Pedro Ferreira
201806093@up.pt

1 Exercise PP. 9

Consider the following optimization problem:

$$\begin{aligned}
\text{minimize: } & x_1^4 - 2x_1^2x_2 + x_1^2 + x_1x_2^2 - 2x_1 + 4 \\
\text{subject to: } & x_1^2 + x_2^2 - 2 = 0 \\
& 0.25x_1^2 + 0.75x_2^2 - 1 \leq 0 \\
& 0 \leq x_1 \leq 4 \\
& 0 \leq x_1 \leq 4
\end{aligned}$$

Solve the constrained minimization using the Steepest Decent Method.

1.1 MATLAB Files

PP9_main.m

```
clear
clc
close(figure(1))

PP9_data

% Open Contour Plot Figure
figure(1)
fcontour(fx1x2,[0 5 0 5])
xlabel('$x_1$','Interpreter','latex')
ylabel('$x_2$','Interpreter','latex')
title('$f(x_1,x_2) = x_1^4 - 2x_1^2 x_2 + x_1^2 +x_1 x_2^2 -2 x_1 +4$', ...
    'Interpreter','latex')
dim =[0.15, 0.8, 0.1, 0.1];
const_str = {'Constraints:', '$x_1^2+x_2^2-2=0$',' ', ...
             '$0.25x_1^2+0.75x_2^2-1 \leq 0$',' ', '$0 \leq x_1 \leq 4$',...
             ' ', '$0 \leq x_2 \leq 4$'};
annotation('textbox',dim,'String', const_str,'Interpreter','latex','BackgroundColor','w')
grid on
hold on

epsilon_min = 0.0001;
epsilon = 0.1;
miu = zeros(length(hx1x2(X(1),X(2))),1);
lambda = zeros(length(gx1x2(X(1),X(2))),1);

```

```
% Search Cycle
k=0;
while k<kmax
    k=k+1;
    [Lx1x2,gradL] = auglag(X,f,g,gx1x2,h,epsilon,miu,lambda);
    [X,X_old] = minimize(X,Lx1x2,gradL,lb,ub,tmax);


    % Plot Search Path
    plot([X_old(1) X(1)],[X_old(2) X(2)],'o-r')

    % KKT Conditions
    [KKT, KKT_norm,flag]=KKT_fun(X,lambda,miu,grad_f,grad_h,grad_g,gx1x2);

    if flag
        break
    end

    [miu,lambda,epsilon] = update(X,hx1x2,gx1x2,miu,lambda,epsilon,epsilon_min);
end
f_obj = fx1x2(X(1),X(2));

% Plot point of minima in a different color
plot(X(1),X(2),'ok',MarkerFaceColor='k')

% Print the results
fprintf([ ...
         'Number of Iterations: %d\n\n', ...
         'Point of Minima: [%.4f , %.4f]\n\n', ...
         'Objective Function Minimum Value after Optimization: %.4f\n\n'], ...
         k,X(1),X(2), f_obj)
```

PP5_data.m

```
% Objective function
syms x1 x2
f(x1,x2) = x1^4 - 2*x1^2*x2 + x1^2 + x1*x2^2 - 2*x1 + 4;
df = gradient(f,[x1,x2]);
fx1x2 = matlabFunction(f);
grad_f = matlabFunction(df);

```

```
% Initial point (k=0)
X = [3;2];
f_obj = fx1x2(X(1),X(2));

% Maximum number of iterations
kmax = 2000;
tmax = 2000;

% Constraints
h = x1^2 + x2^2-2;
g = 0.25*x1^2 + 0.75*x2^2-1;
hx1x2 = matlabFunction(h);
grad_h = matlabFunction(gradient(hx1x2, [x1 x2]));
gx1x2 = matlabFunction(g);
grad_g = matlabFunction(gradient(gx1x2, [x1 x2]));
lb=[0; 0];
ub=[4; 4];
```

auglag.m

```
function [Lx1x2,gradL] = auglag(X,f,g,gx1x2,h,epsilon,miu,lambda)

syms x1 x2
g_aux = gx1x2(X(1),X(2));
aux = zeros(length(gx1x2(X(1),X(2))),1);
aux = sym(aux);

for i=1:length(gx1x2(X(1),X(2)))
    if (-epsilon/2)*lambda(i)>g_aux(i)
        aux(i)=(-epsilon/2)*lambda(i);
    else
        aux(i)=g(i);
    end
end

L(x1,x2) = f + miu'*h + lambda'*sym(aux) + (1/epsilon)*norm(h)^2 + (1/epsilon)*(norm(aux))^2;
dL = gradient(L,[x1,x2]);

Lx1x2 = matlabFunction(L);
gradL = matlabFunction(dL);
```

minimize.m

```
function [X,X_old] = minimize(X,Lx1x2,gradL,lb,ub,tmax)
% Minimizes the function at hand using the Steepest Descent
   Method and the Armijo Linear Search

t=0;
X_old = X;
while t<tmax && norm(gradL(X(1),X(2)))>10^-6
    t=t+1;

    % Search Direction
    d = -gradL(X(1),X(2));

    c = 0.01;
    gama = 0.01;
    delta = 0.5;

    % Define initial step value
    alfa = c*abs(gradL(X(1),X(2))'*d)/(norm(d)^2);
    X_new = X+alfa*d;


    while any(lb > X_new) || any(ub < X_new)
        alfa = alfa*delta;
        X_new = X+alfa*d;
    end

    % Find optimal step
    while Lx1x2(X_new(1),X_new(2)) > Lx1x2(X(1),X(2)) + gama*
       alfa*gradL(X(1),X(2))'*d && all(lb < X_new) && all(X_new
        < ub)
        alfa = alfa*delta;
        X_new = X+alfa*d;
    end

    X = X+alfa*d;
end
```

KKT_fun.m

```
function [KKT, KKT_norm,flag]=KKT_fun(X,lambda,miu,grad_f,
   grad_h,grad_g,gx1x2)
% Karush-Kuhn-Tucker Conditions
```

```

flag = false;
KKT = grad_f(X(1),X(2)) + miu.'*grad_h(X(1),X(2)) + lambda.'*
   grad_g(X(1),X(2));
KKT_norm = norm(KKT);

if KKT_norm<1e-06 && lambda'*gx1x2(X(1),X(2))<1e-06
    flag = true;
```

update.m

```
function [miu,lambda,epsilon] = update(X,hx1x2,gx1x2,miu,lambda
   ,epsilon,epsilon_min)

miu = miu+2/epsilon*hx1x2(X(1),X(2));
lambda = max([zeros(length(gx1x2(X(1),X(2))),1), lambda + (2/
   epsilon)*gx1x2(X(1),X(2))],[],2);
ro = 0.95;
epsilon_old = epsilon;
epsilon = epsilon*ro;

if epsilon < epsilon_min
    epsilon = epsilon_old;
end
```

1.2 Results

```
Number of Iterations: 16
Point of Minima: [1.0000 , 1.0000]
Objective Function Minimum Value after Optimization: 3.0000
```