Methods For Simulating Stock Prices

Ryan Tavolilla

May 15, 2024

1 Introduction

The following is a report on the methods used to simulate stock prices. The report will cover the mathematical models used to simulate stock prices, the mathematical theory that drives those methods, the data acquisition methods, and the analysis and interpretation of the results. The first part of this report will talk about the theories relating to stochastic processes and stochastic differential equations. Then, the report will describe in detail the mathematical model used, specifically going in depth on the methods for parameter estimation as described below. Finally, the report will describe the results from the code and the analysis and interpretation of those results.

The goal of this project was to find a method to simulate the price of a stock through time with the hopes of being able to predict future prices. The code for this project was written in Python. It uses real stock data from the Yahoo Finance API to simulate the price of a stock through time. You can find the entire code base as well as in-depth documentation on the code as well as instructions to run the code at the following GitHub repository: `https://github.com/RyanTavol/Brownian-Motion-Stock-Model`. Please feel free to follow the instructions in the README to run the code and experiment inside of `main.py` to see how the model behaves.

2 Stochastic Processes And Brownian Motion

To understand how to model and simulate the price of a stock, we must first understand how to model seemingly random processes in general. A stochastic process is a series of random events that occur over time. Stochastic processes can also be thought of as a probability distribution over all possible paths. Contrast to a deterministic process, stochastic processes have no fixed trajectory and therefore we do not know before hand which path will occur. This however does not mean that there is no information to be gained from a stochastic process, since we have an understanding of the probability distribution of the paths, we can make predictions about the future [1].

Brownian Motion is a specific type of stochastic process often also referred to as the standard Wiener Process. These two terms will be used interchangeably throughout this report. Brownian Motion is commonly used to model the random movement of particles in a fluid, however it is also often used to model financial data. The Wiener Process is a continuous-time stochastic process $\{W_t\}_{t\geq 0+}$. In other words, W_t is indexed on non negative real numbers. Brownian Motion can also be described as a limit rescaled simple random walks. As $n \to \infty$, the simple random walk converges to Brownian Motion. The standard Wiener process has the following properties:

1. $W_0 = 0$;
2. With probability 1, the function $t \mapsto W_t$ is continuous;
3. The process $\{W_t\}_{t\geq 0+}$ has stationary, independent increments;
4. The increment $W_{t+s} - W_t$ is normally distributed with mean 0 and variance t [2].

3 Stochastic Differential Equations

The movement of most things in life can be described by some form of differential equation, and the movement of stock prices is no different. So, if we can find a way to model and solve "random differential equations", we can model model complicated random processes as well. This is where Stochastic Differential Equations (SDEs) come in. SDEs are used to model systems that have some form of randomness in them and are commonly used to model financial quantities such as asset price, interest rates, and derivatives. Unlike other models such as ODEs or PDEs, SDEs have solutions that are not deterministic and instead are continuous-time stochastic processes.

A typical diffusion process can be described by the following SDE:

$$dX = a(t, X)dt + b(t, X)dW_t$$

Where the first term is known as the drift term and the second term is known as the diffusion term. The drift term describes the average rate of change of the process and the diffusion term describes the randomness of the process. The term dW_t is the differential of the Wiener Process (as described in the previous section). The solution to the SDE is a stochastic process X_t that satisfies the SDE. This differential equation is quite different from ODEs and PDEs for several reasons, but one is that the differential equation is given in differential form rather than derivative form. This is because Brownian motion, among other stochastic processes are continuous but not differentiable [3].

In order to understand SDEs, we must first understand stochastic calculus. Stochastic calculus is a branch of mathematics that operates on stochastic processes. This report will not be able to go into the details of stochastic calculus in this report, but it will provide a brief overview. The most important concept in stochastic calculus that relates to this project is Ito's Lemma, which is truly the cornerstone of stochastic calculus. Ito's Lemma is used to find the differential of a function of a stochastic process (like the SDE described above). Ito's Lemma is used to solve SDEs and is used to find the solution in this model as well. The lemma is as follows: Suppose f is a C^2 function and B_t is a standard Brownian Motion (Wiener Process). Then for every t,

$$f(B_t) = f(B_0) + \int_0^t f'(B_s)dB_s + \frac{1}{2}\int_0^t f''(B_s)ds$$

In differential form, this can be written as:

$$df(B_t) = f'(B_t)dB_t + \frac{1}{2}f''(B_t)dt \quad [4]$$

For the purposes of this project, Ito's Lemma to will be used to solve the SDE that models the stock price. This report will go into more detail on this in the next section. However, it is important to note that many SDEs require the use of numerical methods to solve them. The numerical methods used to solve SDEs are based on many of the same principles as the numerical methods used to solve ODEs, however the methods are adapted to work with stochastic processes. The numerical methods for solving SDEs have both strong and weak convergence properties. The strong convergence property means that the numerical method converges to the true solution of the SDE with probability 1. The weak convergence property means that the numerical method converges to the true solution in expectation. Some common numerical methods for solving SDEs, often inspired from their ODE counterparts, are the Euler-Maruyama method, the Milstein method, and the Runge-Kutta method, and the Taylor Method [3]. There are however a certain class of SDEs that can be solved analytically, which is something will be explore more in the next section when the model for the stock price is introduced.

4 Mathematical Model For Stock Price

Now that we have an understanding of stochastic processes and SDEs, we can begin to model the stock price. A typical model for a "time-evolution" of an asset prices $S(t)$ is given by a Geometric Brownian motion [5]. The term "geometric" in Geometric Brownian Motion refers to the fact that the variable's growth rate that is being modeled is proportional to its current value. The SDE for the Geometric Brownian Motion is given by:

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

In the above SDE, the parameters μ and σ help characterize the growth of the asset. The parameter μ is the drift term and characterizes the average rate of return of the asset. The parameter σ is the diffusion term and characterizes the volatility of the asset. The term $dB(t)$ is the differential of the Wiener Process [3]. The parameters μ and σ are intentionally described generally because it allows the model to appear very simple. However, in practice, these parameters can be very complicated, which is something that will be described more in the next section.

Solving the SDE gives a solution for the stock price $S(t)$, meaning it is a function that describes the price of the stock at any given time, which is the goal of the project. As described in the previous section, most SDEs require numerical methods to solve them. However, the SDE for the Geometric Brownian Motion can be solved analytically. Using Stochastic calculus, the SDE can be solved to get the process describing the stock price. The solution to the SDE is given by:

$$S(t) = S(0)\exp\left(\left(\mu - \frac{1}{2}\sigma^2\right)t + \sigma B(t)\right)$$

However, let's prove this solution. To do this, we will use Ito's Lemma. Let $S = f(t, B_t)$, where $f(t, Y) = S(0)e^Y$. Where $Y = \left(\mu - \frac{1}{2}\sigma^2\right)t + \sigma B(t)$. Then, we can use Ito's Lemma to find the differential of S:

$$dS = S(0)e^Y dY + \frac{1}{2}e^Y dY dY$$

Where $dY = \left(\mu - \frac{1}{2}\sigma^2\right)dt + \sigma dB(t)$ and $dY^2 = \sigma^2 dt$. Using a differential identity from Ito's Lemma, we can simplify the above equation to:

$$\begin{aligned} dS &= S(0)e^Y\left(\mu - \frac{1}{2}\sigma^2\right)dt + S(0)e^Y\sigma dB(t) + \frac{1}{2}S(0)e^Y\sigma^2 dt \\ &= S(0)e^Y\mu dt + S(0)e^Y\sigma dB(t) \\ &= \mu S(t)dt + \sigma S(t)dB(t) \quad [3] \end{aligned}$$

This is the same as the SDE for the Geometric Brownian Motion, so we have proven the solution to the SDE. This solution is important because it allows us to model the stock price. If you notice in the above solution, the stock price is a function of the Wiener Process. This is important because it means that the stock price is a stochastic process. The stock price is also determined by the parameters μ and σ. In theory, if one knew the exact values of μ and σ, they could predict the future price of the stock. This is something that will describe more in the next section, but these parameters can come in many forms. The parameters can be constant values, functions depending on time, functions depending on the stock price, function depending on both, among functions depending on several other variables or processes [6].

5 Parameter Estimation Methods

As described in the previous section, the parameters μ and σ are crucial to the model. The parameters can often be quite complicated, so the goal is to estimate the parameters instead. There are many methods to estimate the parameters of the model. This report is going to describe several methods used in this project to estimate the parameters in this section as well as other methods that could be used which were not used for this project. Some of these methods are based on statistical methods and some are based on financial theory, but almost all of them require some form of data to estimate the parameters. In this case, this came in the form as historical stock data. More of this topic will describe in the next section how this project acquired the data to estimate the parameters. The goal of this project is to estimate these parameters using these different methods and then compare the results of all the methods.

5.1 Constant Parameters

The Fixed Parameter method is based on the theory that in general the parameters μ and σ don't change incredibly much with time. So, if one could get a good approximation of the parameters as constant values, they will give a good idea of the general trend and behavior of the stock. This method is quite simple admittedly, but it can be quite effective especially over longer periods of time. For this project, the specific fixed parameters are used:

$$\mu = 0.08, \quad \sigma = 0.2$$

This report will cover more about this in the next section about data acquisition, but I specifically choose to use stock data from the S&P 500 in my code, so these parameters were chosen based on roughly the average return and volatility of the S&P 500.

5.2 Capital Asset Pricing Model (CAPM)

The Capital Asset Pricing Model (CAPM) is a popular financial model used to determine the expected return on an investment based on its risk. Here is the formula used to estimate μ:

$$\mu = r_f + \beta(r_m = r_f)$$

Where r_f is the risk-free rate of return of an investment with no risk of financial loss, typically the rate of return on a U.S. Treasury Bond. r_m is the expected return of the market, in this case the S&P 500. β is the beta of the stock, which is a measure of the stock's volatility in relation to the market, which can be found on many financial websites. The formula to estimate σ is:

$$\sigma = \frac{s}{\sqrt{t}}$$

Where s is the standard deviation of the stock and t is the time-step interval, often 1/252 (the number of business trading days in a year). This method is quite popular in finance and is used to estimate the expected return of a stock based on its risk [7].

5.3 Bootstrap Method

The Bootstrap Method is a statistical method used to estimate the parameters of a model. It is a resampling technique where multiple samples are drawn with replacement from the original dataset [8].

The Bootstrap Method defines the following formulas to estimate μ:

$$R_t = \frac{S_t - S_{t-1}}{S_{t-1}}, \quad \mu = \frac{1}{M\delta t}\sum_{t=1}^{M} R_t$$

Where R_t is the return of the stock at time t, S_t is the stock price at time t, M is the number of time steps, and δt is the time step interval (amount of stock return data). The Bootstrap Method also defines the following formulas to estimate σ:

$$\sigma_1 = \sqrt{\frac{1}{(M-1)\delta t}\sum_{t=1}^{M}(R_t - \bar{R})^2}, \quad \sigma_2 = \sqrt{\frac{1}{(N-1)\delta t}\sum_{t=2}^{N}(\log(S_t) - \log(S_{t-1}))^2}$$

Where σ_1 is referred to as the common volatility, σ_2 is referred to as the log-volatility, $\bar{R}$ is the average return of the stock, N is the number of stock prices, M is the number of time steps (amount of stock return data), and δt is the time step interval [9]. I implement both types of volatility in my code and will compare them separately when I describe my results.

5.4 Method Of Moments

The Method of Moments is a statistical method used to estimate the parameters of a model. The method is based on the theory that the moments of the data should be equal to the moments of the model. In theory, the Method of Moments aims to find the parameter values that the sample moments (such as the mean, variance, skewness, etc.) match the corresponding population moments of the model.

Here is how the Method of Moments works in general:

1. Calculate the sample moments of the data.
2. Calculate the population moments of the model.
3. Set the sample moments equal to the population moments and solve for the parameters.
4. Use the estimated parameters to simulate the model.
5. Compare the results of the model to the data.
6. Repeat the process until the model fits the data [10].

In this specific implementation, the sample mean and standard deviation of the log returns of the stock were used to estimate the drift and volatility parameters μ and σ of the model. Given a set of stock prices P_t, we first calculate the log returns R_t as follows:

$$R_t = \ln\left(\frac{P_t}{P_{t-1}}\right)$$

Then, the drift parameter μ and the volatility parameter σ are estimated using the Method of Moments as follows:

$$\mu = \frac{1}{\delta t} \cdot \frac{1}{N} \sum_{t=1}^{N} R_t$$

$$\sigma = \frac{1}{\sqrt{\delta t}} \cdot \sqrt{\frac{1}{N-1} \sum_{t=1}^{N} (R_t - \mu)^2}$$

where δt is the time step, N is the number of log returns, and R_t is the log return at time t.

5.5 Other Methods For Estimating Parameters

There are many other methods for estimating the parameters of the model that were not used in this project, but were still worth mentioning. The main reason these methods were not used is due to implementation details, hardware factors, and time constraints. However, these methods are still valid and could be used in future projects. Some of these methods are: Maximum Likelihood Estimation, Bayesian Inference, Stochastic Gradient Descent, and other methods. Many of these methods require some form of optimization, which can be quite computationally expensive. That isn't so say that these methods are not useful, but they require more time and resources to implement, which unfortunately I did not have for this project. However, if I am able to continue this project in the future, I would like to explore these methods further.

6 Data Acquisition

A crucial part of this project was acquiring the stock data to estimate the parameters of the model. This code uses the Yahoo Finance API to get the stock data. The Yahoo Finance API is a free API that allows you to get stock data for any stock on the market. The API provides a wide range of data including the stock price, the stock volume, the stock open price, the stock close price, and many other data points. The API is quite easy to use and is well documented. The API provides the data in JSON format, which is quite easy to work with in Python. The main reason that the Yahoo Finance API was chosen is because it is free and doesn't require the use of an API key. This is important because it allows anyone to run the code without having to sign up for an API key [11] [12].

Because of the Yahoo Finance API, the code was able to use real stock data to model the price of stocks with. Not only is this important for the realism of the model, but it also allows for the model to be tested on real data. For example, using stock data up until the start of 2023 allows me to simulate data from 2023-2024 and compare the simulation to the real stock prices in that time period. This also allows for the model to be tested and analyzed using various statistical metrics, which will be describe in the next section.

An important thing to note is that since random processes are involved in this model, it is important that there is consistency in the simulations. When simulating the code, it is ensured that the random seed is reset before each simulation of each method. This is important because it guarantees that the results of the simulation are consistent and reproducible. The other reason why it is important is because it allows for the comparison of the results of the different methods. If the random seed was not reset, the results of the simulation would be different each time, which would make it difficult to compare the results of the different methods. This ensures that the differences that occur between methods is due to the method itself and not due to randomness.

7 Model Validation

In order to validate the model, I used the data from the Yahoo Finance API to compare the results of the model to the real stock prices. The data from the start of 2023 to the start of 2024 was used to simulate the stock prices from 2023-2024. It is important to have certain metrics to compare the results of the model to the real stock prices. This allows for the model to be tested and analyzed in a quantitative way and allows for the comparison of the results of the different methods. There are 3 main metrics commonly used for comparing methods like these to the true stock price:

1. Correlation Coefficient: The correlation coefficient is a measure of the strength and direction of a linear relationship between two variables. The correlation coefficient is a good metric to use to compare the results of the model to the real stock prices because it gives a good idea of how well the model fits the data. The correlation coefficient is calculated as follows:

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n(\sum x^2) - (\sum x)^2][n(\sum y^2) - (\sum y)^2]}}$$

   The correlation coefficient ranges from -1 to 1. A correlation coefficient of 1 means that the two variables have a perfect positive linear relationship, a correlation coefficient of -1 means that the two variables have a perfect negative linear relationship, and a correlation coefficient of 0 means that the two variables have no linear relationship.

2. Mean Absolute Percentage Error (MAPE): The Mean Absolute Percentage Error (MAPE) is a measure of the accuracy of a model. The MAPE is calculated as the average of the absolute percentage errors of the model. The MAPE is a good metric to use to compare the results of the model to the real stock prices because it gives a good idea of how accurate the model is. The MAPE is calculated as follows:

$$\text{MAPE} = \frac{\sum_{t=1}^{n} \left| \frac{A_t - F_t}{F_t} \right|}{n}$$

   To provide context to the MAPE, a MAPE of 0 means that the model is a perfect fit to the data, a MAPE of less that 10% is considered a highly accurate forecast, a MAPE of 10-20% is considered to be pretty good, and a MAPE between 20% and 50% is considered to be acceptable. A MAPE any higher is considered to be a poor forecast.

3. Percentage Of Correct Predictions (Within Threshold): This will also often be referred to as percentage of inliers. This metric is used to determine the percentage of predictions that are within a certain threshold of the true value. In this case, I used a threshold of 10%. This means this gives the percentage of predicted points that are withing 10% of the true stock price. This is a good metric to use to compare the results of the model to the real stock prices because it gives a good idea of how well the model fits the data [7].

There are many other metrics that could be used to compare the results of the model to the real stock price, but these are the most common and the most useful for this project. Specifically, MAPE was the prevalent metric used in many similar projects, so it will be the main focus in this project.

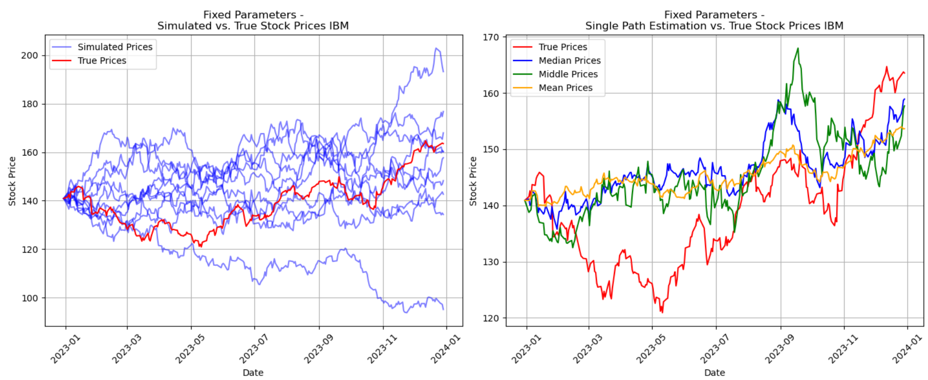
8 Analysis and Interpretation

In this section, I will specifically be analyzing and interpreting the results of a very specific test case of the model. However, it is important to note that this code can be used to model any number of different stocks in the market over any time period. As stated in the introduction, the code is available on GitHub and can be run by anyone. The code is quite simple to use and is well documented for use of testing and validation. This section will describe 3 main things: the results of the model on one specific stock, the aggregate results of the model on several stocks, and using the model to predict future stock prices.

8.1 Individual Stock Analysis (IBM)

IBM was chosen as the stock to analyze in this section. The choice of IBM was quite arbitrary, but it benefits from the fact that it is well-known and has a long history of stock data. This allows for the model to be tested on a wide range of data. I am going to simulate IBM stock prices from the start of 2023 to the start of 2024 using the full stock history of IBM as the data. The stock simulation will be shown as a series of graphs that show the simulated stock prices compared to the true stock prices. Finally, the results of the model will be compared to the true stock prices using the metrics described in the previous section.

To give context to the figures that will be shown below, there will be two plots shown per figure. Each dual plot will be representative of a different method used to estimate the parameters of the model. The left plot of each figure will be a comparison between the true stock prices and several simulated paths of the stock price. This is the best way to visualize the idea of a probability distribution of possible paths. The right plot of each figure will be a comparison between the true stock prices and the 'middles' of the simulated stock prices. This is the best way to visualize the idea of the expected path of the stock price. The 'middles' of the simulated stock prices are given by the mean, median, and middle (minimum distance to all other paths). In general, the comparisons are based on the middle paths, but the other paths are shown for context, as they are important to the model.



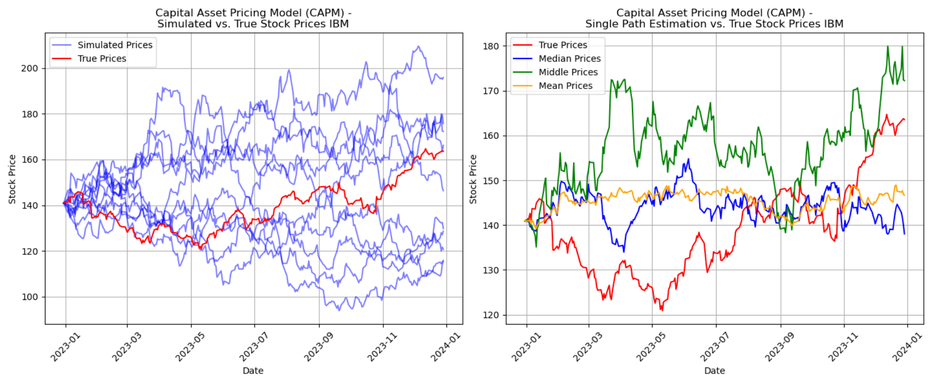
Analysis Group	Correlation Coeff	MAPE	% Inliers
Multiple Paths	0.299209	0.106161	53.9841
Average of Single Paths	0.630222	0.0690384	73.9708

Figure 1: IBM Stock Simulation Using Fixed Parameters

Figure 1 shows the results of the model using the Fixed Parameters method. As we can see, the model does a decent job of simulating the stock price. The model does a good job of capturing the general trend of the stock price, but it does not do a great job of capturing the volatility of the stock price. However, for such a simple method, the model does a good job of simulating the stock price. The correlation coefficient is 0.630222, the MAPE is 0.0690394, and the percentage of inliers is 73.9708. Correlation coefficient is pretty good, however something seen in many of the methods for many different stocks tested is that generally, the correlation coefficient didn't seem to indicate necessarily how well the model predicted the stock price, but it is still important to analyze. The MAPE is quite low, which indicates that the model is quite accurate. Remember, anything around 10% is considered a good forecast. The percentage of inliers is quite high as well. These results are quite surprising because the model is so simple, but it shows that this model can be quite effective if there is a good indication of what the constant parameters of the market look like.

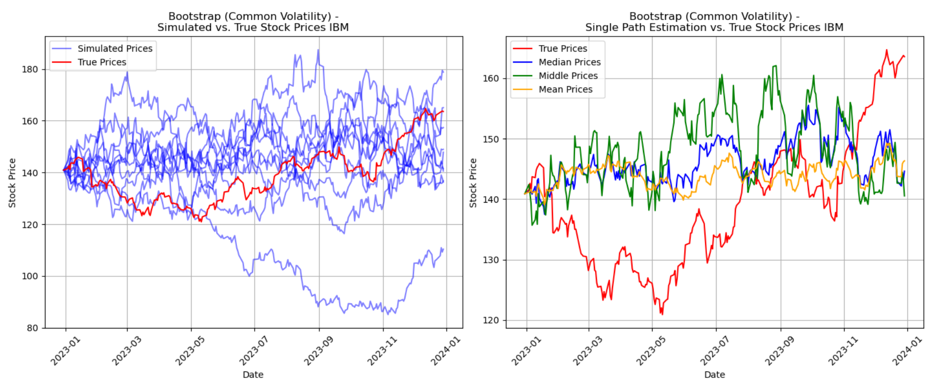
Figure 2 shows the results of the model using the Capital Asset Pricing Model (CAPM) method. As we can see once again, the model using this specific parameter estimation method does a decent job of simulating the stock price. The model does a decent job of capturing the general trend of the stock price, but falls short when capturing the volatility of the stock price. When analyzing the metrics, the correlation coefficient is -0.128619, the MAPE is 0.101011, and the percentage of inliers is 56.1753. The correlation coefficient indicates that there isn't much of a linear relationship between the model and the true stock price. The MAPE is quite low, which indicates that the model is quite accurate and the percentage of inliers is quite high as well. Having roughly 50 % of the predictions within 10% of the true stock price is quite decent.

Figure 3 shows the results of the model using the Bootstrap Method with common volatility. The model does a



Analysis Group	Correlation Coeff	MAPE	% Inliers
Multiple Paths	-0.106906	0.152252	40.4382
Average of Single Paths	-0.128619	0.101011	56.1753

Figure 2: IBM Stock Simulation Using CAPM



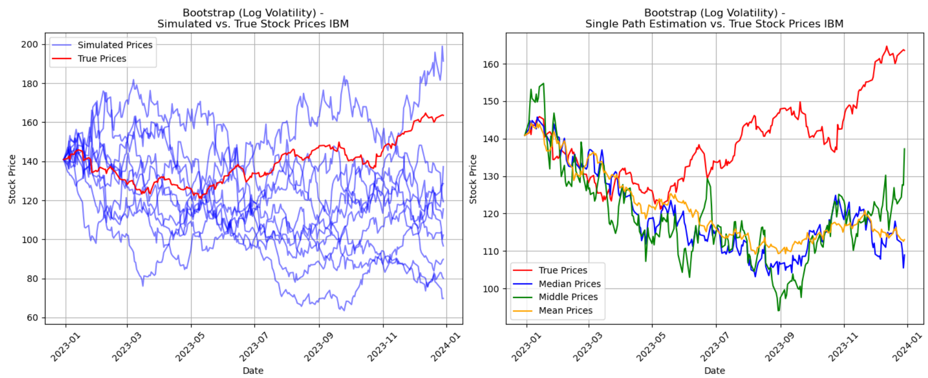
Analysis Group	Correlation Coeff	MAPE	% Inliers
Multiple Paths	0.0577446	0.11127	54.2629
Average of Single Paths	0.151154	0.0828053	60.6906

Figure 3: IBM Stock Simulation Using Bootstrap Method (Common Volatility)

good job of simulating the stock price. In general, all of the simulated prices are quite close to the true stock price. When analyzing the metrics, the correlation coefficient is 0.151154, the MAPE is 0.0828053, and the percentage of inliers is 60.6906. The MAPE is extremely low, indicating that the model is very accurate and the percentage of inliers is quite high as well. In general, the model does a good job of simulating the stock price.

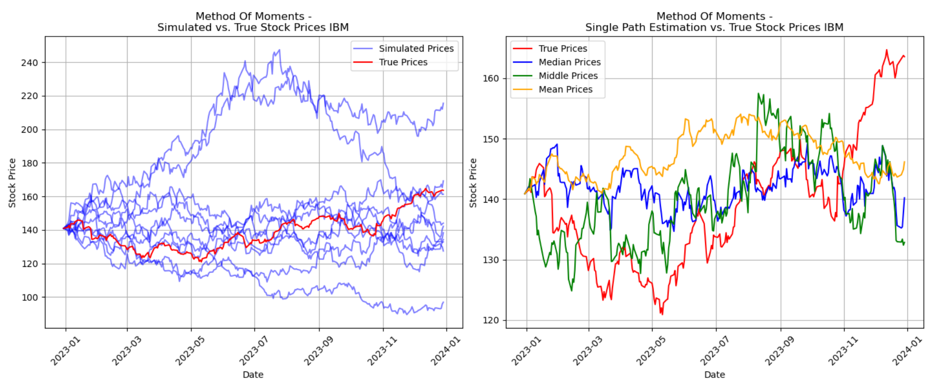
Figure 4 shows the results of the model using the Bootstrap Method with log volatility. As can be viewed in the plots, the model in the first half of the year is an extremely accurate prediction of the true stock price, however that cannot be said for the second half of the year. When analyzing the metrics, the correlation coefficient is -0.260495, the MAPE is 0.141545, and the percentage of inliers is 41.9655. Compared to the other methods seen in previous figures, this model does not perform as well, however it is still quite a decent model. The MAPE, while higher than the rest is still within the realm of what a good to fair forecast would be. The percentage of inliers is definitely lower than the rest, but still quite decent. Having roughly 40% of the predictions within 10% of the true stock price means that generally the middle of the paths was fairly close to the true stock price. And when looking at the plot, the points that were inliers were extremely close to the true plot.

Figure 5 shows the results of the model using the Method of Moments. This model performs very well at simulating the stock price. The model does a good job of capturing the general trend of the stock price and the volatility of the stock price. When analyzing the metrics, the correlation coefficient is 0.24243, the MAPE is 0.0708245, and the percentage of inliers is 71.1819. The correlation coefficient is decent, but as stated before



Analysis Group	Correlation Coeff	MAPE	% Inliers
Multiple Paths	-0.190193	0.184571	32.8287
Average of Single Paths	-0.260495	0.141545	41.9655

Figure 4: IBM Stock Simulation Using Bootstrap Method (Log Volatility)



Analysis Group	Correlation Coeff	MAPE	% Inliers
Multiple Paths	0.0589796	0.158986	51.8327
Average of Single Paths	0.24243	0.0708245	71.1819

Figure 5: IBM Stock Simulation Using Method of Moments

doesn't necessarily indicate how well the model predicted the stock price. However, the MAPE is quite low, which definitely indicates that the model is very accurate. The percentage of inliers is quite high as well, meaning many of the simulated prices were quite close to the true stock price.

In general this model with these parameter estimation methods does a good job of simulating the stock price of IBM from 2023-2024. The model does a good job of capturing the general trend of the stock price and the volatility of the stock price. This is shown both qualitatively and quantitatively through the plots and the metrics. All of the plots generally seem to trend with the true stock price, and the metrics certainly show that the model is quite accurate. All of the MAPE were in the range of what is considered anywhere from a strong forecast to a good forecast. This is also true for the percentage of inliers. This of course is only one specific test case of the model, but it shows that the model can be quite effective at simulating the stock price of certain individual stocks. The next subsection is going to look at how the model performs on a larger number of stocks to see if the results are consistent across many different stocks.

8.2 Aggregate Stock Analysis (S&P 500)

In this subsection, the model is going to be tested on a larger number of stocks to see if the results are consistent across several more stocks. The model is still going to be tested using the same methods as before, but the results are going to be aggregated across several stocks. All the stocks used in this subsection are going to be stocks from the S&P 500 still. The results of the model are going to be compared to the true stock prices using the metrics described in the previous section. Because there are so many stocks in the S&P 500, it is not feasible to show the plots of all the stocks, so instead the results are going to be shown in a table only. The table is going to show the results of the model using the different methods on several stocks. The table is going to show the correlation coefficient, the MAPE, and the percentage of inliers for each stock and each method. The table is going to show the average of the metrics for each method across all the stocks. This is going to give a good idea of how the model performs on a larger number of stocks. The model was simulated on 50 stocks from the S&P 500. The stocks were chosen randomly from the S&P 500. The model was simulated from the start of 2023 to the start of 2024 using the full stock history of each stock as the data similarly to the last section. The results of the model are shown in the table in Figure 6.

Here is the list of the stocks used in the analysis: AAPL, DG, DIS, IBM, INTC, NVDA, PG, SHW, TXN, AMZN, GOOGL, MSFT, TSLA, JNJ, JPM, KO, VZ, NFLX, CSCO, PEP, WMT, UNH, HD, BAC, MA, V, MRK, CMCSA, META, PYPL, ABBV, ABT, ADBE, ADI, ADP, ADSK, AEP, ALGN, ALL, AMAT, AMD, AMGN, RMD, AON, APD, APH, AZO, BA, BAX. As stated before, this list is quite arbitrary, but hopefully should give a good representation of the model across a wide range of stocks across the whole market.

Analysis Group	Correlation Coeff	MAPE	% Inliers
Fixed Parameters	0.122021	0.16489	44.5158
CAPM	0.0402953	0.186529	41.3584
Bootstrap (Common)	0.113854	0.182328	42.9628
Bootstrap (Log)	-0.0599679	0.266048	27.715
Method of Moments	0.108545	0.176828	40.2553

Figure 6: Aggregate Results of Model on S&P 500 Stocks

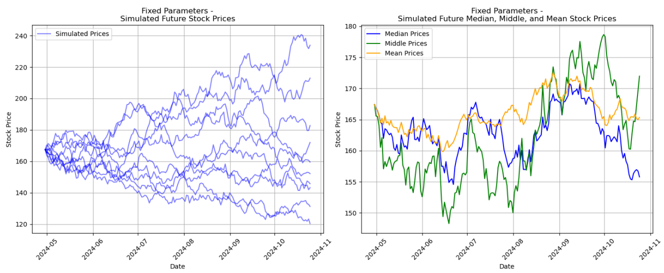
Analyzing the aggregate results shows that our model is quite effective collectively at simulating the stock price of the S&P 500 stocks. Because of the nature of how the model is being compared, it is hard to visualize the results, but the metrics show that the model is quite accurate collectively. Figure 6 shows the average of the metrics for each method across all the stocks.

The correlation coefficient for all of the methods is quite low, indicating that there is practically no correlation between the true stock price and the simulated stock price. However, as stated before, the correlation coefficient doesn't necessarily indicate how well the model predicted the stock price. Collectively, there wasn't much of a correlation, but that doesn't mean that there weren't stocks that had a high correlation or that the model didn't predict the stock price well. The MAPE and percentage inliers will be a better representation of how well the model predicted the stock price. The MAPE for all of the methods is quite low, indicating that the model is quite accurate. Remember that a MAPE between 10% and 20% is considered to be pretty good, considering that the MAPE for almost all of the methods is in this range, it shows that the model is quite accurate and robust as a means of predicting the stock price in general. The percentage of inliers isn't particularly high, but is quite decent still. Having roughly 40% of the predictions within 10% of the true stock price is an indication that the simulations were relatively nearby the true stock price. This is also consistent with what we saw in the individual stock analysis.

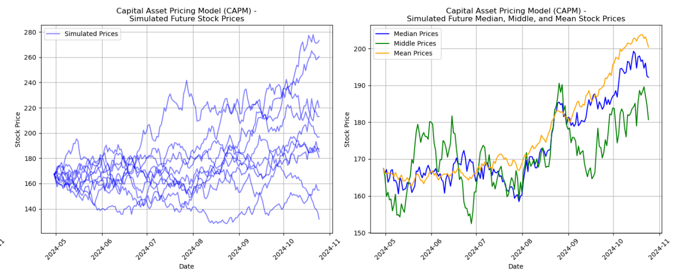
In general, this is an indication that this model performs quite well when it comes to forecasting the stock price over a wide range of stocks. The model is quite accurate and robust and can be used to predict the stock price of any stock in the market. Even the simple methods used in this project were quite effective at predicting the stock price on this time interval. The next subsection is going to look at how the model can be used to predict the future stock price of a stock.

8.3 Predicting Future Stock Prices (AAPL)

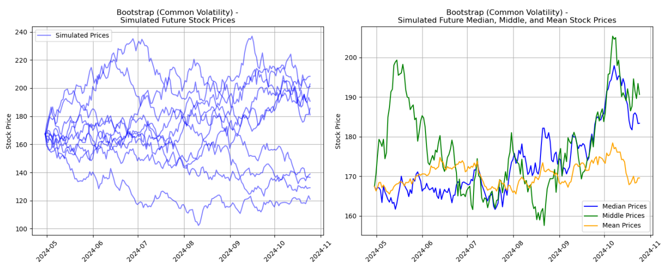
Finally, this subsection is going to explore how the model can be used to predict the future stock price of a stock. Because this model has been shown to be quite accurate and robust, it is safe to say it would probably do a decent job predicting the future price of a stock. The model is going to be used to predict the stock price of AAPL from the start of May 2024 to the start of 2025. The model is going to be simulated using the full stock history as the data. The difference between this section and the prior section is that it will not be possible to compare the results of the model to the true stock price because the true stock price is not known. As such there isn't much analysis that is possible for this subsection, but it is more to showcase how the model can be used to predict the future stock price of a stock. Remember that this code can be changed to simulate any stock in the market along with any time frame. The results will be shown in Figure 7 below and are up to the reader to interpret.



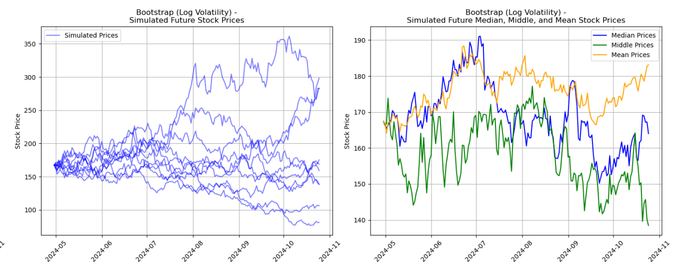
(a) Future Prediction Using Fixed Parameters



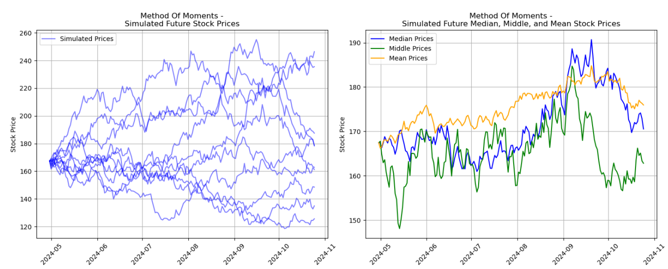
(b) Future Prediction Using CAPM



(c) Future Prediciton Using Bootstrap (Common)



(d) Future Prediction Using Bootstrap (Log)



(e) Future Prediction Using Metho of Moments

Figure 7: AAPL Stock Future Prediction Using Different Methods

References

[1] MIT, "Stochastic processees 1," *Topics in Mathematics with Applications in Finance*, 2013. [Online]. Available: https://ocw.mit.edu/courses/18-s096-topics-in-mathematics-with-applications-in-finance-fall-2013/f5784e4facf3de690210d17c97358eba_MIT18_S096F13_lecnote5.pdf

[2] U. of Chicago, "Brownian motion." [Online]. Available: https://galton.uchicago.edu/~lalley/Courses/313/WienerProcess.pdf

[3] T. Sauer, "Numerical solutions of stochastic differential equations in finance," *George Mason University*. [Online]. Available: https://math.gmu.edu/~tsauer/pre/sde.pdf

[4] Y. Yoo, "Stochastic calculus and black-scholes model," *UChicago*. [Online]. Available: https://math.uchicago.edu/~may/REU2017/REUPapers/Yoo.pdf

[5] K. Siegrist, "Geometric brownian motino," *LibreTexts Statistics*, 2022. [Online]. Available: https://stats.libretexts.org/Bookshelves/Probability_Theory/Probability_Mathematical_Statistics_and_Stochastic_Processes_(Siegrist)/18%3A_Brownian_Motion/18.04%3A_Geometric_Brownian_Motion

[6] QuantStart, "Geometric brownian motion." [Online]. Available: https://www.quantstart.com/articles/Geometric-Brownian-Motion/

[7] K. Reddy and V. Clinton, "Simulating stock prices using geometric brownian motion: Evidence from australian companies," *AABFJ*, 2016.

[8] P. S. University, "Bootstrapping methods," *Applied Statistics*. [Online]. Available: https://online.stat.psu.edu/stat500/lesson/11/11.2/11.2.1

[9] W. F. A. et al, "Stock price prediction using geometric brownian motion," *Journal Of Physics: Conference*, 2018. [Online]. Available: https://iopscience.iop.org/article/10.1088/1742-6596/974/1/012047/pdf

[10] P. S. University, "Method of moments," *Introduction to Mathematical Statistics*. [Online]. Available: https://online.stat.psu.edu/stat415/lesson/1/1.4

[11] "Exploring finance apis," *Python Invest*, 2020. [Online]. Available: https://pythoninvest.com/long-read/exploring-finance-apis

[12] A. Neto, "Python stock price api's," *Medium*, 2023. [Online]. Available: https://medium.com/@aguimarneto/python-stock-price-apis-e67d5310f6e3

[13] T. Areerak, "Mathematical model of stock prices via a fractional brownian motion model with adaptive parameters," *Hindawi*, 2014. [Online]. Available: https://www.hindawi.com/journals/isrn/2014/791418/

[14] J. Lidén, "Stock price predicitons using a geometric brownian motion," *Uppsala University*, 2018. [Online]. Available: https://www.diva-portal.org/smash/get/diva2:1218088/FULLTEXT01.pdf

[15] QuantStart, "Stochastic differential equations." [Online]. Available: https://www.quantstart.com/articles/Stochastic-Differential-Equations/