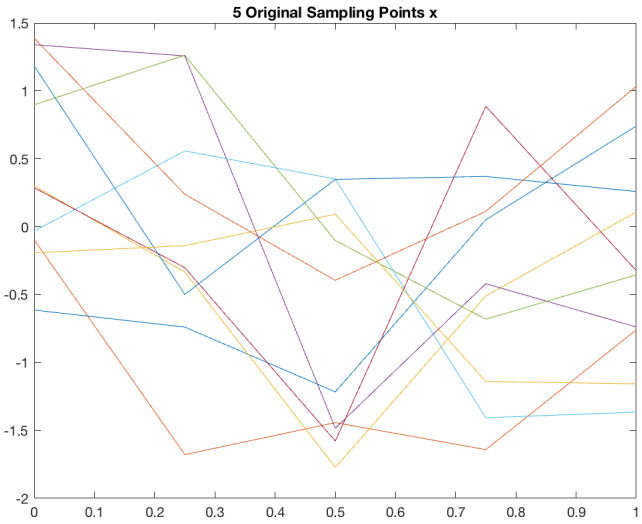
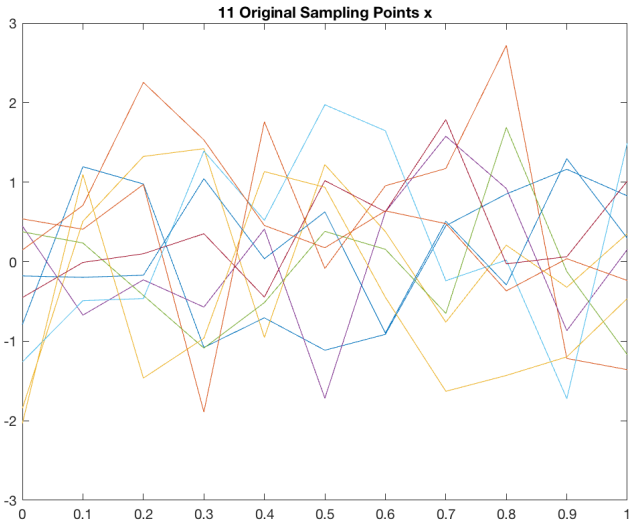
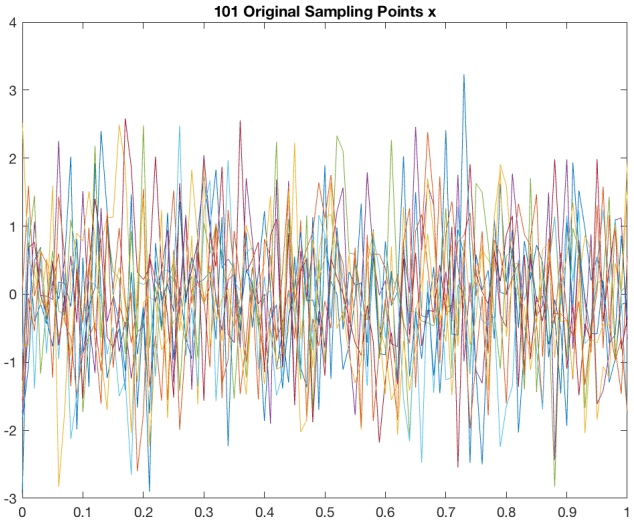
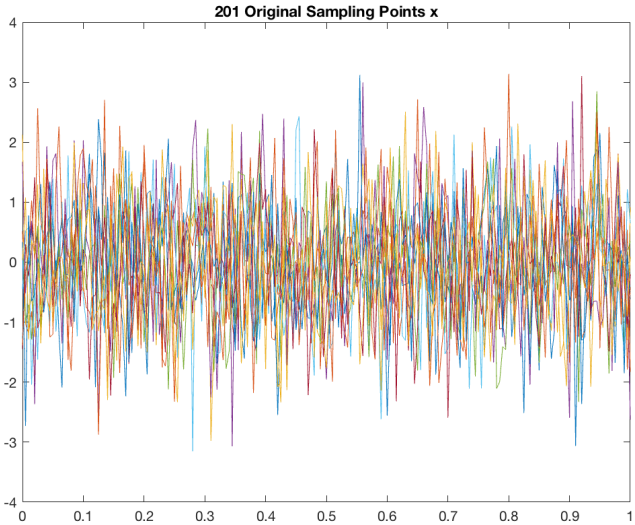
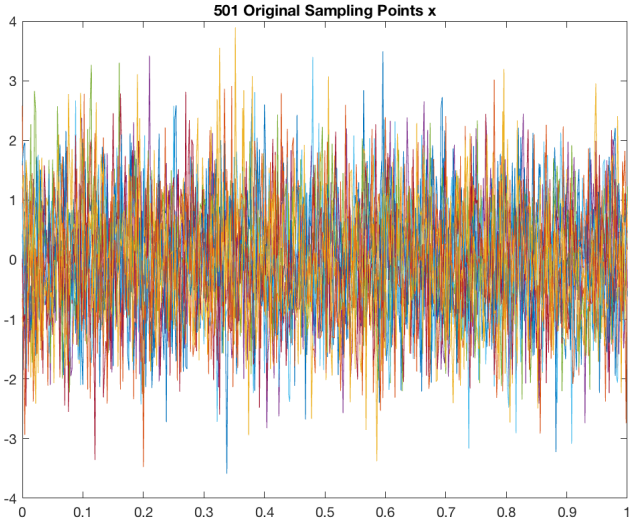
```
%% PROJECT CHAPTER 8
% TUAN NGUYEN
% Machine Learning
% Sampling From a GP prior
clear all; close all;

step = 0.1;
% Define one-dimensional input of L sampling points stored in 1XL matrix x
x = [0 : step : 1];

% Compute the covariance matrix for all pairs of sampling points in x
gam = 10.0;
L = length(x);
C = zeros(L);
for r = 1 : L
    for c = 1 : L
        C(r, c) = exp(gam * (x(r) - x(c))^2);
    end
end

% Add a small constant to the diagonal elements of the covariance matrix
C = C + 1e-6 * eye(L);

% Sample the Gaussian with zero mean and covariance equal to the covariance matrix
y = gausssamp(repmat(0, L, 1), C, 10)
figure(1);

% Plot the sampling points from the Gaussian
plot(x, y)
ti = sprintf('%d Original Sampling Points x', L);
title(ti);

%% Function samples from a Gaussian distribution with a given covariance
function g = gausssamp(mu, sigma, N, sigmachol)
if ~exist('sigmachol')
    [sigmachol, p] = chol(sigma);
    sigmachol = sigmachol';
end
q = randn(length(mu), N);
g = repmat(mu, 1, N)+ sigmachol * q;
g = g';
end
```

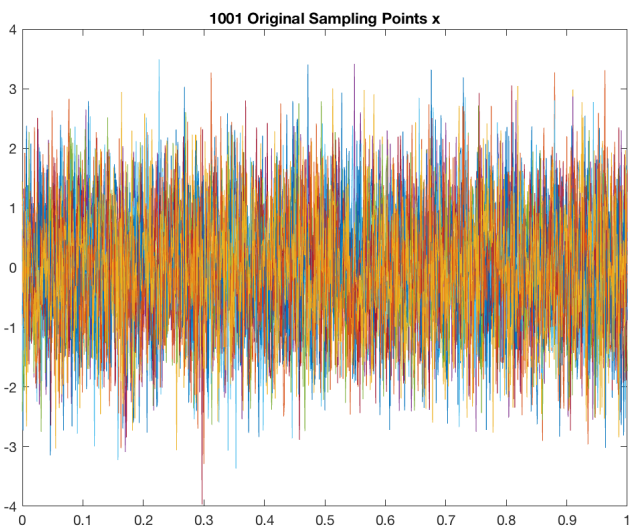

5 Original Sampling Points x


11 Original Sampling Points x


101 Original Sampling Points x


201 Original Sampling Points x


501 Original Sampling Points x


1001 Original Sampling Points x