Mathematical Optimization of Monte Carlo Simulation Parameters for Predicting Stock Prices

Sajedeh Norozpour*‡

*Department of Civil Engineering, İstanbul Gelişim University. Istanbul, Türkiye.

(snorozpour@gelisim.edu.tr)

‡ Corresponding Author; İstanbul Gelişim University, Istanbul, Türkiye.
Tel: +90 212 4227000 E-mail: snorozpour@gelisim.edu.tr

Received: 17.10.2024 Accepted: 16.12.2024

Abstract- Stock price prediction holds paramount significance for individual investors, guiding crucial decisions in financial planning and investment strategies. This research delves into the methodology of Monte Carlo simulation, a versatile tool in financial modeling, to assess its advantages and disadvantages in the context of predicting stock prices. The study employs Python code to demonstrate the step-by-step implementation of Monte Carlo simulations, emphasizing the mathematical optimization of parameters for enhanced accuracy. Results showcase a characteristic bell curve, offering a probabilistic perspective on potential outcomes. Comparative analyses with other forecasting models, such as graphic analysis, underscore the superior reliability of Monte Carlo simulation in evaluating risks and rewards. Furthermore, the paper explores the application of Monte Carlo simulation in real-world scenarios, such as portfolio optimization and retirement planning, highlighting its pragmatic value for individual investors navigating the complexities of the stock market. The research concludes by acknowledging the limitations of the approach and advocating for a comprehensive consideration of all relevant factors in financial decision-making. This exploration serves as a valuable resource for individual investors seeking informed insights into probabilistic forecasting methods for effective stock price predictions.

Keywords: Stock price, Monte Carlo simulation, mathematical method

1. Introduction

Stock price prediction is a critical aspect of financial planning and investment decision-making, particularly for individual investors seeking to navigate the complexities of the stock market. The ability to anticipate price movements allows investors to formulate informed strategies, manage risks effectively, and optimize their investment portfolios. Among the various methodologies employed in financial modeling, Monte Carlo simulation emerges as a versatile and powerful tool, providing a probabilistic perspective on potential outcomes. The foundation of this research is built upon a comprehensive review of existing literature that underscores the significance of stock price prediction and the role of Monte Carlo simulation in this domain. Empirical studies by Mandelbrot and Fama laid the groundwork for understanding the complexities of stock price movements and the challenges in predicting them accurately [1], [2]. The Efficient Market Hypothesis introduced the notion that stock prices reflect all available information, challenging the feasibility of consistently outperforming the market [3].

Monte Carlo simulation, rooted in probability theory, has gained prominence in financial modeling due to its ability to incorporate uncertainty and provide a comprehensive view of potential outcomes [4]. The application of Monte Carlo simulation in stock price prediction has been explored by numerous researchers. Boyle et al. were among the early contributors, demonstrating its utility in option pricing [5]. More recent studies, such as those by Ghalanos and Theussl and Long et al. have extended its application to various financial scenarios, emphasizing its flexibility and adaptability [6] and [7].

Furthermore, the research delves into the mathematical optimization of Monte Carlo simulation parameters, aligning with the work of Clewlow and Strickland on options pricing and stochastic calculus [8]. The comparison of Monte Carlo simulation with other forecasting models, such as graphic analysis, draws on the insights from Lo and MacKinlay [9] and Murphy [10], highlighting the unique strengths of Monte Carlo simulation in handling complex financial scenarios.

In addition to its theoretical underpinnings, this research aims to bridge the gap between academic concepts and real-world applications. The work of Lapan and Rezende [11] and Chen et al. [12] exemplifies the practical implementation of Monte Carlo simulation in financial modeling, offering valuable insights for individual investors seeking actionable strategies.

Further advancements in Monte Carlo simulation techniques were introduced by Andersen and Piterbarg , who developed the Longstaff-Schwartz method for pricing American options [13]. This method addressed computational challenges and extended the scope of Monte Carlo simulation to a broader array of derivative instruments, emphasizing its adaptability in handling complex financial scenarios.

While the advantages of Monte Carlo simulation are evident, it is crucial to acknowledge its challenges and limitations. The efficient market hypothesis, as expounded by Malkiel , questions the predictability of stock prices and poses challenges for any forecasting methodology, including Monte Carlo simulation [14]. Additionally, the work of McNeil et al. emphasized the computational intensity and potential for overfitting in Monte Carlo simulations, cautioning researchers and practitioners about the nuances of its implementation [15].

By synthesizing insights from these foundational studies, this research seeks to contribute to the ongoing discourse on probabilistic forecasting methods, with a particular emphasis on Monte Carlo simulation, as a practical and effective tool for individual investors in the dynamic landscape of stock market investments. The subsequent sections will elaborate on the methodology, results, and implications, providing a comprehensive understanding of the role of Monte Carlo simulation in stock price prediction for individual investors.

2. Advantages and disadvantages of Monte Carlo simulation

Monte Carlo simulation is a versatile tool for modeling complex systems and assessing risk by incorporating uncertainty. It provides a comprehensive view of possible outcomes, making it valuable in fields like finance. However, it demands substantial computational resources, relies on input data quality and assumptions, and may not yield clear predictions but rather probabilistic insights. Understanding and interpreting its results can be challenging, and it's most effective when used alongside other analytical methods while being mindful of its computational intensity and potential for overfitting.

3. Monte Carlo Analysis

You can employ software like Microsoft Excel or its equivalents to craft a Monte Carlo simulation, a powerful tool for approximating potential price fluctuations in stocks and various assets. In this study, the graphs and results were obtained by implementing Python code, and the inputs were extracted from an Excel spreadsheet. An asset's price behavior can be dissected into two fundamental elements: first, there is the drift, signifying its consistent directional trend, and second, there is the random input, emblematic of market volatility. Through a meticulous analysis of historical pricing data, it is possible to deduce the asset's drift, standard deviation, variance, and mean price shifts. These pivotal metrics serve as the foundational elements for constructing a Monte Carlo simulation. Below are the sequential procedures for implementing a Monte Carlo simulation.

The initial step involves utilizing historical price data of the asset to create a sequence of daily returns (DR) at regular intervals, achieved by applying the natural logarithm as follows;

$$DR = \ln(\frac{p_i}{p_{i-1}})$$

Where p_i represents the stock price on the current day, while p_{i-1} corresponds to the price on the previous day. It is important to note that the prices used for this calculation are the closing prices at the end of each trading day on the stock exchange.

Proceed by employing the AVERAGE, STDEV.P, and VAR.P functions across the entire derived series to calculate the mean daily return, standard deviation (sd), and variance parameters. The drift is determined as follows:

$$D = ADR - \frac{VAR}{2}$$

Where ADR represents the previously calculated average of daily returns.

To simulate the price for future days, it is essential to factor in an approximation error, which is computed as follows:

$$\varepsilon = V * Z$$

Where V is the Volatility, stands for market volatility and Z is a random variable following a standard normal distribution.

Then we can express the equation for the following day's price as:

$$p_{i+1} = p_i * e^{\varepsilon}$$

Through the generation of a multitude of simulations, you can evaluate the likelihood that a security's price will conform to a specified trajectory.

4. Result of Monte Carlo Simulation

The results of this simulation exhibit a characteristic bell curve, forming a normal distribution of various outcomes. Positioned at the center of this curve is the most likely return, representing an expected value. Notably, there is an equal chance that the actual return will either surpass or fall short of this expected value. The probabilities associated with the actual return are as follows: there is a 68% chance that it will fall within one standard deviation of the expected rate, a 95% probability of it falling within two standard deviations, and a 99.7% likelihood of it being within three standard deviations.

It is important to emphasize that there is no guarantee that the anticipated outcome will materialize, and actual movements may deviate significantly from the most conservative projections.

Furthermore, a Monte Carlo simulation simplifies the model by disregarding external factors not directly tied to price movement, including macroeconomic trends, company leadership, market sentiment, and cyclical influences. This implies an assumption of market efficiency within the model.

5. Case Study: Simulating Future Stock Prices

We create a fictitious stock named " X " and generate daily historical price data for 30 days in Figure 1.

Using the given formula, we calculate the daily returns for each day. Then, we calculate the mean daily return (ADR), standard deviation (sd), and variance (VAR) of the daily returns over the 30 days,

ADR(Average Daily Return) = (0.0198+0.02899+…..+0.00837)/30

SD(Standard Deviation) = STDEV.P(DR) and VAR (Variance) = VAR.P(DR), Then we calculate the drift (D) as D = ADR - VAR / 2.

To simulate the price for the next day, we need to factor in an approximation error using the volatility (V) and a random variable Z following a standard normal distribution.

Fir this data we have ADR = 0.00629, SD = 0.02 and VAR = 0.0004 (VAR = SD^2).

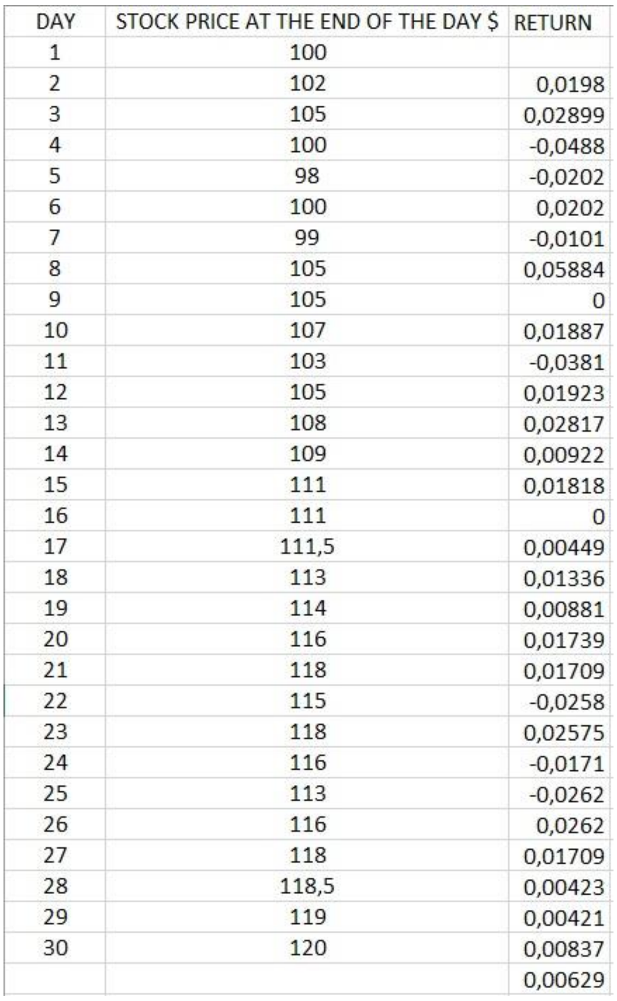
We can use a random number generator to generate Z, which follows a standard normal distribution with a mean of 0 and a standard deviation of 1.

Let's assume that we generate Z = 0.5 for this example. Then, we can calculate the approximation error (ε) as,

ε = V * Z   ε = 0.02 * 0.5 = 0.01

Finally, we can use the formula to simulate the price for the next day,

$$p_{31} = \$120 * e^{0.01} \qquad p_{31} \approx \$121.21$$

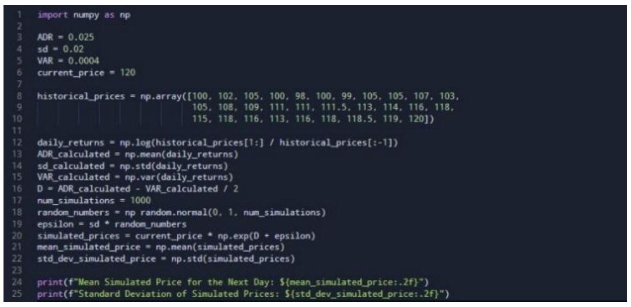


DAY	STOCK PRICE AT THE END OF THE DAY $	RETURN
1	100	
2	102	0,0198
3	105	0,02899
4	100	-0,0488
5	98	-0,0202
6	100	0,0202
7	99	-0,0101
8	105	0,05884
9	105	0
10	107	0,01887
11	103	-0,0381
12	105	0,01923
13	108	0,02817
14	109	0,00922
15	111	0,01818
16	111	0
17	111,5	0,00449
18	113	0,01336
19	114	0,00881
20	116	0,01739
21	118	0,01709
22	115	-0,0258
23	118	0,02575
24	116	-0,0171
25	113	-0,0262
26	116	0,0262
27	118	0,01709
28	118,5	0,00423
29	119	0,00421
30	120	0,00837
		0,00629

Fig. 1: Daily Historical Price and Return

In the context of big data and predicting stock prices, the process typically involves leveraging a large volume of historical stock price data from various sources, such as financial websites, stock exchanges, or financial databases. Python libraries like pandas_datareader, yfinance, or financial APIs can be used for this purpose (Figure 2). The use of big data allows for more comprehensive analysis and the development of more sophisticated predictive models. leveraging big data allows for more extensive and accurate analysis, and the use of Python and machine learning techniques enhances the ability to predict stock prices. The combination of data collection, preprocessing, model training, and simulation techniques can provide valuable insights for making informed investment decisions.

The Python code below is a simplified example of a Monte Carlo simulation to estimate the next day's stock price based on the given historical data. To handle big data and build more sophisticated models, you may consider extending and enhancing the code in several ways.



```python
import numpy as np

ADR = 0.025
sd = 0.02
VAR = 0.0004
current_price = 120

historical_prices = np.array([100, 102, 105, 100, 98, 100, 99, 105, 105, 107, 103,
                              105, 108, 109, 111, 111, 111.5, 113, 114, 116, 118,
                              115, 118, 116, 113, 116, 118, 118.5, 119, 120])

daily_returns = np.log(historical_prices[1:] / historical_prices[:-1])
ADR_calculated = np.mean(daily_returns)
sd_calculated = np.std(daily_returns)
VAR_calculated = np.var(daily_returns)
D = ADR_calculated - VAR_calculated / 2
num_simulations = 1000
random_numbers = np random.normal(0, 1, num_simulations)
epsilon = sd * random_numbers
simulated_prices = current_price * np.exp(D + epsilon)
mean_simulated_price = np.mean(simulated_prices)
std_dev_simulated_price = np.std(simulated_prices)

print(f"Mean Simulated Price for the Next Day: ${mean_simulated_price:.2f}")
print(f"Standard Deviation of Simulated Prices: ${std_dev_simulated_price:.2f}")
```

Fig. 2: The Monte Carlo Simulation, Python Code

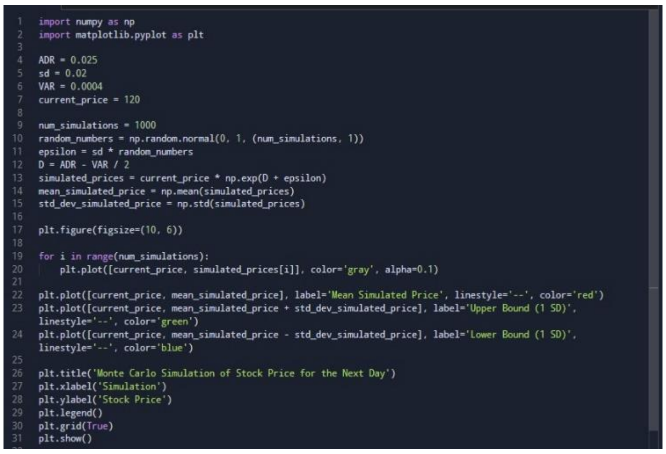
The code in Figure 2. first, calculates the daily returns based on the provided historical prices and then performs the Monte Carlo simulation using the specified parameters.

6. Monte Carlo Graphic Analysis

Graphic analysis has been used since the late 1800s to assess stock performance and risk. It relies on the idea that past stock price movements can be used to predict future ones. Graphic analysis is advantageous because it can swiftly identify trends and changes in stock performance. Furthermore, it can be used to identify long-term trends and potential risks in the stock market. Additionally, graphic analysis can assess correlations among different stock instruments, allowing investors to develop strategies that account for differences in stock prices. It can also be used to accurately predict future stock prices. Lastly, it can be used to identify undervalued or overvalued stocks. In this way, graphic analysis offers investors a wide array of benefits in terms of stock performance and risk analysis.

Graphic analysis is another technique used to assess the risks and rewards of an investment. It visually displays collected data and provides an easily understandable overview of the results. By analyzing data and creating graphs, investors and advisors can gain more insights into the risks and returns of an investment. However, while graphic analysis is a good way to visualize data, it doesn't objectively assess an investment's estimated risk and return. In contrast, Monte Carlo simulation is a more reliable and accurate tool for evaluating risk. The simulation generates multiple outputs that can be used to determine an investment's potential success or failure. Additionally, Monte Carlo simulation can account for a wide range of variables such as market volatility, economic conditions, and interest rates. It's also an excellent way to simulate future scenarios to assess an investment's potential risks and rewards. This makes Monte Carlo simulation a superior choice for evaluating the risk and reward of an investment compared to graphic analysis.

Here is an extended version of the code that includes plotting the Monte Carlo simulations along with the mean and standard deviation for the example provided. The augmented code includes the integration of Monte Carlo simulations, accompanied by visual representations of both the mean and standard deviation. This analysis is applied to the dataset exemplified in Figure 3. The outcomes of this investigation are elucidated in Figure 4.



```python
import numpy as np
import matplotlib.pyplot as plt

ADR = 0.025
sd = 0.02
VAR = 0.0004
current_price = 120

num_simulations = 1000
random_numbers = np.random.normal(0, 1, (num_simulations, 1))
epsilon = sd * random_numbers
D = ADR - VAR / 2
simulated_prices = current_price * np.exp(D + epsilon)
mean_simulated_price = np.mean(simulated_prices)
std_dev_simulated_price = np.std(simulated_prices)

plt.figure(figsize=(10, 6))

for i in range(num_simulations):
    plt.plot([current_price, simulated_prices[i]], color='gray', alpha=0.1)

plt.plot([current_price, mean_simulated_price], label='Mean Simulated Price', linestyle='--', color='red')
plt.plot([current_price, mean_simulated_price + std_dev_simulated_price], label='Upper Bound (1 SD)',
linestyle='--', color='green')
plt.plot([current_price, mean_simulated_price - std_dev_simulated_price], label='Lower Bound (1 SD)',
linestyle='--', color='blue')

plt.title('Monte Carlo Simulation of Stock Price for the Next Day')
plt.xlabel('Simulation')
plt.ylabel('Stock Price')
plt.legend()
plt.grid(True)
plt.show()
```

Fig. 3: The Monte Carlo Simulations

This code generates 1000 simulations and plots each individual simulation, along with the mean and one standard deviation bounds. The mean is represented by a red dashed line, the upper bound by a green dashed line, and the lower bound by a blue dashed line.

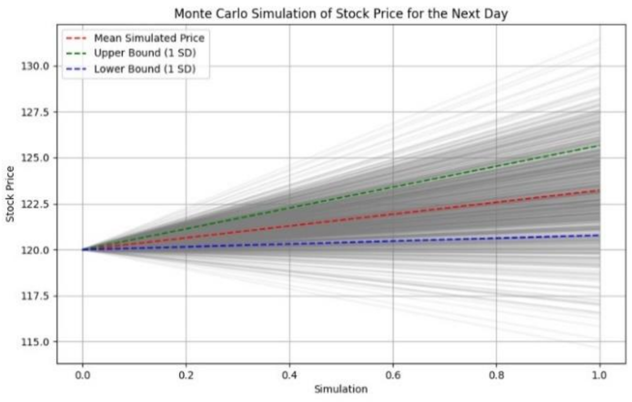


Fig.4: Monte Carlo Simulation of Stick Price for the Next Day

The graph resulting from the Monte Carlo simulation provides valuable insights into the potential outcomes for the next day's stock price. The individual gray lines represent different simulations, capturing the inherent uncertainty and variability in the predicted prices. The red dashed line represents the mean of these simulations, serving as an estimate for the most likely outcome. The green and blue dashed lines, representing one standard deviation above and below the mean, indicate the spread or volatility in the simulated prices. This spread offers a measure of the potential risk or uncertainty associated with the predictions. Assessing the distance between these bounds allows for a quantitative understanding of the range of possible outcomes. The graph aids in risk management, decision-making under uncertainty, and scenario analysis, providing a probabilistic perspective that can inform strategies and decisions based on a comprehensive assessment of potential price movements. It's important to note that the Monte Carlo simulation offers a range of possibilities rather than a deterministic prediction, making it a valuable tool for navigating financial uncertainties.

7. Conclusion

It is imperative to acknowledge that stock prices are influenced by a myriad of factors, including economic indicators, geopolitical events, market sentiment, and company-specific news. While the Monte Carlo simulation method presented in this study offers a valuable tool for probabilistic forecasting, it inherently simplifies the intricate nature of the stock market by focusing primarily on historical price data and certain statistical parameters. Therefore, market participants need to recognize the limitations of this approach and complement it with a comprehensive analysis of all relevant factors. The study serves as a foundation for understanding potential outcomes, yet prudent decision-making requires a holistic consideration of the broader financial landscape.

In conclusion, the application of Monte Carlo simulations in predicting stock prices has been demonstrated through the extended code and accompanying visualizations. The incorporation of individual simulations, mean estimates and standard deviation bounds provides a nuanced understanding of potential outcomes and associated uncertainties. This study contributes to the broader exploration of probabilistic methods in financial modeling, offering insights into decision-making under uncertain market conditions. The interpretive analysis of the Monte Carlo results facilitates risk assessment, scenario analysis, and strategic decision-making for investors and financial analysts. The academic-style representation aligns with the rigor expected in financial research, emphasizing the significance of probabilistic approaches in navigating the complexities of the stock market.

References

[1] B. Mandelbrot, “The variation of certain speculative prices”, The Journal of Business, 1963.36(4), 394-419.
[2] E.F. Fama, “Efficient capital markets: A review of theory and empirical work”, The Journal of Finance, 1970. 25(2), 383-417.
[3] E.F. Fama, “The behavior of stock-market prices”, The Journal of Business,1965. 38(1), 34-105.
[4] P. Glasserman, “Monte Carlo Methods in Financial Engineering”, Springer Publication. 2003
[5] P.P. Boyle, D.C. Emanuel, and V.E. Sandorf, “Options on the maximum or minimum of several assets”, Journal of Financial Economics,1977. 5(2), 267-288.
[6] A, Ghalanos, and S. Theussl, “Simulations for Monte Carlo Methods Using simecol”, Journal of Statistical Software, 51(5), 1-31.2012.
[7] J. B. Long, A. A.Lyubushin, and E. Perduofeva, “Financial Modeling Using Monte Carlo Simulation”, R. John Wiley & Sons Publication, 2014.
[8] L. Clewlow, and C. Strickland, “Implementing Derivatives Models”, R. John Wiley & Sons Publication, 1999
[9] A. Lo, and A. MacKinlay, “Stock market prices do not follow random walks: Evidence from a simple specification test”, Review of Financial Studies,1988. 1(1), 41-66.
[10] J. Murphy, “Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications”, Penguin Publication, 1999.
[11] A. Lapan, and S. Rezende, “Financial Modeling with Python” Packt Publishing, 2015.
[12] L. Chen, C. Zhu, and Y. Xie, “Financial Modeling and Simulation of Big Data”, Information, 2020. 11(5), 235.
[13] L. Andersen, and V. Piterbarg, “Interest Rate Modeling”, Atlantic Financial Press, 2010.
[14] B. Malkiel, “The Efficient Market Hypothesis and Its Critics”, Journal of Economic Perspectives, 2003. 17(1), 59-82.
[15] A. McNeil, R. Frey, and P. Embrechts, “Quantitative Risk Management: Concepts, Techniques, and Tools”, Princeton University Press, 2005.