MAT3007: Optimization - Assignment 9

Ran Hu 116010078

12 December, 2018

Problem 1

```
% Objective: minimize 2*x1^4 + 3*x2^4 + 2*x1^2 + 4*x2^2 + x1*x2 - 3*x1 -
2*x2

function y = f_p1(x)
y = 2*x(1)^4 + 3*x(2)^4 + 2*x(1)^2 + 4*x(2)^2 + x(1)*x(2) - 3*x(1) -
2*x(2);
```

```
function y = gradient_p1(x)

y1 = 8*x(1)^3 + 4*x(1) + x(2) - 3;
y2 = 12*x(2)^3 + 8*x(2) + x(1) - 2;
y = [y1; y2];
```

```
%% Using Gradient Method

%% Setting initial points
x = [0; 0];

%% Setting tolerance factor epsilon
epsilon = 10^(-6);

%% Initialize iteration number
iter = 0;

%% Setting backtracking search parameter
alpha = 0.5;
beta = 0.5;

%% Main Iteration
while norm(gradient_p1(x)) > epsilon

    %% Doing backtracking search
    d = gradient_p1(x);
    t = 1;
    xtemp = x - t * d;

    while f_p1(xtemp) >= f_p1(x) - alpha * t * gradient_p1(x)' * d
```

```
        t = t * beta;
        xtemp = x - t * d;
    end

    %% Output the solution in each step
    iter = iter + 1
    x = xtemp

end
```

Using gradient method, it takes 15 iterations. The optimal solution is: $x_1 = 0.4815$, $x_2 = 0.1809$.

```
function y = hessian_p1(x)
y = zeros(2,2);
y(1,1) = 24*x(1)^2 + 4;
y(1,2) = 1;
y(2,1) = y(1,2);
y(2,2) = 36*x(2)^2 + 8;
```

```
%% Using Newton's Method
clc;

%% Setting initial points
x = [0; 0];

%% Setting tolerance factor epsilon
epsilon = 10^(-6);

%% Initialize iteration number
iter = 0;
alpha = 0.5;
beta = 0.5;

%% Main Iteration
while norm(gradient_p1(x)) > epsilon

    dk = inv(hessian_p1(x)) * gradient_p1(x);
    t = 1;
    xtemp = x - t * dk;

    while f_p1(xtemp) >= f_p1(x) - alpha * t * gradient_p1(x)' * dk
        t = t * beta;
        xtemp = x - t * dk;
    end

    %% Output the solution in each step
    iter = iter + 1
```

```
    x = xtemp
end
```

Using Newton's method, it takes 20 iterations. The optimal solution is: $x_1 = 0.4815$, $x_2 = 0.1809$.

The optimal value is -1.0139 .

Problem 2

```
% Objective: minimize e^(x1+x2+x3) + x1^2 + 2*x2^2 + 3*x3^2 - 2*x1 - 7*x2 -
5*x3

function y = f_p2(x)
y = exp(x(1)+x(2)+x(3)) + x(1)^2 + 2*x(2)^2 + 3*x(3)^2 - 2*x(1) - 7*x(2) -
5*x(3);
```

```
function y = gradient_p2(x)

y1 = exp(x(1)+x(2)+x(3)) + 2*x(1) - 2;
y2 = exp(x(1)+x(2)+x(3)) + 4*x(2) - 7;
y3 = exp(x(1)+x(2)+x(3)) + 6*x(3) - 5;
y = [y1; y2; y3];
```

```
%% Using Gradient Projection Method

%% Setting initial points
x = [4; 0; 0];

%% Setting tolerance factor epsilon
epsilon = 10^(-5);

%% Setting PA
A = [1 2 3]
I = eye(3,3)
PA = I - A'*inv(A*A')*A

%% Initialize iteration number
iter = 0;

%% Setting backtracking search parameter
alpha = 0.5;
beta = 0.5;

%% Main Iteration
while norm(PA*gradient_p2(x)) > epsilon
```

```
    %% Doing backtracking search
    d = PA*gradient_p2(x);
    t = 1;
    xtemp = x - t * d;

    while f_p2(xtemp) >= f_p2(x) - alpha * t * gradient_p2(x)' * d
        t = t * beta;
        xtemp = x - t * d;
    end

    %% Output the solution in each step
    iter = iter + 1
    x = xtemp

end
```

Using gradient method, it takes 15 iterations. The optimal solution is: $x_1 = -0.5183$, $x_2 = 1.2500$, $x_3 = 0.6728$.

```
function y = hessian_p2(x)
y = zeros(3,3);
y(1,1) = exp(x(1)+x(2)+x(3)) + 2;
y(1,2) = exp(x(1)+x(2)+x(3));
y(1,3) = exp(x(1)+x(2)+x(3));
y(2,1) = y(1,2);
y(2,2) = exp(x(1)+x(2)+x(3)) + 4;
y(2,3) = exp(x(1)+x(2)+x(3));
y(3,1) = y(1,3);
y(3,2) = y(2,3);
y(3,3) = exp(x(1)+x(2)+x(3)) + 6;
```

```
%% Using Newton's Method
clc;

A = [1 2 3]
zero = [0]

%% Setting initial points
x = [4; 0; 0];

%% Setting tolerance factor epsilon
epsilon = 10^(-5);

%% Initialize iteration number
iter = 0;
alpha = 0.5;
beta = 0.5;
```

```
%% Main Iteration
dk = [1,1,1]

while norm(dk) > epsilon

    dk = inv(cat(1,cat(2,hessian_p2(x),A'),cat(2,A,zero))) *
cat(1,gradient_p2(x),zero);
    dk(4,:) = [];
    t = 1;
    xtemp = x - t * dk;

    while f_p2(xtemp) >= f_p2(x) - alpha * t * gradient_p2(x)' * dk
        t = t * beta;
        xtemp = x - t * dk;
    end

    %% Output the solution in each step
    iter = iter + 1
    x = xtemp

end
```

Using Newton's method, it takes 6 iterations. The optimal solution is: $x_1 = -0.5183$, $x_2 = 1.2500$, $x_3 = 0.6728$.

```
%% Using cvx

A = [1 2 3]

cvx_begin quiet
    variables x(3)
    minimize exp(x(1)+x(2)+x(3)) + x(1)^2 + 2*x(2)^2 + 3*x(3)^2 - 2*x(1) -
7*x(2) - 5*x(3);
    subject to
        A*x == 4
cvx_end

x
cvx_opt = cvx_optval
```

Using CVX, the optimal solution is: $x_1 = -0.5183$, $x_2 = 1.2500$, $x_3 = 0.6728$.

All three methods give the same solution $x_1 = -0.5183$, $x_2 = 1.2500$, $x_3 = 0.6728$. The optimal value is -2.2524 .