**Computer Problems**

Problem 4.10

1. **Poisson Process**

Code Poisson.m:

function [T,NT]= poisson(Na,lam)

%[T,NT]= poisson(Na,lam)

% lam : lambda of poisson process or arrival rate

% Na : Number of desired arrivals to generate

% T : Times at which arrivals occur (Arrival times)

% NT : Number of arrivals upto each time T

tau = randexp(Na,1,lam); % interarrival times, exp distributed

T = cumsum(tau); % Times at which arrivals occur (Arrival times)

NT = [1:length(T)]'; % Number of arrivals upto each time T

Code:

%% Plot the sample path of poisson process

[T, NT] = poisson(10,1); % lam = 1/5/0.1/10

figure()

stairs(T,NT)

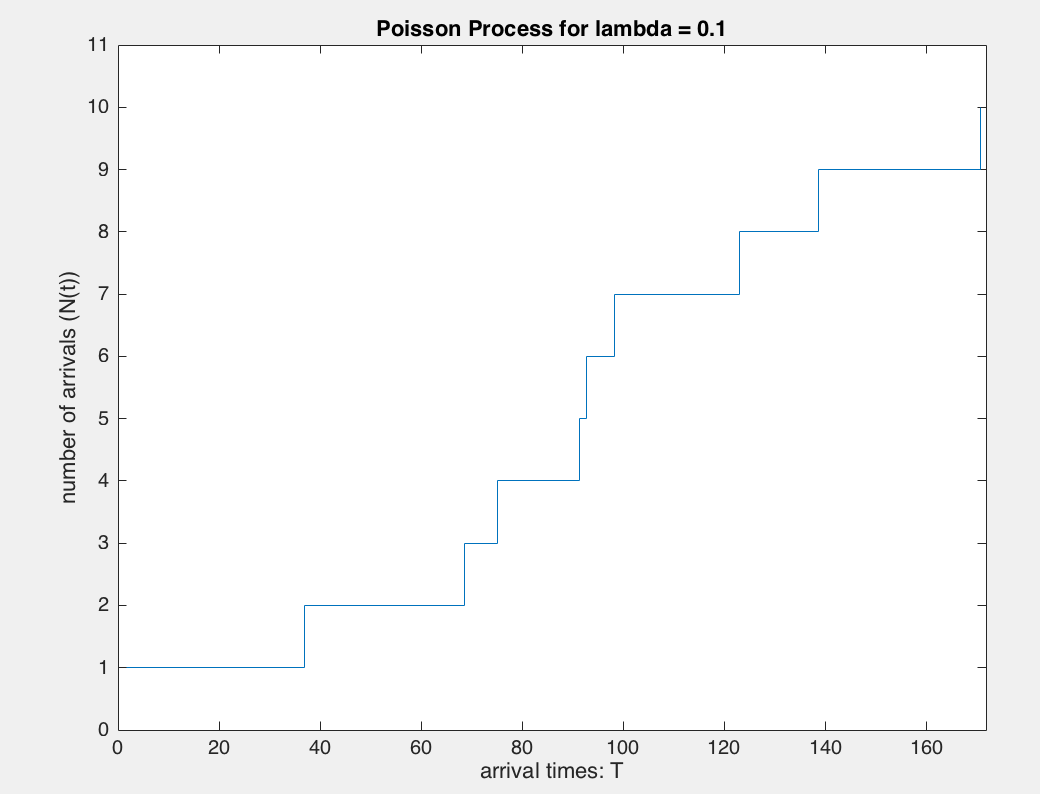
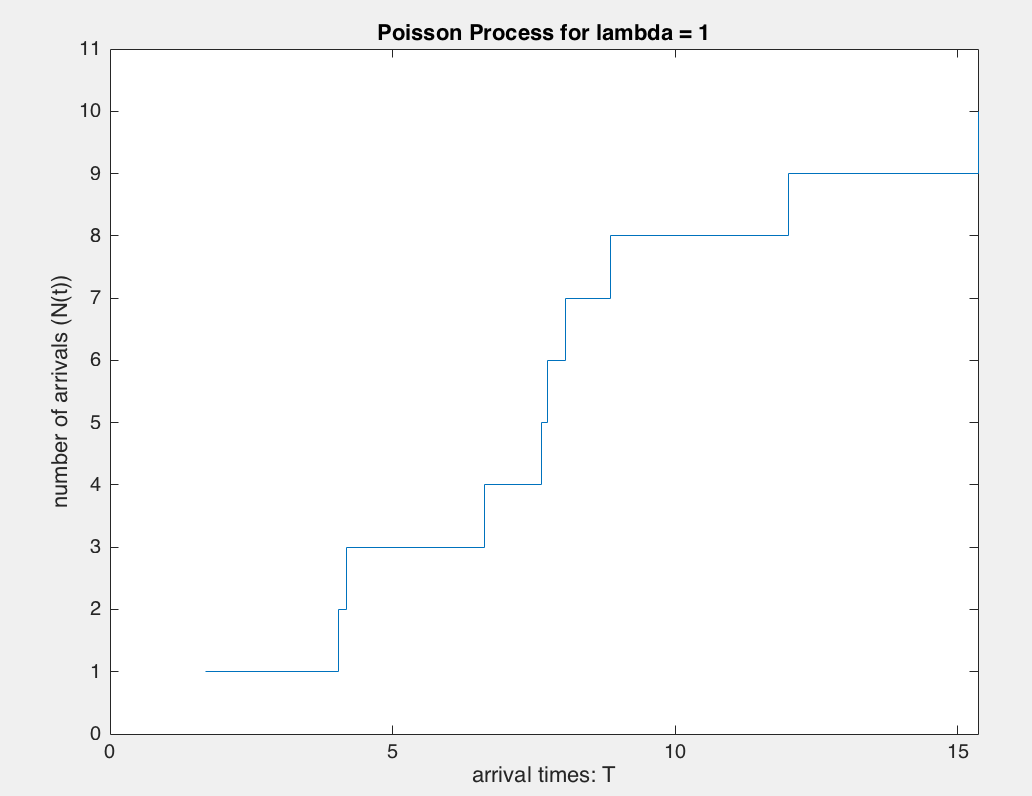
axis([0 max(T+1) 0 max(NT)+1])

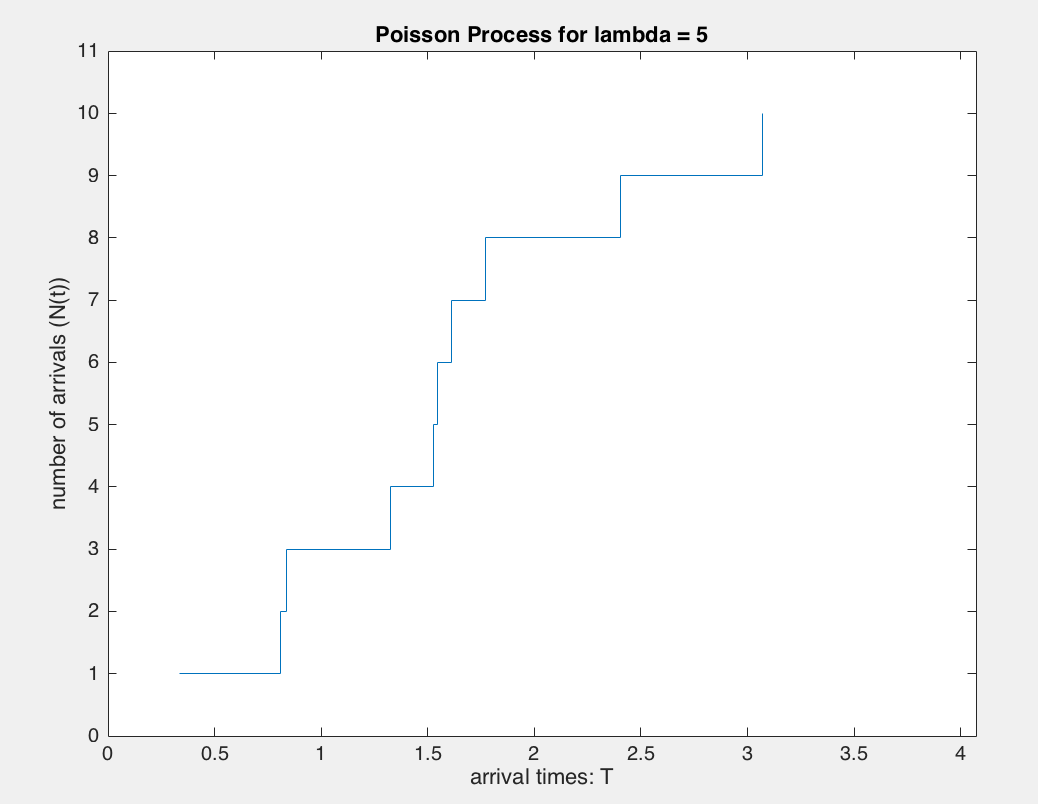
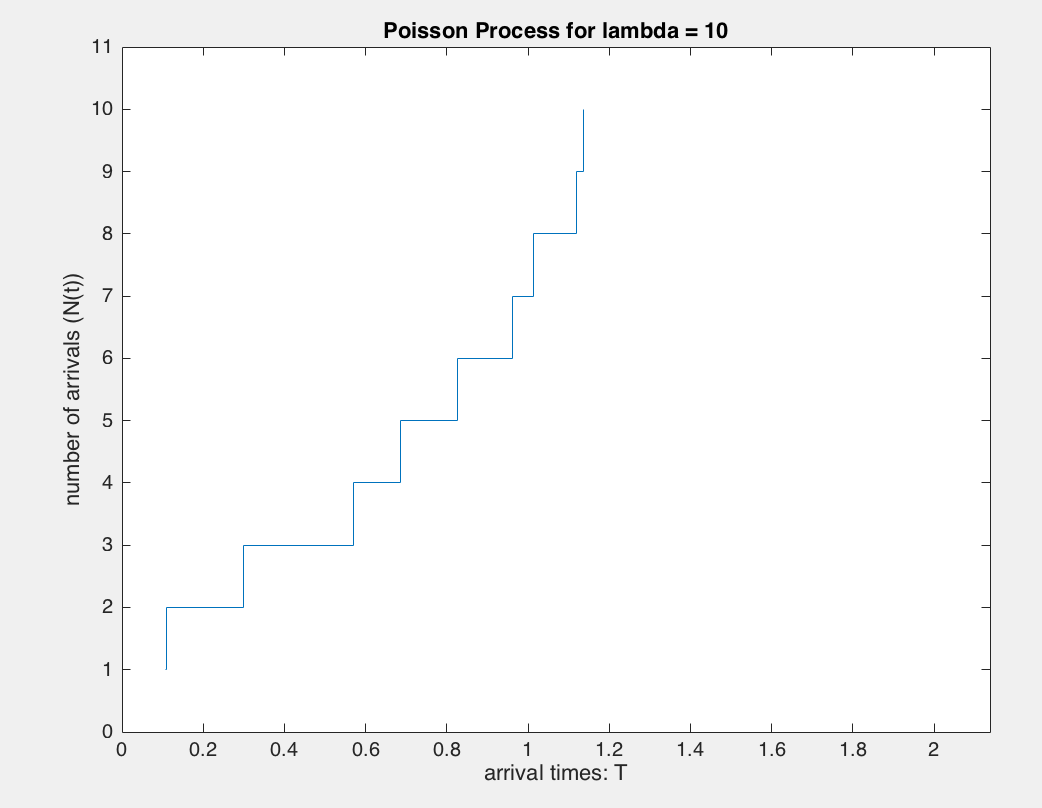
title('Poisson Process for lambda = 10')

ylabel('number of arrivals (N(t))')

xlabel('arrival times: T')

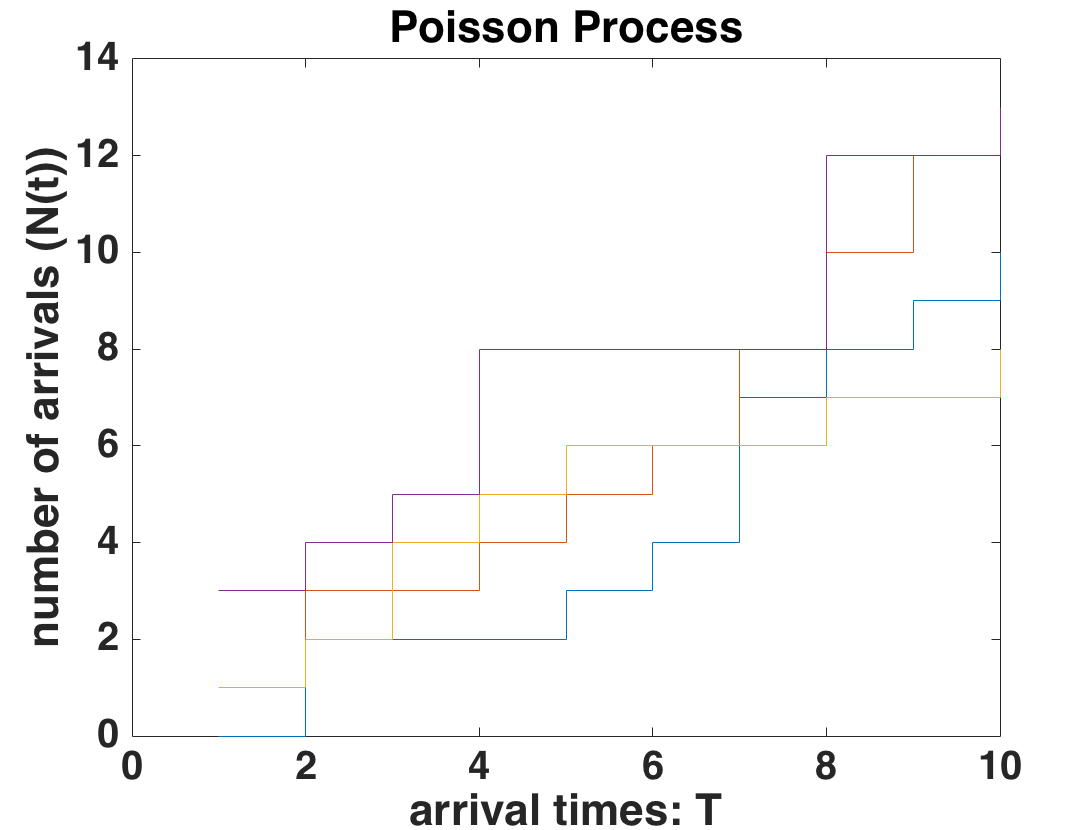
Plotting the sample paths of poisson process by varying lambda using poisson.m:

Since the values are only at discontinous points and no samples elsewhere, ensemble averages obtained wouldn't be proper due to lack of proper samples.

Plotting sample paths using poissrp.m, by generating sample path values of Poisson random processes at fixed sets of times.



Code:

%% Plot the mean and autocovariance function using sample paths generated by poissrp.m

[X,t] = poissrp(10000,[1:10],2);

m\_x = mean(X,1);

autocov = mean((X - repmat(m\_x,10000,1)).^2);

plot(t,mean(X,1))

axis([0 11 0 20])

xlabel('time (t)')

ylabel('mean of poisson process: lambda\*time')

title('mean of poisson process: lambda = 2')

plot(t,autocov)

axis([0 11 0 20])

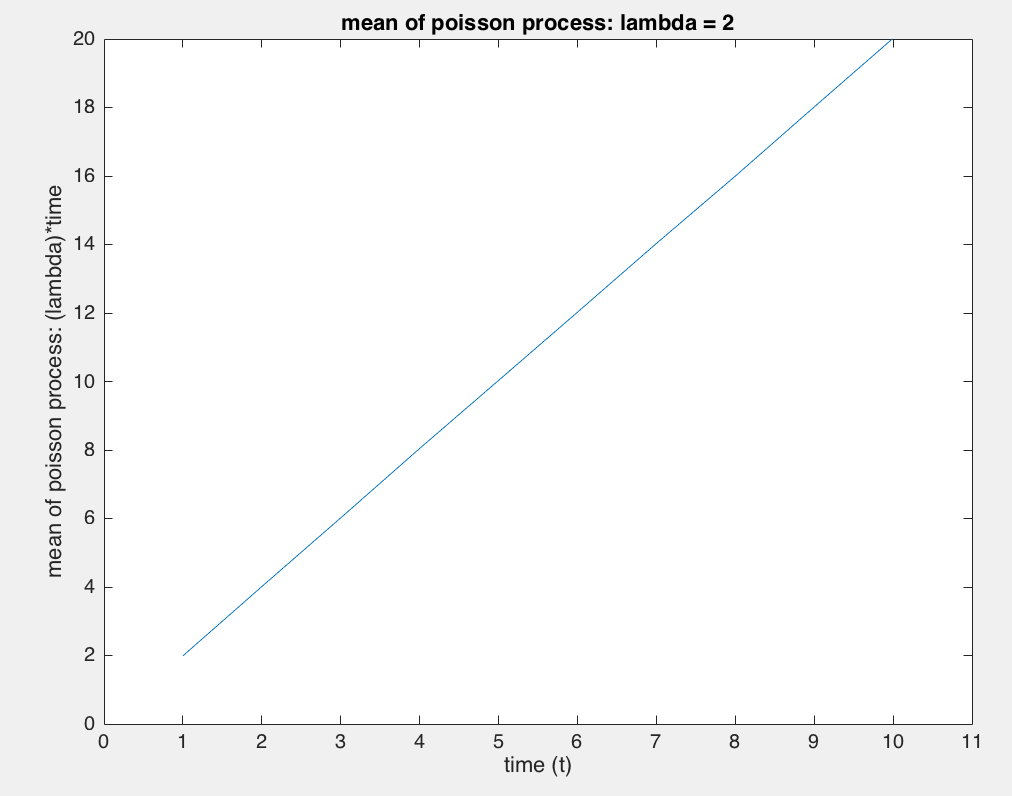
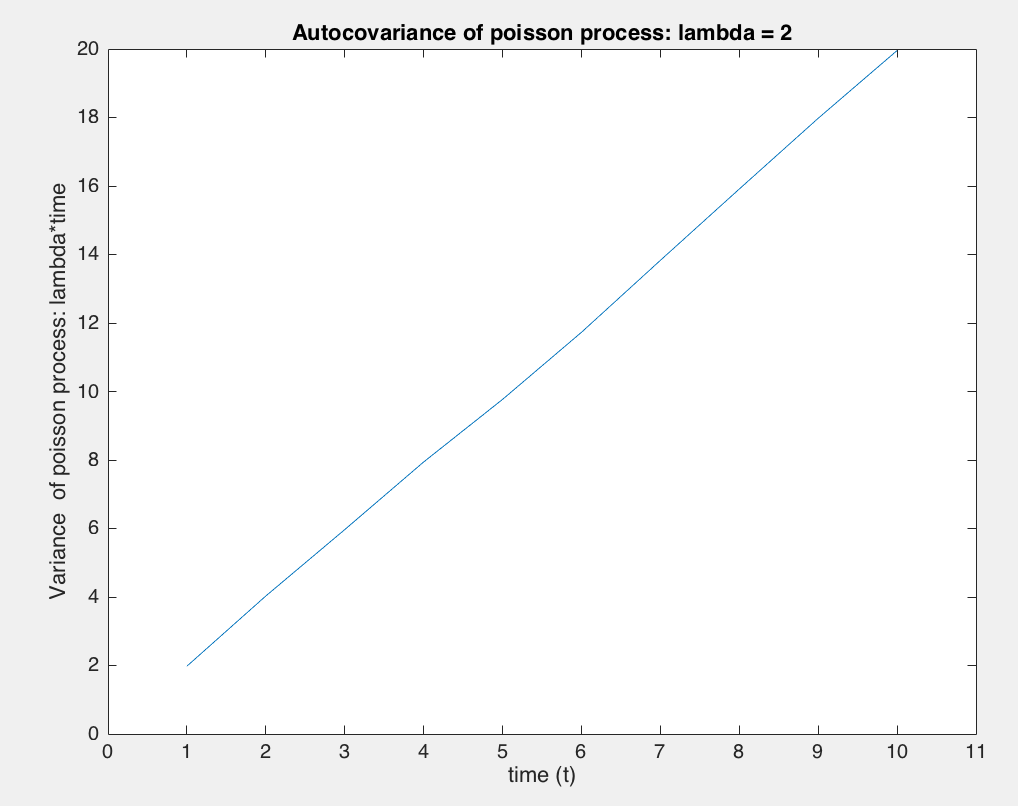
xlabel('time (t)')

ylabel('Variance of poisson process: lambda\*time')

title('Autocovariance of poisson process: lambda = 2')

Plotting mean and autocovariance function of poisson process using sample paths generated from running poissrp.m:

Mean Autocovariance

Theoretical functions (mean = variance = lambda\*time) discussed in the class matched the empirical mean and autocovariance funtions.

1. **Telegraph Process**

Code telerp.m:

function [X,t] = telerp(N,t,l)

%[X,t] = telerp(N,t,l)

% N : Number of sample paths to generate

% t : Vector of time points at which to generate samples

% l : Arrival rate. OPTIONAL. Default l=1.

% X : Matrix of process sample paths. Each row is a sample path, each column

% is a different time point.

% X(t) = Telegraph Random process with rate l

[Nt,t] = poissrp(N,t,l);

U = rand(N,length(t));

p = 0.5;

Z = U<p;

Z = double(Z);

Z(Z==0) = -1;

X = Z.\* (-1).^Nt;

Code:

%% Plot sample paths generated by telerp.m

[X,t] = telerp(10,[0:10],2);

figure()

for i = 1:4

hold on

stairs(t,X(i,:))

end

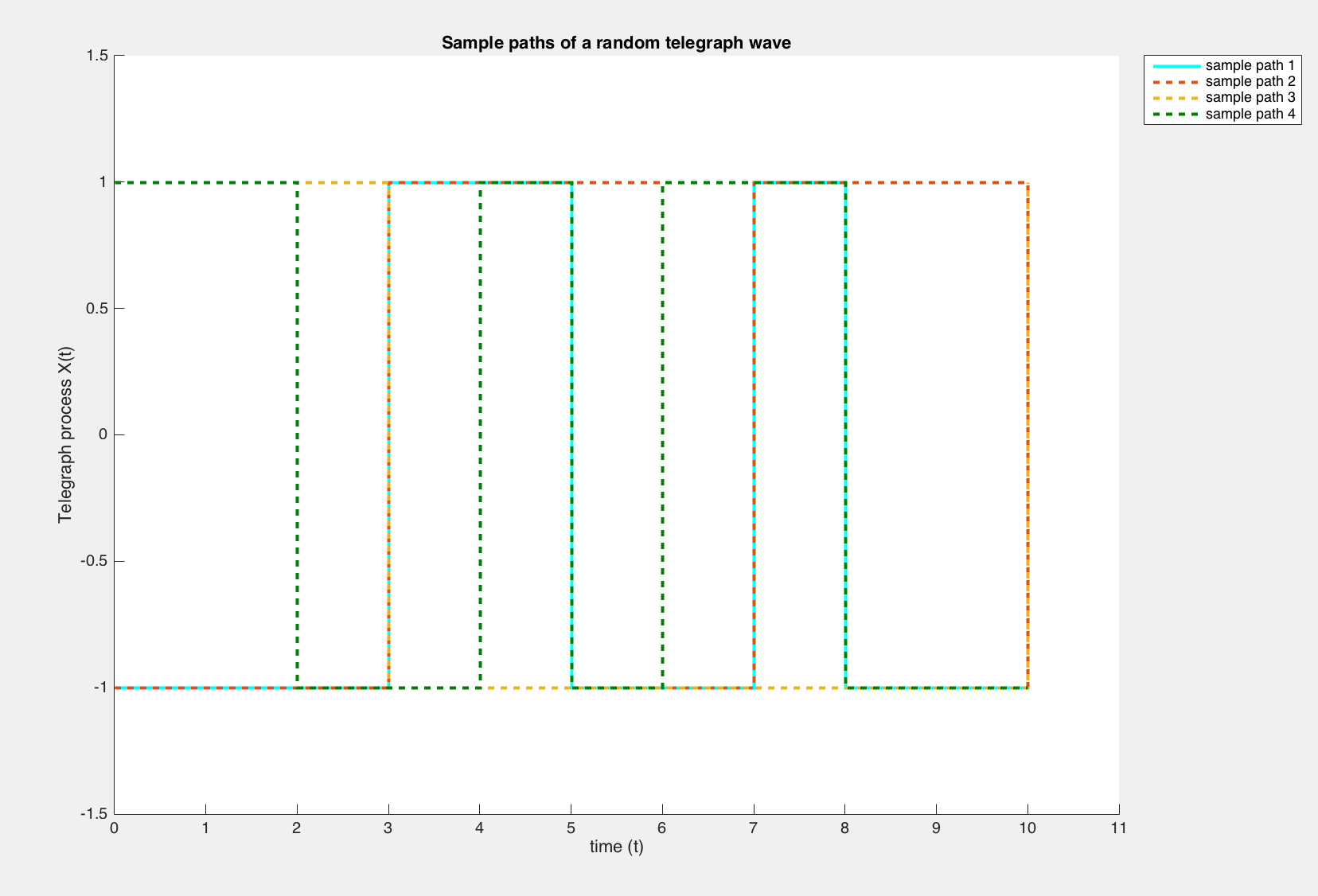
axis([0 11 -1.5 1.5])

xlabel('time (t)')

ylabel('Telegraph process X(t)')

title('Sample paths of a random telegraph wave')

Plotting sample paths using telerp.m, by generating sample path values of Telegraph random processes at fixed sets of times.



Code:

%% Plot the mean and autocovariance function using sample paths generated by telerp.m

[X,t] = telerp(10000,[0:10],2);

m\_x = mean(X,1);

autocov = mean((X - repmat(m\_x,10000,1)).^2);

plot(t,mean(X,1))

axis([0 10 -1 1])

xlabel('time (t)')

ylabel('mean')

title('mean of telegraph random process: lambda = 2')

plot(t,autocov)

axis([0 10 0.9 1.1])

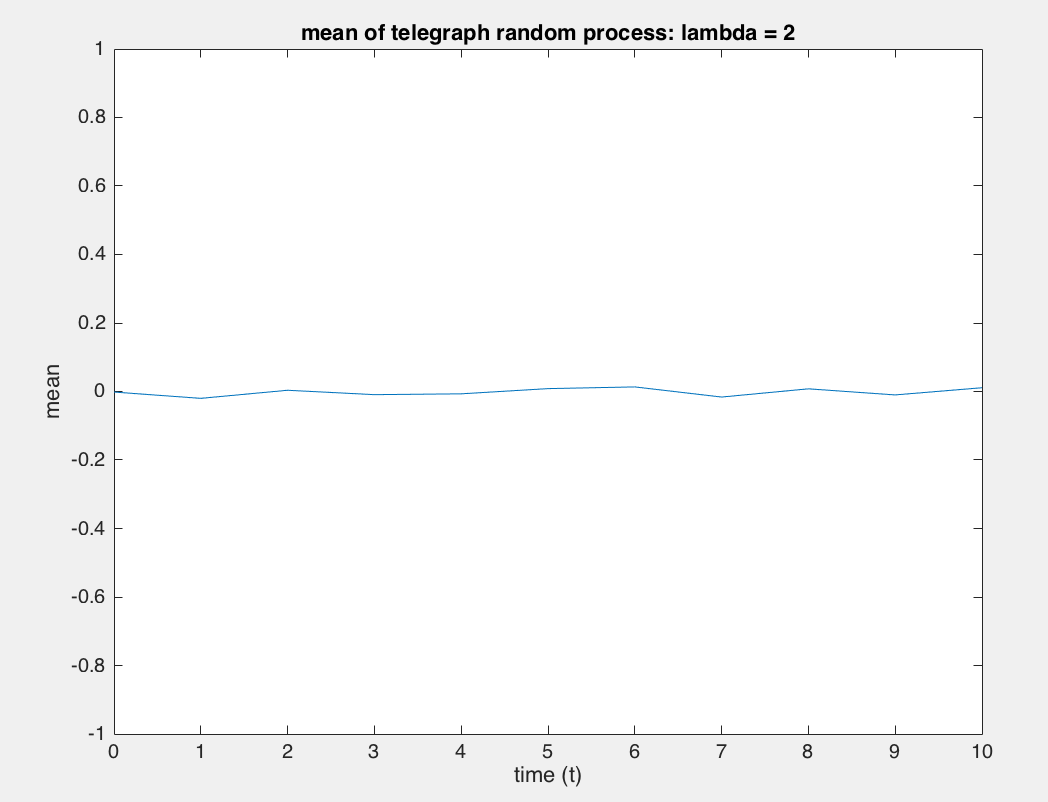
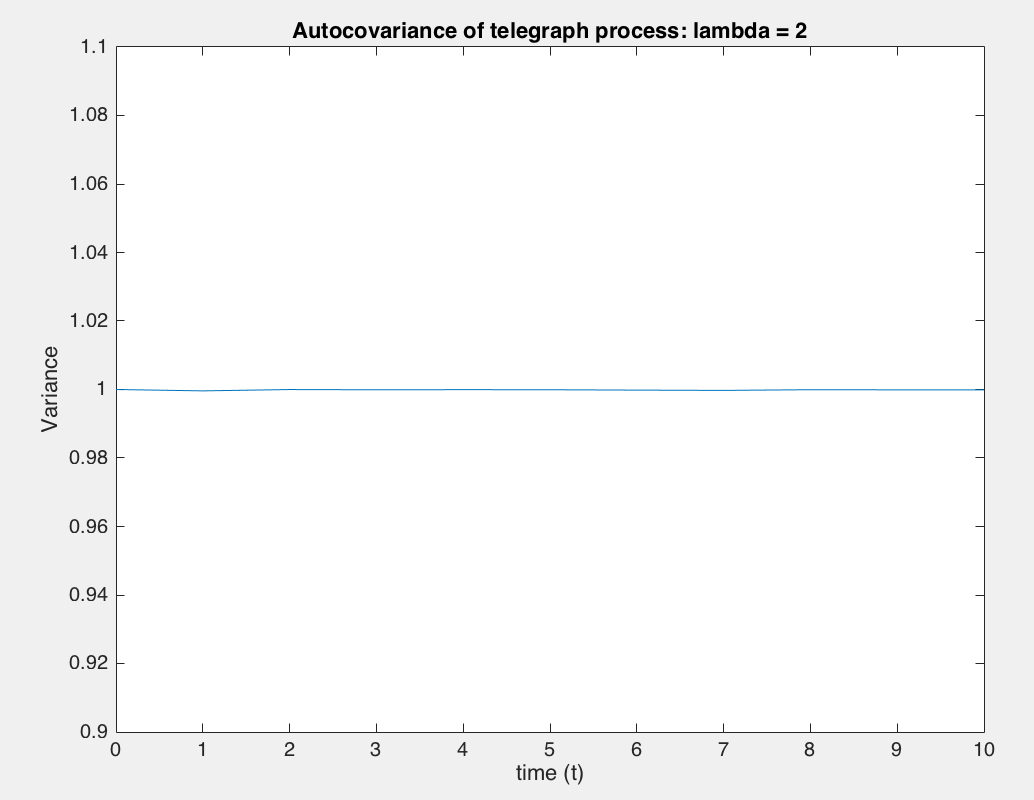
xlabel('time (t)')

ylabel('Variance')

title('Autocovariance of telegraph process: lambda = 2')

Plotting mean and autocovariance function of telegraph process:

Mean Autocovariance

1. **Wiener Process**

Code wiener.m:

function [X,t] = wiener(T,alpha)

% [X,t] = wiener(T,alpha)

% alpha: ratio of the step size squared to time interval s^2/T

% T: time step

% Z: Increment (Bernoulli) vector iid process values at the times in the vector t

t = 0:T:1; % generate the set of time points

s = sqrt(alpha\*T); % set the step size s of the discrete random walk

z = round(rand(length(t),1)); % Bernoulli Trials via rounded uniform RVs

jumps = s\*( sign(z - .5) );

X = cumsum(jumps);

Code:

%% Plot sample paths

[X,t] = wiener(0.01,10);alpha = 1,0.1,10,100

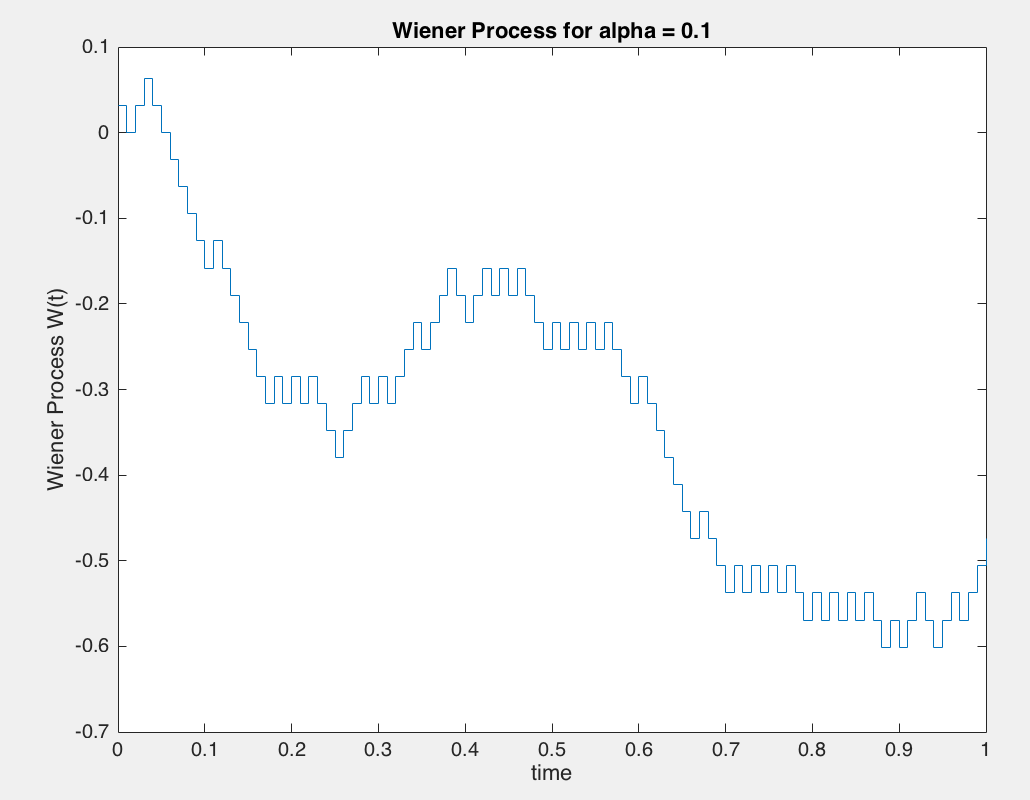
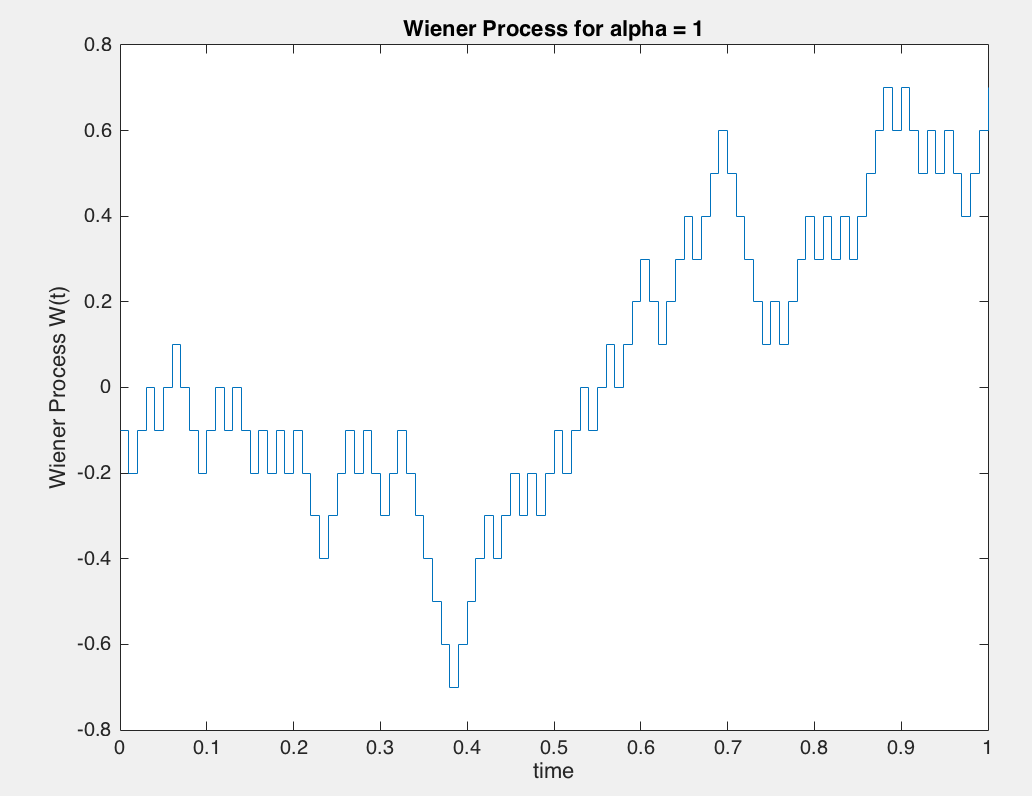
stairs(t,X)

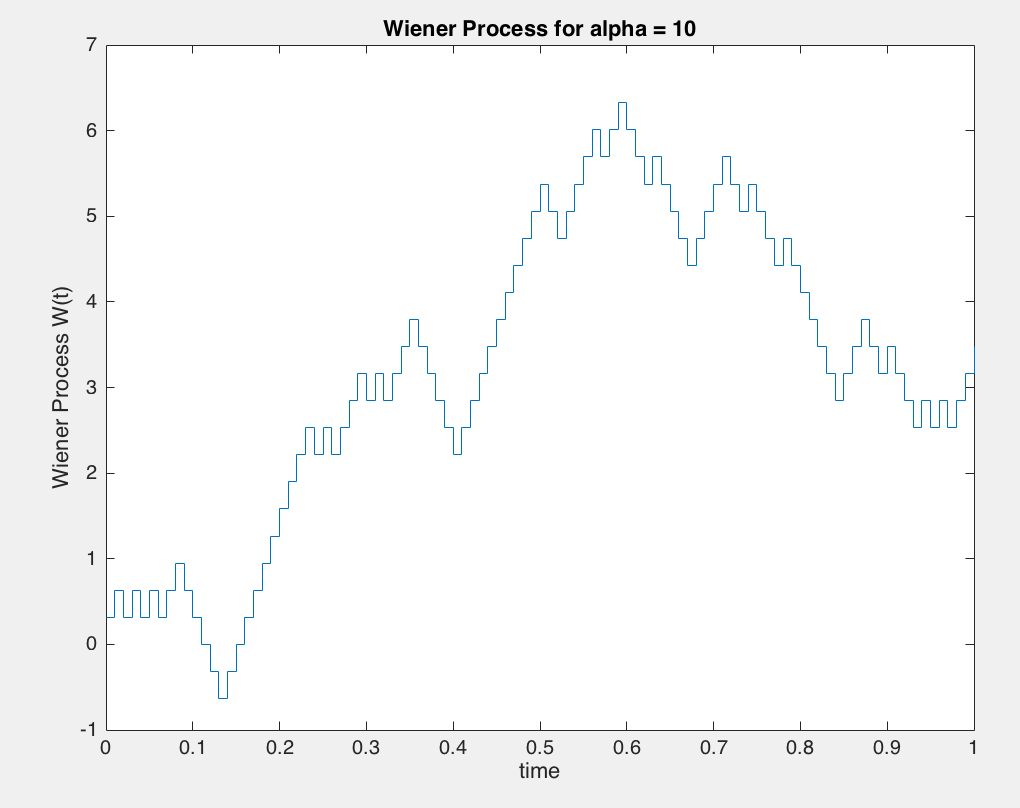
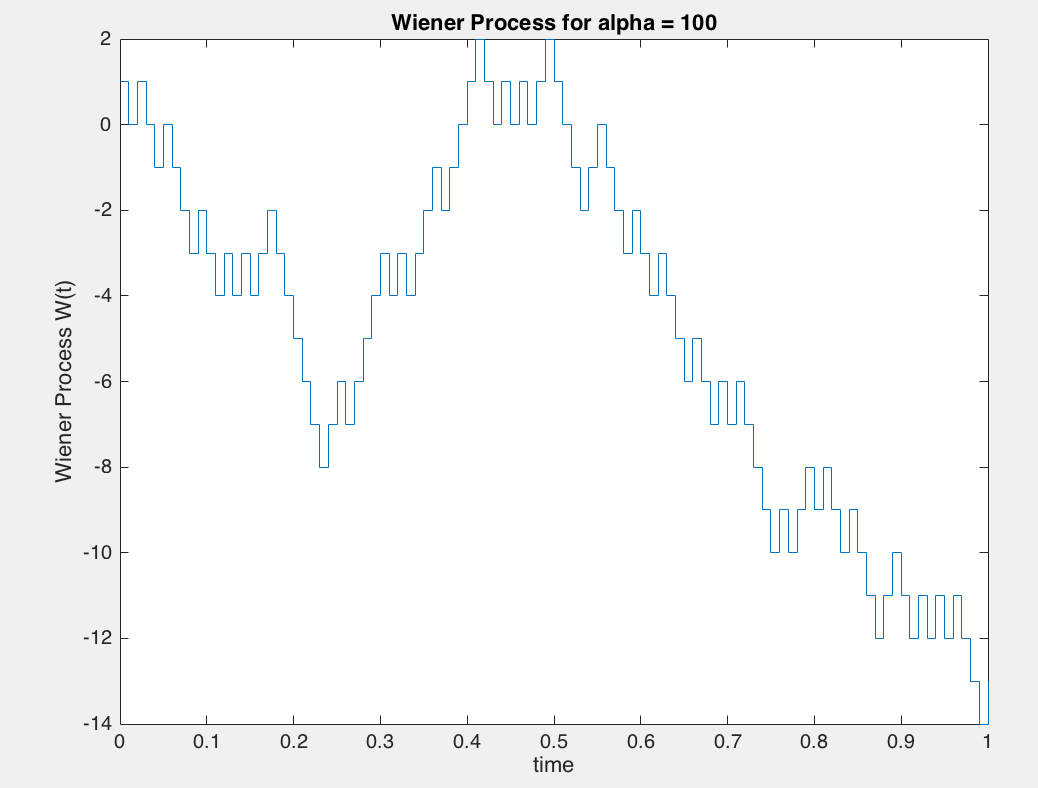
title('Wiener Process for alpha = 10, T = 0.01')

ylabel('Wiener Process W(t)')

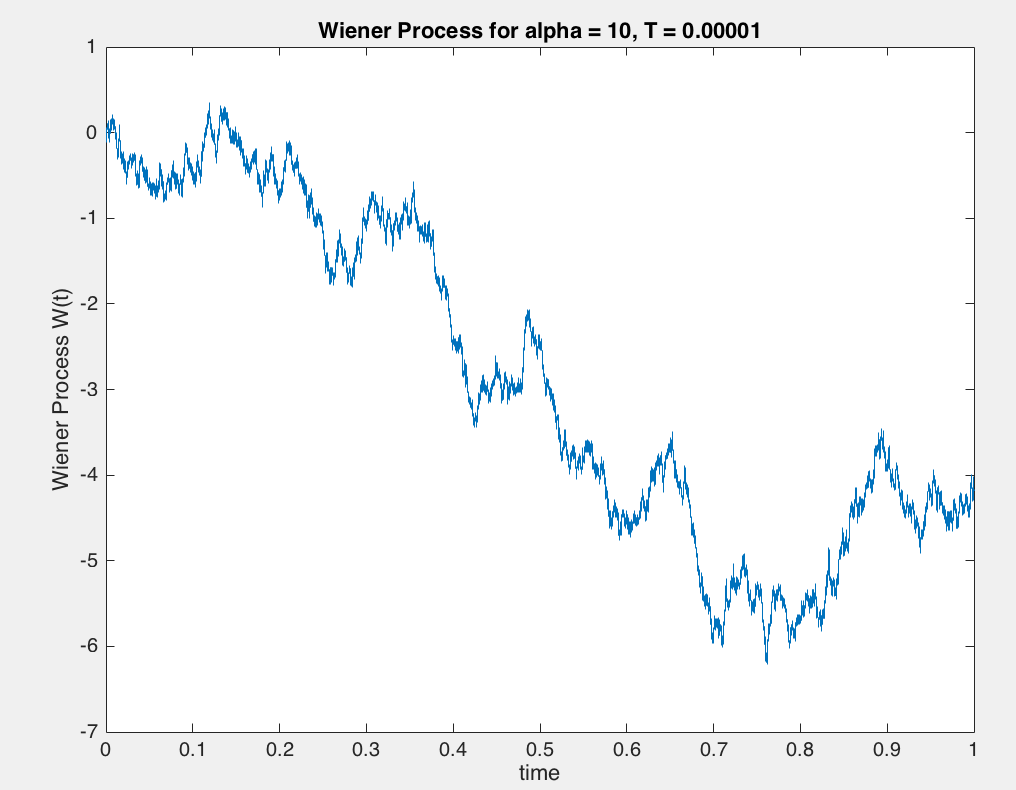
xlabel('time')

Plot sample paths of a wiener process by varying alpha, thus changing the step size of the jumps:

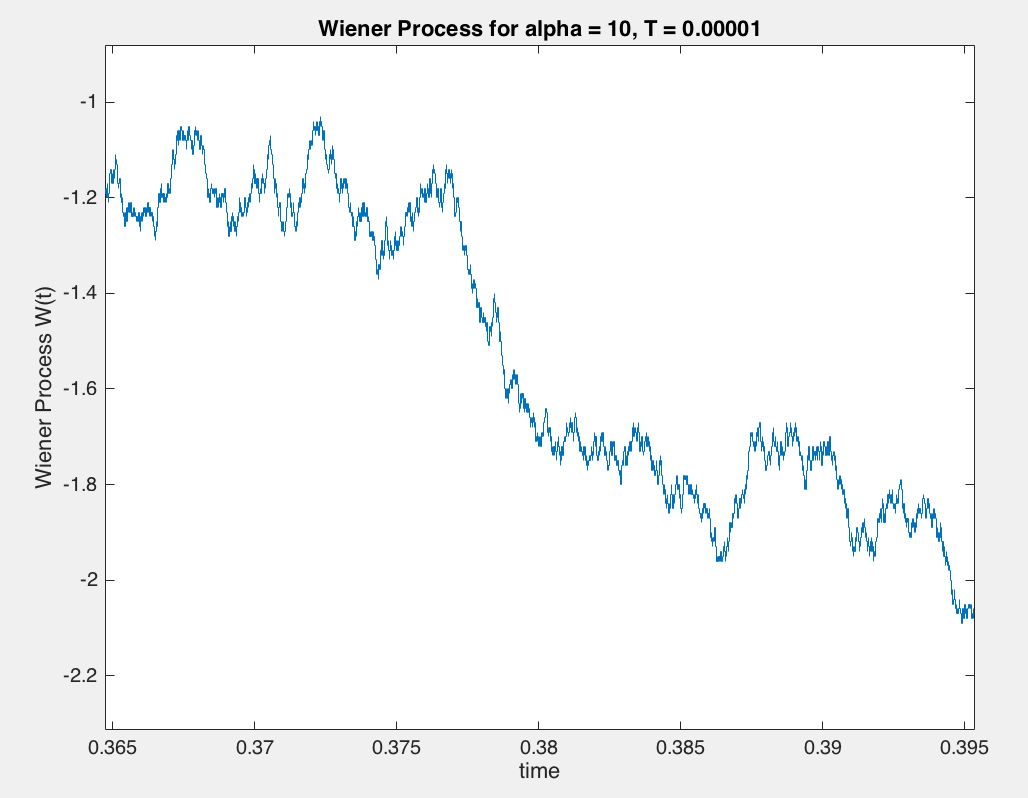
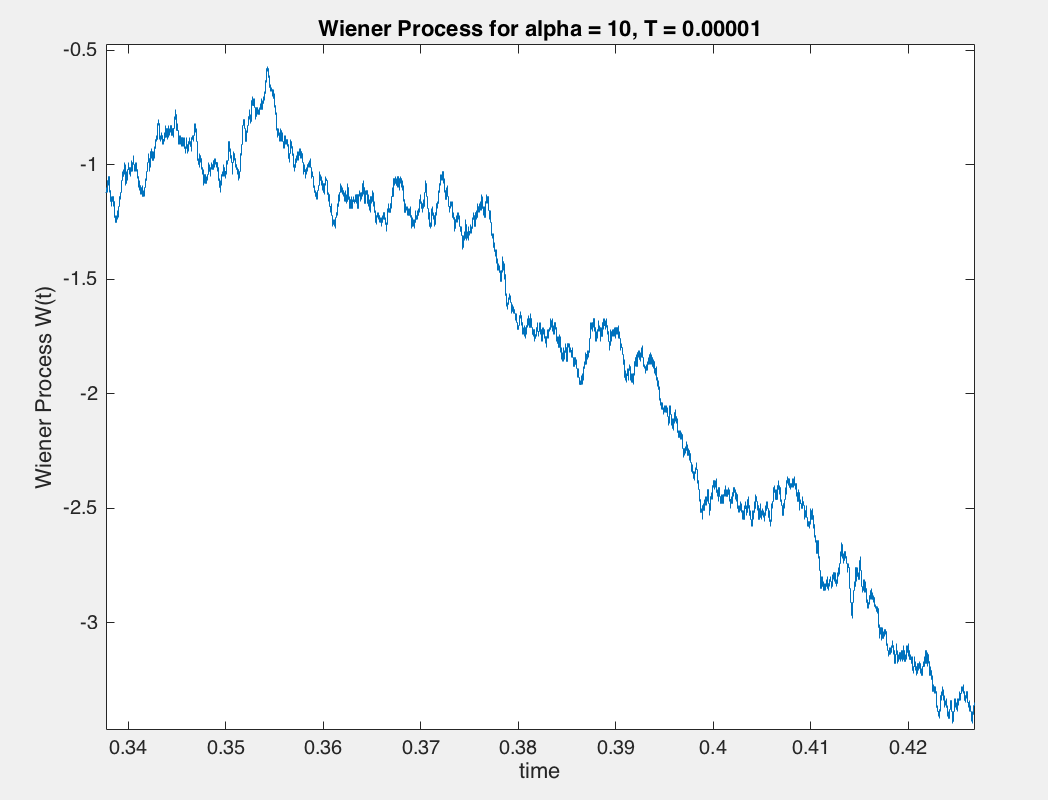
 

Plotting sample path for small value of T:

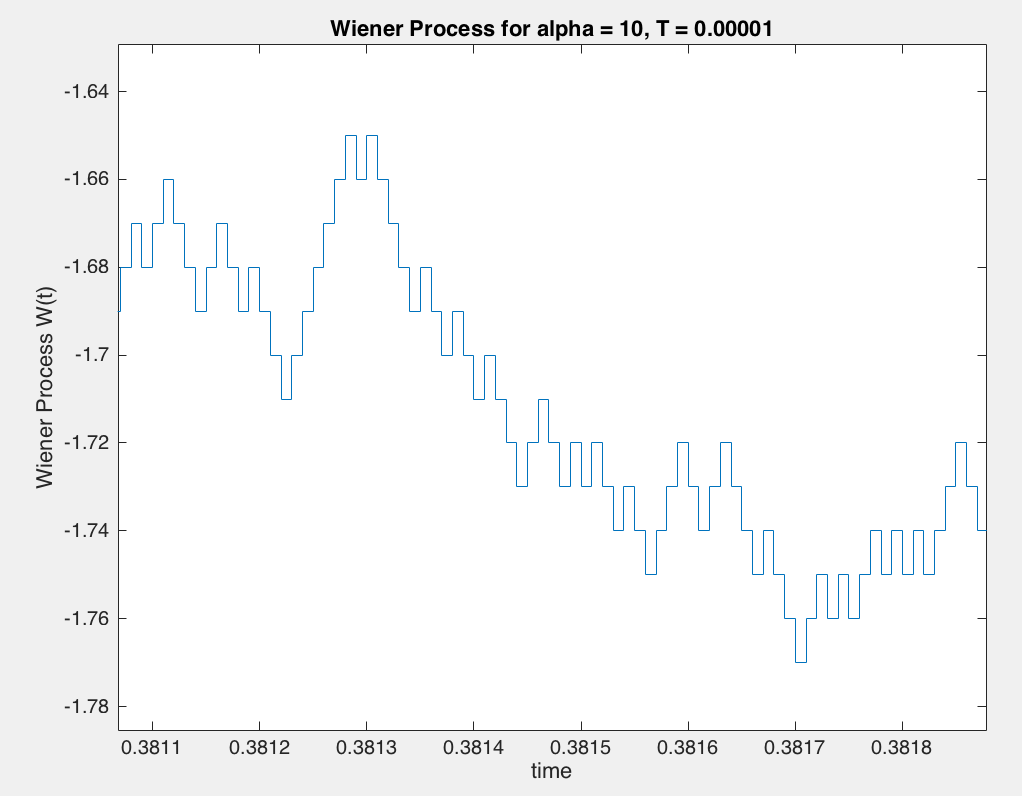


Zoomed in plots of the above original plot:

Zoom1: Zoom in on original Zoom2: Zoom in on Zoom1

Zoom3: Zoom in on Zoom2



Wiener process can be seen to follow fractal patterns meaning the zoomed in components replicate the general shape of the original zoomed out wiener process. Thus it can be said to be self-similar.

Code:

%% Plot wienerp.m sample paths

[X,t] = wienerp(10000,[0:0.1:1],0.001,2);

figure()

for i = 1:4

hold on

stairs(t,X(i,:))

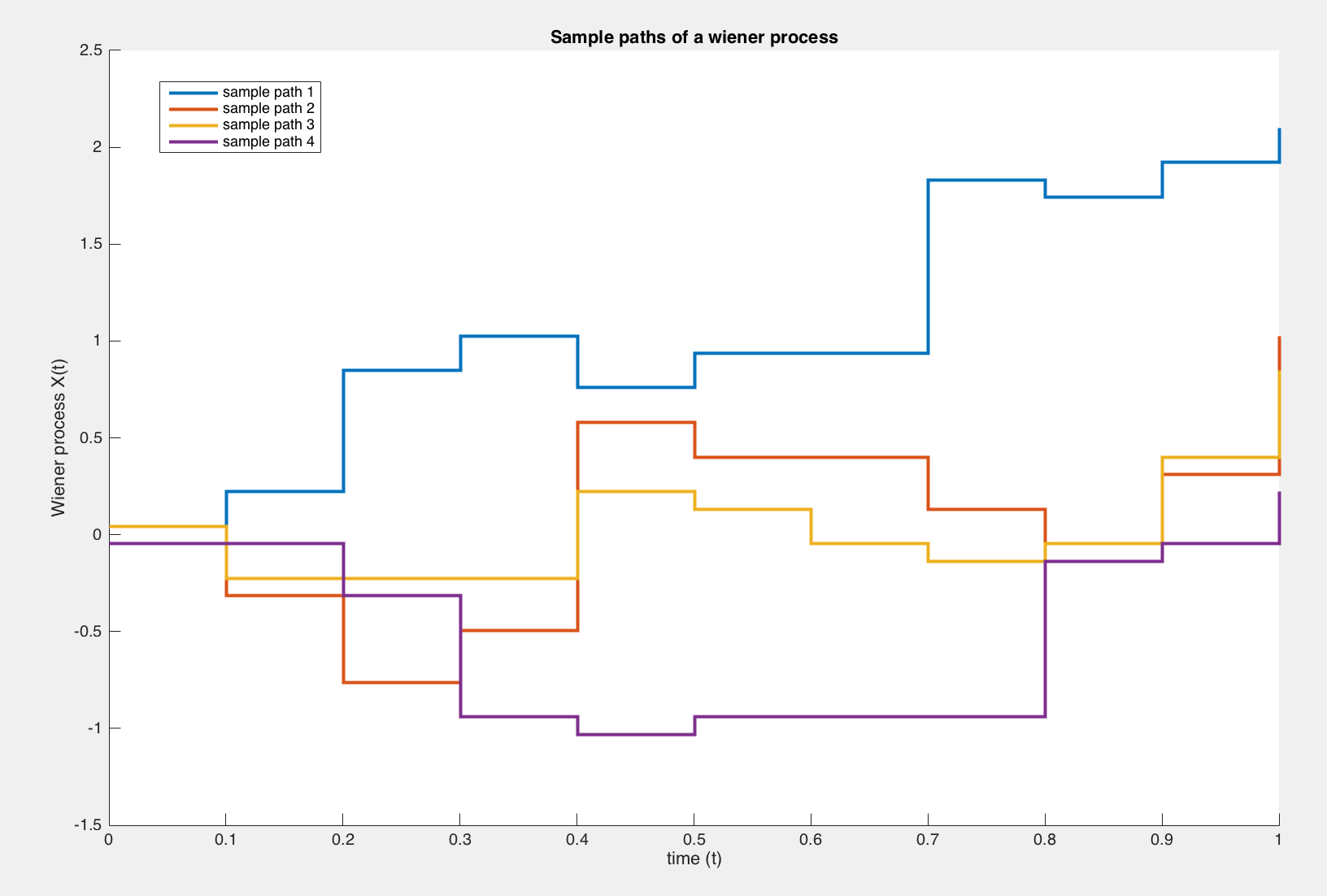
end

xlabel('time (t)')

ylabel('Wiener process X(t)')

title('Sample paths of a wiener process')

Plotting the wiener process using wienerp.m, by generating multiple sample paths of our approximate Weiner process at arbitrary sampling points.



Code:

%% Plot the mean and autocovariance function

m\_x = mean(X,1); % theoretical 0

autocov = mean((X - repmat(m\_x,10000,1)).^2); % theoretical alpha\*time

plot(t,mean(X,1))

axis([0 1 -0.5 0.5])

xlabel('time (t)')

ylabel('mean of wiener process')

title('mean of wiener process: alpha = 2, T = 0.001')

plot(t,autocov)

xlabel('time (t)')

ylabel('Variance of wiener process')

title('Autocovariance of wiener process: alpha = 2, T = 0.001')

Plotting the mean and autocovariance function of Wiener process using sample paths generated from wiener.m:

Mean Autocovariance

