Computational Economics: Problem Set 3

Yangming Bao, ID: 5601239
Cheung Ying Lun, ID: 5441897

1 Problem 1: Simple Functions

The extrema of $f_1(x)$ and $f_2(x)$ are respectively $f_1(-0.5598) = 3.0152$ and $f_2(1) = -0.3679$.

2 Problem 2: Consumption Savings Problem

Writing the Lagrangian

$$\mathcal{L} = \sum_{t=1}^{T} \beta^t u(c_t) - \lambda_t(c_t + a_{t+1} - a_t(1+r) + w_t)$$

we have the first order conditions

$$\beta^t c_t^{-\theta} = \lambda_t \qquad (\tfrac{\partial \mathcal{L}}{\partial c_t})$$
$$\lambda_t = (1+r)\lambda_{t+1} \qquad (\tfrac{\partial \mathcal{L}}{\partial a_{t+1}})$$

We thus obtain the optimal consumption path being

$$c_{t+1} = [(1+r)\beta]^{\frac{1}{\theta}} c_t.$$

Numerical solution for any given θ can be calculated with `Q2.m` which make use of the MATLAB function `fmincon`.

If we now assume $\theta = 1$, the function does not work since the denominator of $u(c_t)$ is zero and the function is not defined. However, we can use the concept of limit, i.e. solve the problem with $\theta \to 1$ instead, by letting $\theta = 1 + \epsilon$ for ϵ being a small number.

3 Problem 3: The Consumption-Savings Problem with Human Capital

1. Consider the Lagrangian

$$\mathcal{L} = f(c_1, c_2) + \lambda_1 g_1 + \lambda_2 g_2 + \mu_1 h_1 + \mu_2 h_2$$

where

$$f(c_1, c_2) = \frac{c_1^{1-\gamma} - 1}{1-\gamma} + \beta \frac{c_2^{1-\gamma} - 1}{1-\gamma}.$$

The Kuhn-Tucker conditions are

$$\begin{aligned}
&\frac{\partial \mathcal{L}}{\partial c_1} = c_1^{-\gamma} + \mu_1 \overset{!}{=} 0 \\
&\frac{\partial \mathcal{L}}{\partial c_2} = \beta c_2^{-\gamma} + \mu_2 \overset{!}{=} 0 \\
&\frac{\partial \mathcal{L}}{\partial i_k} = \lambda_1 + \mu_1 + \mu_2[-(1+r_k)] \overset{!}{=} 0 \\
&\frac{\partial \mathcal{L}}{\partial i_h} = \lambda_2 + \mu_1 + \mu_2[-\eta r_h((1-\delta_h)h_1 + i_h)^{\eta-1}] \overset{!}{=} 0 \\
&\lambda_1 \geq 0, \qquad g_1 = (1-\delta_k)k_1 + i_k \geq 0, \qquad \lambda_1 g_1 = 0 \\
&\lambda_2 \geq 0, \qquad g_2 = i_h \geq 0, \qquad \lambda_2 g_2 = 0
\end{aligned}$$

2. Notice that if $\alpha_i > 0$, $\alpha_i^+ > 0$ and $\alpha_i^- = 0$, which is equivalent to $\lambda_i > 0$ and $g_i = 0$. Similarly, the Kuhn-Tucker conditions are the same as in the case of $\lambda_i = 0$ in Part 2 when $\alpha_i < 0$.

3. Eight unknowns, namely, $\{c_1, c_2, \mu_1, \mu_2, \alpha_k, \alpha_h, i_k, i_h\}$.

4. See `func.m` in `Q3`.

5. The solutions in order $\{c_1, c_2, \mu_1, \mu_2, \alpha_k, \alpha_h, i_k, i_h\}$ are (also list in Table 1):

   (a) $\{5.4546, 5.6052, -0.0336, -0.0306, -0.8178, 0.0925, -0.2811, 0\}$.

   (b) $\{1.8993, 1.9517, -0.2772, -0.2520, -0.6375, -0.3798, -0.5435, 0.1443\}$.

   (c) $\{0.9498, 1.0241, -1.1085, -0.9153, 0.3188, -0.6975, -0.9500, 0.4865\}$.

6. From the result of α_k, α_h, we can calculate the corresponding multiplier λ_k and λ_h. Results are in Table 1, where we can see only in the second case the two

	case 1	case 2	case 3
c_1	5.4546	1.8993	0.9498
c_2	5.6052	1.9517	1.0241
u_1	-0.0336	-0.2772	-1.1085
u_2	-0.0306	-0.2520	-0.9153
α_k	-0.8178	-0.6375	0.3188
α_h	0.0925	-0.3798	-0.6975
i_k	-0.2811	-0.5435	-0.9500
i_h	0.0000	0.1443	0.4865
λ_k	0.0000	0.0000	0.1016
λ_h	0.0086	0.0000	0.0000

Table 1: Solutions

constraints are unbinding as the multipliers (λ) are 0. While in case 1, constraint for i_h is binding and in case 3, constraint for k_2 is binding.

4 Problem 4: A Simple Portfolio Choice Problem

1. Yes, since the CRRA utility function is convex and the set of possible α is a closed convex set. The problem is convex means that if a local minimum is found, it would be the global minimum as well.

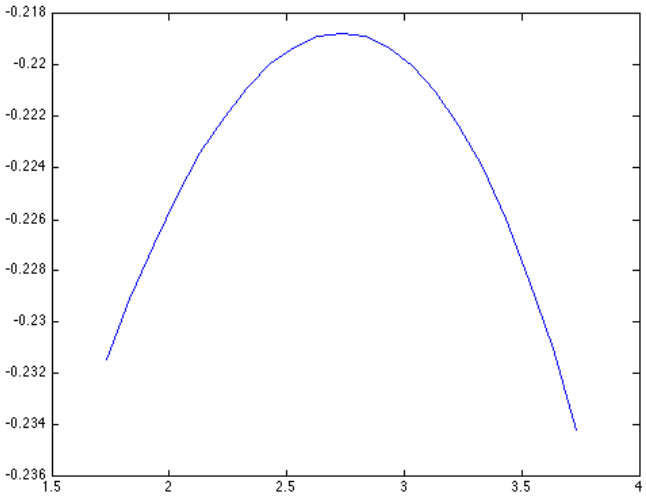
2. (a) Note that w_0 is not random, we can pull it out of the expectation operator. The problem becomes

$$\max_{\alpha} \left\{ w_0^{\phi} \mathbb{E} \left[\frac{1}{\phi} [1 + r^f + \alpha(r - r^f)]^{\phi} \right] \right\}.$$

   Notice that w_0 is just a constant independent of α, the optimal portfolio share is independent of initial wealth.

   (b) The optimal portfolio share is $\alpha^* = 2.7331 > 1$, i.e. the agent short-sells the risk-free asset (i.e. borrow) and buy the risky portfolio.

3. (a) The constraint implies the agent cannot borrow or short sell.



(b) The solutions solved by `fminbnd` and `fmincon` are respectively 0.9999 and 1. The solutions are different because the constraint in `fminbnd` is $0 < \alpha < 1$ while in `fmincon` it is $0 \leq \alpha \leq 1$.

(c) Since the agent chooses to borrow when the constraint is not imposed, the constraint is binding. The best the agent can do is to invest as much as one can in the risky asset, thus yielding $\alpha^* = 1$.

Question 1

`Q1.m`

```
f1=@(x)func1(x);
f2=@(x)func2(x);
[x1,fx1,ef1] = newton(f1,0,[10^-6;10^-6;1e4]);
[x2,fx2,ef2] = newton(f2,0,[10^-6;10^-6;1e4]);
```

`func1.m`

```
function [f1,df1,ddf1] = func1(x)
f1 = 2*x^3-x^2-3*x+2;
df1 = 6*x^2-2*x-3;
ddf1 = 12*x-2;
end
```

`func2.m`

```
function [f2, df2, ddf2] = func2(x)

f2 = -x*exp(-x);
df2 =  (x-1)*exp(-x);
ddf2 = (2-x)*exp(-x);
end
```

`newton.m`

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

tole = cc(1,1);
told = cc(2,1);
maxiter = cc(3,1);
ef = 0;

for j = 1:maxiter
    [fx,dfx,ddfx] = f(x);
    xp = x-ddfx\dfx';
    if norm(x-xp)<=tole*(1+norm(xp))
        ef=2;break;
    else
        x=xp;
    end
```

```
16 end
17
18 if norm(dfx)<=told*(1+norm(xp))
19     if ef == 2; ef=1;
20     else ef =3;
21     end
22 end
```

Question 2

Q2.m

```
1  % Problem 2
2  clear, clc
3
4  %% Parameters
5  theta    = 0.5;
6  T        = 10;
7  W        = [10;zeros(T-1,1)];
8  a0       = 0;
9  beta     = 0.99;
10 r        = 0.05;
11
12 F        = @(C)Obj(C,theta,beta);
13 Cons     = @(C)NonLCon(C,W,a0,r);
14
15 %% Initialization
16 cc       = [0.001;0.001;10000];
17 Hf       = eye(T);
18 C0       = ones(T,1);
19
20 %% Optimization
21 C        = fmincon(@(C)Obj(C,theta,beta),C0,-eye(T),zeros(T,1),[],[],[],[],@
       (C)NonLCon(C,W,a0,r));
22
23 %% theta->1
24 theta    = 1+1e-7;
25 C_star   = fmincon(@(C)Obj(C,theta,beta),C0,-eye(T),zeros(T,1),[],[],[],[],@
       (C)NonLCon(C,W,a0,r));
```

Obj.m

```
function F = Obj(C,theta,beta)
%OBJ computes the function value of the objective function.
%   INPUT:
%           C = T*1 vector of comsumption stream
%       theta = relative risk aversion coefficient
%        beta = discount factor
%
%   OUTPUT:
%           F = function value

T       = length(C);
dis_fac = beta.^((0:T-1)');
if theta==1
    theta = theta+1e-7;
end
F       = -sum(dis_fac.*((C.^(1-theta)-1)/(1-theta)));

disp(['F = ',num2str(-F)])

end
```

NonLCon.m

```
function [Con,ConEq] = NonLCon(C,W,a0,r)
%This function computes the constraint conditions.

T       = length(C);
A       = [a0;zeros(T,1)];
for t=1:T
    A(t+1) = (1+r)*A(t)-C(t)+W(t);
end

Con     = -A(T+1);

ConEq   = -(C+A(2:end)-A(1:end-1)*(1+r)-W);
end
```

Question 3

Q3.m

```
%Q3

clear, clc

ini_val = ones(8,1);


% i)
k1=1; h1=5;
KuhnTucker = @(varible)func(varible,k1,h1);
solution1 = fsolve(KuhnTucker,ini_val);
% lambda = alpha+
lambdak1 = (max(0,solution1(5)))^2;
lambdah1 = (max(0,solution1(6)))^2;

% ii)
k1=1; h1=1;
KuhnTucker = @(varible)func(varible,k1,h1);
solution2 = fsolve(KuhnTucker,ini_val);
lambdak2 = (max(0,solution2(5)))^2;
lambdah2 = (max(0,solution2(6)))^2;

% iii)
k1=1; h1=0.2;
KuhnTucker = @(varible)func(varible,k1,h1);
solution3 = fsolve(KuhnTucker,ini_val);
lambdak3 = (max(0,solution3(5)))^2;
lambdah3 = (max(0,solution3(6)))^2;
```

func.m

```
function KuhnTucker = func(varible,k1,h1)
% This function deals with the Kuhn Tucker equations for Q3

%varibles
c1 = varible(1);
c2 = varible(2);
u1 = varible(3);
```

```
8  u2 = varible(4);
9  alphak = varible(5);
10 alphah = varible(6);
11 ik = varible(7);
12 ih = varible(8);
13
14 %parameters
15 k = 2;
16 gamma = 2;
17 beta = 0.96;
18 rk = 0.1;
19 rh = 1.4;
20 eta = 0.8;
21 sigma_k = 0.05;
22 sigma_h = 0.05;
23
24 %Garcia-Zangwill trick
25 alphak_plus = (max(0,alphak))^k;
26 alphak_minus = (max(0,-alphak))^k;
27 alphah_plus = (max(0,alphah))^k;
28 alphah_minus = (max(0,-alphah))^k;
29
30 % equations for KuhnTucker
31 KuhnTucker = zeros(8,1);
32
33 KuhnTucker(1) = c1^(-gamma)+u1;
34 KuhnTucker(2) = beta*c2^(-gamma)+u2;
35 KuhnTucker(3) = alphak_plus+u1-u2*(1+rk);
36 KuhnTucker(4) = alphah_plus+u1-u2*rh*eta*((1-sigma_h)*h1+ih)^(eta-1);
37 KuhnTucker(5) = alphak_minus-((1-sigma_k)*k1+ik);
38 KuhnTucker(6) = alphah_minus-ih;
39 KuhnTucker(7) = c1-(rk*k1+rh*h1^eta-ik-ih);
40 KuhnTucker(8) = c2-((1+rk)*((1-sigma_k)*k1+ik)+rh*((1-sigma_h)*h1+ih)^eta);
41
42
43
44
45 end
```

Question 4

`Q4.m`

```
%Problem 4
clear, clc

%% Parameters
rf  = 0.02;
rl  = -0.08;
rh  = 0.12;
phi = -3;
p   = 0.1;

%% Q2
f    = @(alpha)-obj_fun(alpha,phi,rf,rh,rl,p);
alpha_q2 = fminunc(f,0.5);
plot(alpha_q2-1:0.1:alpha_q2+1,-f(alpha_q2-1:0.1:alpha_q2+1))

%% Q3
alpha_q3a = fminbnd(f,0,1);
alpha_q3b = fmincon(f,0.5,[],[],[],[],0,1);
```

`obj_fun.m`

```
function U = obj_fun(alpha,phi,rf,rh,rl,p)
%This function calcualtes the expected utility value of given parameters.

U   = p.*(1./phi).*(1+rf+alpha.*(rl-rf)).^phi+...
        (1-p).*(1./phi).*(1+rf+alpha.*(rh-rf)).^phi;

end
```