Assignment 6

Ko-Cheng Chang

October 2020

1 Introduction

In this lab, we are trying to modeling stochastic process, test how in the change of steps (h_i) will effect our Brownian motion and sample path of X, and calculate strong and weak error in this model via Monte Carlo simulation.

2 Assignment 6(Q1)

2.1 Problem

In first question, we modeling Brownian motion on different resolutions($h_i, i = 1, 2, ..10$) with same noise, which means same normal random variables.

2.2 Theory and implementation

To compute this, first, we need to compute increments in

$$h_i = 2^{-10}, i = 10$$

and use grids we obtain to compute other grids in other resolution.

$$\tilde{\eta_1} = \eta_1 + \eta_2$$

2.3 Results and discussion

From figure 1. , we find out the same noise give us some same result in the end and between the path, however ,the path is not similar overall.

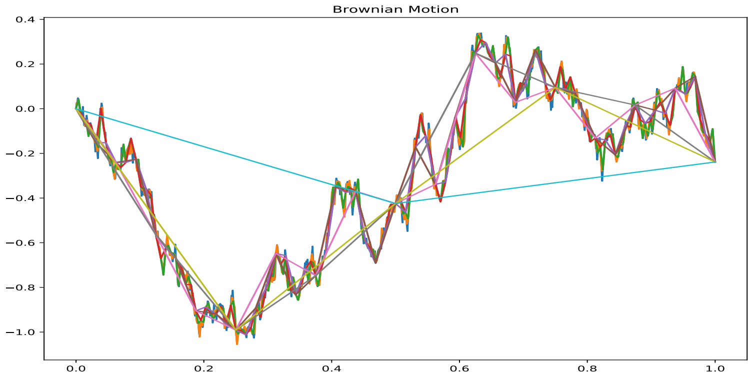

Figure 1: This shows the path of Brownian Motion with same noise and different resolution

3 Assignment 6(Q2)

3.1 Problem

Base on Q1, we now simulate a sample path of X .

3.2 Theory and implementation

We compute path of $X(t)$ through approximation given by the recursion formula

$$X_h(t_n) = (1 + h\mu)X_h(t_{n-1}) + \sigma X_h(t_{n-1})(W(t_n - W(t_{n-1})$$

3.3 Results and discussion

Although we obtain same ending point in $W(1)$ among different i, it's obviously that the path of X is significant different, the reason for this is because in our formula, $(\sigma X_h(t_{n-1})(W(t_n - W(t_{n-1}))$ part, in more resolution formula, this part is more precisely with smaller step, in less resolution ones, smaller step all gather into one big step, therefore, the product of $(\sigma X_h(t_{n-1})(W(t_n$ will differ.

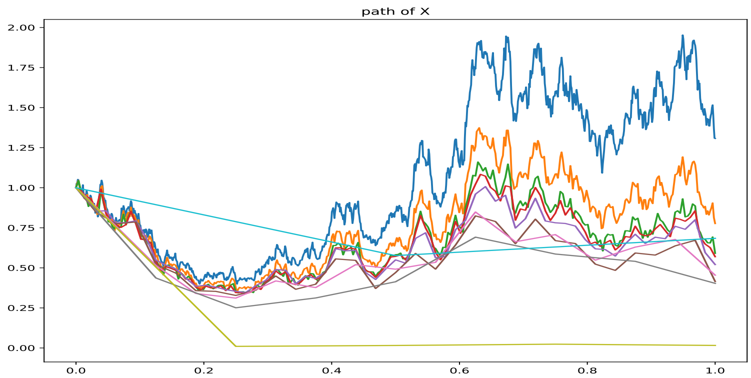

Figure 2: This shows the path of path of X_h(which we treat as approximation as X)with same noise and different resolution

4 Assignment 6(Q3)

4.1 Problem

In Q3, we use Monte Carlo simulation($M = 5000$) to compute strong error with $X(t)$ and $X_h(t)$.

4.2 Theory and implementation

The formula to compute strong error is

$$\frac{1}{M}\sum_{m=1}^{M}((X(T)^{(m)} - X_h(T)^{(m)})^2)^{\frac{1}{2}}$$

From the question, we know that $T = 1$, therefore, we can replace T with 1 and compute the formula.

From the definition, we know that when we have infinite N the error will be 0, however,it's hard to compute very large number, therefore, we are interesting in the convergence in this model when i increasing.

4.3 Results and discussion

In the end we obtain figure below, the figure is in loglog scale, in the x-axis it represent h in log-scale and y-axis is strong error in log-scale, above this, we add a reference slope$h^{\frac{1}{2}}$, the figure shows that the convergence is near $\frac{1}{2}$, which is what we expect.

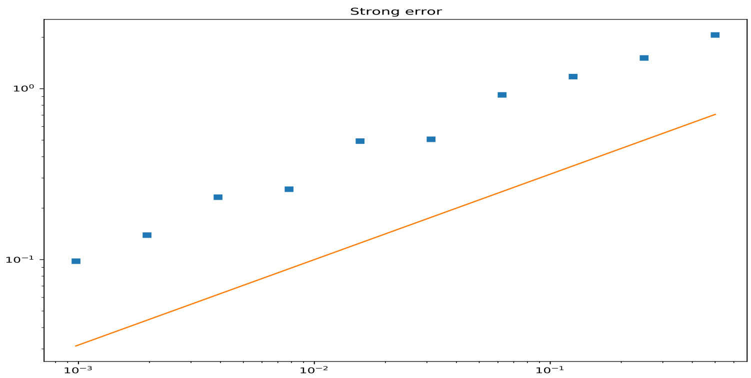

Figure 3: It shows the convergence with strong error when we increasing i

5 Assignment 6(Q4)

5.1 Problem

In this question we compute, another error, which is weak error, with Monte Carlo. simulation($M = 5000$).

5.2 Theory and implementation

The formula to compute weak error given by question is, which imply test function is 1.

$$| \mathbb{E}[X(T) - X_h(T)] |$$

From the question, we know that $T = 1$, therefore, we can replace T with 1 and compute the expectation with Monte Carlo method.

Therefore, the formula to compute weak error is

$$| \mathbb{E}[X(1)] - \frac{1}{M} \sum_{m=1}^{M} (X_h(1))^{(m)} |$$

We have $\mathbb{E}[X(1)] = exp(\mu)$ from the question, therefore , we can compute following figure.

5.3 Results and discussion

The figure, shows that the convergence is not that steady as we using strong error, but it still tell us that there exists convergence when increasing i.

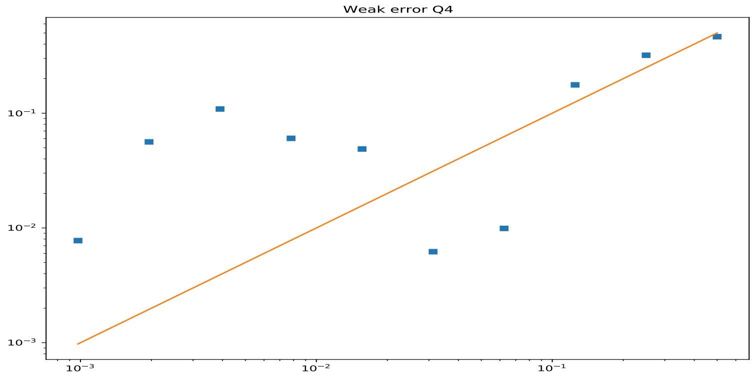

Figure 4: It shows the convergence in weak error when we increasing i

6 Assignment 6(Q5)

6.1 Problem

In this question we compute still compute weak error but with another test function.

6.2 Theory and implementation

The formula to compute weak error is

$$\mid \mathbb{E}[\phi X(T) - \phi X_h(T)] \mid$$

Therefore, the formula to compute weak error is

$$\mid \mathbb{E}[\phi X(1)] - \frac{1}{M} \sum_{m=1}^{M} (\phi X_h(1))^{(m)} \mid$$

Same as previous question, we set $T = 1$.

6.3 Results and discussion

I try to set $\phi(x) = x^2$, although I can obtain data from $\mid \frac{1}{M} \sum_{m=1}^{M} (\phi X_h(1))^{(m)} \mid$, as long as I cannot compute $\mid \mathbb{E}[\phi X(1)] \mid$, it is impossible to obtain real number in weak error.

Appendix - code

```
import numpy as np
import matplotlib.pyplot as plt
X0 = 1
mu = 1
sigma = 1
sigma2 = sigma**2
seed = 518


# Q1

N  = 2**(10)
h = 1/N
np.random.seed(seed)
inc_b = np.random.normal(0, 1, int(N))*np.sqrt(h) #increments
mot_b = np.cumsum(inc_b)  # brownian motion
mot_b = np.insert(mot_b, 0, 0) #adding W(0) = 0
xw = np.linspace(0, 1, len(mot_b))
plt.plot(xw, mot_b)
for o in range(9):
    inc_b = [sum(inc_b[i:i+2]) for i in range(0, len(inc_b), 2)]
    mot_b = np.cumsum(inc_b)
    mot_b = np.insert(mot_b, 0, 0) #adding W(0) = 0
    xw = np.linspace(0, 1, len(mot_b))
    plt.plot(xw, mot_b)
plt.title('Brownian Motion')
plt.savefig('Brownian Motion.pdf')


#Q2


N  = 2**(10)
h = 1/N
np.random.seed(seed)
inc_b = np.random.normal(0, 1, int(N))*np.sqrt(h) #increments
mot_b = np.cumsum(inc_b)  # brownian motion
mot_b = np.insert(mot_b, 0, 0) #adding W(0) = 0
x_h = [1]
for i in range(len(inc_b)):
    x_temp = (1+h*mu)*x_h[i] + sigma * x_h[i]*(inc_b[i])
    x_h.append(x_temp)
xw = np.linspace(0,1,len(x_h))
plt.plot(xw, x_h)
for o in range(9):
    inc_b = [sum(inc_b[i:i+2]) for i in range(0, len(inc_b), 2)]
    mot_b = np.cumsum(inc_b)  # brownian motion
    mot_b = np.insert(mot_b, 0, 0) #adding W(0) = 0
    x_h = [1]
```

```
    for ii in range(len(inc_b)):
        x_temp = (1+h*mu)*x_h[ii] + sigma * x_h[ii]*(mot_b[ii+1]-mot_b[ii])
        x_h.append(x_temp)
    xw = np.linspace(0,1,len(x_h))
    plt.plot(xw, x_h)
plt.title('path of X')
plt.savefig('path of X.pdf')


#Q3
err = []
for o in range(10):
    str_err = []
    for ii in range(5000):# Monte Carlo method
        N  = 2**(o+1)
        h = 1/N
        inc_b = np.random.normal(0, 1, int(N))*np.sqrt(h) #increments
        mot_b = np.cumsum(inc_b) # brownian motion
        mot_b = np.insert(mot_b, 0, 0)
        x_h = [1]
        for i in range(int(N)):
          x_temp = (1+h*mu)*x_h[i] + sigma*x_h[i]*(mot_b[i+1]-mot_b[i])
          x_h.append(x_temp)
        xh_1 = x_h[-1]
        x_1 = np.exp((mu - (sigma2/2))*1 + sigma*mot_b[-1])
        temp = (x_1 - xh_1)**2
        str_err.append(temp)
    str_err = np.mean(str_err)**(1/2)
    err.append(str_err) # gen list of strong error based on different i
h = []
for i in range(10):
    N = 2**(i+1)
    h.append(1/N)
plt.loglog(h,err,'s')
plt.loglog(h, np.sqrt(h))
plt.title('Strong error')
plt.savefig('Strong error.pdf')

#Q4
err = []
for o in range(10):
    weak_err = []
    for ii in range(5000): # Monte Carlo method
        N  = 2**(o+1)
        h = 1/N
        inc_b = np.random.normal(0, 1, int(N))*np.sqrt(h) #increments
        x_h = [1]
        for i in range(int(N)):
          x_temp = (1+h*mu)*x_h[i] + sigma*x_h[i]*inc_b[i]
          x_h.append(x_temp)
        xh_1 = x_h[-1]
```

```
        weak_err.append(xh_1)
    weak_err = np.mean(weak_err)
    err.append(np.abs(np.exp(mu) - weak_err)) # gen list of weak error based on d

h = []
for i in range(10):
     N = 2**(i+1)
     h.append(1/N)
plt.loglog(h,err,'s')
plt.loglog(h, h)
plt.title('Weak error Q4')
plt.savefig('Weak error Q4.pdf')


#Q5
err = []
for o in range(10):
    weak_err = []
    mot_b = []
    for ii in range(5000):# Monte Carlo method
        N  = 2**(o+1)
        h = 1/N
        inc_b = np.random.normal(0, 1, int(N))*np.sqrt(h) #increments
        x_h = [1]
        mot_b.append(np.sum(inc_b))
        for i in range(int(N)):
          x_temp = (1+h*mu)*x_h[i] + sigma*x_h[i]*inc_b[i]
          x_h.append(x_temp)
        xh_1 = x_h[-1]**2
        weak_err.append(xh_1)
```