Optimal Control

HW4

班級：航太四 A
姓名：吳柏勳
學號：407430635
座號：3

#1

```
clear;clc;close all
syms x y z lambda real

% define minimize function
L = -(x*y + 2*x*z + 3*y*z);
% define constrain function
f = x^2 + 2*y^2 + 3*z^2 - 1;
% define Hamiltonian
H = L + lambda*f;
% find gradient of the function
dH = [diff(H, x); diff(H, y); diff(H, z)];
% find Hessian matrix of the function
ddH = [diff(diff(H, x), x) diff(diff(H, x), y) diff(diff(H, x), z);
        diff(diff(H, y), x) diff(diff(H, y), y) diff(diff(H, y), z);
        diff(diff(H, z), x) diff(diff(H, z), y) diff(diff(H, z), z)];
% solve the problem
[x_, y_, z_, lambda_] = vpasolve([dH; f]==0, [x y z lambda]);

% check the solution is minimum
for i = 1:length(x_)
    dH_ = subs(dH, [x y z lambda], [x_(i) y_(i) z_(i) lambda_(i)]);
    ddH_ = subs(ddH, [x y z lambda], [x_(i) y_(i) z_(i) lambda_(i)]);

    if all(eig(ddH_) > -1e-20)
        sol = struct('x', double(x_(i)), 'y', double(y_(i)), 'z', double(z_(i)))
    end
end
```

```
.....................................................................
sol =

   -0.5410   -0.3938   -0.3639


sol =

    0.5410    0.3938    0.3639
```

#2

```
clear;clc;close all
syms x y z real
% setup the functions
L_ = -(x*y + 2*x*z + 3*y*z);
f_ = x^2 + 2*y^2 + 3*z^2 - 1;
dL_ = [diff(L_,x) diff(L_,y) diff(L_,z)];
df_ = [diff(f_,x) diff(f_,y) diff(f_,z)];

% setup the initial value and iteration parameters
state = [0.5; 0.4; 0.5];
iter = 1;
stop = false;
eta = 1;
k = .2;

while ~stop
    x_ = state(1);
    y_ = state(2);
    z_ = state(3);

    % Calculate the parameters
    L = double(subs(L_, [x, y, z], [x_, y_, z_]));
    f = double(subs(f_, [x, y, z], [x_, y_, z_]));
    dL = double(subs(dL_, [x, y, z], [x_, y_, z_]));
    df = double(subs(df_, [x, y, z], [x_, y_, z_]));
    df_star = df'/(df*df');
    lambda = (-dL*df_star)';
    dH = dL + lambda'*df;

    % Calculate the state update
    dy = -eta*df_star*f - k*dH';

    % Stop criteria
    if (norm(dy)>10 || norm(dy)<1e-6 || iter>1000); stop = true; end

    % Update the state
    state = state + dy;
    iter = iter + 1;
```

```
end
sol = state'
```

```
.....................................................................
```

```
sol =

    0.5410    0.3938    0.3639
```

#3

```
clear;clc;close all
L = @(state) -(state(1)*state(2) + 2*state(1)*state(3) + 3*state(2)*state(3));
x0 = [0 0 0];

opt = optimoptions('fmincon', 'Display', 'off');
sol = fmincon(L, x0, [], [], [], [], [], [], @constrains, opt)

function [c, ceq] = constrains(state)
    ceq = state(1)^2 + 2*state(2)^2 + 3*state(3)^2 - 1;
    c = [];
end
```

```
.....................................................................
```

```
sol =

    0.5410    0.3938    0.3639
```

#4

三個題目的結果相似，幾乎沒有差異，但在第一題出現了兩個結果，且兩個結果所算出的 L 皆相同，所以此問題可能有兩個全域解。