MAT3007: Optimization - Assignment 8

Ran Hu 116010078

28 November, 2018

Problem 1

```
xl = 0;
xr = 5;
while xr - xl > 10^(-5)
    xm = 0.5 * (xl + xr)
    if (xm^1.7 - 1.7^xm) > 0
        xr = xm;
    else
        xl = xm;
    end
end
solution = xm
```

The root of $x^{1.7} - 1.7^x = 0$ is 1.7 .

Problem 2

```
% To find the optimal solution to exp(-p) * p / (1 + exp(-p))
% The gradient = exp(-p) * (1 + exp(-p) - p) / (1 + exp(-p))^2

pl = 0;
pr = 10;
phi = (3 - sqrt(5)) / 2;
while pr - pl > 10^(-5)
    p1 = phi * pr + (1 - phi) * pl;
    p2 = phi * pl + (1 - phi) * pr;
    if  exp(-p1) * p1 / (1 + exp(-p1)) <  exp(-p2) * p2 / (1 + exp(-p2))
        pl = p1;
    else
        pr = p2;
    end
end
solution = (pl + pr) / 2
```

The optimal solution is 1.2785 .

Problem 3

```
function y = f(x)

y = exp(1-x(1)-x(2)) + exp(x(1)+x(2)-1) + x(1)^2 + x(1)*x(2) + x(2)^2 +
2*x(1) - 3*x(2);
```

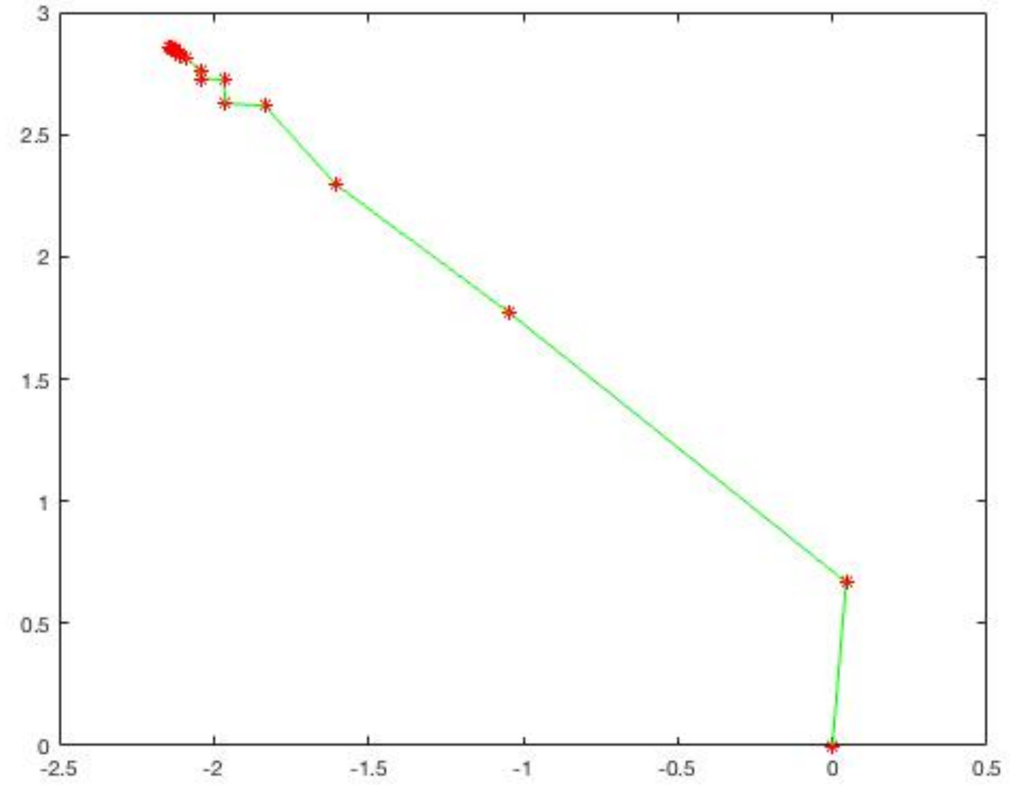
```
function z = gradient(x)

z1 = -exp(1-x(1)-x(2)) + exp(x(1)+x(2)-1) + 2*x(1) + x(2) + 2;
z2 = -exp(1-x(1)-x(2)) + exp(x(1)+x(2)-1) + x(1) + 2*x(2) - 3;
z = [z1; z2];
```

```
% Doing backtracking search
x = [0; 0];
epsilon = 10^(-5);
iter = 0;
alpha = 0.5;
beta = 0.5;

while norm(gradient(x)) > epsilon

    d = gradient(x);
    t = 1;

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

    plot(x(1), x(2), '*r');
    hold on;
    plot([x(1), xtemp(1)], [x(2), xtemp(2)], '-g');
    hold on;

    iter = iter + 1
    x = xtemp

end
```

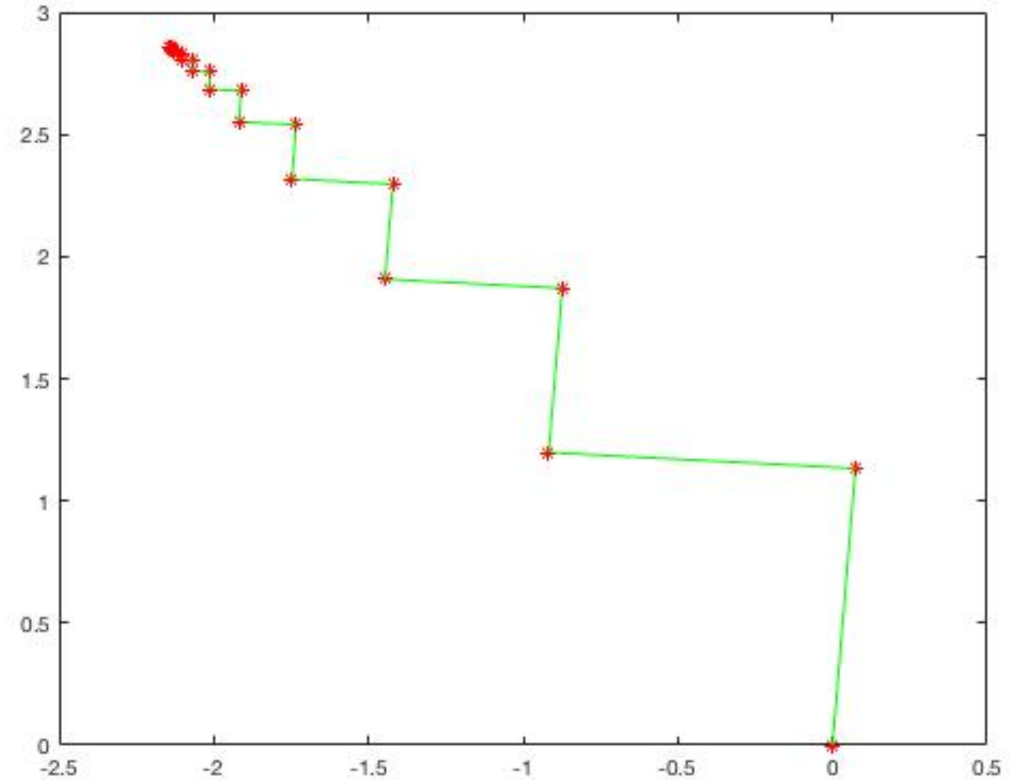
```
% Doing exact line search
x = [0; 0];
epsilon = 10^(-5);
iter = 0;
alpha = 0.5;
beta = 0.5;
```

```
while norm(gradient(x)) > epsilon

    xl = x;
    xr = x - 10 * gradient(x);
    phi = (3 - sqrt(5)) / 2;
    while norm(xr - xl) > 10^(-6)
        x1 = phi * xr + (1 - phi) * xl;
        x2 = phi * xl + (1 - phi) * xr;
        if  f(x1) >  f(x2)
            xl = x1;
        else
            xr = x2;
        end
    end

    xtemp = (xr + xl) / 2;

    plot(x(1), x(2), '*r');
    hold on;
    plot([x(1), xtemp(1)], [x(2), xtemp(2)], '-g');
    hold on;

    iter = iter + 1
    x = xtemp

end
```

The solution path for the backtracking line search method:



The solution path for the exact line search method:



Using the backtracking line search method, we need 40 iterations, and the solution is $x_1 = -2.1418$, $x_2 = 2.8582$.

Using the exact line search method, we need 47 iterations, and the solution is $x_1 = -2.1418,\ x_2 = 2.8582$.