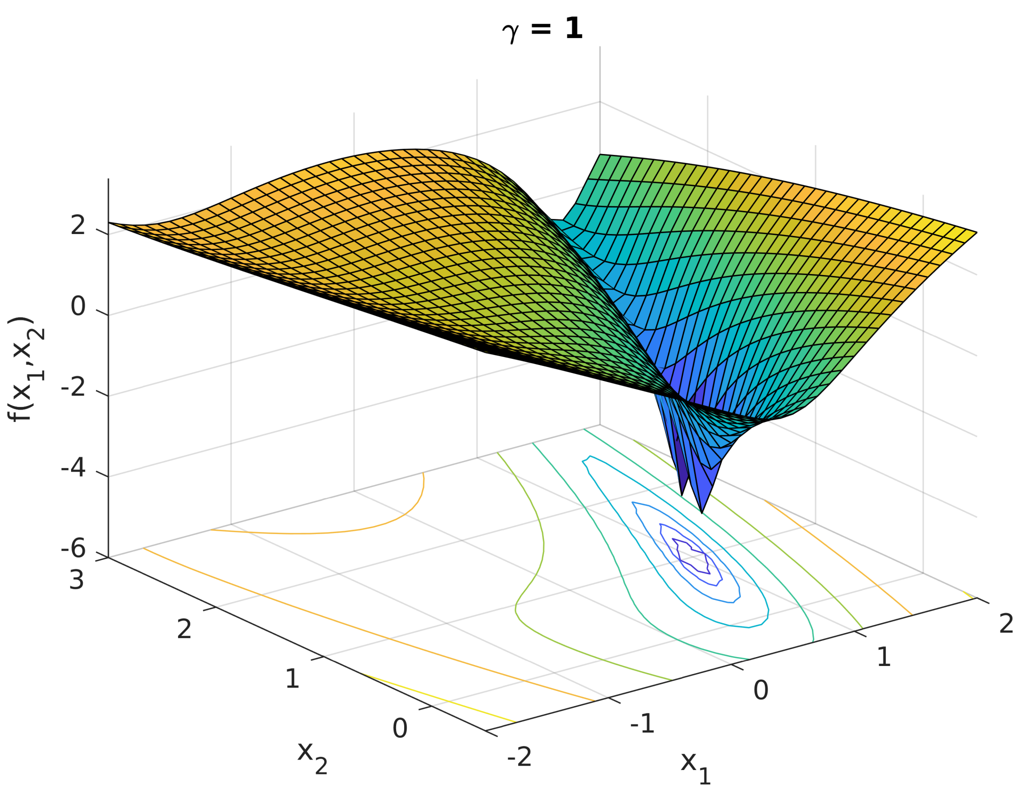
Problem 2

```
clc;
clear;
close all;
```

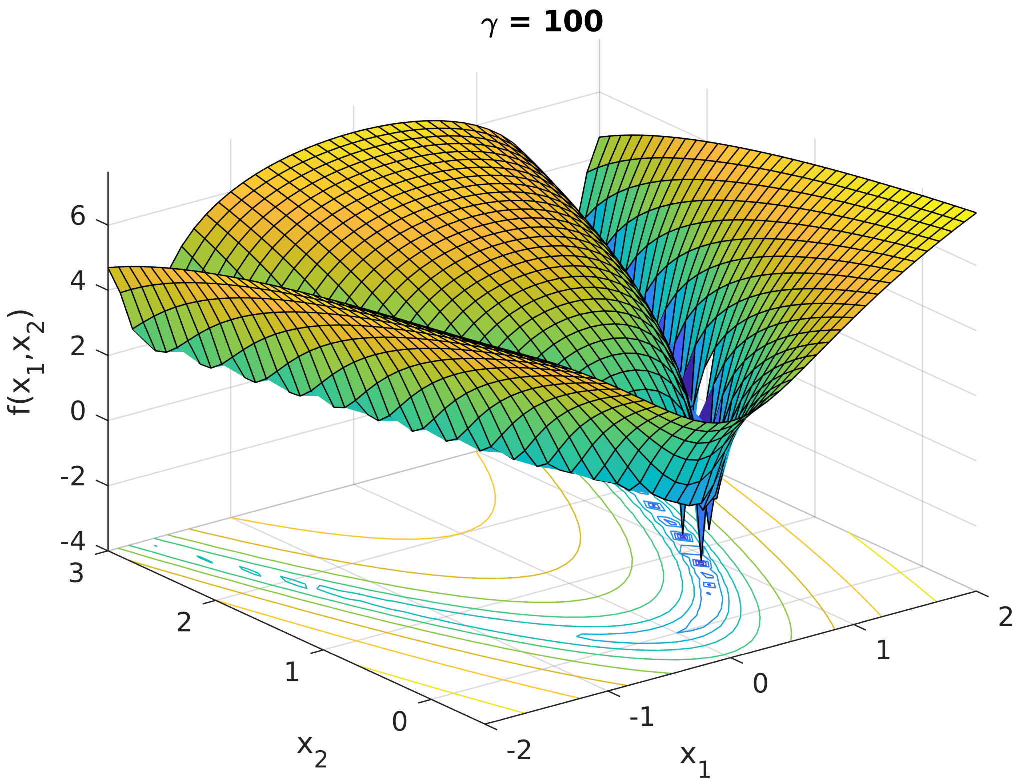
Part A - See Written Portion

```
% Define function, gradient, and hessian
f = @(x1,x2,gamma) gamma*(x1.^2-x2).^2+(x1-1).^2;
g = @(x1,x2,gamma) [4*gamma*x1.*(x1.^2-x2)+2*x1-1; -2*gamma*(x1.^2-x2)];
H = @(x1,x2,gamma) [4*gamma*(x1.^2-x2)+8*gamma*x1.^2+2, -4*gamma*x1;...
                    -4*gamma*x1, 2*gamma];

% Draw 3D contour for gamma = 1 and gamma = 100
[xx1, xx2] = meshgrid(-2:0.1:2,-0.5:0.1:3);
figure;
surfc(xx1,xx2,log(f(xx1,xx2,1)));
title('\gamma = 1'); xlabel('x_{1}'); ylabel('x_{2}'); zlabel('f(x_{1},x_{2})');
```




```
figure;
surfc(xx1,xx2,log(f(xx1,xx2,100)));
```

```
title('\gamma = 100'); xlabel('x_{1}'); ylabel('x_{2}'); zlabel('f(x_{1},x_{2})');
```



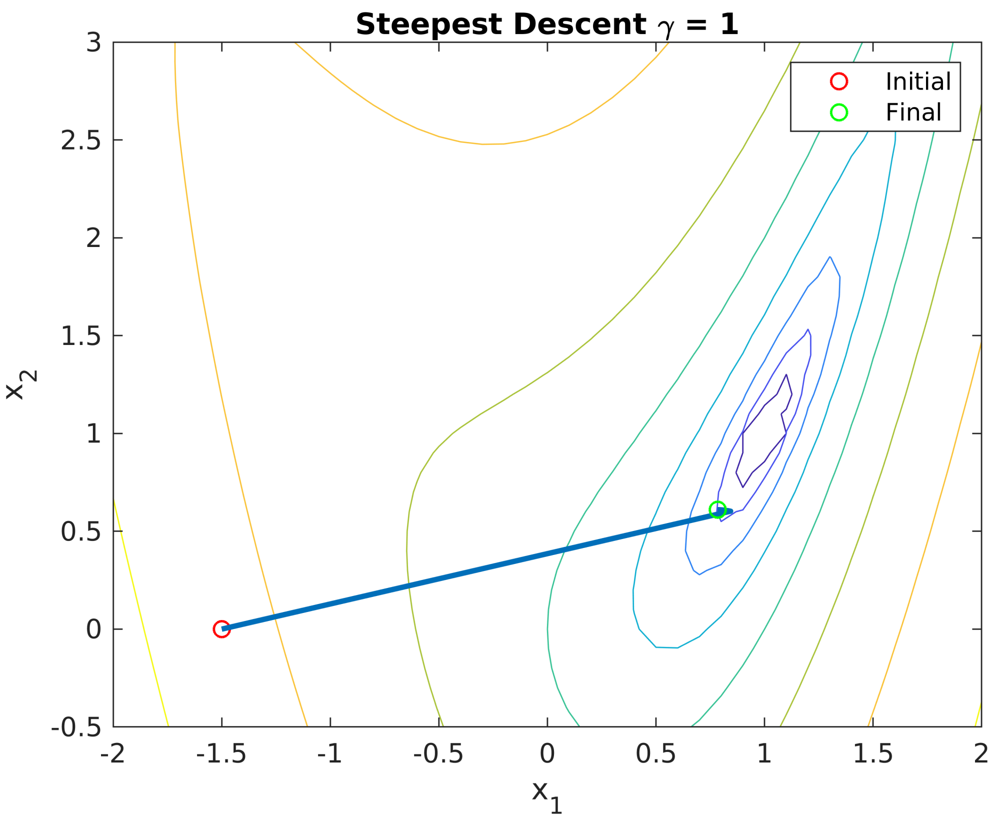
Part B

At $\gamma = 1$, the function has few local minima concentrated in one area. Steepest descent seems to perform much better than Newton at this parameter as can be seen by the sequence of iterates for each method. Steepest descent seems to jump immediately to the optimum, while Newton seems to step down each contour level one-by-one.

At $\gamma = 100$, the function seems to have many local minima spread out across a large area. Here, steepest descent seems to perform worse, where it jumps to many local minima points before stopping. Newton's method seems to proceed in a much more controlled manner.

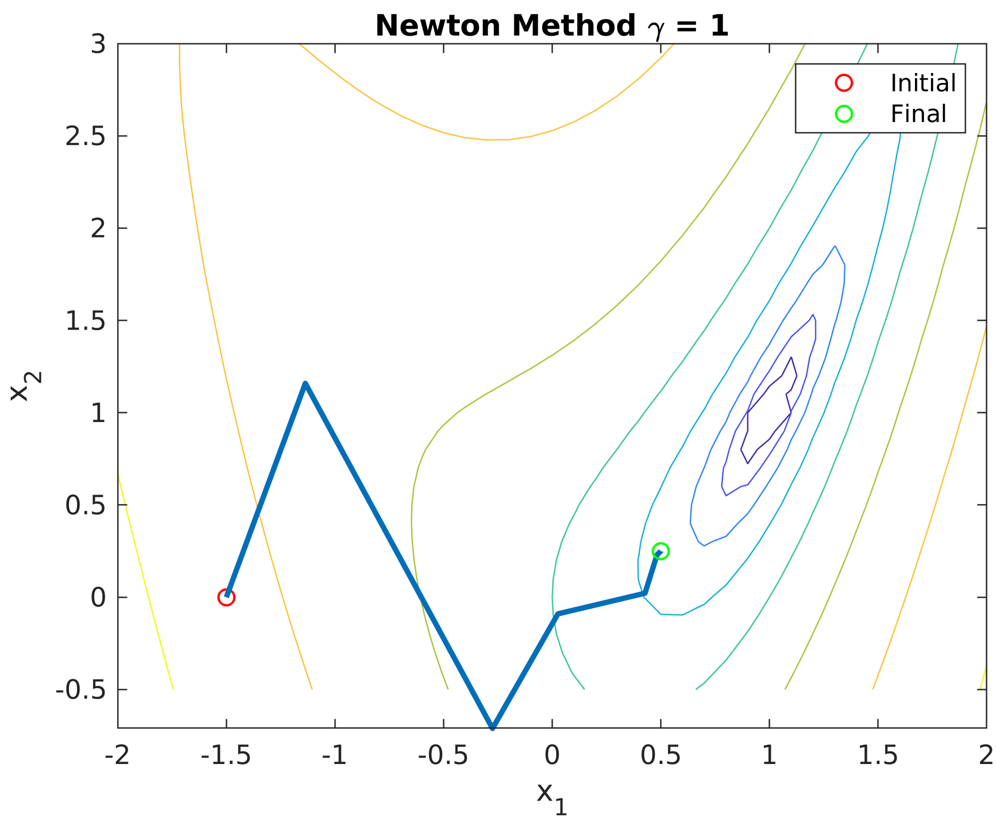
```
% Do steepest descent with gamma = 1
gamma = 1; % set gamma
v = zeros(1000,2); % save each update
v(1,:) = [-1.5,0]; % initial point

% Draw function contour
[xx1, xx2] = meshgrid(-2:0.1:2,-0.5:0.1:3);
figure;
contour(xx1,xx2,log(f(xx1,xx2,gamma)));
hold on;

% plot initial point
s = plot(v(1,1),v(1,2),'or','LineWidth',2);
```

```
% run steepest descent and draw each iteration (do exact line search)
for n=1:999
    pk = -g(v(n,1),v(n,2),gamma);
    fes = @(a) f(v(n,1)+a*pk(1),v(n,2)+a*pk(2),gamma);
    alpha = fminbnd(fes,1e-10,1);
    v(n+1,:) = v(n,:) + (alpha*pk)';
end
plot(v(:,1),v(:,2),'LineWidth',2);

% plot final point
e = plot(v(n+1,1),v(n+1,2),'og','LineWidth',2);
hold off;

% plot config
legend([s,e],{'Initial','Final'})
title('Steepest Descent \gamma = 1');
xlabel('x_{1}'); ylabel('x_{2}');
```



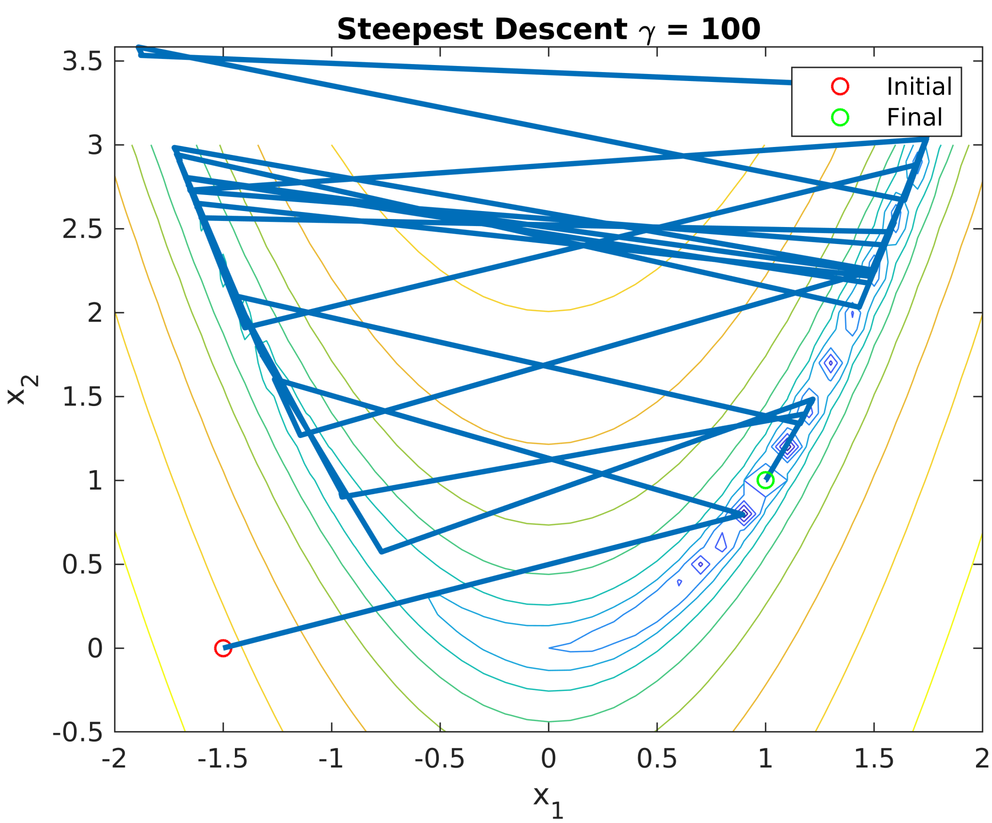

```
% Do Newton's method with gamma = 1
gamma = 1;
v = zeros(1000,2);
v(1,:) = [-1.5,0];

% Draw function contour
[xx1, xx2] = meshgrid(-2:0.1:2,-0.5:0.1:3);
figure;
contour(xx1,xx2,log(f(xx1,xx2,gamma)));
hold on;
```

```
% plot initial point
s = plot(v(1,1),v(1,2),'or','LineWidth',2);

% run Newton's method and draw each iteration (do exact line search)
for n=1:999
    pk = -H(v(n,1),v(n,2),gamma)\g(v(n,1),v(n,2),gamma);
    fes = @(a) f(v(n,1)+a*pk(1),v(n,2)+a*pk(2),gamma);
    alpha = fminbnd(fes,1e-10,1);
    v(n+1,:) = v(n,:) + (alpha*pk)';
end
plot(v(:,1),v(:,2),'LineWidth',2);

% plot final point
e = plot(v(n+1,1),v(n+1,2),'og','LineWidth',2);
hold off;

% plot config
legend([s,e],{'Initial','Final'})
title('Newton Method \gamma = 1');
xlabel('x_{1}'); ylabel('x_{2}');
```



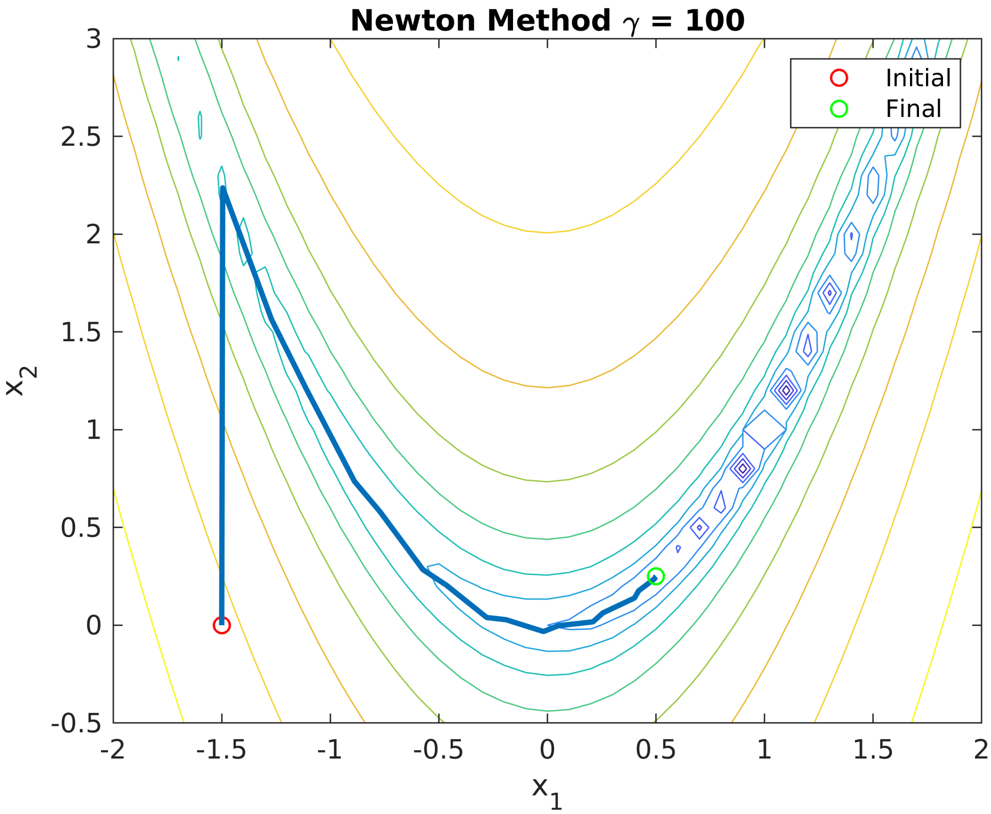

```
% Do steepest descent with gamma = 100
gamma = 100; % set gamma
v = zeros(1000,2); % save each update
v(1,:) = [-1.5,0]; % initial point

% Draw function contour
```

```
[xx1, xx2] = meshgrid(-2:0.1:2,-0.5:0.1:3);
figure;
contour(xx1,xx2,log(f(xx1,xx2,gamma)));
hold on;

% plot initial point
s = plot(v(1,1),v(1,2),'or','LineWidth',2);

% run steepest descent and draw each iteration (do exact line search)
for n=1:999
    pk = -g(v(n,1),v(n,2),gamma);
    fes = @(a) f(v(n,1)+a*pk(1),v(n,2)+a*pk(2),gamma);
    alpha = fminbnd(fes,1e-10,1);
    v(n+1,:) = v(n,:) + (alpha*pk)';
end
plot(v(:,1),v(:,2),'LineWidth',2);

% plot final point
e = plot(v(n+1,1),v(n+1,2),'og','LineWidth',2);
hold off;

% plot config
legend([s,e],{'Initial','Final'})
title('Steepest Descent \gamma = 100');
xlabel('x_{1}'); ylabel('x_{2}');
```




```
% Do Newton's method with gamma = 100
gamma = 100;
```

```
v = zeros(1000,2);
v(1,:) = [-1.5,0];

% Draw function contour
[xx1, xx2] = meshgrid(-2:0.1:2,-0.5:0.1:3);
figure;
contour(xx1,xx2,log(f(xx1,xx2,gamma)));
hold on;

% plot initial point
s = plot(v(1,1),v(1,2),'or','LineWidth',2);

% run Newton's method and draw each iteration (do exact line search)
for n=1:999
    pk = -H(v(n,1),v(n,2),gamma)\g(v(n,1),v(n,2),gamma);
    fes = @(a) f(v(n,1)+a*pk(1),v(n,2)+a*pk(2),gamma);
    alpha = fminbnd(fes,1e-10,1);
    v(n+1,:) = v(n,:) + (alpha*pk)';
end
plot(v(:,1),v(:,2),'LineWidth',2);

% plot final point
e = plot(v(n+1,1),v(n+1,2),'og','LineWidth',2);
hold off;

% plot config
legend([s,e],{'Initial','Final'})
title('Newton Method \gamma = 100');
xlabel('x_{1}'); ylabel('x_{2}');
```



Newton Method γ = 100
Initial
Final
3
2.5
2
1.5
1
0.5
0
-0.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
x_1
x_2