The Brownian Motion

Nguyen Thanh Phuong and Van Tien Duc

Ho Chi Minh city University of Technology
Faculty of Computer Science and Engineering

Abstract. This report is the note that we have learned about the Brownian motion for the last two weeks. The report includes a brief of definition of Stochastic processes, and some form of the Brownian motion.

Keywords: Stochastic process · Brownian motion · Wiener process.

1 Generalities of Stochastic Processes

In its most general expression, a stochastic process is simply a collection of random variables $\{X_t, t \in I\}$. The index t represents for the time, and the set I is the index set of the process. In discrete time case, the index set is usually $I = \{0, 1, 2, ...\}$, and in continuous time case, $I = [0, \infty)$.

Definition. Let $\{\Omega, \mathcal{A}, \mathcal{P}\}$ be a probability space. A real-valued stochastic process is a collection of random variables $\{X_t, t \geq 0\}$ that defines on that sample space, and $X_t : \Omega \to \mathbb{R}$.

Example. (Random walk and Gambler's ruin) A random walker starts at the origin on the integer line. At each discrete unit of time the walker moves either right or left, with respective probabilities p and $1 - p$.This describes a *simple random walk* in one dimension. The stochastic process is built as follows. Let $X_1, X_2...$ be a sequence of i.i.d. random variables with

$$\begin{aligned} X_t &= +1, \qquad \textit{with probability } p, \\ &= -1, \qquad \textit{with probability } 1-p \end{aligned}$$

for $t \geq 1$. Set

$$S_n = \sum_{i=1}^{n} X_i, \quad for\ n \geq 1,$$

with $S_0 = 0$. Then, S_n is the random walk's position after n steps.

Example. (Monopoly) The popular board game *Monopoly* can be modeled as a stochastic process. Let $X_0, X_1, X_2...$ represent the successive board positions of an individual player. That is, X_k is the players board position after k plays.
The sample space is $\{1, ..., 40\}$ denoting the 40 squares of a Monopoly boardfrom Go to Boardwalk. The index set is $\{0, 1, 2, ...\}$. Both the index set and sample space are discrete.

1.1 Filtration

Consider a probability space $\{\Omega, \mathcal{A}, \mathcal{P}\}$. A *filtration* $\{\mathcal{F}_t, t \geq 0\}$ is an increasing family of sub-σ-algebras of $\mathcal{A}$ indexed by $t \geq 0$ such that

- if $s < t$, then $\mathcal{F}_s \subset \mathcal{F}_t$.
- $\mathcal{F}_0 = \{\Omega, \emptyset\}$.
- $\mathcal{F}_t = \sigma(X_s; 0 \leq s \leq t)$. This σ-algebra is the smallest set of subsets of Ω that makes it possible to assign probabilities to all the events related to the process X up to time t.

1.2 Conditional Expectation

We start with continuous case. Let X, Y be continuous random variables. We define the conditional density of X given Y to be

$$f_{X|Y}(x|y) = \frac{f_{X,Y}(x,y)}{f_Y(y)}$$

Then

$$P(a \leq x \leq b|Y = y) = \int_a^b f_{X|Y}(x|y)dx$$

Conditioning on Y=y is conditioning on an event with probability zero. This is not defined, so $f_Y(y) > 0$.
We then define the conditional expectation of X given $Y = y$ to be

$$\mathbb{E}[X|Y = y] = \int_{-\infty}^{+\infty} x f_{X|Y}(x|y)dx$$

Now we review the discrete case. Let X, Y be discrete random variables. We define the conditional mass probability of X given Y to be

$$P(X = x|Y = y) = \frac{P(X = x, Y = y)}{P(Y = y)}, \qquad (P(Y = y) > 0)$$

Then we define the conditional expectation of X given Y=y to be

$$\mathbb{E}[X|Y = y] = \sum_x xP(X = x|Y = y)$$

Example. Suppose that X and Y are independent geometric random variables with parameter p. Suppose that $n > 2$ is a positive integer. Find the conditional mass function of X given $X+Y = n$ and the conditional expectation of X given $X + Y = n$.

Solution. First, we need to compute the mass probability of $X + Y = n$

$$\begin{aligned}
P(X + Y = n) &= \sum_{k=1}^{n-1} P(X = k, Y = n - k) \\
&= \sum_{k=1}^{n-1} P(X = k)P(Y = n - k) \qquad (\textit{Since } X \textit{ and } Y \textit{ are independent}) \\
&= \sum_{k=1}^{n-1} (1 - p)^{k-1} p(1 - p)^{n-k-1} p \\
&= p^2(n - 1)(1 - p)^{n-2}
\end{aligned}$$

Thus, the conditional mass function will be computed as below

$$\begin{aligned}
P(X|X + Y = n) &= \frac{P(X, X + Y = n)}{P(X + Y = n)} \\
&= \frac{P(X = k)P(Y = n - k)}{P(X + Y = n)} \\
&= \frac{p^2(1 - p)^{n-2}}{p^2(n - 1)(1 - p)^{n-2}} \\
&= \frac{1}{n - 1}
\end{aligned}$$

And finally, we compute the conditional expectation of X given $X+Y=n$ as the following

$$\begin{aligned}\mathbb{E}[X|X+Y=n] &= \sum_{k=1}^{n-1} kP(X|X+Y=n)\\ &= \frac{1}{n-1}\sum_{k=1}^{n-1} k\\ &= \frac{1}{n-1}.\frac{n(n-1)}{2}\\ &= \frac{n}{2}\end{aligned}$$

1.3 Martingale

Given a probability space $\{\Omega, \mathcal{A}, \mathcal{P}\}$ and a filtration $\{\mathcal{F}_t, t \geq 0\}$ on $\mathcal{A}$, a martingale is a stochastic process $\{X_t, t \geq 0\}$ such that $\mathbb{E}[X_t] < \infty$ for all $t > 0$, it is adapted to a filtration $\{\mathcal{F}_t, t \geq 0\}$, for each $0 \leq s \leq t < \infty$, it holds true that

$$\mathbb{E}[X_t|\mathcal{F}_s] = X_s$$

Example. A gambler's fortune is a martingale if all the betting games which the gambler plays are fair. To be more specific: suppose X_n is a gambler's fortune after n tosses of a fair coin, where the gambler wins \$1 if the coin comes up heads and loses \$1 if it's tails. The gambler's conditional expected fortune after the next trial, given the history, is equal to his present fortune. This sequence is thus a martingale.

2 The Brownian Motion

Brownian motion is a continuous time, continuous sample space stochastic process. The name also refers to a physical process, first studied by the botanist Robert Brown in 1827. Brown observed the seemingly erratic, zigzag motion of tiny particles ejected from pollen grains suspended in water. He gave a detailed study of the phenomenon but could not explain its cause. In 1905, Albert Einstein showed that the motion was the result of water molecules bombarding the particles. We can see a 2D simulation of the Brownian motion in **Figure 1**

2.1 Definition

The Brownian motion is also called the Wiener process. There are several ways to characterize and define the Wiener process $W = \{W(t), t \geq 0\}$, and one is following:

- It is a Gaussian process with continuous paths.
- $W(0) = 0$ with probability 1.
- $W(t) - W(s) \stackrel{d}{=} \mathcal{N}(0, t-s)$, for $0 \leq s \leq t < \infty$,
- Stationary independent increment.

Simulation of the trajectory of the Brownian motion
Given a fixed time increment $t > 0$, one can easily simulate a trajectory of the Wiener process in the time interval $[0, T]$. Indeed, for $W_{\Delta t}$ it holds true that

$$W(\Delta t) = W(\Delta t) - W(0) \stackrel{d}{=} \mathcal{N}(0, \Delta t) \stackrel{d}{=} \sqrt{\Delta t}\, \mathcal{N}(0,1)$$

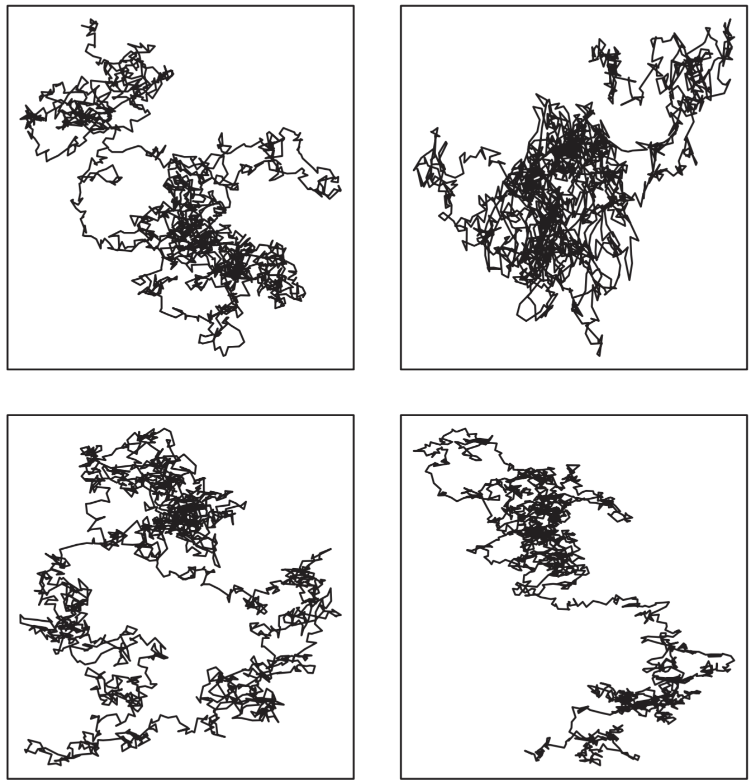

Fig. 1. Simulations of two-dimensional Brownian motion [2]

and the same is also true for any other increment $W(t+\Delta t)-W(t)$; i.e.,

$$W(t+\Delta t)-W(t) \stackrel{d}{=} \mathcal{N}(0,\Delta t) \stackrel{d}{=} \sqrt{\Delta t}\,\mathcal{N}(0,1)$$

The following is the code of simulation of the Wiener process

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

m=5
n=300
# Initializing time
t=np.linspace(0,1,n+1)
# Getting time step
h=np.diff(t[0:2])
# Generating random values over time
dw=np.sqrt(h) * np.random.randn(n,m)
# Initializing zeros at the time t=0
zeros = np.zeros((1,m))
# Creating processes
w_t = np.r_[zeros,dw]
w = np.cumsum(w_t, axis=0)
```

```
plt.figure(figsize=(10,6))
plt.plot(t,w)
plt.xlabel('t')
plt.ylabel('W')
plt.title('The graph of 5 Wiener processes')
plt.savefig('Wiener Process.png')
plt.show()
```

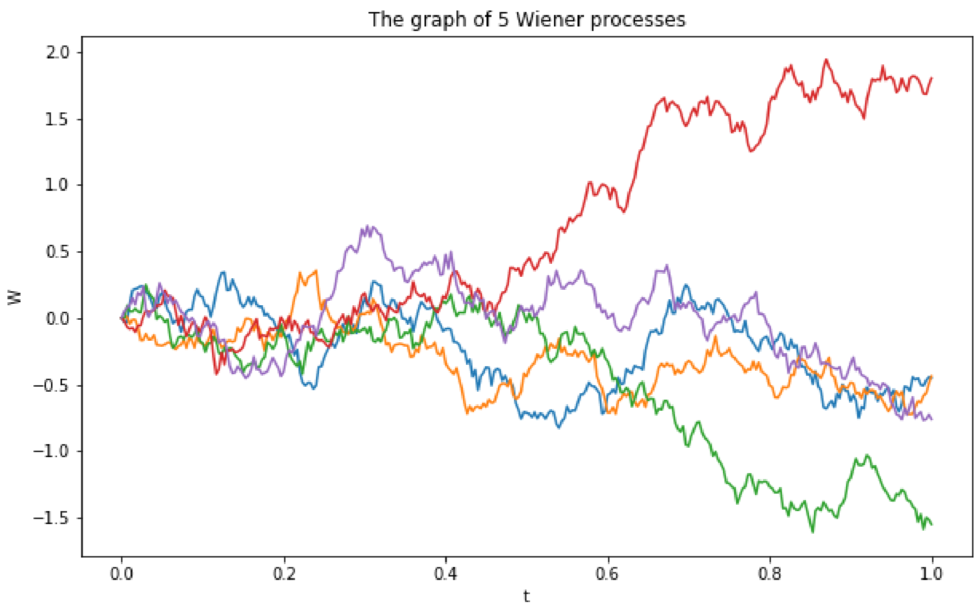

Fig. 2. The simulation of Wiener process.

2.2 The Brownian Motion as the Limit of a Random Walk

One characterization of the Brownian motion says that it can be seen as the limit of a random walk in the following sense. Given a sequence of i.i.d. random variables $X_1, X_2, ..., X_n$, taking only two values -1 and +1 with equal probability and considering the partial sum,

$$S_n = \sum_{i=1}^{n} X_i$$

Then, as $n \to \infty$

$$P\left(\frac{S_{[nt]}}{\sqrt{n}} < x\right) \to P(W(t) < x)$$

Simulation of the trajectory of the Random walk

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

n=[10,100,1000] # number of times
m=1 # number of processes
T = 6 # the end time
t = [np.linspace(0,T,x+1) for x in n]
# the sequence of random variables taking values -1 and 1 with equal probability
X_t = [2*(np.random.rand(x,m) > 0.5)-1 for x in n]
# The Wiener process
S_n = [np.cumsum(x_t, axis=0) for x_t in X_t]
W_t = S_n/np.sqrt(n)
# The start W(0)
zeros = np.zeros((1,m))
W_t = [np.r_[zeros,w_t] for w_t in W_t]

plt.figure(figsize=(16,6))
plt.plot(t[0],W_t[0])
plt.plot(t[1],W_t[1])
plt.plot(t[2],W_t[2])
plt.legend(['n=10','n=100','n=1000'])
plt.xlabel('t')
plt.ylabel('W')
plt.title('Simulation of Random walk')
plt.savefig('Random walk.png')
plt.show()
```

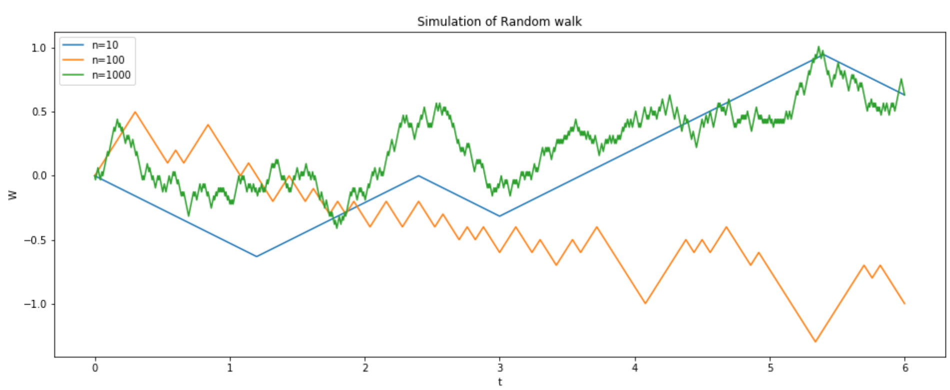

Fig. 3. The simulation of Random walk.

2.3 Geometric Brownian Motion

A process used quite often in finance to model the dynamics of some asset is the so-called *geometric Brownian motion*. This process has the property of having independent multiplicative increments and is defined as a function of the standard Brownian motion

$$S(t) = S(0)exp\left\{\left(r - \frac{\sigma^2}{2}\right)t + \sigma W(t)\right\}, t > 0,$$

Simulation of the trajectory of Geometric Brownian motion

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

r = 1
sigma = 0.5
n = 1000
T = 1
S0 = 10
Delta = T/n # step time

Z = np.random.normal(0,1,n) * np.sqrt(Delta) # the sequence of random variables
t = np.linspace(0,T,n)
W_t = np.cumsum(Z) # the Wiener process

S_t = S0 * np.exp((r-np.power(sigma,2)/2)*t+sigma*W_t)

plt.figure(figsize=(16,6))
plt.plot(t,S_t)
plt.xlabel('t')
plt.ylabel('W')
plt.title("Geometric Brownian Motion")
plt.savefig('Geometric_Brownian_Motion.png')
plt.show()
```

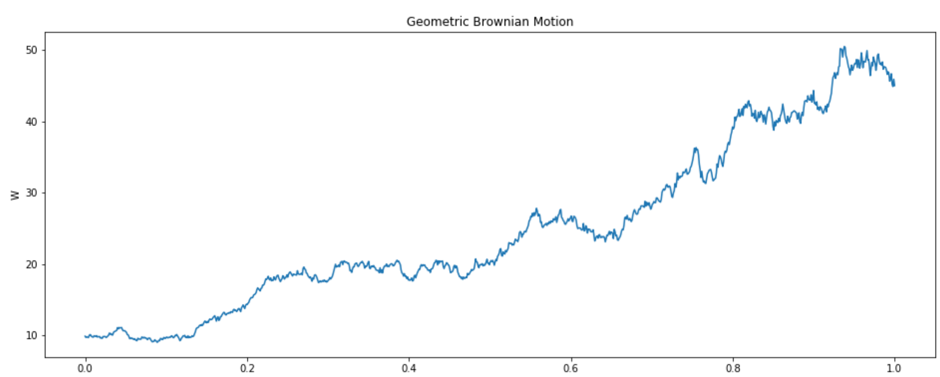

Fig. 4. The simulation of Geometric Brownian Motion

2.4 Brownian Bridge

Another useful and interesting manipulation of the Wiener process is so-called Brownian Bridge, which is the Brownian motion start at x at time t_0 and passing through some point y at time T, $T > 0$. It is defined as:

$$W_{t_0,x}^{T,y}(t) = x + W(t - t_0) - \frac{t - t_0}{T - t_0}(W(T - t_0) - y + x)$$

More precisely, this is the process $\{W(t), t_0 < t < T | W(t_0) = x, W(T) = y\}$
Simulation of the trajectory of Brownian bridge

```
import numpy as np
import matplotlib.pyplot as plt
n = 100
T = 1
Delta = T/n
Z = np.random.normal(0,1,n) * np.sqrt(Delta)
W_t = np.cumsum(Z)
W_t = np.insert(W_t,0,0)
t = np.linspace(0,T,n+1)
x = 0
y = -1
BB = x + W_t - (t/T)*(W_t[-1]-y+x)
plt.figure(figsize=(16,6))
plt.plot(t,BB)
plt.hlines(-1,xmin=0,xmax=1,linestyle='dashed')
plt.xlabel('t')
plt.ylabel('W')
plt.title("Brownian Bridge")
plt.savefig('Brownian_Bridge.png')
plt.show()
```

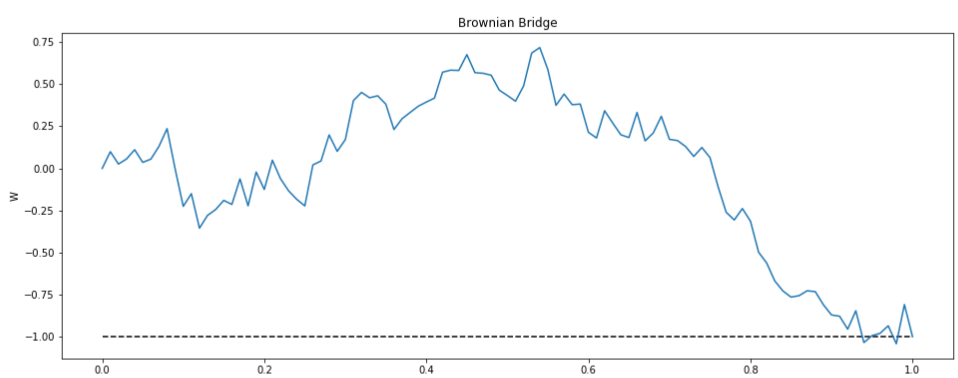

Fig. 5. The simulation of Brownian Bridge.

References

1. Stefano M. lacus, Simulation and Inference for Stochastic Differential Equations with R examples, 2008
2. Robert P. Dobrow, Introduction to Stochastic Processes with R, 2016