Indian Institute of Technology, Gandhinagar

Modelling Stock Market Prices using Numerical Methods

MA 202 PROJECT - II REPORT
24 APRIL 2022

TEAM MEMBERS
Aryan Gupta 20110026
Chirag Sarda 20110047
Dhairya Shah 20110054
Dhruv Parekh 20110058
Harshvardhan Vala 20110075

Under the guidance of:
PROF. SATYAJIT PRAMANIK
PROF. UDDIPTA GHOSH
PROF. TANYA SRIVASTAVA

Contents

Abstract

Brownian Motion describes a random motion and hence it has b=often been compared to the stock market model. We have derived an equation to incorporate this random nature into our mathematical model and by approximating the Weiner function as a random noise function, we have obtained data which helps us to analyse the trends of the stock market.

Problem Statement

The Stock Market has always been a fascinating matter of discussion, and there have been countless times people have tried to create models and equations to forecast the price movements. This has been a long-term key issue in financial studies. Such forecasting models are needed to predict stocks with less investment risk and to have a clear understanding of the marketplace. Hence this has inspired tremendous research in this domain.

There have been innumerable models that have been created to mimic the nature of the stock market, like the ‘random walk model.’ This model reiterates the fact that the stock market is unpredictable. The fluctuations in the stock market cannot be anticipated. We have assumed the stock market follows the standard Brownian motion process, which considers them to follow a random walk model. Brownian Motion is the model described for the random motion of particles, and the un-predictive nature of the stock market inspires us to adopt a similar model.

By adopting a dynamic equation that determine the stock index for a stochastic process that follows Geometric Brownian Motion, we have modelled a way to quantize the stock index's value over time.

This equation has elements of drift coefficient and diffusion coefficient. Our goal is to use numerical methods (namely Dekker’s Method and Inverse Interpolation), to solve an approximate of this equation, compare their results for values of drift and diffusion coefficient, and ultimately try to estimate the nature of the stock index, which will allow us to predict certain trends.

Assumptions and Parameters

Assumptions

- Stock prices are continuous. They also evolve continuously with time and are driven by the Brownian motion process.
- Markets are frictionless, i.e., no transaction fees are charged. Only buying and selling of stocks take place.
- There are no sudden changes in the market due to external factors. The market is somewhat stable.
- We are considering a financial market of n+1 stocks.

Parameters

- B_t - set of random variables at different values of t in the given interval
- μ = Drift coefficient
- σ = Diffusion coefficient
- SI(t) = The value of the stock market index at time t
- N - Market Size

Mathematical Model Formation

Geometric Brownian Motion

A geometric Brownian motion is a continuous-time stochastic process in which the logarithm of a randomly fluctuating variable follows the equation of Brownian motion. This process is similar to the Weiner process. It's a good example of a stochastic process that can fulfill a stochastic differential equation. It is used in mathematical finance to simulate stock prices in particular.

Stochastic process: A stochastic process, sometimes called a random process, is a set of random variables indexed by a mathematical set. The random process involves observations at certain points in time, and the outcome, i.e., the observed value at each point in time, is a random variable. Each probability and the random process has a distinct relationship with one of the set's elements. The indexed mathematical set collects random variables used to index them.

Brownian motion and the Wiener process: The Wiener process is a continuous-time stochastic process with real values used to analyze the mathematical features of one-dimensional Brownian motion.

Mathematics of Brownian Motion

Brownian Motion, $B_t, 0 \leq t \leq T$, is a set of random variables at different values of t in the given interval. This collection of random variables must have following properties:

- B_t is continuous and $B_0 = 0$.
- For each t, B_t is normally distributed with expected value 0 and variance t, and they are independent of each other.
- For any t and s the random variables B_{t+s} and B_s are independent. Also $B_{t+s} - B_s$ has variance t.

The above states some properties, which Brownian motion must possess. However, just because we want something with certain properties does not guarantee that such a thing exists.

It can be shown that Brownian motion does indeed exist, and section 5.9 of The Mathematics of Finance Modeling and Hedging by Stampfli and Goodman indicates one way to construct a Brownian motion. There is one important fact about Brownian motion is the process satisfies the stochastic differential equation

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

The crucial fact about Brownian motion, which we need is

$$(dB)^2 = dt$$

According to equation 2: $(dB)^2$ is determinant, not random, and its magnitude is dt i.e, the amount of change in $(dB)^2$caused by a change dt in the parameter is equal to dt.

To partially justify this statement we compute the expected value of $(B_{t+\Delta t} - B_t)^2$.

$$E\left[(B_{t+\Delta t} - B_t)^2\right] = Var\left[(B_{t+\Delta t} - B_t)\right] = \Delta$$

For the formulation of our mathematical model, we will require Ito's lemma in order to form the equation for stock prices. Any function that depends on the Brownian motion can be solved using Ito's Lemma

Ito's Lemma

We know that for a function f(x, y):

$$\mathrm{df} = \frac{\partial f}{\partial x}\mathrm{dx} + \frac{\partial f}{\partial y}\mathrm{dy} \quad (4)$$

Suppose that $\mathrm{f} = \mathrm{f}(\mathrm{t}, \mathrm{B}_t)$, where B_t denotes Brownian motion.

We expand df using Taylor's formula; this time keep the terms involving the second derivatives of f:

$$\mathrm{df} = \frac{\partial f}{\partial t}\mathrm{dt} + \frac{\partial f}{\partial B_t}\mathrm{dB}_t + \frac{1}{2} * \frac{\partial^2 \mathrm{f}}{\partial \mathrm{t}^2}(dt)^2 + \frac{\partial 2\mathrm{f}}{\partial \mathrm{t}\,\partial \mathrm{Bt}}(\mathrm{dt})(\mathrm{dB}_t) + 1/2\ \frac{\partial 2\mathrm{f}}{\partial B_t^2}(dB_t)^2 + \text{higher order terms (5)}$$

We ignore all terms involving dt to a power higher than 1. This gives us the following expression for df:

$$\mathrm{df} = \frac{\partial f}{\partial t}\ \mathrm{dt} + \frac{\partial f}{\partial Bt}\ \mathrm{dB}_t + \frac{1}{2}\left(\frac{\partial^2 \mathrm{f}}{\partial \mathrm{B}_t^2}\right)(dB_t)^2 \quad (6)$$

We next use the fact that $(dB)^2 = dt$ and rearrange to get:

$$\mathrm{df} = (\frac{\partial f}{\partial t} + \frac{1}{2}\frac{\partial^2 \mathrm{f}}{\partial \mathrm{B}_t^2})\mathrm{dt} + \frac{\partial f}{\partial B_t}\,\mathrm{dB}_t \quad (7)$$

Equation (7) is called Ito's lemma, and gives us the correct expression for calculating differentials of composite functions which depend on Brownian processes.

Relevance to the Problem Statement

Geometric Brownian motion implies that the change in the logarithm of the stock index value is a random variable.

Formulation of the logarithmic return of stock market index

In our stochastic process, the variable $t \in (0, T)$ is the time moment, and T is the total time, while $\Delta t = \frac{1}{T}$ being the time lag. Then the logarithmic return R(t) is given by the equation:

$$R(t) = ln(\frac{SI(t+\Delta t)}{SI(t)}) = lnSI(t+\Delta t) - lnSI(t) = \Delta lnSI \qquad (1)$$

A generalized stochastic process denoted by $(\mathrm{B(t)}, \mathrm{t} \geq 0)$ can be considered as a standard Brownian motion (BM) if it satisfies the following three conditions:

1) For any interval of time $0 \leq t_1 \leq t_2 \leq \ldots t_{n-1} \leq t_n$, the increments $\mathrm{B}(\mathrm{t}_2) - \mathrm{B}(\mathrm{t}_1), \ldots, \mathrm{B}(\mathrm{t}_n) - \mathrm{B}(\mathrm{t}_{n-1})$ are independent;

2) Each increment is a Gaussian random variable of zero mean and variance $\mathrm{t} - \mathrm{s}, \mathrm{so}: \mathrm{B(t)} - \mathrm{B(s)} \sim \mathrm{N}(0, \mathrm{t} - \mathrm{s}), \forall \mathrm{s} < \mathrm{t}$;

3) $\mathrm{B}(0) = 0$.

A Brownian motion {B(t), t ≥ 0} is a Markov process, where Markov process is defined below:
Given the present, a Markov process is a random process in which the future is independent of the past. As a result, Markov processes are the stochastic analogues of deterministic processes defined by differential and difference equations. They belong to one of the most significant categories of stochastic processes.

A real-valued stochastic process $(X_t)_{t\geq 0}$is a Markov process if for all Borel set $\mathrm{A} \subset \mathrm{R}$ and all real numberst $> \mathrm{s} > 0$, $P(X_t \in A | X_u, u \leq s) = P(X_t \in A | X_s)$.

We have used the following characteristics of Brownian Motion:

1) The trajectories of the motion are continuous but not derivable.

2) Stochastic processes$\{B(t), t \geq 0\}$, and $B^2(t) - t;\ t \geq 0$ are martingale, and, reciprocally, if SI(t) is a continuous process such that SI(t) and $(SI^2(t) - t,\ t \geq 0)$ are martingale, then SI(t) is a BM;
3) Stochastic processes $\{e^{[\lambda B(t) - \frac{1}{2}\lambda^2 t]}, t \geq 0\}$ is a martingale for any λ real and reciprocally. If $SI(t)$ is a continuous process such that $\{e^{[\lambda B(t) - \frac{1}{2}\lambda^2 t]}, t \geq 0\}$ is a martingale for any λ real, then $SI(t)$ is a Brownian motion.

Definition 3. A stochastic process $\{SI(t), t \geq 0\}$ follows a geometric Brownian motion (GBM) if the value of the stock index is described by the following dynamics equation

$$dSI(t) = \mu SI(t)dt + \sigma SI(t)dB(t) \quad ...(2)$$

where μ is the drift coefficient, σ is the diffusion coefficient, dB is the infinitesimal variation of a standard Brownian.

Using the stock index's dynamics equation $dSI(t) = \mu SI(t)dt + \sigma SI(t)dB(t)$ and applying the Itô lemma, it follows

$$dlnSI = \left(\mu - \frac{\sigma^2}{2}\right)dt + \sigma dB(t) \quad ...(3)$$

By integrating Equation (3), we obtain

$$lnSI(t) = lnSI(0) + lne^{\left[\int_0^t \left(\mu - \frac{\sigma^2}{2}\right)ds + \int_0^t \sigma dB(s)\right]} \quad ...(4)$$

where $SI(0)$ is the initial value of the stock market index and $SI(t)$ is the value of the stock market index at time t in the future.

Equation (4) can also be written as

$$SI(t) = SI(0) + e^{[\mu t - \frac{\sigma^2 t}{2} + \sigma B(t)]} \quad ...(5)$$

equivalent

$$ln\ \frac{SI(t)}{SI(0)} = \mu t - \frac{\sigma^2 t}{2} + \sigma B(t) \quad ...(6)$$

Or, over a time lag, the solution is

$$SI(t + \Delta t) = SI(t) + \exp\left\{\left(\mu - \frac{\sigma^2}{2}\right)\Delta t + \sigma[B(t + \Delta t) - B(t)]\right\} \quad ...(7)$$

where $B(t + \Delta t) - B(t) \sim N(0, \Delta t)$

Further, from Equation (7), the logarithmic return of the stock market index modelled by a GBM for a time lag (Δt) can be written as follows

$$\Delta lnSI = R(t) = \left(\mu - \frac{\sigma^2}{2}\right)\Delta t + \sigma[B(t + \Delta t) - B(t)] \quad ...(8)$$

with $t \geq 0$.

Moreover, given that $B(t + \Delta t) - B(t) \sim N(0, \Delta t)$ the logarithmic distribution of the stock index return modelled by a GBM can be written as follows:

$$R(t) \sim N\left[\left(\mu - \frac{\sigma^2}{2}\right)\Delta t, \sigma\sqrt{\Delta t}\right] \quad \ldots (9)$$

B(t) is a random function that gives random values for different t_s by the brownian function. It was not possible to conduct a numerical analysis using a random function. Hence, we have mimicked the randomness by using a noise function given by:

$$\frac{4\sin(100x) + 3\cos(150x) + 5\sin(170x) + 8\cos(190x) + 0.9}{50}$$

Algorithms Used

1. Dekker Method
2. Inverse Quadratic Interpolation Method

Dekker Method

Dekker's method is root finding algorithm combining the bisection method and the secant method. The algorithm tries to use the potentially fast converging secant method with the reliability of bisection method.

Dekker gave this in 1969

Initial input:

We want to solve the equation $f(x) = 0$

Assuming that f is continuous in $[a_0, b_0]$ and $f(a_0)$ and $f(b_0)$ have opposite signs.

Three points are used at every step for finding the root

1. b_k represents the current iterate and this is equal to x_r: current root of the f
2. a_k is other bound of the interval $[a_k, b_k]$ such that $f(a_k)$ and $f(b_k)$ have opposite signs and there is guaranteed a root in the interval $[a_k, b_k]$.
   Assumption: $|f(b_k| \leq |f(a_k|$ and the b_k is a better guess than a_k.
3. b_k represents the previous iteration and for initial iteration $b_{k-1} = a_0$

Iterative formula:

$$x_r = \begin{cases} b_k - \dfrac{b_k - b_{k-1}}{f(b_k) - f(b_{k-1})} f(b_k) & if\ f(b_k) \neq f(b_{k-1}) \\ m & else \end{cases}$$

The second solution is due to contribution of bisection method thus $m = \frac{a_k + b_k}{2}$

Finding the b_{k+1}:

$$b_{k+1} = \begin{cases} x_r & x_r\ lies\ strictly\ between\ b_k and\ m \\ m & else \end{cases}$$

Finding the new contra point:

$$a_{k+1} = \begin{cases} a_k & if\ f(a_k)\ and\ (b_{k+1})\ have\ opposite\ signs \\ b_k & else\ f(b_{k+1})\ and\ f(b_k)\ will\ have\ opposite\ signs \end{cases}$$

Finally putting constraint on a_{k+1} and b_{k+1}

If $|f(a_{k+1}| \leq |f(b_{k+1}|$ this means that a_{k+1} is the better guess thus we will swap the a_{k+1} and b_{k+1}
We will continue this until we get $|b_k - b_{k-1}| < tolerance$
Code for Dekker's Method:

```
from math import sin
from math import cos
from math import e

def w(x):
    y =(4*sin(100*x) + 3*cos(150*x) + 5*sin(170*x)+ 8*cos(190*x) +0.09)/50
    return abs(y)

#function for which we have to find the root
def f(x):
    # y = 7*e**(-x)*sin(x)-1
    y = e**(mu*x - (sigma*sigma*x/2) + sigma*w(x)) - k
    return y

#dekker method
def dekker(a,b,fun_tol,maxit):
    ak = a
    bk = b
    fak = f(ak)
    fbk = f(bk)
    iteration = 0;

    #checking whether the intial bounds are not the solution
    if abs(fbk) <= fun_tol:
        root = bk
        return bk,iteration

    elif abs(fak) <= fun_tol:
        root = ak
        return ak,iteration

    #assumption
    if abs(fak) < abs(fbk):
        ak,bk = bk,ak
        fak,fbk = fbk,fak


    #function has no root in this interval
    if fak*fbk >0:
```

```
        print("Stock Price Can't be doubled ")
            #the existence of root in such an interval is not guaranteed
        return 0,iteration

    #multiple iteration
    bk_1 = ak
    while abs(bk - bk_1) > fun_tol and iteration <maxit:
        fbk_1 = f(bk_1);
        m = (ak + bk)/2
        if fbk_1 != fbk: #secant method
            xr = bk - fbk*(bk - bk_1)/(fbk - fbk_1)
            if bk<m:
                if xr<m and xr>bk:
                    bk1 = xr
                else:
                    bk1 = m
            elif bk>m:
                if xr>m and xr<bk:
                    bk1 = xr
                else:
                    bk1 = m
        else: #bisection method
            xr = (ak + bk)/2
            bk1 = m

        iteration= iteration + 1
        fbk1 = f(bk1)
        fxr = f(xr)

        if abs(fxr) <= fun_tol:
            return xr,iteration
        if fak*fxr < 0:
            ak1 = ak
        else:
            ak1 = bk

        if abs(f(ak1)) < abs(f(bk1)):
            ak,bk = bk,ak
            fak,fbk = fbk,fak

        bk_1 = bk;
        bk = bk1;
        ak = ak1;
        fak = f(ak);
```

```
        fbk = f(bk1);
        if iteration >= maxit:
            return xr,iteration


#input ---------------------
mu = .006
sigma = 0.000012
k = 2
a=1
b = 4000
fun_tol = 1e-12;
maxit = 1000
#-----------------------------------

root,it_count = dekker(a,b,fun_tol,maxit)
print(root)
print(it_count)
```

Inverse Quadratic Interpolation Method

In numerical analysis, inverse quadratic interpolation is a root-finding algorithm, meaning that it is an algorithm for solving equations of the form $f(x) = 0$. The idea is to use Lagrange quadratic interpolation to approximate the $f(x)$.
The Methods requires three initial guesses x_0, $x_1$and x_2. Starting with initial values x_0, $x_1$and x_2, we construct a Lagrange interpolating polynomial of 2nd order of inverse of $f(x)$ using the initial points given.

$$f^{-1}(y) = \frac{(y - f(x_1))(y - f(x_2))(x_0)}{(f(x_0) - f(x_1))((f(x_0) - f(x_2))} + \frac{(y - f(x_0))(y - f(x_2))(x_1)}{(f(x_1) - f(x_0))((f(x_1) - f(x_2))} + \frac{(y - f(x_1))(y - f(x_0))(x_2)}{(f(x_2) - f(x_1))((f(x_2) - f(x_0))}$$

We put $y = 0$ in the Lagrange interpolating polynomial of 2nd order of inverse of $f(x)$ to get

$$x = \frac{(f(x_1)(f(x_2)(x_0)}{(f(x_0) - f(x_1))((f(x_0) - f(x_2))} + \frac{(f(x_0)(f(x_2)(x_1)}{(f(x_1) - f(x_0))((f(x_1) - f(x_2))} + \frac{(f(x_1)(f(x_0)(x_2)}{(f(x_2) - f(x_1))((f(x_2) - f(x_0))}$$

We then use this new value of x as x_3 and repeat the process, using x_1, x_2 and x_3 instead of x_0, x_1 and x_2. We continue this process, solving for x_5, x_6, etc., until we reach a sufficiently high level of precision ($abs(f(x_1)) < accuracy$.

Code for Inverse Quadratic Interpolation

```
mu=0.006; % drift
sigma=0.000012; % diffusion
n=100; % number of stocks
k=1.1; % the final value of stock is k times the initial value
f_x = @(x)exp(mu*x-
(sigma*sigma*x/2)+(sigma*(4*sin(100*x)+3*cos(150*x)+5*sin(170*x)+8*cos(190*x)+0.9)))/50
-k;
x_0=1000; % initial guess 1
x_1=2000; % initial guess 2
x_2=3000; % initial guess 3
accuracy=0.00001; % to converge the method
inverse_quadratic_method(x_0,x_1,x_2,f_x,accuracy);


function[Root]=inverse_quadratic_method(x_0,x_1,x_2,f_x,accuracy)
while true
  count=count+1;
  if count>1000 % number of iterations becomes large
    fprintf('The method doesnt converge')
    break
  elseif f_x(x_1)==0 % x_1 is already the root
      Root=x_1;
      fprintf('Number of iterations are %f\n',count)
      fprintf('Root is %f\n',Root)
      break
  elseif f_x(x_2)==0 % x_2 is already the root
      Root=x_2;
      fprintf('Number of iterations are %f\n',count)
      fprintf('Root is %f\n',Root)
      break
  elseif f_x(x_0)==0 % x_0 is already the root
      Root=x_0;
      fprintf('Number of iterations are %f\n',count)
      fprintf('Root is %f\n',Root)
      break
  elseif f_x(x_1)==f_x(x_2) % denominator in lagrange interpolation becomes zero
      fprintf('The denominator becomes zero. Try other initial guesses')
      break
  elseif f_x(x_0)==f_x(x_2) % denominator in lagrange interpolation becomes zero
      fprintf('The denominator becomes zero. Try other initial guesses')
      break
  elseif f_x(x_0)==f_x(x_1) % denominator in lagrange interpolation becomes zero
      fprintf('The denominator becomes zero. Try other initial guesses')
      break
```

```
  else
    if abs(f_x(x_2))<accuracy % condition for convergence of root
       Root=x_2;
       fprintf('Number of iterations are %f\n',count)
       fprintf('Root is %f\n',Root)
       break
    elseif abs(f_x(x_1))<accuracy % condition for convergence of root
       Root=x_1;
       fprintf('Number of iterations are %f\n',count)
       fprintf('Root is %f\n',Root)
       break
    elseif abs(f_x(x_0))<accuracy % condition for convergence of root
       Root=x_0;
       fprintf('Number of iterations are %f\n',count)
       fprintf('Root is %f\n',Root)
       break
    else
       % hardcoding the inverse lagrange interpolating polynomial
       t=x_2;
       x_2=(f_x(x_2)*f_x(x_1)*x_0)/((f_x(x_0)-f_x(x_1))*(f_x(x_0)-
f_x(x_2)))+(f_x(x_2)*f_x(x_0)*x_1)/((f_x(x_1)-f_x(x_2))*(f_x(x_1)-
f_x(x_0)))+(f_x(x_0)*f_x(x_1)*x_2)/((f_x(x_2)-f_x(x_1))*(f_x(x_2)-f_x(x_0)));
       x_0=x_1;
       x_1=t;
    end
  end
end
end
```

Result and Graphical Explanation

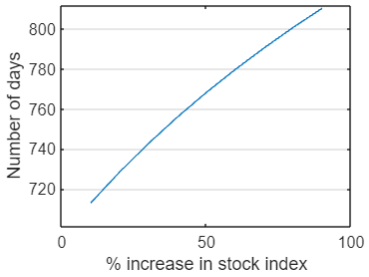
After using numerical analysis by the given methods, we plotted graphs to see the trend of how much time the stock index takes to grow.

As expected for both the methods we get the same graph.

For the sake of plotting time against growth value of stock index we have put in the values of the drift coefficient as 0.006 and that of the diffusion coefficient as 0.000012

For these values, the graph is almost linear as the volatility (defined as $\mu - \frac{\sigma^2}{2}$) is of a very small value.

In this way we estimated the time when our stock index will hit various value checkpoints like a 10 percent growth or a 100 percent growth.



Acknowledgments

We, as a team, would like to extend our gratitude to all the three instructors of MA202 who gave us this opportunity to work on such an exciting topic. We would also like to thank Prof Satyajit Pramanik, Prof Uddipta Ghosh, and Prof Uddipta Ghosh for providing guidance and assistance.

References

[1]https://www.diva-portal.org/smash/get/diva2:1218088/FULLTEXT01.pdf

[2]https://ecommons.udayton.edu/cgi/viewcontent.cgi?article=1010&context=mth_epumd

[3]https://www.math.tamu.edu/~stecher/425/Sp12/brownianMotion.pdf

[4]http://reports.ias.ac.in/report/18641/implementation-of-brent-dekker-and-a-better-root-finding-method-and-brent-dekker-methods-parallelization

[5]https://www.researchgate.net/publication/237429557_NEW_INVERSE_INTERPOLATION_METHODS