Answers to Problem Set 5
Group name: Ferienspass

Sebastian Kühnl: 5642348
Alexander Dück (as: reebyte): 5504077
Patrick Blank (as: paddyblank): 6729110
Christian Wierschem: 6729288

1 Question 1

2 Question 2

```
close all;
clear;
clc;

y_1 = 1.02;
Var_ln_eta = (0.25)^2;
Mu_ln_eta = -Var_ln_eta/2;
y_2 = 1.06;
n = 11;                                              %11 nodes
[ln_eta,w]=qnwnorm(n,Mu_ln_eta,Var_ln_eta);          % Distribution of log(eta)
eta=exp(w'*ln_eta);                                  % Expectation of eta

%%%%%%%%%%%%%% Question 1 %%%%%%%%%%%%%%%

p_1=y_1;                                             % Expected Payoff of Project 1
p_2=y_2*eta;                                         % Expected Payoff of Project 2

if p_1 < p_2
    disp('Project 2 yields the greater expected payoff.');
elseif p_1 == p_2
    disp('Project 1 and project 2 yield the same expected payoff.');
else
    disp('Project 1 yields the greater expected payoff.');
```

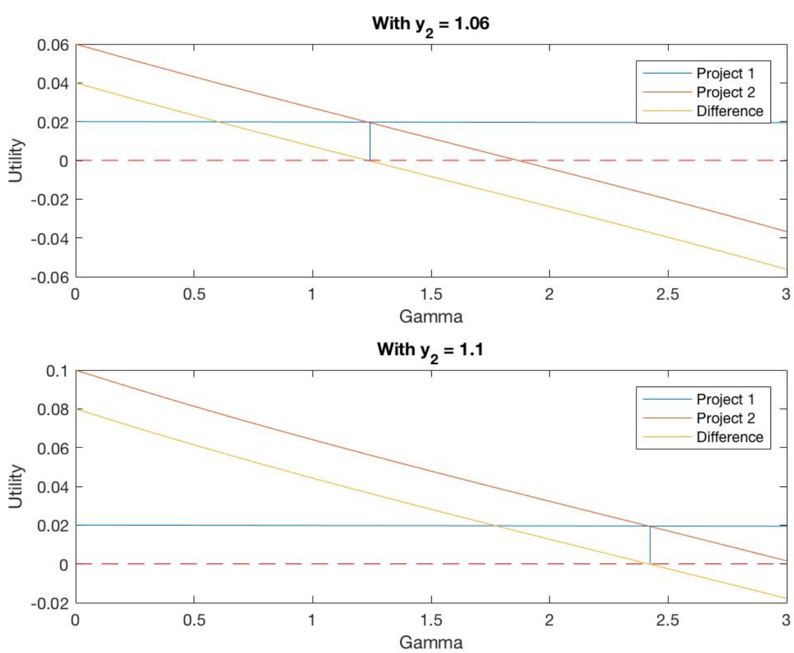
```
end

%%%%%%%%%%%%%% Question 2 %%%%%%%%%%%%%%%

gamma = 1.5;

u_1 = utility(y_1,gamma);
u_2 = w'*utility(exp(ln_eta)*y_2,gamma);

if u_1 < u_2
    disp('Household will prefer to invest in project 2.');
elseif u_1 == u_2
    disp('Household will be indifferent betwee project 1 and project 2.');
else
    disp('Household will prefer to invest in project 1.');
end

%%%%%%%%%%%%%% Question 3 %%%%%%%%%%%%%%%

gamma = linspace(0,3,100);                        %Gamma is now a vector of different values (for plotting only)
y_2= [1.06 1.1];
u_1=nan(1,100);
u_2=nan(1,100);
difference = nan(2,100);
%Plot intersection point
figure('Name','PS5Q2Sub3_Utility')
for j=1:2
for i=1:100
u_1(1,i) = utility(y_1,gamma(1,i));
u_2(1,i) = w'*utility(exp(ln_eta)*y_2(1,j),gamma(1,i));
difference(j,i) = u_2(1,i)-u_1(1,i);
end
[Min,Index] = min(abs(difference(j,:)));

subplot(2,1,j)
plot(gamma,u_1,gamma,u_2,gamma,difference(j,:))
line([min(gamma),max(gamma)],[0,0],'Color','red','LineStyle','--')
line([gamma(1,Index),gamma(1,Index)],[0,u_2(1,Index)])
title(['With y_2 = ', num2str(y_2(1,j))])
```

```
legend('Project 1','Project 2','Difference')
xlabel('Gamma')
ylabel('Utility')
end

figure('Name','PS5Q2Sub3_Quad_Diff')
plot (gamma, (difference(1,:)).^2,gamma, (difference(2,:)).^2)
title('Squared differnces')
xlabel('Gamma')
ylabel('Difference in Utility')

%Find intersection via grid search
[Min1,Index1] = min(abs(difference(1,:)));
[Min2,Index2] = min(abs(difference(2,:)));

disp('As one can see clearly, changing y_2 changes the gamma at which both projects yield the same expected utility.');
fprintf(['\n In the first plot, gamma = ', num2str(gamma(1,Index1)),' produced the smallest difference. \n For a value close to this, the household will be indifferent between the two projects, given y_2 = 1.06 \n '] );
fprintf(['\n In the second plot, gamma = ', num2str(gamma(1,Index2)),' produced the smallest difference. \n For a value close to this, the household will be indifferent between the two projects, given y_2 = 1.1 \n'] );

%Find intersection point numerically using Newton. This is equal to finding
%the gamma for which the difference is equal to zero --> Root finding Problem

params = [ln_eta;w;y_1;y_2(1,1)];
f = @(x) Utility_Difference(x,params);
y = [1 2];                                              % initial guess
cc =[0.1;0.1;1000];                                     % criteria
y_sol_1=newton(f,y(1,1),cc);

%More to come here!!

%Newton
```

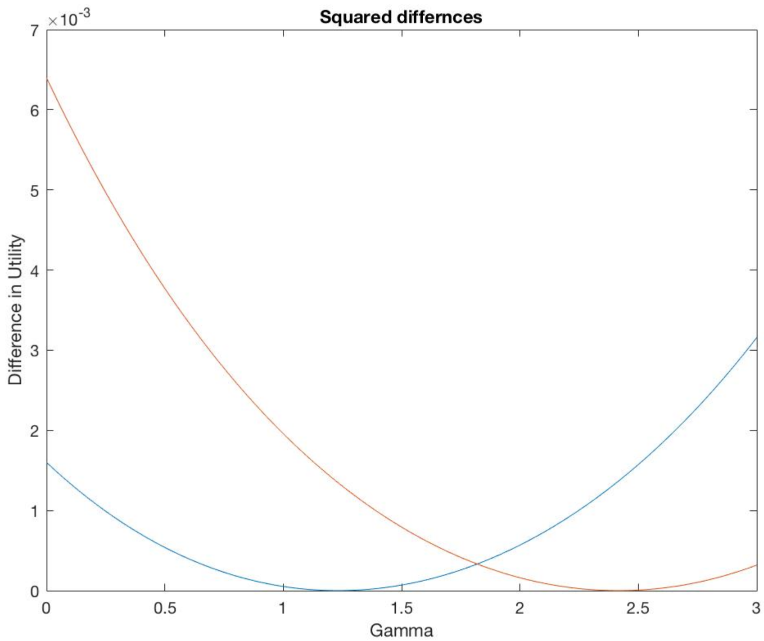
```
function [x,fx,ef,iter] = newton(f,x,cc)

% convergence criteria
tole = cc(1,1); told = cc(2,1); maxiter = cc(3,1);

% newton algorithm
for j = 1:maxiter
    [fx,dfx] = f(x);

    xp = x - dfx\fx;
    D = (norm(x-xp) <= tole*(1+norm(xp)) && norm(fx) <= told);
    if D == 1
        break;
    else
        x = xp;
    end
end
ef = 0; if D == 1; ef = 1; end
iter = j;
end

%Function whose root if to be found
function [fx,dfx] = Utility_Difference(x,y)

weight=[y(1,1);y(2,1);y(3,1);y(4,1);y(5,1);y(6,1);y(7,1);y(8,1);y(9,1);y(10,1);y(11,1)];
rv=[y(12,1);y(13,1);y(14,1);y(15,1);y(16,1);y(17,1);y(18,1);y(19,1);y(20,1);y(21,1);y(22,1)];

y_1=y(23,1);
y_2=y(24,1);

fx = utility(y_1,x) - weight'*utility(exp(rv)*y_2,x);

dfx = derivate(y_1,x)-weight'*derivate(exp(rv)*y_2,x);


end

% Declare CRRA utility function
function u= utility(x,gamma)
    if gamma == 1
        u=log(x);
    else
        u=(x.^(1-gamma)-1)./(1-gamma);
```

```
            disp(u);
        end
end

function dgu = derivate(x,gamma)
    if gamma == 1
         dgu = 0.1; %Guess?
    else
        dgu=((x.^(1-gamma)-1)-(x.^(1-gamma)-1).*(1-
            gamma)-1)./((1-gamma).^2);
    end

end
```

The approximate point where the household is indifferent between the two projects can be found via grid search. Graphically, the point where the two utility curves intersect in the point where the household becomes indifferent between the two projects. In both graphs, this point is marked by the vertical line going upwards from the horizontal line crossing through zero. Alternatively,



this point can also be found where the squared difference comes close to zero. However, for a truly satisfying result, grid search is not sufficient. The root of the difference function has to be found numerically. Here, the Newton algorithm is applied. For this, the first derivative with respect to γ has to be computed.



Since this is possible analytically, the derivation is provided below.

$$u(c_t) = \frac{c_t^{1-\gamma} - 1}{1-\gamma} = \frac{c_t^{1-\gamma}}{1-\gamma} - \frac{1}{1-\gamma} \tag{1}$$
$$= \frac{e^{(1-\gamma)\ln(c_t)}}{1-\gamma} - \frac{1}{1-\gamma} \tag{2}$$

The first derivative with respect to γ can then be calculated:

$$\frac{\partial u(c_t)}{\partial \gamma} = \frac{(-1)e^{(1-\gamma)\ln(c_t)}(1-\gamma) - (-1)e^{(1-\gamma)\ln(c_t)}}{(1-\gamma)^2} - \frac{0-(-1)*1}{(1-\gamma)^2} \tag{3}$$
$$= \frac{e^{(1-\gamma)\ln(c_t)} - e^{(1-\gamma)\ln(c_t)}(1-\gamma)}{(1-\gamma)^2} - \frac{1}{(1-\gamma)^2} \tag{4}$$
$$= \frac{e^{(1-\gamma)\ln(c_t)} - e^{(1-\gamma)\ln(c_t)}(1-\gamma) - 1}{(1-\gamma)^2} \tag{5}$$
$$= \frac{c_t^{1-\gamma} - c_t^{1-\gamma}(1-\gamma) - 1}{(1-\gamma)^2} \tag{6}$$
$$\tag{7}$$

Which can be used for the calculation of the Newton Algorithm. In the case that $\gamma = 1$

$$\frac{\partial u(c_t)}{\partial \gamma} = \frac{\partial \ln c_t}{\partial \gamma} = 0. \tag{8}$$

$$\tag{9}$$