and $eta2$ represent jump size, with the following relationship; an upward jump will have a magnitude of $(\frac{1}{\eta_1 - 1}) * 100\%$, while a downward jump will have a magnitude of $(\frac{-1}{\eta_2 + 1}) * 100\%$. Parameter la represents λ and is the average number of jumps in one period. P is the probability of an upward jump, note $0 \leq p \leq 1$. Sig is σ, the implied volatility. The rate of return is r. The underlying asset price is $bigS$, the strike price is $bigK$, the time (in years) of an option's life is $bigT$. The calculation involves infinite series, and $nStep$ declares how many terms to include in the calculation. It has been shown that anywhere from 10 to 15 steps will provide enough accuracy for most applications. The fixed parameters used for this model are as follows; $\eta_1 = 15$, $\eta_2 = 20$, $\lambda = 1$, $p = 0.6$ and $r = 0.03$. The other parameters are obtained from Google's option data.

The final step is to again use put-call parity to determine the corresponding put values of each Google stock option. Please refer to figure A.6 and table A.3 in the appendix for more details.

CHAPTER 4

RESULTS

So far, we have discussed three models for pricing European style options. We have compared the "tried and true" Black-Scholes method to two newer Lévy models; the Merton Jump Diffusion and Kou's Double Exponential Jump Diffusion. In this chapter we will discuss the results obtained from each model.

Since our data set is so large, we will need to take a random sample to work with. Our GOOG option data has over 650,000 different options. In order to assess which model is the most accurate, a sample of around 200 options would be ideal. In order to accomplish this using Excel, we can create a new column and use `rand()` to randomly assign a value between 0 and 1 for each cell in this new column. Then we can sort by this column and select the first 200 options from the top of the spreadsheet and we now have a random sample that is of a manageable size.

Calibration charts and results for each model are listed in figures A.4, A.5 and A.6 in the appendix. The goal here was to plot the real world strike price versus option price as circles on a chart, and the models' price as x's on the same graph. The fixed parameters were tweaked and experimented with until the difference between the market prices and model prices were minimized in terms of least-squares. As the model's grew more complicated and had more parameters, it became more time consuming to nail down accurate parameters. However, we were able to calibrate each model correctly and obtained accurate model prices.

There are several standard global measures of fit that we will use to determine which model is most accurate. They are the root mean square error (*rmse*), the average absolute error as a percentage of the mean price (*ape*), the average absolute error (*aae*) and the average relative percentage error (*arpe*). Below are the formulas for calculating each of these four measures of fit.

$$rmse = \sqrt{\sum_{options} \frac{(\text{Market price - Model Price})^2}{\text{number of options}}}$$

$$ape = \frac{1}{\text{mean option price}} \sum_{options} \frac{|\text{Market price - Model price}|}{\text{number of options}}$$

$$aae = \sum_{options} \frac{|\text{Market price - Model price}|}{\text{number of options}}$$

$$arpe = \frac{1}{\text{number of options}} \sum_{options} \frac{|\text{Market price - Model price}|}{\text{Market price}}$$

Our next step is to calculate each of these four measures of fit for each of our three models. This is straightforward to do with Microsoft Excel. We see from the results that Kou's model is clearly the best model of the three we examined. Kou's Double Exponential Jump Diffusion has the lowest values for each of the four measures used.

	BS	**MJD**	**KDEJD**
rmse	6.126633645	0.78423714	0.565239579
ape	0.037991633	0.00532576	0.00001695
aae	3.485423353	0.49253839	0.344834803
arpe	0.08736599	0.15233925	0.07820416

Table 4.1: Model accuracy

CHAPTER 5

CONCLUSION

Since its inception in 1973, the Black Scholes model has been widely used as a fast and accurate method of pricing options. Its tractability and ease of use have been its most attractive features. The assumption that Google's stock returns must follow a lognormal distribution poses some problems. It is clear from the QQ-plot below that Google's stock returns are not normally distributed, therefore violating one of the assumptions needed to use the Black-Scholes pricing model.

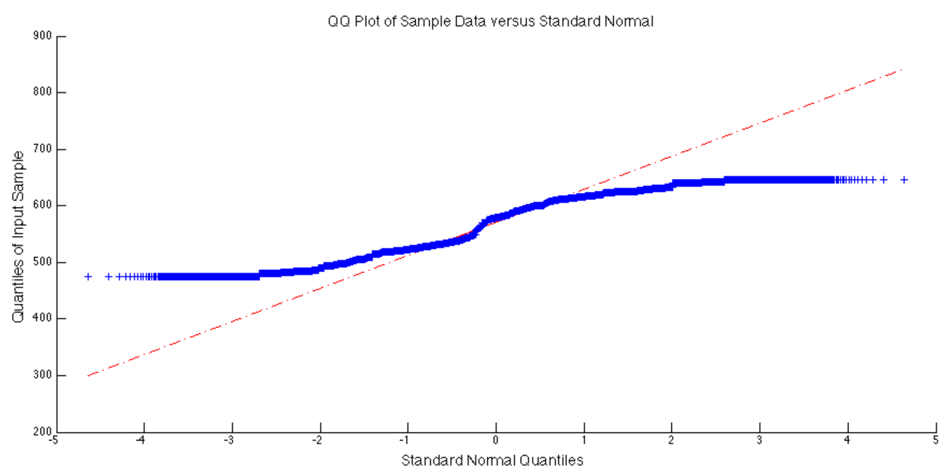

Figure 5.1: QQ-Plot of Google Stock price

Merton's Jump diffusion was the second model we examined. It made some improvements upon the standard Black Scholes model through introducing a discretized Poisson jump process to the underlying Brownian motion. This model was much more accurate than Black Scholes for our option pricing. This is due to the fact that this model was able to capture the jumps in the underlying stock price and use that to calculate the option price. This model is *slightly* less tractable than the Black Scholes model, but with MATLAB we were able to use the model with little trouble.

The final model examined was Kou's Double Exponential Jump diffusion. Kou's model includes parameters for the probability of an upward jump and probability of a downward jump. The jump sizes in Kou's model are iid random variables with the same double exponential distribution. There are a total of 10 parameters that are used when pricing an option with this model. These parameters capture: the rate of jumps in a period, the probability of an upward/downward jump as well as the mean upward and downward jump magnitudes. Due to the complexity of this model, it was the least tractable of the models we explored. Even so, with software like mathematica and modern processing technology we were able to price options reasonably quickly, and with an incredibly high level of accuracy.

APPENDIX A

FIGURES

A.1 MODEL PATH SIMULATIONS

A.1.1 Black Scholes

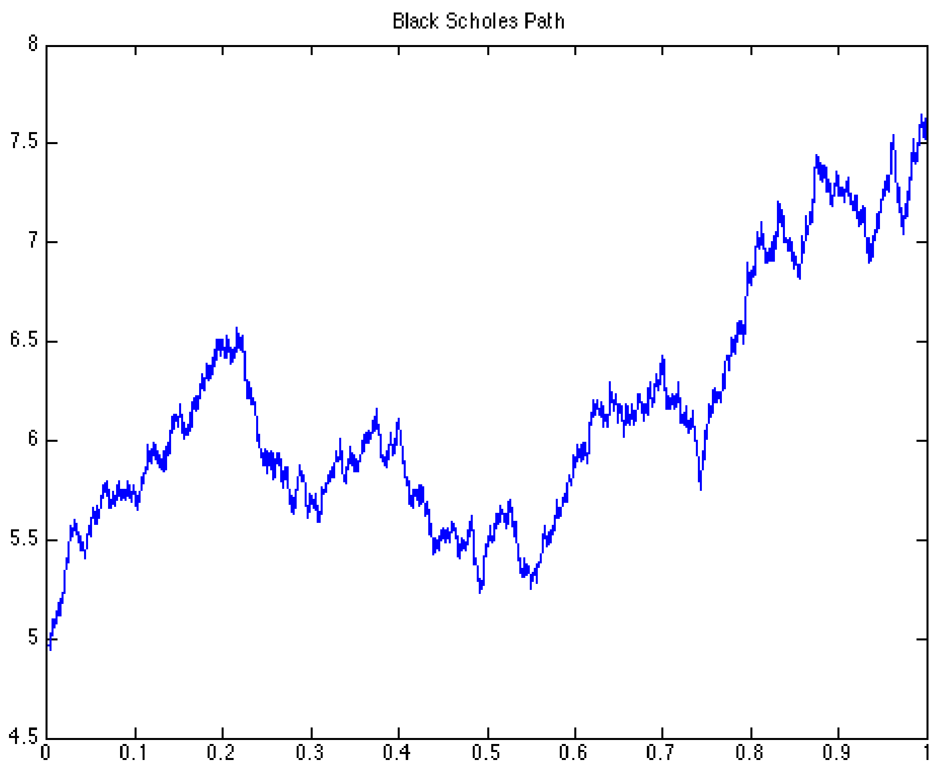

Figure A.1: Simulation of Black Scholes path

A.1.2 Merton's Jump Diffusion

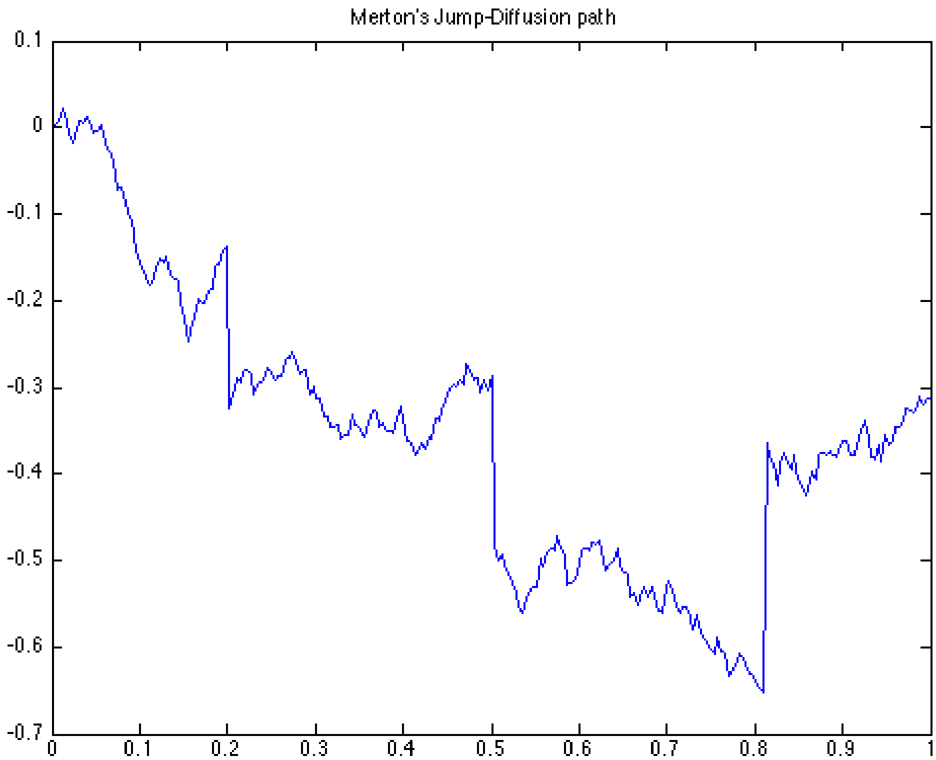

Figure A.2: Simulation of Merton Jump Diffusion path

A.1.3 Kou's Double Exponential Jump Diffusion

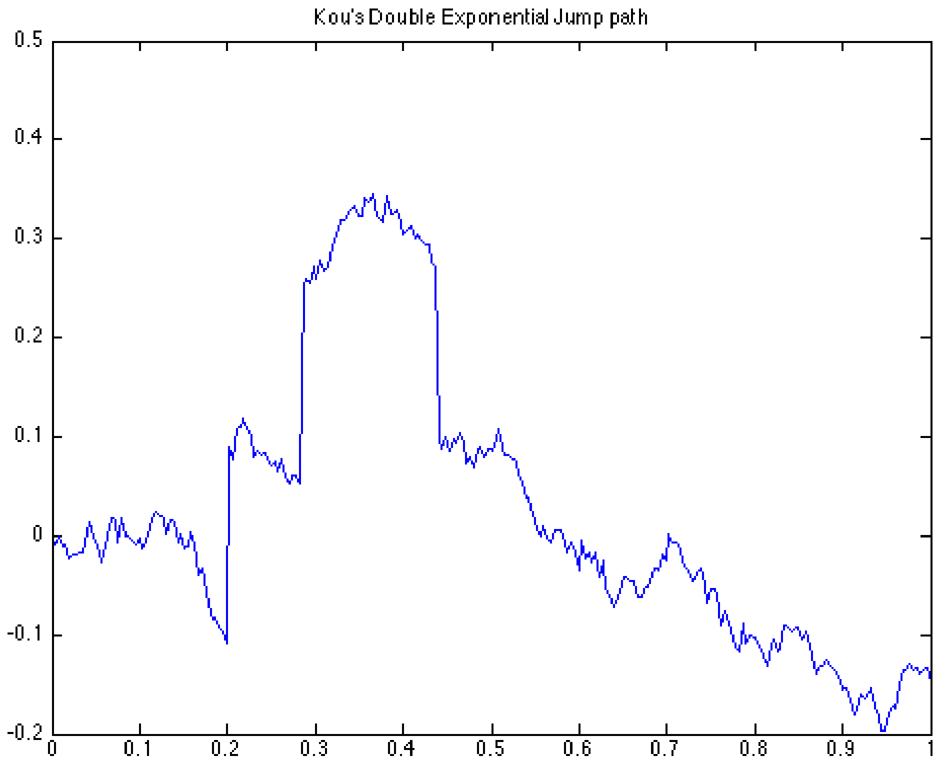

Figure A.3: Simulation of Kou's Double Exponential Jump Diffusion path

A.1.4 Black Scholes Calibration

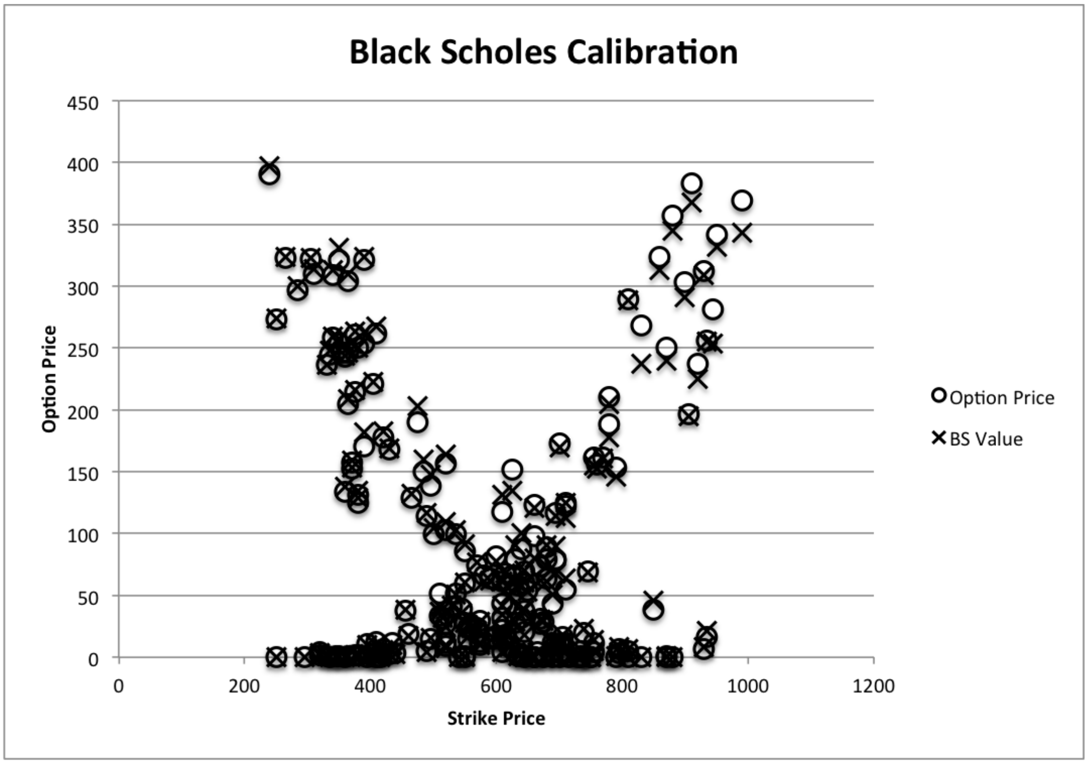

Figure A.4: 200 GOOG options "o" vs BS model price "x"

Underlying Price	Type	Length Years	Strike	Volatility	Option Price	Black Scholes Model Price
521.53	call	0.1642	730	0.2926	0.05	0.05659
597.50	call	0.2163	490	0.382	114.1	116.569
524.84	call	0.13689	700	0.355	0.4	0.43243
524.84	call	0.13689	370	0.6175	157.7	158.98
522.18	call	0.03012	250	2.1483	273.4	273.602
537.97	put	0.13963	660	0.3301	123.3	120.882
586.31	call	1.2347	485	0.3816	150.2	160.038
574.18	call	0.027378	635	0.3915	1	1.61048
539.08	put	0.147841	415	0.4928	3.4	3.24052
569.99	put	0.262834	625	0.2566	65.9	62.4326
542.56	call	0.00547	750	1.5507	0.05	0.05198
539.00	put	0.43258	460	0.376	19.2	17.6017
520.90	call	0.06023	550	0.1757	1.15	1.24456
630.37	put	0.0821	630	0.1974	4.3	4.23936
611.04	put	0.04928	770	0.6604	161.3	160.34
628.15	call	0.013689	380	2.3901	250.1	250.247
529.02	put	0.64339	910	0.3923	383.3	367.875
600.87	call	0.33128	700	0.2975	11.4	12.48
480.22	call	0.15879	455	0.3039	37.5	38.809
623.77	call	0.5284	795	0.267	7	8.09344

Table A.1: Sample of Black Scholes calibration data.

A.1.5 Merton Jump Diffusion Calibration

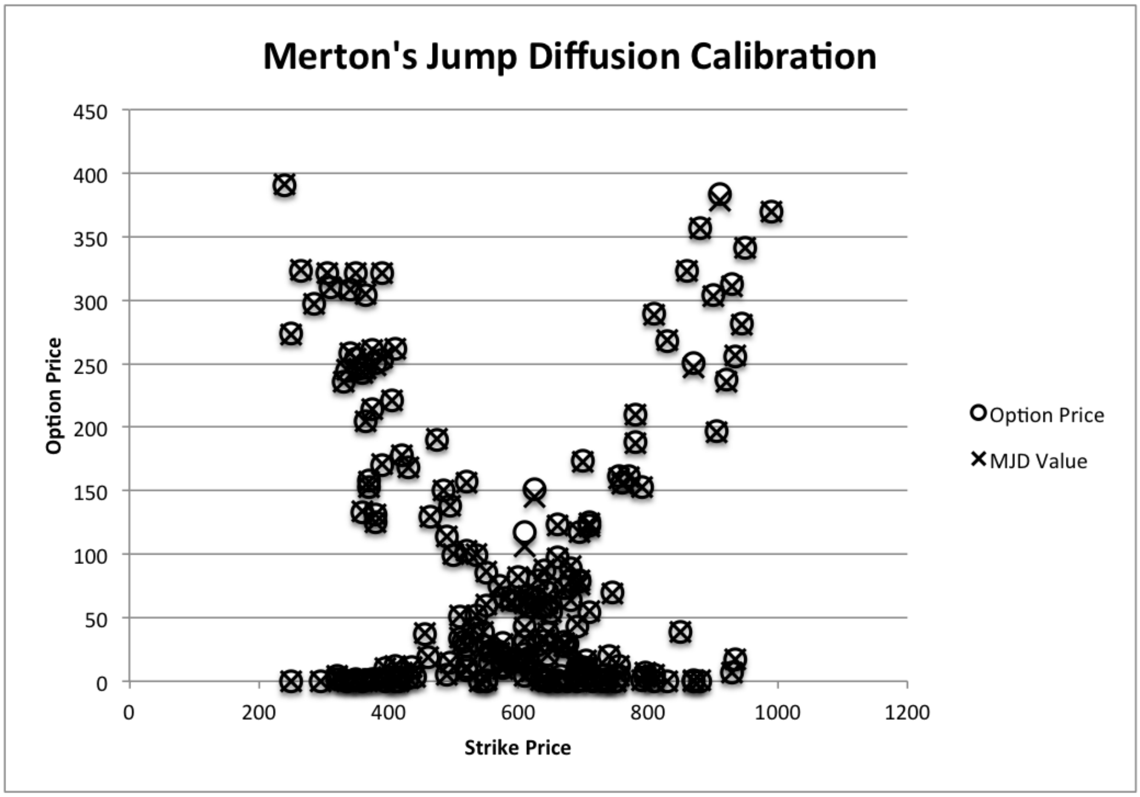

Figure A.5: 200 GOOG options "o" vs MJD model price "x"

Underlying Price	Type	Length Years	Strike	Volatility	Option Price	MJD Model Price
521.53	call	0.1642	730	0.2926	0.05	0.0417997
597.50	call	0.2163	490	0.382	114.1	114.285
524.84	call	0.13689	700	0.355	0.4	0.322262
524.84	call	0.13689	370	0.6175	157.7	157.48
522.18	call	0.03012	250	2.1483	273.4	273.061
537.97	put	0.13963	660	0.3301	123.3	122.44
586.31	call	1.2347	485	0.3816	150.2	150.371
574.18	call	0.027378	635	0.3915	1	0.847578
539.08	put	0.147841	415	0.4928	3.4	2.8299
569.99	put	0.262834	625	0.2566	65.9	64.9545
542.56	call	0.00547	750	1.5507	0.05	0.029587
539.00	put	0.43258	460	0.376	19.2	18.943
520.90	call	0.06023	550	0.1757	1.15	1.3036
630.37	put	0.0821	630	0.1974	4.3	4.271
611.04	put	0.04928	770	0.6604	161.3	160.508
628.15	call	0.013689	380	2.3901	250.1	248.576
529.02	put	0.64339	910	0.3923	383.3	378.35
600.87	call	0.33128	700	0.2975	11.4	11.2714
480.22	call	0.15879	455	0.3039	37.5	37.7112
623.77	call	0.5284	795	0.267	7	7.1804

Table A.2: Sample of Merton's Jump Diffusion calibration data.

A.1.6 Kou's Double Exponential Jump Diffusion Calibration

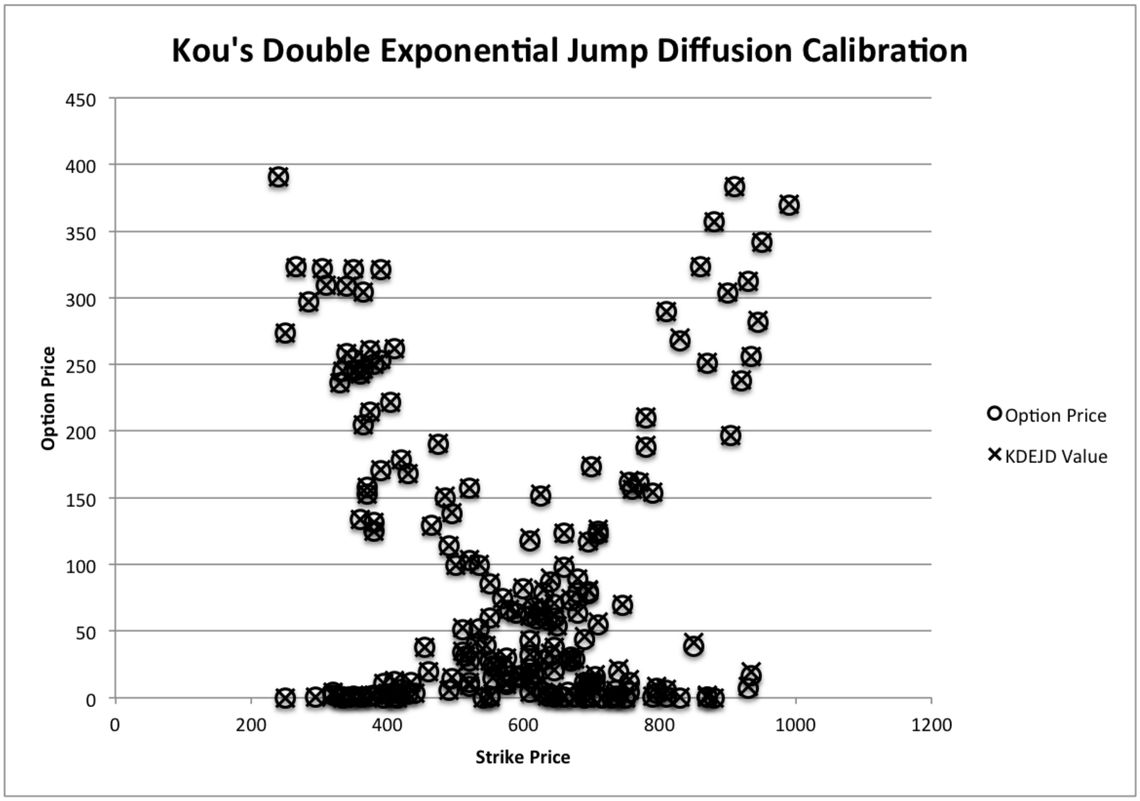

Figure A.6: 200 GOOG options "o" vs KDEJD model price "x"

Underlying Price	Type	Length Years	Strike	Volatility	Option Price	KDEJD Model Price
521.53	call	0.1642	730	0.2926	0.05	0.35763
597.50	call	0.2163	490	0.382	114.1	114.228
524.84	call	0.13689	700	0.355	0.4	0.68749
524.84	call	0.13689	370	0.6175	157.7	157.745
522.18	call	0.03012	250	2.1483	273.4	273.4072
537.97	put	0.13963	660	0.3301	123.3	123.596
586.31	call	1.2347	485	0.3816	150.2	151.176
574.18	call	0.027378	635	0.3915	1	1.1017
539.08	put	0.147841	415	0.4928	3.4	3.43548
569.99	put	0.262834	625	0.2566	65.9	66.4392
542.56	call	0.00547	750	1.5507	0.05	0.06278
539.00	put	0.43258	460	0.376	19.2	19.4504
520.90	call	0.06023	550	0.1757	1.15	1.3308
630.37	put	0.0821	630	0.1974	4.3	4.32475
611.04	put	0.04928	770	0.6604	161.3	161.44
628.15	call	0.013689	380	2.3901	250.1	250.115
529.02	put	0.64339	910	0.3923	383.3	383.741
600.87	call	0.33128	700	0.2975	11.4	12.1886
480.22	call	0.15879	455	0.3039	37.5	37.675
623.77	call	0.5284	795	0.267	7	8.2352

Table A.3: Sample of Kou's Double Exponential Jump Diffusion calibration data.

Underlying Price	Type	Expiration	Quote Date	Strike	Last	Bid	Ask	Volume	Volatility
604.35	call	1/7/11	1/3/11	540	60.1	62.6	64.3	24	0.01
604.35	call	1/7/11	1/3/11	550	53.9	53.7	54.3	44	0.01
604.35	call	1/7/11	1/3/11	560	44.2	43.8	44.3	54	0.01
604.35	call	1/7/11	1/3/11	570	33.7	33.9	34.2	132	0.01
604.35	call	1/7/11	1/3/11	580	24.2	24.1	24.5	350	0.208
604.35	call	1/7/11	1/3/11	590	15	14.7	15	684	0.191
604.35	call	1/7/11	1/3/11	600	7.4	7	7.4	3557	0.198
604.35	call	1/7/11	1/3/11	610	2.6	2.55	2.7	5023	0.2008
604.35	call	1/7/11	1/3/11	620	0.85	0.75	0.85	1236	0.2156
604.35	call	1/7/11	1/3/11	630	0.35	0.25	0.35	828	0.2483
604.35	call	1/7/11	1/3/11	640	0.2	0.15	0.2	464	0.2903
604.35	call	1/7/11	1/3/11	650	0.1	0	0.1	10	0.32
604.35	call	1/7/11	1/3/11	660	0	0	0.05	0	0.3458
604.35	put	1/7/11	1/3/11	540	0.1	0.1	0.15	249	.4992
604.35	put	1/7/11	1/3/11	550	0.15	0.15	0.2	92	.4466
604.35	put	1/7/11	1/3/11	560	0.2	0.15	0.25	168	.3866
604.35	put	1/7/11	1/3/11	570	0.25	0.2	0.3	770	.3206
604.35	put	1/7/11	1/3/11	580	0.45	0.35	0.45	1316	.2624
604.35	put	1/7/11	1/3/11	590	1.1	1	1.1	2478	.2265
604.35	put	1/7/11	1/3/11	600	3.3	3.2	3.4	2277	.2137
604.35	put	1/7/11	1/3/11	610	8.9	8.5	8.8	466	.2213
604.35	put	1/7/11	1/3/11	620	17	16.7	17.2	128	.2641
604.35	put	1/7/11	1/3/11	630	28.6	26.2	27.8	20	.4044
604.35	put	1/7/11	1/3/11	640	36.7	36.1	36.5	58	.3894
604.35	put	1/7/11	1/3/11	650	47.8	46	46.4	66	.4547
604.35	put	1/7/11	1/3/11	660	0	55.2	58	0	.6922

Table A.4: Snapshot of one day's raw data of Google stock options

BIBLIOGRAPHY

[1] Applebaum, D. (2004). *Lévy processes and stochastic calculus.* Cambridge, UK : Cambridge University Press.

[2] Black, F. and M. Scholes (1973). The pricing of options and corporate liabilities. J. Polit. Econ. 81, 637-654.

[3] Hull, J. C. (2009). *Options, futures, and other derivatives.* Upper Saddle River, NJ : Pearson Education, Inc.

[4] Kou, S. G. (2002). A Jump-Diffusion Model for Option Pricing Management Science, Vol. 48, No. 8, 1086-1101.

[5] Kyprianou, Andreas, & Wilmott, Paul. (2005). Exotic Option Pricing and Advanced Lévy Models. West Sussex, UK : John Wiley & Sons.

[6] Macbeth, James D. (1979). An Empirical Examination of the Black-Scholes Call Option Pricing Model *The Journal of Finance Vol. 34,* 1173-1186.

[7] Merton, R. C. (1976). Option pricing with discontinuous returns. *Bell J. Financial Economics 3,* 145-166.

[8] Sato, K. I. (1999). *Lévy processes and infinitely divisible distributions.* Cambridge, UK : Cambridge University Press.

[9] Wu, Liuren (2008). Modeling Financial Security Returns Using Lévy Processes An overview of Lévy processes in finance. Birge, J.R., Linetsky, V. (Eds.), Handbooks in Operations Research and Management Science, Vol. 15, 117-162.