**Problem 1:**

(a)

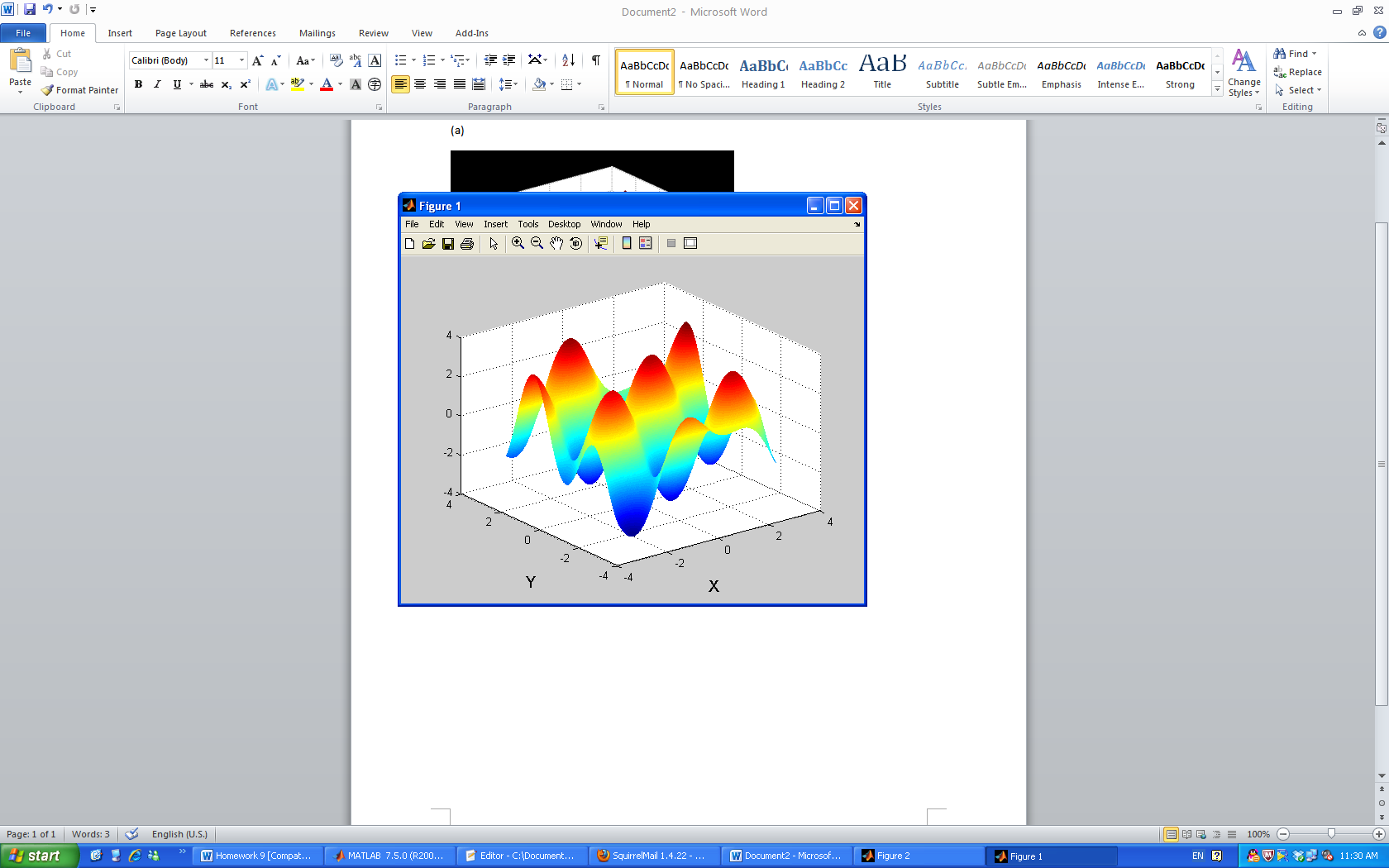


Figure 1: Mesh plot for the function

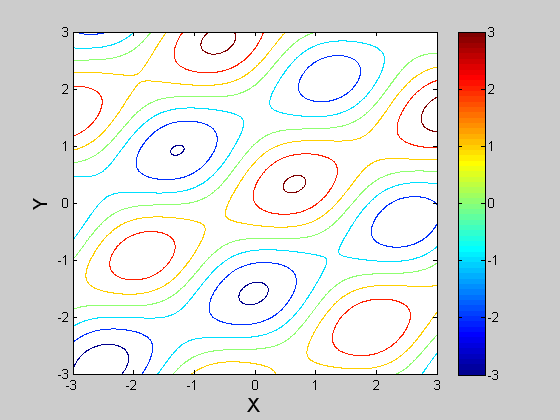
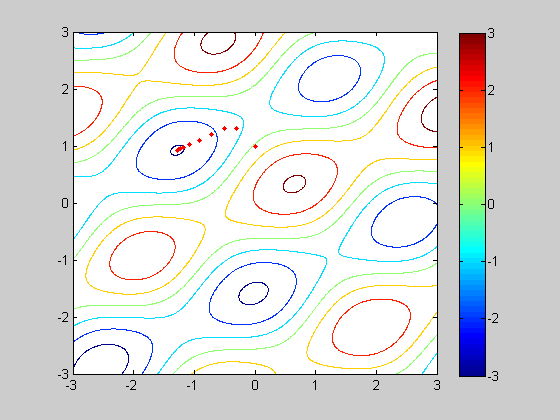


Figure 2: Contour plot for the function

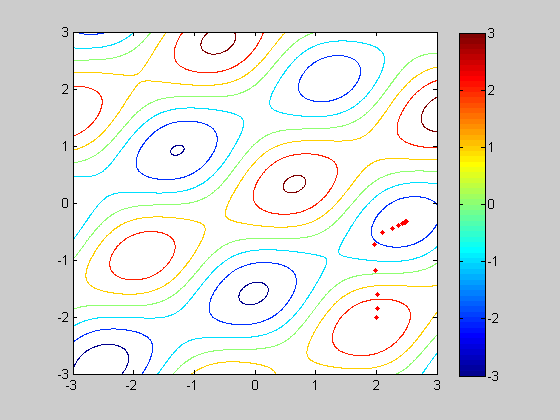
(b) Find the local minima using Gradient Method:

**Alpha = 0.1**

Initial point [0,1]:



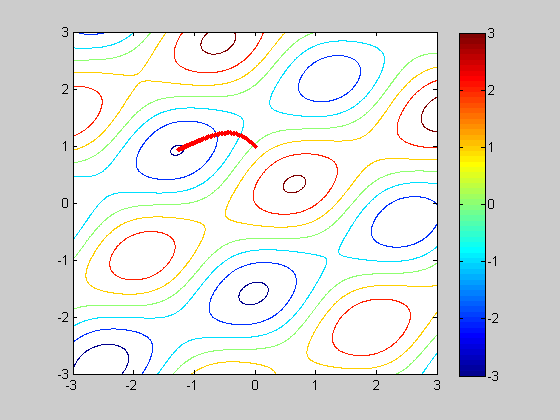
Initial point [2,-2]:



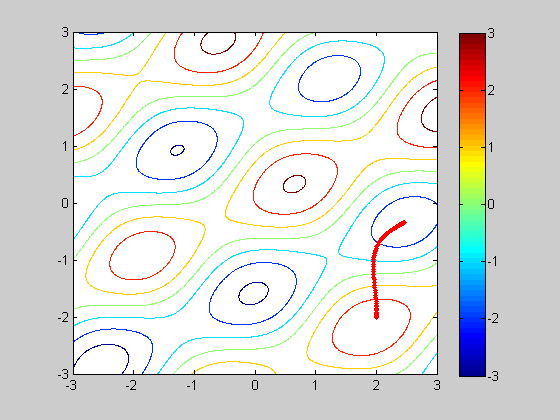


**Alpha = 0.01**

Initial point [0,1]:

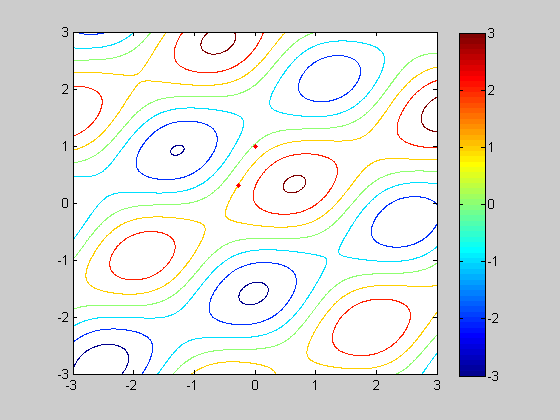


Initial point [2,-2]:

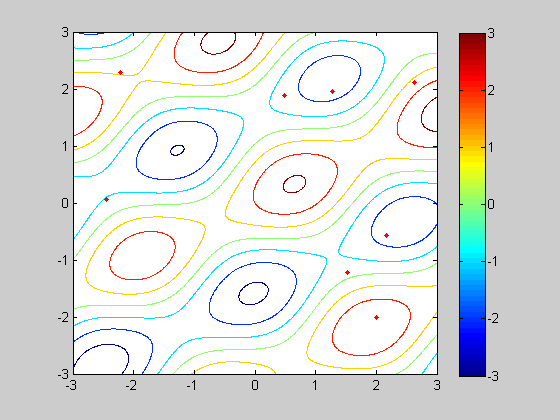


**Alpha = 1**

Initial point [0,1]:



Initial point [2,-2]:



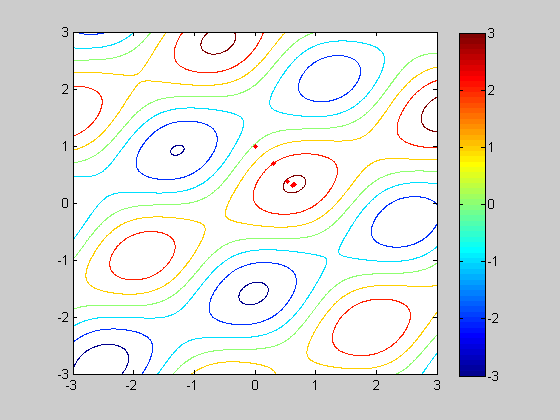
Comment:

1. I try different initial points; using Gradient algorithm converges to the local minimum correctly depending on the initial values.
2. Then I try different step size Alpha = 0.1, 0.01, 1. For the smaller the step size Alpha=0.01 or 0.1, it converges to the local minimum correctly. But for the large step size, this method could not find the correct local minimum because the steps are too large and miss the minimum.

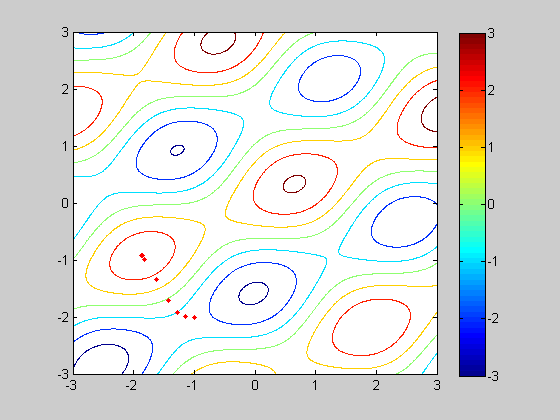
(c) Find the local maxima using Gradient Method:

**Alpha = 0.1**

Initial point [0,1]:

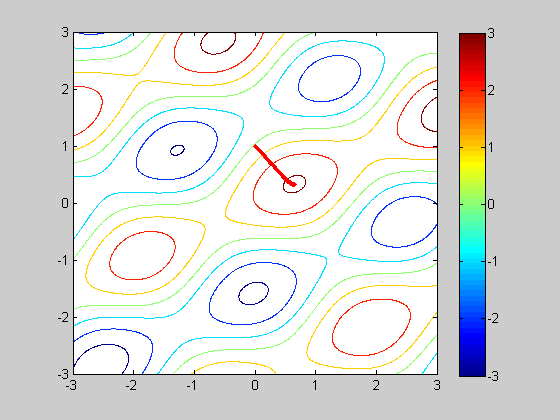


Initial point [-1,-2]:

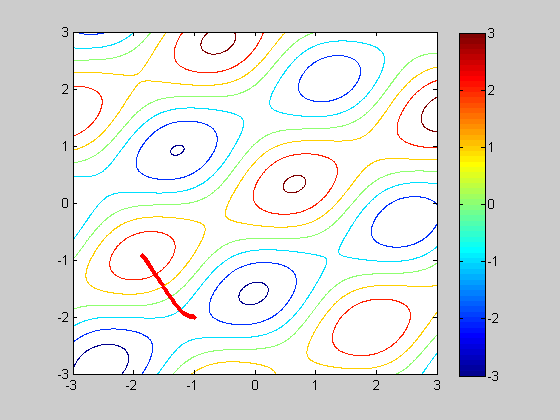


**Alpha = 0.01**

Initial point [0,1]:

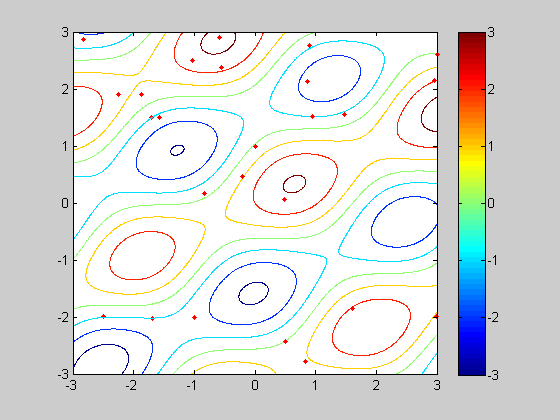


Initial point [-1,-2]:

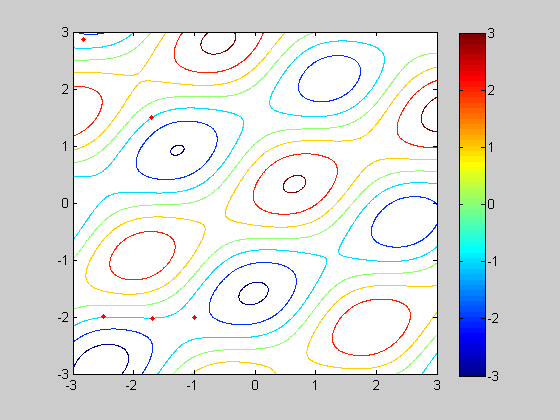


**Alpha = 1**

Initial point [0,1]:



Initial point [-1,-2]:



Comment:

1. I try different initial points; using Gradient algorithm converges to the local maximum correctly depending on the initial values.
2. Then I try different step size Alpha = 0.1, 0.01, 1. For the smaller the step size Alpha=0.01 or 0.1, it converges to the local maximum correctly. But for the large step size, this method could not find the correct local maximum because the steps are too large and miss the maximum.

**Code for Problem 1:**

clear all; close all;

h = inline('sin(2\*x+y)+2.\*cos(x-2\*y)+0.1\*(x+y)','x','y');

dhx = inline('2\*cos(2\*x+y)-2\*sin(x-2\*y)+0.1','x','y');

dhy = inline('cos(2\*x+y)+4\*sin(x-2\*y)+0.1','x','y');

x = -3:0.01:3;

y = -3:0.01:3;

for i=1:length(x)

z(:,i)=h(x(i),y);

end

figure(1);

mesh(x,y,z);

xlabel('X','fontsize',14);

ylabel('Y','fontsize',14);

figure(2);

contour(x,y,z);

xlabel('X','fontsize',14);

ylabel('Y','fontsize',14);

colorbar;

alpha = 1;

%try initial point

theta(:,:,1) = [-1,-2];

h\_value(1) = h(theta(1,1,1),theta(1,2,1));

for i = 2:100

theta(:,:,i) = theta(:,:,i-1) + alpha\*[dhx(theta(1,1,i-1),theta(1,2,i-1)),dhy(theta(1,1,i-1),theta(1,2,i-1))];

h\_value(i) = h(theta(1,1,i),theta(1,2,i));

end;

figure(3);

[X,Y] = meshgrid(x,y);

Z=h(X,Y);

contour(X,Y,Z);

colorbar;

hold on;

for i=1:100

plot(theta(1,1,i),theta(1,2,i),'r.');

hold on;

end;

**Problem 2:**

(a)

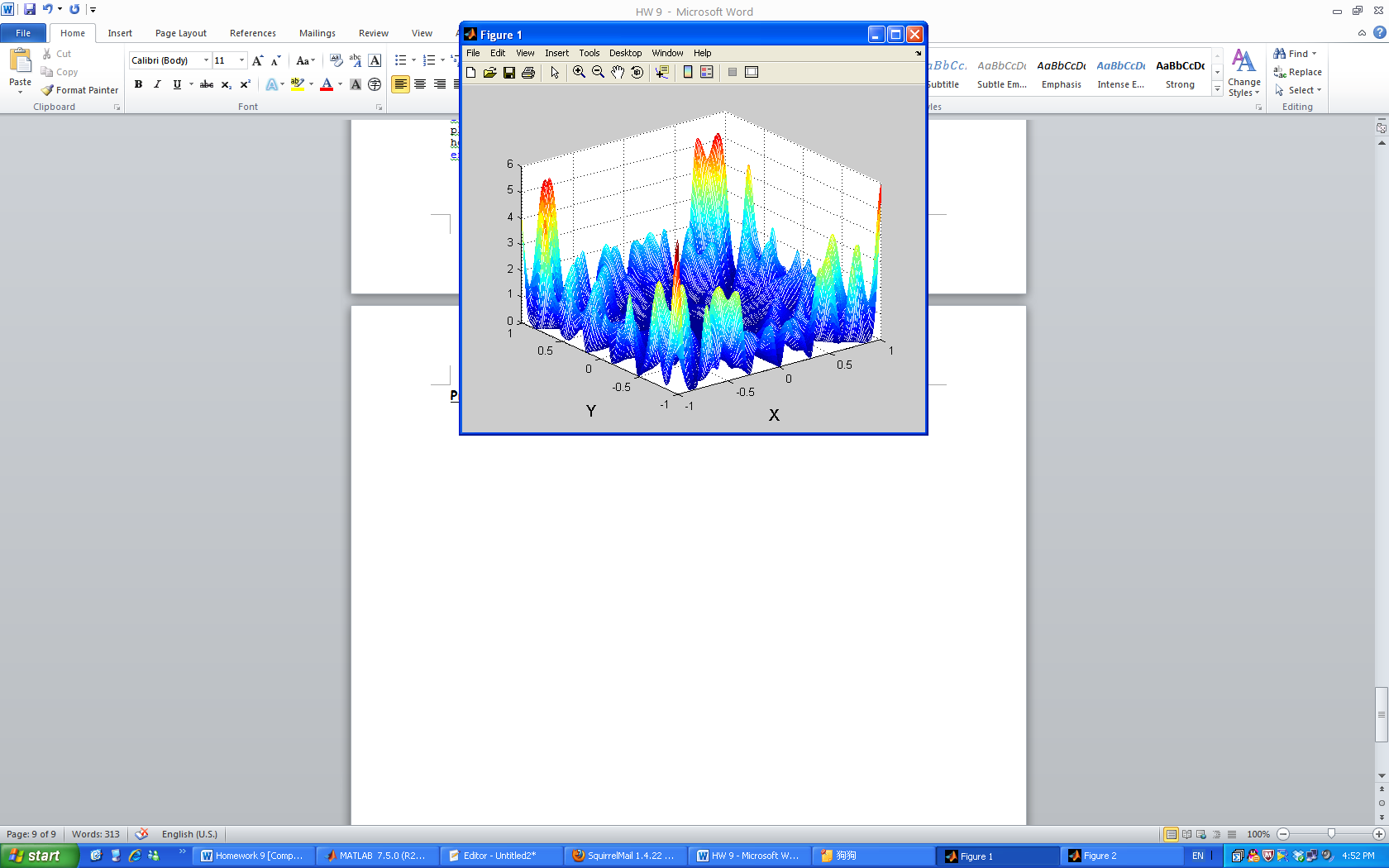


Figure 1: Mesh plot for the function

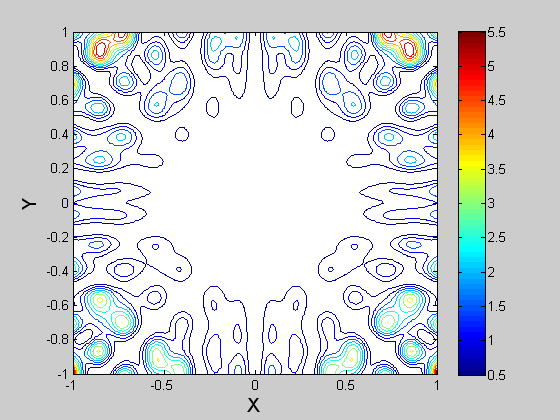
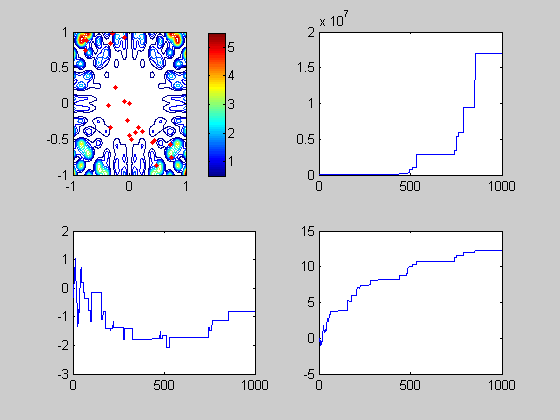


Figure 2: Contour plot for the function

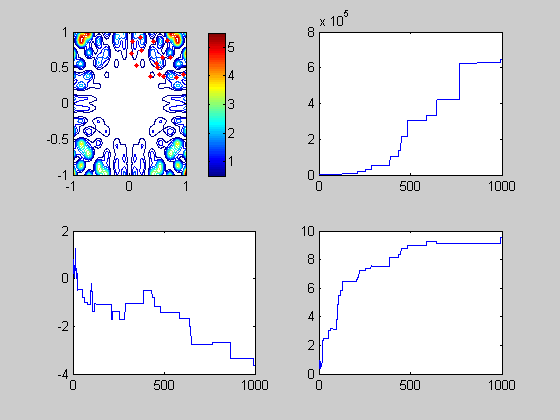
(b)

**Sigma = 0.1**

Initial point [0,0]:

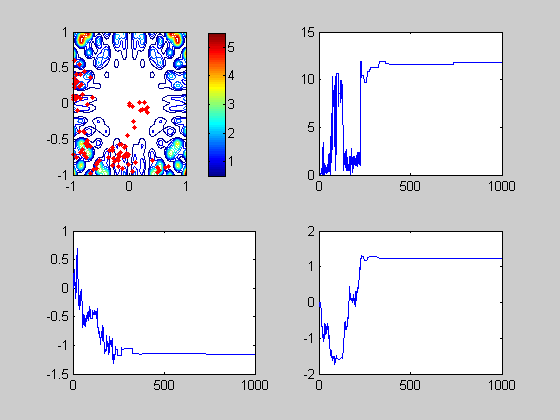


Initial point [0.5,0.5]:

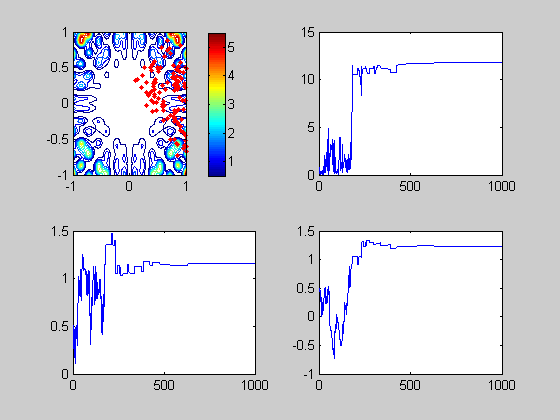


**Sigma = 0.01**

Initial point [0,0]:

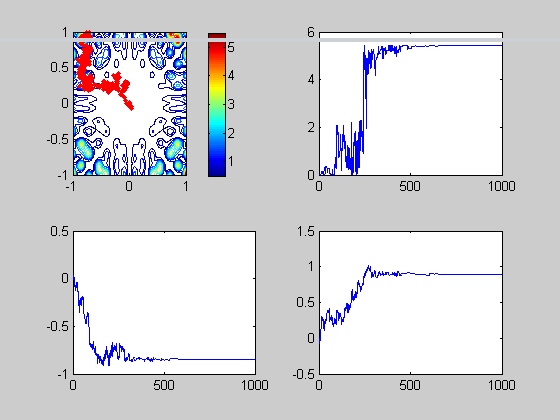


Initial point [0.5,0.5]:

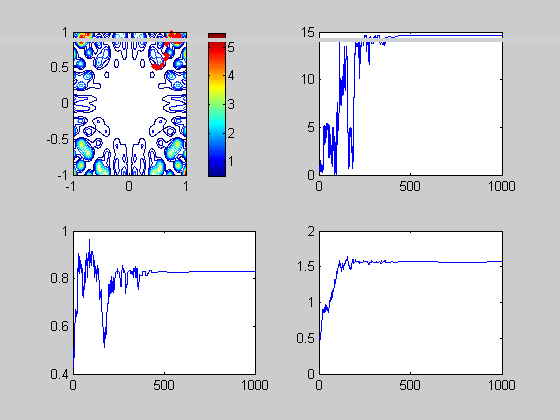


**Sigma = 0.001**

Initial point [0,0]:



Initial point [0.5,0.5]:



Comment:

1. When Sigma is small, the points spread out very slow to the maximize place with higher altitude, less chance to jump out. Maybe trapped in.
2. When Sigma is larger, the points spread out faster to the maximize place with higher altitude, more chance to jump out.

**Code for Problem 2:**

clear all; close all;

h = inline('(x.\*sin(20\*y)+y.\*sin(20\*x)).^2.\*cosh(sin(10\*x).\*x) + (x.\*cos(10\*y)-y.\*sin(10\*x)).^2.\*cosh(cos(20\*y).\*y)','x','y');

alpha=0.99;

x = -1:0.01:1;

y = -1:0.01:1;

for i=1:length(x)

z(:,i)=h(x(i),y);

end

figure(1);

mesh(x,y,z);

xlabel('X','fontsize',14);

ylabel('Y','fontsize',14);

figure(2);

contour(x,y,z);

xlabel('X','fontsize',14);

ylabel('Y','fontsize',14);

colorbar;

% %% Optimization using M-H Simulated Annealing

T(1) = 10;

sigma = [.001 0; 0 .001];

%initial point

theta(1,:)=[0.5 0.5];

for i=2:1000;

T(i) = T(i-1)\*alpha;

zeta = mvnrnd(theta(i-1,:),sigma);

rho = min(1,exp((h(zeta(1),zeta(2))-h(theta(i-1,1),theta(i-1,2)))/T(i)));

U = rand;

if (U>rho)

theta(i,:) = theta(i-1,:);

else

theta(i,:) = zeta;

end

end

figure (5);

subplot(2,2,1)

[X,Y]=meshgrid(x,y);

Z = h(X,Y);

contour(X,Y,Z); hold on;

colorbar;

for i=1:1000

plot(theta(i,1),theta(i,2),'r.');

hold on;

end;

%plot the function

subplot(2,2,2)

plot(1:1000,h(theta(:,1),theta(:,2)));

subplot(2,2,3)

plot(1:1000,theta(:,1));

subplot(2,2,4)

plot(1:1000,theta(:,2))