PARAMETER ESTIMATION IN MERTON JUMP DIFFUSION MODEL

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

TUĞCAN ADEM ÖZDEMIR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
STATISTICS

JULY 2019

Approval of the thesis:

PARAMETER ESTIMATION IN MERTON JUMP DIFFUSION MODEL

submitted by **TUĞCAN ADEM ÖZDEMIR** in partial fulfillment of the requirements for the degree of **Master of Science in Statistics Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** ________________

Prof. Dr. Ayşen Dener Akkaya
Head of Department, **Statistics** ________________

Assist. Prof. Dr. Ceren Vardar Acar
Supervisor, **Statistics Department, METU** ________________

Examining Committee Members:

Assist. Prof. Dr. Fulya Gökalp Yavuz
Statistics Department, METU ________________

Assist. Prof. Dr. Ceren Vardar Acar
Statistics Department, METU ________________

Assoc. Prof. Dr. Aslı Yıldız
Mathematics Department, Hacettepe University ________________

Date:

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Tuğcan Adem Özdemir

Signature :

ABSTRACT

PARAMETER ESTIMATION IN MERTON JUMP DIFFUSION MODEL

Özdemir, Tuğcan Adem
M.S., Department of Statistics
Supervisor: Assist. Prof. Dr. Ceren Vardar Acar

July 2019, 84 pages

Over the years, jump diffusion models become more and more important. They are used for many purposes in several branches such as economics, biology, chemistry, physics, and social sciences. The reason for prevalent usage of these jump models is that they capture stochastic movements and they are sensitive to jump points. It is possible to measure sudden decreases/increases caused by some reasons such as wars, natural disasters, market crashes or some dramatic news, by jump diffusion models.

Recently, US Dollar to Turkish Lira exchange rate has showed dramatic increases/decreases. It is very difficult to model this exchange rate data with classical modeling methods. In this thesis, we try to model this data with Merton model which is among the well-known jump diffusion models. To obtain true parameter estimation algorithm, we simulate a data by using Merton structure. The values of parameters are found with Maximum Likelihood Estimation (MLE). The initial parameter values in simulated data and the estimated parameter values are compared to control the parameter estimation is true or not. Also, the values of Euler-Maruyama numerical approximation method and analytical solution values are checked whether convergence is good or not. After the true parameter estimation algorithm is found, US Dollar to Turkish Lira exchange rate data is used. This data is between date of $01.02.2019$ and $21.06.2019$. By using this data, the parameter estimation is made and prediction is made for between date of $23.06.2019$ and $02.07.2019$ for both Merton Jump Diffusion model and Black-Scholes model. Finally, the fitting and forecasting accuracy performances of them are compared.

Keywords: Merton Jump Diffusion model, Euler-Maruyama method, Exchange rate, Stochastic Differential Equation

ÖZ

MERTON SIÇRAMALI DİFÜZYON MODELLERİNDE PARAMETRE TAHMİNİ

Özdemir, Tuğcan Adem

Yüksek Lisans, İstatistik Bölümü

Tez Yöneticisi: Dr. Öğr. Üyesi. Ceren Vardar Acar

Temmuz 2019 , 84 sayfa

Sıçramalı difuzyon modelleri her geçen yıl daha önemli bir konuma gelmektedir. Bu modeller birçok amaç için ekonomi, biyoloji, kimya, fizik ve sosyal bilimler gibi farklı branşlarda kullanılmaktadır. Sıçramalı Difüzyon modelleri stokastik hareketlere ve sıçramalara karşı duyarlı oldukları için yaygın bir şekilde kullanılmaktadır. Savaş, doğal afet, finansal kriz veya dramatik haberler nedeniyle veride oluşan ani azalış veya artışları bu modeller ile ölçümlemek mümkündür.

Son zamanlarda, Dolar/TL döviz kurunda ani artış ve azalışlar görülmektedir. Bu veriyi geleneksel yöntemler ile modellemek çok zordur. Tez çalışmasında, en çok bilinen sıçramalı modellerden biri olan Merton Sıçramalı Difuzyon modeli ile bu veri modellenmeye çalışılmıştır. Parametre tahmini için oluşturulan algoritmayı doğru bir şekilde oluşturabilmek için Merton yapısına göre bir veri simülasyonu yapılmıştır. Bu verideki parametre başlangıç değerleriyle Maksimum Olabilirlik Tahmini yöntemi ile bulunan parametre tahminleri karşılaştırılarak algoritmanın doğruluğuna karar verilmiştir. Ayrıca, Euler-Maruyama yöntemi ile numerik yaklaşım ve analitik çözümün birbirlerine ne kadar yakınsadıkları kontrol edilmiştir. Parametre tahmini için elde edilen algoritma kullanılarak 01.02.2019 - 21.06.2019 tarihleri arasındaki Dolar/TL döviz kuru verisi ile parametre tahmini; 23.05.2019 - 02.07.2019 tarihleri arası için ise öngörü yapılmıştır. Bu işlemler hem Merton Difüzyon modeli hem de Black-Scholes modeli için uygulanmıştır. Son olarak, bu iki model tahmin ve veriye uyumluluk açısından karşılaştırılmıştır.

Anahtar Kelimeler: Merton Şıçramalı Difüzyon Modeli, Euler-Maruyama yöntemi, Döviz kuru, Stokastik Diferansiyel Denklem

To my beloved family...

ACKNOWLEDGMENTS

Firstly, I want to declare my appreciations to my advisor Assist. Prof. Ceren Vardar Acar. She helped to me with great devotion. In every step of my work, she was gentle and very understanding.

Secondly, I would like to say about my examining committee members, Assist. Prof. Fulya Gökalp Yavuz, Assoc. Prof. Dr. Aslı Yıldız. I appreciated to spare their valuable time to review my thesis and to declare their precious suggestions.

I also appreciate my friend Semih Ergişi. During the thesis, he gave his technical support to me especiall in the subject of Brownian Motion.

Finally, my greatest thanks to my family and my wife Neval Özdemir for endless support and great understanding. Their encouragements when the times got rough are much appreciated and duly noted.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

AIC	Akakike Information Criteria
GBM	Geometric Brownian Motion
MAPE	Mean Absolute Percentage Error
MLE	Maximum Likelihood Estimation
ODE	Ordinary Differential Equation
SDE	Stochastic Differential Equation
PDE	Partial Differential Equation

CHAPTER 1

INTRODUCTION

In many areas, algebraic methods are used to evaluate value of many things. For example, area of a place, speed of a car, density of liquid, etc. Although these methods are frequently used, they are inadequate to meet needs for evaluating some unstable situations; for instance heat loss, seismic waves detection, fluctuation in population. These cases show changes in their situations [4]. At this point, differential equations become a vital alternative. It has ability to measure to change. Differential equations break down into two parts namely deterministic differential equation and stochastic differential equation (SDE). Deterministic equations are used in many cases in nature, finance, and technology to model this cases. However, this modeling does not consider stochastic increases/decreases and is not proper for some areas such as stock prices, population dynamics, and biometry. To handle stochastic movements, SDEs are used since they arise in modeling including random dynamics. It is possible to study on SDE through two parts namely SDE with no jump and SDE with jump. Data including radical changes or sudden increasing/decreasing is frequently seen in many areas such as economics, biology, chemistry, physics, and social sciences. Measuring these dramatic changes has become more and more important over the years. For instance, a stock price is modeled by Geometric Brownian Motion (GBM) which is SDE with no jump in financial sector. However, when a radical change takes place due to some situations such as wars, natural disasters, market crashes or some dramatic news, it is better to prefer the model which is GBM containing jump terms [2].

Black-Scholes model is among the frequently used model especially in finance. However, this model is unsatisfactory for data including sudden changes and it is only used for continuous sample paths. In 1976, Robert Merton [15] showed a model for stock price including a finite number of discrete jumps. This model can be used for a sample path which is composed of continuous and jump processes. The magnitude of these jumps are normally distributed and their intensity has Poisson distribution. In the thesis, we mainly focus on Merton Jump Diffusion model. We present comparison of Merton model and Black-Scholes counterpart through the models including jump.

This master thesis is planned as follows. In Chapter 2, we present concept of differential equation and SDE. Also, stochastic integral and stochastic process are mentioned. In this chapter, we also talk about numerical approximation methods. In the Chapter 3, we approach to SDEs with jump, discretization procedure and we also give information about well-known jump diffusion models. In Chapter 4, Merton Jump Diffusion model is explained in detail. Model derivation, characteristic function and convolution for transition density are also mentioned. In Chapter 5, we show parameter estimation and comparison of the results obtained in this estimation and the initial values. Parameter estimation is made by using MLE method. Euler-Maruyama convergence and analytical solution for Merton Jump Diffusion model are also controlled. Their convergence check, firstly, is controlled by graphically. Then, it is checked how the convergence is as time interval increases. In the Chapter 6, US Dollar to Turkish Lira exchange rate data is taken for between the date of $01.02.2019$ and $21.06.2019$. US Dollar to Turkish Lira exchange rate data includes sudden changes especially recently. Thus, we consider that the data can be suitable for jump diffusion models. To check this situation, we compare Merton Jump Diffusion model and Black-Scholes framework. Akaike Information Criterion (AIC) values are found for both models and we predict the values for between the date of $23.06.2019$ and $02.07.2019$. Then, two models are compared by checking mean absolute percentage error (MAPE).

CHAPTER 2

STOCHASTIC DIFFERENTIAL EQUATIONS

2.1 Motivation on Differential Equations

In the universe, many phenomena can be described by algebraic methods. These methods are used to evaluate some static situations. That is, value of area, speed, density, price, size etc. can be obtained by these methods. However, the most striking cases generally are related to altered circumstances. For example heat loss, seismic waves detection, fluctuation in population are cases which show change in their situations [4]. At this point, algebraic methods are not enough, and differential equations take place. Differential equations are preferred in many areas such as economy, biology, physics, and engineering. As a philosophical description, a differential equation shows a rate of change in a variable which depends on other variables in the equation. In mathematical description, a differential equation is composed of an unknown function and its derivatives with respect to independent variable [19]. To gain richer understanding, we provide a real life example.

Example 1: Consider a mice population on an island. It is assumed that there are no predators in this place. Under this condition mice population increases at a rate proportional to existing population.

$$\frac{dp(t)}{dt} = rp(t) \tag{2.1}$$

where p symbolizes the existing mice population, t is time in months, and r is growth rate. The equation 2.1 gives the result of change in mice population along the time. From the equation 2.1, it can be seen that the change in mice population is multiplication of growth rate and present mice population.

Consider another scenario for mice population. Suppose that some predators live on this island and hunt 10 mice per day. Then, the differential equation becomes

$$\frac{dp(t)}{dt} = rp(t) - 300 \tag{2.2}$$

As seen in the equation 2.2, the number of hunted mice is given as in day but time is measured in months in the equation. Thus, it is transformed to amount per month which is 300 mice.

Briefly, solving a differential equation means to find the unknown function ($p(t)$ in the above example) satisfying this differential equation.

Differential equations are classified according to several aspects. One of them is related to the number of dependent variables in the equation. According to this, it is divided into two groups namely ordinary differential equations and partial differential equations [26].

2.1.1 Ordinary Differential Equations

An ordinary differential equation (ODE) is defined as a differential equation which contains ordinary derivatives of at least one dependent variable with respect to an independent variable [26].

Suppose that t is an independent variable and $y = f(t)$ is an unknown function. An ODE is generally written as:

$$F(t, y, y', y'', y''', ...y^n) = 0. \tag{2.3}$$

Differential equations can be written by different notations:

- $ay'' + by' = cy$ (Lagrange's notation)
- $af''(x) + bf'(x) = cf(x)$ (Functional notation)
- $a\frac{d^2y}{dx^2} + b\frac{dy}{dx} = cy$ (Leibnitz notation)

Example 2: Suppose that 100 bacteria live in a culture and their population increases with a rate which is proportional to the current number of bacteria. The population doubles in 2 hours. The number of bacteria 5 hours later in that culture can be found in the following way:

Let y be the population of bacteria, t denotes time and so the growth rate of this population is $\frac{dy}{dt}$. If $c \ (c > 0)$ is proportionality constant, then

$$\frac{dy}{dt} = cy. \tag{2.4}$$

Separating the variables,

$$\frac{dy}{y} = cdt. \tag{2.5}$$

Then, we integrate both sides of the equation 2.5 and obtain

$$\ln y = ct + c_1 \tag{2.6}$$

where c_1 is an integration constant. Equivalently, we have

$$\begin{aligned} y_t &= Ae^{ct} \\ A &= e^{c_1} \end{aligned} \tag{2.7}$$

At the beginning, that is at time $t = 0$, the number of bacteria is 100. Hence

$$y(0) = A = 100 \tag{2.8}$$

In 2 hours, the population becomes 200.

$$y(2) = 100e^{2c} = 200 \tag{2.9}$$

which gives $c = \frac{1}{2}\ln 2$. Thus the function $y(t)$ defining the population of bacteria is

$$y(t) = 100e^{(\frac{1}{2}\ln 2)t} = 200 \tag{2.10}$$

Therefore, 5 hours later the population becomes

$$y(5) = 100e^{(\frac{5}{2}\ln 2)t} \approx 566 \tag{2.11}$$

2.1.2 Partial Differential Equations

A partial differential equation (PDE) is defined as a differential equation which contains partial derivatives of at least one dependent variable with respect to more than one independent variable [26]. The general form of a PDE is

$$F(x, y, z, \frac{\partial z}{\partial x}, \frac{\partial z}{\partial y}) = F(x, y, z, a, b) = 0 \tag{2.12}$$

where x, y are independent variables, z is dependent variable, and a and b are partial derivatives of z with respect to x and y respectively. An example is given below to explain clearly.

Example 3: Consider temperature as a function including some parameters such as time, latitude, longitude, and altitude. To find the temperature changing in time, the derivative of temperature function with respect to time is taken by keeping the parameters of latitude, longitude, and altitude as constant.

PDEs are widely used in many disciplines e.g. evolution of gases in fluid dynamics, formation of galaxies, nature of quantum mechanics etc [4].

Solution Sets of Algebraic and Differential Equation: Solution sets of a differential equation and an algebraic equation are different. While an algebraic equation's solution is value or value set, a differential equation solution is function or function set.

Example 4: Suppose there exist two equations. One of them is algebraic equation and the other one is differential equation. Their structures and solutions have the following forms:

- Algebraic Equation:

$$\begin{aligned} &x^2 - 3x + 28 = 0 \\ &x = -4, x = 7 \qquad \text{(solutions of the algebraic equation)} \end{aligned} \tag{2.13}$$

- Differential Equation:

$$\begin{aligned} &y'' + 2y' = 3y \\ &y(x) = c_1 e^{-3x} + c_2 e^{x} \quad \text{(solutions of the differential equation)} \end{aligned} \tag{2.14}$$

  where c_1 and c_2 are constant.

Differential equations can be also divided by their orders. The highest degree of derivative indicates the order of a differential equation [26].

Importance of Stochastic Approaching: Up to now, we mentioned about deterministic process and differential equations to gain better understanding for SDEs. Although many phenomena in nature, technology, economy or some other areas are modeled by deterministic approach, they are not sufficient to describe some cases such as stock prices, population dynamics, and biometry etc. The inadequacy of these approaches arises from omitting stochastic fluctuations. To handle stochastic movements, SDEs are used since they arise in modeling random dynamics [6]. Before talking about SDE, it is better to mention about stochastic process. However, firstly, we will give basic notations of probability theory which are frequently used in stochastic process, stochastic integrals, and other topics in this thesis.

2.2 Basic Notations of Probability Theory

We introduce probability space and random variable in this part. They are frequently used in the stochastic processes. To describe the measure of probability, it is necessary to define probability space. A probability space has a triplet including three parts namely $(\Omega, \mathcal{F}, P)$. They refer to sample space, sigma algebra, and probability measure respectively. Sample space has all outcomes of a random experiment and denoted by Ω. A subset of the outcomes is known as an event. The notation of $\mathcal{F}$ shows set of events. $\mathcal{F}$ is called sigma-algebra if it has the following properties [5].

- $\emptyset \in \mathcal{F}$ where $\emptyset$ is empty set,
- Let $A \in \mathcal{F}$ then $A^{\mathcal{C}} \in \mathcal{F}$ where $A^{\mathcal{C}} = \Omega - A$,
- $\{A_i\}_{i\geq 1} \in \mathcal{F}$ then $\mathrm{U}_{i=1}^{\infty} A_i \in \mathcal{F}$.

The probability P is the set function and it maps A into $[0, 1]$. As long as it satisfies the following requirements, then it is called as probability measure.

- $P(\Omega) = 1$,

- $P(A^C) = 1 - P(A)$,
- $P(U_{i=1}^{n} A_i) = \sum_{i=1}^{n} P(A_i)$, if $A_i \cap A_j = \emptyset$ for $i \neq j$.

Let the probability space have the triplet $(\Omega, \mathcal{F}, P)$ and a random variable X is defined as a measurable function

$$X : \Omega \to R \tag{2.15}$$

2.3 Stochastic Process

Stochastic process is a set of random variables $\{X(t), t \in T\}$ described on a triplet $(\Omega, \mathcal{F}, P)$. That is, time point t in T, $X(t)$ is observed. Due to the fact that it is defined on probability space, it can be said that stochastic process is the probabilistic version of deterministic process [1]. Although exact value at any time in deterministic process is known , it is only known distribution of possible values at any time in stochastic process. Since the values of stochastic process vary in time with an indefinite way, the stochastic process is called discrete or continuous based on its time $t \in T$.

Definition 1. ***(Discrete Time Process)*** $\{X(t), t \in T\}$ *is called discrete time process if the set of* T *is finite or countable. In other words, time points represent specific locations in time space [1].*
e.g: $T = \{0, 1, 2, 3, \ldots\}$ *and* $\{X(0), X(1), X(2), X(3), \ldots\}$ *is discrete time process which is a random number associated with* $t = 0, 1, 2, 3, \ldots$.

Definition 2. ***(Continuous Time Process)*** $\{X(t), t \in T\}$ *is called continuous time process if the set of* T *is infinite or uncountable. Time points can take any positive values [1].*
e.g: $T = [0, \infty)$ *or* $T = [0, c]$ *for some constant c. Thus,* $\{X(t), t \in T\}$ *has a random number* $X(t)$ *associated with every instant in time.*

In this thesis, two well-known stochastic process are frequently used namely Wiener (Brownian) Process and Poisson Process. First one is an example of continuous stochastic process, the last one is an example of discrete stochastic process.

2.3.1 Brownian Motion

The standard Brownian Motion is a continuous stochastic process, also known as Wiener Process, $W(t)$ is widely used in many areas such as physics, biology, finance to model random movements (molecule movements of gas or change in asset price) and satisfies the following three conditions:

1. $W(0) = 0$ and with probability 1 the sample path $t \to W(t;\omega)$ is continuous for every t.

2. The increments $W(t_n) - W(t_{n-1}), W(t_{n-1}) - W(t_{n-2}), ..., W(t_2) - W(t_1)$ are independent random variables for time points $\{t_1, t_2, ..., t_n\}$. (Independent increments).

3. The increments $W(t_n) - W(t_{n-1}), W(t_{n-1}) - W(t_{n-2}), ..., W(t_2) - W(t_1)$ are normally distributed with mean zero and variance $(t_n - t_{n-1}), (t_{n-1} - t_{n-2}), ..., (t_2 - t_1)$ respectively. (Normally distributed increments) [7].

The Brownian motion which has a drift and diffusion coefficients is defined with the SDE given below [1]:

$$dX_t = \mu dt + \sigma dW_t \tag{2.16}$$

where dt and dW_t are increments of time and Wiener process respectively and μ and σ are the drift and diffusion coefficients. The solution X_t has distribution $N(\mu t, \sigma^2 t)$

and the increment $dW_t = W_{t+dt} - W_t$ in equation 2.16 is distributed $N(0, dt)$.
To simulate Brownian paths, the following algorithm can be used:

Algorithm 1 Generation of Brownian Paths

- Let $W(t)$ be Brownian Motion defined on the time interval $[0, T]$. Divide this interval into n parts as $\Delta t = \frac{T}{n}$.

- Take initial value as $W(0) = 0$.

- Produce next states based on $W(t + \Delta t) = W(t) + Z\sqrt{(\Delta t)}$ where Z is the set of random variables which are distributed standard normal.

Example 5: Let $T = 1, n = 100, \Delta t = \frac{T}{n} = 0.01$. Utilizing the algorithm given above a realization of Brownian path is simulated and presented in Figure 2.1.

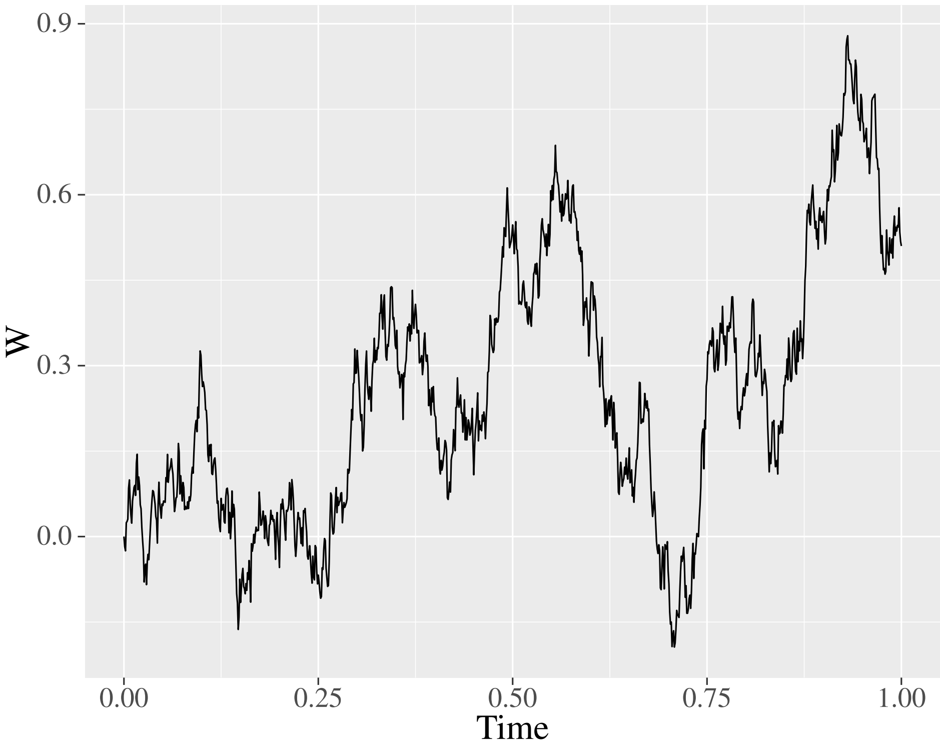

Figure 2.1: Simulated Brownian Path

2.3.1.1 Geometric Brownian Motion

In this thesis, it is focused on the Merton Jump Diffusion model. This model requires GBM structure. Thus, it is useful to give an information about this structure. Brownian motion can take both negative and positive values but sometimes, as in Merton structure, the positive values are only used. In this case, it is preferred to use GBM which is the non-negative variation version of Brownian motion [22]. GBM with drift can be showed as follows:

$$dX_t = \mu X_t dt + \sigma X_t dW_t, \tag{2.17}$$

where μ and σ are drift and diffusion coefficients. X_t is solution of SDE and it has the following form:

$$X(t) = X(0)e^{Y(t)}, \tag{2.18}$$

where $Y(t) = (\mu - \frac{1}{2}\sigma^2)t + \sigma W_t$ is Brownian motion with drift. The solution of SDE is found as:

Separate the variable in the equation 2.17,

$$\frac{dX_t}{X_t} = \mu dt + \sigma dW_t, \tag{2.19}$$

Take the integration of both side,

$$\int \frac{dX_t}{X_t} = \int (\mu dt + \sigma dW_t)dt, \tag{2.20}$$

We know that $\frac{dX_t}{X_t}$ relates to derivative of $\ln(X_t)$ and we obtain the following equation by using Ito calculus,

$$\ln(\frac{dX_t}{X_t}) = (\mu - \frac{1}{2}\sigma^2)t + \sigma W_t, \tag{2.21}$$

Take the exponential of both side,

$$X_t = X_0 \text{exp}((\mu - \frac{1}{2}\sigma^2)t + \sigma W_t) \tag{2.22}$$

When the logarithm of the solution is taken, the following form is produced and it has also Brownian motion structure with normal distribution:

$$Y(t) = \ln(\frac{X(t)}{X(0)}), \tag{2.23}$$

$$\ln(X(t)) = \ln(X(0)) + Y(t) \sim \text{Normal}((\mu - \frac{1}{2}\sigma^2)t + \ln(S(0)), \sigma^2 t). \tag{2.24}$$

For each t, $X(t)$ has a lognormal distribution.

2.3.2 Poisson Process

It is discrete stochastic process, $N(t)$. It is widely used in scenarios when counting the occurrences of certain events with certain occurrence rate. It must satisfy the following conditions:

Let the triplet $(\Omega, \mathcal{F}, P)$ having state space $\mathbb{N} = \{1, 2, 3, ...\}$ and a Poisson Process which is defined on this triplet with parameter λ. This process is the set of $N(t)$ where $t \in [0, \infty)$. A Poisson process must satisfy three crucial requirements.

1. $N(0) = 0$ with probability 1.

2. The increments $N(t_n) - N(t_{n-1}), N(t_{n-1}) - N(t_{n-2}), ..., N(t_2) - N(t_1)$ are independent for $\{t_1, t_2, ..., t_n\}$.

3. Let $0 \leq s < t < \infty$ and the increment $N(t) - N(s)$ has Poisson distribution with parameter λ. Let $k \in \mathbb{N}$ and the distribution of increment explicitly given as follows [17]:

$$P([N(t) - N(s)] = k) = \frac{\lambda^k (t-s)^k}{k!} e^{-\lambda(t-s)}. \tag{2.25}$$

Poisson process is similar to Wiener process as they come from family of Levy processes. We will defined Levy processes in Chapter 5. Both have stationary independent increments. However, they are different from each other with respect to the probability distribution of increments while Wiener process increment is Normally distributed with mean and variance $(0, s - t)$ respectively. Poisson increment has a Poisson distribution with mean $\lambda(s - t)$ [13].

To simulate Poisson paths, the following algorithm can be used:

Algorithm 2 Generation of Poisson Paths, [13]

- Define $T_j \sim \exp(\lambda)$ as interarrival times of random numbers which are generated.

- Find the cumulative sum S_n of those random times up to T which is the last time point.

- So, $N(T) = \min\{n : S_n > T\} - 1$

Example 6: Let intensity parameter $\lambda = 3$ and $T = 4$. Then, the following path in Figure 2.2 can be obtained by using the above algorithm.

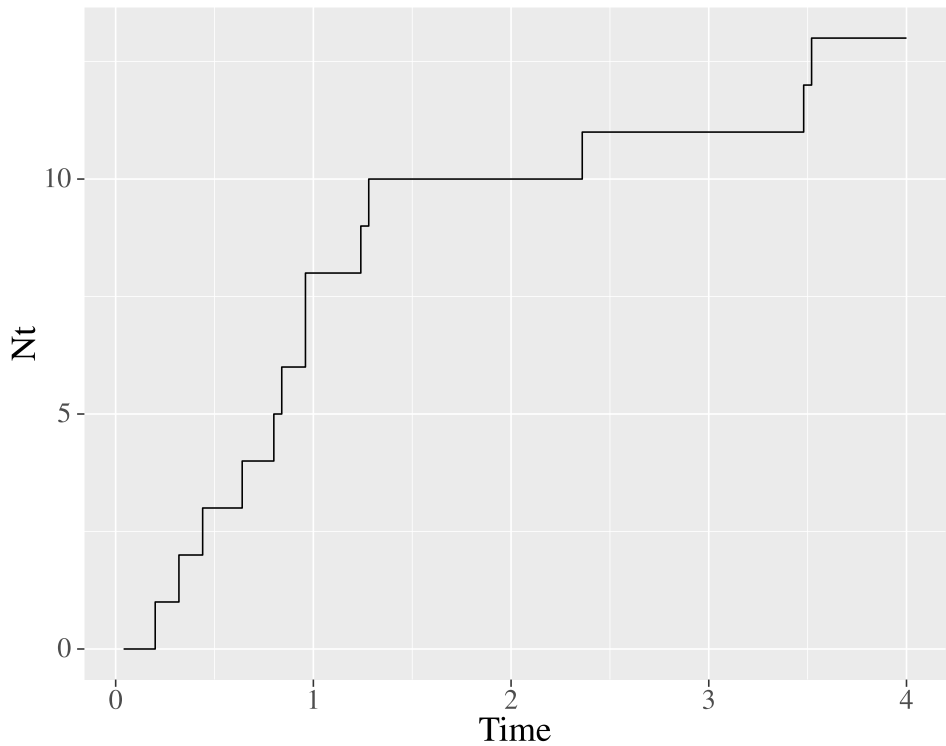

Figure 2.2: Simulated Poisson Path

where N_t is number of jump at time point t. In this thesis, while measuring the jump effect, it is applied to compound Poisson process. Thus, let us now introduce this notation at this point.

2.3.2.1 Compound Poisson Process

$\{Q_i\}_{i\geq 1}$ is a set of random variables which are identically and independently distributed. $\{N_t\}_{t\geq 0}$ symbolize Poisson process with parameter λ. The compound Poisson process $\{CT_t\}_{t\geq 0}$ is described as:

$$CT_t = \sum_{i=1}^{N_t} Q_i. \tag{2.26}$$

The parameter of λ is used as jump intensity in the jump models which are investigated in this thesis. The sum of jumps between t and $t+\Delta t$ has same distribution with $\sum_{i=1}^{\Delta N_t} Q_i$. The number of jumps ΔN_t is distributed $\text{Poisson}(\lambda\Delta t)$. The detailed explanation for simulation of compound Poisson process will be explained in Chapter 3 [21].

Markov Process: A stochastic process is called a Markov process if the conditional probability of future (conditioned on present and past states) depends only on the present state not the past. The processes satisfying this property are also known as memoryless processes [25]. The stochastic process which has independent increments is considered as Markovian process. Therefore, Brownian and Poisson processes, which are previously mentioned in this thesis, have Markov property. Markov property can be described in mathematical form as follows [3]:

$$P^x(X(s+t) \in \mathcal{A}|\mathcal{F}_s) = P^{X(s)}(X(t) \in \mathcal{A}). \tag{2.27}$$

Here the set of $\{P^x\}$ has Markov property in terms of filtration $\mathcal{F}_s$ for each $x \in S$ and every $s, t \geq 0$. S is the state space and $\mathcal{F}_s$ is the filtration of $\mathcal{F}$. $\mathcal{F}_t$ can be assumed as $\sigma - \text{algebra}$ yielded by $\{X(s) : s \leq t\}$. $\{P^x\}$ is the probability measure on $(\Omega, \mathcal{F})$.

Markov chains are required for working on the stochastic process. Suppose that $Y_1, Y_2, ...$ is discrete time process with the corresponding set of values $\{y_1, y_2, ..., y_N\}$. Y_n and Y_{n+1} are present and immediate future states respectively. $Y_1, Y_2, ...Y_{n-1}$ represent past states. Then, Markov property becomes as follows:

$$P\{Y_{n+1}|Y_1, Y_2, ..., Y_n\} = P\{Y_{n+1}|Y_n\}. \tag{2.28}$$

Let P be NxN matrices and its entries are all non-negative and rows sum to 1, $P(n) = [p^{ij}(n)]$ with $n = 1, 2...$ where

$$p^{ij}(n) = P\{Y_{n+1} = y_j|Y_n = y_i\}\text{for} i, j = 1, 2, ..., N \tag{2.29}$$

The process is known as Markov chain and $P(n)$ is accepted as transition matrix. The transition densities have the following properties:

$$0 \leq p^{ij}(n) \leq 1 \tag{2.30}$$

$$\sum_{j=1}^{N} p^{ij}(n) = 1 \tag{2.31}$$

where j=1,2,...,N and n=1,2,3...

2.4 Stochastic Differential Equation

In many cases in nature, finance, technology, it is possible to see using deterministic equations for modeling these cases. However, this modeling ignore stochastic movements and is not appropriate for some areas such as stock prices, population dynamics, and biometry etc. To handle stochastic behaviors, SDEs are used since they arise in modeling including random dynamics.

Definition

Definition 3. *SDE is a form of differential equation including stochastic process. In general, a SDE is defined as following form*

$$\partial X(t) = f(X(t), t; \theta)\partial t + g(X(t), t; \theta)\partial W(t) \tag{2.32}$$

or it can be written in integral form as follows:

$$X(t+s) = X(t) + \int_t^{t+s} f(X(u), u; \theta)\partial u + \int_t^{t+s} g(X(u), u; \theta)\partial W(u) \tag{2.33}$$

where $W(t)_{t\geq 0}$ is a Wiener Process and $\theta \in R^m$ is unknown parameter set [1].

The name of functions f and g are drift and diffusion coefficients. Stochastic process can be called differently such as GBM, Ornstein-Uhlenbeck, etc. This depends on the coefficients of drift and diffusion parts.

In the equation 2.33, there exist two types of integral. First integral is Riemann Stieltjes integral because of the deterministic structure of this term. The other integral is known as Ito or stochastic integral. To solve the stochastic equation, it is necessary to know how to solve these two types of integral.

Firstly, it is necessary to know Riemann Sum to handle these integrals.

Definition 4. *Let us define a function, f, and it is described in $[a, b]$. Divide the interval into n subintervals $[x_0, x_1], [x_1, x_2], ..., [x_{n-1}, x_n]$ where $a = x_0 < x_1 < x_2 ... < x_n = b$. For each $i = 1, 2, 3, ..., n$, it is defined x_i^* in $[x_{i-1}, x_i]$. Then the Riemann sum of the interval $[a, b]$ can be described as follows:*

$$\sum_{i=1}^{n} f(x_i^*)\Delta x_i. \tag{2.34}$$

2.4.1 Riemann Stieltjes Integral

The definite integral of a continuous function f on interval $[a, b]$ can be given as:

$$\int_a^b f(x)dx = \lim_{n\to\infty} \sum_{i=1}^{n} f(x_i^*)\Delta x_i \tag{2.35}$$

for any choice of x_i^* in $[x_{i-1}, x_i]$ where $x_i = a + i\Delta x$ with $\Delta x = \frac{b-a}{n}$. Here, there exist n sub-intervals and their lengths are not necessarily equal.

The first integral in the equation 2.33 can be solved by this method. The corresponding Riemann integral as follows:

$$\int_a^b f(t)dx = \lim_{\Delta t\to\ 0} \sum_{i=1}^{n} f(t_i^*)\Delta t_i \tag{2.36}$$

where $\Delta t_i = t_i - t_{i-1}$, $t_{i-1} \leq t_i^* \leq t_i$, $a = t_0 < t_1 < t_2 ... < t_n = b$ [16].

2.4.2 Stochastic Integral

A random variable is called the stochastic integral if it satisfies the following condition:

$$\int_a^b g(t)dW_t = \lim_{\Delta t\to\ 0} \sum_{i=1}^{n} g(t_{i-1})\Delta W_i$$

or it can be given as follows: (2.37)

$$\lim_{n\to\infty} E[(\int_a^b g(t)dW_t - \sum_{i=1}^{n} g(t_{i-1})\Delta W_i)] = 0$$

where $E(\cdot)$ is expectation and $\Delta W_i = W_{t_i} - W_{t_{i-1}}$, $t_{i-1} \leq t_i^* \leq t_i$, $a = t_0 < t_1 < t_2 ... < t_n = b$ [16].

In the first integral which is calculated by Riemann integral, any point in the interval (t_{i-1}, t_i) can be used to evaluate the integral but the left end point for same interval is used for stochastic integral. This difference is due to having random variables of the function $g(t)$, W_t, and solution of stochastic integral. Stochastic integral cannot be solved in classical way due to the unbounded variation. This process is almost surely non-differentiable.

It is necessary to be known chain rule to obtain analytic solution of SDE. This rule is also known as Ito Formula.

2.4.3 Ito Calculus

Suppose that X_t is Ito process, that is, it satisfies the following SDE [12]:

$$dX_t = \mu_t d_t + \sigma_t dW_t. \tag{2.38}$$

Now, let the function $f(t, X_t) : [0, \infty) \times R$ be a twice differentiable function with respect to X and differentiable with respect to t and let $Z_t = f(t, X_t)$ then following Ito calculus

$$\begin{aligned} dZ_t &= \frac{\partial f}{\partial t}(t, X_t)dt + \frac{\partial f}{\partial X}(t, X_t)dX_t + \frac{1}{2}\frac{\partial^2 f}{\partial X^2}(t, X_t)dX_t dX_t \\ &= (\frac{\partial f}{\partial t}(t, X_t) + \frac{\partial f}{\partial X}(t, X_t)\mu_t + \frac{1}{2}\frac{\partial^2 f}{\partial X^2}(t, X_t)(\sigma_t)^2)d_t + \frac{\partial f}{\partial X}(t, X_t)\sigma_t dW_t. \end{aligned} \tag{2.39}$$

Compared to deterministic differential equation the SDE given above has the following extra term,

$$\frac{1}{2}\frac{\partial^2 f}{\partial X^2}(t, X_t)dX_t dX_t \tag{2.40}$$

and here $dX_t dX_t$ is analyzed using the identities given below:

$$\begin{aligned} & dtdt = dtdW_t = dW_t dt = 0 \\ & dW_t dW_t = dt. \end{aligned} \tag{2.41}$$

The extra term in mention in equation 2.40 is observed from the quadratic variation and covariation. To understand the application of Ito calculus for SDEs better, we present the following examples [24].

Example 7: Suppose that we have a GBM, that is,

$$X_t = X_0 \text{exp}^{\mu_t + \sigma W_t} \tag{2.42}$$

where X_0 is a constant. Equivalently, $X_t = f(t, x) = X_0 \text{exp}^{\mu_t + \sigma x}$. According to this form, we have

- $f'_t = X_0 \text{exp}^{\mu_t + \sigma x}\mu$
- $f'_x = X_0 \text{exp}^{\mu_t + \sigma x}\sigma$
- $f''_{xx} = X_0 \text{exp}^{\mu_t + \sigma x}\sigma^2$

Now, we apply the Ito formula to obtain the SDE for dX_t and

$$\begin{aligned} dX_t &= (X_0 \mu \text{exp}^{\mu t + \sigma W_t} + \frac{1}{2}X_0 \sigma^2 \text{exp}^{\mu t + \sigma W_t})dt + X_0 \sigma \text{exp}^{\mu t + \sigma W_t}. \\ &= (\mu + \frac{1}{2}\sigma^2)X_t dt + \sigma X_t dW_t. \end{aligned} \tag{2.43}$$

Example 8: Let SDE has the form:

$$dZ = 3W_t^2 dW_t + 3W_t dt. \tag{2.44}$$

Now, to show solution of the SDE as $Z = W_t^3$, take the function on $Y = f(t, x) = x^3$

According to the given function, the elements of Ito formula are:

- $f_t' = 0,$
- $f_x' = 3x^2,$
- $f_{xx}'' = 6x$

Then, the solution of the SDE is:

$$\begin{aligned} dY &= \frac{df}{dt}(t,x)dt + \frac{df}{dx}(t,x)dx + \frac{1}{2}\frac{d^2 f}{d^2 x}(t,X)dxdx \\ &= 0 + 3x^2 dx + \frac{1}{2}6xdxdx \\ &= 3x^2 dx + 3xdxdx \\ &= 3W_t^2 dW_t + 3W_t dW_t dW_t \\ &= 3W_t^2 dW_t + 3W_t dt \\ &= 3W_t^2 dW_t + 3W_t dt \end{aligned} \tag{2.45}$$

In these examples, analytical solutions of SDEs can be found. However, the exact solution of SDE is generally difficult to obtain. Most of the SDEs do not have closed form solutions. At this time, it is useful to approximate the solution by using some numerical approximation techniques such as Euler-Maruyama, Milstein and Runge-Kutta. In this thesis, only the Euler-Maruyama method is used, therefore, we present the following introductory explanation for this method.

Euler-Maruyama Method: It is the well-known numerical approximation method for SDEs. When Ito's formula of the stochastic Taylor series after the first order terms is truncated, the Euler-Maruyama method is obtained. Suppose that there exists a differential equation in the following form:

$$\partial X(t) = f(X(t), t; \theta)\partial t + g(X(t), t; \theta)\partial W(t), \tag{2.46}$$

where the initial condition is $X(0) = X_0$ and $0 \leq t \leq T$. To apply Euler-Maruyama method, initially, the interval of $[0, T]$ must be discretized. Let $\Delta t = \frac{T}{N}$ for some positive integer N and $\tau_j = j\Delta t$. The numerical approximation to $X(\tau_j)$ is denoted by X_j. Then, from the Euler-Maruyama method we have [20]:

$$X_j = X_{j-1} + f(X_{j-1})\Delta t + g(X_{j-1})(W(\tau_j) - W(\tau_{j-1})), \tag{2.47}$$

where f and g are scalar functions with the initial condition $X(0)$, and $j = 1, ..., N$. In order to obtain above form the following approximation is used:

$$\begin{aligned} \int_{\tau_{j-1}}^{\tau_j} g(s, X_s) dW_s &\approx g(\tau_{j-1}, X_{j-1})\Delta W_{j-1}, \\ \int_{\tau_{j-1}}^{\tau_j} f(s, X_s) ds &\approx f(\tau_{j-1}, X_{j-1})\Delta t. \end{aligned} \tag{2.48}$$

In this thesis, the discretized Brownian paths and Poisson paths will be computed and will be used to generate the corresponding increments in the Euler-Maruyama form for jump diffusion model. The detailed explanation will be given in Chapter 3.

CHAPTER 3

STOCHASTIC DIFFERENTIAL EQUATION WITH JUMP

3.1 Motivation on Stochastic Differential Equation with Jump

Up to now, we have dealth with SDEs with no jump. However, in many cases, we encounter data including radical changes or sudden increasing/decreasing. Measuring these dramatic changes has become more and more important over the years. Thus, SDEs with jump are used in many areas such as economics, biology, chemistry, physics and social sciences, etc. For example, in financial sector, a stock price is modeled by GBM. However, when it is encountered a radical change due to some reasons such as wars, natural disasters, market crashes or some dramatic news, it is better to prefer the model which is Geometric Brownian Motion containing jump terms [2].

Definition 5. *[10] SDE with jump can be showed in differential form as follows:*

$$dX(t) = f(X(t), t)dt + g(X(t), t)dW_t + h(X(t), t)dN_t, \quad X(0) = X_0, \tag{3.1}$$

or it can be written in integral form as:

$$X(t) = X(0) + \int_0^t f(X(s), s)ds + \int_0^t g(X(s), s)dW_s + \int_0^t h(X(s), s)dN_s, \tag{3.2}$$

where N_t is Poisson proocess and W_t is Brownian motion.

Theorem 1. *[23] Solution of a SDE with jump, X(t), can be written as summation of a drift term, a Brownian stochastic integral, and compound Poisson process:*

$$X_t = X_0 + \int_0^t f(X(s), s)ds + \int_0^t g(X(s), s)dW_s + \sum_{i=1}^{N_t} Q_i, \tag{3.3}$$

where f and g are continuous functions, N_t and W_t are Poisson process and Brownian motion respectively and Q_i is the jump size.

The equation 3.1 can be rewritten in different form to analyze the structure of jump diffusion model in detail. Suppose that $(\Omega, \mathcal{F}, P)$ is the triplet which shows a probability measure and Markov process X on a domain $D \to \mathbb{R}^d$ is a solution of SDE with jump. Then, the SDE is [8]:

$$dX_t = \mu(X_t)dt + \sigma(X_t)dW_t + \int_M \Delta(X_{t^-}, z)p(dz, dt), X_0 = x_0 \tag{3.4}$$

where $\mu : D \to \mathbb{R}^d$, $\sigma : D \to \mathbb{R}^{d\times m}$ and $D \times M \to \mathbb{R}^d$ are deterministic functions, W is an m-dimensional standard Brownian motion, d and m are bigger than or equal to 1, $p(dz, dt)$ is a random counting measure on $M \times (0, \infty)$ with M which is a subset of Euclidean space, and $p(dz, dt)$ has $\lambda_t(dz)$ as intensity measure. This measure can be written as $\Lambda(X_t)\nu(dz)$, where $\Lambda : D \to \mathbb{R}_{++}$ and ν is probability measure on M. $\Lambda(X)$ is jump arrival rate and ν is the distribution jump size.

3.2 Exact Solution for Jump Process

As seen in the SDE with no jump, there exists also analytical solution for jump model. To find this solution, stochastic integral is again needed. Although an analytical solution is generally difficult to obtain, we will explain how to construct an analytical solution structure. Before talking about stochastic integral for jump models, it is better to mention about notion of stochastic process in jump cases.

3.2.1 Stochastic Jump Process

Suppose there exists a function h and it is defined on $[0, T] \to \mathbb{R}^n$. It is said to be right continuous with left limit at $t \in [0, T]$ if

$$h(t^+) := \lim_{s\to\, t^+} h(s) \text{ and } h(t^-) := \lim_{s\to\, t^-} h(s) \text{ exist and } h(t^+) = h(t).$$

It is said to be left continuous and right limit at $t \in [0, T]$ if

$$h(t^+) := \lim_{s\to\, t^+} h(s) \text{ and } h(t^-) := \lim_{s\to\, t^-} h(s) \text{ exist and } h(t^-) = h(t).$$

Definition 6. *[23] If h is right continuous with left limit at t, then $\Delta h(t) = h(t) - h(t^-)$ is called jump of h at t. If h is left continuous with right limit at t, then $\Delta h(t) =$*

$h(t^+) - h(t)$ is called jump of h at t. Then, a stochastic process $X = (X_t)_{t \geq 0}$ is known as jump process when the sample paths $s \to X_s$ is left or right continuous for every s.

In this thesis, Poisson and compound Poisson process are used as stochastic jump processes. The explanations for them are given in Chapter 2.

3.2.2 Stochastic Integral for Jump Process

The stochastic or Ito jump diffusion process can be written in the following form

$$X_t = X_0 + \int_0^t f(X_s, s; \theta) ds + \int_0^t g(X_s, s; \theta) dW_s + \int_0^t h(X_s, s; \theta) dN_s. \quad (3.5)$$

Here f, g and h are coefficients of drift, diffusion and jump respectively. W_t is a Brownian motion and N_t is a Poisson process. In order to obtain solution for SDE with jump, Ito calculus is needed. Ito calculus for jump process is written for equation 3.5 [17]:

$$\begin{aligned} k(X_t) =& k(X_0) + \int_0^t \frac{\partial k}{\partial x}(X_s) f_s ds + \frac{1}{2} \int_0^t \frac{\partial^2 k}{\partial x^2}(X_s) g_s^2 ds + \int_0^t \frac{\partial k}{\partial x}(g_s) dW_s \\ &+ \int_0^t (k(X_{s^-} + h(X_s)) - k(X_{s^-})) dN_s. \end{aligned} \quad (3.6)$$

3.3 Numeric Solution For Jump Process

As in the no jump stochastic models, the approximation methods are alternative to analytical solutions when the explicit solution does not exist or is hard to obtain.

3.3.1 Discretization Scheme

One way is to discretize the jump diffusion using standard discretization scheme. If the jump intensity is deterministic or bounded, this way can be used. However, many models include unbounded jump intensity. For this situation, the different approach can be used. Let us explain this approach briefly.

Building Jump-Diffusion with Time-Scaling: [8] Suppose that $(\Omega, \mathcal{F}, P)$ is the triplet which shows a probability measure and $\{\varepsilon_n, Z_n\}$ is a set of random variables defined on this probability space. Here ε_n is standard exponentially distributed and Z_n is distribution of jump size. The jumps are represented as an increasing sequence $0 = \tau_0 < \tau_1 < \tau_2 < \tau_3 < ...$ recursively according to

$$X_t = X_{\tau_n} + \int_t^{\tau_n} \mu(X_s)ds + \int_t^{\tau_n} \sigma(X_s)dW_s \tag{3.7}$$

for $t \in [\tau_n, \tau_{n+1})$, with

$$\tau_{n+1} = \inf\{t > \tau_n : \int_{\tau_n}^t \Lambda(X_s)ds \geq \varepsilon_{n+1}\} \tag{3.8}$$

and the jump update is

$$X_{\tau_{n+1}} = X_{\tau_{n+1^-}} + \Delta(X_{\tau_{n+1^-}}, Z_{n+1}). \tag{3.9}$$

Let $p(dz, dt)$ be the random counting measure and it has intensity measure as a form of $\Lambda(X_t)\nu(dz)$. There exist a process A which is defined by

$$A_t = \int_0^t \Lambda(X_s)ds \tag{3.10}$$

and has continuous and increasing sample paths. $A_s^{-1} = \inf\{t : A_t \geq s\}$ is the inverse of the process A which is also continuous and increasing. Then, it can be said that A_s^{-1} describes a stochastic change of time.

The Process of Discretization: [8] Suppose that X^h and A^h are Euler approximations of X and A. Their initial values are X_0 and 0 respectively. The step size is $h = \frac{T}{N_{step}}$. The discretization times and approximate jump times are symbolized by t and τ_n^h. The approximations are

$$X^h_{t_{i+1^-}} = X^h_{t_i} + \mu(X^h_{t_i})(t_{i+1} - t_i) + \sigma(X^h_{t_i})(W_{t_{i+1}} - W_{t_i}) \tag{3.11}$$

$$A^h_{t_{i+1}} = A^h_{t_i} + \Lambda(X^h_{t_i})(t_{i+1} - t_i) \tag{3.12}$$

for $t \in [t_i, t_{i+1})$. The n^{th} approximate jump time is

$$\tau_n^h = \inf\{t : A_t^h \geq E_n\} \tag{3.13}$$

where $E_n = \sum_{k \leq n} \varepsilon_k$ and $k = 1, 2, ...$ are identically independent distributed (i.i.d.) standard exponential random variables. Then the approximate jump time can be written as

$$\tau_n^h = \eta_n^h + \frac{E_n - A^h_{\eta_n^h}}{\Lambda(X^h_{\eta_n^h})}, \tag{3.14}$$

where $\eta_n^h = \inf\{t_i : A_{t_i}^h + \lambda(X_{t_i}^h)((\lfloor \frac{t_i}{h} \rfloor + 1)h - t_i) > E_n\}$ is the last discretization time before the jump. At τ_n^h, the jump is updated according to

$$X_{\tau_n^h}^h = X_{\tau_n^{h-}}^h + \Delta(X_{\tau_n^{h-}}^h, Z_n). \tag{3.15}$$

Algorithm for Simulating Data: [8] Suppose that $\{Z_j, j = 1, 2, ...\}$, $\{\varepsilon_n, n = 1, 2, ...\}$ are sequences of i.i.d. standard normal and standard exponential random variables respectively and $\{Q_n, n = 1, 2, ...\}$ presents jump sizes with some distribution ν. Then the complete discretization algorithm is given as follows:

Algorithm 3 Discretization Algorithm, [8]

---

1. Set $i \leftarrow 0$, $j \leftarrow 0$, $n \leftarrow 0$, $s \leftarrow 0$, $X_s^h \leftarrow x_0$, $A_s^h \leftarrow 0$ and $E \leftarrow \varepsilon_1$.
2. Compute $A_{temp}^h = A_s^h + \Lambda(X_s^h)((i+1)h - s)$.
3. $A_{temp}^h \geq E$, a jump has occured between s and $(i+1)h$, so
   - Compute $\tau_n^h = s + \frac{E - A_s^h}{\Lambda(X_s^h)}$,
   - Compute $X_{\tau_n^{h-}}^h = X_s^h + \mu(X_s^h)(\tau_n^h - s) + \sigma(X_s^h)\sqrt{\tau_n^h - s}Z_j$,
   - Compute $X_{\tau_n^h}^h = X_{\tau_n^{h-}}^h + \Delta(X_{\tau_n^{h-}}^h, Q_n)$,
   - Set $s \leftarrow \tau_n^h$, $A_s^h \leftarrow E$, $n \leftarrow n + 1$ and $E \leftarrow E + \varepsilon_n$,

   else no jump occured between s and $(i+1)h$, so

   - Compute $X_{(i+1)h}^h = X_s^h + \mu(X_s^h)((i+1)h - s) + \sigma(X_s^h)\sqrt{(i+1)h - s}Z_j$,
   - Set $S \leftarrow (i+1)h$, $A_s^h \leftarrow A_{temp}^h$ and $i \leftarrow i + 1$.
4. $j \leftarrow j + 1$. When $s = T$ simulation has completed, otherwise go to step 2.

3.4 Jump Diffusion Models

Jump diffusion models have skill to capture heavy tail characteristic of observations. If excessive kurtosis (heavy tails) is seen in the distribution, jump models can be a valuable option for modeling. These models are preferred to handle discontinuous behavior in distribution of observation caused by radical or unexpected events. At this point, we would like to explain two famous jump diffusion models:

3.4.1 Merton Model

This model includes jump and diffusion components. The diffusion component comes from family of GBM. The jump term has lognormal jumps driven by Poisson process. SDE for Merton Jump Diffusion model can be described as following form:

$$dX_t = (\alpha - \lambda k)X_t dt + \sigma X_t dW_t + (y_t - 1)X_t dN_t. \tag{3.16}$$

Here W_t and N_t are Brownian motion and Poisson process respectively, λ is intensity of Poisson process, α is instantaneous expected return, $y_t - 1$ is relative jump size and its mean is k, σ is diffusion term. The solution for Merton Jump Diffusion model is:

$$X_t = X_0 \exp[(\alpha - \frac{\sigma^2}{2} - \lambda k)t + \sigma W_t + \sum_{i=1}^{N_t} Q_i], \tag{3.17}$$

where α, σ are the instantaneous expected return and diffusion volatility terms [14]. W_t and N_t are standard Brownian motion and Poisson process respectively. $\sum_{i=1}^{N_t} Q_i$ is compound Poisson jump process [14]. Merton model will be explained in detail in Chapter 4.

3.4.2 Pareto-Beta Model

This model is composed of Pareto and Beta distributions. Sometimes negative or positive jump magnitudes have different effects. For example, in financial sector, people respond differently according to news. Bad news and good news take different reaction. Bad news have more effect than good news. The reason is that people are in panic when the spectacular movements occurred and this causes more destructive chain reaction. To evaluate these differences, this model can be preferred. This jump model assumes that the good news use Pareto distribution, and the bad news use Beta distributions. Both are generated by Poisson process but magnitudes comes from these two distributions. We have

$$\frac{dX(t)}{dX(t-)} = \mu dt + \sigma dW(t) + \sum_{j=u,d} (V^j_{N^j(\lambda^j t)} - 1) dN^j(\lambda^j t), \tag{3.18}$$

where μ, σ, W, V^j are drift, diffusion, Brownian motion, and the jump magnitude respectively. $N^j(\lambda^j)$ is independent Poisson process. The symbol j represent upward and downward jump and can take values u and d. Upward jumps are distributed Pareto and downward jumps are distributed Beta [18].

- The up-jump magnitudes V^u has $\text{Pareto}(\eta_u)$ distribution and its density is given as:

$$f_{V^u}(x) = \frac{\eta_u}{X^{\eta_u+1}}, V^u \geq 1. \tag{3.19}$$

- The down-jump magnitudes V^d has $\text{Beta}(\eta_d, 1)$ distribution and its density is given as:

$$f_{V^d}(x) = \eta_d X^{\eta_d - 1}, 0 < V_d < 1. \tag{3.20}$$

CHAPTER 4

MERTON JUMP DIFFUSION MODEL

In this Chapter, Merton Jump Diffusion model will be explained in detail. In this thesis, we mainly focus on this model and we do the analysis based on it. This model is created by Robert Merton in 1976. He aimed to model the stock price behavior including small diffusive movements with large random jumps. The model overcomes the problem of crash scenarios which cause dramatic effect. For this model, Matsuda's paper [14] is used.

4.1 Model Type

Merton jump diffusion model is a member of Levy family which is a family of Levy process. Levy process is stochastic process having stationarity and independence increments with right continuous and left limits paths. Levy process distribution is characterized by its characteristic function given in the Levy–Khintchine representation. This representation is described as follows:

Definition 7. *Levy–Khintchine representation is an expression of a characteristic function* $\phi(\omega)$ *and a characteristic exponent* $\psi(\omega)$ *of a finite variation Levy process* $\{X_t; t \geq 0\}$*:*

$$\phi(\omega) = E[\exp(i_t)] = \exp[t\psi(\omega)] \tag{4.1}$$

$$\psi(\omega) = ib\omega - \frac{\sigma^2\omega^2}{2} + \int_{-\infty}^{\infty} \{\exp(i\omega x) - 1\} l(dx) \tag{4.2}$$

where $l(dx) = \lambda f(dx)$ *with* λ *is jump intensity and* $f(dx)$ *is jump size density which is called Levy measure.*

Merton Jump Diffusion has the following form:

$$X_t = X_0 e^{L_t} \tag{4.3}$$

where the stock price process $\{X_t; 0 \leq t \leq T\}$ is constructed as an exponential of a Levy process $\{L_t; 0 \leq t \leq T\}$. This process includes two parts. One part is a continuous diffusion process which is Brownian motion with drift. The other part is discontinuous jump process which is represented by compound Poisson process. This Levy process has the following equality:

$$L_t = (\alpha - \frac{\sigma^2}{2} - \lambda k)t + \sigma W_t + \sum_{i=1}^{N_t} Q_i \tag{4.4}$$

where $\{W_t; 0 \leq t \leq T\}$ is a standard Brownian motion process. The part of $(\alpha - \frac{\sigma^2}{2} - \lambda k)t + \sigma W_t$ is called as Brownian motion with drift process and $\sum_{i=1}^{N_t} Q_i$ is known as compound Poisson jump process. $\sum_{i=1}^{N_t} Q_i$ is only difference between the Black-Scholes and Merton Jump Diffusion models. This term includes two sources of randomness which are random timing and random jump size. Random timing means that the asset price randomly jumps according to Poisson process dN_t with intensity λ which is average number of jumps per unit of time. Also, when the jump occurs its magnitude is important. Merton assumes that log stock prices jump size, (dx_i), has normal distribution with parameter μ and δ^2:

$$f(dx_i) = \frac{1}{\sqrt{2\pi\delta^2}} \exp\{-\frac{(dx_i - \mu)^2}{2\delta^2}\}. \tag{4.5}$$

Merton says that these two sources of randomness are independent. In Black-Scholes model, there exist two parameters for drift and diffusion terms. However, in Merton Jump Diffusion model, there are three additional parameters λ, μ, δ compared to the Black-Scholes model. Merton Jump Diffusion model is used to handle negative skewness and excess kurtosis of the log return density.

When the intensity is multiplied by jump size density, Levy measure $l(dx)$ of a compound Poisson process is obtained,

$$l(dx) = \lambda f(dx). \tag{4.6}$$

If the Levy measure $l(dx)$ is finite (i.e. number of jumps per unit time is finite), then a compound Poisson process is known as finite activity Levy process that is

$$\int_{-\infty}^{\infty} l(dx) = \lambda < \infty. \tag{4.7}$$

An asset price X_t model is family of exponential Levy process L_t. That is, log-return of X_t, $\ln(\frac{X_t}{X_0})$, can be defined as a Levy process such that

$$\ln(\frac{X_t}{X_0}) = L_t = (\alpha - \frac{\sigma^2}{2} - \lambda k)t + \sigma W_t + \sum_{i=1}^{N_t} Q_i. \tag{4.8}$$

4.2 Model Derivation

In Merton Jump Diffusion model, variation in asset price has continuous diffusion and discontinuous jump components. First one is normal component which is represented by Brownian motion with drift process and second one is abnormal component which is modeled by compound Poisson process. The jumps in the model are accepted as independently and identically random variables. The asset price jumps in tiny time interval, dt, can be showed by using Poisson process dN_t such that

$$\text{P(number of jumps in } dt\text{)=} \begin{cases} P\{dN_t = 1\} \cong \lambda dt \\ P\{dN_t \geq 2\} \cong 0 \\ P\{dN_t = 0\} \cong 1 - \lambda dt \end{cases}$$

where $\lambda \in \mathbb{R}^+$ is intensity of the jump process. The asset price jumps occur from X_t to $y_t S_t$ in the tiny time interval dt. Thus, the relative price jump size is

$$\frac{dX_t}{X_t} = \frac{y_t X_t - X_t}{X_t} = y_t - 1, \tag{4.9}$$

where y_t comes from the lognormal distribution. That is,

$$\begin{aligned} &\ln(y_t) \sim N(\mu, \delta^2), \\ &E[y_t] = e^{\mu + \frac{1}{2}\delta^2}, \\ &E[(y_t - E[y_t])^2] = e^{2\mu+\delta^2}(e^{\delta^2}), \\ &y_t \sim \text{Lognormal}(e^{\mu+\frac{1}{2}\delta^2}, e^{2\mu+\delta^2}(e^{\delta^2})). \end{aligned} \tag{4.10}$$

When the above properties are considered, the asset price Merton Jump Diffusion model takes the form as:

$$\frac{dX_t}{X_t} = (\alpha - \lambda k)dt + \sigma dW_t + (y_t - 1)dN_t, \tag{4.11}$$

where α, σ are the instantaneous expected return and volatility of the asset respectively. W_t and N_t are standard Brownian motion and Poisson process respectively.

The term of $y_t - 1$ represents relative price jump size. It is accepted that W_t, N_t, and $y_t - 1$ are independent. The relative price jump size $y_t - 1$ has Lognormal distribution with mean $E[y_t - 1] = e^{\mu+\frac{1}{2}\delta^2} - 1 = k$ and the variance $E[(y_t - 1 - E[y_t - 1])^2] = e^{2\mu+\delta^2}(e^{\delta^2} - 1)$, that is,

$$(y_t - 1) \overset{\text{i.i.d.}}{\sim} Lognormal(k = e^{\mu+\frac{1}{2}\delta^2} - 1, e^{2\mu+\delta^2}(e^{\delta^2} - 1)). \qquad (4.12)$$

In other words, Merton considers that the log price jump size $\ln y_t = Q_t$ and log-return jump size $\ln(\frac{y_t X_t}{X_t})$ is normally distributed as

$$\ln(\frac{y_t X_t}{X_t}) = \ln(y_t) = Q_t \overset{\text{i.i.d.}}{\sim} \text{Normal}(\mu, \delta^2) \qquad (4.13)$$

At this point, it should be noted that

$$E[y_t - 1] = e^{\mu+\frac{1}{2}\delta^2} - 1 = k \quad \neq \quad E[\ln(y_t)] = \mu \qquad (4.14)$$

due to

$$\ln E[y_t - 1] \neq E[\ln(y_t - 1)] = E[\ln(y_t)]. \qquad (4.15)$$

For the jump part dN_t in tiny time interval dt, the expected relative price change is $\lambda k d_t$ because $E[(y_t - 1)dN_t] = E[y_t - 1]E[dN_t] = k\lambda d_t$. It is called the predictable part of the jump. Thus, the instantaneous expected return on the asset αdt is adjusted by $-\lambda k dt$. By this way, it can be motivated on unpredictable part of the jump as follows:

$$\begin{aligned} E[\frac{dX_t}{X_t}] &= E[(\alpha - \lambda k)dt] + E[\sigma dW_t] + E[(y_t - 1)dN_t], \\ &= (\alpha - \lambda k)dt + 0 + \lambda k dt = \alpha dt. \end{aligned} \qquad (4.16)$$

When the asset price has no jump in time interval dt, then the jump diffusion process turns into the Brownian motion with drift process

$$\frac{dX_t}{X_t} = (\alpha - \lambda k)dt + \sigma dW_t. \qquad (4.17)$$

When the asset price jump is one in dt, we have

$$\frac{dX_t}{X_t} = (\alpha - \lambda k)dt + \sigma dW_t + (y_t - 1) \qquad (4.18)$$

Solution For Merton Jump Diffusion Model: Merton Jump Diffusion model in equation 4.11 can be rewritten as

$$dX_t = (\alpha - \lambda k)X_t dt + \sigma X_t dW_t + (y_t - 1)X_t dN_t. \qquad (4.19)$$

Cont and Tankov [23] suggest the Ito formula for the jump diffusion process as:

$$\begin{aligned} df(X_t,t) =& \frac{\partial f(X_t,t)}{\partial t}dt + b_t \frac{\partial f(X_t,t)}{\partial x}dt + \frac{\sigma^2}{2}\frac{\partial^2 f(X_t,t)}{\partial x^2}dt \\ &+ \sigma_t \frac{\partial f(X_t,t)}{\partial x}dW_t + [f(X_{t^-} + \Delta X_t) - f(X_{t^-})] \end{aligned} \tag{4.20}$$

where b_t, σ_t are drift and diffusion terms of jump diffusion process respectively. It has the following form:

$$X_t = X_0 + \int_0^t b_s ds + \int_0^t \sigma_s dW_s + \sum_{i=1}^{N_t} Q_i. \tag{4.21}$$

By using the Ito calculus for Merton Jump Diffusion model, we obtain

$$\begin{aligned} d\ln X_t =& \frac{\partial \ln X_t}{\partial t}dt + (\alpha - \lambda k)X_t \frac{\partial \ln X_t}{\partial X_t}dt + \frac{\sigma^2 X_t^2}{2}\frac{\partial^2 \ln X_t}{\partial X_t^2}dt + \sigma X_t \frac{\partial \ln X_t}{\partial X_t}dW_t \\ &+ [\ln y_t X_t - \ln X_t] \\ &= (\alpha - \lambda k)X_t \frac{1}{X_t}dt + \frac{\sigma^2 X_t^2}{2}(-\frac{1}{X_t^2})dt + \sigma X_t \frac{1}{X_t}dW_t + [\ln y_t + \ln X_t - \ln X_t] \\ &= (\alpha - \lambda k)dt - \frac{\sigma^2}{2}dt + \sigma dW_t + \ln y_t \end{aligned}$$

Hence,

$$\ln X_t - \ln X_0 = (\alpha - \frac{\sigma^2}{2} - \lambda k)(t-0) + \sigma_t(W_t - W_0) + \sum_{i=1}^{N_t} \ln y_i$$

We have

$$\exp(\ln X_t) = \exp\left\{ \ln X_0 + (\alpha - \frac{\sigma^2}{2} - \lambda k)t + \sigma_t W_t + \sum_{i=1}^{N_t} \ln y_i \right\}$$

yielding

$$X_t = X_0 \exp[(\alpha - \frac{\sigma^2}{2} - \lambda k)t + \sigma W_t] \prod_{i=1}^{N_t} y_i.$$

All in all, the exact solution of Merton Jump Diffusion model can be written as follow:

$$X_t = X_0 \exp[(\alpha - \frac{\sigma^2}{2} - \lambda k)t + \sigma W_t + \sum_{i=1}^{N_t} Q_i] \tag{4.22}$$

where $\ln(y_i) = Q_i$. As it is seen in the solution structure, the process $\{X_t; 0 \leq t \leq T\}$ is modeled as an exponential Levy model of the form.

$$X_t = X_0 e^{L_t} \tag{4.23}$$

where $L_t = (\alpha - \frac{\sigma^2}{2} - \lambda k)t + \sigma W_t + \sum_{i=1}^{N_t} Q_i$. The equation 4.22 can be also written in the following form:

$$X_t = X_{t-1}\exp[(\alpha - \frac{\sigma^2}{2} - \lambda k)d_t + \sigma dW_t + \sum_{i=1}^{N_{d_t}} Q_i] \tag{4.24}$$

where $d_t = t_n - t_{n-1}$ and $dW_t = W_t - W_{t-1}$ are time increment and Brownian increment respectively. In the model assumption assessing, we use this structure.

4.3 Convolution For Transition Density in Merton Model

The classical Black-Scholes model suggests that log return $\ln(\frac{X_t}{X_0})$ has normal distribution as follows:

$$\ln(\frac{X_t}{X_0}) \sim \text{Normal}[(\alpha - \frac{\sigma^2}{2})t, \sigma^2 t] \tag{4.25}$$

However, in the Merton Jump Diffusion model, the compound Poisson jump process $\sum_{i=1}^{N_t} Q_i$ makes log return non-normal. In Merton model, as a result of having normal distribution of log-return jump size, the probability density of log-return $Y_t = \ln(\frac{S_t}{S_0})$ can be acquired as following converging form:

$$P(Y_t; \lambda, \alpha, \sigma, \mu, \delta) = \sum_{i=0}^{\infty} \frac{e^{-\lambda t}(\lambda t)^i}{i!} N(Y_t; (\alpha - \frac{\sigma^2}{2} - \lambda k)t + i\mu, \sigma^2 t + i\delta^2). \tag{4.26}$$

where α, σ are the instantaneous expected return and diffusion volatility terms. The λ, μ and δ represent jump intensity, mean and standard deviation of jump size distribution respectively. As it is seen in the convolution form, it can be said that the log-return density of Merton Jump Diffusion model is the weighted average of the Black-Scholes normal density.

At this point, we give the following theorem which is about log-normal jump diffusion transition density and its proof to gain better understanding.

Theorem 2. *[14] The log-normal jump-diffusion log-return differential d[ln(X(t))] has transition density as:*

$$\phi_{dln(X(t))}(z) = \sum_{k=0}^{\infty} p_k(\lambda dt)\phi_n(z; \mu_d dt + \mu_j k, \sigma_d^2 dt + \sigma_j^2 k), \tag{4.27}$$

where p_k and ϕ_n represent $Poisson$ and $Normal$ distribution respectively. μ_d and μ_j are diffusion coefficient and jump mean respectively where volatility is symbolized by σ_d for diffusion and σ_j for jump.

Proof:[11] The logic behind this derivation is about finding the density for summation of two independent variables. Let $X = \mu_d dt + \sigma_d dZ(t)$ be diffusion plus log drift term. This term is normally distributed with $\phi_n(z; \mu_d dt, \sigma_d^2 dt)$. The other independent variable YZ is compound Poisson process $\sum_{i=1}^{dP(t)} Q_i$ where $dP(t)$ and Q are jump intensity and jump size respectively. It sums the jump size up to jump intensity. This jump size is normally distributed with density $\phi_n(y; \mu_j, \sigma_j^2)$ and $Z = dP(t)$ is differential Poisson process. It is necessary to obtain summation of these two independent random variables namely X and YZ.

The density of a sum of independent random variables is given via convolution of densities as follow:

$$\phi_{X+YZ}(z) = (\phi_X * \phi_{YZ})(z) = \int_{-\infty}^{\infty} \phi_X(z-y)\phi_{YZ}(y)dy \tag{4.28}$$

Firstly we give density of the compound $Poisson - Normal$ process and then evaluate the convolution for density of compound random variable $\phi_{YZ}(x)$.

The compound Poisson process YZ represents the sum of k independent variables which are distributed normally. The number of jumps k in the process is determined by Poisson distribution. By the law of total variability, we have

$$\begin{aligned} \phi_{YZ}(x) = P\left[\sum_{i=1}^{dP(t)} Q_i \le x\right] &= \sum_{k=0}^{\infty} p_k(\lambda dt) P\left[\sum_{i=1}^{k} Q_i \le x\right] \\ &= \sum_{k=0}^{\infty} p_k(\lambda dt)\phi_{(\sum_i^k Q_i)}(x). \end{aligned} \tag{4.29}$$

The k^{th} jump sum have distribution which is set of nested convolutions of i.i.d. random variables Q_i. Suppose that this part and the diffusion density are merged, then the following form will be acquired:

$$\begin{aligned} \phi_{X+YZ}(z) &= \sum_{k=1}^{\infty} p_k(\lambda dt)\left(\left(\prod_{i=1}^{k} \phi_{Q_i} *\right)\phi_X\right)(z) \\ &= \sum_{k=1}^{\infty} p_k(\lambda dt)((\phi_Q *)^k \phi_X)(z). \end{aligned} \tag{4.30}$$

In the equation 4.30, the last step is written according to properties of identically independent distribution. As a result of the convolution of two normal densities we obtain a normal density which is also a normal density. Its mean and variance are sum of the means and variances leading to a normal density of each k jump counts upon recursion.This last normal density has mean which is sum of the means and variance which is the sum of variances. This situation comes from the identity for two normal distribution multiplication. It is derived from using the completing square technique merging a product of two normal densities.

$$\begin{aligned}\phi_n(X;\mu_1,\sigma_1^2).\phi_n(X;\mu_2,\sigma_2^2) =&\phi_n\left(X;\frac{\mu_1\sigma_2^2+\mu_2\sigma_1^2}{\sigma_1^2+\sigma_2^2},\frac{\sigma_1^2\sigma_2^2}{\sigma_1^2+\sigma_2^2}\right)\\ &\frac{1}{\sqrt{2\pi(\sigma_1^2+\sigma_2^2)}}\text{exp}\left(-\frac{(\mu_1-\mu_2)^2}{2(\sigma_1^2+\sigma_2^2)}\right).\end{aligned} \tag{4.31}$$

Note that X_i's are independent normal random variables which have density $\phi_{Xi}(x)$, mean μ_i, and variance σ_i^2. For $i = 1, ..., K$, let us apply the equation 4.31:

$$\begin{aligned}&I_2(x) = (\phi_{X_1} * \phi_{X_2})(x) = \int_{-\infty}^{\infty}\phi_{X_1}(x-y)\phi_{X_2}(y)dy\\ &= \frac{1}{\sqrt{2\pi(\sigma_1^2+\sigma_2^2)}}\text{exp}\left(-\frac{(x-\mu_1-\mu_2)^2}{2(\sigma_1^2+\sigma_2^2)}\right)\int_{-\infty}^{\infty}\phi_n\left(y;\frac{(x-\mu_1)\sigma_2^2+\mu_2\sigma_1^2}{\sigma_1^2+\sigma_2^2}\frac{\sigma_1^2\sigma_1^2}{\sigma_1^2+\sigma_2^2}\right).\end{aligned} \tag{4.32}$$

In above, the value of K is taken as 2. In general:

$$I_K(x) \equiv \left(\left(\prod_{i=1}^{K-1}\phi_{X_i} *\right)\phi_{X_K}\right)(x) = \phi_n\left(x;\sum_{i=1}^{K}\mu_i,\sum_{i=1}^{K}\sigma_i^2\right) \tag{4.33}$$

As seen, convolution for two normal distribution can take the form of equation 4.27 in Theorem 2.

4.4 Characteristic Function For Merton Jump Diffusion Model

The characteristic function of Merton-Jump Diffusion model is evaluated by using Fourier transforming the Merton log return density function with FT parameters (a,b)=(1,1):

$$\begin{aligned}\phi(\omega) &= \int_{-\infty}^{\infty}\text{exp}(i\omega x_t)\mathbb{P}(x_t)dx_t\\ &= \text{exp}\left[\lambda t\text{exp}\left\{\frac{1}{2}\omega(2i\mu-\delta^2\omega)\right\}-\lambda t(1+i\omega k)-\frac{1}{2}t\omega\{-2i\alpha+\sigma^2(i+\omega)\}\right].\end{aligned} \tag{4.34}$$

After doing some algebraic manipulation, the characteristic function is obtained as follow:

$$\phi(\omega) = \text{exp}[t\psi(\omega)], \tag{4.35}$$

where the characteristic exponent or known as cumulant generating function:

$$\psi(\omega) = \lambda\left\{\exp(i\omega\mu - \frac{\delta^2\omega^2}{2}) - 1\right\} + i\omega(\alpha - \frac{\sigma^2}{2} - \lambda k) - \frac{\sigma^2\omega^2}{2}. \tag{4.36}$$

In the equation 4.36, the symbol k is equal to $e^{\mu+\frac{1}{2}\delta^2} - 1$. This characteristic exponent can be written in differently by changing Levy measure in the Merton Jump Diffusion model in the equation 4.6 with the Levy-Khinchin representation of the finite variation type.

Levy measure of the Merton Jump Diffusion model:

$$l(dx) = \frac{\lambda}{\sqrt{(2\pi\delta^2)}}\exp\left\{-\frac{(dx-\mu)^2}{2\delta^2}\right\} = \lambda f(dx) \tag{4.37}$$

where $f(dx) \sim N(\mu, \delta^2)$. Applying this measure into Levy-Khinchin representation:

$$\begin{aligned}
\psi(\omega) =& ib\omega - \frac{\sigma^2\omega^2}{2} + \int_{-\infty}^{\infty}\{\exp(i\omega x) - 1\}l(dx) \\
=& ib\omega - \frac{\sigma^2\omega^2}{2} + \int_{-\infty}^{\infty}\{\exp(i\omega x) - 1\}\lambda f(dx) \\
=& ib\omega - \frac{\sigma^2\omega^2}{2} + \lambda\int_{-\infty}^{\infty}\{\exp(i\omega x) - 1\}f(dx) \\
=& ib\omega - \frac{\sigma^2\omega^2}{2} + \lambda\left\{\int_{-\infty}^{\infty}\exp(i\omega x)f(dx) - \int_{-\infty}^{\infty}f(dx)\right\}
\end{aligned} \tag{4.38}$$

It is noticed that $\int_{-\infty}^{\infty}\exp(i\omega x)f(dx)$ is the characteristic function of $f(dx)$:

$$\int_{-\infty}^{\infty}\exp(i\omega x)f(dx) = \exp(i\mu\omega - \frac{\delta^2\omega^2}{2}) \tag{4.39}$$

In conclusion, the characteristic exponent take the following form:

$$\psi(\omega) = ib\omega - \frac{\sigma^2\omega^2}{2} + \lambda\left\{\exp(i\mu\omega - \frac{\delta^2\omega^2}{2}) - 1\right\}. \tag{4.40}$$

where $b = \alpha - \frac{\sigma^2}{2} - \lambda k$.

Characteristic exponent and characteristic function have the following relation:

$$\ln(\phi(\omega)) = \psi(\omega).$$

Then, the n^{th} cumulant is defined as:

$$\text{cumulant}_n = \frac{1}{i^n}\frac{\partial^n\psi(\omega)}{\partial\omega^n}|_{\omega=0}.$$

Characteristic exponent produces cumulants as follows:

$$\begin{aligned}
\text{cumulant}_1 &= \alpha - \frac{\sigma^2}{2} - \lambda k + \lambda\mu \\
\text{cumulant}_2 &= \sigma^2 + \lambda\delta^2 + \lambda\mu^2 \\
\text{cumulant}_3 &= \lambda(3\delta^2\mu + \mu^3) \\
\text{cumulant}_4 &= \lambda(3\delta^4 + 6\mu^2\delta^2 + \mu^4)
\end{aligned} \tag{4.41}$$

These cumulants are used for evaluating mean, variance, skewness, and excess kurtosis of the log return density.

$$\begin{aligned}
&E[X_t] = \text{cumulant}_1 = \alpha - \frac{\sigma^2}{2} - \lambda(e^{\mu+\frac{1}{2}\delta^2} - 1) + \lambda\mu, \\
&\text{Variance}[X_t] = \text{cumulant}_2 = \sigma^2 + \lambda\delta^2 + \lambda\mu^2, \\
&\text{Skewness}[X_t] = \frac{\text{cumulant}_3}{(\text{cumulant}_2)^{\frac{3}{2}}} = \frac{\lambda(3\delta^2\mu + \mu^3)}{(\sigma^2 + \lambda\delta^2 + \lambda\mu^2)^{\frac{3}{2}}}, \\
&\text{Excess Kurtosis}[X_t] = \frac{\text{cumulant}_4}{(\text{cumulant}_2)^{\frac{4}{2}}} = \frac{\lambda(3\delta^4 + 6\mu^2\delta^2 + \mu^4)}{(\sigma^2 + \lambda\delta^2 + \lambda\mu^2)^2}.
\end{aligned} \tag{4.42}$$

It is seen that the log-return density of Merton Jump Diffusion model $\mathbb{P}(X_t)$ has some important features. For example, the sign of expected log-return jump size μ specify the sign of skewness. If μ is less than zero then the log-return density of Merton Jump Diffusion model $\mathbb{P}(X_t)$ is negatively skewed. Another example is about intensity λ. If the intensity is high, the log-return density of Merton Jump Diffusion model $\mathbb{P}(X_t)$ becomes fatter-tailed. The value of excess kurtosis is smaller in high jump intensity compared to low counterpart. Finally, it can be said that Merton log-return density shows higher peak and fatter tails, that is, it is more leptokurtic than Black-Scholes normal counterpart.

CHAPTER 5

SIMULATING MERTON JUMP DIFFISON MODEL AND PARAMETER ESTIMATION

In this chapter, the application of Merton Jump Diffusion model will be presented. This application covers numerical approximation with Euler-Maruyama method, parameter estimation with MLE steps. The analysis is based on simulation data. For simulation process, firstly, five initial parameters which are required for Merton Jump Diffusion model are arbitrarily determined, and the number of observation is chosen as 1000. This simulation data is produced for 100 sample paths with these initial values. After this process, we obtain 100 different paths. In the analysis, firstly, 10 paths out of these 100 paths will be selected randomly and the structure of these paths will be examined. Then, we apply the numerical approximation step. For this procedure, we use the technique Euler-Maruyama, which we mention in Chapter 3, will be used. The obtained numerical approximation results will be checked whether it converges to analytical solution or not. For this purpose, this convergence process is going to apply for selected two paths by graphically. In this graphs, we control how the analytical and numerical solutions are close to each other. Moreover, we check how the convergence changes as the time interval increases. We expect the difference between analytical solution and numerical approximation decreases as the time interval becomes smaller . Then, the process of parameter estimation will be analyzed for this model. The parameters of Merton Jump Diffusion model will be estimated by using MLE method. The initial parameters using for simulated data and estimation values of parameters which are produced by MLE will be compared. Then, we will show how the value of initial parameters in simulated data and MLE results close to each other. Finally, we will focus on jump detection in this chapter after convergence process and jump detection is analyzed. Actually, this jump detection is not used for this analysis due to using different discretization method. The standard discretization scheme is used for this model because the model has deterministic boundless jump intensity. However, many models in real life do not have these attributes. This detection technique can be used when the jump intensity is state dependent but bounded. Thus, it is meaningful explain this discretization method.

5.1 Simulating Data by Using Analytical Solution

Analytical solution of Merton Jump Diffusion model is used to simulate the data. The simulation is created with 1000 observations for 100 paths. As Merton used in his model, it is considered that simulating data is stock price values for this case. Merton model has exponentially Levy model form of $X_t = X_0 e^{L_t}$. In this model, X_t and X_0 are stock price and initial stock price values respectively. L_t is Levy process $\{L_t; 0 \leq t \leq T\}$. Recall that L_t has the following form:

$$L_t = (\alpha - \frac{\sigma^2}{2} - \lambda k)t + \sigma W_t + \sum_{i=1}^{N_t} Q_i$$

where $\{W_t; 0 \leq t \leq T\}$ is a standard Brownian motion process. The part of $(\alpha - \frac{\sigma^2}{2} - \lambda k)t + \sigma W_t$ is called as Brownian motion with drift process and $\sum_{i=1}^{N_t} Q_i$ is known as compound Poisson jump process.

According to Levy structure, Merton has the following analytical solution given by equation 4.22

$$X_t = X_0 \text{exp}[(\alpha - \frac{\sigma^2}{2} - \lambda k)t + \sigma W_t + \sum_{i=1}^{N_t} Q_i].$$

In the analytical solution structure, there are five parameters namely $\alpha, \sigma, \mu_j, \delta, \lambda$. They are referred to instantaneous expected return, volatility of the asset, expected value of jump size, standard deviation of jump size, and jump intensity, respectively. W_t and N_t are standard Brownian motion and Poisson process respectively. The term of k is equal to $e^{\mu_j + \frac{1}{2}\delta^2} - 1$. In the simulation procedure, the five parameters are determined as initial values and the simulation is created by their values. For the simulation, the parameter values are taken as $\alpha = 2$, $\sigma = 1.5$, $\mu_j = 0.1, \delta = 0.3, \lambda = 40$. To obtain analytical solution, the following R script is used:

```
### Generate Analytical Solution ###
## stochastic increment of Brownian process ##
for(i in 1:M)
  dw[,i]=sqrt(dt)*rnorm(N)
## cumulative stochastic increment of Brownian process ##
for(i in 1:M)
  w[,i]=cumsum(dw[,i])

## create jump part ##
Jump=matrix(0,N,M) #create empty matrix for jump size
Cumjump=matrix(0,N,M) #create empty matrix for cumulative jump size
X=matrix(0,N,M)     #create empty matrix for process of X
dN=matrix(0,N,M) #create empty matrix for Poisson increment
# constructing Poisson process #
for(j in 1:M){
for(i in 1:N)
{dN[i,j]=rpois(1,lambda*dt)}}#define N numbers from Poisson(lambda*dt)

# according to existing of jump, describe jump sizes
for(j in 1:M){
for(i in 1:N){
if(dN[i,j]==1){Jump[i,j]=rnorm(1,muj,sigmaj)} #define the jump sizes
else{Jump[i,j]=0}}} #there exist no jump then jump size is 0
for(i in 1:M)
Cumjump[,i]=cumsum(Jump[,i]) #define cumulative values for jump sizes.

## constructing analytical solution ##
for(i in 1:N){
for(j in 1:M){
X[i,j]=Xs*exp((alpha-0.5*sigma^2-lambda*k)*t[i]+sigma*w[i,j]+Cumjump[i,j])
    }}
```

Listing 5.1: Simulation of Data for Merton Model in R

As seen in the R script, three types increment are created namely time increment, Brownian increment, and Poisson increment. Poisson increment does not always occur. When there exists no jump, Poisson increment does not yield. When there exists a jump, Poisson increment also exists and it is normally distributed for Merton Jump Diffusion model.

According to the given information, the following simulation paths are yielded. In Figure 5.1, ten simulated paths are presented. As said before, the simulations are created for 100 paths but ten paths are only given graphically. The reason is to avoid complex graphical presentation. As seen in the figure, the paths are nearly exponentially distributed. This pattern is suitable with Levy family. Some paths can be seen as straight lines. The reason of that these paths take very small values but the others take higher values. All in all, it can be said that this graphical representation shows the simulations are exponentially distributed as expected.

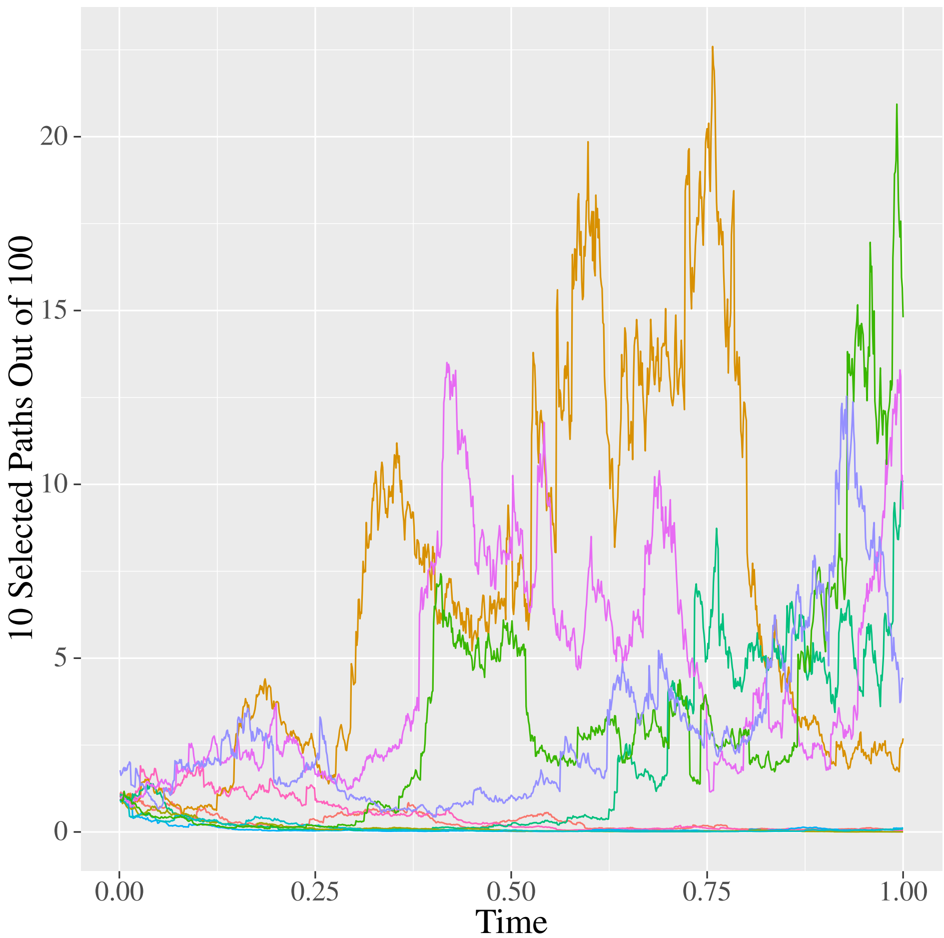

Figure 5.1: Simulated Merton Paths

5.2 Convergence of Numerical Approximation to Analytical Solution

Generally, it is not possible to obtain the analytical solution of SDEs. At this time, some numerical approximation methods are used to obtain the solution. As mentioned in Chapter 2, Milstein method, Runge-Kutta method and Euler-Maruyama method are well-known numerical approximation methods. Euler-Maruyama method is easy to compute and it is more suitable for the structure of models which are investigated in this work. For these reasons, Euler-Maruyama approximation method is used for numerical approximation procedure. Then, the numerical approximation values are checked how they converge to the analytical solution. In the simulation process, 100 paths are yielded. Two of them are chosen randomly and their analytical solutions and numerical solutions are compared in Figure 5.2. In this case, 1000 observations are used. The time interval is $\frac{1}{1000} = 0.001$ because $dt = \frac{T}{N}$ where $N = 1000$ and $T = 1$. For numerical approximation method, the following Euler-Maruyama form is used:

$$X_{temp} = X_s + (\alpha - \lambda k)dtX_s + \sigma dW X_s, \tag{5.1}$$

$$X_{app} = X_{temp} + (X_{temp}(\exp(z) - 1)), \tag{5.2}$$

where X_{app} and X_{temp} are approximate values for jump case and no-jump case respectively. The parameters $\alpha, \sigma, \mu_j, \delta, \lambda$ are instantaneous expected return, volatility of the asset, expected value of jump size, standard deviation of jump size, and jump intensity, respectively. z is distribution of jump size where $z \sim N(\mu_j, \delta^2)$. X_s is previous state values for approximation chains. To illustrate the convergence performance of Euler-Maruyama approximation method, one can check Figure 5.2.

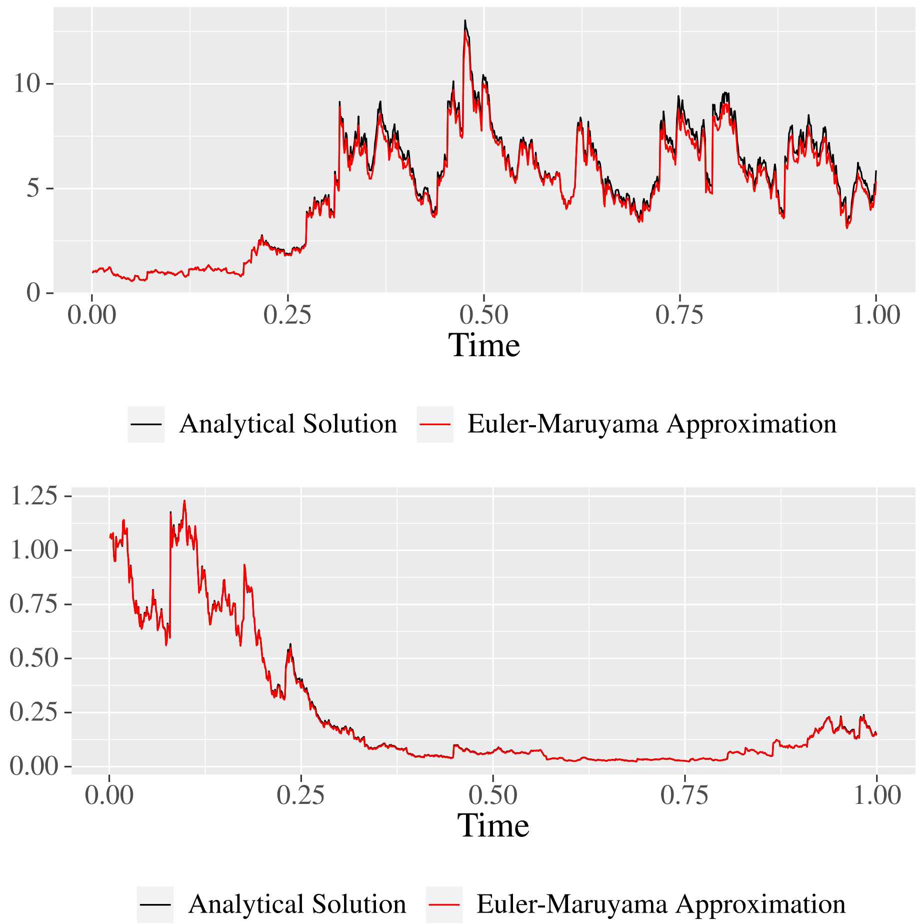

Figure 5.2: Convergence Between Analytical Solution and Euler Approximation

As seen in Figure 5.2, there exist two graphs. Each one represents one path. Two paths are chosen out of 100 paths to illustrate graphically. Both paths numerical approximation perfectly fit to their analytical solutions.

Although the graphical presentation says that the convergence is fine, it is better to confirm this result by checking convergence performance as time interval changes. That is, it should be also checked whether all paths with different step sizes have good convergence or not. For each time interval, average of absolute difference of all paths for analytical values and numerical approximation values are calculated. Then, these results are presented in Table 5.1.

Time Interval:	20dt	10dt	5dt	2dt	dt
Mean of Error Through M Paths:	0.7643	0.4311	0.2698	0.1458	0.0887

Table 5.1: Convergence Rates For Different Time Intervals

As seen in Table 5.1, five time intervals are used for 100 paths. The table shows as the time interval decreases the convergence between analytical solution and numerical approximation becomes better. This is the expected result of the convergence process.

5.3 Parameter Estimation of Merton Jump Diffusion Model

In the beginning of the simulation, the initial values are determined for the parameters $\alpha, \sigma, \mu_J, \delta, \lambda$. According to these parameters, the simulation is executed. After that, we estimate these parameters by using MLE. To do this, number of observations are taken 1000 and the simulation is repeated 100 times. That is, each parameter is estimated 100 times and the average of these estimations are taken for each parameter separately. In Merton model, as a result of having normal distribution of log-return jump size, the probability density of log-return $Y_t = \ln(\frac{X_t}{X_0})$ can be acquired as following converging form:

$$P(Y_t \in A) = \sum_{i=0}^{\infty} \frac{e^{-\lambda t}(\lambda t)^i}{i!} N(Y_t; (\alpha - \frac{\sigma^2}{2} - \lambda k)t + i\mu_j, \sigma^2 t + i\delta^2),$$

where α, σ, μ_j, δ, λ are instantaneous expected return, volatility, mean of jump size, standard deviation of jump size, and jump intensity, respectively. Likelihood function of this density function is

$$L(\theta; Y) = \prod_{t=1}^{T} P(Y_t). \tag{5.3}$$

We minimize negative log-likelihood function to make the calculation easier.

$$-\ln L(\theta; Y) = -\sum_{t=1}^{T} \ln P(Y_t). \tag{5.4}$$

We have likelihood function. Thus, we are able to construct MLE. This function is found by using transition density function in the equation 4.26. The following code is used to obtain parameter estimation and then its output is given in Table 5.2.

```r
### MLE of SDE with Jump ###
dx=matrix(0,N,M) # create empty matrix for difference of X states
for(j in 1:M){
dx[,j]=c(log(X[1,j])-log(1),diff(log(X[,j])))} # find difference of X states
f=matrix(0,N,10) # create empty matrix for transition density
estimate=matrix(0,M,5)
for(v in 1:M){
dif=dx[,v]
likelihood=function(theta,dif,dt) # create a function for likelihood
{
Q1=theta[1] # symbolize alpha parameter
Q2=theta[2] # symbolize sigma parameter
Q3=theta[3] # symbolize muj parameter
Q4=theta[4] # symbolize sigmaj parameter
Q5=theta[5] # symbolize lambda parameter
#assign element of transition density matrix for N elements and k jump possibilities.
for(i in 1:N){
for(j in 1:10){
f[i,j]=(exp(-Q5*dt)*(Q5*dt)^(j-1)/factorial(j-1))*(1/sqrt(2*pi*(Q2^2*dt+(j-1)*Q4^2)))*exp(-(dif[i]-((Q1-Q5*k-Q2^2/2)*dt+(j-1)*Q3))^2/(2*(Q2^2*dt+(j-1)*Q4^2)))}}
R=rowSums(f)
LL=-sum(log(R)) # find -loglikelihood
return(LL)
}
## Minimize -loglikelihood function ##
estimation=optim(c(1,1,1,1,1),likelihood,gr=NULL,dif,dt,method="L-BFGS-B",
lower=c(-Inf,0,-Inf,0,-Inf), upper = c(Inf,Inf,Inf,Inf,Inf),hessian=T)
options(scipen=999)
estimate[v,]=estimation\$par # assign each parameter estimation set to estimate matrix.
}
parameter_estimations=colSums(estimate)/M # give the estimated parameter value for each parameter
```

Listing 5.2: Maximum Likelihood Estimation in R

When this algorithm works, the following output is obtained:

	α	σ	μ_j	δ	λ
Initial Parameter Values:	2	1.5	0.1	0.3	40
Estimated Parameter Values:	1.7157	1.4972	0.1053	0.2887	38.2364

Table 5.2: Comparison of Initial and Estimated Parameter Values

As seen in Table 5.2, the estimation results are close to initial parameter values. Thus, this confirms that the parameter estimation process works well. The estimation process should be made many times and the averages of these estimations must be taken to reach the true values because there exists random number generation in the parameter estimation algorithm. This random situation causes different estimation results after the process works. Thus, the parameter estimation gives better and stable results if the process works many times.

In addition to parameter estimation process, now, we will mention the jump detection method by using alternative discretization approach which is different from the method used in this thesis. In the analysis of this thesis, we do not use this method but we want to give this additional information. This can be helpful for other studies including scenarios given below.

5.4 Jump Detection in Merton Jump Diffusion Model

Although discretization scheme which is used in this thesis is classical approach, alternative discretization scheme is preferred in some situations. When the jump intensity has bounded and state dependent, this alternative method is used. This method is based on constructing the jump times by a stochastic time-change of a standard Poisson process. To illustrate this case, an example is given below. In this example, a data is simulated and their jump times are determined based on the algorithm given in Chapter 3. The simulation is created by the following information in Table 5.3.

Parameter:	α	σ	μ_j	δ	λ	dt	N	T
Value:	3	2	0	1	5	0.01	100	1

Table 5.3: Simulation Initial Values

where α, σ, μ_j, δ, λ are instantaneous expected return, volatility, mean of jump size, standard deviation of jump size, and jump intensity, respectively. The number of observation $N = 100$ and time is $(0 \leq t \leq 1)$ and so $T = 1$. The time interval dt is equal to $\frac{T}{N} = \frac{1}{100} = 0.01$. Therefore we obtain the Figure 5.3.

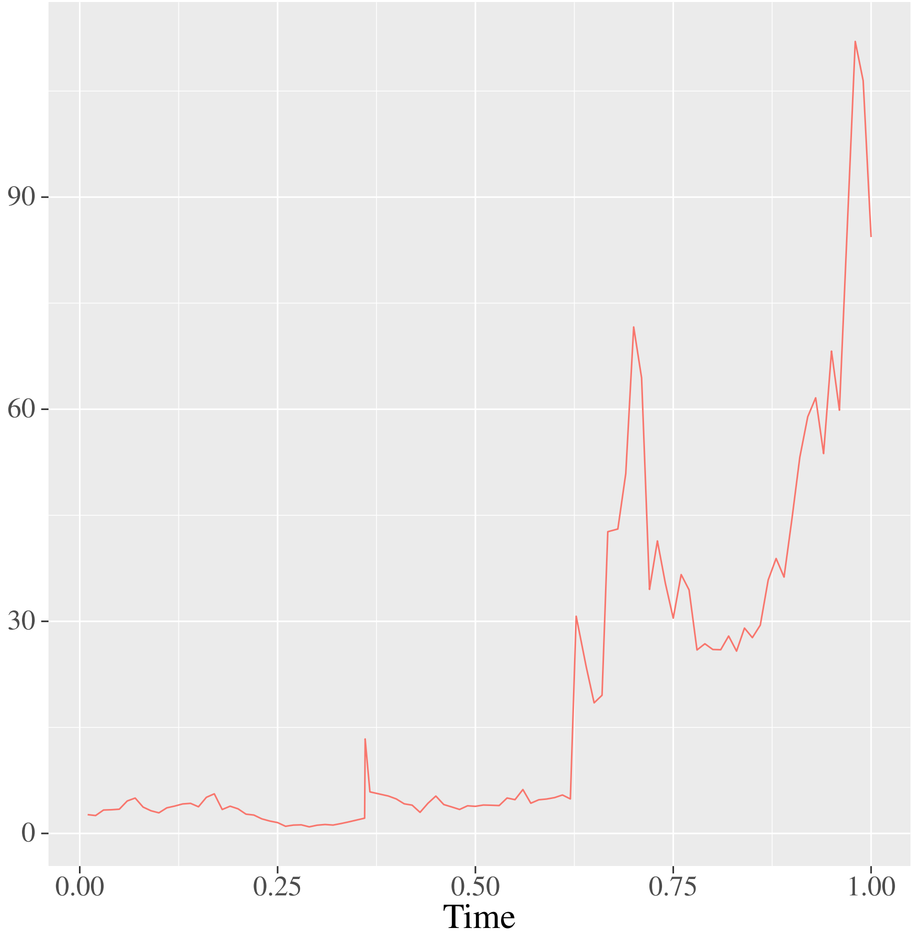

Figure 5.3: Simulated Merton Data by Using Discretization Method

In the Figure 5.3, there seems to become jumps at six time points, in (0.25, 0.50) one jump, in (0.50, 0.75) three jumps and in (0.75, 1.00) two jumps. However, this does not show that they are jump points for this simulation. To confirm that, jump times are determined by using algorithm in Chapter 3. The result of this algorithm is represented in Table 5.4.

Jump Times:	0.361	0.367	0.627	0.667
Jump Sizes:	11.207	-7.486	25.841	23.133
Corresponding Values at Jump Points:	13.363	5.877	30.719	42.677

Table 5.4: Jump Sizes, Jump Times and Corresponding Values at Jump Points

As seen in Table 5.4, the jumps become at 0.361, 0.367, 0.627, 0.667 time points. The corresponding jump values are 13.363, 5.877, 30.719, 42.677. The highest jump size is seen at time point 0.627. All jumps are upward except at time point 0.367. It can be said that, there are four jump points in the simulation, although there seems to be six jump points in the graphic. As a result of being member of Levy family, this figure shows exponential increasing. Although, the second jump point value is lower than the other jump values, there exist a increasing as a whole.

CHAPTER 6

APPLICATON OF MERTON JUMP DIFFISION MODEL ON DOLLAR/TL EXCHANGE RATE

As mentioned before, the unexpected situations are often seen in many areas such as economics, biology, chemistry, physics, and social sciences. These situations cause dramatic changes in value of response. Wars, economic crisis, natural disasters or similar cases create unexpected effects on models. In financial sector, some sharp increases and decreases has been seen in market data. The recently seen movements in exchanges rates of Turkish Lira to foreign currencies are one of the examples for this case. We worked on the data of daily United States (US) Dollar to Turkish Lira currency exchange rate. We considered this data has sudden changes and it is suitable for Merton Jump Diffusion model. We aim to model this data and to forecast. In this chapter, Merton Jump Diffusion model and Black-Scholes framework will be used for modeling the data. For both models , some assumptions required for doing analysis will be controlled and then the analysis will be made. Firstly, the parameter estimations will be applied. Then, AIC will be used to determine the better fit to empirical data. Moreover, MAPE for both two models are calculated to see the forecasting accuracy performance of them.

6.1 Dollar/TL Exchange Rate Data and Jump Detection

In the analysis, the used data, which is daily US Dollar to Turkish Lira exchange rate, is taken between the date of $01.02.2019$ and $21.06.2019$. In this interval, there are 121 data points. The data is given in the Figure 6.1. As seen in the figure, the data includes some sudden increasing/decreasing. It has an overall increasing trend.

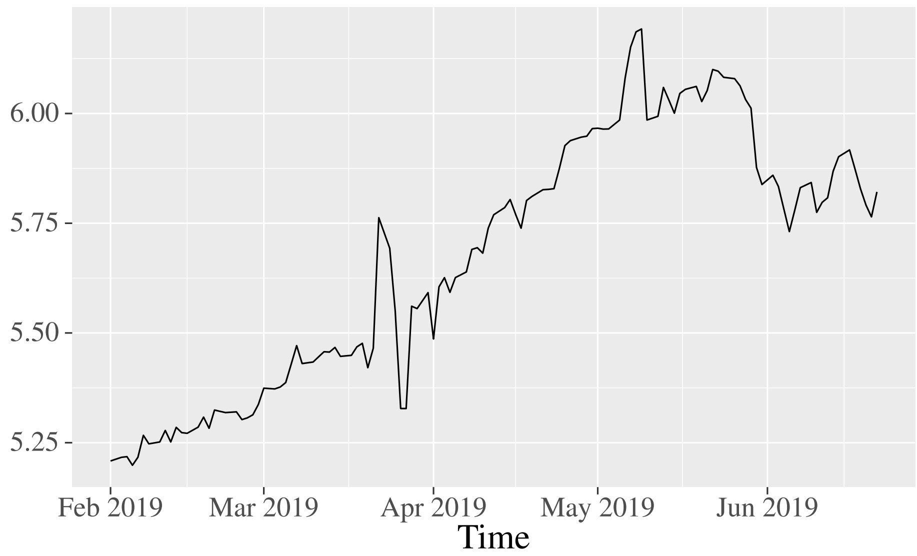

Figure 6.1: Dollar to Turkish Lira Exchange Rate

The log-return of data is taken to determine jump points and the Figure 6.2 is created . We define the $(-0.0125, 0.0125)$ as cut-off points for log-return exchange rate data. As seen in this figure, these points are determined by red line. The points which are out of these lines are accepted jump points. According to the Figure 6.2 and the calculations based on given threshold, there exist 9 jump points.

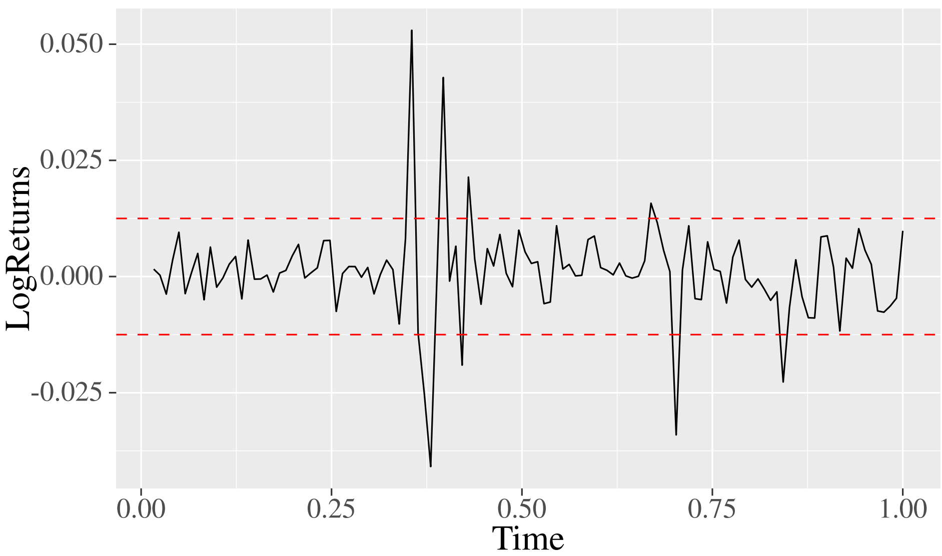

Figure 6.2: Log-Return Exchange Rate For Detecting Jump Points

6.2 Assessing The Model Assumptions

Model assumptions are requirement for testing the data. Before applying parameter estimation, we should mention model assumptions and their checking. As said before, Merton Jump Diffusion model includes Brownian motion and Poisson process, Black-Scholes model includes Brownian motion. These processes must satisfy three conditions namely stationarity, normality and independence. Thus, we check three assumptions according to the following log-return model:

$$\ln(\frac{X_t}{X_{t-1}}) = (\alpha - \frac{\sigma^2}{2} - \lambda k)dt + \sigma W_t + \sum_{i=1}^{N_t} Q_i \qquad (6.1)$$

6.2.1 Stationarity

Stationary time series causes constant sample statistics such as mean, variance and autocorrelation in time. Many forecasting methods need the stationarity assumption to obtain accurate prediction. Otherwise, unexpected acts in the analysis can be seen. For example, a test statistics, t, is not following t-distribution. The mean or variance can be increased with sample size if the time series persistently increases in time. This causes to undervalue the mean and variance in coming periods. In this application, stationarity is checked for Merton and Black-Scholes model according to log-return $\ln(\frac{X_t}{X_{t-1}})$. In the analysis, we use two tests namely KPSS test and ADF test to check stationarity.

- Hypothesis Test for KPSS Test:
  H_0: Distribution is stationary.
  H_1: Distribution is not stationary.

- Hypothesis Test for ADF Test:
  H_0: Distribution is not stationary.
  H_1: Distribution is stationary.

If the null hypothesis is rejected in KPSS test, then time series does not have unit root or it is not stationary. If the null hypothesis is rejected in ADF test, then time series is stationary.

6.2.2 Normality

Normality are checked for Merton model according to the following form which is distributed standard normal N(0,1).

$$\frac{\ln(\frac{X_t}{X_{t-1}}) - \sum_{i=1}^{N_t} Q_i - \mu t}{\sigma\sqrt{t}} \tag{6.2}$$

where μ and σ are mean and standard deviation of jump size distribution. Black-Scholes model normality assumpiton is controlled for $\ln(\frac{X_t}{X_{t-1}})$. Although the easiest way of detecting normality is graphical presentation. This is not accurate so we prefer to use statistical test Shapiro-Wilk and Jarque-Bera test. The hypothesis for normality is:

- Hypothesis Test for Shapiro-Wilk or Jarque-Bera Test:
  H_0: Distribution is normally distributed.
  H_1: Distribution is not normally distributed.

If the null hypothesis is rejected, then the normality assumption is satisfied.

6.2.3 Independence

Box-Ljung or Box-Pierce test can be used as statistical test to obtain exact result of independence checking. With this test, we can see whether the increments are independent or not. Independence are checked for both model according to the log-return $\ln(\frac{X_t}{X_{t-1}})$.

- Hypothesis Test for Box-Pierce Test:
  H_0: Distribution is independently distributed.
  H_1: Distribution is not independently distributed.

If the null hypothesis does not rejected, then the independent increment assumption is satisfied.
Now, let us check the these three assumption whether they are satisfied or not. To do this, we will construct some test statistics mentioned above. The results are in Table 6.1:

	Merton Jump Diffusion Model			Black-Scholes Model		
	Stationarity	Normality	Independence	Stationarity	Normality	Independence
KPSS	0.10	-	-	0.10	-	-
ADF	0.01	-	-	0.01	-	-
Shapiro-Wilk	-	0.06	-	-	≈0	-
Jarque-Bera	-	0.75	-	-	≈0	-
Box-Pierce	-	-	0.35	-	-	0.35

Table 6.1: P-Values of Some Test Statistics for Model Assumption Checkings

As seen in Table 6.1, stationarity assumption is satisfied for both Merton Jump Diffusion model and Black-Scholes model because p-value for KPSS test is bigger than critical value 0.05 and p-value for ADF test is less than 0.05. This indicates there is no enough evidence to reject the increments are stationary. In the normality assumption for Merton Jump Diffusion model, Shapiro-Wilk and Jarque-Bera tests say that p values are bigger than critical values. That is, the null hypothesis which is increments are normally distributed is not rejected. Thus, normality assumption is satisfied. However, this assumption is not satisfied for Black-Scholes model due to the very small p-values. The last assumption which is independence for both models is satisfied. Box-Pierce test confirms this result with the p-values which are bigger than critical value 0.05. Moreover, Merton Jump Diffusion model has two additional assumptions. The distribution of logarithm of jump size is normal and interarrival times of jump point is exponentially distributed. The normality is checked for the following form:

$$\frac{\ln(\Delta_j) - \mu}{\sigma}$$

Here, Δ_j is j^{th} jump size where j is the number of jump. μ, σ are mean and standard deviation of jump sizes respectively. We check the standard normality based on this form. According to the Shapiro-Wilk test p-value for normality is 0.22. This indicates the logarithm of jump size distribution is standard normal. Since, the p-value is less the critical value 0.05. The other assumption which is exponential distributed jump arrival times is controlled by using Kolmogorov-Smirnov test. In this test, the null hypothesis is the interarrival times of jumps is distributed exponential. The p-value obtained from this test is 0.18, that is, the null hypothesis is not rejected. Since, p-value is less than the critical value 0.05. Therefore, we can say the jump arrival times is exponentially distributed.

6.3 Parameter Estimation of Merton Jump Diffusion Model For USD/TL Exchange Rate

The following Merton Jump Diffusion model is given by the equation 4.19.

$$dX_t = (\alpha - \lambda k)X_t dt + \sigma X_t dW_t + (y_t - 1)X_t dN_t$$

Here W_t and N_t are Brownian motion and Poisson process, respectively. The symbol of λ is intensity of Poisson process, α is instantaneous expected return, σ is diffusion term, $y_t - 1$ is relative jump size, and its mean is k. In Merton model, as a result of having normal distribution of log-return jump size, the probability density of log-return $Y_t = \ln(\frac{X_t}{X_0})$ can be acquired as the following converging form:

$$P(Y_t \in A) = \sum_{i=0}^{\infty} \frac{e^{-\lambda t}(\lambda t)^i}{i!} N(Y_t; (\alpha - \frac{\sigma^2}{2} - \lambda k)t + i\mu_j, \sigma^2 t + i\delta^2),$$

where α, σ, μ_j, δ, λ are instantaneous expected return, volatility, mean of jump size, standard deviation of jump size, and jump intensity, respectively. Likelihood function of this transition density function is given by the equation 5.3.

$$L(\theta; Y) = \prod_{t=1}^{T} P(Y_t).$$

After the process of MLE estimation is completed, the following results come out:

	α	σ	μ_j	δ	λ
Estimated Parameter Values:	0.238	0.063	-0.015	0.058	4.440

Table 6.2: Merton Jump Diffusion Model Parameter Estimations For Exchange Rate

As seen in Table 6.2, we find the five parameters values for Merton Diffusion model after the process of parameter estimation. According to this, the expected return and volatility are 0.238 and 0.063 respectively. The jump size distribution's expected value and standard deviation are -0.015 and 0.058, respectively. The jump intensity is 4.44. In other words , we expect 4 jumps in the US Dollar to Turkish Lira currency exchange rate data between the date of $01.02.2019$ and $23.06.2019$ averagely.

6.4 Parameter Estimation of Black Scholes Model For USD/TL Exchange Rate

The parameters of daily US Dollar to Turkish Lira currency exchange rate data will also be estimated by using Black-Scholes framework. To do this, the following Black-Scholes model is used:

$$d_t = \mu X_t dt + \sigma X_t dW_t, \tag{6.3}$$

where μ is drift parameter and σ is diffusion parameter. The transition density function is

$$P(Y_t) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(dX_t - \mu X_t dt)^2}{2\sigma^2 X_t dt} \right), \tag{6.4}$$

where $dX_t = X_t - X_{t-1}$ and $dt = t_n - t_{n-1}$

After the process of MLE estimation is completed, the following results come out:

	μ	σ
Estimated Parameter Values:	0.112	0.114

Table 6.3: Black-Scholes Model Parameter Estimations For Exchange Rate

As seen in Table 6.3, the drift parameter μ takes the value of 0.112 and diffusion parameter σ takes the value of 0.114.

6.5 Comparison of Merton Jump Diffusion Model and Black Scholes Model For USD/TL Exchange Rate

The daily US Dollar to Turkish Lira exchange rate data was modelled considering both two structure of Black-Scholes and Merton Jump Diffusion. To determine which model has better fit to this data, AIC is used for both models. AIC is calculated as follows:

$$AIC = -2\ln L + 2k \tag{6.5}$$

where L is the value of likelihood function, N is the number of observations, and k is the number of parameters. The best model is the one that has minimum AIC among the other models. The following results are found when the AIC calculated for two models:

	Merton Model	Black-Scholes Model
AIC:	352.04	356.61

Table 6.4: AIC For Merton and Black-Scholes Models

The better model in the comparison has the smallest AIC value. Thus, Merton Jump Diffusion model is better model than Black-Scholes model for daily US Dollar to Turkish Lira exchange rate data according to Table 6.4.

6.6 Prediction on US Dollar to Turkish Lira Exchange Rate

We made parameter estimation for Black-Scholes and Merton Jump Diffusion model by using MLE for USD/TL exchange rate data between date of $01.02.2019$ and $21.06.2019$. In this interval, there are 121 data points. Now, we make prediction for the exchange rate between date of $23.06.2019$ and $02.07.2019$. To do this, we used these two models with estimated parameter values. Moreover, MAPE is used to measure how accurate the forecasting results. It is calculated for two models according to the following structure:

$$MAPE = \Big(\frac{1}{n}\sum\frac{|Actual - Forecasting|}{|Actual|}\Big)100, \tag{6.6}$$

where n is number of observations. If the results are less than 10 we can consider the model forecasting accuracy performance is good.

In the following table, the prediction values which are found for between the date of 23.06.2019 and 02.07.2019 are presented. As seen in the table, the prediction values of Merton model is more close to actual values in all days. However, Black-Scholes predictions have more deviation from the actual data set. After the fifth day, Black-Scholes model very largely deviate from the actual values. Also, the prediction MAPE values are calculated for both model to determine which model has better forecasting accuracy. According to the results in Table 6.5, Merton Jump Diffusion model prediction values are closer to the actual values. Merton jump Diffusion model has smaller MAPE value which is 2.91% and Black-Scholes model has 4.71% MAPE value. It can be said that Merton Jump Diffusion model has better forecasting accuracy performance than Black-Scholes model for this data set.

Date	True Value	Merton Prediction	Black-Scholes Prediction
23.06.2019	5.76	5.84	5.92
24.06.2019	5.81	5.87	5.91
25.06.2019	5.80	5.86	5.91
26.06.2019	5.78	5.89	5.97
27.06.2019	5.77	5.86	5.99
28.06.2019	5.79	5.92	6.17
30.06.2019	5.74	5.98	6.14
01.07.2019	5.65	5.99	6.08
02.07.2019	5.65	6.02	6.08
MAPE:		2.91	4.71

Table 6.5: Merton and Black-Scholes Model Forecasting Performance

From the Figure 6.2, we can see that Merton model has almost same path with actual values up to fifth day. After this point, both model prediction values deviate from actual values but this deviation is higher for Black-Scholes model. Also, we can see from the figure Merton model has closer values at all points to actual values than Black-Scholes model.

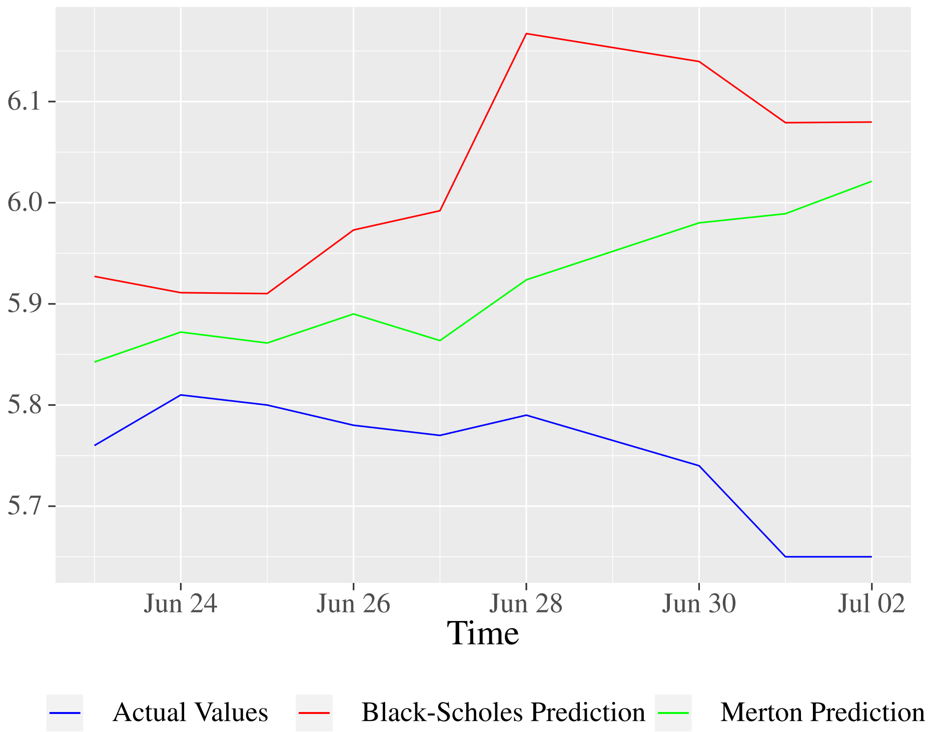

Figure 6.3: Comparison of Actual Values, Merton Jump Diffusion Model and Black-Scholes Model Predictions

CHAPTER 7

CONCLUSION

In many branches, measuring the rate of change has become more and more important over the years. This situation has made the differential equation more widely used. Moreover, stochastic movement in differential equations and so SDE are frequently used for many purposes. By using SDE, it is possible to have information about randomness or uncertainty in the models. In the thesis, we mainly focus on SDE with jump models. These models have ability to measure the radical changes in quantities due to some reasons such as wars, natural disasters, market crashes or some dramatic news as well as moderate changes.

In this work, we have two purposes. First one is to estimate parameter values in Merton Jump Diffusion models. In Chapter 5, the data was simulated according to Merton Jump Diffusion model. We found that the initial parameter values used for the simulating data and parameter estimation values by using MLE are very close to each other as seen in the Table 5.3. To obtain parameter values truly, the estimation process was made many times due to including random number generation from Normal and Poisson distribution. After that, the average of these estimation process was taken. As iteration number of estimation process increases, the estimation results are closer to initial parameter values. The reason of this situation is eliminating the potential bias yielded by random number generator [9]. In Chapter 5, we also checked whether numerical approximation and analytical solution values are close to each other or not. As seen in Figure 5.2, the convergence seems to be perfect. Euler-Maruyama method was used for numerical approximation. As mentioned earlier, most of the times analytical solutions are difficult to obtain or sometimes not even obtained. For this kind of situations, these numerical solution values can be used instead of analytical solution. Therefore, it is important that the algorithm, which we adapted to obtain numerical approximation, works well. Moreover, we adapted jump detection algorithm in Chapter 3 to determine jump times and jump sizes. However, this algorithm was not used in our analysis because this jump detection algorithm is used for unbounded jump intensity. In Merton model, jump intensity is constant and so the classical discretization scheme was used. We mentioned this discretization algorithm to become reference for other studies including Jump Diffusion models with unbounded jump intensity.

The second purpose of this study is to show Merton model has better fit to US Dollar to Turkish Lira exchange rate data than Black-Scholes counterpart. The US Dollar to Turkish Lira exchange rate data has showed sudden increase/decrease in recent years. Therefore, it includes many jumps especially upward jumps. We considered jump diffusion models can be compatible with the trend of this data and it has better fit to this data than the other diffusion models. Among various jump diffusion models we have preferred trying the Merton Jump Diffusion model since the data set satisfies its assumptions. Once, the Merton model was fitted for the data the parameter estimations were conducted both for the Merton Jump Diffusion model and the Black-Scholes model. Then, these two models were compared in terms of forecast performance and model fit. Firstly, we checked the AIC values for both models in Chapter 6 to control model fit performances. Merton Jump Diffusion model has better fit to the data because AIC values for Merton model is lower than AIC for Black-Scholes model as seen in Table 6.4. This result was found as expected because the exchange rate data includes many jumps and we know that Black-Scholes model is insensitive to this jumps. Secondly, we made 9 days prediction by using the exchange rate data with Merton and Black-Scholes model. The predictions were controlled with MAPE values for both models. As expected, Merton Jump Diffusion model has smaller MAPE value. Thus, the predictions of Merton Jump Diffusion model is more close to original values. Also, we present the forecasting trends for these two models with actual data set graphically. This also confirms that Merton prediction is more close to actual data for all days. Therefore, Merton Jump Diffusion model has better forecasting accuracy performance than Black-Scholes model for US Dollar to Turkish Lira exchange rate data.

Both Merton model and Black-Scholes model have some assumptions to study on them. Before doing analysis, we must check the all assumptions. In Chapter 6, we investigated three assumptions for both models namely stationarity, normality and independence of increments. Also, we checked two additional assumptions for Merton Jump Diffusion model. First one is that the logarithm of jump sizes is distributed normal. Second one is that interarrival times of jumps is distributed exponential. The stationarity assumption is satisfied for both models. That is, the probability of any increments does not depend location of time interval. They only depend on length of time intervals. Stationarity prevents the unreliable and spurious outputs obtained in the analysis. These outputs can cause weak understanding and forecasting. However, the normality assumption is only satisfied for Merton model. Non-normality obstructs exact inference on estimations of coefficients. This assumption is not satisfied for Black-Scholes model and this result is expected. Since, the data includes some jump points and this made the distribution of $\ln(\frac{X_t}{X_{t-1}})$become non-normal. This is also answer the why this data is more suitable with Merton Jump Diffusion model. The last assumption independent increments is satisfied for both models. The Merton additional assumptions which are normally distributed logarithm of jump size and exponentially distributed interarrival times of jumps are also satisfied.

In conclusion, Merton Jump Diffusion model is a better fit than the Black-Schooles model for US Dollar to Turkish Lira exchange rate data set since it captures the jump points as well as the continuous parts. The Black-Scholes framework is insensitive to jumps in the data set. The US Dollar to Turkish Lira exchange rate data has both discrete and continuous parts. Thus, Merton Jump Diffusion model explains the data better.

REFERENCES

[1] ALLEN, E. *Modeling with Itô Stochastic Differential Equations*, vol. 22. Springer Netherlands, 2007.

[2] BASS, R. F. Stochastic differential equations with jumps. *Probab. Surveys 1* (2004), 1–19.

[3] BHATTACHARYA, R., AND WAYMIRE, E. *Stochastic Processes with Applications*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, 2009.

[4] BOYCE, W., AND DIPRIMA, R. *Elementary Differential Equations and Boundary Value Problems, 10th Edition*. Wiley, 2012.

[5] CARLSSON, J., MOON, K.-S., SZEPESSY, A., TEMPONE, R., AND ZOURARIS, G. *Stochastic Differential Equations: Models and Numerics 1*. KTH, 2019.

[6] DITLEVSEN, S., AND SAMSON, A. Introduction to stochastic models in biology. *Lecture Notes in Mathematics 2058* (01 2010).

[7] GEISS, S. Lecture notes: Stochastic differential equations. = http://users.jyu.fi/ geiss/lectures/stochasticsde.pdf, March 2018.

[8] GIESECKE, K., AND TENG, G. Numerical solution of jump-diffusion sdes. Tech. rep., Stanford University, 2013.

[9] GONZÁLEZ-URTEAGA, A., NOVALES, A., AND RUBIO, G. Estimation of stochastic volatility models with jumps in returns for stock market indices. Tech. rep., Universidad del País Vasco, 2019.

[10] HANSON, F. B. *Applied Stochastic Processes and Control for Jump-Diffusions: Modeling, Analysis, and Computation (Advances in Design and Control)*. SIAM.

[11] HANSON, F. B., AND WESTMAN, J. J. Stochastic analysis of jump-diffusions for financial log-return processes. In *Stochastic Theory and Control* (Berlin, Heidelberg, 2002), B. Pasik-Duncan, Ed., Springer Berlin Heidelberg, pp. 169–183.

[12] HAUGH, M. A brief introduction to stochastic calculus. Tech. rep., Columbia University, 2016.

[13] IBE, O. *Markov Processes for Stochastic Modeling*. Stochastic modeling. Elsevier Science, 2008.

[14] MATSUDA, K. Introduction to merton jump diffusion model. Tech. rep., The City University of New York, 2004.

[15] MERTON, R. C. Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics 3*, 1-2 (1976), 125–144.

[16] MIKOSCH, T. *Elementary Stochastic Calculus with Finance in View*. Advanced series on statistical science & applied probability. World Scientific, 1998.

[17] PRIVAULT, N. Lecture notes: Stochastic calculus for jump processes. = http://www.ntu.edu.sg/home/nprivault/MA5182/stochastic-calculus-jump-processes.pdf.

[18] REDUCHA, W. Parameter estimation of the pareto-beta jump-diffusion model in times of catastrophe crisis. Tech. rep., Halmstad University, Applied Mathematics and Physics (CAMP), 2011.

[19] ROSS, S. *Differential Equations*. Wiley, 1984.

[20] SAUER, T. Numerical solution of stochastic differential equations in finance. In *Handbook of Computational Finance*, J.-C. Duan, W. K. Hardle, and J. E. Gentle, Eds. Springer, Berlin-Heidelberg, 2012, ch. 19, pp. 529–550.

[21] SERFOZO, R. *Basics of Applied Stochastic Processes*. Probability and Its Applications. Springer Berlin Heidelberg, 2009.

[22] SIGMAN, K. Lecture notes: Geometric brownian motion, modeling stock prices in continuous time, black-scholes option pricing formula. `http://www.`

columbia.edu/~ks20/FE-Notes/4700-07-Notes-GBM.pdf, 2006.

[23] TANKOV, P. *Financial Modelling with Jump Processes*. Chapman and Hall/CRC Financial Mathematics Series. CRC Press, 2003.

[24] WHITT, W. A quick introduction to stochastic calculus. = http://www.columbia.edu/ ww2040/4701Sum07/lec0813.pdf.

[25] WIKIPEDIA. Markov property, 2019. [Online; last edited 3-July-2019].

[26] WINKEL, B. Shen, wen. 2015. lecture notes for math 251: Introduction to ordinary and partial differential equations. = https://www.simiode.org/resources/2947, Oct 2016.

APPENDIX A

ALL R SCRIPTS USED IN THE THESIS

```
library(ggplot2)
library(tikzDevice)
n=1000;  T=1
delta=T/n
set.seed(123)
W=cumsum(c(0,rnorm(n,0,1)*sqrt(delta)))
t=seq(0,T,delta)
data=data.frame(t,W)
colnames(data)=c("Time", "W")
tikz("BM.tex",width=5,height=5)
ggplot(data,aes(x=Time,y=W))+geom_line()
dev.off()
```

Listing A.1: Simulated Brownian Path

```
library(ggplot2)
library(tikzDevice)
T=4
n=100
lambda=3
dt=T/n
t=seq(dt,T,dt)
set.seed(123)
N=cumsum(rpois(length(t),lambda*dt))
data=data.frame(c=t,y=N)
colnames(data)=c("Time", "Nt")
tikz("pois.tex",width=5,height=5)
ggplot()+geom_step(data, mapping=aes(x=Time, y=Nt))
dev.off()
```

Listing A.2: Simulated Poisson Path

```
### Generate Analytical Solution ###
set.seed(127)# create reproducible results
## define the initial parameter values for process ##
alpha=2 #instantaneous expected return
sigma=1.5 # volatility of asset
muj=0.1 # expected value for distribution of jump size
sigmaj=0.3 # standard deviation for distribution of jump size
lambda=40 #jump intensity
k=exp(muj+0.5*sigmaj^2)-1 #mean of relative jump size
M=100 #number of paths
Xs=1 #starting value for process of X
N=1000 #number of observations
T=1 # length of the interval and time is in [0 , T]
dt=T/N #time increment
t=seq(dt,1,dt) #time increase by dt.
dw=matrix(0,N,M)#create empty matrix for dw
w=matrix(0,N,M)#create empty matrix for w

## stochastic increment of Brownian process ##
for(i in 1:M)
  dw[,i]=sqrt(dt)*rnorm(N)
## cumulative stochastic increment of Brownian process ##
for(i in 1:M)
  w[,i]=cumsum(dw[,i])
## create jump part ##
Jump=matrix(0,N,M) #create empty matrix for jump size
Cumjump=matrix(0,N,M) #create empty matrix for cumulative jump size
X=matrix(0,N,M)      #create empty matrix for process of X
dN=matrix(0,N,M) #create empty matrix for Poisson increment
# constructing Poisson process #
for(j in 1:M){
for(i in 1:N)
{dN[i,j]=rpois(1,lambda*dt)}#define N numbers from Poisson(lambda*dt)}
# according to existing of jump, describe jump sizes
for(j in 1:M){
for(i in 1:N)
{if(dN[i,j]==1){Jump[i,j]=rnorm(1,muj,sigmaj)} #define the jump sizes
else
{Jump[i,j]=0} #there exist no jump then jump size is 0}}
for(i in 1:M)
```

```
Cumjump[,i]=cumsum(Jump[,i]) #define cumulative values for jump sizes.

## constructing analytical solution ##
for(i in 1:N){
for(j in 1:M){
X[i,j]=Xs*exp((alpha-0.5*sigma^2-lambda*k)*t[i]+sigma*w[i,j]+Cumjump[i,j])}}
###################################
## produce the graph of Merton simulations ##
library(ggplot2)
library(tikzDevice)
t=seq(dt,T,dt)
data=data.frame(t,X)
tikz("Merton_Simulations.tex",width=5,height=5)
f1=geom_line(aes(x=t,y=X[,10],color = "black"))
f2=geom_line(aes(x=t,y=X[,20],color = "red"))
f3=geom_line(aes(x=t,y=X[,30],color = "blue"))
f4=geom_line(aes(x=t,y=X[,40],color = "maroon"))
f5=geom_line(aes(x=t,y=X[,50],color = "green"))
f6=geom_line(aes(x=t,y=X[,60],color = "brown"))
f7=geom_line(aes(x=t,y=X[,70],color = "pink"))
f8=geom_line(aes(x=t,y=X[,80],color = "magenta"))
f9=geom_line(aes(x=t,y=X[,90],color = "cyan"))
f10=geom_line(aes(x=t,y=X[,100],color = "peru"))
ggplot(data)+f1+f2+f3+f4+f5+f6+f7+f8+f9+f10+xlab("Time") +ylab("10 Selected Paths Out of 100")+ theme(legend.position = "none")
dev.off()
##################################
## Numerical Approximation by Using Euler-Maruyama Method ##
X_app=matrix(0,N,M)#create empty matrix for approximation values for process of X
Xs=1
for(j in 1:M){
for(i in 1:N){
xapptemp=Xs+dt*(alpha-lambda*k)*Xs+sigma*Xs*dw[i,j]
X_app[i,j]=xapptemp+(xapptemp*(exp(Jump[i,j])-1))*dN[i,j]
Xs=X_app[i,j]}
Xs=1}

#show the convergence of analytical solution and Euler-Maruyama
```

```
    approximations#
install.packages("ggplot2")
install.packages("tikzDevice")
library(ggplot2)
library(tikzDevice)
data1=data.frame(t,X[,25])
colnames(data1)=c("time", "value")
data2=data.frame(t,X_app[,25])
colnames(data2)=c("time", "value")
p=ggplot()+geom_line(data=data1,aes(x=time,y=X[,25],color="Analytical
    Solution"))+geom_line(data=data2,aes(x=time,y=X_app[,25],color="Euler-
    Maruyama Approximation"))+xlab("Time") +ylab("")+ theme(legend.position
    ="bottom")+scale_color_manual(values=c("black", "red"))+ theme(legend.
    title = element_blank())
p1=p + theme(legend.text = element_text(margin = margin(r =15,l = 5, unit =
     "pt"), hjust = 0))
data3=data.frame(t,X[,75])
colnames(data3)=c("time", "value")
data4=data.frame(t,X_app[,75])
colnames(data4)=c("time", "value")
p=ggplot()+geom_line(data=data3,aes(x=time,y=X[,75],color="Analytical
    Solution"))+geom_line(data=data4,aes(x=time,y=X_app[,75],color="Euler-
    Maruyama Approximation"))+xlab("Time") +ylab("")+ theme(legend.position
    ="bottom")+scale_color_manual(values=c("black", "red"))+ theme(legend.
    title = element_blank())
p2=p + theme(legend.text = element_text(margin = margin(r = 15,l =5, unit =
     "pt"), hjust = 0))
#to combine two graphs in one panel, use following packages
install.packages("gridExtra")
library(gridExtra)
tikz("convergence.tex",width=5,height=5)
grid.arrange(p1,p2,nrow=2)
dev.off()
#################################
## Euler convergence check##
Xs=1 # initial value
delta=5 # number of different time interval (dt) sizes
Xerr=matrix(0,M,delta) # create empty matrix for error
# Analytical solution
for(j in 1:M){
```

```
Xs=1
Xtrue=Xs*exp((alpha-0.5*sigma^2-lambda*k)*t[N]+sigma*w[N,j]+Cumjump[N,j])
R=c(1,2,5,10,20) #multiplier of dt
for(p in 1:5){
Dt=R[p]*dt #time interval for euler approximation
L=N/R[p] #number of observations for euler approximation
Xs=1
for(i in 1:L){
Winc=sum(dw[(R[p]*(i-1)+1):(R[p]*i),j]) #Brownian increment for euler approximation
Ninc=sum(Jump[(R[p]*(i-1)+1):(R[p]*i),j]) #Poisson increment for euler approximation
xtemperr=Xs+Dt*(alpha-lambda*k)*Xs+sigma*Xs*Winc
X_em=xtemperr+(xtemperr*(exp(Ninc)-1)) #Euler approximation
Xs=X_em}
Xerr[j,p]=abs(X_em-Xtrue) #assign each difference between analytical and approximation result to this matrix}}
#Present the result of mean of error through the M paths
colSums(Xerr)/100
################################
### MLE of SDE with Jump ###
dx=matrix(0,N,M) # create empty matrix for difference of X states
for(j in 1:M){
dx[,j]=c(log(X[1,j])-log(1),diff(log(X[,j]))) # find difference of X states }
f=matrix(0,N,10) # create empty matrix for transition density
estimate=matrix(0,M,5)
for(v in 1:M){
dif=dx[,v]
likelihood=function(theta,dif,dt) # create a function for likelihood{
Q1=theta[1] # symbolize alpha parameter
Q2=theta[2] # symbolize sigma parameter
Q3=theta[3] # symbolize muj parameter
Q4=theta[4] # symbolize sigmaj parameter
Q5=theta[5] # symbolize lambda parameter
#assign element of transition density matrix for N elements and k jump possibilities.
for(i in 1:N){
for(j in 1:10)
{f[i,j]=(exp(-Q5*dt)*(Q5*dt)^(j-1)/factorial(j-1))*(1/sqrt(2*pi*(Q2^2*dt+(j
```

```
    -1)*Q4^2)))*exp(-(dif[i]-((Q1-Q5*k-Q2^2/2)*dt+(j-1)*Q3))^2/(2*(Q2^2*dt
    +(j-1)*Q4^2))))}}
R=rowSums(f)
LL=-sum(log(R)) # find -loglikelihood
return(LL)}
## Minimize -loglikelihood function ##
estimation=optim(c(1,1,1,1,1),likelihood,gr=NULL,dif,dt,method="L-BFGS-B",
lower=c(-Inf,0,-Inf,0,-Inf), upper = c(Inf,Inf,Inf,Inf,Inf),hessian=T)
options(scipen=999)
estimate[v,]=estimation$par # assign each parameter estimation set to
    estimate matrix.
}
parameter_estimations=colSums(estimate)/M
##############################
```

Listing A.3: Merton Parameter Estimation in R

```
##Create data with jump points
set.seed(123)
N=100
T=1
h=T/N
lambda=5
muj=0
sigmaj=1
alpha=3
sigma=2
k=exp(muj+0.5*sigmaj^2)-1
z=rnorm(N)
e=rexp(100)
v=rnorm(N,muj,sigmaj)
i=0
n=0
s=0
X_s=3
A_s=0
E=cumsum(e)
X=NULL
Xjump=NULL
tjump=NULL
t=NULL
while(s<T){
A_temp=A_s+lambda*((i+1)*h-s)
if(A_temp>=E[n+1]){
tjump[n+1]=s+(E[n+1]-A_s)/lambda
Xjumpleft=X_s+(alpha-lambda*k)*X_s*(tjump[n+1]-s)+sigma*X_s*sqrt(tjump[n+1
    ]-s)*z[i+1]
X[i+1]=Xjumpleft+Xjumpleft*(exp(v[n+1])-1)
Xjump[n+1]=X[i+1]
t[i+1]=tjump[n+1]
s=tjump[n+1]
A_s=E[n+1]
X_s=X[i+1]
n=n+1}
else{
X[i+1]=X_s+(alpha-lambda*k)*X_s*((i+1)*h-s)+sigma*X_s*sqrt((i+1)*h-s)*z[i+1
    ]
```

```
s=(i+1)*h
t[i+1]=(i+1)*h
A_s=A_temp
X_s=X[i+1]}
i=i+1}
data=data.frame(t,X)
colnames(data)=c("time", "value")

library(ggplot2)
library(tikzDevice)

##Plot the data##
library(ggplot2)
tikz("Jump_detection.tex",width=5,height=5)
ggplot(data,aes(x=time,y=value))+geom_line()
dev.off()
```

Listing A.4: Jump Detection R Script

```
data=read.table("d10.txt",header=TRUE)
Y=data[,2]
X=data[,1]
Sys.setlocale("LC_TIME", "English")
####################################################
## produce the graph of data##
install.packages("ggplot2")
install.packages("tikzDevice")
library(ggplot2)
library(tikzDevice)
tikz("Exchangepath.tex",width=5,height=3)
ggplot(data = data, aes(x = as.Date(X,format='%d.%m.%Y'), y = Y))+geom_line
    (color = "maroon")+ scale_x_date(date_labels = "%b %Y", breaks='1 month
    ')+xlab("Time") +ylab("")+ theme(legend.position = "none")
dev.off()
```

Listing A.5: Dollar to Turkish Lira Exchange Rate

```
data=read.table("d10.txt",header=TRUE)
X=data[,2]
dx=diff(log(X))
N=length(X)
T=1
dt=T/N
##########################################################
### MLE of SDE with Jump ###
dx=diff(log(X)) # find difference of X states
f=matrix(0,N-1,10) # create empty matrix for transition density
likelihood=function(theta,dx,dt) # create a function for likelihood
{
Q1=theta[1] # symbolize alpha parameter
Q2=theta[2] # symbolize sigma parameter
Q3=theta[3] # symbolize muj parameter
Q4=theta[4] # symbolize sigmaj parameter
Q5=theta[5] # symbolize lambda parameter
k=exp((Q3)+0.5*(Q4)^2)-1
#assign element of transition density matrix for N elements and k jump
    possibilities.
for(i in 1:N-1){
for(j in 1:10)
{f[i,j]=(exp(-Q5*dt)*(Q5*dt)^(j-1)/factorial(j-1))*(1/sqrt(2*pi*(Q2^2*dt+(j
   -1)*Q4^2)))*exp(-(dx[i]-((Q1-Q5*k-Q2^2/2)*dt+(j-1)*Q3))^2/(2*(Q2^2*dt+(
   j-1)*Q4^2)))}}
R=rowSums(f)
LL=-sum(log(R)) # find -loglikelihood
return(LL)}
likelihood(c(1,1,1,1,1),dx,dt)
## Minimize -loglikelihood function ##
estimation=optim(c(1,1,1,1,1),likelihood,gr=NULL,dx,dt,method="Nelder-Mead"
    ,hessian=T)
options(scipen=999)
estimation$par
AIC=-2*likelihood(c(1,1,1,1,1),dx,dt)+2*5
```

Listing A.6: Merton Jump Diffusion Model Parameter Estimations For Exchange Rate

```
data=read.table("d10.txt",header=TRUE)
X=data[,2]
N=length(X)
T=1
dt=T/N
dx=diff(log(X)) # find difference of X states
f=matrix(0,N-1,1) # create empty matrix for transition density
likelihood_bs<-function(theta,dx,dt)
{
Q1<-theta[1]
Q2<-theta[2]
for(i in 1:N-1)
{
f[i]=(1/(sqrt(2*pi*Q2^2*dt)))*exp(-(dx[i]-Q1*dt)^2/(2*(Q2^2)*dt))
}
LL=-sum(log(f)) # find -loglikelihood
return(LL)
}
likelihood_bs(c(1,1),dx,dt)
## Minimize -loglikelihood function ##
estimation=optim(c(1,1),likelihood_bs,gr=NULL,dx,dt,method="Nelder-Mead",
    hessian=T)
options(scipen=999)
estimation$par
AIC=-2*likelihood_bs(c(1,1),dx,dt)+2*2
```

Listing A.7: Black-Scholes Model Parameter Estimations For Exchange Rate

```
alpha=0.238 #instantaneous expected return
sigma=0.063 # volatility of asset
muj=-0.015 # expected value for distribution of jump size
sigmaj=0.058 # standard deviation for distribution of jump size
lambda=4.44 #jump intensity
k=exp(muj+0.5*sigmaj^2)-1 #mean of relative jump size
M=1 #number of paths
Xs=5.82 #starting value for process of X
N=9 #number of observations
dt=1/121
dw=matrix(0,N,M)#create empty matrix for dw
w=matrix(0,N,M)#create empty matrix for w
t=seq(dt,1,dt)
set.seed(718)
## stochastic increment of Brownian process ##
for(i in 1:M)
  dw[,i]=sqrt(dt)*rnorm(N)
## cumulative stochastic increment of Brownian process ##
for(i in 1:M)
  w[,i]=cumsum(dw[,i])
## create jump part ##
Jump=matrix(0,N,M) #create empty matrix for jump size
Cumjump=matrix(0,N,M) #create empty matrix for cumulative jump size
X=matrix(0,N,M)     #create empty matrix for process of X
dN=matrix(0,N,M) #create empty matrix for Poisson increment
# constructing Poisson process #
for(j in 1:M){
for(i in 1:N)
{dN[i,j]=rpois(1,lambda*dt)}#define N numbers from Poisson(lambda*dt)
}
# according to existing of jump, describe jump sizes
for(j in 1:M){
for(i in 1:N)
{if(dN[i,j]==1){Jump[i,j]=rnorm(1,muj,sigmaj)} #define the jump sizes
else
{Jump[i,j]=0} #there exist no jump then jump size is 0
}}
for(i in 1:M)
Cumjump[,i]=cumsum(Jump[,i]) #define cumulative values for jump sizes.
## constructing analytical solution ##
```

```
for(i in 1:N){
for(j in 1:M){
X[i,j]=Xs*exp((alpha-0.5*sigma^2-lambda*k)*t[i]+sigma*w[i,j]+Cumjump[i,j])
    }}
Y=matrix(c(5.76,5.81,5.80,5.78,5.77,5.79,5.74,5.65,5.65),N,M) #actual
    values
sum(abs(X-Y)/Y)/N*100 # Give MAPE value
```

Listing A.8: Dollar to Turkish Lira Exchange Rate Merton Prediction

```
mu=0.112#instantaneous expected return
sigma=0.114 # volatility of asset
M=1 #number of paths
Xs=5.8214 #starting value for process of X
N=9 #number of observations
dt=1/121
dw=matrix(0,N,M)#create empty matrix for dw
w=matrix(0,N,M)#create empty matrix for w
t=seq(dt,59*dt,dt)
set.seed(603)
## stochastic increment of Brownian process ##
for(i in 1:M)
  dw[,i]=sqrt(dt)*rnorm(N)
## cumulative stochastic increment of Brownian process ##
for(i in 1:M)
  w[,i]=cumsum(dw[,i])
X=matrix(0,N,M)
## constructing analytical solution ##
for(i in 1:N)
{
for(j in 1:M)
{
X[i,j]=Xs*exp((mu-0.5*sigma^2)*t[i]+sigma*w[i,j])}}
Y=matrix(c(5.76,5.81,5.80,5.78,5.77,5.79,5.74,5.65,5.65),N,M) #Actual
    values
sum(abs(X-Y)/Y)/N*100 #MAPE value
```

Listing A.9: Dollar to Turkish Lira Exchange Rate Black-Scholes Prediction

```
data=read.table( "d10.txt" ,header=TRUE)
q1=data[,2][-length(data[,2])]
q2=data[,2][-1]
b=log(q2/q1)
dt=1/length(length(data[,2]))

##########Jump Detection##########
epsilon=0.0125
c=which(abs(b)>epsilon)
index=matrix(0,length(b),1)
for(i in 1:length(c)){
index[c[i]]=1}
k=b-index*(b)
a=(k-mean(k))/sd(k)
*************************************************************
install.packages("ggplot2")
library(ggplot2)
t=seq(2*dt,1,dt)
data2=data.frame(t,b)
colnames(data2)=c("Time", "LogReturns")
ggplot(data2,aes(x=Time,y=LogReturns))+geom_line()
**************************************************************
##########Model Assumptions##########
library(tseries)
#######Stationarity#######
kpss.test(b)
adf.test(z)
#######Normality#######
shapiro.test(a)
jarque.bera.test(z)
#######Independency#######
Box.test (b, lag = 1)
```

Listing A.10: Jump Detection and Model Assumption Check

APPENDIX B

DOLLAR/TL EXCHANGE RATE DATA

Date	Exchange Rate	Date	Exchange Rate
1.02.2019	5.2084	8.03.2019	5.4303
3.02.2019	5.2167	10.03.2019	5.4338
4.02.2019	5.2182	11.03.2019	5.4455
5.02.2019	5.1985	12.03.2019	5.4572
6.02.2019	5.2168	13.03.2019	5.4564
7.02.2019	5.2668	14.03.2019	5.4671
8.02.2019	5.2474	15.03.2019	5.4467
10.02.2019	5.2517	17.03.2019	5.449
11.02.2019	5.2779	18.03.2019	5.4683
12.02.2019	5.2516	19.03.2019	5.4764
13.02.2019	5.285	20.03.2019	5.4208
14.02.2019	5.2729	21.03.2019	5.4652
15.02.2019	5.2714	22.03.2019	5.7627
17.02.2019	5.2854	24.03.2019	5.693
18.02.2019	5.3083	25.03.2019	5.5503
19.02.2019	5.2828	26.03.2019	5.3279
20.02.2019	5.3244	27.03.2019	5.3278
21.02.2019	5.3214	28.03.2019	5.561
22.02.2019	5.3186	29.03.2019	5.5555
24.02.2019	5.3203	31.03.2019	5.5919
25.02.2019	5.3026	1.04.2019	5.4863
26.02.2019	5.3066	2.04.2019	5.605
27.02.2019	5.3136	3.04.2019	5.626
28.02.2019	5.3371	4.04.2019	5.5925
1.03.2019	5.3742	5.04.2019	5.6262
3.03.2019	5.3725	7.04.2019	5.639
4.03.2019	5.3769	8.04.2019	5.6903
5.03.2019	5.387	9.04.2019	5.6943
6.03.2019	5.4288	10.04.2019	5.6818
7.03.2019	5.4712	11.04.2019	5.7388

Date	Exchange Rate	Date	Exchange Rate
12.04.2019	5.7693	19.05.2019	6.0617
14.04.2019	5.7856	20.05.2019	6.0272
15.04.2019	5.8041	21.05.2019	6.0525
16.04.2019	5.7703	22.05.2019	6.1002
17.04.2019	5.7387	23.05.2019	6.0964
18.04.2019	5.8018	24.05.2019	6.0825
19.04.2019	5.8112	26.05.2019	6.0794
21.04.2019	5.8264	27.05.2019	6.0629
22.04.2019	5.8272	28.05.2019	6.0319
23.04.2019	5.8287	29.05.2019	6.0121
24.04.2019	5.8753	30.05.2019	5.8772
25.04.2019	5.9268	31.05.2019	5.8382
26.04.2019	5.9382	2.06.2019	5.8593
28.04.2019	5.9462	3.06.2019	5.8339
29.04.2019	5.9483	4.06.2019	5.7824
30.04.2019	5.9656	5.06.2019	5.731
1.05.2019	5.9665	6.06.2019	5.7801
2.05.2019	5.9645	7.06.2019	5.831
3.05.2019	5.9648	9.06.2019	5.8428
5.05.2019	5.9851	10.06.2019	5.7747
6.05.2019	6.0803	11.06.2019	5.7976
7.05.2019	6.1511	12.06.2019	5.808
8.05.2019	6.186	13.06.2019	5.8682
9.05.2019	6.1926	14.06.2019	5.9015
10.05.2019	5.9851	16.06.2019	5.917
12.05.2019	5.9936	17.06.2019	5.8733
13.05.2019	6.0594	18.06.2019	5.8284
14.05.2019	6.0306	19.06.2019	5.7914
15.05.2019	6.0007	20.06.2019	5.7644
16.05.2019	6.0457	21.06.2019	5.8214
17.05.2019	6.055		