

Applications of Stochastic Processes in Asset Price Modeling

TJHSST Computer Systems Lab Senior Research Project
2008-2009
Preetam D'Souza

May 26, 2009

Abstract

Stock market forecasting and asset price modeling have recently become important areas in the financial world today. The increasing complexity of the stock market and the lucrative field of investment management has fueled breakthrough developments in mathematical stock price modeling. New financial instruments that rely on an underlying asset's price in the future to determine their current price require accurate methods of stock modeling. One method of mathematical modeling uses random or pseudorandom methods known as stochastic processes to determine an asset's price in the future. This project aims to demonstrate the flexibility and accuracy of these stochastic models by implementing them in code and testing them against empirical data. The main model tested in this project will be the Geometric Brownian Motion diffusion process and it will be analyzed with a variety of stocks, both stable and volatile, to appreciate an understanding of its strengths and weaknesses during differing market conditions.

Keywords: Stochastic processes, Brownian Motion, Financial Derivatives, Asset Price Modeling

Contents

1	Introduction	3
1.1	Scope of Study	3
1.2	Expected results	3
1.3	Rationale	3
2	Theory	4
3	Procedures and Methodology	4
3.1	Structure	4
3.2	Model Inputs	5
3.3	Model Accuracy	5
4	Current Results	5
A	Additional Simulation Runs	9
A.1	Google	9
A.2	Wal-Mart	10
A.3	Honda Motor Company	11
B	Source code	12
B.1	Main model class	12
B.2	Geometric Brownian class	13
B.3	Statistics class	13

1 Introduction

1.1 Scope of Study

This project examines stochastic processes to predict stock price movements. Given the current price, volatility, and expected return of an arbitrary stock, several stochastic models exist to predict changes in price. The main model that will be implemented and tested is Geometric Brownian Motion (GBM), an adaptation of the standard Brownian Motion process. A standard Brownian Motion model assumes that stock prices themselves follow a random walk process. GBM, on the other hand, assumes that stock price returns, not specifically the price, follow a stochastic process. The goal of this project is to extensively test both of these models against empirical data for a single stock (IBM) to determine accuracy. Additionally, this project seeks to develop possible variance reduction techniques that improve the validity of both models.

In order to test the models against empirical data, historical prices for a specific stock (such as IBM) in the past over an arbitrary time period are required. Yahoo! Finance has free stock price data in the past for many large companies and data from this website will be used for this experiment.

1.2 Expected results

I expect that the stochastic modeling techniques will approximate a stock's change in price after running many simulated trials and fine-tuning the model. Over several runs, the model should converge to the actual stock price fluctuations. The results of the project can be shown visually through graphs. For example, historical IBM stock prices can be plotted along with the simulated run of the stock to show the accuracy of the model.

If this project is successful, it could be of use to financial companies that use investment models to determine how to hedge their portfolios against risk. Improved methods of variance reduction to improve accuracy of these models also hold value for derivative pricing. Results from this project could also be used to further develop the implemented algorithms to more accurately model stock prices.

In general, data garnered in this experiment will presumably reveal whether stochastic based models are accurate in predicting complex stock price movements. Calibration testing will also potentially reveal possible improvements to the current models.

1.3 Rationale

Implementation of these stochastic models and extensive testing will lead to results that help to determine the accuracy and validity of these models when they are used to predict stock prices. Several financial firms tend to use these models to price their complex financial instruments and thus require a high degree of certainty that their models are correct to prevent risk and potential losses. Invalid models used to price these instruments can lead to potential mishaps for the entire economy; this can easily be seen in the housing market collapse and subsequent chaos on Wall Street where firms did not know the correct value of their mortgage-backed securities. Eliminating these models and developing improvements to them can lead to greater fundamental knowledge behind human behavior and more accurate asset pricing methods.

2 Theory

Stochastic processes are can be represented with stochastic differential equations (SDEs) that describe changes in different quantities.

Let S be the stock price, μ the drift rate (or mean), σ the volatility (or variance) of the stock, and let dZ represent a Wiener Process.

$$dZ = \phi\sqrt{dt}$$

where ϕ is drawn from a normal distribution $N \sim (0, 1)$. The SDE for Geometric Brownian Motion is given by:

$$\frac{dS}{S} = \mu dt + \sigma dZ$$

Here $\frac{dS}{S}$ represents the return on the stock. Multiplying by S to both sides of the equation one obtains:

$$dS = \mu S dt + \sigma S dZ$$

This shows that the stock price cannot change once $S = 0$, which is a requirement for this model to accurately represent stock prices.

3 Procedures and Methodology

3.1 Structure

After research of the theory behind these models, the actual models were implemented in code. Java was chosen as the prime programming language for all phases of this project. First, an RSS based class was created to retrieve real-time stock price data for an given stock from a free financial website. This will be used in later development to predict present movements of a stock price. A main statistics class was also created to act as a simple resource for calculating the mean, variance, and standard deviation for a list of a stock's historical prices over an arbitrary time period.

The main model class is responsible for data parsing and simulation. This class reads in historical price data and utilizes the aforementioned statistical convenience class to calculate inputs to the model. Once these are determined, the simulation process begins. Currently, a Geometric Brownian Motion model is being used, but this class can easily be adopted to other models as long as they follow the same convention. These stochastic models were implemented using a discrete iterative algorithm to approximate the continuous time forms of the theoretical models. During the simulation, price changes over the given trading period (usually 1 day) are printed out to a file formatted to easily be plotted with Gnuplot. This model supports simultaneous simulations so that several different sample paths for a stock price can be plotted on the same graph.

Stock prices over one year can be intuitively plotted on graphs to display the price fluctuations with a smooth curve drawn through the discrete points. Charts of key prices over a year can also be provided to demonstrate potential variations in the model from the empirical data.

3.2 Model Inputs

There are two main inputs to the Geometric Brownian Motion model that are obtained from empirical data in the past. The first is μ , the drift (mean) of the stock price's returns over the chosen period for analysis. This attribute is useful in determining the overall general drift rate of the stock and hence signals the direction towards which a stock tends to move over a long period of time. This is obtained by calculating the mean of the stock price's returns. Let S_i represent the i th stock price, then:

$$\mu = \frac{1}{n} \sum_{i=1}^n S_i$$

The other input is the volatility, or standard deviation, of the empirical data from the past. Let σ represent the volatility, then a reasonable algorithm for calculating the volatility is¹:

$$\sigma = \sqrt{\frac{1}{n-1} \left[\sum_{i=1}^n S_i^2 - \frac{1}{n} \left(\sum_{i=1}^n S_i \right)^2 \right]}$$

3.3 Model Accuracy

The accuracy of the model with respect to empirical data can be estimated by calculating the Root Mean Squared Deviation (RMSD) of the data sets. This is a measure of the average of the squared errors between the model and the empirical data. Let S_i represent the empirical price and \hat{S}_i represent the simulated price. The RMSD is then defined as:

$$RMSD = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{S}_i - S_i)^2}$$

A large value indicates large deviations between the empirical data and the model, while a value close to zero represents a good fit.

4 Current Results

The current model calculates simulated price changes for a stock and outputs them to a file that can be easily plotted along with the empirical price graph. Figures 1 and 2 on the next page demonstrate two different simulation runs of the Geometric Brownian model. The empirical data used for these runs are the IBM opening prices from January 1990 to January 1991. IBM was chosen to be an initial stock used for analysis due to its status as a blue-chip stock and hence its general stability over several years. Each run is plotted with the empirical price curve and three simulation paths. In Figure 1, notice how one simulation path runs far off the drift line while the other two move with the empirical data. In Figure 2, all three simulations hug the drift line and closely approximate the historical price changes. Different sample runs of this model can produce dramatically different results, although they can all be considered valid paths for a stock price to take. Note that these figures are only

¹This algorithm is justified by the following relationship in probability theory: $Var(X) = E(X^2) - E(X)^2$, where $Var(X)$ is the variance of a random variable X and $E(X)$ is the expected value of X .

demonstrating how accurately the Geometric Brownian process can fit empirical data in the past.

Figures 3 and 4 on the next page demonstrate the accuracy of the Geometric Brownian model for the time period January 1991 to January 1992 for IBM stock. 1991-1992 was an overall bear session for IBM stock as its value steadily declined from \$113 to \$89. In Figure 3, it can easily be observed how the GBM model overestimates IBM growth for this period due to the fact that the drift rate is positive. Each simulation tends to be centered around the mean line, and thus the averaged simulation predicts a price that is a full \$30 above the actual closing price. However, sometimes the GBM model can produce accurate results even in a down market, which can be seen in Figure 4. Seemingly against all odds, each run in this simulation runs below the drift line and yields an averaged ending price that is within \$10 from the empirical closing price.

The following graphs display just how variable each simulation can be; sometimes the model can produce accurate results but many other times it can either overestimate or underestimate a stock's potential for growth. A large part of the model is based on past data, so the importance of the choice of data for calculating the model inputs should not be neglected. See Appendix A for sample simulation runs on Google, Wal-mart, and Honda Motor Company shares.

5 Conclusions

From the numerous trials of the Geometric Brownian Motion model under varying market conditions, it appears that it is capable of accurately representing stock price movements. The stochastic differential equations that govern the GBM model seem to be a fairly reasonable mathematical model for anticipating price changes based on historical price data. However, it is prudent to keep in mind that there is no certain consistency to the model's predictions. The inherent randomness associated with the general Weiner process that governs GBM gives rise to an unpredictable and occasionally highly volatile model. One way of dealing with volatility issues is to run several simulations and average them all together to obtain an averaged simulation run. Unfortunately, extensive averaging smooths out the stochastic nature of the model and destroys its defining characteristic of seemingly random motion. In this case, the model approaches the linear drift line of the given stock. In this project, three simulations were averaged together in order to mitigate volatility while preserving the model's stochastic nature.

Given the strengths and weaknesses of the GBM model, it is insufficient to wholly concentrate on this individual model when determining investment decisions in the equity markets. Human judgment and other traditional forms of investment research should be used in conjunction with the GBM model in order to offset potential errors in the model's predictions. The GBM model may be especially useful in determining complex derivative prices that cannot be priced with the standard Black-Scholes formula. When used carefully, the GBM model can be a powerful tool in influencing investment decisions.

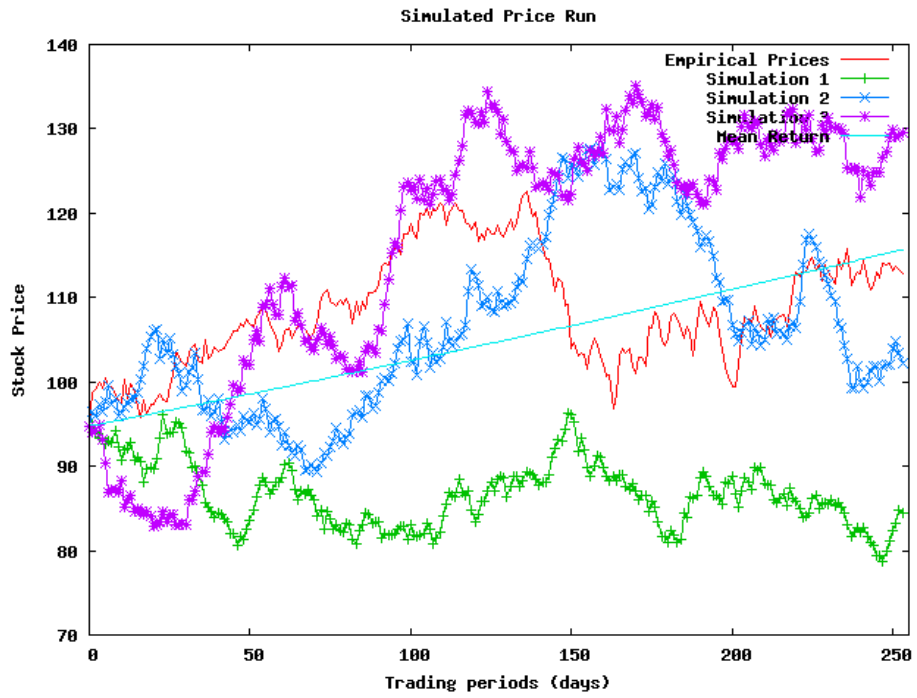


Figure 1: Simulation 1 for IBM stock during January 1990-1991

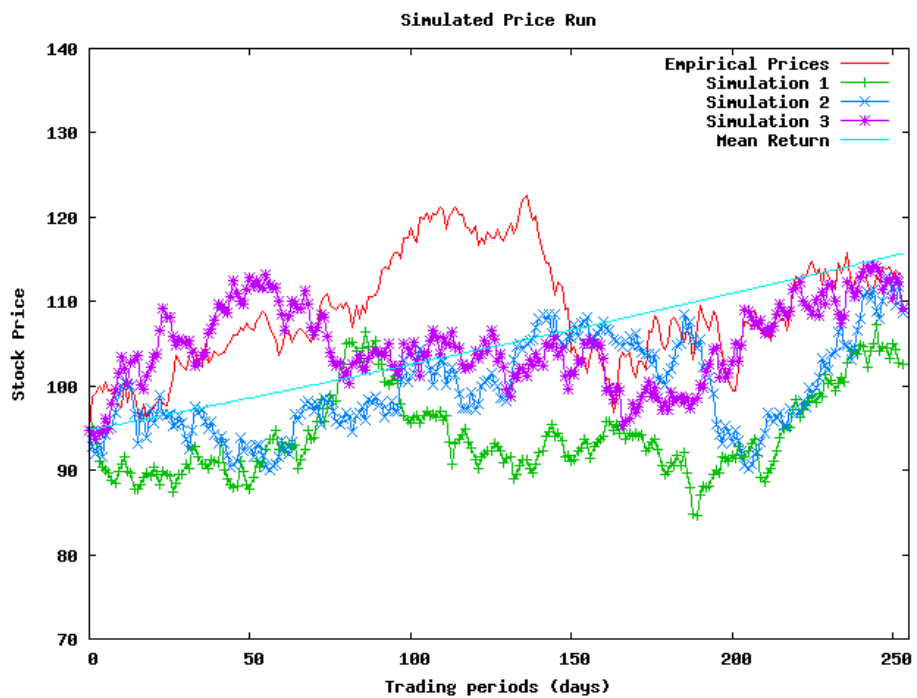


Figure 2: Simulation 2 for IBM stock during January 1990-1991

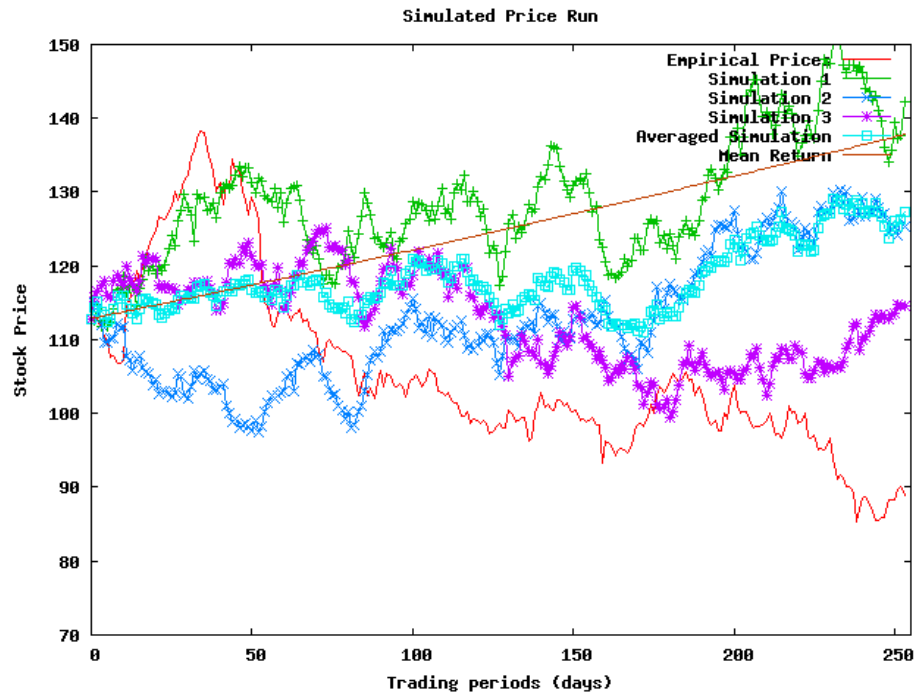


Figure 3: Simulation 3 for IBM stock during January 1991-1992

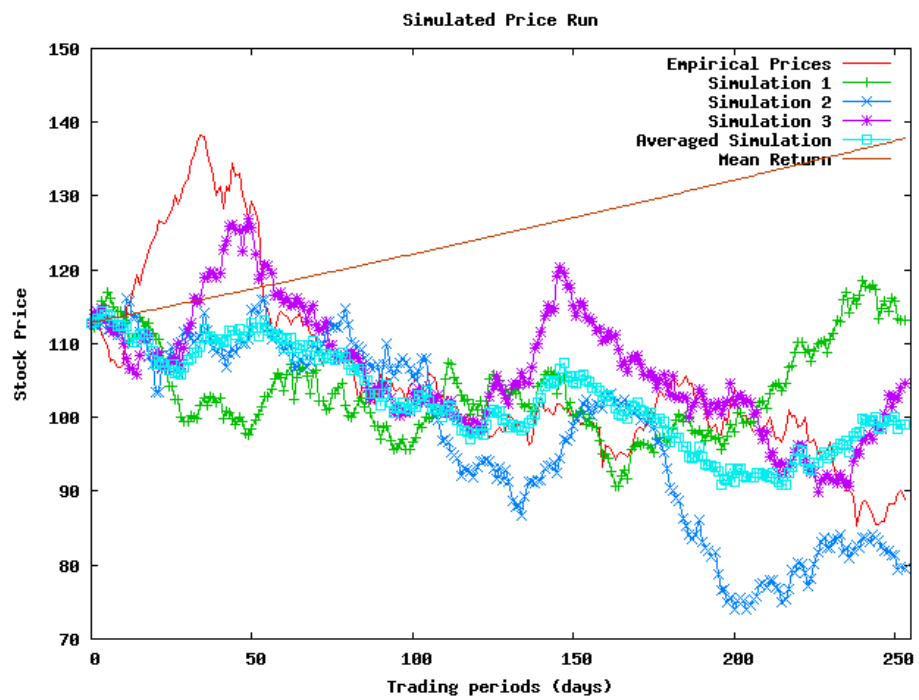


Figure 4: Simulation 4 for IBM stock during January 1991-1992

A Additional Simulation Runs

A.1 Google

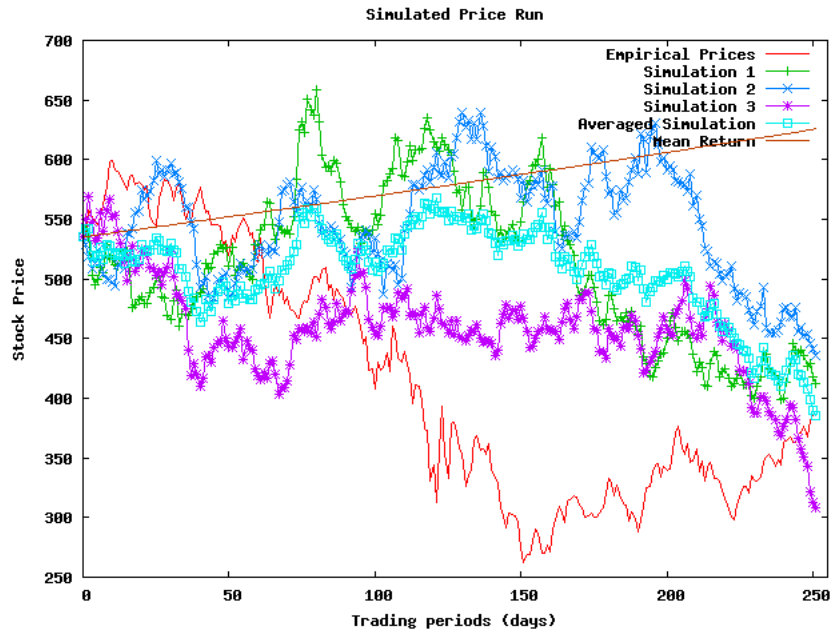


Figure 5: Simulation 1 for Google stock during April 2008-2009

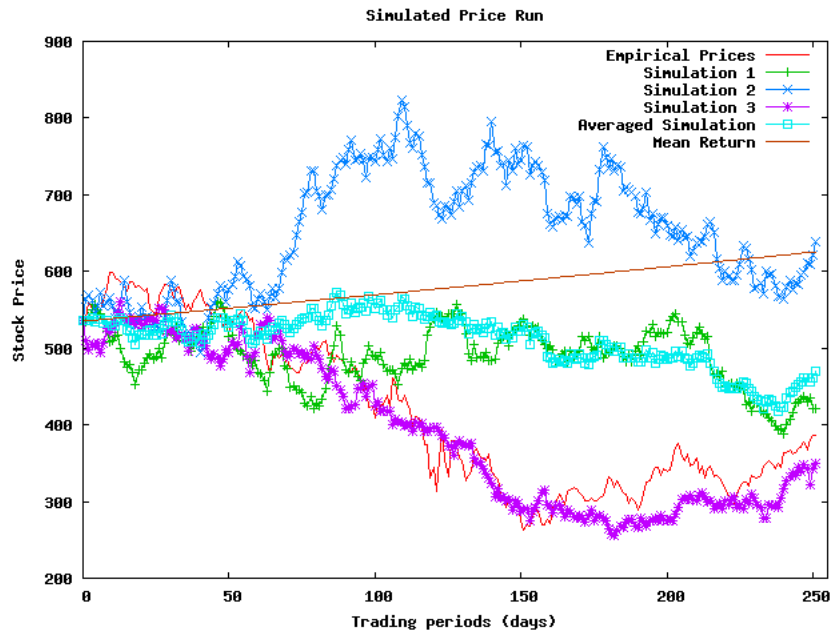


Figure 6: Simulation 2 for Google stock during April 2008-2009

A.2 Wal-Mart

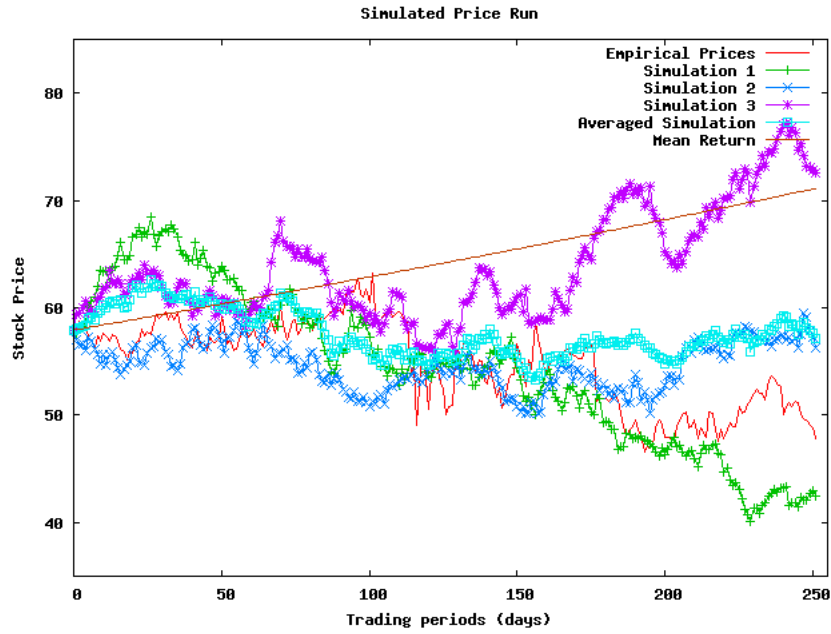


Figure 7: Simulation 1 for Wal-Mart stock during April 2008-2009

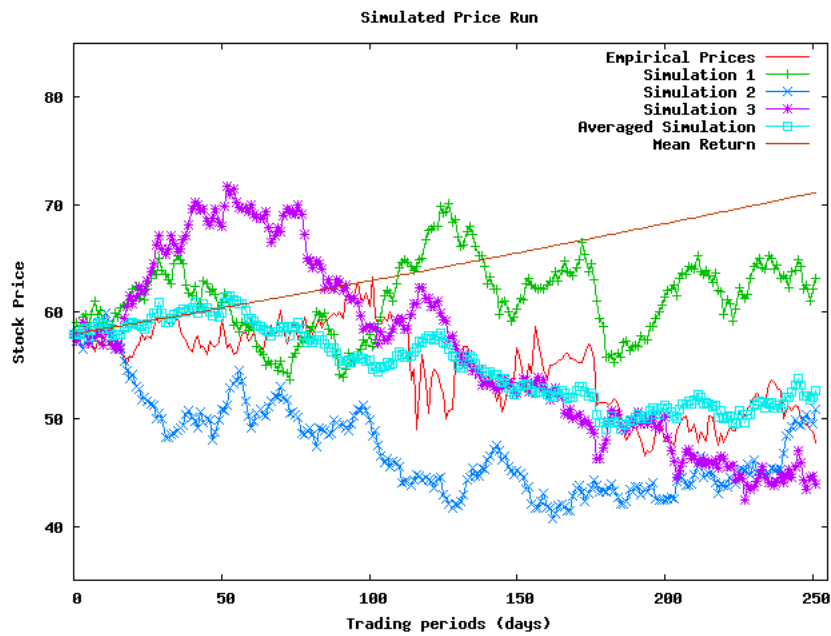


Figure 8: Simulation 2 for Wal-Mart stock during April 2008-2009

A.3 Honda Motor Company

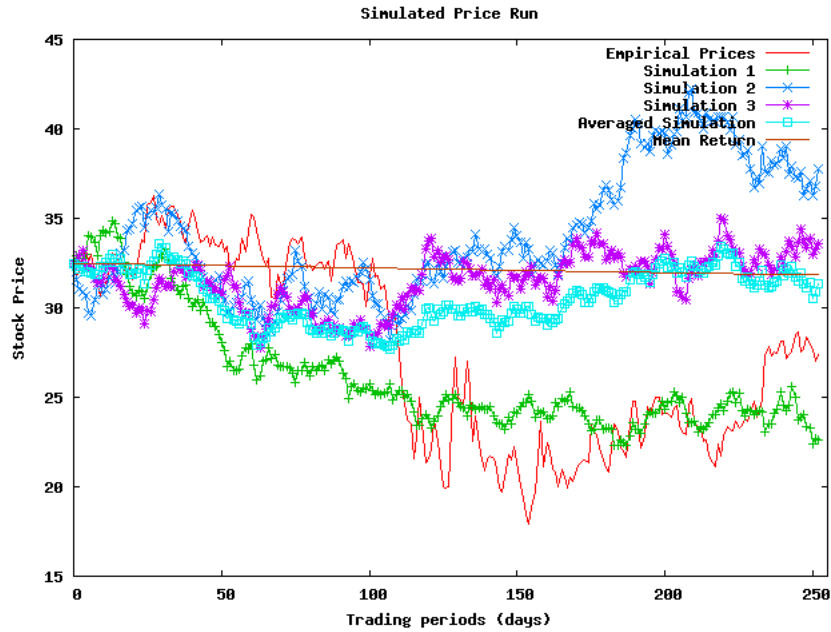


Figure 9: Simulation 1 for Honda stock during April 2008-2009

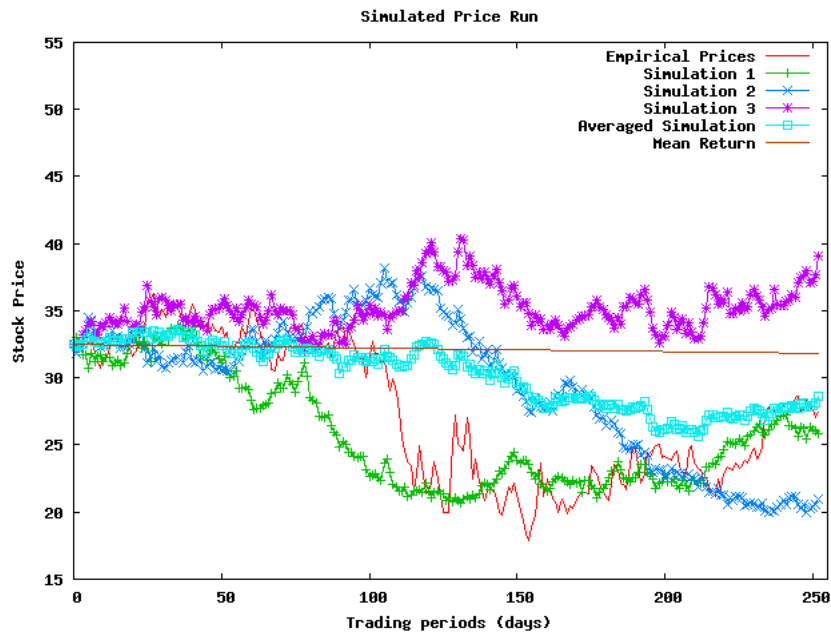


Figure 10: Simulation 2 for Honda stock during April 2008-2009


```

        for(int i=0;i<g.length;i++) {
            g[i].process(dt);
            avg += g[i].getPrice();
            sout.printf("%.5f_", g[i].getPrice());
        }
        sout.printf("%.5f_%.5f\n", avg/((double)NUM), mean_line);
        mean_line += (mean_line*drift);
    }
    System.out.println("Done!\n");

    sout.printf("#Mean:_%f\n#Volatility:_%f\n", drift, volatility);
    sc.close();
    pout.close();
    sout.close();
}
}

```

B.2 Geometric Brownian class

```

// Preetam D'Souza
/*
 * This class simulated discrete changes in stock price
 * using a geometric brownian process.
 */
import java.util.*;

public class GeoBrownian
{
    private double cur_price;
    private double mu;
    private double sigma;

    public GeoBrownian(double price, double drift, double volatility)
    {
        cur_price = price;
        mu = drift;
        sigma = volatility;
    }

    public double getPrice()
    { return cur_price; }

    public void setPrice(double price)
    { cur_price = price; }

    public double getDrift()
    { return mu; }

    public void setDrift(double drift)
    { mu = drift; }

    public double getVolatility()
    { return sigma; }

    public void setVolatility(double volatility)
    { sigma = volatility; }

    // price - current stock price
    // mu - drift parameter
    // sigma - stock volatility
    // dt - time step (usually 1/trading periods)

    public double process(double dt)
    {
        double phi;
        double dS, dZ;

        Random r = new Random();
        phi = r.nextGaussian();
        dZ = phi*Math.sqrt(dt);
        dS = cur_price*(mu*dt + sigma*dZ);
        cur_price += dS;
        return dS;
    }
}

```

B.3 Statistics class

```

// Preetam D'Souza
// *Based on Philip Barker's DataDisperion package*

/* Convenience class to implement statistics
 * algorithms for calculating the mean, variance
 * and standard deviation for a data set.
 */

```

```
*/
// Stat.java
import java.util.*;
public class Stat
{
    public Stat() {}

    public static double mean(double[] x)
    {
        double total=0.0;
        for(int i=0;i<x.length;i++)
            total += x[i];
        return total/x.length;
    }

    public static double variance(double v1[])
    {
        double sumd=0.0;
        double total=0.0;
        for(int i=0;i<v1.length;i++) {
            total += v1[i];
            sumd += Math.pow(v1[i],2);
        }
        return (sumd-(total*(total/v1.length)))/((v1.length)-1);
    }

    public static double sdev(double v1[])
    {
        return Math.sqrt(variance(v1));
    }
}
```

References

- [1] Balaji, Raman. "Introduction to Stochastic Finance", University of Connecticut.
- [2] Barker, Philip. Java Methods for Financial Engineering, Springer Publishing, May 2007.
- [3] Charnes, John. "Using Simulation for Option Pricing" School of Business, The University of Kansas.
- [4] Chance, Don. "Essays in Derivatives", Wiley Publishing, August 1998.
- [5] Forsyth, Peter. "An Introduction to Computational Finance without Agonizing Pain", School of Computer Science, University of Waterloo.
- [6] Straja, Sorin. "Stochastic Modeling of Stock Prices", Montgomery Investment Technology, Inc.