Option Pricing: Lévy Process

By

Lan Mi

Alessandro Veneziani

Adviser

Applied Mathematics

Alessandro Veneziani

Adviser

Nicholas Valerio III

Committee Member

James Nagy

Committee Member

2013

Option Pricing: Lévy Process

By

Lan Mi

Alessandro Veneziani

Adviser

An abstract of

a thesis submitted to the Faculty of Emory College of Arts and Sciences

of Emory University in partial fulfillment

of the requirements of the degree of

Bachelor of Sciences with Honors

Mathematics

2013

Abstract

Option Pricing: Lévy Process

By Lan Mi

We present an introduction and implementation of exponential Lévy model for option pricing. First of all, we provide a short introduction on two of most popular option pricing models, Binomal Price Tree and the celebrated Black-Scholes Model. Through the introduction, we also illustrate several mathematical and numerical concepts related to solve our Lévy model. Then, we demonstrate the idea to use Lévy process to approximate financial market movements and to estimate option prices. Finally, we develop an explicit-implicit finite difference scheme for solving the exponential Lévy process, which has a parabolic partial integro-differential equation (PIDE) with jump-diffusion process. We implement this scheme based on European put option, and we calculate the call option price based on Put-Call Parity. We also compare the numerical results by our Lévy model and the Black-Scholes model.

Option Pricing: Lévy Process

By

Lan Mi

Alessandro Veneziani

Adviser

A thesis submitted to the Faculty of Emory College of Arts and Sciences

of Emory University in partial fulfillment

of the requirements of the degree of

Bachelor of Sciences with Honors

Mathematics

2013

Contents

1 Introduction **1**

1.1 Outline . . . 2

1.2 Terminology . . . 2

2 Derivatives and Option Pricing **4**

2.1 European options . . . 4

2.2 The binomial price tree model . . . 5

2.3 The Black-Scholes model . . . 6

2.3.1 Brownian motion: Wiener process . . . 7

2.3.2 Itô Processes . . . 8

2.3.3 The Black-Scholes Equation . . . 10

3 Lévy Processes **13**

3.1 Basic facts . . . 14

3.1.1 Simulating Lévy processes . . . 14

3.1.2 Characteristic function of a Lévy process . . . 18

3.2 No-arbitrage condition . . . 19

3.3 Explicit-implicit finite difference scheme . . . 21

3.3.1 Truncate integral . . . 23

3.3.2 Divide Lu into explicit and implicit parts . . . 24

4 Numerical Approximation and Its Implementation **25**

4.1 Explicit-implicit finite difference scheme . . . 25

4.2 A sketch of algorithm . . . 26

4.3 MATLAB code . . . 26

4.4 Explanation of MATLAB code . . . 33

5 Numerical Results **34**

5.1 Different parameters and implied volatility errors 34

5.2 Results with comparison to the Black-Scholes Model 37

6 Conclusion **39**

7 Appendix **40**

7.1 Existence/uniqueness and errors: exp. Lévy model 40

7.1.1 Lipschitz Condition 40

7.1.2 Viscosity solutions 41

7.1.3 Bound on approximation error 42

7.2 Bound on localization error 44

8 Reference **45**

List of Figures

1 The Number of Call Contracts 1
2 Binomial Price Tree - Stock 5
3 Binomial Price Tree - Payoff of Call 6
4 Wiener Process Simulation 8
5 Poisson Process Simulation 15
6 Lévy Process with Merton's Model 1 16
7 Lévy Process with Merton's Model 2 16
8 Lévy Process with Merton's Model 3 17
9 Lévy Process with Merton's Model 4 17
10 Results with Changed A 35
11 Results with Changed M 36
12 Results with Changed N 36
13 Results by Lévy compared to the Black-Scholes 37

List of Tables

1 Fundamental Multiplication Rule of Wiener Process 9
2 Numerical Methods for Partial Integro-Differential Equations 21
3 Put Price Results with Changed Order of A 34

1 Introduction

First introduced in the eighteenth century, put and call options did not embrace a rapid growth in transactions until the 1970s with the exchange markets open to trade options. From the chart below, we can see that the market of options is still growing even since 2007 to 2011, except for a slightly decrease during the credit crisis. Actually, derivatives markets have become even more important at the time that the world was and is experiencing a severe credit crisis. It is critical to professionals in the financial services industry to understand how the derivatives markets work. Also, it is essential for academics to develop better applications to estimate large market movements and more accurate valuations of options.

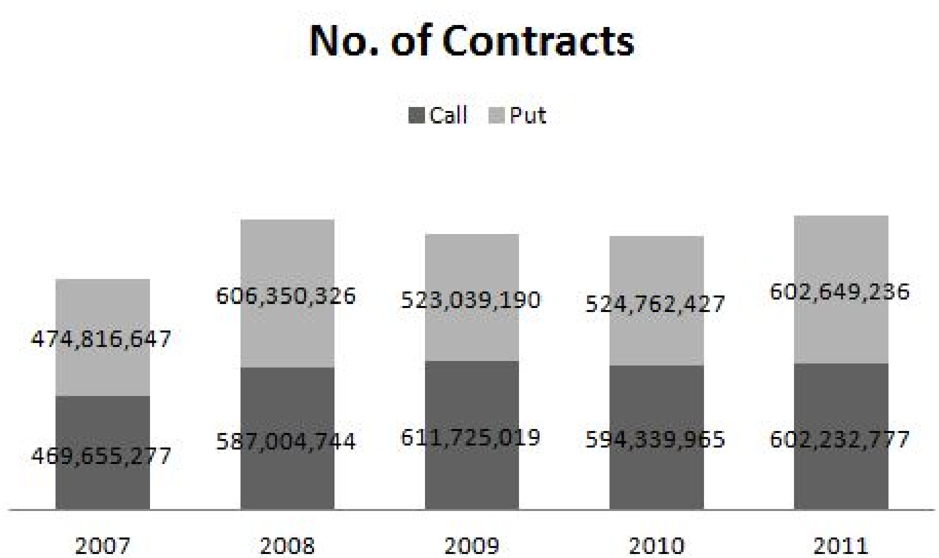

Figure 1: The Number of Call Contracts from the year 2007 to 2011 - *Data from the Chicago Board Options Exchange*

Achdou[1], Cont[5], Daffy[6] and Tankov[13] all have stressed that the exponential Lévy processes can give an improved approach to compute option prices. The celebrated Black-Scholes model assuumes that the probability distribution of the underlying asset price (here we use stocks) to be lognormal. If the market price moves apart from the lognormal distribution, the Black-Scholes model will under-estimate or over-estimate option prices. Models such as Lévy processes allow jumps in asset price movements, which are more realistically representing the price dynamics. Also, those models allow the users to choose different probability measures, such as the Merton diffusion model and variance Gamma models. This open choice tenders the flexibility to tailor our model in order to better approximate the market movements.

1.1 Outline

Section 2 introduces basic facts about derivatives, especially European options. Then, it investigates two option pricing models, with a concentration in the celebrated Black-Scholes model. Section 3 introduces the Lévy processes and discusses how we can use its properties to evaluate European options. In Section 3, we also introduce several mathematical and numerical concepts, such as viscosity solutions, used to determine the existence, uniqueness and accuracy of our explicit-implicit finite difference scheme to solve the Lévy model.

In Section 4, we continue our discussion on how to develop an explicit-implicit finite difference scheme. As well, we present a full MATLAB algorithm. Section 5 provides results on option pricing by our model, and gives a comparison to results by the Black-Scholes model.

1.2 Terminology

V = V(S,t): value function

C(S,t): value of call option

P(S,t): value of put option

K: exercise / strike price

S: current stock price

S_t: stock price at time t

S_T: stock price at maturity

T: expiry date

PV(*): present value of * $(= e^{-rt}*)$

σ: volatility of underlying asset

r: riskless rate

X_T: a stochastic process, given a probability space (Ω, A, P) and a measurable space (S, Σ), is a collection of random variables on Ω, indexed by a totally ordered set T(time), such that, $X_t : t \in T$. X_t is an S-valued random variable on Ω and S is called the state space of the process. A is a subset of Ω, and P is a non-negative probability measure on Ω such that $P(\Omega) = 1$.[1]

2 Derivatives and Option Pricing

In this section, we present a summary on the European options. Then, we investigate in two option pricing models: the Binomial Price Tree model and the Black-Scholes model. We focus on the evolution of the Black-Scholes equation. Through introducing these two models, we also illustrate definitions and properties to some concepts, such as Brownian process and Itô formula. Those concepts will also give us a hand in developing a finite difference scheme to solve the Lévy processes.

2.1 European options

Options are financial products. By definition[11], the European call and put options have payoffs depending on the price behavior of the underlying asset, oftentime stocks. The seller of such contracts is called the subscriber; the counterparty is called the holder.

At the drawing up time of the contract (time t = 0) and **exercise** or **strike price** K is fixed. Thus, at the **expiry date** $\boldsymbol{T}$,

1. under a call contract: the holder can exercise the contract and purchase the asset at K, oftentime when $S_T > K$, so the holder can pay less than the market price;

2. under a put contract: the holder can exercise the contract and sell the asset at K, oftentime when $S_T < K$, so the holder can receive more than the market price.

Contrast to forward and future contracts, option contracts do not require the holders to exercise the option. Without assigning any obligation, ption contracts give to the holder a right to get a payoff: $(S_T - K)^+$ for a call contract and $(K - S_T)^+$ for a put contract. Thus, the holder has to pay extra price for option contracts.

An option pricing model is designed to find the "right" price of options based on fair approximation of its future payoffs. Those models are based on no-arbitrage principle, which means that, in the hypothesis of efficient market, if an arbitrage opportunity exists (opportunity to invest with cash inflow today and no cash outflow in the future), the market will correct it automatically immediately. We also need to know an important equalibrium: the European Put-Call Parity,

$$C - P = S - PV(K)$$

Later in this paper, we will use this equation to transfer the price between puts and calls. In the following sub-section, we will introduce two basic option pricing models and some concepts related to our exponential Lévy process model.

2.2 The binomial price tree model

The binomial price tree model involves dividing the life of the option into several subintervals. It begins with the current stock price and assumes that stock price changes only to two directions, either up or down movement. Option price is calculated backward, assuming at maturity, which is the end of this model, option only has intrinsic value ($(S-K)^+$ for call). And all the proceeds at each step are discounted back to the current stage. The major limitation on this model is that it is not sufficient to estimate large market movements, if the number of subintervals is not large enough. Also, it is difficult to know the correct discount rate to use for payoffs at each stage[14].

$t=0$	$t=1$	$t=2$
		u^2S
	uS	
S		udS
	dS	
		d^2S

Figure 2: Binomial Price Tree - Stock

This model is easy to implement and to use, especially for those underlying assets having discontinuous dividents. It is also fundamental model for many option pricing extensions, such as the Black-Scholes model.

$t=0$	$t=1$	$t=2$
		C_{uu}
	C_u	
C		C_{ud}
	C_d	
		C_{dd}

Figure 3: Binomial Price Tree - Payoff of Call

2.3 The Black-Scholes model

In the early 1970s, Fischer Black, Myron Scholes, and Robert Merton invented the most celebrated approach to evaluate options, the Black-Scholes model, also known as Black-Scholes-Merton model. Black and Scholes used the capital asset pricing model to calibrate the factors of stock price movements with time.

This model assumes the *market efficiency*[11]: (a) the market responds instantaneously to new information on the asset; (b) the price has no memory: its past history is fully stored in the present price, without further information.

These two properties imply that the stock movement dS has the following characteristics:
1. Stationary distribution: the probability distribution of dS does not change over time;
2. **Stochastic processes**: dS changes over time with uncertainty. Look up in *Terminology* for a rigorous definition;
3. **Markov property**: a stochastic process has the Markov property if it has the independence of the future process X_{t+s} from the past (absence of memory) when the present X_s is known and reflects the absence of memory of the random walk[11, for a rigorous definition]. The process is called a Markov process, such as Brownian motion;
4. Diffusion process: continuous-time stochastic process.

Thus, in the time interval from t to $t+dt$, the Black-Scholes model considers that S undergoes a change from S to $S+dS$. this model also assumes that the **return** dS/S is consisted of two terms: an average growth term with time and a stochastic term.
The average growth rate of S is measured by a drift μ: $\frac{dS}{S} = \mu dt$. From basic calculus, we know that $d(\log S) = \frac{dS}{S} = \mu dt$. Thus, S(t) has an exponential growth: $S(t) = S(0)e^{\mu t}$. To

include the random walks in stock price movements, the stochastic term contains an increment of a Brownian motion: σdB. The coefficient σ is called the **volatility** and measures the standard deviation of the return; this model assumes σ to be constant.

Now we have,

$$\tfrac{dS}{S} = \mu dt + \sigma dB.$$

In order to get a solution for the above system, we need to learn about Brownian motion (Wiener Processes) and Itô Processes. The following two sub-sections further explain these two processes.

2.3.1 Brownian motion: Wiener process

Definition: a stochastic process ȳ(t) has the following properties[14]:

1. $\bar{y}(t)$ has Gaussian normal distribution;

2. The covariance between $\bar{y}(t_i)$ and $\bar{y}(t_j)$ is zero, for $i \neq j$.

Wiener Process is also known as **standard Brownian motion**, a Brownian motion with Gaussian distribution. It is a particular type of Markov stochastic process, with mean at zero and standard deviation at 1. Its first property implies that for any two non-overlapping intervals, values of $\bar{y}(t)$ are independent. To transform this process from $N(0, 1)$ to $N(0, dt)$, we consider the change in a process B:

$$B(t + \Delta t) = B(t) + \bar{y}(t)\sqrt{\Delta t}$$

$$\Delta B(t) = B\sqrt{\Delta t}$$

$$If\ \Delta t \to 0, B = \bar{y}\sqrt{(dt)},$$

where $dB \sim N(0, dt)$, is an increment of a Brownian process[11].

In our case, the Black-Scholes model makes the random aspects as a stochastic contribution: σdB. Combining with the average growth rate term of S, we have,

$$d(\log S) = \frac{dS}{S} = \mu dt + \sigma dB,$$

which is a **stochastic differential equation**[11]. In order to integrate dS/S between 0 to t, we need to use **Itô formula**, a stochastic version of the chain rule[11]. The next sub-section will explain what Itô formula is, and how it can be applied to option pricing.

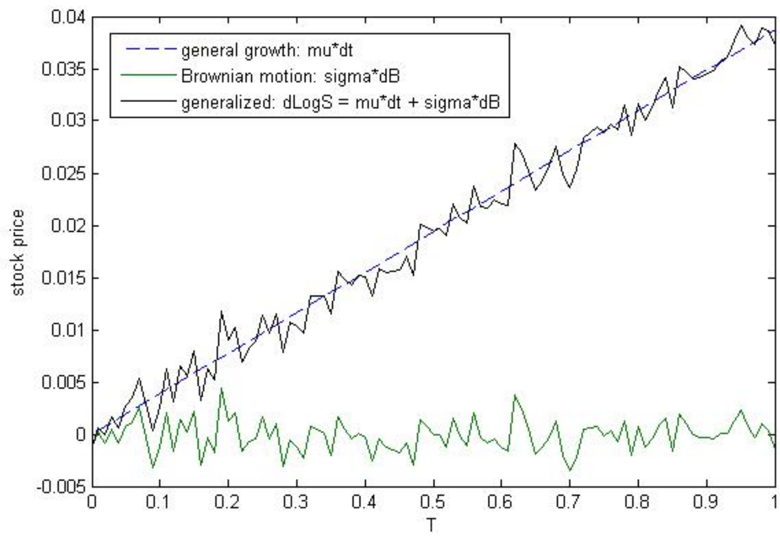

Figure 4: Wiener Process Simulation

To better understand these random movements, we build a generalized Wiener process with the form, $d\bar{x} = adt + \sigma d\bar{z}$. As shown in Figure 4, the generalized Wiener process goes up and down, with a dominating trend of rising.

2.3.2 Itô Processes

Definition: an Itô process X = X(t) is a solution of a stochastic differential equation of the type[11],

$$dX = a(X,t)dt + \sigma(X,t)dB$$

where a is the drift term and σ is the volatility coefficient.

Now, we consider a smooth function $F = F(x,t)$, which has the form:

$$dF = F_t dt + F_x dX = \{F_t + aF_x\}dt + \sigma F_x dB.$$

Numerically, we use the Taylor's formula to approximate the integration of F, letting $X(0) = X_0$,

$F(X,t) = F(X_0, 0) + F_t dt + F_x dX + \frac{1}{2}\{F_{xx}(dX)^2 + 2F_{xt}dXdt + F_{tt}(dt)^2\} + ...$

We need to find the differential of F along the trajectories of Itô formula.Thus, we only consider linear terms,

$$F_t dt + F_x dX = \{F_t + aF_x\}dt + \sigma F_x dB.$$

Excluding the non linear terms $2F_{xt}dXdt$ and $F_{tt}(dt)^2$, we now check the term $(dX)^2$),

$$(dX)^2 = [adt + \sigma dB]^2 = a^2(dt)^2 + 2a\sigma dBdt + \sigma^2(dB)^2 = \sigma^2 dt,$$

based on fundamental multiplication rule of Wiener process:

	dB	**dt**
dB	dt	0
dt	0	0

Table 1: Fundamental Multiplication Rule of Wiener Process[14], because $(dt)^2$ and $dBdt$ are non linear with respect to dt and dX[11].

Now, we have,

$$dF = \{F_t + aF_x + \frac{1}{2}\sigma^2 F_{xx}\}dt + \sigma F_x dB,$$

where:

$$F_t = \frac{\partial F}{\partial t}, F_x = \frac{\partial F}{\partial x}, F_{xx} = \frac{\partial^2 F}{\partial x^2}$$

This is called **Itô's Lemma**. We will use this equation later to derive the finite difference system for the Lévy process.

Let $F(S) = \log S$. Since $dS = \mu S dt + \sigma S dB$, and

$$F_t = 0, F_S = \frac{1}{S}, F_{SS} = -\frac{1}{S^2},$$

Thus, we have,

$$dF = d(\log S) = \frac{dS}{S} = \{0 + \mu\frac{1}{S} + \frac{1}{2}\sigma^2(-\frac{1}{S^2})\}dt + \sigma\frac{1}{S}dB$$
$$d\log S = (\mu - \frac{1}{2}\sigma^2)dt + \sigma dB.$$

After integrating between 0 and t, we have,

$$\log S_t = \log S_0 + (\mu - \tfrac{1}{2}\sigma^2)t + \sigma B(t).$$

Since μ and σ are constant, $Y = \log S$ must follow a generalized Wiener process, with a drift at $\mu - \frac{1}{2}\sigma^2$ and constant variance rate at σ^2. Therefore, Y has a normal distribution with mean $\log S_0 + (\mu - \frac{1}{2}\sigma^2)t$ and variance $\sigma^2 t$. Clearly, the density of S is a **lognormal density**.

2.3.3 The Black-Scholes Equation

The Black-Scholes Equation has the following hypothesis[11]:

1. S follows a lognormal law;
2. The volatility σ is constant and known;
3. There are no transaction costs or dividends;
4. It is possible to buy or sell any number of the underlying asset;
5. There is an interest rate r ¿ 0, for riskless investment;
6. The market is arbitrage free.

The fifth assumption implies that, at time $t = T$ from the prospect of $t = 0$, the value of $\$1 = e^{rT}$. The sixth assumption is important. Under the assumption of risk neutral and no-arbitrage markets, the value of an option should be matched with its synthetic portfolio, which is created by buying (for call) Δ shares of underlying stock and borrowing Π dollars at riskless rate,

$$V = S\Delta + \Pi, \longrightarrow \Pi = V - S\Delta,$$

where, $d\Pi = r\Pi dt$, due to no-arbitrage principle. Starting from the Itôs Lemma, we calculate the return of the evolution of $V(S,t)$, the **value function**. We have,

$$dS = \mu S dt + \sigma S d\bar{z},\ dV = V_t + \mu S V_s + 1/2\sigma^2 S^2 V_{ss} dt + \sigma S V_s dB.$$

And,

$$\begin{aligned} d\Pi &= dV - \Delta dS \\ &= \{V_t + \mu S V_S + \frac{1}{2}\sigma^2 S^2 V_{SS}\}dt + \sigma S V_S dB - \Delta(\mu S dt + \sigma S dB) \\ &= \{V_t + \mu S V_s + \frac{1}{2}\sigma^2 S^2 V_{SS} - \mu S \Delta\}dt + \sigma S(V_S - \Delta)dB. \end{aligned}$$

Instead of fully integrating the stochastic component (the dB term), the Black-Scholes model eliminates this term by letting,

$$V_S = \Delta;\ \text{then},$$

$$d\Pi = \{V_t + \tfrac{1}{2}\sigma^2 S^2 V_{SS}\}dt$$

Since $d\Pi = r\Pi dt$ according to no-arbitrage principle,

$$d\Pi = \{V_t + \tfrac{1}{2}\sigma^2 S^2 V_{SS}\}dt = r\Pi dt,$$

$$V_t + \tfrac{1}{2}\sigma^2 S^2 V_{SS} - r\Pi = 0.$$

Since $\Pi = V - S\Delta$ and $V_S = \Delta$,

$$V_t + \tfrac{1}{2}\sigma^2 S^2 V_{SS} + rSV_S - rV = 0.$$

Substituting

$$\Pi = V - S\Delta = V - V_S S,$$

we obtain,

$$LV = V_t + \tfrac{1}{2}\sigma^2 S^2 V_{SS} + rSV_S - rV = 0.$$

This is the **Black-Scholes Equation**. We notice that the Black-Scholes model simply gets rid of the stochastic component, which gives rise to partial integro-differential equation (PIDE). This assumption renders the accuracy of the Black-Scholes model on the true movements in the underlying assets. However, the Lévy model retains this term and becomes more realistic to the financial market than the Black-Scholes model.

Since our focus is not Black-Scholes model, we just give the solution. Based on final payoff function $C(S,T) = (S-K)^+$ and boundary conditions of European call option, this problem can be derived into:

$$\text{For call option, } V(S,t) = SN(d1) - PV(K)N(d2);$$

$$\text{For put option, } V(S,t) = -SN(-d1) + PV(K)N(-d2);$$

where,

$$d_1 = \frac{\log(S/K)+(r+\frac{1}{2}\sigma^2)t}{\sigma\sqrt{t}}, d_2 = d_1 - \sigma\sqrt{t}.$$

We will use this solution in Section 5, to compare our model's result of option price to that of the Black-Scholes.

Remark: the Black-Scholes model has stock price which does not go up or down more than the current expected value. However, in the reality, stock price does increase or decrease out of expectations based on the riskless rate. Thus, in the real financial market, stock price moves within expectation most of time, and jumps. Later, this paper discusses the Lévy process, which allows a semi-martingale environment.

3 Lévy Processes

The Black-Scholes model is based on smooth function in continuous time range, not allowing jumps in stock movements. However, in actuality, stock price does jump, and some risks cannot be handled within continuous-path models[4,5,13]. The **Exponential Lévy model** is a choice to include jumps allowing more accurate representation of the market movements. Lévy process tenders a more realistic model of price dynamics than Black-Scholes model. It also allows greater flexibility to calibrate the model to implied volatility smile in option markets. Furthermore, because of its mathematical tractability, the Lévy model has become popular since the late 1990s and early 2000s.

On the other hand, Lévy process is much more complicated to implement. The Poisson process and the Wiener process (also interpreted as Brownian motion, as discussed in the previous section) are fundamental components of Lévy processes. This section introduces basic facts about Lévy processes, and use finite difference and trapezoidal quadrature approximation to develop an explicit-implicit time-stepping scheme to approach the no-arbitrage exponential Lévy model. Then, we will present a detailed implementation of the finite difference scheme in the next section.

Recall the Black-Scholes equation:

$$LV = V_t + \tfrac{1}{2}\sigma^2 S^2 V_{SS} + rSV_S - rV = 0.$$

The equation for a Lévy process is as following,

$$V_t + \tfrac{1}{2}\sigma^2 S^2 V_{SS} + rSV_S - rV + \int_R (V(Se^y, t) - V - S(e^y - 1)V_S)\nu(y)dy = 0.$$

While the Black-Scholes model simplies the option pricing problem by eliminating the stochastic process term, the Lévy process adds it back. Through this section, we will see how the equation for Lévy process evolves.

3.1 Basic facts

Definition: A Lévy Process is a stochastic process X_t with stationary independent increments which is continuous in probability, satisfying[1,4]:

1. $X_0 = 0$;

2. Independent increments: for any $u \leq s < t, X_t - X_s$ is independent of X_u;

3. Stationary increments: for any $s, t > 0, X_{s+t} - X_s$ has the same distribution as $X_t - X_0$.

3.1.1 Simulating Lévy processes

This section explores approach to simulate Lévy process for the readers interest. As we know, Lévy process includes Brownian and Poisson processes. There are many simulation methods available for different Lévy process, for instance, simulation with variance gamma and normal inverse Gaussian. In this paper, we will generate an algorithm and use it to simulate compound Poisson process with Excel. Then, we will use Wolfram Demonstrations Project[15] to present several Mertons Jump Diffusion with different inputs.

(a) Simulation of compound Poisson process

Compound Poisson process can be simulated exactly and the computational time grows linearly with intensity. We make some assumptions, in order to make the simulation simplified and satisfied the Poisson process. A Poisson process is a counting process, which satisfies the following properties[4,7,12]:

1. $X(t) \geq 0, X(0) = 0$;

2. $X(t)$ is integer-valued;

3. $X(s) < X(t)$ if $s < t$;

4. $X(t) - X(s)$ equals the number of events that have occured in the interval (s, t).

We consider the jump size distribution is standard normal, T is 1, the drift is 3 and the jump intensity is equal to 10. The algorithm is as following,

1. Generate a random variable N from Poisson distribution with parameter $\lambda * T$, which yields 6 in our case. N is the total number of jumps on [0,T];

2. Set $t = 0$ and $X = 0$;

3. Generate U, which is uniformly distributed on [0,T];

4. Generate jump size Y with standard normal distribution and law $\nu(dx)/\lambda$, which in our case is ln(U)/10;

5. Update new location X: $X = bt + Y$;

6. Go back to Step 3 and continues, until we have $\sum_{i=1}^{N} t_i \geq T$.

The trajectory of our compound Poisson process is as shown below. The places of vertical arrows represent jumps in our process. This graph tenders an idea of Poisson process: it is discontinuous, and has smaller and larger jumps randomly.

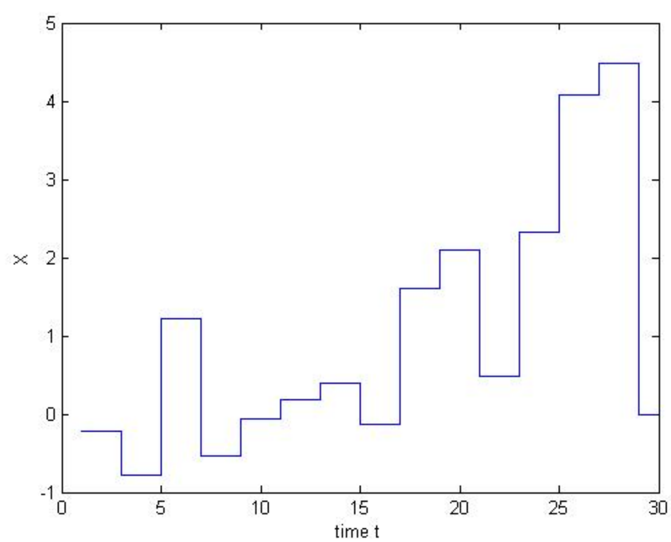

Figure 5: Poisson Process Simulation

(b) Simulation of Mertons jump diffusion model

A Lévy process of jump-diffusion type has the following form:

$$X_t = \gamma t + \sigma W_t + \sum_{i=1}^{N_t} Y_i,$$

where $(N_t)_t \geq 0$ is the Poisson process counting the jumps of X and Yi are jump sizes.

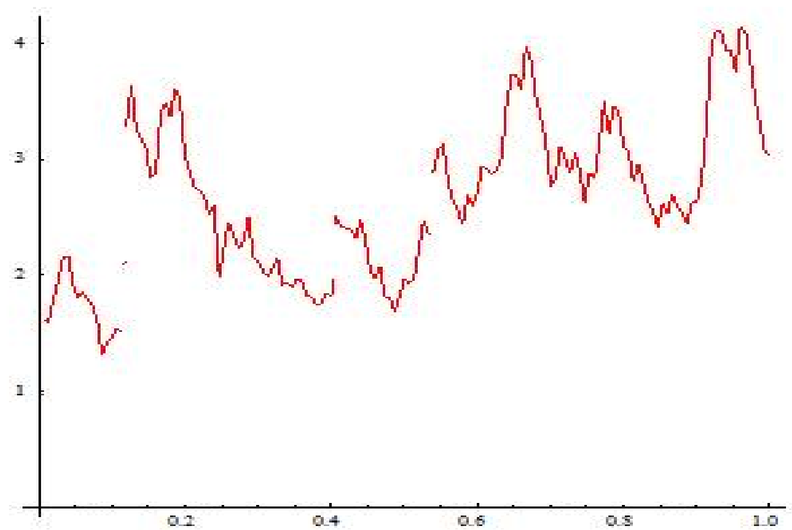

Figure 6: Lévy Process with Merton's Model 1

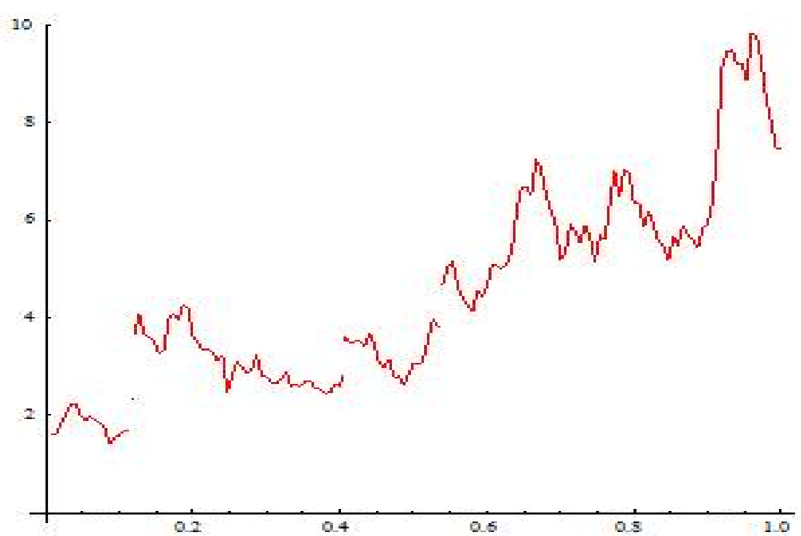

Figure 7: Lévy Process with Merton's Model 2

In the Merton model, it keeps X following the Poisson distribution and assumes that Yi has a Gaussian distribution. The factors influencing this motion are the jump intensity, jump mean size and jump standard deviation. A sketch of algorithm is as following,

1. Simulate N independent centered Gaussian random variables;
2. Simulate the compound Poisson part as discussed in the previous section;
3. Update the trajectory with

$$X(t_i) = bt_i + \sum_{k=1}^{i} G_k + \sum_{j=1}^{N} 1_{U_i<t_i} Y_j.$$

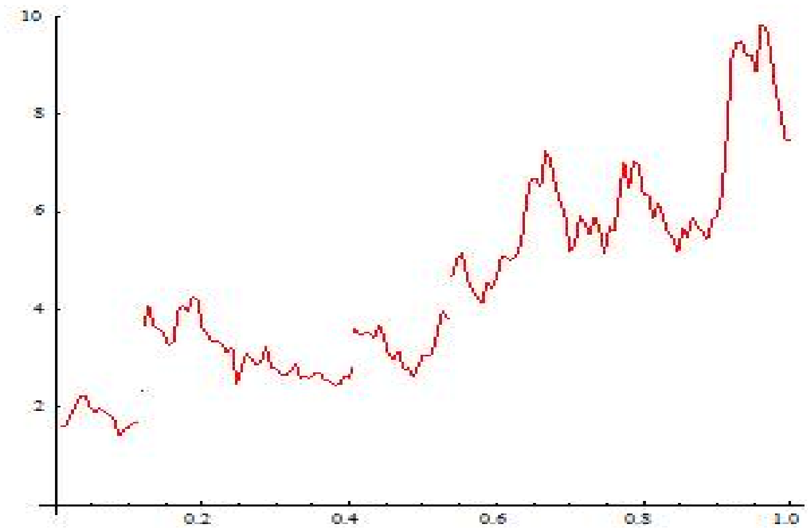

Figure 8: Lévy Process with Merton's Model 3

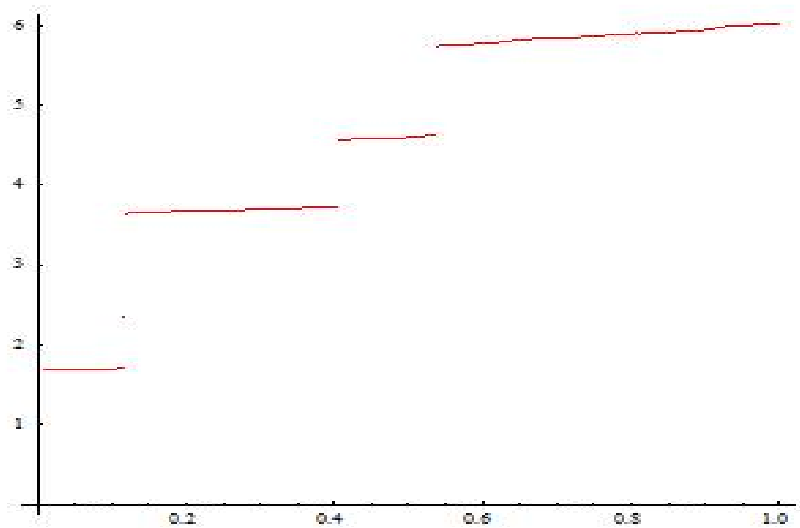

Figure 9: Lévy Process with Merton's Model 4

This trajectory of a Lévy process is with Mertons jump diffusion model. Here jump size is normally distributed with mean at 0 and standard deviation at 0.5. The number of jumps is 5, jump intensity is 10, the diffusion volatility is 1 and the drift is 0.1. If we change the drift to 1, we have Figure 7. The increase in the drift makes the process to have a higher trend of rising in general. Back to the original graph, if we change the diffusion volatility from 1 to 0.01, we have Figure 8. The change in diffusion volatility has smoothed out this process, and makes it have an appearance of the compound Poisson process (Figure 9).

3.1.2 Characteristic function of a Lévy process

Let $(X_t)_t \geq 0$ be a Lévy process on R^d. There exists a continuous function $\psi : R_d \to R$ called the characteristic exponent of X, such that:

$$E[e^{izX_t}] = e^{t\psi(z)}, z \in R^d.$$

The classic Lévy model assumes $S_t = e^{r_t + X_t}$, where X_t indicates the Lévy process. This model combines Brownian motion and Poisson process. According to the **Lévy-Khintchine formula**[1,4], it has the following characteristic function of X_t:

$$E[e^{izX_t}] = e^{-t\psi(z)}$$

$$= \exp\{t(-\tfrac{\sigma^2 z^2}{2} + i\gamma z + \int_{-\infty}^{+\infty}(e^{ixz} - 1 - izx1_{|x|\leq 1})\nu(dx))\},$$

where $\sigma > 0$ and γ are real constants, and ν is a positive measure verifying

$$\int_{-1}^{+1}(x^2\nu(dx)) < +\infty, \int_{|x|>1}\nu(dx) < +\infty.$$

where, the measure ν is called the Lévy measure of X. In the characteristic function, we can see that the integral part in the exponential is compound Poisson process and the rest two terms presents Brownian motion. Thus, X is a collection of Brownian motion and Poisson processes. x denotes the jump size, and $\nu(dx)$ is the intensity of jumps of size x.

Since we use finite difference method to approximate the Lévy processes, we do not need to take small jumps in account. Thus, we can eliminate the term of the Lévy-Khinchin representation can be reduced to

$$E[\exp\{izX_t\}] = \exp\{t(-(\sigma^2 z^2)/2 + i\gamma_0(\nu)z + \int_{-\infty}^{+\infty}(e^{izx} - 1)\nu(dx))\},$$

Also, since Lévy process is Markov process, its infinitesimal generator L^X: $f \to L^X f$ is an integro-differential operator as following,

$$L^X f(x) = \lim_{t\to 0} \frac{E[f(x+X_t)]-f(x)}{t}$$

$$= \frac{\sigma^2}{2}\frac{\partial^2 f}{\partial x^2} + \gamma\frac{\partial f}{\partial x} + \int \nu(dy)[f(x+y) - f(x) - y1_{|y|\leq 1}\frac{\partial f}{\partial x}(x)].$$

Also, similar to the Black-Scholes model, we assume that the underlying asset S_t has a exponential of a Lévy process: $S_t = S_0 e^{rt+X_t}$. X_t is a Lévy process with characteristic triplet (σ, γ, ν).

3.2 No-arbitrage condition

Definition: Let $f(x)$ be a probability density and h be a real number. Its **Esscher transform**[4] is defined as

$$f(x;h) = \frac{e^{hx}f(x)}{\int_{-\infty}^{+\infty} e^{hx}f(x)dx} = \frac{e^{hx}f(x)}{E[e^{hx}f(x)]}.$$

The flexibility of Lévy model allows us to obtain a variety of measures by changing the distribution of jumps, by altering the gamma-function, which is one of the three characteristics of the Lévy random process as stated in the previous section.

In the book *Financial Modelling with Jump Processes* by Roma Cont and Peter Tankov, if the exponential Lévy model is arbitrage-free, *if the trajectories of X are neither almost surely increasing nor almost surely decreasing.* Also, *there exists a probability measure Q equivalent to P such that* $(e^{-rt}S_t)_{t\in[0,T]}$ *is a Q-martingale, where r is the interest rate.*

Applying the Esscher transform, we have

$$\frac{dQ|_{F_t}}{dP|_{F_t}} = \frac{e^{\theta X_t}}{E[e^{\theta X_t}]} = e^{\theta X_t + \gamma(\theta)t}$$

where $\gamma(\theta) = -lnE[e^{\theta X_1}]$ is the log of the moment generating function of X_1, given by the characteristic exponent of the Lévy process X. This is called the Radon-Nikodym derivative corresponding to the Esscher transform. It entails the following properties in order for an exponential Lévy model to be no-arbitrage:

1. X has a nonzero Gaussian component: $\sigma > 0$;
2. X has infinite variation $\int_{-1}^{+1} |x|\nu(dx) = +\infty$;
3. X has both positive and negative jumps;
4. X has positive jumps and negative drift or negative jumps and positive drift.

This proposition can be justified by intuition. If there surely exists increase or decrease in stock price, based on the theory of Efficient Market, investors will behave accordingly in the same direction to absorb the current values without embedded risk in the future, which is clearly an arbitrage opportunity phenomenon.

According to Rama Cont and Peter Tankov [5], the properties above present the following conditions to a no-arbitrage exponential Lévy model,

$$\int_{|y|>1}(dy)e^y < +\infty,$$

$$\gamma = \gamma(\sigma, \nu) = -\frac{\sigma^2}{2} - \int (e^y - 1 - y1_{|y|\leq 1})\nu(dy).$$

Thus, the infinitesimal generator, in the previous sub-section, becomes,

$$L^X f(x) = \frac{\sigma^2}{2}\left[\frac{\partial^2 f}{\partial x^2} - \frac{\partial f}{\partial x}\right] + \int \nu(dy)\left[f(x+y) - f(x) - (e^y - 1)\frac{\partial f}{\partial x}(x)\right].$$

Thus, for $Y_t = rt + X_t$, the infinitesimal generator of Y_t is,

$$Lf = L^X f + r\frac{\partial f}{\partial x}.$$

For Lévy process: $S_t = S_0 e^{rt + X_t}$,

$$V_t + \frac{1}{2}\sigma^2 S^2 V_{SS} + rSV_S - rV + \int_R (V(Se^y, t) - V - S(e^y - 1)V_S)\nu(y)dy = 0.$$

3.3 Explicit-implicit finite difference scheme

Method	**Advantages**	**Drawbacks**
Multinomial tree/lattice	Monotonicity, ease of implementation	Inaccurate representationof jumps, slow convergence
Analytic method of lines	Fast when feasible	Needs closed form Wiener-Hopf factor
Explicit/Implicit finite difference scheme	Fast, simple to implement, monotone, handles barriers efficiently	Collocation method: solution is computed only in some points
Crank-Nicholson/FFT	Fast	Boundary conditions not handled efficiently
Finite elements	Extends to American options	Mathematical formulation is more difficult
Wavelet-Galerkin	Extends to American options	Implementation is difficult

Table 2: Numerical Methods for Partial Integro-Differential Equations *[4](Rama Cont and Peter Tankov, 2004), and edited by Dr. Veneziani - Cont and Tankov originally stated that the drawback to finite elements method was dense matrix to solve. Dr. Veneziani asserted that this is not usual for finite elements method that are the contrary featuruing sparse matrices. He also added that in this case of solving Lévy processes, there is the integral that in principle may lead to a full matrix. This in practice doesn't happen because we truncate the integral, thanks to its exponential decay. For this reason, we have modified the table by Cont and Tankov.*

As listed above in the chart, this paper uses explicit-implicit finite difference scheme to solve the exponential Lévy model, because it works fine without too many difficulties to implement. We start with the European call value function:

$$C(t,S) = E[e^{-r(T-t)}(S_T - K)^+ | St = S].$$

For numerical reason, we replace time t by the time to maturity. We assume $\tau = T-t, x = ln(S/S_0)$ and $S_t = S_0 e^x$, then,

(1) move the $e^{r\tau}$ part into the expectation function, and (2) since $S_t = Se^{r\tau + X_t}$ by definition; (3) by our assumption; let the payoff function $H(S_T) = (S_T - K)^+$ for call and $H(S_T) =$

$(K - S_T)^+$ for put option, $h(x) = H(S_0 e^x))$ and $Y_\tau = r\tau + X_\tau$, then

$$\begin{aligned} u(\tau, S) &= e^{r\tau} C(T - \tau, S_0 e^x) \\ &= e^{r\tau} E[e^{-r(T-t)}(S_T - K) + |S_t = S] \\ &= E[(S_T - K)^+ | S_t = S] \\ &= E[H(Se^{r\tau + X_\tau})] \\ &= E[H(S_0 e^{x + r\tau + X_\tau}] \\ &= E[h(x + Y_\tau)]. \end{aligned}$$

Applying Itô formula to u(t,X_t) on (0,T), we have

$$\frac{\partial u}{\partial \tau} = Lu \text{ on } (0, T)R, \text{ (Cauchy Problem)}$$

where the initial-boundary is $u(0, x) = h(x)$, for $x \in R$. Now we have a starting point, $\frac{\partial u}{\partial \tau} \approx \frac{u^{n+1} - u^n}{\Delta t}$ by finite difference. We still need to explore how Lu can be interpreted.

Recall the infinitesimal generator function. Since X_t is a Lévy process, $Y_t = rt + X_t$ has the infinitesimal generator as, $Lf = L^X f + r\frac{\partial u}{\partial x}$. Thus,

$$Lu = L^X u + r\frac{\partial u}{\partial x} = \frac{\sigma^2}{2}[\frac{\partial^2 u}{\partial x^2} - \frac{\partial u}{\partial x}] + \int \nu(dy)[u(\tau, x + y) - u(x) - (e^y - 1)\frac{\partial u}{\partial x}(x)] + r\frac{\partial f}{\partial x}.$$

Another thing we know about the Lévy model is that it involves integration of an unbounded area; however, in numerical aspect, we cannot provide an exact solution to problems involves infinity. We assume O is an open interval $(a, b) \subseteq R$, and $g \in C_p^+([0, T]xR)$. Then, we have an initial boundary condition: u(0,x) = h(x) for $x \in O$, and $u(\tau, x) = g(\tau, x)$ if $x \notin O$.

We would like to divide this equation into explicit and implicit parts (in time), and use a finite difference scheme to find a solution. First, we need to truncate the integral.

Remark: Here, we substitute $u(\tau, S)$ into the infinitesimal generator, which has another advantage. $u(\tau, S)$ is the forward value of a European option defined by $u(\tau, S) = E[h(x+Y_\tau)]$. $u(\tau, S)$ is the unique viscosity solution of the Cauchy problem $\frac{\partial u}{\partial \tau} = Lu$ on $(0, T) \times R$, if we can prove the payoff function $H(S_T)$ is Lipschitz and h(x) has quadratic growth at infinity. Read *Appendix 1* for more discussion on existence, uniqueness and errors of this solution.

3.3.1 Truncate integral

As stated in the previous section, we need to truncate the integral into a bounded region in order to numerically approximate the solution. Here, we ignore the small jumps in the Lévy processes and focus on the truncation of large jumps. Assume Bl is the left-hand boundary and Br is the right-hand boundary. We now define a new process $\bar{X}_\tau$ with the Lévy characteristic triplet $(\bar{\gamma}, \sigma, \nu 1_{x\in[Bl,Br]})$, such that

$$\bar{\gamma} = -\tfrac{\sigma^2}{2} - \textstyle\int_{B_l}^{B_r} (e^y - 1 - y 1_{|y|\le 1} \nu(dy)),$$

which still keeps $e^{rt+\bar{X}_t}$ the status of a martingale and the properties of no-arbitrage exponential Lévy models.

We are using the following density function to approximate,

$$\nu(x) = 0.1 \tfrac{e^{-x^2/2}}{\sqrt{2\pi}}$$

which is a Merton model with Gaussian jumps. We would like to choose two boundaries B_l and B_r in such way that,

$$| \textstyle\int_{\infty}^{\infty} f(x)dx - \int_{B_l}^{B_r} f(x)dx| < \varepsilon,$$

where ε is a given tolerance. Since the probability measure $\nu(x)$ converges very fast towards to infinity, we focus on the interval, where,

$$\nu(x) \ge \varepsilon \iff -\sqrt{-2\log(10\varepsilon\sqrt{2\pi})} \le x \le +\sqrt{-2\log(10\varepsilon\sqrt{2\pi})}.$$

Thus, $B_r = +\sqrt{-2\log(10\varepsilon\sqrt{2\pi})}$, and $B_l = -B_r$. We will talk more about bounds on error later Section 3.4.

3.3.2 Divide Lu into explicit and implicit parts

Recall, we have

$$\begin{aligned} Lu &= \frac{\sigma^2}{2}[\frac{\partial^2 u}{\partial x^2} - \frac{\partial u}{\partial x}] + \int \nu(dy)[u(\tau, x+y) - u(x) - (e^y - 1)\frac{\partial u}{\partial x}(x)] + r\frac{\partial f}{\partial x} \\ &= \frac{\sigma^2}{2}\frac{\partial^2 u}{\partial x^2} - (\frac{\sigma^2}{2} - r)\frac{\partial u}{\partial x} + \int_{B_l}^{B_r} \nu(dy)[u(\tau, x+y) - u(x) - (e^y - 1)\frac{\partial u}{\partial x}(x)]. \end{aligned}$$

Let $v(R) = \lambda < +\infty$ and $\alpha = \int_{B_l}^{B_r} \nu(dy)(e^y - 1)$, then

$$Lu = \tfrac{\sigma^2}{2}\tfrac{\partial^2 u}{\partial x^2} - (\tfrac{\sigma^2}{2} - r + \alpha)\tfrac{\partial u}{\partial x} - \lambda u + \int_{B_l}^{B_r} \nu(dy)u(\tau, x+y).$$

We define $Lu = Du + Ju$, then,

$$Du = \tfrac{\sigma^2}{2}\tfrac{\partial^2 u}{\partial x^2} - (\tfrac{\sigma^2}{2} - r + \alpha)\tfrac{\partial u}{\partial x} - \lambda u, Ju = \int_{B_l}^{B_r} \nu(dy)u(\tau, x+y) - u(\tau).$$

4 Numerical Approximation and Its Implementation

4.1 Explicit-implicit finite difference scheme

Finite Difference Method is numerical approach to use Taylor series to approximate derivatives. For example,

$$u'(x) \approx \frac{u(x+h)-u(x)}{h},$$
$$u'(x) \approx \frac{u(x+h)-u(x-h)}{2h}.$$

The second one is also called as Centered Difference Approximation. It has $O(h^2)$, which is considered to be smaller than that of the first method which has $O(h)$. However, for numerical purpose, we use the first method to implement our finite difference scheme. We introduce a uniform grid on $[0,T]x[-A,A] : \tau_n = n\Delta t, n = 0,...,M, x_i = -A + i\Delta x, i = 0,...,N$. Then, $\Delta t = T/M$ and $\Delta x = 2A/N$. Let u_i^n be the solution of the numerical scheme. Then,

$$Du = \frac{\sigma^2}{2}\frac{\partial^2 u}{\partial x^2} - (\frac{\sigma^2}{2} - r + \alpha)\frac{\partial u}{\partial x} - \lambda u, and\ numerically,$$
$$Du_i = \frac{\sigma^2}{2}\frac{u_{i+1}-2u_i+u_{i-1}}{(\Delta x)^2} - (\frac{\sigma^2}{2} - r + \alpha)\frac{u_{i+1}-u_i}{\Delta x} - \lambda u,$$

where $\alpha = \int_{B_l}^{B_r} \nu(dy)(e^y - 1)$.

The remaining problem is how to numerically approximate the integrations. We use the trapezoidal quadrature rule with the equalized step sizex. Let K_l, K_r be such that
$[B_l, B_r] \subset [(K_l - 0.5)\Delta x, (K_r + 0.5)\Delta x]$. Then,

$$Ju \approx \int_{B_l}^{B_r} \nu(dy)u(\tau, x+y) - u(\tau) \approx \sum_{j=K_l}^{K_r} \nu_j(u_{i+j} - u_i),$$

where $\nu_j = \int_{(j-0.5)\Delta x}^{(j+1/2)\Delta x} \nu(dy)$. Thus, $\alpha = \int_{B_l}^{B_r} \nu(dy)(e^y - 1) \approx \sum_{j=K_l}^{K_r} \nu_j(e^{y_j} - 1)$.

4.2 A sketch of algorithm

Initialization:

$$u_i^0 = h(x_i), for\ i = 0, ..., N;$$

$$u_i^0 = g(0, x_i), fori \neq 0, ..., N.$$

Build matrix D and J. **For n = 0,..., M - 1:**

Solve:

$$(I - \Delta t D)u^{n+1} = (I + \Delta t J)u^n;$$

Where

$$(Du^{n+1})_i = [\frac{\sigma^2}{2(\Delta x)^2} - \frac{1}{\Delta x}(\frac{\sigma^2}{2} - r + \alpha)]u_{i+1}^{n+1} - [\frac{\sigma^2}{(\Delta x)^2} + \frac{1}{\Delta x}(\frac{\sigma^2}{2} - r + \alpha)]u_i^{n+1} + \frac{\sigma^2}{(2(\Delta x)^2}u_{i-1}^{n+1},$$

$$(Ju^n)_i = \sum_{j=K_l}^{K_r} \nu_j u_{i+j}^n - \lambda u_i^n.$$

Continue until reach stopping criteria.

4.3 MATLAB code

Need to solve: $(I - \Delta t D)u^{n+1} = (I + \Delta t J)u^n$

Example on how to use this algorithm:

```
Ao = 10;
T = 1;
M = 50;
N = 100;
sigma = 0.15;
r = 0.05;

S0 = 100;
K = 100;
```

```
>> [l,bs] = LevyMerton(Ao, T, M,N,S0,K,sigma,r,0);
```

To compute the prices:

```
function [ Levy, BS ] = LevyMerton...
    ( Ao, T, M, N, S0, K, sigma, r, CallOrPut, hFun)
%
%   This function takes inputs initialized by the user, and uses
%   an implicit-explicit finite difference scheme of Levy process
%   to calculate the price of call; also gives a comparison to
%   the Black-Scholes
%   model's result.
%
%   Input
%           Ao: the order of truncated upper bound on x of the grid
%           T: maturity date
%           M: the number of subintervals of time
%           N: the number of subintervals of x movements
%           sigma: the volatility of the underlying stock
%           r: risk-free rate
%           hx: vector of initial conditions
%           S0: initial stock price
%           K: strike price
%           CallOrPut: 0 for Call, 1 for Put
%           hFun: initial conditions, if not passing hFun, default
%                   will be h(x) = (1 - e^x)+
%   Output
%           Levy: our result by the Levy processes with Merton's model
%           BS: result by the Black-Scholes model

%-----------------------------------------------------------------------
% How to use this algorithm?
%
%   We need to initialize some values.
%   Here since we implement the Merton's model for probability
%   measure, we have,
%       Initial conditions: h(x) = (1 - e^x)+; (by default)
%                           g(tao, x) = 0;
%       Probability measure:v(x) = 0.1 * exp(-x .* x * 0.5) / sqrt(2*pi)
%-----------------------------------------------------------------------

%   1.
%   Initial values

myFun = @(x) x .* x .* (0.1 * exp(-x .* x * 0.5) / sqrt(2*pi));
```

```
A = Ao * sqrt(T * (sigma * sigma + quadgk(myFun,-inf, inf)));

dx = 2 * A / N;
dt = T / M;

if nargin == 9
    if CallOrPut == 0
        hFun = @(x) exp(x) - 1;
    else if CallOrPut == 1
        hFun = @(x) 1 - exp(x);
        end
    end
end
hx = zeros(N+1,1);
x = zeros(N+1,1);
t = linspace(0,T,M+1)';

for i = 1:(N+1)
    x(i) = -A + i * dx;
    hx(i) = max(0, hFun(x(i)));

    if hx(i) < 0
        hx(i) = 0;
    end
end

%   2.
%   localization conditions
%           ep: given tolerance
%           Bl: left hand truncation of the integral
%           Br: right hand truncation of the integral
%   Without losing generality, Bl < -1, Br > 1, and
%       ep needs to be smaller than 1/sqrt(2*pi).
%
ep = 1e-3;
Bl = -sqrt(-2*log(10*ep*sqrt(2*pi)));
Br = - Bl;
%   Since [Bl, Br] is part of [(Kl - 0.5)dx, (Kr + 0.5)dx],
Kl = max(floor(Bl / dx + 0.5),-N);
Kr = - Kl;

%   3.
%   Simulate vectors u with function BuildU.m
[u, C] = BuildU( A, T, M, N, Kl, Kr, sigma, r, hx,K,S0 );
```

```
%   4.
%   Calculate the expected price of C.Price is desired calculated price.
sumC = sum(sum(C));
Levy = sumC / N / M;
d1 = (log(S0/K) + (r + sigma*sigma / 2) * T) / (sigma * sqrt(T));
d2 = d1 - sigma * sqrt(T);
PV_K = exp(-r*T) * K;
if CallOrPut == 0
    BS = S0 * normcdf(d1,0,1) - PV_K * normcdf(d2,0,1);
else if CallOrPut == 1
        BS = PV_K * normcdf(-d2,0,1) - S0 * normcdf(-d1,0,1);
    end
end
end
```

To construct the u_{n+1} updating system:

```
function [ u, C, flag ] = BuildU( A, T, M, N, Kl, Kr, sigma, r, hx, K, S0 )
%   function BuildU
%
%   This algorithm builds the vectors u_i one by one and compact
%   into a matrix u. First, we use function BuildD and BuildJ to
%   build left-hand and right-hand sides of this equation:
%               (I - dt*D) u_(n+1) = (I + dt*J) u_n.
%   Since this linear system involves large assymmetric and sparce
%   matrix on both sides, we use GMRES iteration to solve it.
%
%   Input:  A: truncated upper bound on x of the grid
%           T: maturity date
%           M: the number of subintervals of time
%           N: the number of subintervals of x movements
%           sigma: the volatility
%           r: risk-free rate
%           hx: vector of initial conditions
%
%   Output: u: a matrix containing u_i's as column vectors

% Initialization:
dt = T / M;
dx = 2*A / N;

u = zeros(N+1, M);
C = zeros(N+1, M);
u(:,1) = hx;
C(:,1) = u(:,1);

% build J and D.
[J, alpha, lambda] = BuildJ(N, dx, dt, Kl, Kr);
D = BuildD(sigma, dx, dt, r, N, alpha, lambda);

flag = zeros(M,1);
tao = 0;

for n = 1:M
    tao = tao + dt;
    b = J * u(:,n);
    [u(:,n+1), flag(n)] = gmres(D,b);
    C(:,n+1) = u(:, n+1) * exp(-r*tao) * K / S0;
end
end
```

To construct the left-hand side matrix:

```
function [ D ] = BuildD( sigma, dx, dt, r, N, alpha, lambda )
%
%   function BuildD
%
%   This function constructs the matrix D we use to pair with the
%   u_n+1. And, we scale it by (I + dt*D). To computational purpose,
%   we keep this matrix sparce; since it is tri-diagonal and asymmetric,
%   we will use iterative methods such as GMRES to solve this problem.
%
%   Input:  sigma: the volatility constant
%           dx: increment in step
%           dt: increment in time
%           r: risk-free rate
%           n: size of the matrix
%           alpha: coefficient for first order derivative
%
%   Output: D: nxn desired sparce matrix
%

C = sigma * sigma / (2 * (dx * dx)) - (1/dx) * (sigma*sigma/2 - r + alpha);
B = sigma*sigma / (dx*dx) + (1/dx) * (sigma*sigma/2 - r + alpha) + lambda;
A = sigma * sigma / (2 * dx * dx);

v1 = A * ones(N, 1);
v2 = - B * ones(N+1, 1);
v3 = C * ones(N, 1);

Diag = [[v1;0], v2, [0;v3]];
D = speye(N+1, N+1) - dt * spdiags(Diag, [-1,0,1], N+1, N+1);

end
```

To construct the right-hand side matrix:

```
function [ J, alpha, lambda ] = BuildJ( N, dx, dt, Kl, Kr )
%   function BuildJ
%
%   This algorithm build the matrix J which we need for the
%   right-hand side computation. Here, we use the Merton's
%   model to construct the jump steps function v:
%
%          vFun = @(x) 0.1 * exp(-x .* x * 0.5) / sqrt(2*pi);
%
%   We then calibrate the matrix J into the complete right-
%   hand side form: (I + dt * J).
%
%   Input:  N: the number of subintervals of x
%           dx: movement step size
%           dt: time step size
%
%   Output: J: (N+1)x(N+1) matrix of the right-hand side
%             information to update the vector u_n to u_n+1
%           alpha: coefficient for localization of boundaries
%

vFun = @(x) 0.1 * exp(-x .* x * 0.5) / sqrt(2*pi);

v = zeros(Kr - Kl + 1, 1);
lambda = 0;
alpha = 0;
J = zeros(N+1,N+1);

for i = Kl : Kr
    lower = (i - 0.5) * dx;
    upper = (i + 0.5) * dx;
    k =i + (1 - Kl);
    y = Kl * dx;
    v(k) = quad(vFun, lower, upper);
    lambda = lambda + v(k);
    alpha = alpha + (exp(y)-1)*v(k);
    V = ones(N+1-abs(i),1) * v(k);
    J = J + diag(V, i);
end

J = eye(N+1, N+1) + dt * J;
end
```

4.4 Explanation of MATLAB code

The first algorithm gives an example on how to use those self-built MATLAB functions to compute desired option prices. In the example, we have expiry date T as 1, the number of time intervals M as 50, the number of movement intevals as 100, the volatility σ as 15% and the riskfree rate as 5%.

Another term we need to initialize is 'Ao', which stands for the order of A. 'A' is the truncation boundary as discussed in Section 3.3. 'The order of A' means a ratio of the domain size A to the standard deviation of the Lévy process X_T,

$$A = Ao\sqrt{T(\sigma^2 + \int y^2 \nu(dy))}$$

Since we use the Merton's model with Lévy density,

$$\nu(x) = 0.1\frac{e^{-x^2/2}}{\sqrt{2\pi}},$$

we compute the domain size A with the following code (as shown in *function LevyMerton*),

```
myFun = @(x) x .* x .* (0.1 * exp(-x .* x * 0.5) / sqrt(2*pi));
A = Ao * sqrt(T * (sigma * sigma + quadgk(myFun,-inf, inf)));
```

The *function LevyMerton* also allows the user to pass in a different function for initial condition, while the default is $h(x) = 1 - e^x$ for put option.

The *function BuildU* implements the sketch of algorithm in Section 4.2. It calls *function BuildD* and *function BuildJ* to construct the left-and right-hand matrix of the finite difference scheme. Then, since our system is a large sparce linear system, it uses *gmres*[8] to update vector u of option payoffs at current time level to the next level. Since our algorithm uses the idea of backward integration, thus, u is updated towards to the initial time.

5 Numerical Results

We use the MATLAB codes as displayed in the previous section, to compute the European put prices with our model, and compare them to that of the Black-Scholes model.

We assume it is an at-money European call option: $S_0 = K = 100$, maturity of T at 1, $\sigma = 15\%$, $r = 5\%$. **The put price by the Black-Scholes model is 3.7146**. The *Implied Volatility Error in %* in our analysis is calculated by the Black-Scholes Implied Volatility with MATLAB *function blsimpv*. For example,

```
[put,bs] = LevyMerton(Ao, T, Ms(i),N,S0,K,sigma,r,1);
vol(i) = blsimpv(S0, K, r, T, put,[],[],[],false);
```

where "false" means Put option and "true" means Call option. Then, we calculate the error with the following code,

```
error = abs(vol-sigma)/sigma;
```

where "sigma" is the pass-in volatility σ.

5.1 Different parameters and implied volatility errors

First, we conduct a sensitive analysis with different orders of A. As shown,

Ao	*A*	**put by Lévy**	**put by Black-Scholes**
5	1.40	91.3556	3.7146
7	2.10	13.9428	3.7146
9	2.80	5.5594	3.7146
11	3.85	3.2337	3.7146
13	4.55	2.2921	3.7146
15	5.25	1.7965	3.7146

Table 3: Put Price Results with Changed Order of A

The right graph in Figure 10, the put price is significantly influenced by the order of A. Also, we can see the the order larger than 5 gives much better results.

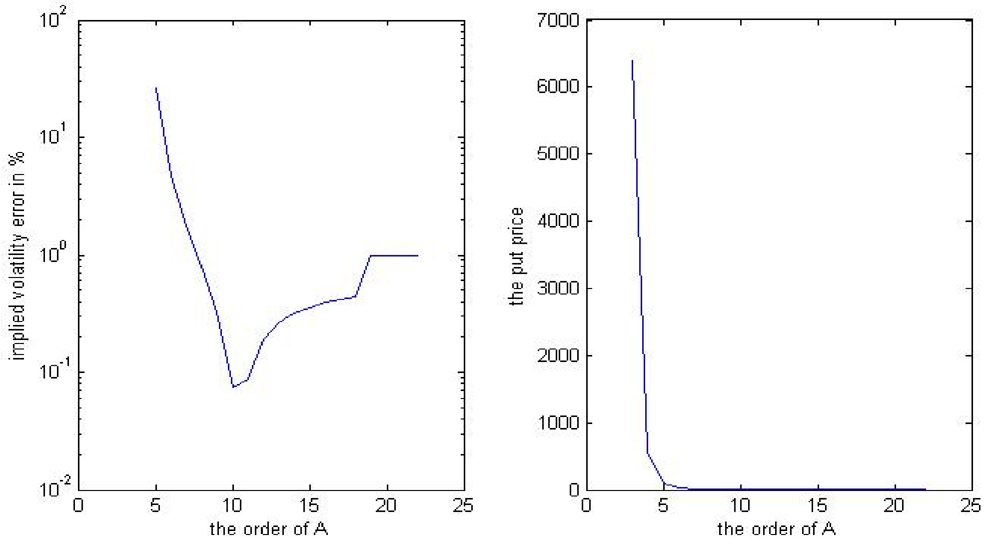

Figure 10: Results with Changed A - the left shows how the Black-Scholes implied volatility changes with different order of A; the right shows how the put price changes with the order of A.

The difference in put prices between by Lévy model and the Black-Scholes model can be justified by the flexibility of the Lévy model on probability measure on jumps in asset price dynamics. From the left graph in Figure 10, we recognize that 10 is a nice order of A to compute the put price.

Then, we conduct a similar sensitive test with changed values of M, the number of time intervals. As shown in Figure 11, the error can be effectively eliminated by the increase in M. Also, we can see from the right graph in Figure 11, the put price is not heavily influenced by the size of M. The difference is actually negligible compared to the changes dominated by the order of A.

Again, we conduct a sensitive analysis based the put prices and the changed size of N, as shown in Figure 12. We expected the implied volatility error decreased with increasing size of N, and the put price converged. Unfortunately, the error and the put price do not behave as we expected.

One possible reason is that, the increased N tenders greater opportunity for random jumps in price dynamics. This consequantially leads to an increase in the put prices and

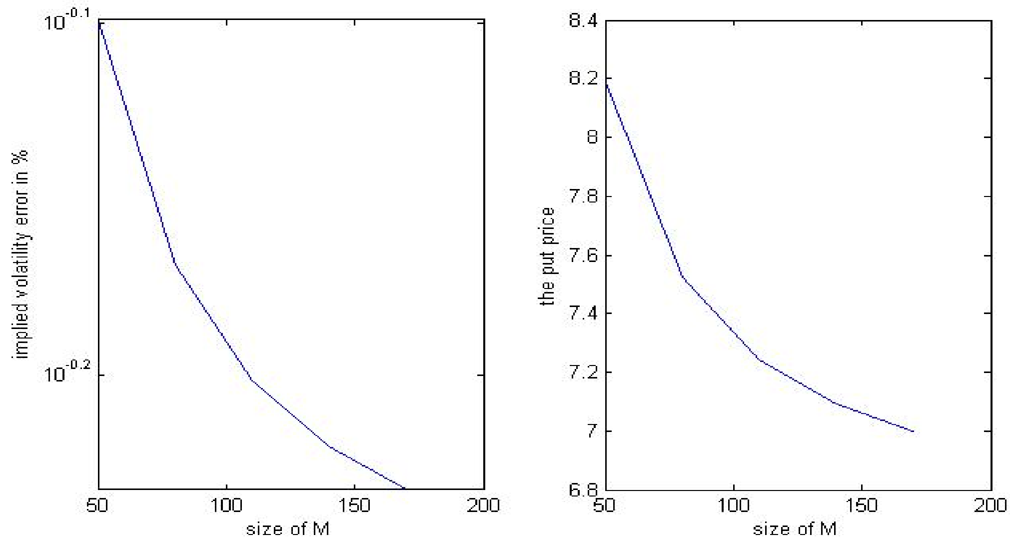

Figure 11: Results with Changed M - with the order of A at 8 and the number of asset movements N at 100.

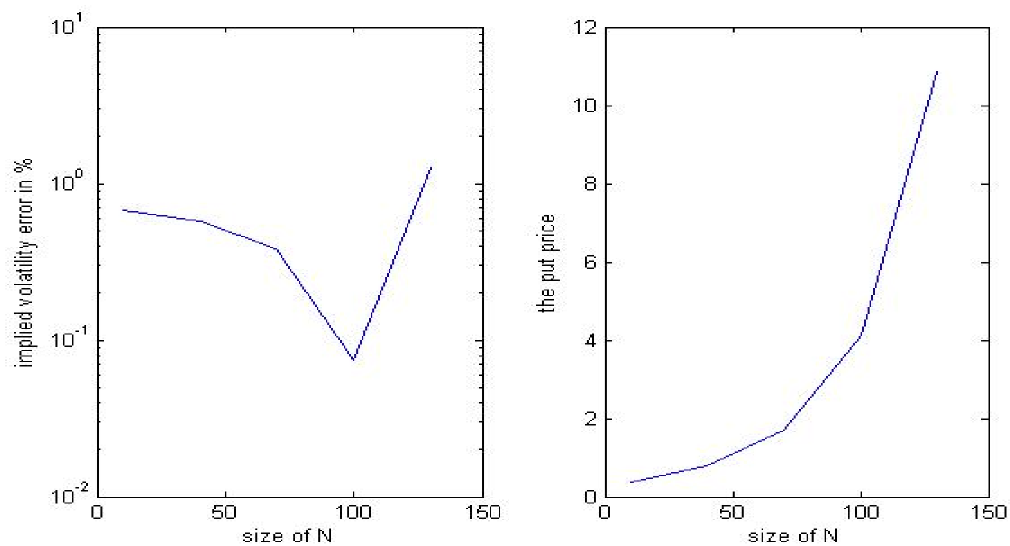

Figure 12: Results with Changed N

the implied volatility error. Another possible reason for such turn-out is that our boundary condition $g(x_i, t) = 0$ for $i \notin \{0, 1, 2, ..., N\}$ is not good enough. For further study, the reader can design a better boundary condition. Due to time constraint, we assume $dx = \frac{2A}{N}$ is between 0.06 to 0.2 with corresponding values of N; i.e.,if $A = 3$, N is between 30 to 100.

5.2 Results with comparison to the Black-Scholes Model

From the discussion of the previous sub-section, we notice that the order of A is a major factor on our option pricing model. Thus, we conduct a sensitive analysis with comparison to the Black-Scholes model. We use the following MATLAB code,

```
T = 1;
M = 80;
N = 100;
sigma = 0.15;
r = 0.05;
S0 = 100;
K = 100;
orderA = linspace(5,15,11)';
for i = 1:11
    [put(i),bs(i)] = LevyMerton(orderA(i),T,M,N,S0,K,sigma,r,1);
end
plot(orderA,put,orderA,bs,'--')
ylabel('the put price'),xlabel('order of A')
legend('by Levy','by Black-Scholes')
```

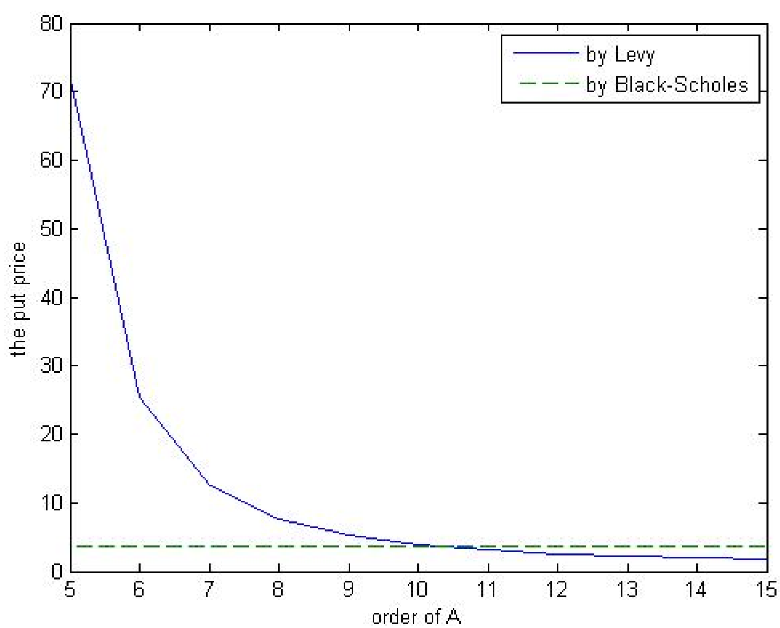

Figure 13: Results by Lévy compared to the Black-Scholes

As shown in Figure 13, the put prices by our model varies a lot at lower orders of A; however, it soon converges when the order of A is larger than 7. We consider the differences in the put prices by our model and by the Black-Scholes model are mainly caused by the different considerations in volatility. Based on our results, the user can choose a put price with his/her own aspects on the market volatility. For example, if the user forecasts there will be a economic downturn within the duration of an option contract, he/she can choose the price with the order of A at 9, which is slightly larger than the put price calculated by the Black-Scholes model.

6 Conclusion

As we have seen through the paper, the Lévy process model is more difficult to implement and involves more computations compared to the Black-Scholes model. Thus, the question is whether it is worth to implement a Lévy process model. We summarize that the Lévy process model does have certain advantages over the Black-Scholes model:

1. Jumps in stock price: more realistic than the Black-Scholes model;

2. Flexibility: we can use different probability measures (or return innovation distribution) to capture different return type;

3. Tranctability: our model can implement any tractable $\psi(z)$ in the Lévy process characteristic function. This can allow us to generate a Fourier transform for any price dynamics problem;

4. Market coverage: using different jump probability measures, our model is capable to capture some other market aspects such as defaults, sudden movements. For example, the economic downturn starting in the end of 2008.

Based on the market expectation and risk aversion, the investors can wisely choose between the Lévy model and the Black-Scholes model. However, we should always keep in mind that it is impractical to model the stock in truly reality, because it is inherent from the incomplete market.

7 Appendix

7.1 Existence/uniqueness and errors: exp. Lévy model

Accuracy is yielded in the way of numerical approximation. This section will talk about the errors incurred during evaluation of the explicit-implicit exponential Lévy model.

7.1.1 Lipschitz Condition

Definition: the function f(t,y) satisfies the Lipschitz condition[8],
if for all $t \in [a, b]$ and for all y and ŷ,

$$|f(t,\hat{y}) - f(t, y)| \leq L|\hat{y} - y|, \text{ where L is a constant.}$$

Remark: if $\frac{\partial f}{\partial y}$ exists and is continuous, then $|f(t,\hat{y}) - f(t, y)| \leq |\frac{\partial f}{\partial y}(t, \xi)||\hat{y} - y|$, where ξ is between ŷ and y.

The upper bound limit brought by the Lipschitz condition will help us find the limit of errors incurred during our numerical approximation.
We consider a European call with maturity T and payoff $H(S_T)$. Then,

$$H(S_T) = (S_T - K)^+, \text{ assume } H \text{ has Lipschitz condition:}$$

$$|H(\hat{y}) - H(y)| \leq L|\hat{y} - y|, \text{ where L is a positive constant.}$$

Since we consider the option to be a European call with $H(x)$ as $(x - K)^+$, we can show that

$$
\begin{aligned}
|H(\hat{y}) - H(y)| &= |(\hat{y} - K)^+ - (y - K)^+|, \text{ assume } \hat{y} > y \\
&= |\hat{y} - y|, \quad \text{if } \hat{y}, y > K; \\
&= |\hat{y} - K|, \quad \text{if } \hat{y} > K > y; \\
&= 0\ , \quad \text{if } \hat{y}, y < K.
\end{aligned}
$$

Thus we have, $|H(\hat{y}) - H(y)| < L|\hat{y} - y|$, and $L = 1$. Since H is Lipschitz, C is also Lipschitz,

$$\begin{aligned} |C(t, S_1) < C(t, S2)| &= |E[e^{-r\tau}H(S_1e^{r\tau} + X_\tau)] - E[e^{-r\tau}H(S_2e^{r\tau} + X_\tau)]| \\ &= e^{-r\tau}|E[H(S_1e^{r\tau} + X_\tau)] - E[H(S_2e^{r\tau} + X_\tau)]| \\ &\leq L|S_1 - S_2| \end{aligned}$$

7.1.2 Viscosity solutions

Viscosity solution yields existence and uniqueness for jump models with finite variation. This paper also uses viscosity solutions to determine the error between the true value and approximation of payoff function.

A function is viscosity solution if this function is both a viscosity subsolution or supersolution of the Cauchy problem, which means the difference between this function and any test function has a global maximum and a global minimum at a given point, say at $(\tau,\, x)$. Then, this function is also continuous on the entire domain of this problem.

The definition above implies that, if u is a subsolution in the upper semicontinuous with an initial condition u_0, and ν is a supersolution in the lower semicontinuous with initial condition $\nu(0)$, then if $u_0 \leq \nu_0$, we can conclude that u is smaller or equal to ν at any point on the given domain $]0, T] \times R$. This proposition is also called as the Comparison Principle. It leads to another important principle in the financial interpretation, the Arbitrage Inequality[4]:

If the terminal payoff of a European option dominates the terminal payoff of another one, then their values should verify the same inequality.

In other words, if a call option C_1 terminal payoff is greater than that of C_2, then C_1 should cost more than C_2. Otherwise, arbitrage opportunity will occur. Since our model is entirely based on no-arbitrage environment, the arbitrage inequality holds.

7.1.3 Bound on approximation error

Since numerical computations can only be performed on finite domains, we need to reduce the problem into a bounded domain, as discussed previously. We assume A to be a number large enough in the exponential sence, so that $(-A, A)$ is our desired artificial bounded domain with limited error under our control. We have defined our finite difference grid based on this. Since we add artificial bounds, we also need to create corresponding boundary conditions. Here, we have,

$$u_i^0 = h(x_i), i = 0, , N;$$
$$u_i^0 = g(0, x_i), i \neq 0, , N;$$
$$u_i^n = g((n+1), x_i), i \neq 0, , N;$$

We need to be careful with these boundary conditions, since nonlocality, brought in by the integral part in our integro-differential equation, may destroy the regularity of our established boundary of functions.

Proposition: Bound on localization error[4]

If $||h||_{L^\infty} < \infty$, and $\exists \alpha > 0, \int_{|x|>1} \nu(dx) e^{\alpha|x|} < +\infty$ then

$$|u(\tau, x) uA(\tau, x)| \leq 2C_{\tau,\alpha}||h||_\infty e^{-\alpha(A-|x|)}.$$

where $C_{\tau,\alpha} = Ee^{\alpha M_\tau}$, $M_\tau = \sup_{t\in[0,\tau]} |X_t|$.

Proof is shown in *Appendix 2*.

(sup(X) means the supremum or least upper bound of a set X, and sup(X).)

Proposition: Bound on errors of truncation of large jumps[4]

According to the Lipschitz condition, the error due to the truncation of large jumps can be estimated as $|u(\tau, x) - \tilde{\ } u(\tau, x)| \leq ||h'||_{L^\infty} \tau (C_1 e^{-\alpha_l |B_l|} + C_2 e^{-\alpha_r |B_r|})$.

As we observe from these two bounds on approximation errors, they contain negative tails of those exponential components. The exponentially decreasing feature tells that the approximation errors are exponentially small.

7.2 Bound on localization error

Proof: recall $u(\tau, S) = E[h(x+Y_\tau)]$. Define, $M_\tau^x = \sup_{t\in[0,\tau]} |Y_t + x|$. Then

$$\begin{aligned} u(\tau, S) &= E[h(x+Y_\tau)] \\ uA(\tau, S) &= E[h(x+Y_\tau)1_{M_\tau^x<A}] \\ or\ uA(\tau, S) &= E[h(x+Y_\tau)1_{M_\tau^x<A} + h(Y_{\theta(x)}+x)1_{M_\tau^x<A}]. \end{aligned}$$

For $g = 0$, (condition function for out-of-boundary situation), according to the Lipschitz condition:

$$\begin{aligned} |u(\tau, S) - uA(\tau, S)| &= |E[h(x+Y_\tau)(1 - 1_{M_\tau^x<A})]| \\ &= |E[h(x+Y_\tau)1_{M_\tau^x<A}]| \\ &\leq ||h||_\infty Q(M_\tau^x \geq A) \end{aligned}$$

For g = h(x),

$$\begin{aligned} |u(\tau, S) - uA(\tau, S)| &\leq |E[h(x+Y_\tau)(1 - 1_{M_\tau^x<A})]| + E[h(x+Y_{\theta(x)})1_{M_\tau^x\geq A}]| \\ &\leq 2||h||_\infty Q(M_\tau^x \geq A). \end{aligned}$$

Also, we have $C_{\tau,\alpha} = Ee^(\alpha M_\tau^0) < \infty$. After applying Chebyshevs inequality, we have,

$$Q(M_\tau^0 \geq A) \leq C_{\tau,\alpha}e^{-\alpha A}$$

Since $\sup|Y_t + x| \leq \sup|Y_t| + |x|$, and $\sup|Y_t + x| \geq A$ and $\sup|Y_t| + |x| \geq A$, thus,

$$\begin{aligned} Q(M_\tau^x \geq A) &= Q(\sup|Y_t + x| \geq A) \leq Q(\sup|Y_t| + |x| \geq A) \\ &= Q(M_\tau^0 \geq A - |x|) \leq C_{\tau,\alpha}e^{-\alpha(A-|x|)}. \end{aligned}$$

Thus,

$$|u(\tau, x) - uA(\tau, x)| \leq 2C_{\tau,\alpha}||h||_\infty e^{-\alpha(A-|x|)}.$$

8 Reference

[1]Achdou, Yves, and Olivier Pironneau. *Computational Methods for Option Pricing.* N.p.: SIAM Frontiers in Applied Mathematics, n.d. Print.

[2]Broadie, Mark, and Jerome B. Detemple. "Option Pricing: Valuation Models and Applications." *Management Science* 50.9 (2004): 1145-77. Print. An extensive review of valuation methods for European- and American-style claims

[3]*CBOE Market Statistics.* Chicago Board Options Exchange, n.d. Web. 3 Apr. 2013. <http://www.cboe.com>. The year of 2008, 2009, 2010, 2011

[4]Cont, Rama, and Peter Tankov. *Financial Modelling with Jump Processes.* N.p.: Chapman & Hall/CRC Financial Mathematics Series, 2004. Print.

[5]Cont, Rama, and Ekaterina Voltchkova. "A Finite Difference Scheme for Option Pricing in Jump Diffusion and Exponential Lévy." *SIAM Journal on Numerical Analysis* 43.4 (2006): n. pag. Print.

[6]Duffy, Daniel J. "Numerical Analysis of Jump Diffusion Models: A Partial Differential Equation Approach." *Datasim. Quantnet*, n.d. Web. 3 Apr. 2013. numerical methods to approximate the solution of the Partial Integro Differential Equation

[7]Gander, Matthew P.S., and David A. Stephens. "Simulation and Inference for Stochastic Volatility Models Driven by Lévy Processes." *Biometrika* 94.3 (2007): 627-46. Print. Ornstein-Uhlenbeck stochastic processes driven by Lévy processes

[8]Nagy, James. "Numerical Analysis." MATH516. Emory University. 2013. Lecture. Finite

difference method, Lipschitz condition, etc.

[9]Protter, Philip, and Denis Talay. ”The Euler Scheme for Lévy Driven Stochastic Differential Equations.” Annals of Probability, the 25.1 (1997): 393-423. Print.

[10]Rosinski, Jan. ”Simulation of Lévy Processes.” Department of Mathematics. University of Tennessee. Reading.

[11]Salsa, Sandro. Partial Differential Equations in Action: From Modeling to Theory.

[12]Sigman, Karl. Poisson Process, and Compound (batch) Poisson Processes. N.p.: n.p., 2007. Print.

[13]Tankov, Peter. ”Financial Modeling with Lévy Processes.” Centre De Mathematiques Appliquees: n. pag. Print.

[14]Valerio, Nicholas, III. ”Advanced Derivatives.” BUS 484. Goizueta Business School, Emory Universtiy. 2013. Speech. Black-Scholes Model, option pricing, etc.

[15]Wolfram Demonstration Projects. Wolfram Methematica, n.d. Web. 3 Apr. 2013. <http://demonstrations.wolfram.com/MertonsJumpDiffusionModel/>. Merton’s Jump Diffusion Model