1. Exercise PP. 1

Minimize

$$f(x_1, x_2) = x_1 - x_2 + 2x_1^2 + 2x_1x_2 + x_2^2 \tag{1}$$

from the starting point $X_1(x_1, x_2) = (0, 0)$ using the Steepest Descent Method, using the Armijo condition.

1.1. MATLAB Files

PP1_main.m

```
% Main  file
% Calls:PP1_data.m, PP1_search , PP1_plot, armijo

clear
clc
close(figure(1))    % Close open figure

% Initialize Data
PP1_data

% Open Contour Plot Figure
figure(1)
% Draw Conturs at increments of 0.3
fcontour(f)
xlabel('x_1')
ylabel('x_2')
title('f(x_1,x_2) = x_1 - x_2 + 2x_1^2 + 2x_1x_2 + x_2^2')
hold on

% Search Cycle
t=0;
while t<tmax && norm(grad_f(x(1),x(2))) > precision
    t=t+1;

    % Select the search direction
    PP1_search

    % Determine the step size
    alpha=armijo(f,d,x,grad_f);

    % New search point
    x_old=x;
    f_old=f_obj;
```

```
34
35      x=x+alpha*d;
36      f_obj=f(x(1),x(2));
37
38      % Plot the current search path
39      PP1_plot
40  end
41
42  % Display the results
43  fprintf('Number of Iterations: %d\n\n', t);
44  fprintf('Point of Minima: [%d , %d]\n\n', x(1),x(2));
45  fprintf('Objective Function Minimum Value after Optimization: %
      d\n\n',f_obj);
```

PP1_data.m

```
1  % Data Initialization
2
3  % Objective Function
4  f  = @(x1,x2) x1-x2 + 2.*x1.^2 + 2.*x1.*x2 + x2.^2;
5  grad_f = @(x1,x2) [1+2.*x2+4.*x1 -1+2.*x1+2.*x2];
6
7  % Maximum number of iterations
8  tmax = 2000;
9
10 % Initial Point
11 x = [0 0];
12 f_obj(1) = f(x(1),x(2));
13
14 % Precision
15 % (may not be achivied if the necessary number of iterations is
16 % greater than the maximum defined above)
17 precision = 1E-10;
```

PP1_search.m

```
1  % Computes the search direction as defined in the Steepest
2  % Descent Method.
3  d=-grad_f(x(1),x(2));
```

PP1_plot.m

```
% Plots the iterated points.
x_coord = [x_old(1),x(1)];
y_coord = [x_old(2),x(2)];

plot(x_coord,y_coord,'o-r')
```

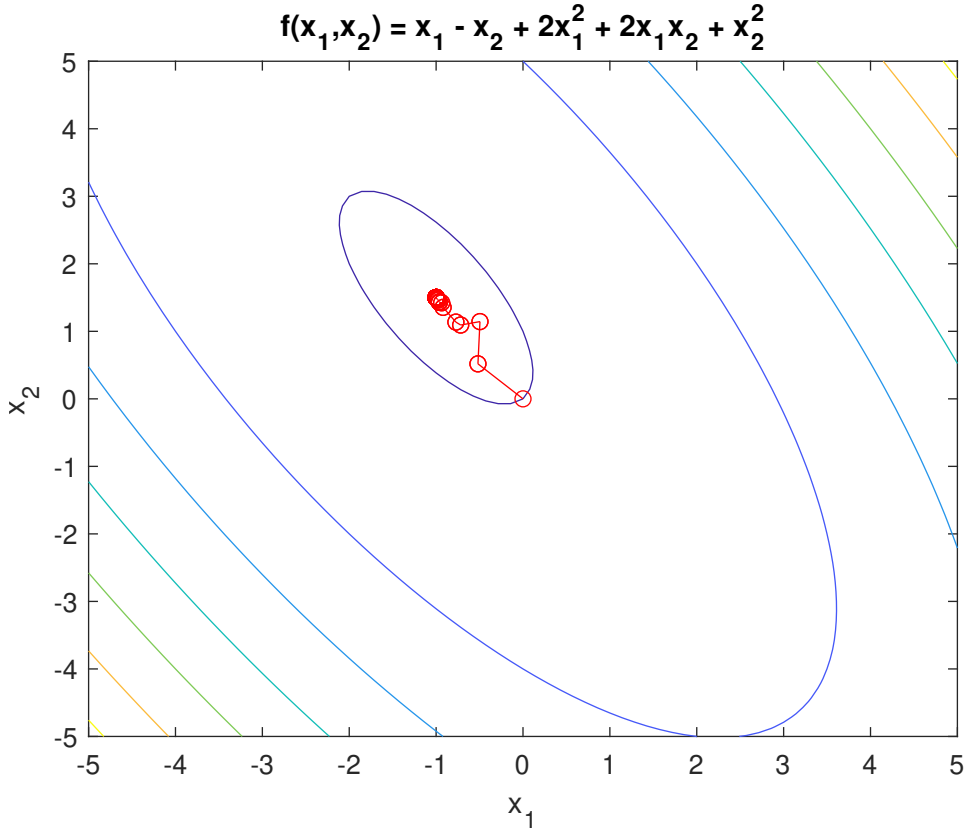
armijo.m

```
function [alpha]=armijo(f,d,x,grad_f)
% Armijo Linear Search
% f - objective function
% d - search direction
% x - search point
% grad_f - gradient of the obejctive function

% Defines delta between (0,1), gamma between (0,1/2) and
% c between (0,1)
delta=rand(1);
gamma=rand(1)*0.5;
c=rand(1);

% Initial guess for the step size alpha
a=c*abs(grad_f(x(1),x(2))*d.')/(norm(d))^2;

% Application of the Armijo Condition
while true
    if f(x(1)+a*d(1),x(2)+a*d(2))<=f(x(1),x(2))+gamma*a*grad_f(
       x(1),x(2))*d.'
        break
    else
        a=delta*a;
    end
end
alpha=a;
```

1.2. Results

Executing the script file *PP1_main.m*, the results obtained are the following:

```
Number of Iterations: 89
Point of Minima: [-1.000000e+00 , 1.500000e+00]
Objective Function Minimum Value after Optimization: -1.250000e+00
```



The number of iterations may vary because of the randomized values used for δ, γ and c in *armijo.m*.

2. Exercise PP. 2

Solve the following problem using the steepest descent (Cauchy) method.

$$f(x_1, x_2) = x_1 - x_2 - 2x_1x_2 - +x_1^2 - 2x_2^2 \tag{2}$$

2.1. MATLAB Files

The solution of this exercise is very similar to the Exercise PP. 1. In fact, the the only files we have to modify are the *PP1_data.m* and the *PP1_main.m*.

PP2_main.m

Using *PP1_main.m* as a template, line 9 will be changed to *PP2_data*, in order to initialize the data related to this exercise. Line 17 will also be changed so the title of the plot agrees with the input data. The rest is the same as in *PP1_main.m*

```
1  ...
2  % Initialize Data
3  PP2_data
4
5  % Open Contour Plot Figure
6  figure(1)
7  % Draw Conturs at increments of 0.3
8  fcontour(f)
9  xlabel('x_1')
10 ylabel('x_2')
11 title('f(x,y) = x_1^2 + x_2^2 + x_1x_2-1')
12 hold on
13 ...
```

PP2_data.m

Taking *PP1_data.m*, the modified parameters will be the objective function, its gradient, and the initial search point.

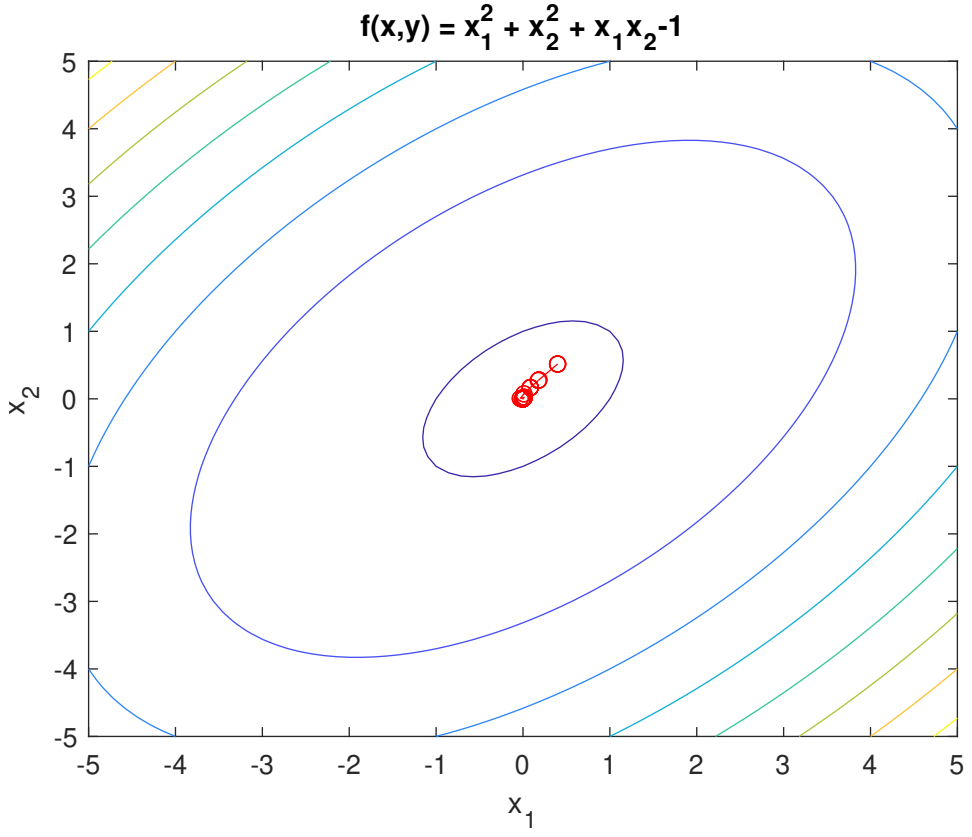
```
1  % Data Initialization
2
3  % Objective Function
4  f   = @(x1,x2) x1.^2+x2.^2-x1.*x2-1;
5  grad_f = @(x1,x2) [2.*x1+x2 x1+2.*x2];
6
7  % Maximum number of iterations
8  tmax = 2000;
9
10 % Initial Point
```

```
x = rand(1,2); % The inital point is not given so one will be
               % randomized
f_obj(1) = f(x(1),x(2));

% Precision
% (may not be achivied if the necessary number of iterations is
% greater than the maximum defined above)
precision = 1E-10;
```

2.2. Results

Executing the script file *PP2_main.m*, the results obtained are the following:

```
Number of Iterations: 59
Point of Minima: [-0.0000000000 , 0.0000000000]
Objective Function Minimum Value after Optimization: -1
```



In this exercise, the number of iterations, also, may vary because of the randomized values used for δ, γ and c in *armijo.m* and because of the randomized initial search point.