FINANCIAL ENGINEERING PROJECT

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Bawana Road, Delhi-110042

SIMULATION AND ANALYSIS OF STOCK PRICING TECHNIQUES

Submitted By:

Jai Garg (2K18/MC/044)

Jatin Papreja (2K18/MC/049)

INDEX

INTRODUCTION

Businesses today are extremely fragile. Organizations invest an enormous amount of money in order to keep their business running and fully functional. But one cannot just keep on spending lavishly or not spend at all. It can be said that **finance is the fuel of business** today, but its management is equally important for organizations as well as individuals to emerge successful from the risk offered by financial markets. **Risk** is inherent in financial and commodity markets.

All investment instruments in the financial markets face risks in terms of the constant fluctuation in prices, which increases the Investor's exposure to such risks. To derive positive returns, an investor must accurately predict the movement of the market, whether it will move up or down. A wrong prediction guarantees a **substantial loss**. Thus, **predicting Stock Price accurately** plays an important role in order to gain advantage in the financial market.

Figure 1: Stock Price Graph

STOCK PRICING

A **stock** is a security that represents the ownership of a fraction of a corporation. This entitles the owner of the stock to a proportion of the corporation's assets and profits equal to how much stock they own. Units of stock are called "**shares**." Stocks are bought and sold predominantly on stock exchanges.

There are two main types of stock:

- **Common Stock:** Common stock usually entitles the owner to vote at shareholders' meetings and to receive any dividends paid out by the corporation.
- **Preferred Stock:** Preferred stockholders generally do not have voting rights, though they have a higher claim on assets and earnings than the common stockholders.

In financial markets, stock pricing is the method of calculating theoretical values of companies and their stocks. The main use of these methods is to predict future market prices. Thus, to profit from price movement – stocks that are judged **undervalued** (with respect to their theoretical value) are bought, while stocks that are judged **overvalued** are sold, in the expectation that undervalued stocks will overall rise in value, while overvalued stocks will generally decrease in value.

FACTORS AFFECTING STOCK PRICES

The factors affecting the movement of a stock price can be broken down into three categories:

Fundamental Factors: Fundamental factors drive stock prices based on a company's earnings and profitability from producing and selling goods and services. These include:

- The level of the earnings base measured by earnings per share, cash flow per share and dividends per share.
- The expected growth in the earnings base.
- The discount rate, which is itself a function of inflation.
- The perceived risk of the stock.

Technical Factors: Technical factors relate to a stock's price history in the market pertaining to chart patterns, momentum, and behavioral factors of traders and investors. These include:

- **Inflation/Deflation**: It signifies the gain/loss in pricing power for companies.
- **Economic Strength of Market and Peers**: Company stocks tend to track with the market and with their sector or industry peers. A negative outlook for one retail stock often hurts other retail stocks as "guilt by association" drags down demand for the whole sector.
- **Incidental Transactions**: Incidental transactions are purchases or sales of a stock that are motivated by something other than belief in the intrinsic value of the stock. These transactions include executive insider transactions, which are often pre-scheduled or driven by portfolio objectives.
- **Demographics of Customer Base**: Demographic data is collected to build a profile for the organization's customer base. The hypothesis is that the greater the proportion of middle-aged investors among the investing population, the greater the demand for equities and the higher the valuation multiples.
- **Liquidity**: It refers to how much interest from investors a specific stock attracts. Large-cap stocks have high liquidity—they are well followed and heavily transacted. Many small-cap stocks suffer from an almost permanent "liquidity discount" because they simply are not on investors' radar screens.

Market Sentiment: Market sentiment refers to the psychology of market participants, individually and collectively. Market sentiment is often **subjective, biased, and obstinate**. It starts with the assumption that markets are apparently not efficient much of the time, and this inefficiency can be explained by psychology and other social science disciplines.

Different types of investors depend on different factors. **Short-term investors** tend to incorporate and may even prioritize technical factors. **Long-term investors** prioritize fundamentals and recognize that technical factors play an important role.

DERIVATIVES

A **derivative** is a financial security with a value that is reliant upon or derived from, an **underlying asset** or group of assets—a benchmark. The derivative itself is a contract between two or more parties, and the derivative derives its price from fluctuations in the underlying asset. The most common underlying assets for derivatives are **stocks, bonds, commodities, currencies, interest rates, and market indexes**. These assets are commonly purchased through brokerages. Derivatives can be used to hedge a position, speculate on the directional movement of an underlying asset, or give leverage to holdings. Common forms of Derivatives include: **Future Contract, Forward Contract, Swaps and Options.**

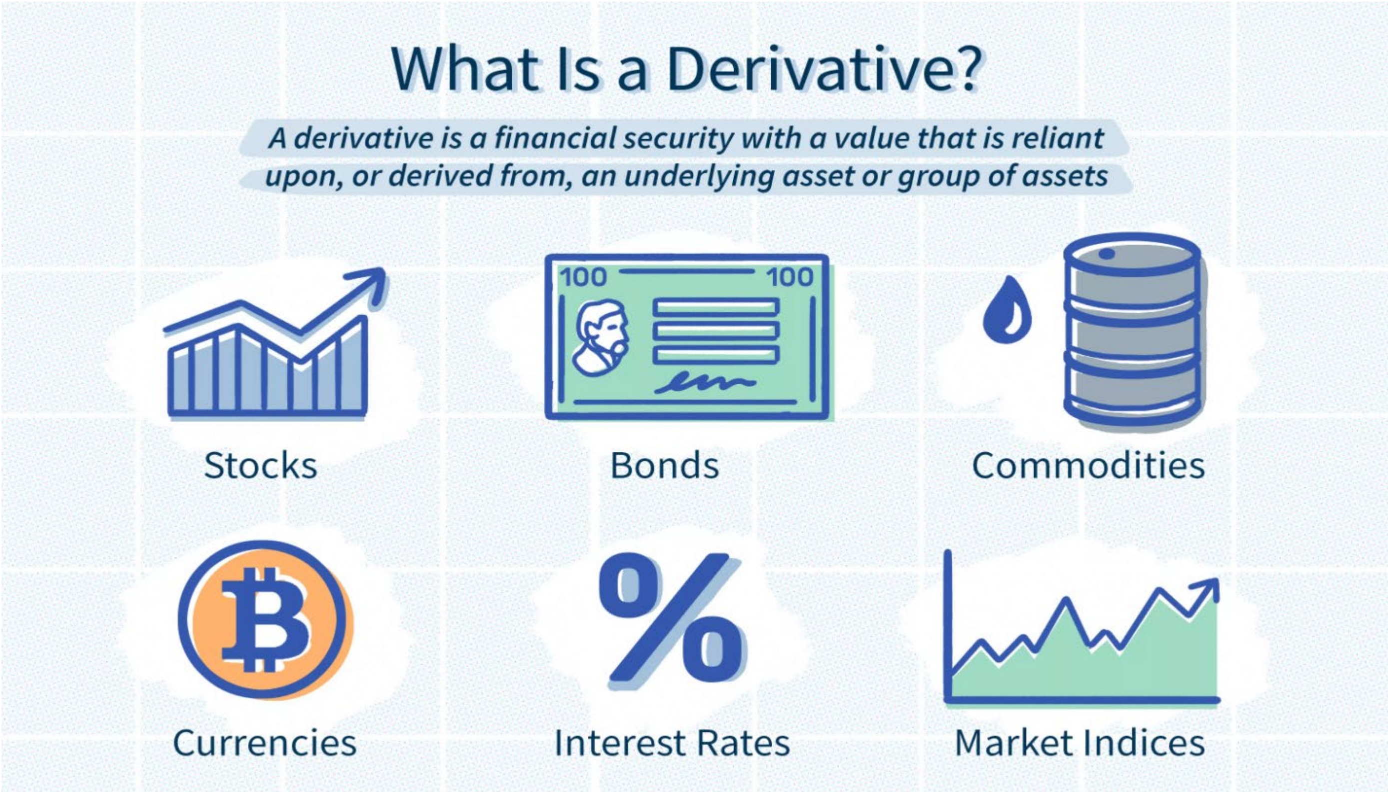


Figure 2: Underlying Assets for Derivatives

FINANCIAL MATHEMATICS

Financial Mathematics describes the application of mathematics and mathematical modeling to solve **financial problems**. It is sometimes referred to as **quantitative finance, financial engineering,** and **computational finance**. It combines tools from statistics, probability, and stochastic processes with economic theory. Most of the products have been developed through techniques in the field of financial engineering. Using mathematical modeling and computer science, financial engineers are able to test and issue new tools such as **new methods of investment analysis, new debt offerings, new investments, new trading strategies, new financial models,** etc. Investment banks, commercial banks, hedge funds, insurance companies and regulatory agencies apply the methods of financial mathematics to such problems as derivative securities valuation, portfolio structuring, risk management, and scenario simulation.

BACKGROUND KNOWLEDGE

STOCHASTIC PROCESS

A **stochastic process** is defined as a collection of random variables defined on a common **probability space** $(\boldsymbol{\Omega}, \boldsymbol{\mathcal{F}}, \boldsymbol{P})$, where $\boldsymbol{\Omega}$ is a **sample space**, $\boldsymbol{\mathcal{F}}$ is a $\boldsymbol{\sigma}$ - **algebra**, and $\boldsymbol{P}$ is a **probability measure**; and the random variables, indexed by some set $\boldsymbol{T}$, all take values in the same mathematical space $\boldsymbol{S}$, which must be measurable with respect to some $\boldsymbol{\sigma}$ - **algebra** $\sum$. In other words, for a given probability space $(\boldsymbol{\Omega}, \boldsymbol{\mathcal{F}}, \boldsymbol{P})$ and a measurable space $(\boldsymbol{S}, \sum)$, a stochastic process is a collection of $\boldsymbol{S}$-valued random variables, which can be written as:

$$\{\, \boldsymbol{X}(t) : t \in \boldsymbol{T} \,\}$$

BROWNIAN MOTION

In mathematics, the **Brownian Motion** is a real valued continuous-time stochastic process. It plays an important role in both pure and applied mathematics. In pure mathematics, it gave rise to the study of **continuous time martingales**. It is a key process in terms of which more complicated stochastic processes can be described.
A Stochastic Process (SP) $\{W(t), t \geq 0\}$ is said to be a Brownian Motion (BM) if it satisfies the following properties:

- $W(0) = \ 0$ ie, it starts from zero.
- for $t > 0$, the sample path of $W(t)$ is continuous.
- $W(t), t \geq 0$ has independent and stationary increment.
- for $0 \leq s < t < \infty$, $W(t)$ - $W(s)$ normally distributed random variable with mean '0' and variance $t - s$.

STOCHASTIC DIFFERENTIAL EQUATION

A **stochastic differential equation (SDE)** is a differential equation in which one or more of the terms are a stochastic process, resulting in a solution, which is also a stochastic process. SDEs are used to model various phenomena such as unstable stock prices or physical systems subject to thermal fluctuations.
A **typical SDE** is of the form:

$$dX(t) = b\big(t, X(t)\big)\, dt + \sigma\big(t, X(t)\big) dW(t)$$

where W denotes a Wiener Process or Brownian Motion.
It can equivalent be written as:

$$X(t) = X(0) + \int_0^t b\big(s, x(s)\big)\, ds + \int_0^t \sigma\big(s, x(s)\big)\, dW\,(s)$$

which is known as **Stochastic Integral Equation.**

GEOMETRIC BROWNIAN MOTION

A **Geometric Brownian Motion (GBM)** (also known as **Exponential Brownian Motion**) is a continuous-time stochastic process in which the logarithm of the randomly varying quantity follows a Brownian motion (also called a Wiener process) with drift. It is an important example of stochastic processes satisfying a stochastic differential equation (SDE); in particular, it is used in mathematical finance to model stock prices in the **Black–Scholes model**.

A Stochastic Process $S(t)$ is said to follow a GBM if it satisfies the following Stochastic Differential Equation:

$$dS(t) = \mu S(t)\,dt + \sigma S(t) dW(t)$$

where
$W(t)$ is a Brownian Motion,
μ is drift which is used to model deterministic trends,
σ is volatility which is often used to model a set of unpredictable events

The Solution of SDE of GBM is given as:

$$S(t) = S(0)e^{\left(\mu - \frac{\sigma^2}{2}\right)t + \sigma W(t)}$$

Here, $S(t)$ follows Lognormal Distribution where

$E\big(S(t)\big) = S(0)e^{\mu t}$

$Var\big(S(t)\big) = S^2(0)e^{2\mu t}\left(e^{\sigma^2 t} - 1\right)$

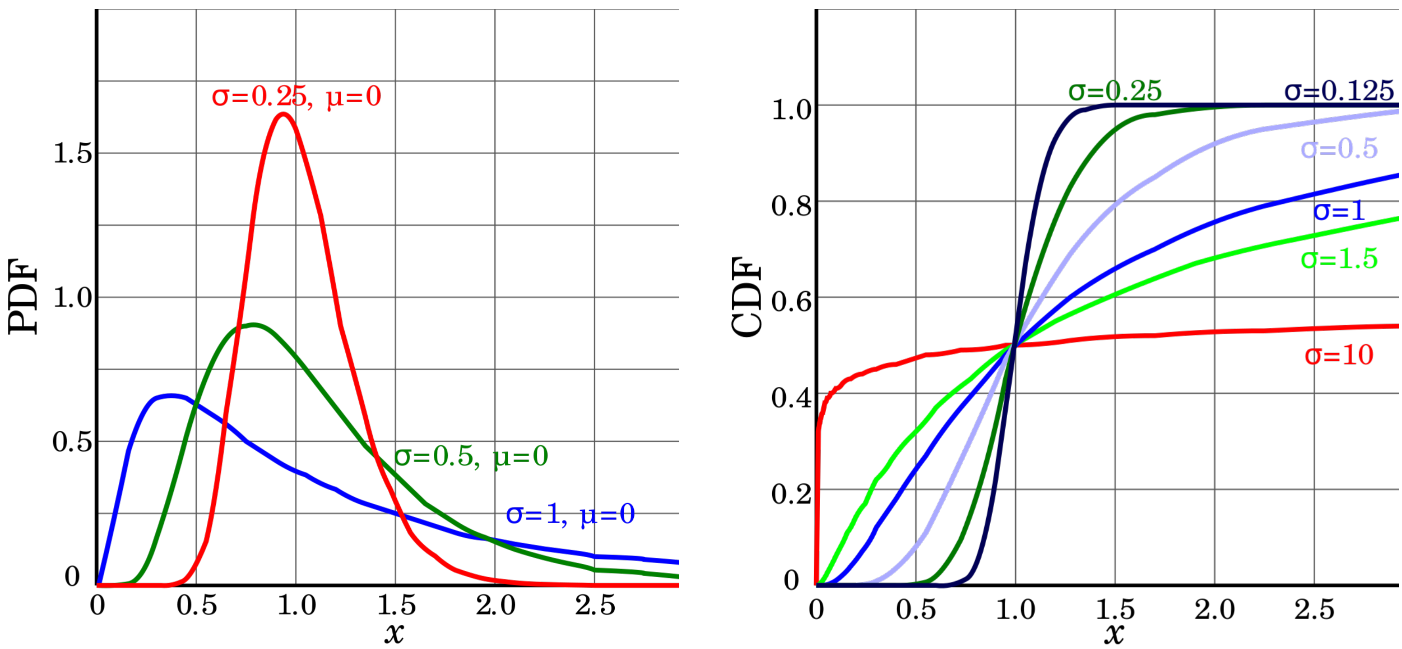


Figure 3: PDF and CDF of Lognormal Distribution

However, GBM is not a completely realistic model due to following assumptions:

- **Volatility is assumed to be constant**.
- **Path is assumed to be continuous i.e. no jumps are considered**.

MERTON-JUMP DIFFUSION

Geometric Brownian Motion assumes that the path is continuous and no erratic movement known as **Jumps** occur during the sampling. A more realistic model should consist of jump component to better simulate true nature of a process. The best-known model of this type in finance is the **Merton-Jump Diffusion model.**

This model superimposes a **jump component** on a **diffusion component**. The diffusion component is the familiar Geometric Brownian motion. The jump component is composed of lognormal jumps driven by a Poisson process. It models the sudden changes in the stock price because of the arrival of important new information.

A Stochastic Process $S(t)$ is said to follow Merton-Jump Diffusion model if it satisfies the following Stochastic Differential Equation:

$$dS(t) = S(t)\left(\mu - \lambda\bar{k}\right) dt + \sigma dW(t) + kdQ(t)$$

where
$Q(t)$ is compound Poisson Process with intensity λ.
k denotes the magnitude of the random jump.
$\bar{k}$ denotes the expected value of k.

$$\bar{k} = e^{\gamma+\frac{\delta^2}{2}} - 1$$

The Solution of SDE of Merton-Jump Diffusion is given as:

$$S(t) = S(0)e^{\left(\mu-\lambda\bar{k}-\frac{\sigma^2}{2}\right)t+\sigma W(t)}U\left(N(t)\right)$$

where
$N(t)$ is Poisson Process with intensity λ

$$U\left(N(t)\right) = \prod_{i=0}^{N(t)}(1 + k_i)$$

$N(t)$ denotes the number of jumps that occur up to time t. As k > −1, stock prices will stay positive. The Geometric Brownian Motion, the lognormal jumps, and the Poisson Process are assumed to be independent.

HESTON PROCESS

In finance, the **Heston model** is a mathematical model describing the evolution of the volatility of an underlying asset. It is a stochastic volatility model, which assumes that the volatility of the asset is not constant, nor even deterministic, but follows a random process. It is more realistic model as compared to GBM, which assumes constant volatility. The basic Heston model assumes that $S(t)$, the price of asset, is determined by a stochastic process.

$$dS(t) = \mu S(t)\, dt + \sqrt{v(t)} S(t) dW_s(t)$$

Where $v(t)$, the instantaneous variance given as:

$$dv(t) = K(\theta - v(t))\, dt + \xi \sqrt{v(t)} dW_v(t)$$

And $W_s(t)$ and $W_v(t)$ are Brownian Motions with correlation ρ, or equivalently, with covariance $\rho\, dt$. The parameters in the above equations represent the following:

- μ is the rate of return of the asset.
- θ is the long variance, or long run average price variance.
- κ is the rate at which $v(t)$ reverts to θ.
- ξ is the volatility of the volatility, determines the variance of $v(t)$.

If the parameters obey the following condition known as **Feller condition**, then the process $v(t)$ is strictly positive:

$$2\kappa\theta > \xi^2$$

CONCEPT OF OPTION PRICING

Option pricing theory estimates a value of an option contract by assigning a price, known as a premium, based on the calculated probability, that the contract will finish **In The Money (ITM)** at expiration. The longer that an investor has to exercise the option, the greater the likelihood that it will be ITM and profitable at expiration. This means that longer-dated options are more valuable. Similarly, the more volatile the underlying asset, the greater the odds that it will expire ITM. Thus, option pricing theory provides an evaluation of an option's fair value, which traders incorporate into their strategies. Models account for variables such as current market price, strike price, volatility, interest rate, and time to expiration to determine the theoretical value an option. Some commonly used models to value options are Black-Scholes and Binomial Option Pricing.

ROOT MEAN SQUARED ERROR

Root Mean Squared Error (RMSE) is the square root of the mean of the square of all of the errors. The use of RMSE is very common, and it is considered an excellent general-purpose error metric for numerical predictions.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (S_i - O_i)^2}{n}}$$

where O_i are the observations, S_i are predicted values of a variable, and n is the number of observations available for analysis.

RMSE is a good measure of accuracy, but only to compare prediction errors of different models or model configurations for a particular variable and not between variables, as it is scale-dependent.

SIMULATIONS AND ANALYSIS

• GEOMETRIC BROWNIAN MOTION

The figure below shows the Monte-Carlo Simulation of Stock Prices under GBM consisting of 500 paths for a stock with initial stock price, $S(0)$ = \$100, μ = 0.8, σ = 0.8, T = 3 months and is traded only for Business Days.

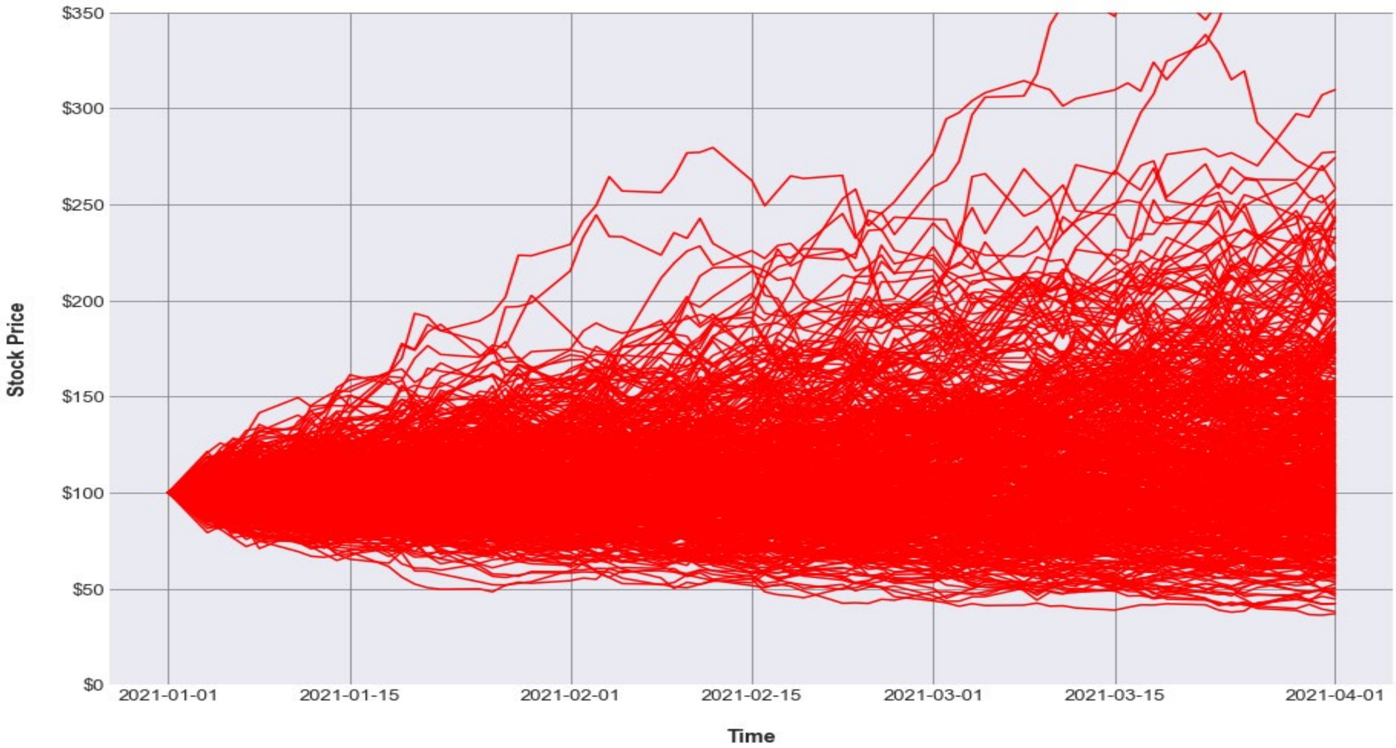


Figure 4: Monte-Carlo Simulation of Stock Prices under Geometric Brownian

The following figure shows the Python Code for generating paths for Monte-Carlo Simulation. It considers the stock price at previous time step as initial stock price and applies Geometric Brownian Motion to predict the stock price at current time step. This process is repeated for all paths at every time step.

```python
def generate_paths(self, fixed_seed=False, day_count=365.):
    if self.time_grid is None:
        self.generate_time_grid()

    M = len(self.time_grid)
    I = self.paths

    paths = np.zeros((M, I))
    paths[0] = self.initial_value
    short_rate = self.short_rate

    for t in range(1, len(self.time_grid)):
        dt = (self.time_grid[t] - self.time_grid[t-1]).days / day_count
        ran = np.random.normal(loc = 0.0, scale = np.sqrt(dt), size = (1, I))

        paths[t] = paths[t - 1] * np.exp((short_rate - (0.5 * (self.volatility**2)))*dt + (self.volatility*ran))

    self.instrument_values = paths
```

Figure 5: Code Cell for Generating Paths for Monte-Carlo Simulation under GBM

PARAMETER ANALYSIS

The figure below shows the movement of Stock Price simulated for different values of Drift. Drift refers to the deviation from the initial Stock Price. As the drift increases, rate of increase of Stock Price increases with respect to time.

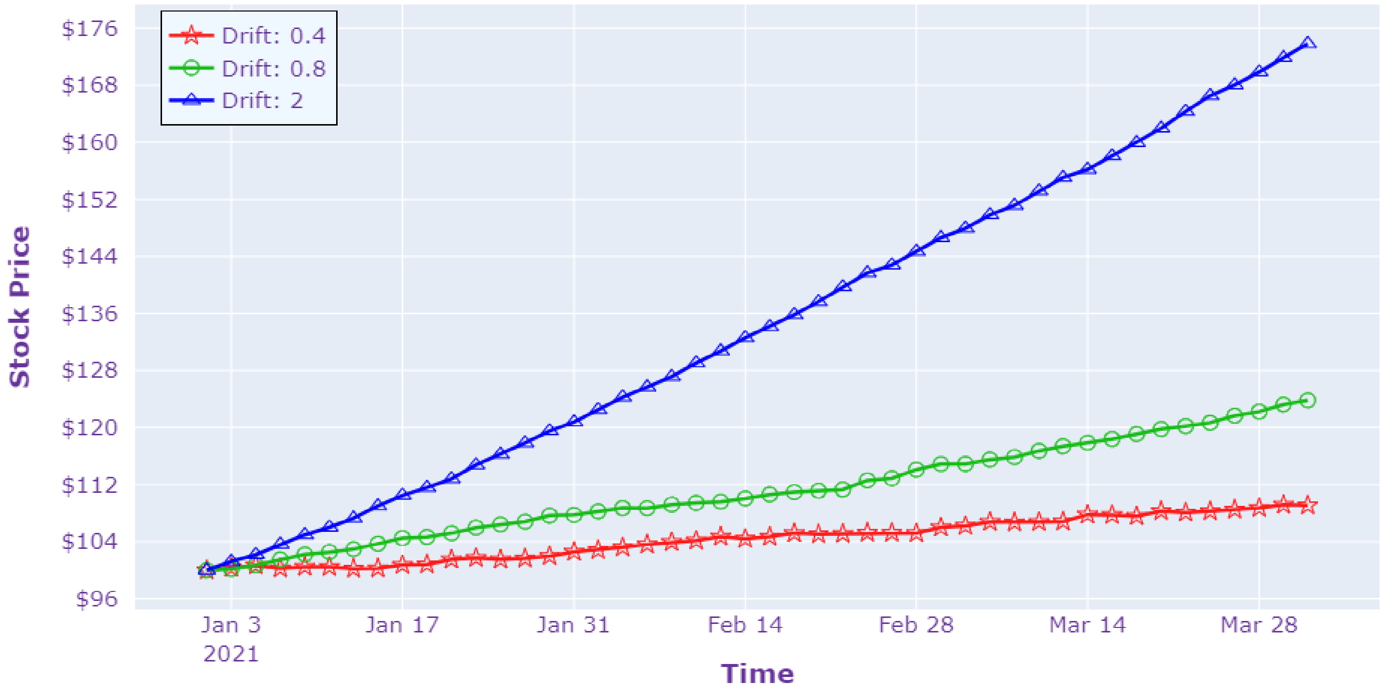


Figure 6: Simulation of Geometric Brownian Motion for Different Values of Drift

The figure below shows the movement of Stock Price simulated for different values of Volatility. As the volatility increases, the nature of stock price curve becomes more 'volatile' i.e. it shows more erratic and drastic changes. This shows the increase in risk as volatility increases.

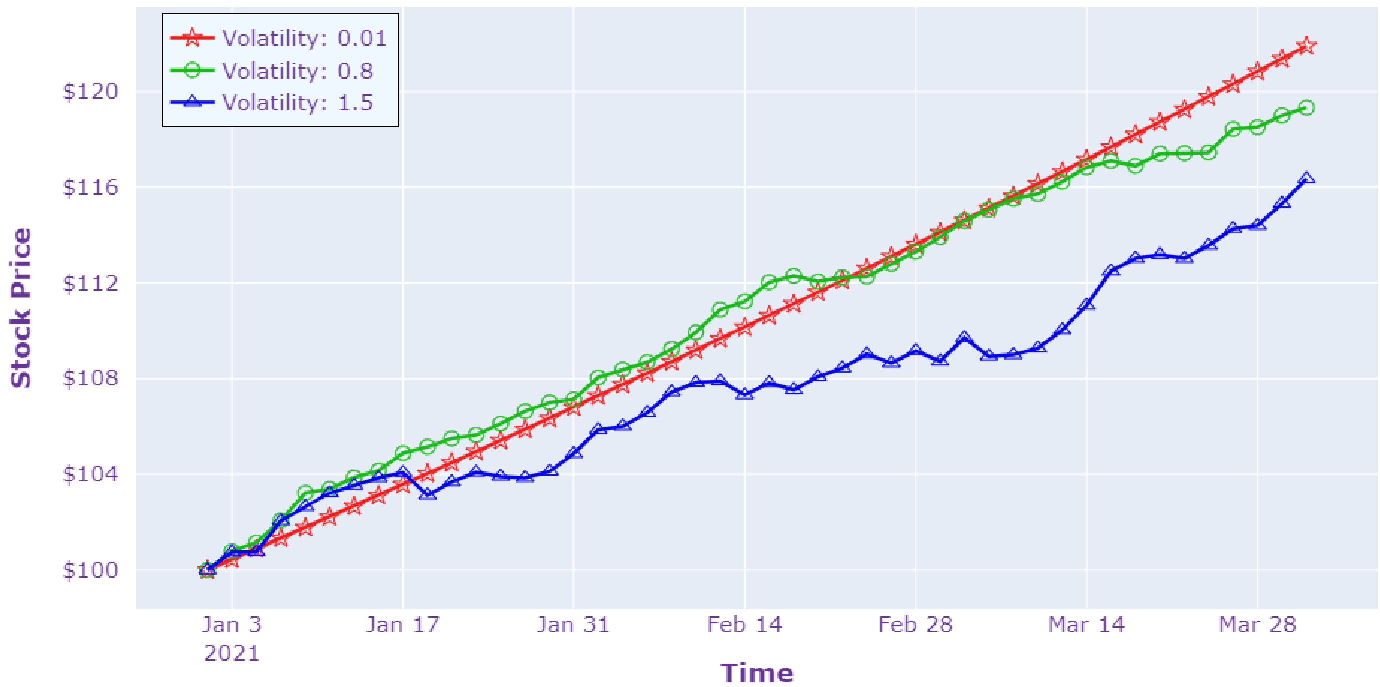


Figure 7: Simulation of Geometric Brownian Motion for Different Values of Volatility (σ)

- ## MERTON-JUMP DIFFUSION

The figure below shows the Monte-Carlo Simulation of Stock Prices under Merton-Jump Diffusion consisting of 500 paths for a stock with initial stock price, $S(0)$ = \$100, μ = 0.8, σ = 0.8, λ = 1, γ = 0.4, δ = 0.05, T = 3 months and is traded only for Business Days. The Jumps are highlighted by darker shade and are clearly visible in some of the paths.

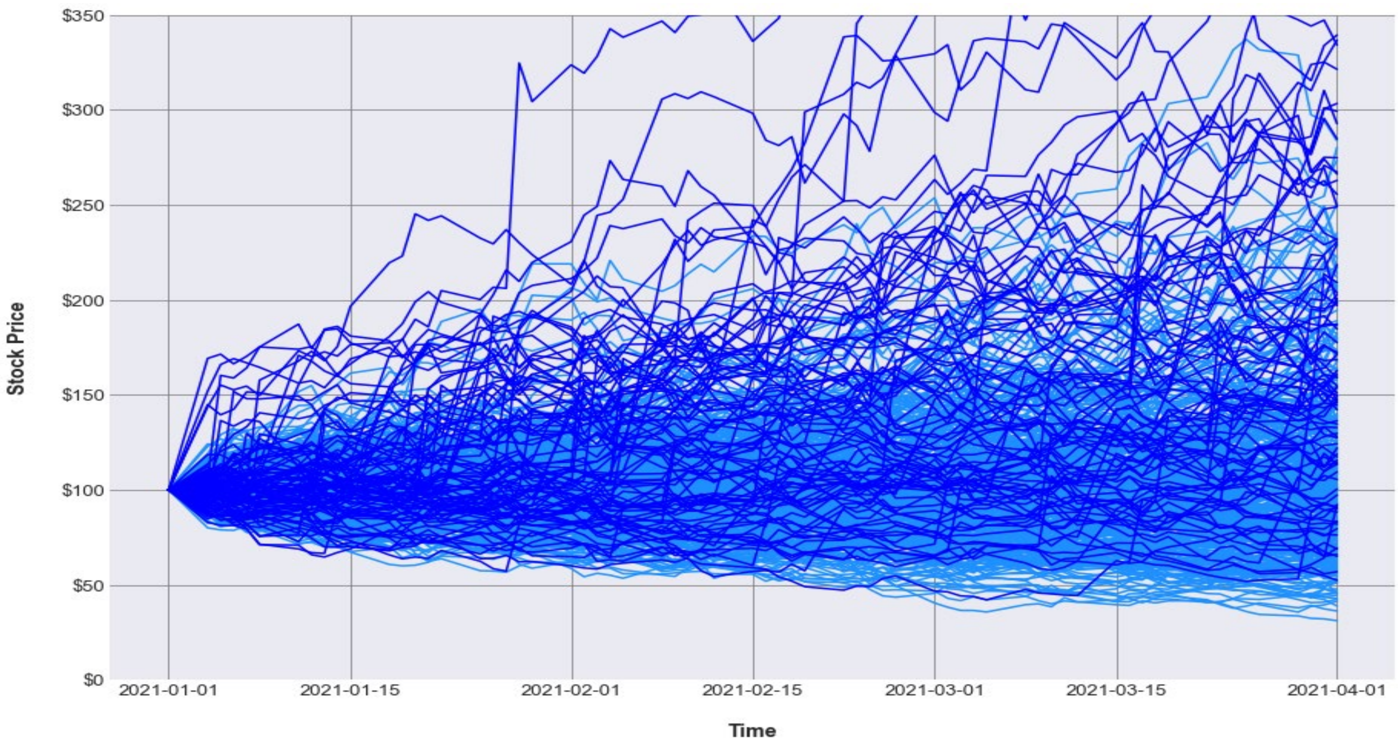


Figure 8: Monte-Carlo Simulation of Stock Prices under Merton Jump Diffusion

The following figure shows the Python Code for generating paths for Monte-Carlo Simulation. It considers the stock price at previous time step as initial stock price and applies Merton-Jump Diffusion to predict the stock price at current time step.

```
def generate_paths(self, fixed_seed=True, day_count=365.):
    if self.time_grid is None:
        self.generate_time_grid()

    M = len(self.time_grid)
    I = self.paths
    paths = np.zeros((M, I))
    paths[0] = self.initial_value
    rj = self.lamb * (np.exp(self.mu + 0.5 * self.delt**2) - 1)
    short_rate = self.short_rate

    for t in range(1, len(self.time_grid)):
        dt = (self.time_grid[t] - self.time_grid[t-1]).days / day_count
        ran1 = np.random.standard_normal((1, I))
        ran2 = np.random.standard_normal((1, I))
        poi = np.random.poisson(self.lamb * dt, I)

        for i in range(self.paths):
            if poi[i] == 1:
                self.index[i] = 1

        paths[t] = paths[t-1] * (np.exp((short_rate - rj - (0.5 * (self.volatility ** 2)))*dt
                                        + (self.volatility*np.sqrt(dt)*ran1))
                                 + ((np.exp(self.mu + self.delt * ran2) - 1) * poi))

    self.instrument_values = paths
```

Figure 9: Code Cell for Generating Paths for Monte-Carlo Simulation under Merton Jump

PARAMETER ANALYSIS

The figure below shows the movement of Stock Price simulated for different values of Lambda. Lambda gives the expected number of jumps. As the value of Lambda increases, the number of jumps increases. As value of Lambda reaches 5, the Stock price curve becomes highly erratic and shows sudden changes.

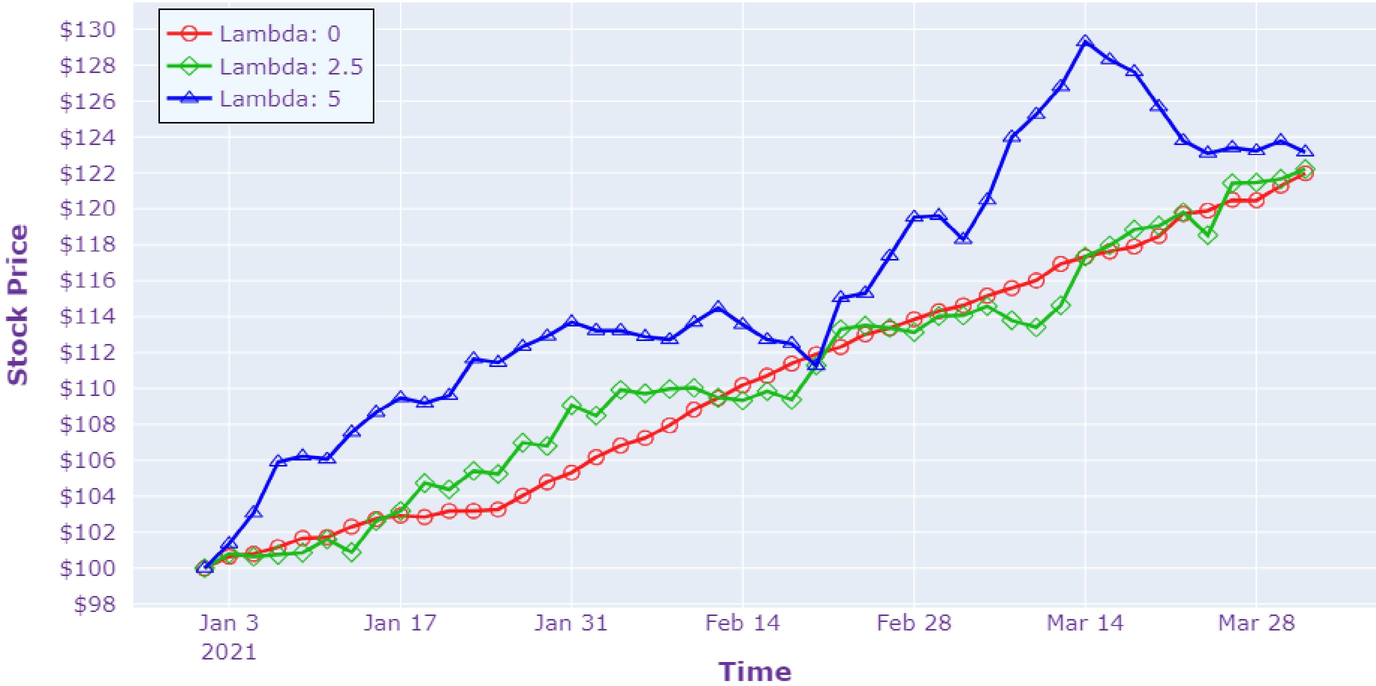


Figure 10: Simulation of Merton-Jump Diffusion for Different Values of Lambda (λ)

The figure below shows the movement of Stock Price simulated for different values of Gamma. Gamma is the expected jump size in Stock Price. As Gamma increases, the stock price curve follows a zigzag pattern indicating big jumps in both upward and downward direction. Smaller values of Gamma give smoother Stock Price curves.

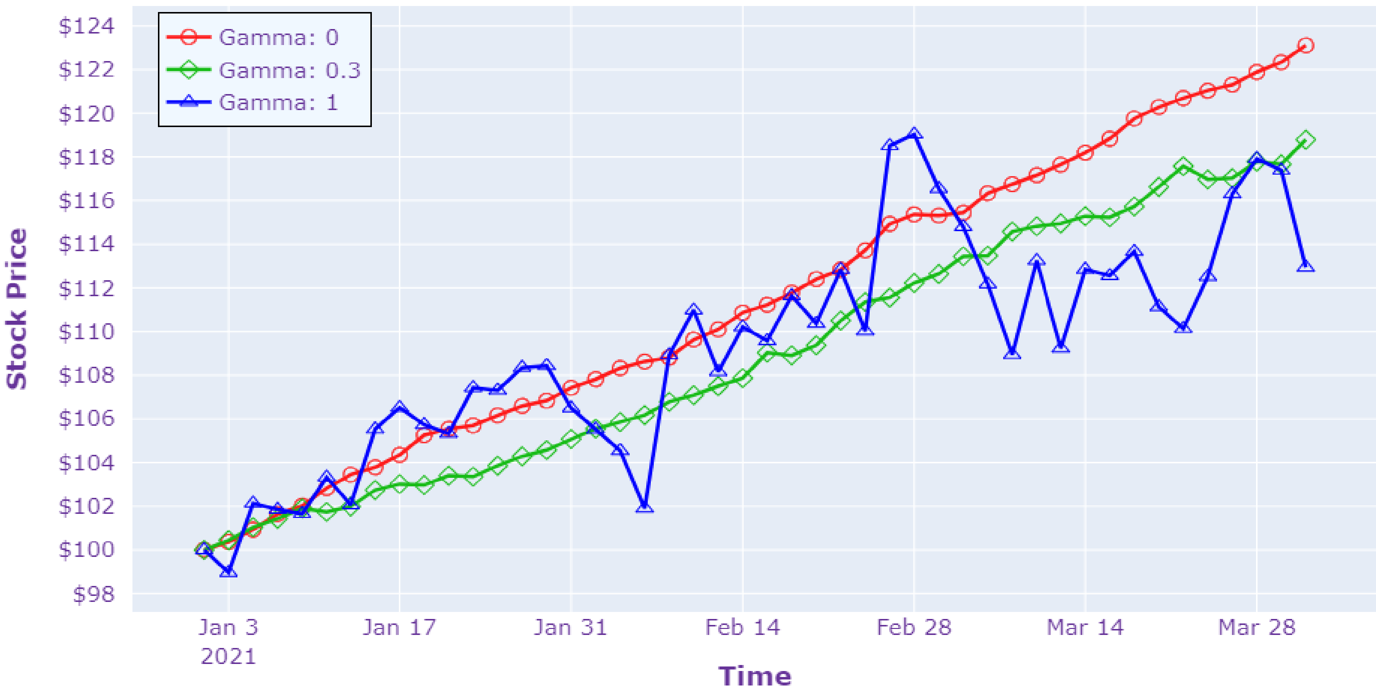


Figure 11: Simulation of Merton-Jump Diffusion for Different Values of Gamma (γ)

- **HESTON PROCESS**

The figure below shows the Monte Carlo Simulation of Stock Prices under Heston Process consisting of 500 paths for a stock with initial stock price, $S(0)$ = \$100, μ = 0.8, σ = 0.8, ρ = 0.3, κ = 4, θ = 1.5, T = 3 months and is traded only for Business Days.

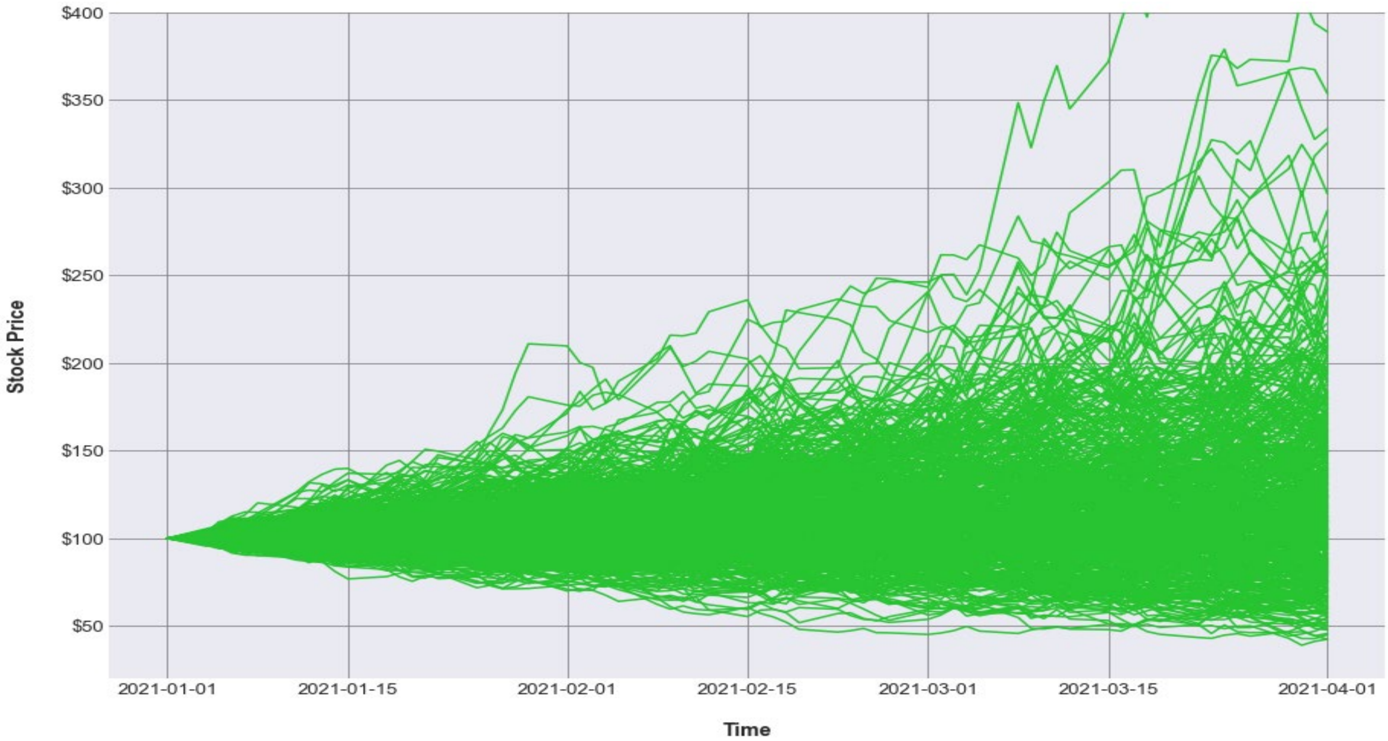


Figure 12: Monte-Carlo Simulation of Stock Prices under Heston Process

The following figure shows the Python Code for generating paths for Monte-Carlo Simulation. It considers the stock price at previous time step as initial stock price and applies Heston Process to predict the stock price at current time step.

```
def generate_paths(self, fixed_seed=True, day_count=365.):
    if self.time_grid is None:
        self.generate_time_grid()
    M = len(self.time_grid)
    I = self.paths
    paths = np.zeros((M, I))
    paths[0] = self.initial_value
    short_rate = self.short_rate
    X0 = np.log(self.initial_value)
    v0 = 0.04
    Y0 = np.log(v0)
    MU = np.array([0, 0])
    COV = np.matrix([[1, self.rho], [self.rho, 1]])
    W = ss.multivariate_normal.rvs( mean=MU, cov=COV, size=(I, M), random_state = 0)
    W_S = W[:,:,0]
    W_v = W[:,:,1]
    Y = np.zeros((I, M))
    Y[:,0] = Y0
    X = np.zeros((I,M))
    X[:,0] = X0
    v = np.zeros(M)

    for t in range(0, len(self.time_grid)-1):
        dt = (self.time_grid[t+1] - self.time_grid[t]).days / day_count
        v = np.exp(Y[:, t])
        v_sq = np.sqrt(v)
        Y[:,t+1] = Y[:,t] + (1/v)*(self.kappa*(self.theta - v) - 0.5*self.volatility**2 )*dt
                          + self.volatility * (1/v_sq) * np.sqrt(dt) * W_v[:,t]
        X[:,t+1] = X[:,t] + (self.mu - 0.5*v)*dt + v_sq * np.sqrt(dt) * W_S[:,t]

    paths = np.exp(X.T)
    self.instrument_values = paths
```

Figure 13: Code Cell for Generating Paths for Monte-Carlo Simulation under Heston Process

PARAMETER ANALYSIS

The figure below shows the movement of Stock Price simulated for different values of Rho. Rho is the covariance between Stock Price process and Stochastic Volatility process. As Rho deviates from zero, Stock Price increases with respect to time.

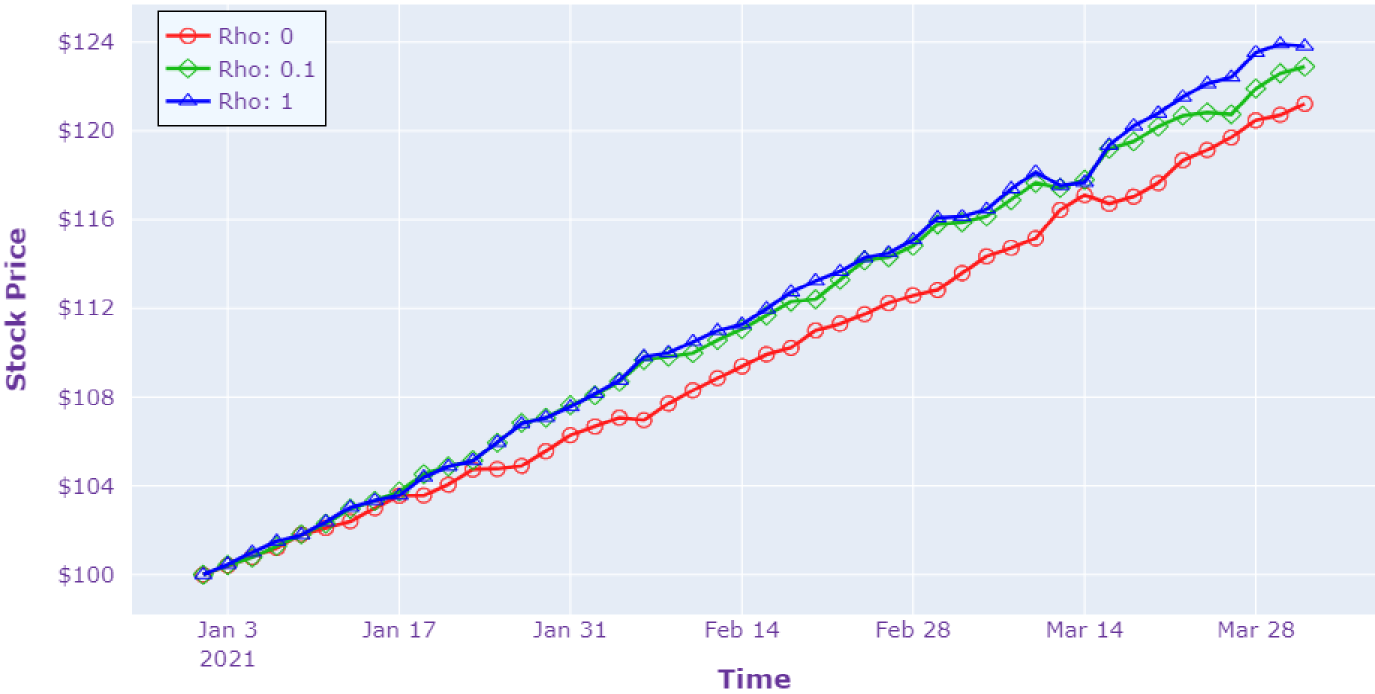


Figure 14: Simulation of Heston Process for Different Values of Rho (ρ)

The figure below shows the movement of Stock Price simulated for different values of Theta. Theta refers to the long term mean of the Volatility (variance) Process. As Theta increases, same results are obtained as on increasing Volatility in GBM model. Kappa refers to the mean reversion coefficient of the variance process and has similar impact.

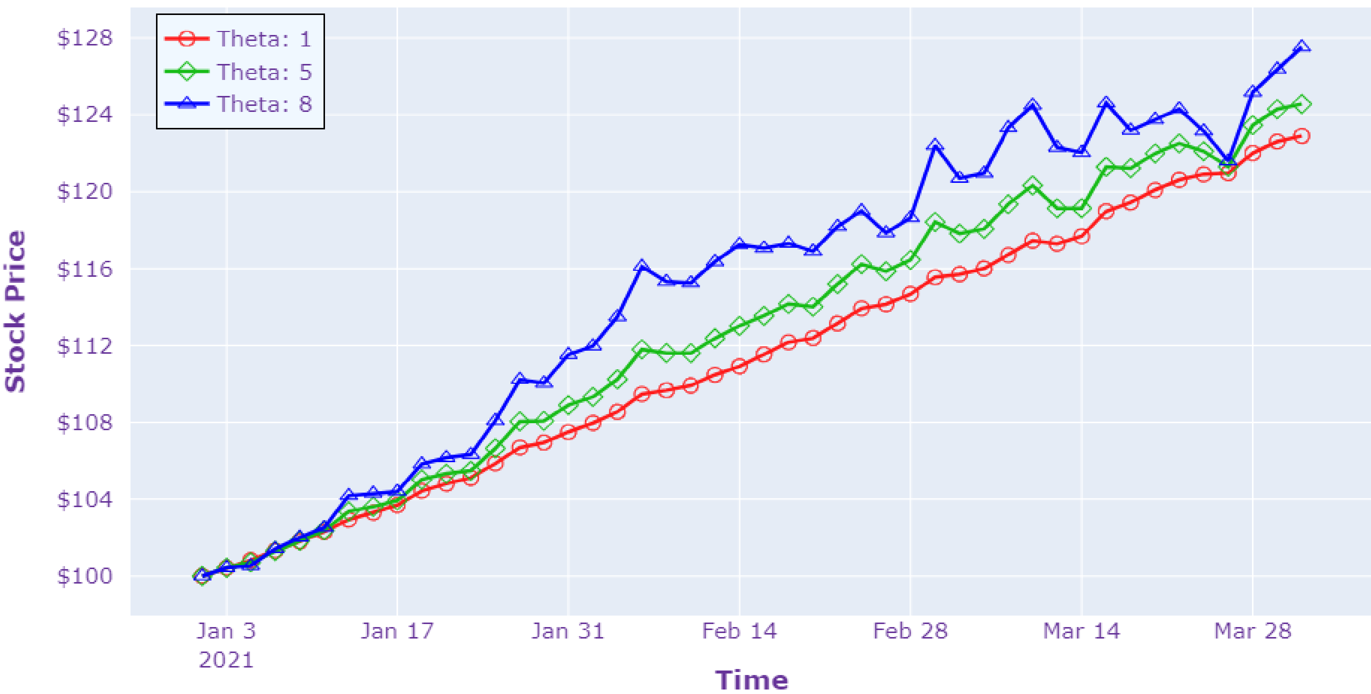


Figure 15: Simulation of Heston Process for Different Values of Theta (Θ)

CASE STUDY

- **APPLE STOCK PRICE PREDICTIONS**

The figure below shows the Comparison of Stock Price Predictions of three models for Apple Stock with initial stock price, $S(0)$ = \$266.94, μ = 0.8, σ = 0.8, T = 70 days and is traded only for Business Days.

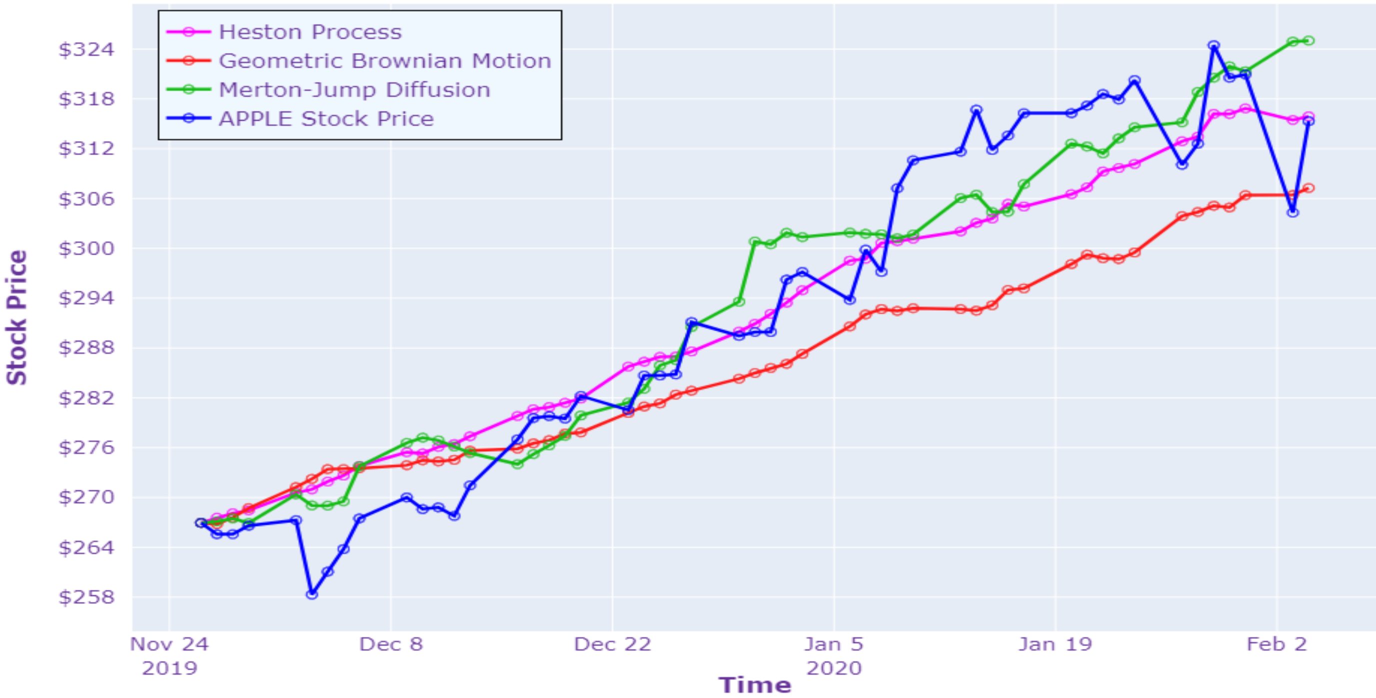


Figure 16: Comparison of Apple Stock Price Predictions with Actual Stock

The following figure shows the Apple Stock Price Predictions given by GBM. The GBM curve follows the Actual Price curve correctly for initial half but then fails to replicate the sudden jump in Stock Price in 2^{nd} week of January. This causes the GBM predictions to diverge from Actual Stock Price curve for later stages.

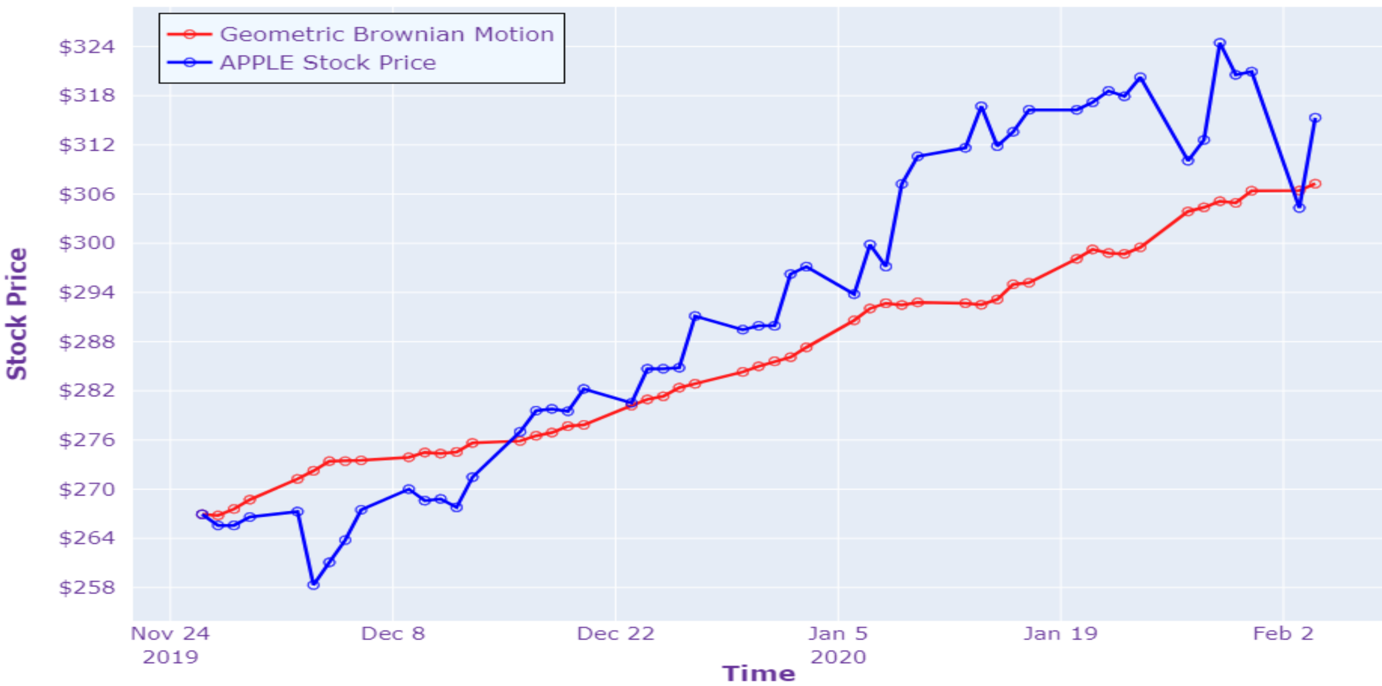


Figure 17: Apple Stock Price Predictions using GBM

The figure below shows the Apple Stock Price Predictions given by Merton-Jump Diffusion. Lambda, Gamma and Delta are taken as 1, 0.6, 0.6 resp. The predictions curve follows the Actual Price curve correctly for majority of the time duration and gives very good predictions on some days. This model correctly simulates the erratic and jump-prone nature of Actual Stock Price curve.

Figure 18: Apple Stock Price Predictions using Merton-Jump Diffusion

The following figure shows the Apple Stock Price Predictions given by Heston Process. Rho, Kappa and Theta are taken as 0.3, 4, 1.5 resp. The predictions curve follows the Actual Price curve correctly for initial half of the time duration but then again catches the Actual Stock Price curve towards the end to correctly predict final Stock Price.

Figure 19: Apple Stock Price Predictions using Heston Process

The figure below shows the Monte-Carlo Simulations of all three models for Apple Stock Price prediction overlapping each other. GBM spreads uniformly over the time duration as compared to Heston Process which spreads highly towards later stages. The Jumps in Merton-Jump Diffusion are clearly visible in the background.

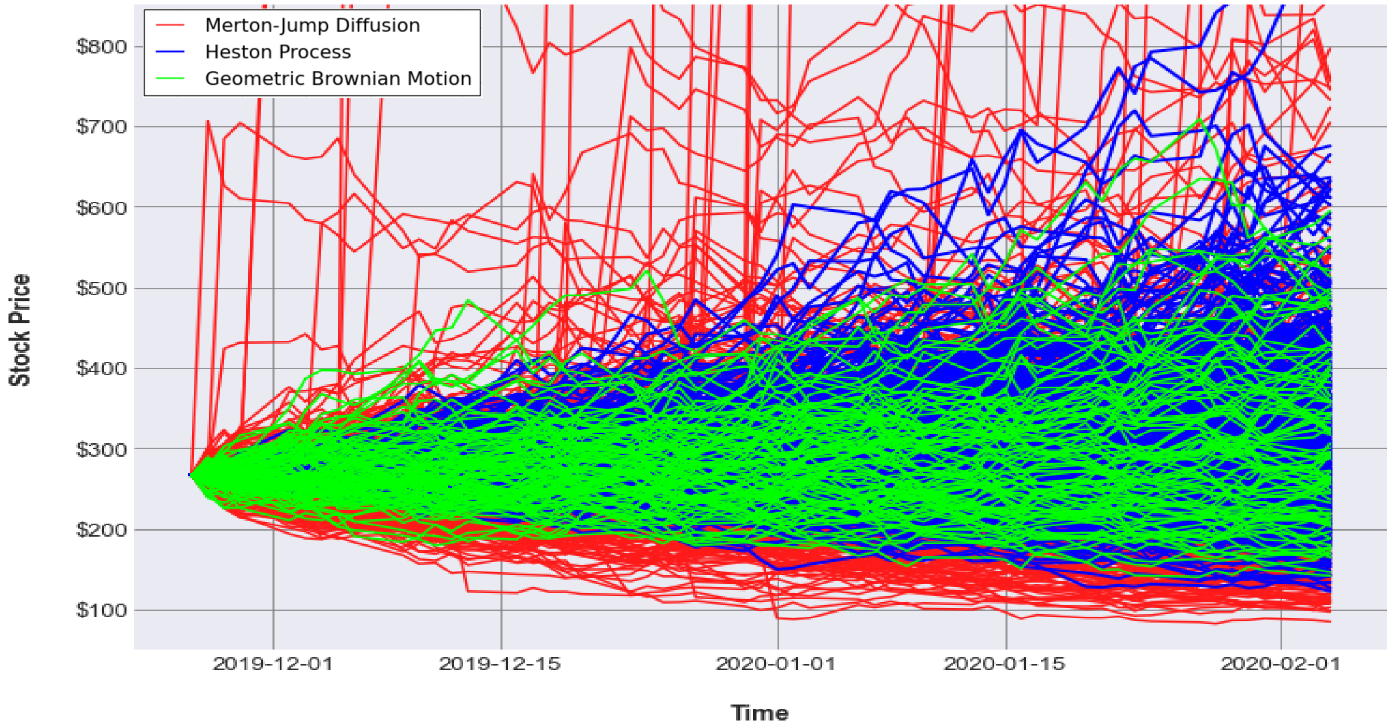


Figure 20: Monte-Carlo Simulations of Merton-Jump Diffusion, GBM and Heston

ERROR ANALYSIS FOR APPLE STOCK PRICE PREDICTIONS

The figure below shows the percentage error for all three models at each time step. Merton-Jump Diffusion and Heston follow very similar trajectories, whereas the GBM shows large error (5-8%) in the second half.

```
Root Mean Square Error:
Geometric Brownian Motion: 0.03701
Merton Jump Diffusion: 0.02231
Heston Process:  0.02195
```

Figure 19: RMSE For Each Method

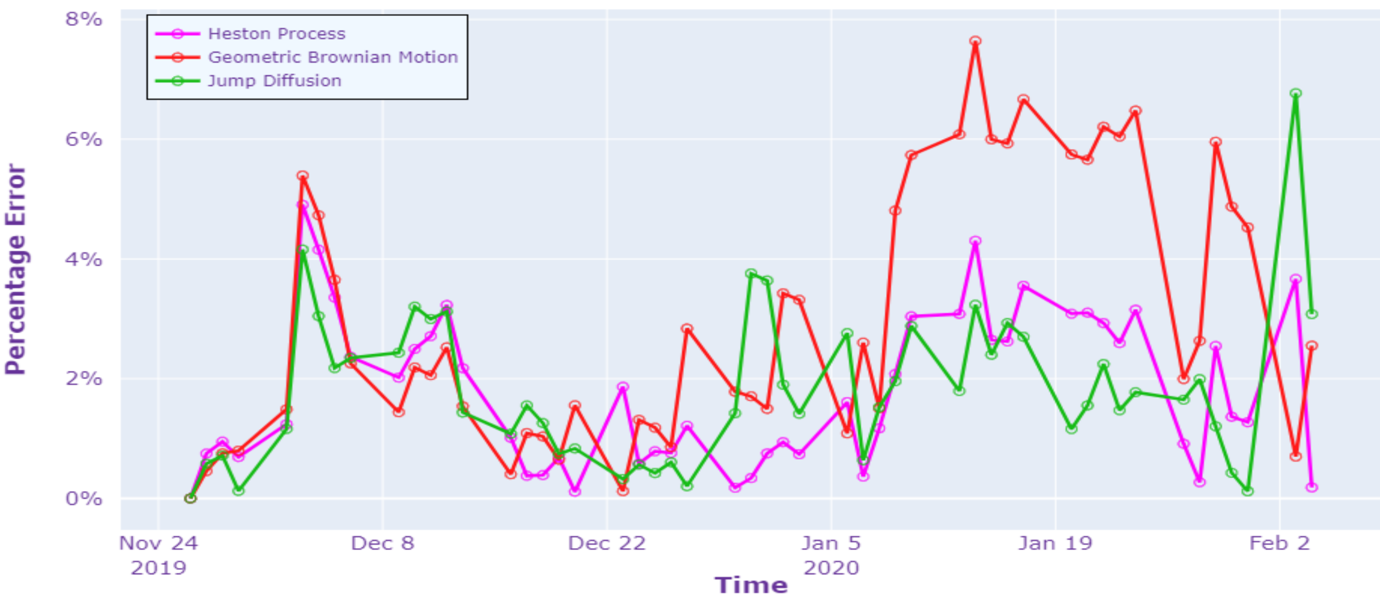


Figure 21: Comparison of Percentage Error for Apple Stock Price Predictions

CALL OPTION PRICING FOR APPLE STOCK

The following figure shows the Call Option Prices for various Strike Prices using Merton-Jump Diffusion for Stock Price predictions. The Call Option Prices vary from $30 to $55 for the Strike Prices in the range $295 to $270 at the end of 70 days.

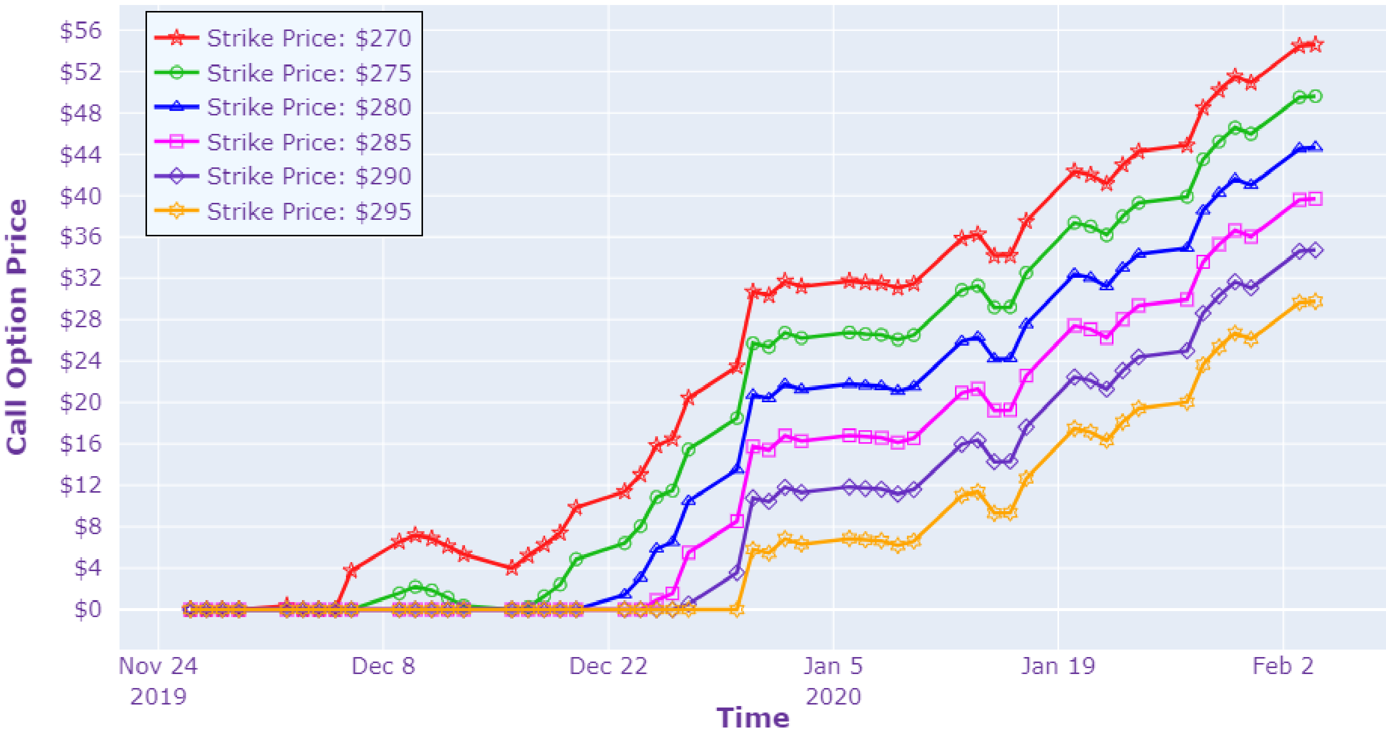


Figure 22: Call Option Pricing using Merton-Jump Diffusion for various Strike Prices

The figure below shows the Call Option Prices for various Strike Prices using Heston Process for Stock Price predictions. The Call Option Prices vary from $21 to $46 for the Strike Prices in the range $295 to $270 at the end of 70 days.

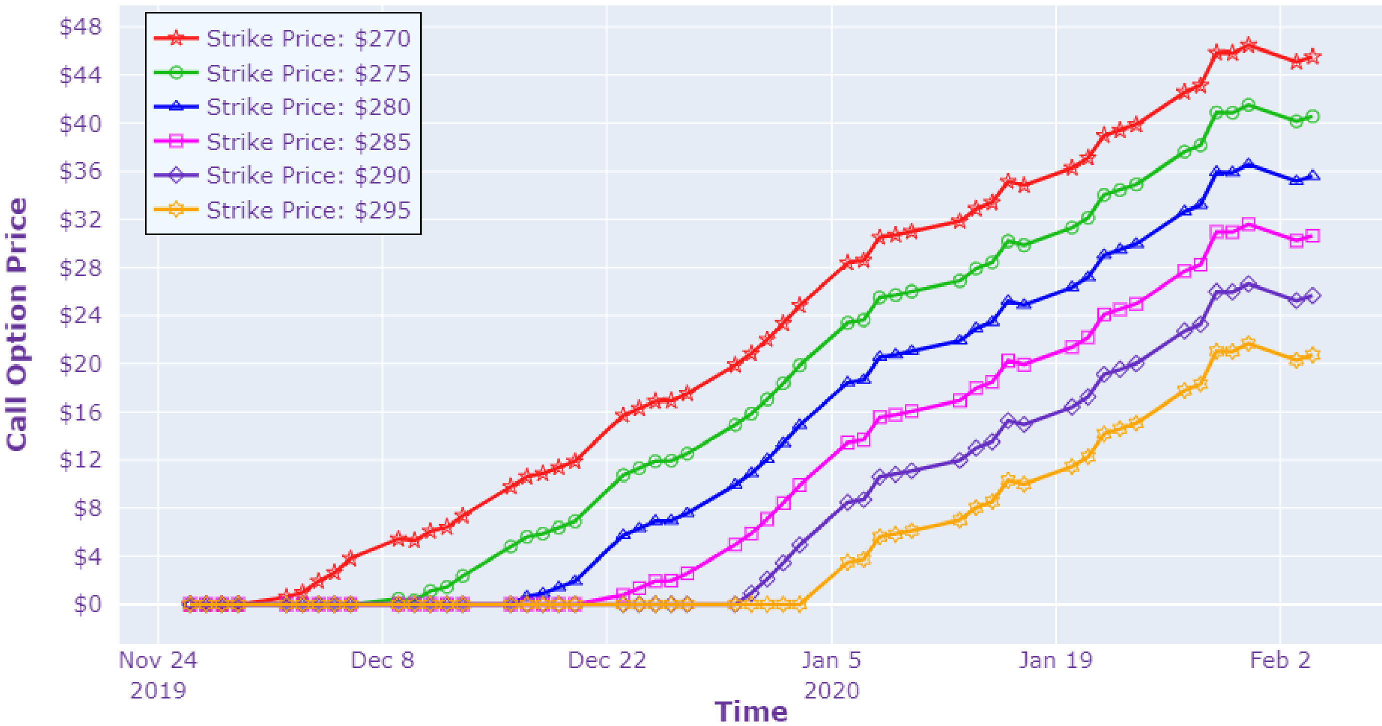


Figure 23: Call Option Pricing using Heston Process for various Strike Prices

- **INTEL STOCK PRICE PREDICTIONS**

The figure below shows the Comparison of Stock Price Predictions of three models for Apple Stock with initial stock price, $S(0)$ = \$53.12, μ = 0.8, σ = 1, T = 70 days and is traded only for Business Days.

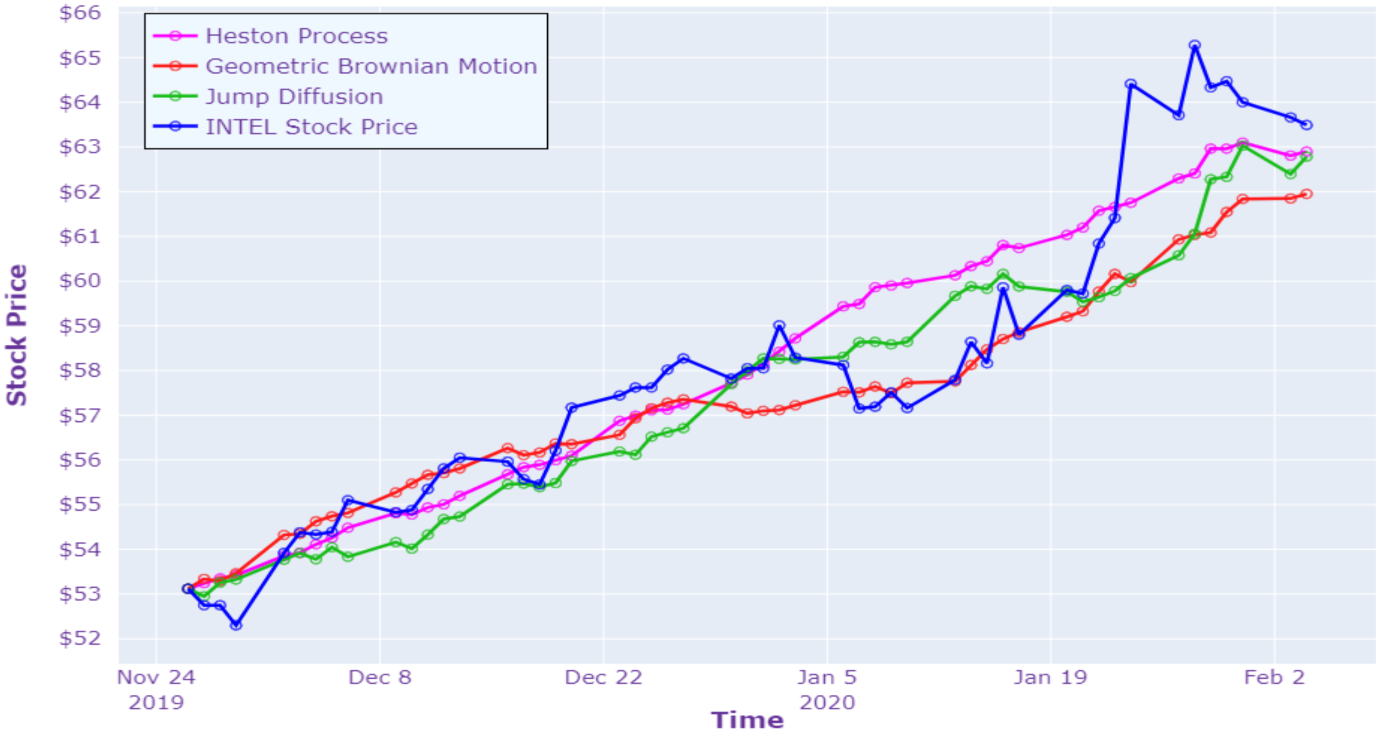


Figure 24: Comparison of Intel Stock Price Predictions with Actual Stock Price

The following figure shows the Intel Stock Price Predictions given by GBM. The GBM curve follows the Actual Price curve correctly till 22nd January but then again fails to replicate the sudden jump in Stock Price. This causes the GBM predictions to diverge from Actual Stock Price curve for later stages.

Figure 25: Intel Stock Price Predictions using GBM

The figure below shows the Intel Stock Price Predictions given by Merton-Jump Diffusion. Lambda, Gamma and Delta are taken as 1, 0.5, 0.5 resp. The predictions curve remains in touch with the Actual Price curve for majority of the time. This model correctly predicts the jump in later stages except for the jump size in Stock Price.

Figure 26: Intel Stock Price Predictions using Merton-Jump Diffusion

The following figure shows the Intel Stock Price Predictions given by Heston Process. Rho, Kappa and Theta are taken as 0.3, 4, 1.5 resp. The predictions curve follows the Actual Price curve correctly for initial half but then again catches the Actual Stock Price curve towards the end to almost correctly predict final Stock Price.

Figure 27: Intel Stock Price Predictions using Heston Process

The figure below shows the Monte-Carlo Simulations of all three models for Intel Stock Price prediction overlapping each other. GBM spreads uniformly over the time duration as compared to Heston Process which spreads highly towards later stages. The Jumps in Merton-Jump Diffusion are clearly visible in the background.

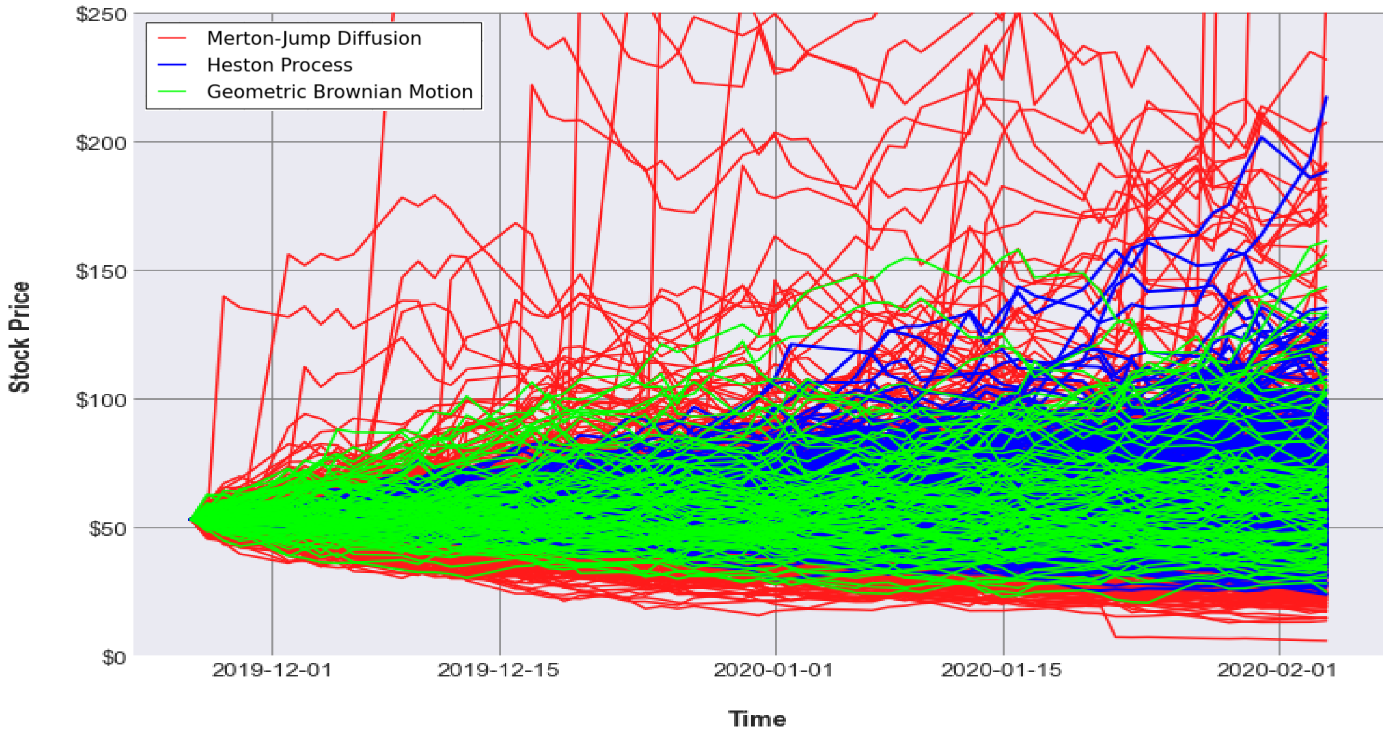


Figure 28: Monte-Carlo Simulations of Merton-Jump Diffusion, GBM and Heston

ERROR ANALYSIS FOR INTEL STOCK PRICE PREDICTIONS

The figure below shows the percentage error for all three models at each time step. Merton-Jump Diffusion and GBM show high error (6-7%) in last few weeks, whereas Heston Process shows relatively smaller error (4%).

```
Root Mean Square Error:
Geometric Brownian Motion: 0.02178
Merton Jump Diffusion: 0.02292
Heston Process:  0.02148
```

Figure 28: RMSE For Each Method

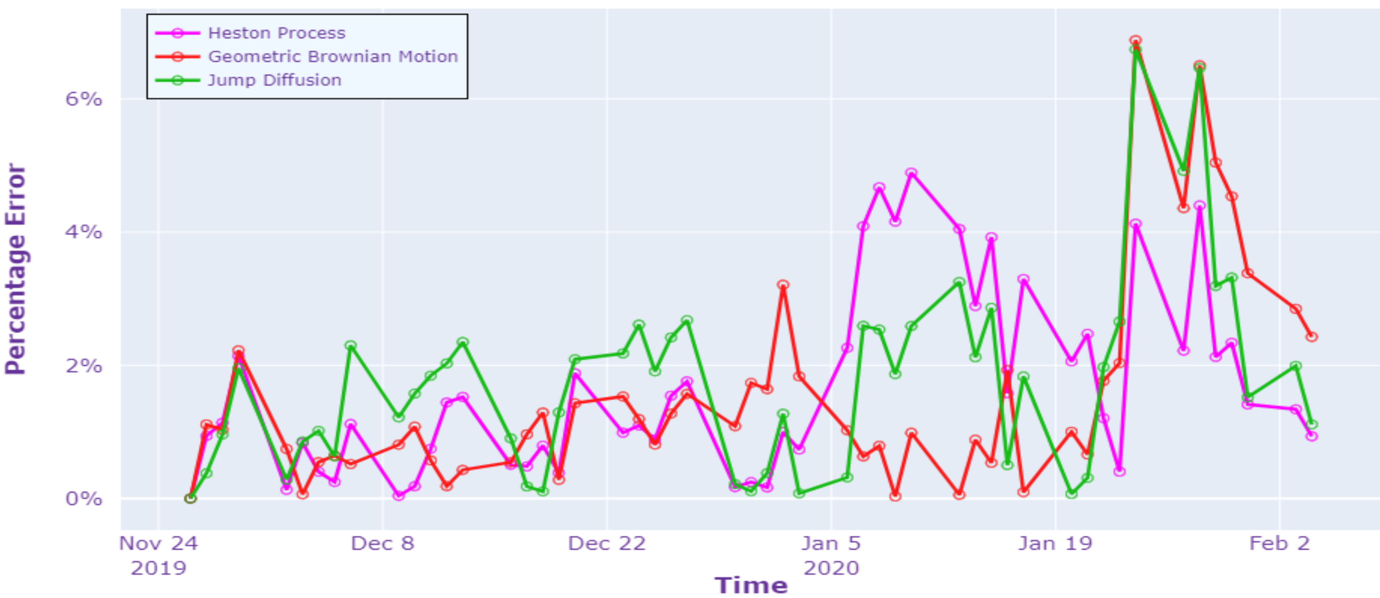


Figure 29: Comparison of Percentage Error for Intel Stock Price Predictions

CALL OPTION PRICING FOR INTEL STOCK

The following figure shows the Call Option Prices for various Strike Prices using Merton-Jump Diffusion for Stock Price predictions. The Call Option Prices vary from $0 to $9 for the Strike Prices in the range $64 to $54 at the end of 70 days.

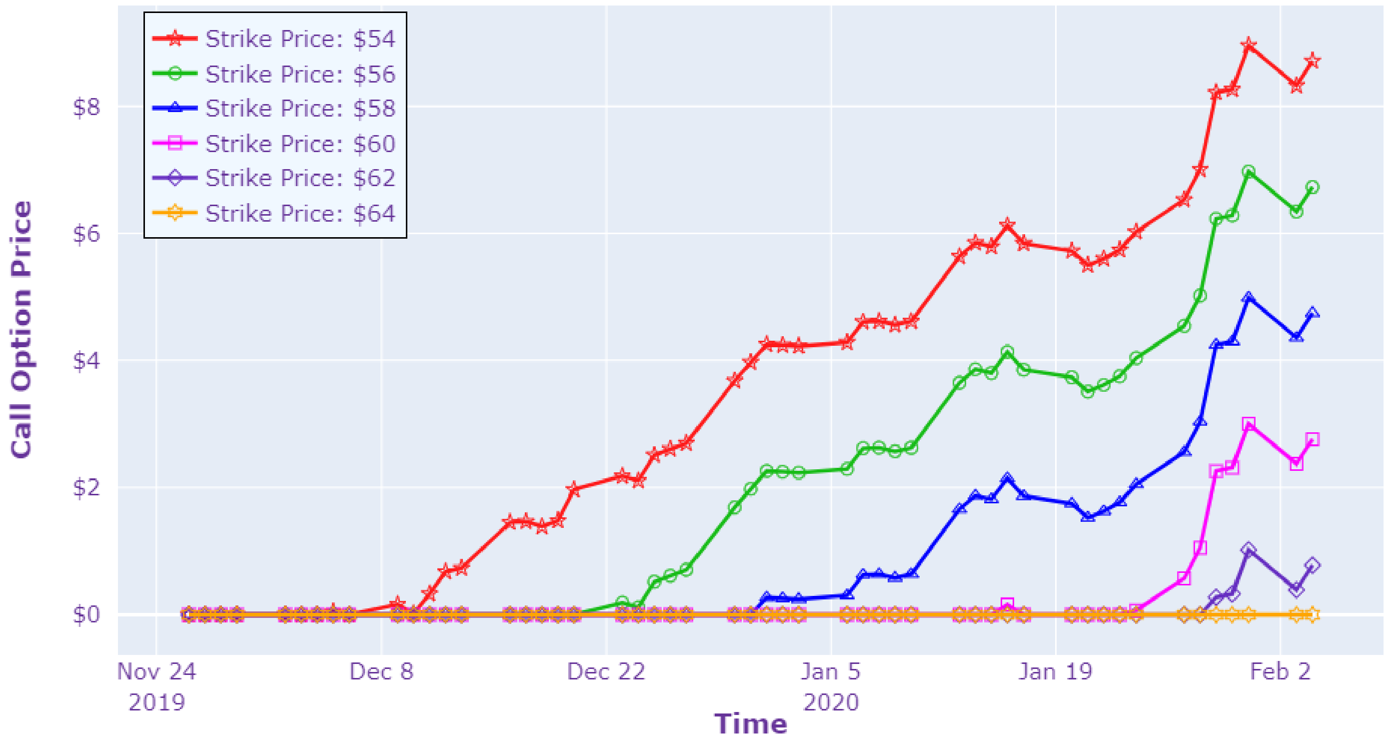


Figure 30: Call Option Pricing using Merton-Jump Diffusion for various Strike Prices

The figure below shows the Call Option Prices for various Strike Prices using Heston Process for Stock Price predictions. The Call Option Prices vary from $0 to $9 for the Strike Prices in the range $64 to $54 at the end of 70 days.

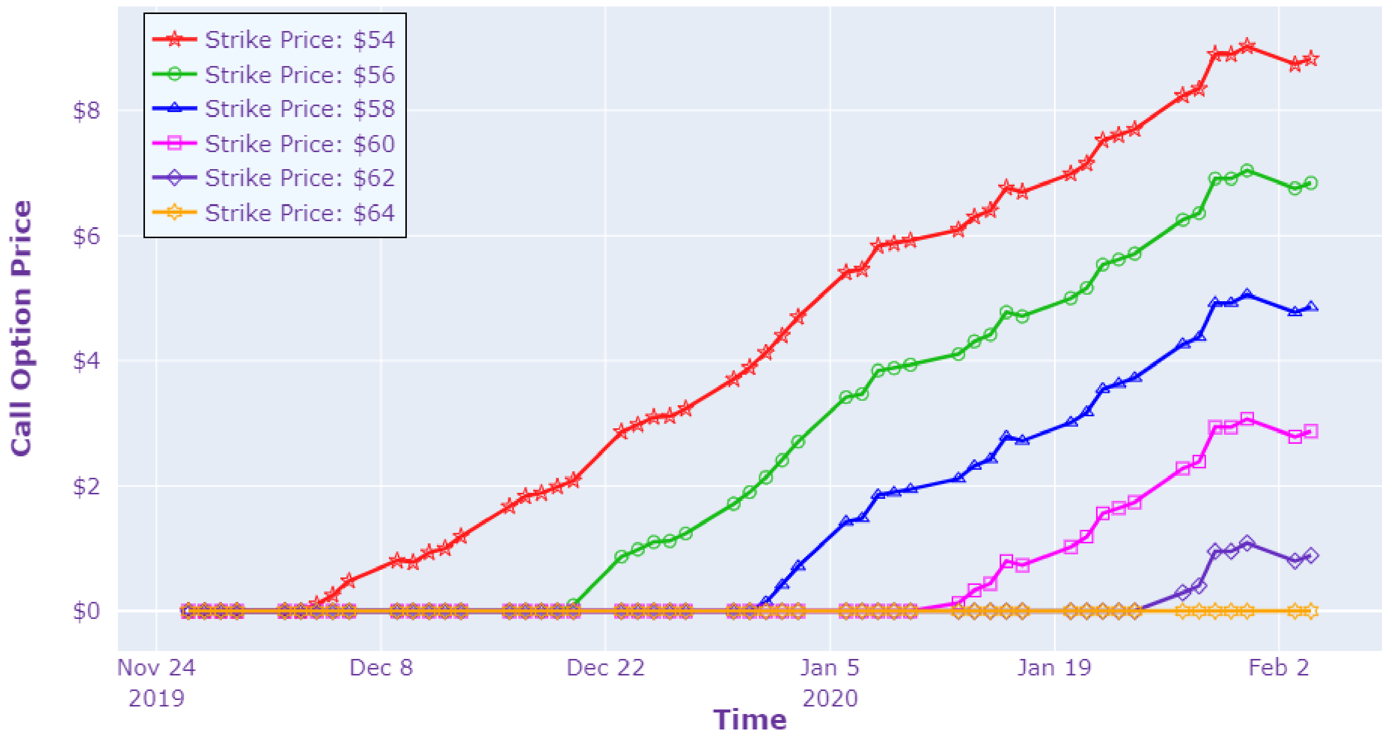


Figure 31: Call Option Pricing using Heston Process for various Strike Prices

APPLICATIONS OF STOCHASTIC DIFFERENTIAL EQUATIONS

Stochastic process models play a prominent role in a range of application areas, including **biology, chemistry, epidemiology, mechanics, microelectronics, economics,** and **finance**. Idea of modeling natural phenomena in terms of ordinary and stochastic differential equations can be exposed to a range of modern applications.

- **Physics:** In physics, SDEs have widest applicability ranging from **molecular dynamics** to **neurodynamics** and to the **dynamics of astrophysical objects**. More specifically, SDEs describe all dynamical systems, in which quantum effects are considered as perturbations. SDEs can be viewed as a generalization of the dynamical systems theory to models with noise. This is an important generalization because real systems cannot be completely isolated from their environments and for this reason always experience external stochastic influence.
- **Textile Industry:** In the cotton system, **fiber breakage** occurs in all mechanical processes undergone by the lint from the field to the spinning mill. The direct impact of breakage on fiber length distribution and on the incidence of short fibers represents a long-lasting concern in the cotton industry. A SDE model is developed for fibers undergoing breakage during textile processing steps. This model has provided more understanding of the fiber breakage phenomenon and the origination of different fiber length distributions.

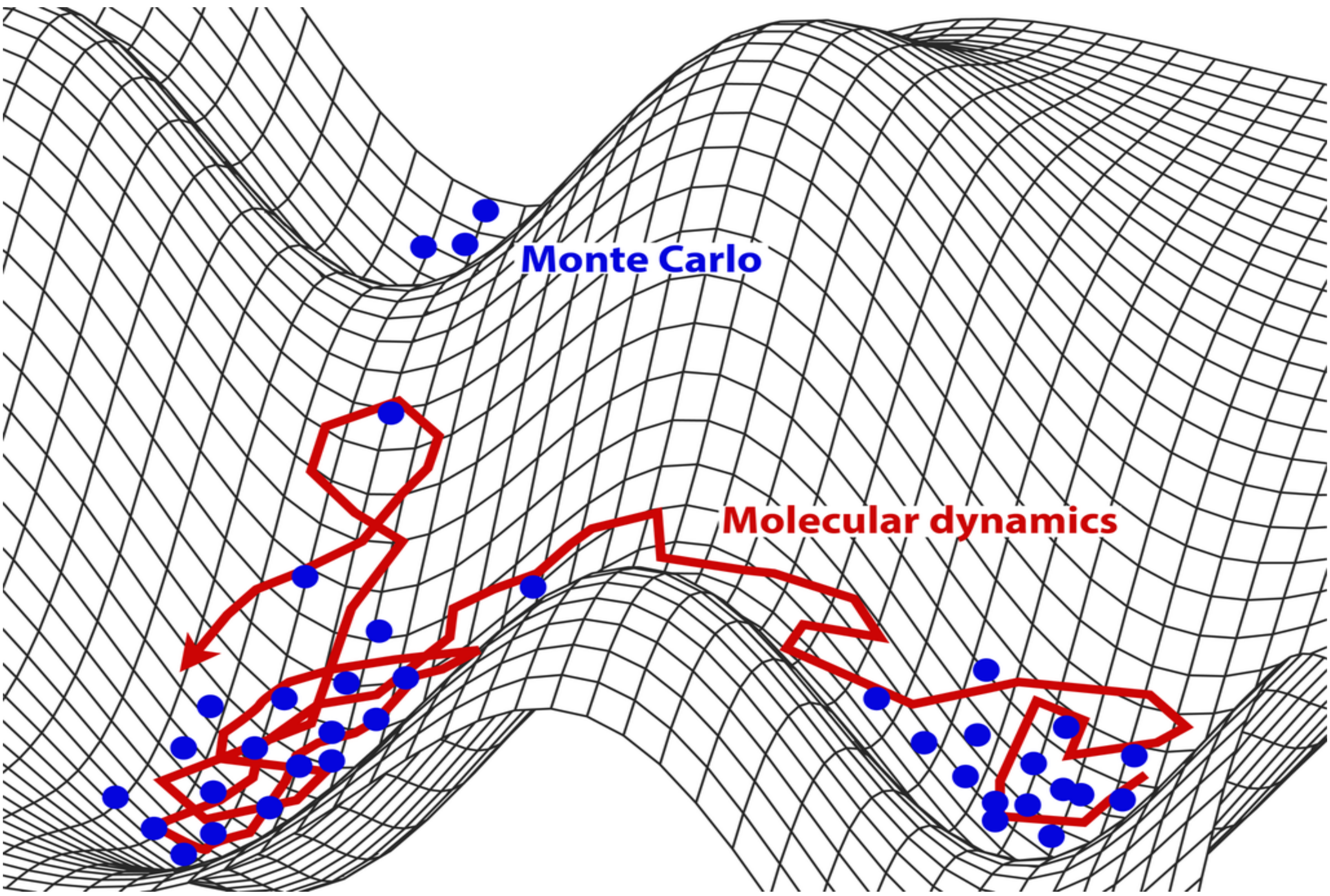
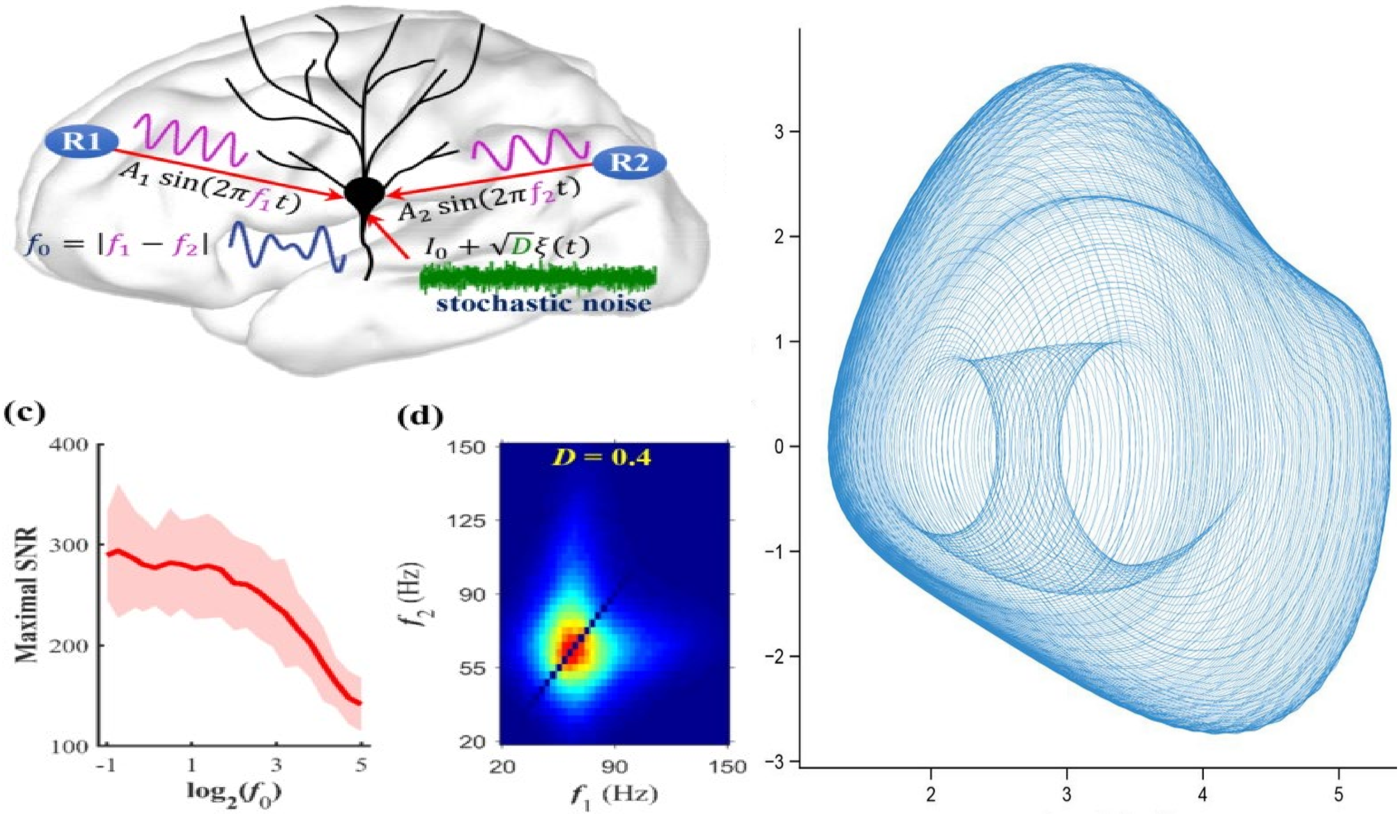


Figure 32: Monte-Carlo Simulation of Molecular Dynamics using Stochastic Differential Equations

- **Weather Forecasting:** Stochastic methods are a crucial area in contemporary **climate research** and are increasingly being used in **comprehensive weather** and **climate prediction models** as well as reduced order climate models. Stochastic approaches in numerical weather and climate prediction models also lead to the reduction of model biases.
- **Stochastic Resonance:** Stochastic resonance is an effect in which the response of a system to an external periodic signal is amplified by the presence of noise. Small periodic changes in the **Earth's orbital parameters** are amplified by the presence of random changes in the weather, thus leading to significant glacial cycles. **Neurons of crickets** is another example in which they try to detect an approaching predator, a bird, by detecting the periodic up- and downturns of air pressure caused by the bird's wing beats. Systems displaying stochastic resonance are often modeled using stochastic differential equations.
- **Equations with Delays:** A Stochastic Delay Differential Equation (SDDE) is a stochastic differential equation where the increment of the process depends on values of the process of the past. These equations can be used to model processes with a memory. An example is the influence of the ocean in a coupled atmospheric-ocean model of the climate. Since the oceans have a high heat capacity, but transport processes can take years, decades or even centuries, the oceans have a "long memory" compared to the atmosphere: The heat feedback from the oceans back to the atmosphere may depend on conditions that date back decades or even centuries.



a) b)

Figure 33: a) Stochastic Neural Dynamics b) Stochastic Delay Differential Equation Curve

CONCLUSION

In this project, we have simulated and analyzed some of the Stock Price Prediction Models namely:

- **Geometric Brownian Motion**
- **Merton-Jump Diffusion**
- **Heston Process**

Geometric Brownian Motion is a Standard Financial Model which can be used to find the future stock prices given the current stock price and some additional parameters. We performed parameter analysis to understand the impact of parameters such as Drift and Volatility on Stock Price Prediction. We also highlighted the short-comings of GBM and compared it to other models which overcame these short-comings.

Merton-Jump Diffusion Model is an extension of Geometric Brownian Motion. It considers two components namely: Diffusion Component and Jump Component. Diffusion Component refers to Standard GBM. Jump Component comprises of Compound Poisson Process which provides random jumps in Stock Prices as seen in real life Stock Price movement.

Heston Process is another extension of Geometric Brownian Motion. It considers Stochastic Volatility instead of Constant Volatility as in GBM. Stochastic Volatility captures the unpredictable nature of Stock Price Market in a better manner due to ever changing market conditions and situations.

Finally, we analyzed the performance all three models in two case studies namely: **Apple Stock Price Prediction** and **Intel Stock Price Prediction**. The different natures of three models can be clearly seen through the graphical analysis. **Error Analysis** is done in both case studies to compare the performance of different models. Based on the **Stock Price Predictions, Call Option Prices** for different Strike Prices and Maturities were also evaluated. All Models performed well in the case studies. **Thus, these models act as strong mathematical tools for Financial Markets.**

LEARNING

In this project, we studied different Financial Models for Stock Price Predictions. By understanding these models mathematically using Stochastic Differential Equations, we came to know how core mathematical concepts can be used to simulate real life processes. This project helped us to have a better **understanding of Financial Markets** and become familiar with different mathematical and financial tools available to predict the stock prices. During the course of this project, we used various Python3 libraries such as **NumPy, Matplotlib, Plotly, Scipy, Math** etc. We performed **Monte-Carlo Simulations** using Graphical Plotting Modules for different Financial models.

Generally, this project helped us to learn how to break a natural and complex phenomenon into simpler algorithms that can be understood by people from different domains. We also learnt how to write modular codes, which in turn makes debugging easier and makes the presentation more sophisticated.

REFERENCES

- https://www.csie.ntu.edu.tw/~lyuu/finance1/2015/20150513.pdf
- https://www.oreilly.com/library/view/python-for-finance/9781492024323/
- https://en.wikipedia.org/wiki/Stochastic_differential_equation
- https://en.wikipedia.org/wiki/Geometric_Brownian_motion
- https://corporatefinanceinstitute.com/resources/knowledge/trading-investing/heston-model/
- https://www.investopedia.com/articles/07/montecarlo.asp
- https://valuationmasterclass.com/introduction-to-derivatives-market/
- https://www.investopedia.com/terms/d/derivative.asp
- https://www.investopedia.com/articles/basics/04/100804.asp

LINKS FOR DATASET

- **Apple Stock Price Dataset**

  https://www.kaggle.com/tarunpaparaju/apple-aapl-historical-stock-data

- **Intel Stock Price Dataset**

  https://finance.yahoo.com/quote/intc/history/