448 HW4 Report

Mingfu Liang, Student ID:3146919, NetID:MLQ4767

March 2019

1. Problem a

In problem a we simulate the Brownian ratchet with $dF = \sigma dW$ with W the Wiener process. First we need to determine how small the Δt we should choose at this question. The intuition is that we don't want the Δt too large since it will cut down the time step size and cannot represent the behavior of $X(t)$ satisfactorily. We also don't want the Δt to be too small since it will be time-wasting and memory-wasting to generate some much point. Therefore, the perfect Δt should be compared to the Δt with a factor of 2, they are quite close to each other and all the points are almost touch with each others. For example, as shown in Figure 1, we can see that when $\Delta t = 0.061$, it satisfy the requirement, so I will choose this one to complete part a.

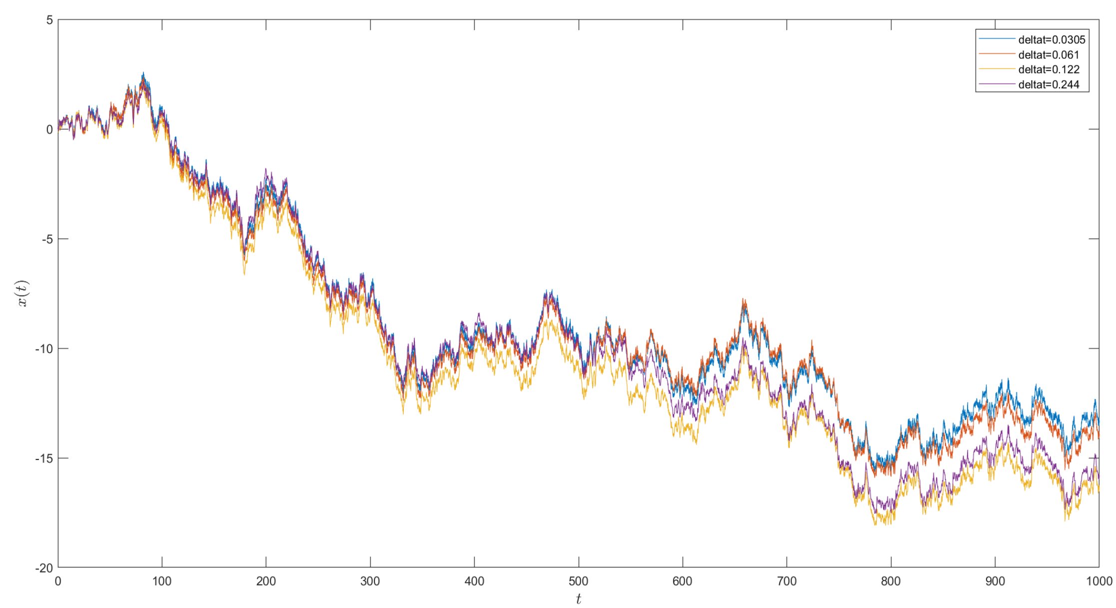

Figure 1: Determine the δ_t

1.1 Problem a(i)

In Figure 2 there are several trajectories $X(t)$ for $\sigma = 1$ from $M_{trials} = 1000$

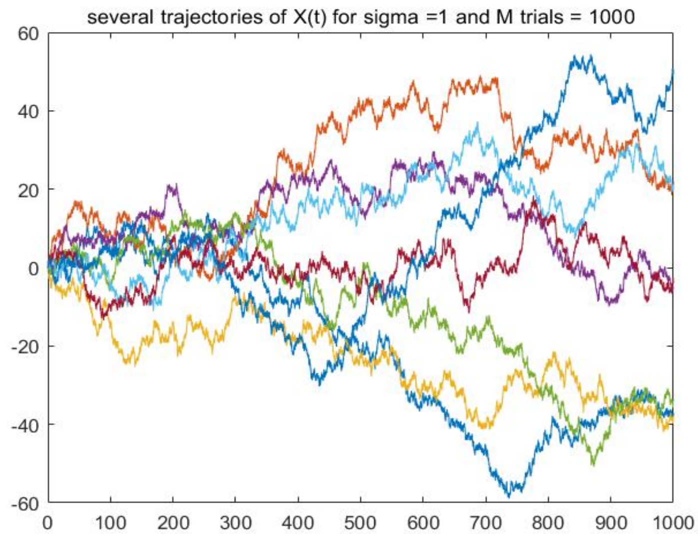

Figure 2: Several trajectories $X(t)$ for $\sigma = 1$

From Figure 3 and 4 is the expectation value $< X(t) >$ of $X(t)$ and the standard derivation of $X(t)$. Actually each time we run the code, the expectation value of $< X(t) >$ will change, but the shape of the standard derivation of $X(t)$ would be stable and quite similar each run. And actually from the Figure 5 that when M_{trials} is increasing, the shape of the standard derivation of $X(t)$ will be smoother.

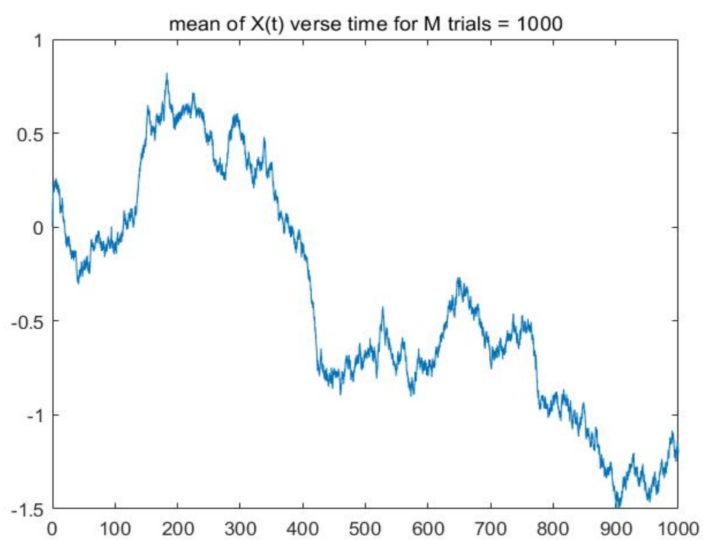

Figure 3: Expectation value $< X(t) >$ of $X(t)$

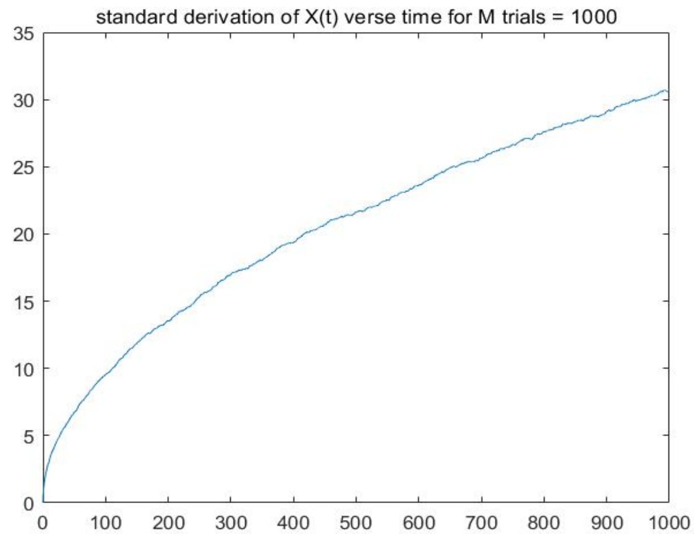

Figure 4: Standard deviation of $X(t)$

1.2 Problem a(ii)

Here I am going to compute the standard error of the mean $< X(T) > /T$ for $T = 1000$ and $\sigma = 1$. Here I first define the formula of the standard error of the mean as:

$$\sqrt{\frac{\sum_{i}^{M}((X_i(t)/T) - < (X_i(t)/T) >)^2}{M * (M-1)}} \tag{1}$$

where $X_i(t)$ denotes ith trials of realization, M denotes the total size of realization and $< X_i(t) >$ denote the mean of $X_i(t)$. First I take a try of $M = 32$ and I find that the standard error of the mean is not less than 0.005, which is 0.00532. I tried this M many times and I found that it can not get lower than 0.005, Then I try to increase the M with a factor of 2 so that $M = 64$ and I found that the standard error of mean is 0.0037264, which is smaller than 0.005, and it can sufficiently smaller than 0.005. Then since based on the equation 1, it can be divide into two part actually, one part is the estimation of the standard variance, which is $\sqrt{\frac{\sum_{i}^{M}((X_i(t)/T) - <(X_i(t)/T)>)^2}{(M-1)}}$, the other part is $\sqrt{\frac{1}{M}}$. And since when the M becomes larger and larger, the estimation of standard variance would be kind of constant, and then the standard error of the mean will be proportional to the $\sqrt{\frac{1}{M}}$. Since we want to estimate how many trials would be needed to obtain a standard error of the mean of 0.0001, which is smaller than 0.005 with a factor of 50, therefore I calculate the trials and it seems to be 1600000. Then I run that many trials to verify my thought and I found that the standard error of the mean is $7.7423 * 10^{-5}$, which is sufficiently smaller than 0.0001. Actually, I have also tried 90000, 100000, 110000, 120000, 130000, 140000 and I found that the decrease of standard error of the mean is quite small. Therefore, I also conclude that the decrease of the standard error of the mean will first drop down quickly as M increasing and then drop down very slow when you increase the M after a sufficiently large M.

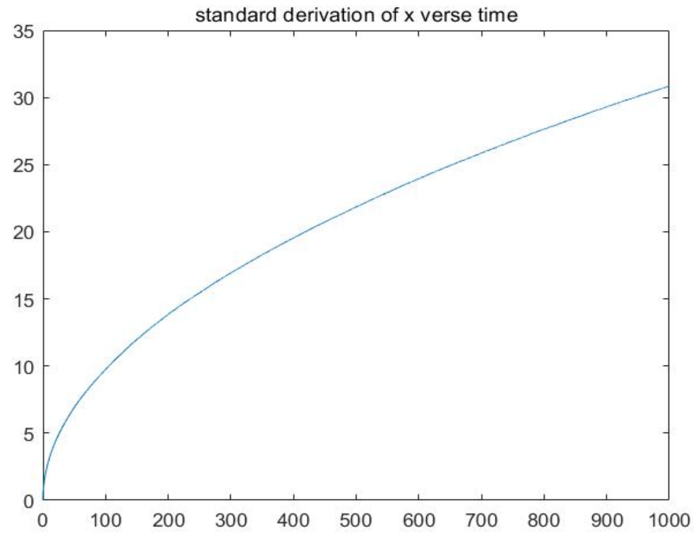

Figure 5: The standard deviation of $X(t)$ when $M_{trials} = 1600000$

1.3 Problem a(iii)

For $\sigma = 0.2$ and $T = 2000$ and $M = 100$ I obtain a histogram of the potential energy $V(X)$ for all times t after a transient time of $t_T = 100$. As shown in Figure (6), this is the histogram on a linlog scale and I find that it decrease exponentially. By observation, I think it quite related to the Boltzmann Distribution in an equilibrium thermal system but it is not easy to proof it mathematically :)

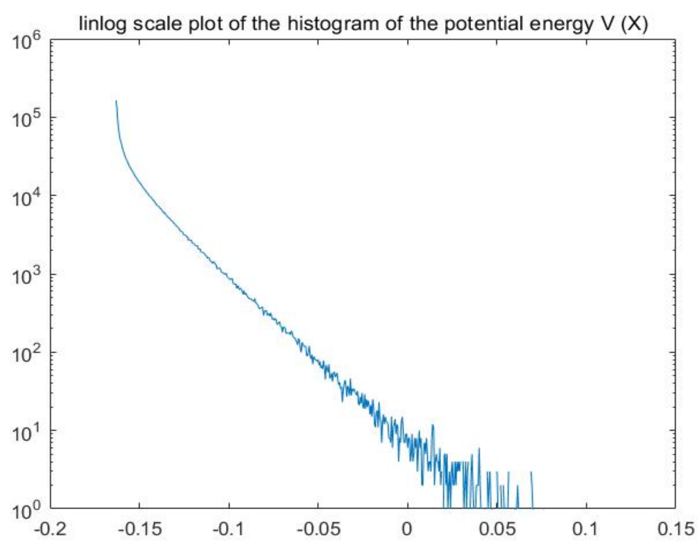

Figure 6: Linlog scale of the histogram of the potential energy $V(X)$ for all times t after a transient time of $t_T = 100$

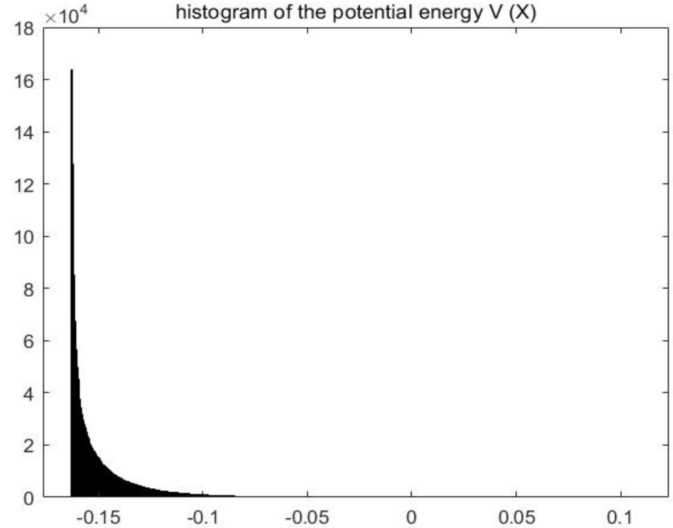

Figure 7: Histogram of the potential energy $V(X)$ for all times t after a transient time of $t_T = 100$

Actually, if we plot the mean of $X(t)$, as shown in Figure 8, we may see that the mean of $X(t)$ will suddenly go up to 0.5 and then oscillate in around this value. Remember that in we set $\sigma = 0.2$ here, which means that the noise is not that strong, so the motion of the particle will just goes down along the valley and then stuck in the valley as shown in Figure 9. Therefore I roughly conclude that if the noise is not that large, then the value of $X(T)$ will not blow up.

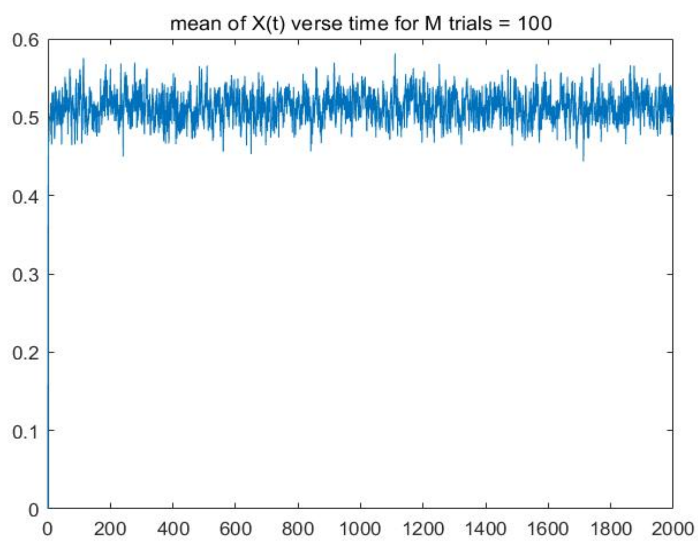

Figure 8: mean of $X(t)$ when $\sigma = 0.2$

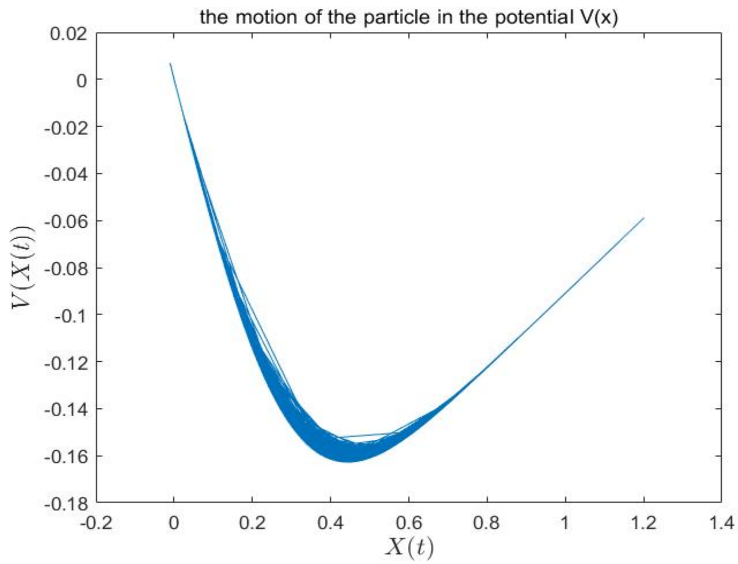

Figure 9: motion of the particle in the potential $V(x)$

2. Problem b

For part (b) we are going to simulate the Brownian ratchet for $dF = Gdt$ with

$$dG = -\frac{1}{\tau}Gdt + \frac{1}{\sqrt{\tau}}\widetilde{\sigma}\,dW \tag{2}$$

First I want to explain how I vectorized calculation of all the realizations in one time loop. In the code below I set $X(t)$ and $G(t)$ as a matrix with the row size is the trials M and column size is the time step size. Then in the calculation, I calculate all the M trials in the same time by using $G(:,)$ and $x(:,)$ and vectorized calculation dot. This exactly speed up my code.

- **Vectorizing the calculation:**

Listing 1: Vectorized calculation of all the realizations in one time loop

```
%%%%%%%%%%%%%%% initialize the parameter %%%%%%%%%%%%
      M_trials = M; % define the trials numbers
      nstepp=13; % define the factor of time step
      nsteps=2^nstepp; % define the total size of the time step
      tmax=T; % define the max time
      variance=tmax/nsteps; % define the delta t
      K = (6435*pi)/(16384); % define the parameter K
      g0=1;

      realization=randn(M_trials,nsteps); % define the W_0
      x=zeros(M_trials,nsteps); % define the matrix of X(t), which have M
          trials rows and nsteps time step columns
      Delta_W_n = realization*sqrt(variance); % for W_n_0
```

```
    G = zeros(M_trials,nsteps); % define the matrix of the G(t), which have the same size of X(t)

    ntj =1;
    dt(ntj)=tmax/nsteps;

%%%%%%%%%%%% compute all of the samples simultaneously
%%%%%%%%%%%% using a single time-stepping loop.
    for i=1:nsteps
        G(:,i+1)= G(:,i)+(-1/tau).*G(:,i).*dt(ntj)+(1/sqrt(tau)).*sigma.*Delta_W_n(:,i);
        x(:,i+1)=x(:,i)+dt(ntj).*(((cos(x(:,i))).^16)/K - 1/(2*pi))+g0.*G(:,i).*dt(ntj);
    end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

First we get a feeling for the system by plotting the motion of the particle in the potential $V(x)$ like Figure 10 when $\tilde{\sigma} = 0.5$ and $G(0) = 0$.

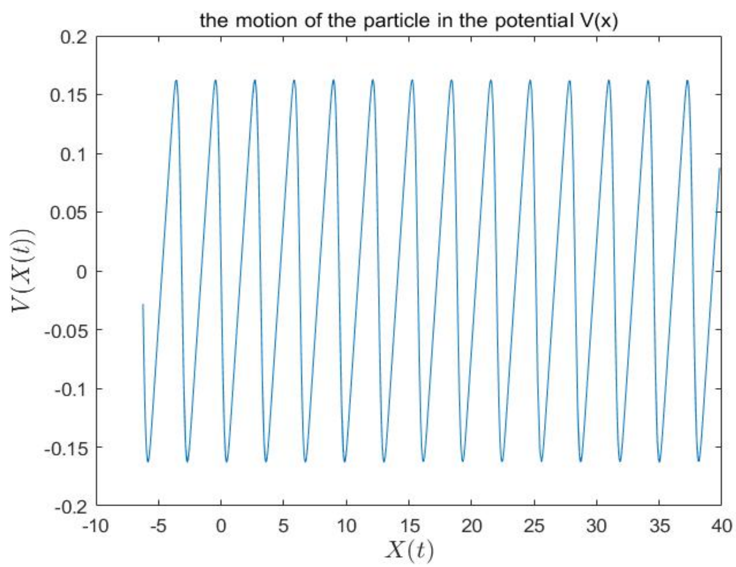

Figure 10: plot the motion of the particle in the potential $V(x)$

2.1 Problem b (i)

In this problem I am going to find a sufficiently small step size of Δt to obtain a weak error of $X(T)$ that is smaller than 1%. Here is how I calculate the weak error: For each Δt I do M realization, and for each realization I exact the $X(T)$ and then take the average over M trials. Here I choose $M = 10000$ and then I evaluate the difference between the Δt_i and the Δt_{i+1} which is large than Δt_i with a factor of 2. Then I get the Figure 11, the weak error line and the Δt line are comparing with each other and we can see that they seem like similar with each other except for when the Δt is large, and we can see that the weak error become suddenly large. A reasonable explanation is that since when we want to approximate the solution, we ignore the higher order term of Δt, therefore, if the Δt is large, the error would be large. Finally I choose the $\Delta t = 0.25$ since at this time the weak error is 0.0054, which is sufficiently smaller than 1%.

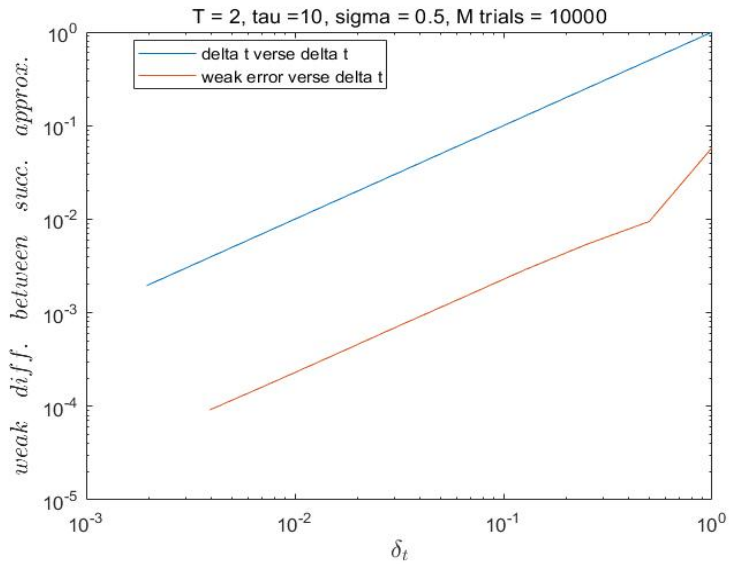

Figure 11: loglog plot of the weak difference between succ. approximation

2.2 Problem b (ii)

To show that $G(t)$ is correlated with $G(t^{'})$, as shown in Figure 12 and 13, we see that the autocorrelation values get max when Δt become zero, which means the autocorrelation of $G(t)$ itself. And as the Δt becomes larger, the autocorrelation value decays exponentially. This result is same in different $\tilde{\sigma}$.

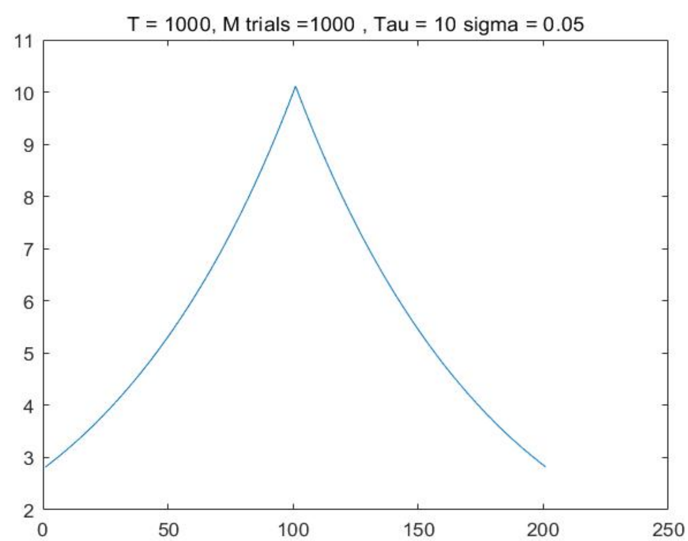

Figure 12: Autocorrelation function $C(\Delta t)$ when $\tau = 10$, $\tilde{\sigma} = 0.05$

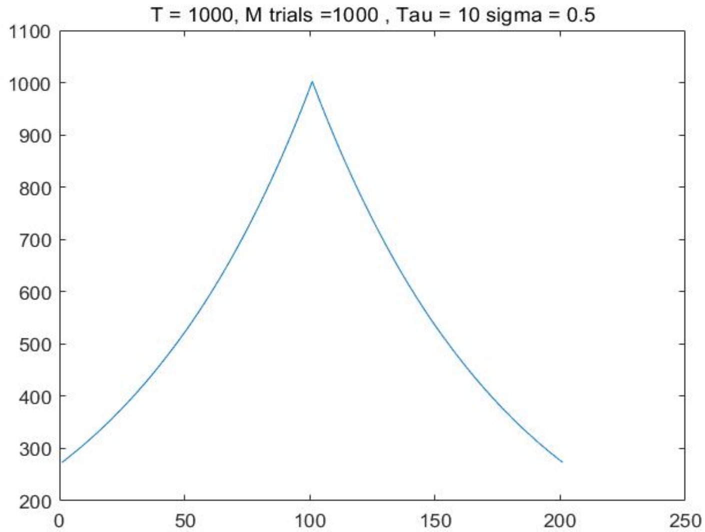

Figure 13: Autocorrelation function $C(\Delta t)$ when $\tau = 10$, $\tilde{\sigma} = 0.5$

2.3 Problem b (iii)

In the last subproblem, for $T = 5000$ and the step size $\Delta t = 0.25$, I plot $< X(T) > /T$ as a function of τ in the range $0.1 \leqslant \tau \leqslant 40$. We can see that when τ is sufficiently large, the mean drift speed $< X(T) > /T$ will suddenly increase.

Actually, the correlations between dW at one time and dW at another time in the Wiener process is equal to 0, and the correlation between $G(t)$ and $G(t^{'})$ depends on τ. When τ is equal to zero, then the $t^{'}$-integral of $G(t')$ in Ornstein-Uhlenbeck process can be proved as $W(t)$ as shown in the Note. This is quite similar to the Part 1, and as shown in Figure 14, when τ is small, the mean drift shift will be quite small, which is the same as I found in Problem a (iii) when $\sigma = 0.2$, which means for a weak noise, the mean drift shift is $2.3944 * 10^{-4}$. They are quite similar in the scale when the noise is weak when $dF = \sigma dW$ and when the τ is small when $dF = Gdt$. And when the τ is sufficiently large, in my opinion, the mechanism that drives the net motion would be the strong noise, which is quite similar to the part a when $\sigma = 1$ as we see in the part a (ii) that the mean drive shift would be large if we only do small trials. When the mean drive shift is large, the particle can get out of the valley as we shown in Figure 9 and which leads to sufficiently increase in $X(t)$. Well actually, we can also look at the equation (2) and when τ is sufficiently large, the term will $\frac{1}{\sqrt{\tau}}$ will dominate.

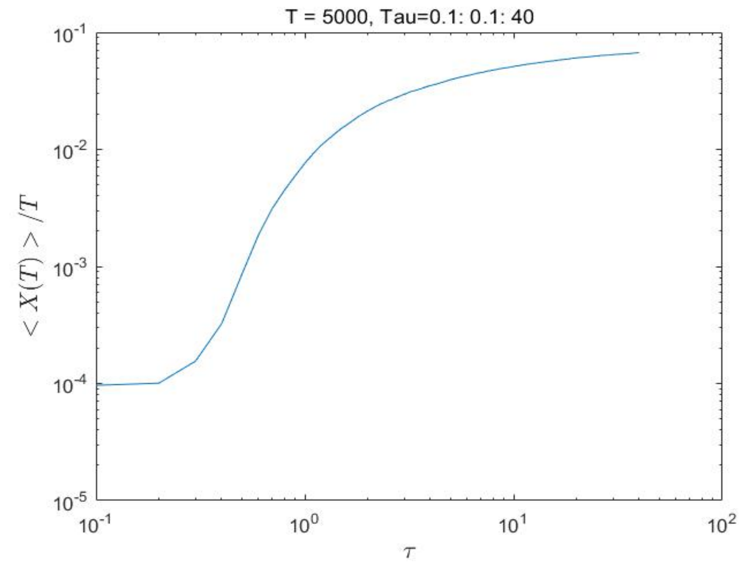

Figure 14: loglog plot of the $< X(T) > /T$ as a function of τ in the range $0.1 \leqslant \tau \leqslant 40$

3. Part c MATLAB code

```matlab
function X = Liang_Mingfu_hw4(M,T,sigma,tau)

%%%% Homework 4 Part c %%%%%%%%
%%%% author: Mingfu Liang %%%%%%%
%%%% date: 03/21/2019 %%%%%%

%%%%%%%%%%%%%% initialize the parameter %%%%%%%%%%%
    M_trials = M; % define the trials numbers
    nstepp=13; % define the factor of time step
    nsteps=2^nstepp; % define the total size of the time step
    tmax=T; % define the max time
    variance=tmax/nsteps; % define the delta t
    K = (6435*pi)/(16384); % define the parameter K
    g0=1;

    realization=randn(M_trials,nsteps); % define the W_0
    x=zeros(M_trials,nsteps); % define the matrix of X(t), which have M trials rows and nsteps time step columns
    Delta_W_n = realization*sqrt(variance); % for W_n_0
    G = zeros(M_trials,nsteps); % define the matrix of the G(t), which have the same size of X(t)

    ntj =1;
```

```
    dt(ntj)=tmax/nsteps;

%%%%%%%%%%%%% compute all of the samples simultaneously
%%%%%%%%%%%%% using a single time-stepping loop.
    for i=1:nsteps
        G(:,i+1)= G(:,i)+(-1/tau).*G(:,i).*dt(ntj)+(1/sqrt(tau)).*sigma.*
            Delta_W_n(:,i);
        x(:,i+1)=x(:,i)+dt(ntj).*(((cos(x(:,i))).^16)/K - 1/(2*pi))+g0.*G(:,
            i).*dt(ntj);
    end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%% Extract the X(T) %%%%%%%%%%%%%
    X = x(:,end);

 %%%%%%%%%%%%%%%%%% plot X(t) for one realization %%%%%%%%%%%%%%%%%%

%   figure;
%   plot(0:dt(ntj):tmax, x(1,:))
%   ylabel('$X(t)$','Interpreter','latex','FontSize',13)
%   xlabel('$t$','Interpreter','latex','FontSize',13)

 %%%%%%%%%%%%%%%%%% plot <X(t)> for all realization %%%%%%%%%%%%%%%%%%

%   figure;
%   plot(0:dt(ntj):tmax, mean(x))
%   ylabel('$<X(t)>$','Interpreter','latex','FontSize',13)
%   xlabel('$t$','Interpreter','latex','FontSize',13)

%%%%%%%%%%%%%%%%%% autocorrelation of G(t) %%%%%%%%%%

%       G_cov=zeros(M_trials,2*100+1);
%       for i =1:M_trials
%           G_cov(i,:) = xcov(G(i,:),100);
%       end
%
%       figure;
%       plot(mean(G_cov))
%       title(['xcov of G whenT = ',num2str(tmax),', M trials =', num2str(
```

```
    M_trials), ' , Tau = ', num2str(tau),' sigma = ',num2str(sigma)])

end
```

4. Part a MATLAB code

4.1 Problem a (i) and (ii)

```
%%%% Homework 4 Part a (i) and (ii) %%%%%%%%
%%%% author: Mingfu Liang %%%%%%%
%%%% date: 03/21/2019 %%%%%%

tic
M_trials = 8; % define the trials numbers
nstepp=12; % define the factor of time step
nsteps=2^nstepp; % define the total size of the time step
tmax=64; % define the max time
variance=tmax/nsteps;  % define the delta t
g0=1;
realization=randn(M_trials,nsteps); % define the W_0
x=zeros(M_trials,nsteps); % define the matrix of X(t), which have M trials
    rows and nsteps time step columns
Delta_W_n = realization*sqrt(variance); % for W_n_0
K = (6435*pi)/(16384);
ntj = 1;
ntfactor=2^(ntj-1);
nt=nsteps/ntfactor;
dt(ntj)=tmax/nt;

%%%%%%%%%%%%%%%%%% calculate the X(t) %%%%%%%%%%

for  i=1:nt
          x(:,i+1)=x(:,i)+dt(ntj).*(((cos(x(:,i))).^16)/K - 1/(2*pi))+g0.*
              Delta_W_n(:,i);
end

%%%%%%%%%%%% several plot of X(t) %%%%%%%%%%%

figure;
plot(0:dt(ntj):tmax,x(1,:));
hold on
```

```
plot(0:dt(ntj):tmax,x(2,:));
hold on
plot(0:dt(ntj):tmax,x(3,:));
hold on
plot(0:dt(ntj):tmax,x(4,:));
hold on
plot(0:dt(ntj):tmax,x(5,:));
hold on
plot(0:dt(ntj):tmax,x(6,:));
hold on
plot(0:dt(ntj):tmax,x(7,:));
hold on
plot(0:dt(ntj):tmax,x(8,:));
hold off
ylabel('$x(t)$','Interpreter','latex','FontSize',13)
xlabel('$t$','Interpreter','latex','FontSize',13)
title(['several trajectories of X(t) for sigma =1 and M trials = ',num2str(M_trials)])

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

average_rate = mean(x(:,nsteps+1))/1000;
Standard_deviation = std(x);
Standard_deviation_T = std(x(:,nsteps+1)/1000);
Standard_error_mean = Standard_deviation_T/sqrt(M_trials);
mean_x = mean(x);
figure;
plot(0:dt(ntj):tmax,mean_x);
title(['mean of X(t) verse time for M trials = ', num2str(M_trials)])
figure;
plot(0:dt(ntj):tmax,Standard_deviation)
title(['standard derivation of X(t) verse time for M trials = ', num2str(M_trials)])
toc

fprintf(['Need ', num2str(M_trials), ' to achieve standard error of mean ', num2str(Standard_error_mean)],'\n')
```

4.2 Problem a (iii)

```
%%%%% Homework 4 Part a(ii) %%%%%%%%%
```

```matlab
%%%%% author: Mingfu Liang %%%%%%%
%%%%% date: 03/21/2019 %%%%%%

tic
M_trials = 100; % define the trials numbers
nstepp=14;  % define the factor of time step
nsteps=2^nstepp; % define the total size of the time step
tmax=2000;  % define the max time
variance=tmax/nsteps; % define the delta t
nsteps_100 = round(nsteps * (100/2000)); % define the T=100
g0=0.2; % sigma =0.2 in this question
realization=randn(M_trials,nsteps); % define the W_0
x=zeros(M_trials,nsteps);  % define the matrix of X(t), which have M trials
    rows and nsteps time step columns
Delta_W_n = realization*sqrt(variance); % for W_n_0
K = (6435*pi)/(16384);

ntj = 1;
ntfactor=2^(ntj-1);
nt=nsteps/ntfactor;
dt(ntj)=tmax/nt;

%%%%%%%%%%%%%%%% calculate the X(t) %%%%%%%%%%

for i=1:nt
        x(:,i+1)=x(:,i)+dt(ntj).*(((cos(x(:,i))).^16)/K - 1/(2*pi))+g0.*
            Delta_W_n(:,i);
end

%%%%%%%%%%%%%%%% calculate the V(t) %%%%%%%%%%

V= (-1/(12870*pi))*(5720*sin(2.*x)+2002*sin(4.*x)+728*sin(6.*x)+(455/2)*sin
    (8.*x) ...
     + 56*sin(10.*x)+10*sin(12.*x)+(8/7)*sin(14.*x)+(1/16)*sin(16.*x));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure;
plot(x(1,:),V(1,:))
ylabel('$V(X(t))$','Interpreter','latex','FontSize',13)
```

```matlab
xlabel('$X(t)$','Interpreter','latex','FontSize',13)
title('the motion of the particle in the potential V(x)')

mean_x_T = mean(x(:,end))/tmax;

figure;
h = histogram(V(:,nsteps_100:end));
title('histogram of the potential energy V (X)')
figure;
semilogy(h.BinEdges(1:end-1),h.Values)
title('linlog scale plot of the histogram of the potential energy V (X)')

toc
```

5. Part b MATLAB code

5.1 Problem b (i)

```matlab
%%%%% Homework 4 Part b(i) %%%%%%%%
%%%%% author: Mingfu Liang %%%%%%%
%%%%% date: 03/21/2019 %%%%%%

M_trials =10000; % define the trials numbers
tau = 10;
sigma = 0.5;
nstepp=10; % define the factor of time step
nsteps=2^nstepp; % define the total size of the time step
ntjmax=10; % define how many different delta t I would want to evaluate
tmax=2; % define the max time
g0=1;
variance=tmax/nsteps; % define the delta t
posit = 1: nsteps; % define the index variables
Delta_W_n =zeros(nstepp,nsteps,M_trials);
realization=randn(1,nsteps,M_trials); % define the W_0

 % define the 3 dimension matrix of X(t), which have M trials 2 dimension matrix, in each two dimension matrix
 % it has ntjmax row denote different each delta t and nsteps column for each delta t
x=zeros(ntjmax,nsteps,M_trials);

```

```
G = zeros(ntjmax,nsteps,M_trials); % define the matrix of the G(t), which have the same size of X(t)
Delta_W_n(1,:,:) = realization(1,:,:)*sqrt(variance); % for W_n_0
K = (6435*pi)/(16384);

%%%%%%%%%%%%%%%%%% calculate W_n %%%%%%%%%%%%%%%%%%%%%%%%

for k=2:ntjmax
    Delta_W_n(k,1:2^(nstepp-k+1),:) = Delta_W_n(k-1,1:2:length(Delta_W_n(2,1:2^(nstepp-(k-1)+1)))-1,:)+ Delta_W_n(k-1,2:2:length(Delta_W_n(k-1,1:2^(nstepp-(k-1)+1))),:);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

mean_M_trials_X_t = zeros(ntjmax,nsteps+1); % define the mean of X(t) over M trials

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%% loop for all different delta t and find out which one would have
%%%%%% smallest weak error

for ntj = ntjmax:-1:1
        ntfactor=2^(ntj-1);
        nt=nsteps/ntfactor;
        dt(ntj)=tmax/nt;
        for i=1:nt
        G(ntj,i+1,:)= G(ntj,i,:) + (-1/tau).*G(ntj,i,:).*dt(ntj)+(1/sqrt(tau)).*sigma.*Delta_W_n(ntj,i,:);
        x(ntj,i+1,:)=x(ntj,i,:)+dt(ntj).*(((cos(x(ntj,i,:))).^16)/K - 1/(2*pi))+g0.*G(ntj,i,:).*dt(ntj);
        end
        M_trials_X_t = reshape(x(ntj,1:nt+1,:),[nt+1,M_trials]).';
        mean_M_trials_X_t(ntj,1:nt+1) = mean(M_trials_X_t(:,1:nt+1));
        mean_M_trials_X_T(ntj,1) = mean_M_trials_X_t(ntj,nt+1);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```
weak_error_mean_shift =abs(mean_M_trials_X_T(2:end,1) -mean_M_trials_X_T(1:
    end-1,1));

figure();
loglog(dt,dt);
hold on
loglog(dt(2:nstepp),weak_error_mean_shift)
ylabel('$ weak \quad diff. \quad between \quad succ. \quad approx.$','
    Interpreter','latex','FontSize',13)
xlabel('$\Delta {t}$','Interpreter','latex','FontSize',13)
legend('delta t verse delta t','weak error verse delta t')
title(['T = ', num2str(tmax), ', tau =', num2str(tau), ', sigma = ', num2str
    (sigma), ', M trials = ', num2str(M_trials)])
hold off
```

5.2 Problem b (ii)

```
%%%% Homework 4 Part b (ii) %%%%%%%%
%%%% author: Mingfu Liang %%%%%%%
%%%% date: 03/21/2019 %%%%%%

M_trials =1000; % define the trials numbers
tau = 10;
sigma = 0.5;
nstepp=13; % define the factor of time step
nsteps=2^nstepp; % define the total size of the time step
tmax=1000; % define the max time
variance=tmax/nsteps; % define the delta t
amp=0;
field=0;
g0=1;
K = (6435*pi)/(16384);

posit = 1: nsteps; % define the index variables
realization=randn(M_trials,nsteps); % define the W_0
x=zeros(M_trials,nsteps); % define the matrix of X(t), which have M trials
    rows and nsteps time step columns
Delta_W_n = realization*sqrt(variance); % for W_n_0
G = zeros(M_trials,nsteps); % define the matrix of the G(t), which have the
    same size of X(t)

```

```
ntj = 1;
ntfactor=2^(ntj-1);
nt=nsteps/ntfactor;
dt(ntj)=tmax/nt;

%%%%%%%%%%% compute all of the samples simultaneously
%%%%%%%%%%% using a single time-stepping loop.

for i=1:nt
        G(:,i+1)= G(:,i)+(-1/tau).*G(:,i).*dt(ntj)+(1/sqrt(tau)).*sigma.*Delta_W_n(:,i);
        x(:,i+1)=x(:,i)+dt(ntj).*(((cos(x(:,i))).^16)/K - 1/(2*pi))+g0.*G(:,i).*dt(ntj);
end

%%%%%%%%%%%%%%%%%%%%% vectorizing calculate V(X(t)) %%%%%%%%%%%%%%%%%%%%%%%%%

V= (-1/(12870*pi))*(5720*sin(2.*x)+2002*sin(4.*x)+728*sin(6.*x)+(455/2)*sin(8.*x) ...
   + 56*sin(10.*x)+10*sin(12.*x)+(8/7)*sin(14.*x)+(1/16)*sin(16.*x));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure;
plot(x(1,:),V(1,:))
ylabel('$V(X(t))$','Interpreter','latex','FontSize',13)
xlabel('$X(t)$','Interpreter','latex','FontSize',13)
title('the motion of the particle in the potential V(x)')

%%%%%% plot the autocorrelation averaging over large trials %%%%%%%%%%%

mean_x = mean(x);
G_cov=zeros(M_trials,2*100+1);
for i =1:M_trials
G_cov(i,:) = xcov(G(i,:),100);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```
figure;
plot(mean(G_cov))
title(['T = ',num2str(tmax),', M trials =', num2str(M_trials), ' , Tau = ', num2str(tau),' sigma = ',num2str(sigma)])
```

5.3 Problem b (iii)

```
%%%% Homework 4 Part b (iii) %%%%%%%%
%%%% author: Mingfu Liang %%%%%%%
%%%% date: 03/21/2019 %%%%%%

tic
M_trials =100; % define the trials numbers
tau_head = 100; % define the max of tau
tau_bottom = 0.1; % define the min of tau
delta_tau = 0.1; % define the change of tau
tau_range = round((tau_head-tau_bottom)/delta_tau); % define the range of tau
sigma = 0.5; % define sigma
nstepp=15; % define the factor of time step
nsteps=2^nstepp; % define the total size of the time step
tmax=5000; % define the max time
variance=tmax/nsteps; % define the delta t
g0=1;
K = (6435*pi)/(16384);

realization=randn(M_trials,nsteps); % define the W_0
x=zeros(M_trials,nsteps);  % define the matrix of X(t), which have M trials rows and nsteps time step columns
Delta_W_n = realization*sqrt(variance); % for W_n_0
G = zeros(M_trials,nsteps); % define the matrix of the G(t), which have the same size of X(t)

ntj = 1;
ntfactor=2^(ntj-1);
nt=nsteps/ntfactor;
dt(ntj)=tmax/nt;
mean_X_T_vec = zeros(tau_range,1);
k =1;

%%%% evaluate the mean drift shif over different tau %%%%%%%%%%%%%%%%
```

```
for tau = tau_bottom:delta_tau:tau_head

    for i=1:nt
        G(:,i+1)= G(:,i)+(-1/tau).*G(:,i).*dt(ntj)+(1/sqrt(tau)).*sigma.*Delta_W_n(:,i);
        x(:,i+1)=x(:,i)+dt(ntj).*(((cos(x(:,i))).^16)/K - 1/(2*pi))+g0.*G(:,i).*dt(ntj);
    end

    mean_X_T = mean(x(:,end))/tmax;
    mean_X_T_vec(k,1)=mean_X_T;
    k = k+1;

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure;
loglog(tau_bottom:delta_tau:tau_head,mean_X_T_vec);
xlabel('$\tau$','Interpreter','latex','FontSize',14)
ylabel('$<X(T)>/T$','Interpreter','latex','FontSize',14)
toc
```