Part I

1. $\boldsymbol{J}(\boldsymbol{x}) = \begin{pmatrix} -x_2\sin(x_1) & \cos(x_1) \\ 0 & 4x_2 \end{pmatrix}$
2. $\boldsymbol{g}(\boldsymbol{x}^{(0)}) = \begin{pmatrix} \cos(1) \\ 2-1 \end{pmatrix} = \begin{pmatrix} 0.54 \\ 1 \end{pmatrix} \neq \begin{pmatrix} 0 \\ 0 \end{pmatrix}$
3. $\boldsymbol{x}^{(1)} = \boldsymbol{x}^{(0)} - \boldsymbol{J}^{-1}(\boldsymbol{x}^{(0)})\boldsymbol{g}(\boldsymbol{x}^{(0)}) = \begin{pmatrix} 1 \\ 1 \end{pmatrix} - \begin{pmatrix} -1.1884 & 0.1605 \\ 0 & 0.25 \end{pmatrix}^{-1} \begin{pmatrix} \cos(1) \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{1.48} \\ \mathbf{0.75} \end{pmatrix}$
   a. $\boldsymbol{g}(\boldsymbol{x}^{(1)}) = \begin{pmatrix} \mathbf{0.0668} \\ \mathbf{0.1250} \end{pmatrix}$
4.
   a. $\boldsymbol{x}^{(2)} = \begin{pmatrix} 1.48 \\ 0.75 \end{pmatrix} - \begin{pmatrix} -1.3387 & 0.0398 \\ 0 & 0.3333 \end{pmatrix}^{-1} \begin{pmatrix} 0.0668 \\ 0.1250 \end{pmatrix} = \begin{pmatrix} \mathbf{1.5661} \\ \mathbf{0.7083} \end{pmatrix}$
      i. $\boldsymbol{g}(\boldsymbol{x}^{(2)}) = \begin{pmatrix} 0.0034 \\ 0.0035 \end{pmatrix}$
   b. $\boldsymbol{x}^{(3)} = \begin{pmatrix} 1.5661 \\ 0.7083 \end{pmatrix} - \begin{pmatrix} -1.4118 & 0.0024 \\ 0 & 0.3529 \end{pmatrix}^{-1} \begin{pmatrix} 0.0034 \\ 0.0035 \end{pmatrix} = \begin{pmatrix} \mathbf{1.5708} \\ \mathbf{0.7071} \end{pmatrix}$
      i. $\boldsymbol{g}(\boldsymbol{x}^{(3)}) = \begin{pmatrix} 0.5765 \\ 0.3004 \end{pmatrix} \times 10^{-5}$

Code:

```
clc; clear; close all
syms x1 x2
g = [x2.*cos(x1); 2.*x2.^2 - 1];

fprintf('Solution to question 1:')
J = jacobian(g)

fprintf('Solution to question 2:')
g = @(x) [x(2).*cos(x(1)); 2.*x(2).^2 - 1];
x_0 = [1 1];
g_at_x_nought = g(x_0)

fprintf('Solution to question 3:')
Jinv = @(x) [-1/(x(2)*sin(x(1))) cos(x(1))/(4*x(2)^2*sin(x(1))); 0 1/(4*x(2))];
x_1 = (x_0' - Jinv(x_0)*g(x_0))'
g_at_x_1 = g(x_1)

fprintf('Solution to question 4:')
Jinv = @(x) [-1/(x(2)*sin(x(1))) cos(x(1))/(4*x(2)^2*sin(x(1))); 0 1/(4*x(2))];
x_2 = (x_1' - Jinv(x_1)*g(x_1))';
g_at_x_2 = g(x_2)
x_3 = (x_2' - Jinv(x_2)*g(x_2))';
g_at_x_3 = g(x_3)
```

Solution to question 1:
J =

$\begin{pmatrix} -x_2\sin(x_1) & \cos(x_1) \\ 0 & 4\,x_2 \end{pmatrix}$

```
Solution to question 2:
g_at_x_nought = 2×1
    0.5403
    1.0000

Solution to question 3:
x_1 = 1×2
    1.4816    0.7500

g_at_x_1 = 2×1
    0.0668
    0.1250

Solution to question 4:
g_at_x_2 = 2×1
    0.0034
    0.0035

g_at_x_3 = 2×1
10^-5 ×
    0.5765
    0.3004
```


```
clc; clear; close all
syms x1 x2
g = [x2.*cos(x1); 2.*x2.^2 - 1];
```

```
fprintf('Solution to question 1:')
J = jacobian(g)
```

```
fprintf('Solution to question 2:')
g = @(x) [x(2).*cos(x(1)); 2.*x(2).^2 - 1];
x_0 = [1 1];
g_at_x_nought = g(x_0)
```

```
fprintf('Solution to question 3:')
Jinv = @(x) [-1/(x(2)*sin(x(1))) cos(x(1))/(4*x(2)^2*sin(x(1))); 0 1/(4*x(2))];
x_1 = (x_0' - Jinv(x_0)*g(x_0))'
g_at_x_1 = g(x_1)
```

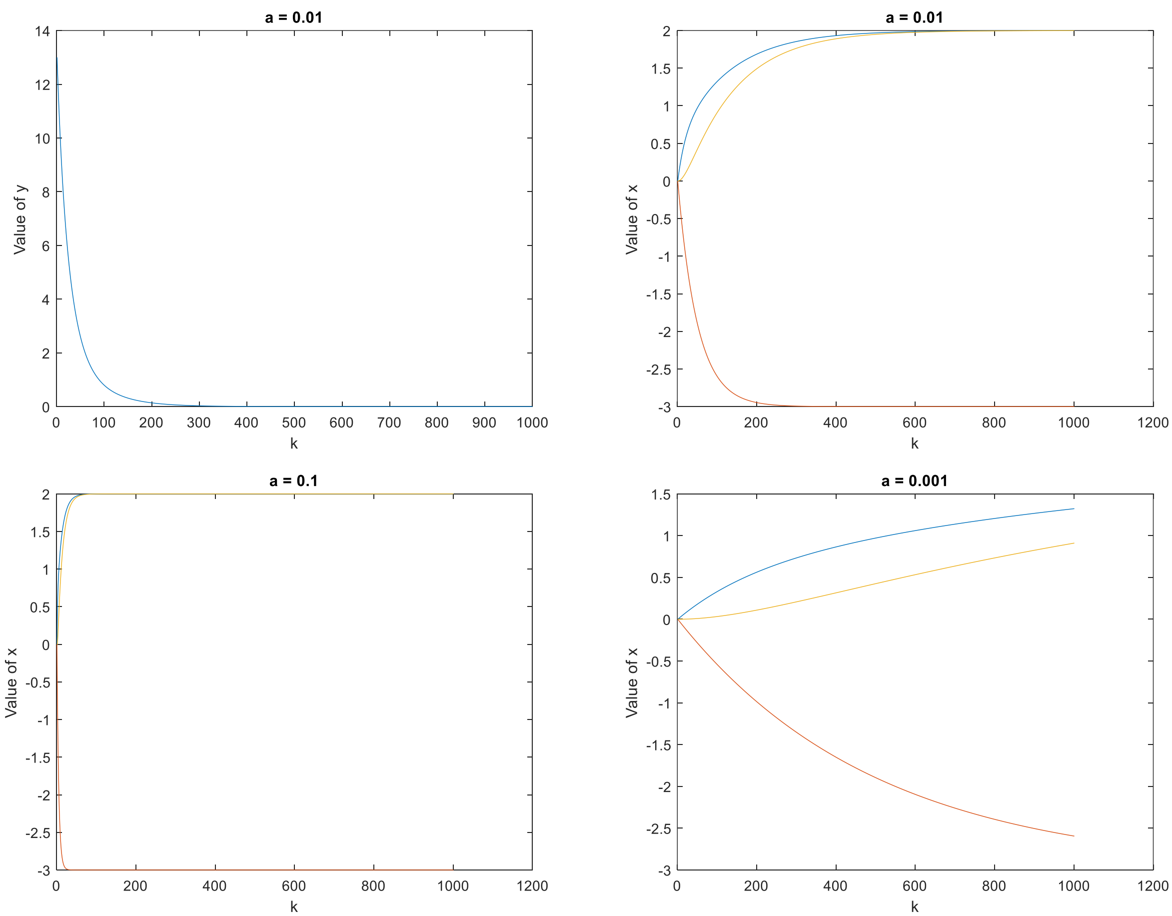
```
fprintf('Solution to question 4:')
Jinv = @(x) [-1/(x(2)*sin(x(1))) cos(x(1))/(4*x(2)^2*sin(x(1))); 0 1/(4*x(2))];
x_2 = (x_1' - Jinv(x_1)*g(x_1))';
g_at_x_2 = g(x_2)
x_3 = (x_2' - Jinv(x_2)*g(x_2))';
g_at_x_3 = g(x_3)
```

Part II

Code:

```
clear; close all; clc;
a = 0.01;
x = zeros(1000,3);
y = zeros(1000,1);
f = @(x) (x(1)-2)^2 + (x(2)+3)^2 + (x(3)-x(1))^2;
g = @(x) [2*(x(1)-2) - 2*(x(3)-x(1)); 2*(x(2)+3); 2*(x(3)-x(1))];

 for k = 1:1000
     x(k+1,1:3) = (x(k,1:3)' - a*g(x(k,1:3)))';
     y(k) = f(x(k,1:3));
 end

figure(1)
plot(1:1000,y)
title('a = 0.01')
xlabel('k')
ylabel("Value of y")
figure(2)
plot(x)
title('a = 0.01')
xlabel('k')
ylabel("Value of x")
%-----------------------------------------------------------------------------
clear;
a = 0.1;
x = zeros(1000,3);
y = zeros(1000,1);
f = @(x) (x(1)-2)^2 + (x(2)+3)^2 + (x(3)-x(1))^2;
g = @(x) [2*(x(1)-2) - 2*(x(3)-x(1)); 2*(x(2)+3); 2*(x(3)-x(1))];

 for k = 1:1000
     x(k+1,1:3) = (x(k,1:3)' - a*g(x(k,1:3)))';
     y(k) = f(x(k,1:3));
 end

figure(3)
plot(x)
title('a = 0.1')
xlabel('k')
```

```
ylabel("Value of x")
%------------------------------------------------------------------------
clear;
a = 0.001;
x = zeros(1000,3);
y = zeros(1000,1);
f = @(x) (x(1)-2)^2 + (x(2)+3)^2 + (x(3)-x(1))^2;
g = @(x) [2*(x(1)-2) - 2*(x(3)-x(1)); 2*(x(2)+3); 2*(x(3)-x(1))];

 for k = 1:1000
     x(k+1,1:3) = (x(k,1:3)' - a*g(x(k,1:3)))';
     y(k) = f(x(k,1:3));
 end

figure(4)
plot(x)
title('a = 0.001')
xlabel('k')
ylabel("Value of x")
```



Changing alpha modifies the size of the step in the gradient descent. Therefore, a larger alpha will result in a steeper step downhill and a more rapid approach of the variables to their optimal values.