KAUNAS UNIVERSITY OF TECHNOLOGY

Faculty of Mathematics
and Natural Sciences

Optimization Methods

Laboratory work report
Lab 2

Names:Mohamed Abdelmonem,Alexandros Veremis
Group Number: 26
Submission date: 2020-05-08

(Erasmus- Spring 2020)

KAUNAS

REPORT FOR LABORATORY 2:

Firstly, our group number is 26 ⇒ (26%20+1)=7, so we have Task 1 function #3 and Task 2 function #2.
Task 1 function #3: $f(x,y)=(x+2y-7)^2+(2x+y-5)^2$
Subject to: $g1(x,y)=x+y\le 0$; $g2(x,y)=x^2+y^2-1.5\le 0$.
Task 2 function #2: $f(x,y)=(x+2)^2+(y+2)^2$
Subject to: $\hbar(x,y)=y-x=0$

Task 1

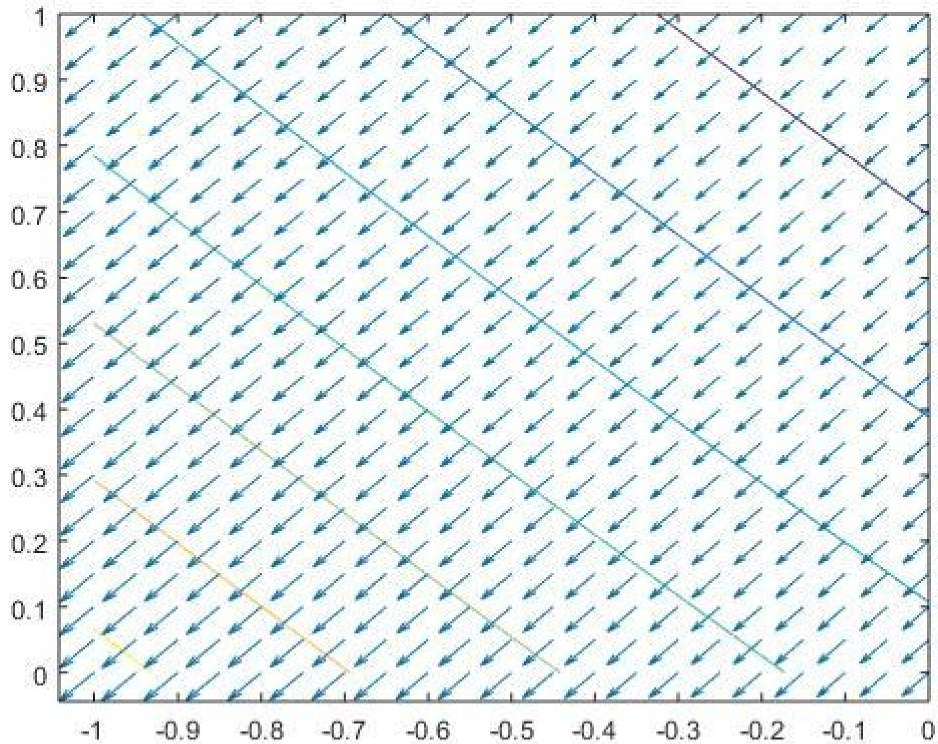
Use confun to minimize the function described in Instructions for the Preparation of the Report. Check which constraints are active by means of the output parameter options. Compute gradients of the target function and the constraints at the solution produced by confun. Check if Karush-Kuhn-Tucker Theorem conditions do hold true at this point. Plot the field of contour lines and gradient fields for the target functions and constraints. Visualize gradient vectors at this point. How the solution depends on initial conditions?

By using fmincon to minimize the function we get the minimum point to be: [x,y]=[-0.866 , 0.866]
Both constraints are zero at this point, which means that both of them are active.
The gradients that were asked:
gradf1x=6*x+6*y-24; ⇒ gradf1x(min)=-24
gradf1y=6*x+6*y-24; ⇒ gradf1y(min)=-24
gradg1x=1; ⇒ gradg1x(min)=1
gradg1y=1; ⇒ gradg1y(min)=1
gradg2x=2*x; ⇒ gradg2x(min)=-1.732
gradg2y=2*y; ⇒ gradg2y(min)=1.732

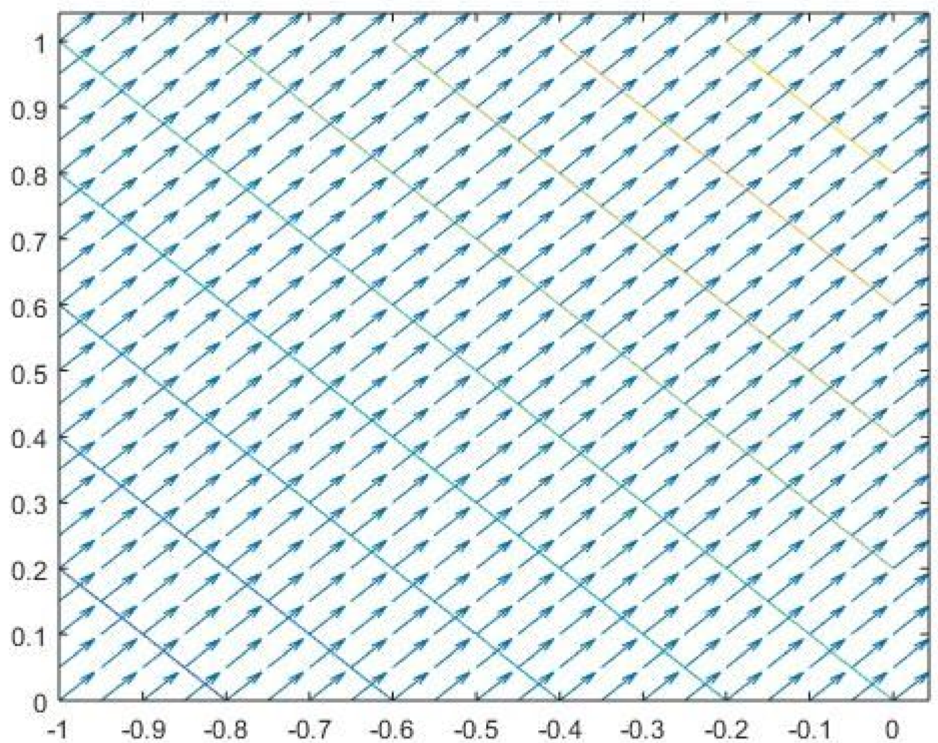
As, we checked all of the KKT Theorem conditions hold true at the minimum point.

The solution does not depend on the initial point, as long as the algorithm manages to find the right minimum. Because we only use the initial point at the start for fmincon. After that we do not use it again, so that is all the dependence! We only need it in order to find the minimum point , so a good choice for the initial point is needed in difficult functions, in others it does not play a significant role.

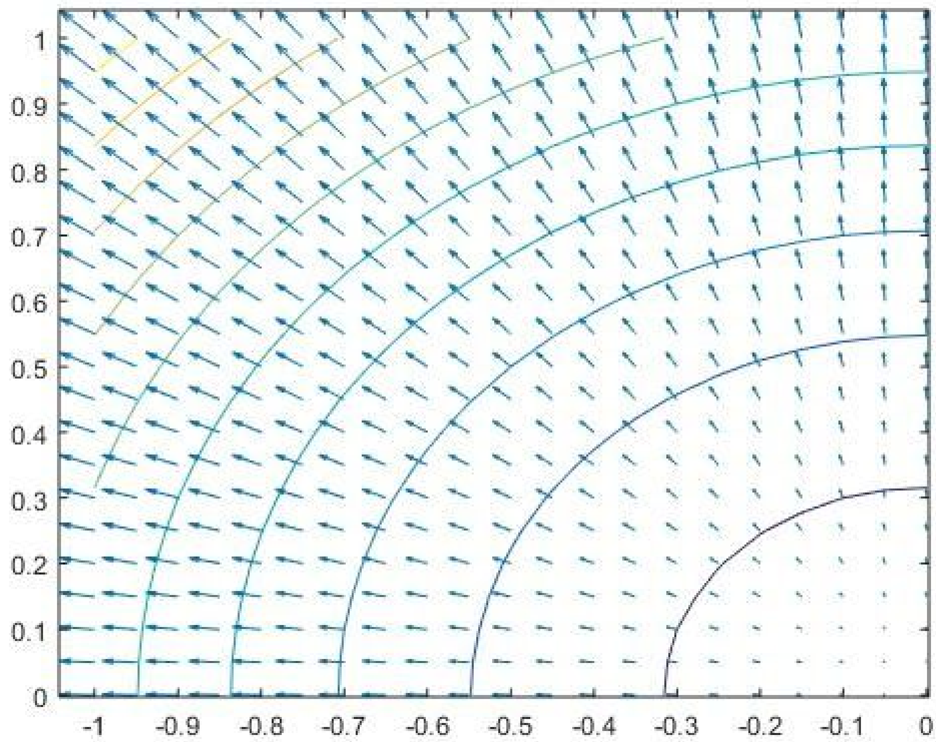
Output:



Field of countour lines and gradient field for the target function f.



Field of contour lines and gradient field for constraint g1.



Field of contour lines and gradient field for constraint g2.

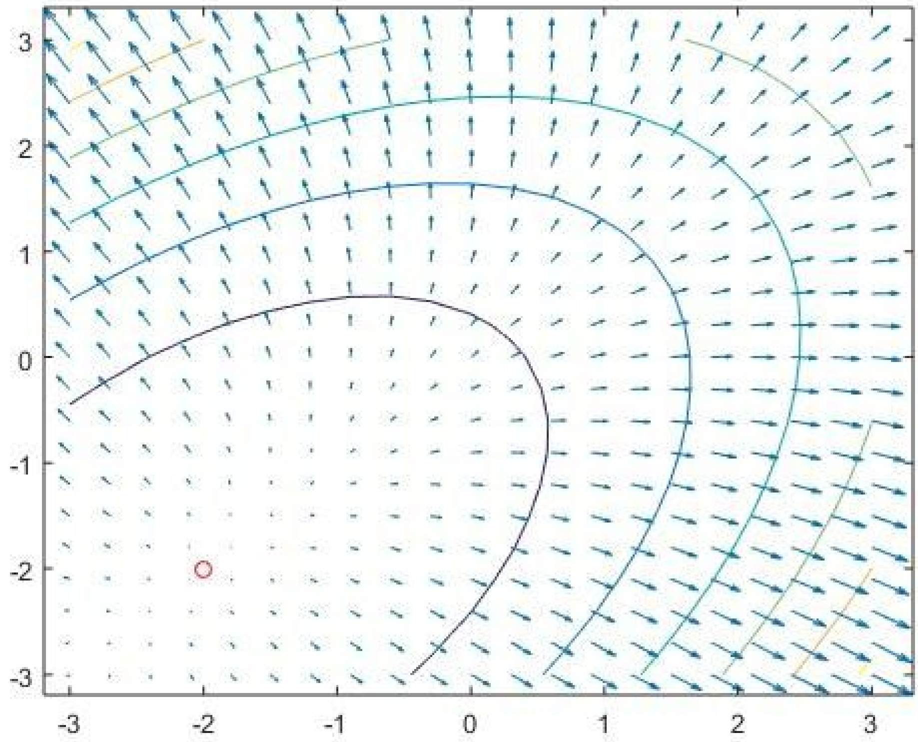
Task 2

Use the algorithm described in the example to minimize the function described in Instructions for the Preparation of the Report by the penalty function method.
Check if Karush-Kuhn-Tucker Theorem conditions do hold true at the solution. Comment the results.
Why it is impossible to select the value of parameter R almost equal to 0 at the first iteration? Explain the answer in details, provide results of computational experiments.

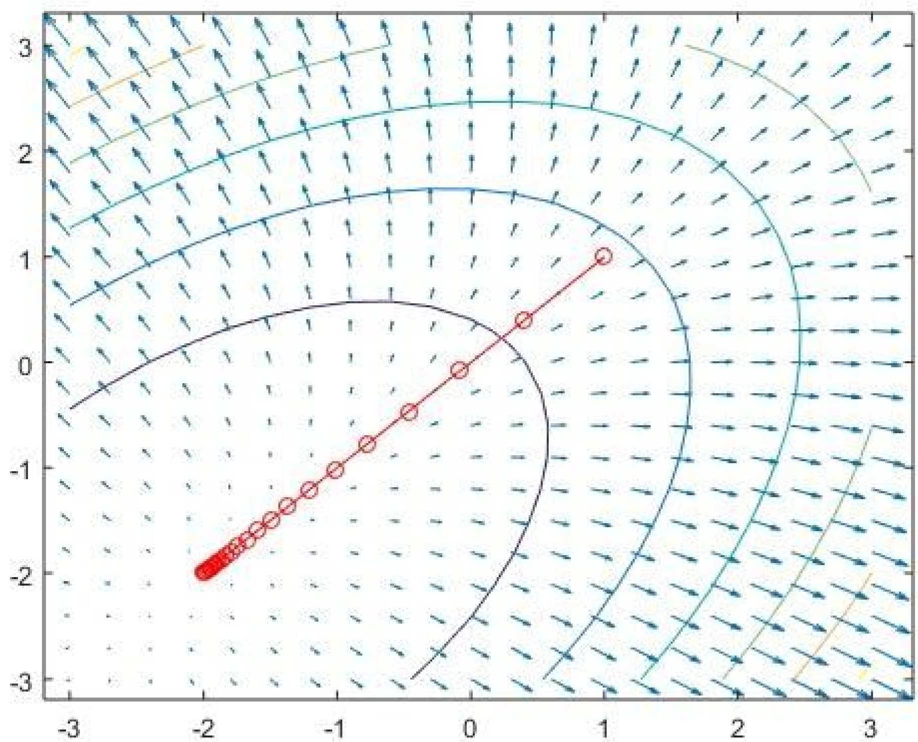
For the minimum point: if we use as [1,1] as our initial point then our minimum point turns to be [-1.9963,-1.9963].
On the other hand, if we start from [-2,-2] then our minimum point is [-2,-2].
As fas as the KKT Theorem conditions are concerned:
As we checked we ended up to the conclusion that only if we take [-2 -2] as our starting point, we will end with the right result about the minimum which is [-2 -2]. Otherwise, we will end up with a precision near [-2 -2] like [-1.9963,-1.9963] which will not satisfy the KKT conditions.
The KKT conditions under [-2 -2] : $\lambda 1$ ends up being 0
Therefore, $\lambda 1 * h(x)=0$ and lamda1=0>=0.
Because, R is the denominator and it has to decrease in every iteration in order to show the difference and eventually reach a value close to zero. R could be almost equal to 0 at the first iteration if R was on the nominator. It would be problematic if it was close to 0 in our case, since it would not be helpful in order to find the minimum point.

Below stand the computational experiments with different values of R. They show us that when R is almost equal to zero at the first iteration, then the fields do not surround the minimum point but are more like parallel straight lines ; rather than curved ones.

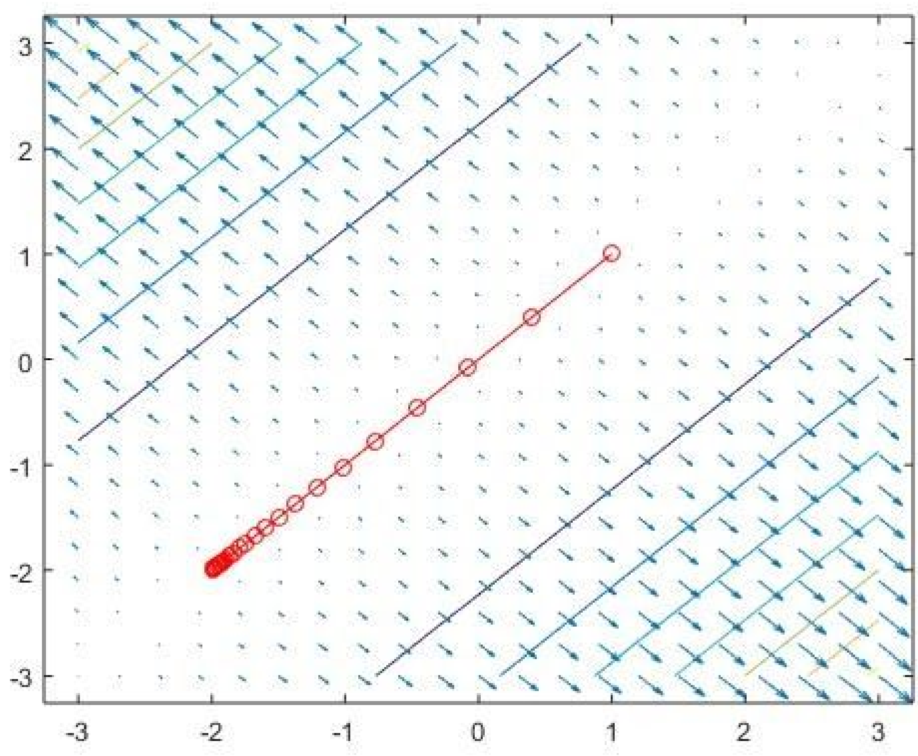
Output:



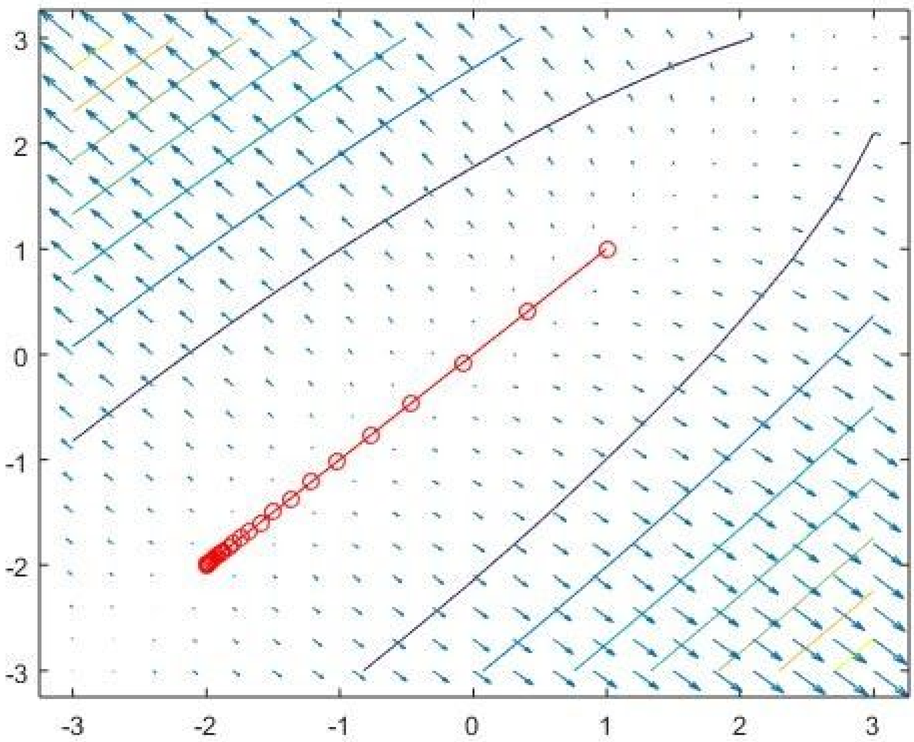
Field of contour lines and gradient field for the penalty function with the initial point being [-2 -2], and with the red spot being the minimum point calculated in every iteration



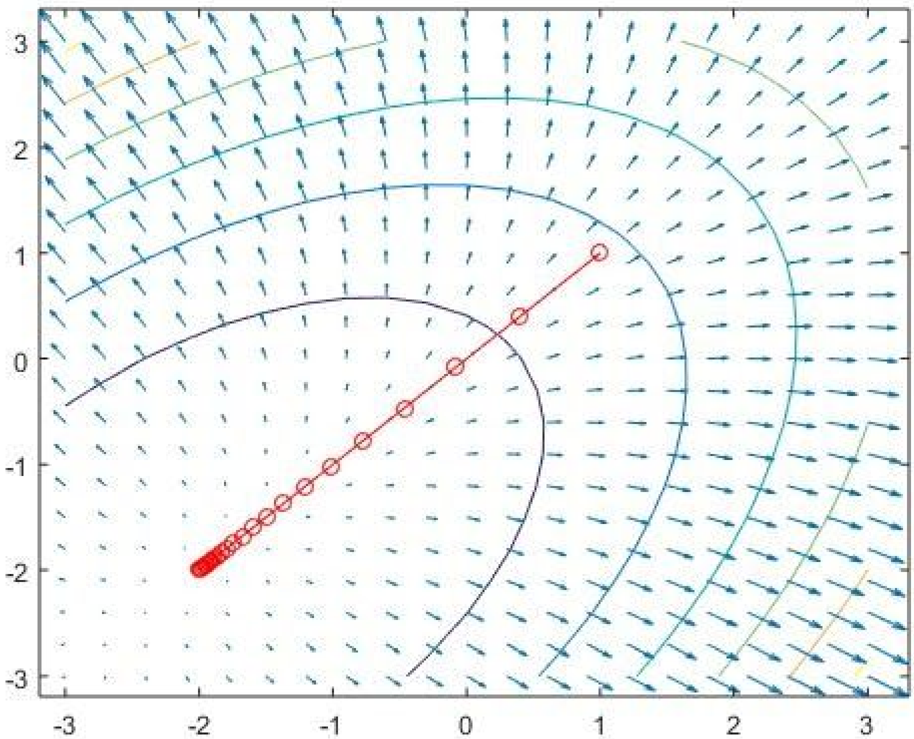
Field of contour lines and gradient field for the penalty function with the initial point being [-2 -2], and with the red spot being the minimum point calculated in every iteration



When R=0.00001



When R=0.1



When R=1

Task 3

Replace the equality constraint by the inequality constraint. Check that the minimum points of the target function would not belong to the feasible region. Construct the iterative minimization procedure. Visualize iterations and comment the results.

Since we have an inequality constraint instead of an equality constraint we must not penalize our function if the equality constraint is satisfied. In order to do so we add the same quadratic penalty function as in Task 3, but change it so when y - x is positive the penalty is zero. We do so using the following term:

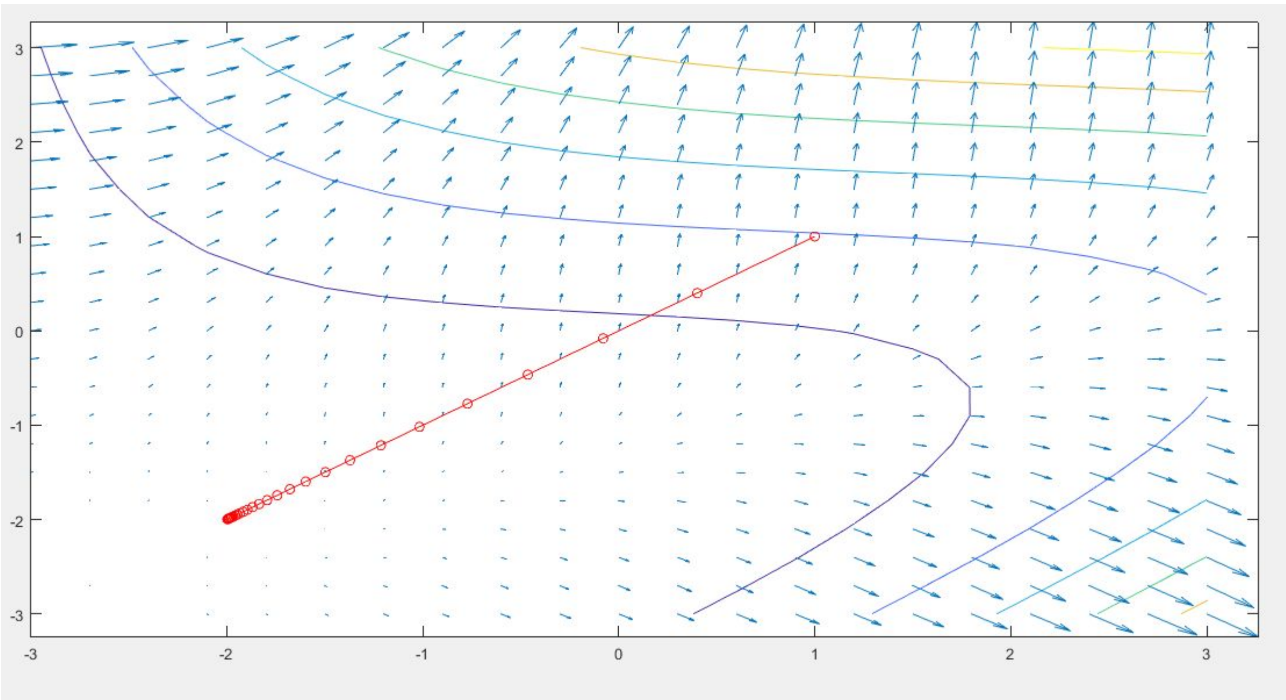
*(y-x).^2*abs((sign(y-x)-1)/2)*

If y - x is positive, meaning the point is in the feasible set then *abs((sign(y-x)-1)/2)* will be zero and the penalty will disappear.

if y - x is negative meaning the point is not in the feasible set then *abs((sign(y-x)-1)/2)* will be 1 and the penalty remains the same as in Task 2.

The minimum result is -1.9963   -1.9963 which is in the feasible set, as well as the result from every other iteration.

Output:



Program Code :

task1.m

close all;

clear all;

%TASK 1

% make an initial guess:

```
x0 = [0 0];
% Setup the optimization parameters:
% turn off large-scale algorithms
% turn on Display options for visualization of transient results
options = optimset('LargeScale','off','Display','iter');
% non-explicit constraints are replaced by []
[x,fval,exitflag,output]=fmincon('f1',x0,[],[],[],[],[],[],'constraints',options)

%Check which constraints are active by means of the output parameter options.
checkActive=constraints(x);

%Compute gradients of the target function and the constraints at the solution produced by confun.
MINX=x;
%both are zero ==> active
%gradf1x=6*x+6*y-24;
%gradf1y=6*x+6*y-24;
%gradg1x=1;
%gradg1y=1;
%gradg2x=2*x;
%gradg2y=2*y;
gradf1xmin=gradf1x(x);
gradf1ymin=gradf1x(x);
gradg1xmin=1;
gradg1ymin=1;
gradg2xmin=gradg2x(x);
gradg2ymin=gradg2y(x);


%KKT
A1=-24+lamda1*1+lamda2*(-1.732);
A2=-24+lamda1*1+lamda2*(1.732);
lamda1=24;
lamda2=0;
%so  both lamdas are >= 0
result1=lamda1*(MINX(1)+MINX(2)); %equals 0
%lamda2*g2min=0*g2min=0 ; equals 0
%then every condition is fullfilled

%the right plot
[x,y]=meshgrid(-1:.05:0,0:.05:1);
f=(x+2*y-7).^2+(2*x+y-5).^2;
[dx,dy]=gradient(f);
figure
contour(x,y,f), hold on
quiver(x,y,dx,dy), hold off
z1=x+y;
[dx,dy]=gradient(z1);
figure
contour(x,y,z1), hold on
quiver(x,y,dx,dy), hold off
z2=x.^2+y.^2-1.5;
```

```
[dx,dy]=gradient(z2);
figure
contour(x,y,z2), hold on
quiver(x,y,dx,dy)
```

gradf1x.m

```
function f=gradf1x(x)
f=6*x(1)+6*x(2)-24;
end
```

gradg2x.m

```
function f=gradg2x(x)
f=2*x(1);
end
```

gradg2y.m

```
function f=gradg2y(x)
f=2*x(2);
end
```

f1.m

```
function f=f1(x)
        f=(x(1)+2*x(2)-7).^2+(2*x(1)+x(2)-5).^2;
end
```

constraints.m

```
function [c,ceq]=constraints(x)
c=[x(1)+x(2);x(1).^2+x(2).^2-1.5];
ceq=[];
end
```

task2.m

```
close all;
clear all;

%TASK 2
%f(x,y)=(x+2).^2+(y+2).^2
%h(x,y)=y-x=0
%P(x,y,R)= (x+2).^2+(y+2).^2 +((y-x).^2)/R

R=1;
e=0.001;

[x,y]=meshgrid(-3:.3:3);
P=(x+2).^2+(y+2).^2 +((y-x).^2)/R;
figure(1)
hold off
contour(x,y,P)
[dx,dy]=gradient(P,.2,.2);
hold on
quiver(x,y,dx,dy)
%starting point
x=[1,1];
xs=x; % dummy variable required for the iterative process
```

```
step=0.1; % the step size
previousXS=xs+10;
while(abs(xs-previousXS)>e)
previousXS=xs;
% compute the next point
x = xs - step*gradientfortask2(xs,R);
% plot the step
plot([xs(1),x(1)],[xs(2),x(2)],'r',[xs(1),x(1)],[xs(2),x(2)],'ro')
% refresh the variables
xs=x;
R=R/5;
end

%KKT
%2*(x(1)+2)-lamda1*(-1)=0;
%2*(x(2)+2)-lamda1*(1)=0;
%So only if we take as starting point [-2 -2] we will end with the right result about the minimum
```

gradientfortask2.m

```
function g=gradientfortask2(x,R)
% partial derivative in respect of x(1)
g(1)=2*x(1)+4+(-2/R)*(x(2)-x(1));
% partial derivative in respect of x(2)
g(2)=2*x(2)+4+(2/R)*(x(2)-x(1));
end
```

task3.m

```
close all;
clear all;

R=1;
e=0.001;

[x,y]=meshgrid(-3:.3:3);
P=(x+2).^2+(y+2).^2 +(y-x).^2*(abs((sign(y-x)-1))/2)/R;
figure(1)
hold off
contour(x,y,P)
[dx,dy]=gradient(P,.2,.2);
hold on
quiver(x,y,dx,dy)
%starting point
x=[1,1];
xs=x; % dummy variable required for the iterative process
step=0.1; % the step size
previousXS=xs+10;
while(abs(xs-previousXS)>e)
previousXS=xs;
% compute the next point
x = xs - step*gradientfortask3(xs,R);
% plot the step
```

```
plot([xs(1),x(1)],[xs(2),x(2)],'r',[xs(1),x(1)],[xs(2),x(2)],'ro')
% refresh the variables
if x(2)-x(1) < 0
   disp('answer not in feasible set')
end
xs=x;
R=R/5;
end
disp(x)
```

gradientfortask3.m

```
function g=gradientfortask3(x,R)
% partial derivative in respect of x(1)
g(1)=2*x(1)+4+(-2/R)*(x(2)-x(1))*(abs(sign(x(2)-x(1)))-1)/2;
% partial derivative in respect of x(2)
g(2)=2*x(2)+4+(2/R)*(x(2)-x(1))*(abs(sign(x(2)-x(1)))-1)/2;
end
```

<u>Control Work-Alexandros Veremis :</u>

Alexandros Veremis

Control Work for Laboratory 2.

Problem 1. $f(x,y) = x^2 + y^2$

$g_1(x,y) = -\frac{3}{2}x - y + 3 \le 0$

$g_2(x,y) = -x + 1 \le 0$

Check is $(x^*, y^*) = (1, 3/2)$ is a min. point (using KKT Theorem cond.).

∴ KKT conditions:

a) $$\left.\begin{aligned} \frac{\partial f}{\partial x} + \lambda_1 \frac{\partial g_1}{\partial x} + \lambda_2 \frac{\partial g_2}{\partial x} = 0 \\ \frac{\partial f}{\partial y} + \lambda_1 \frac{\partial g_1}{\partial y} + \lambda_2 \frac{\partial g_2}{\partial y} = 0 \end{aligned}\right\}.$$

$$\left.\begin{aligned} 2x + \lambda_1\left(-\frac{3}{2}\right) + \lambda_2(-1) = 0 \\ 2y + \lambda_1(-1) + 0 = 0 \end{aligned}\right\} \xrightarrow{(x^*, y^*)}$$

$$\left.\begin{aligned} 2 \cdot 1 + \lambda_1\left(-\frac{3}{2}\right) + \lambda_2(-1) = 0 \end{aligned}\right\} \Rightarrow$$

$\lambda_1 = 3$

$$\left.\begin{aligned} \lambda_2 = -9/2 + 2 = -5/2 \\ \lambda_1 = 3 \end{aligned}\right\}$$

So $\lambda_2 = -5/2 < 0 \Rightarrow$ 3rd condition of KKT Theorem is not satisfied $\Rightarrow$ $\left(1, \frac{3}{2}\right)$ is not a minimum point.

because KKT Theorem doesn't hold true at this point.

Problem 3). $f(x,y) = x^2 + 2y^2$

$g(x,y) = 2x - y + 1 \le 0$

Find the minimum point (using KKT).

$$\left.\begin{aligned} \frac{\partial f}{\partial x} + \lambda_1 \frac{\partial g}{\partial x} = 0 \\ \frac{\partial f}{\partial y} + \lambda_1 \frac{\partial g}{\partial y} = 0 \end{aligned}\right\}.$$

$$\left.\begin{aligned} 2x + \lambda_1 \cdot 2 = 0 \\ 4y + \lambda_1(-1) = 0 \end{aligned}\right\} \quad \begin{aligned} 2x + 2\lambda_1 = 0 \\ \lambda_1 = 4y. \end{aligned}$$

$x + \lambda_1 = 0 \Rightarrow x + 4y = 0 \Rightarrow$

$x = -4y.$

$$g(x,y) \xlongequal{x=-4y} -8y - y + 1 \le 0$$
$$= -9y + 1 \le 0.$$

So, $-9y + 1 \le 0$

$1 \le 9y \Rightarrow \frac{1}{9} \le y.$

Because, we are looking for the minimum point we will take the equality of this inequation $\Rightarrow y_{min} = \frac{1}{9}$

$\Rightarrow x_{min} = -\frac{4}{9}.$

So, $\lambda_1 = \frac{4}{9}.$

Condition b). $\lambda_1 \cdot g(x_{min}, y_{min}) =$

$$\frac{4}{9} \cdot \left(-\frac{8}{9} - \frac{1}{9} + 1\right) = \frac{4}{9} \cdot 0 = 0.$$

Condition c) $\lambda_1 = 4/9 \geq 0$

So, all conditions of the KKT Theorem hold true, which means that the minimum point really is $(x^*, y^*) = \left(-\frac{4}{9}, \frac{1}{9}\right)$.

Problem 2|. We now have $g_k \geq 0$, which is the same as $-g_k \leq 0$. Therefore, we can define $\rho_k = -g_k$. Then again, we have the constraints to be: $\rho_k \leq 0$ and if we now apply the KKT theorem, we will get:

a) $\nabla f(X_*) + \sum_{k=1}^{m} \lambda_k \nabla \rho_k(X_*) = 0$

b) $\lambda_k \cdot \rho_k(X_*) = 0$

c) $\lambda_k \geq 0$

Now, if we replace ρ_k with $-g_k$ again, we will get:

a) $\nabla f(X_*) - \sum_{k=1}^{m} \lambda_k \nabla g_k(X_*) = 0 \quad (E1)$

b) $-\lambda_k \cdot g_k(X_*) = 0 = \lambda_k g_k(X_*)$

and c) $\lambda_k \geq 0$.

because of how we defined (E1). So, as we see the only thing that changes when we have $g_k \geq 0$ is that: at the first condition (E1) instead of having a "plus symbol" we have a "minus symbol". Everything else remains ~~changed~~ the same.