ECON 6140 - Problem Set # 3

Julien Manuel Neves

March 3, 2018

Open Sector Growth Model via Dynamic Programming

(1) Let $\tau_t^c = \tau^c$ and $\tau_t^x = \tau^x$ be constant. The functional equation is given by

$$(Tv)(k) = \max_{0 \le k' \le \frac{f(k)}{(1+\tau^x)} + (1-\delta)k} \left\{ u\left(\frac{f(k)}{(1+\tau^c)} - \frac{(1+\tau^x)}{(1+\tau^c)}(k' - (1-\delta)k) \right) + \beta v(k') \right\}$$

(2) Our state variable is the current value of k and our control is the future value of capital k'.

In our setting, this makes sense since we can't affect the current value of capital, but we can decide how much to consumes/invest to pinpoint capital in the next period.

Note that we can't set our state space equal to $[0, 1]$. In fact, part 4-7 shows a steady state value of capital that is not even remotely close to be inside $[0, 1]$.

(3) Note that T is defined on a the set of bounded function defined on $[0.\infty)$.

To show that T is a contraction, we can use Blackwell's theorem. In fact, we only need to show monotonicity and discounting for T

(i) Monotonicity

Let $v(x) \le w(x)$ and $g_v(k)$, $g_w(k)$ be the respective policy function. Then,

$$\begin{aligned}
(Tv)(k) &= \max_{0 \le k' \le \frac{f(k)}{(1+\tau^x)} + (1-\delta)k} \left\{ u\left(\frac{f(k)}{(1+\tau^c)} - \frac{(1+\tau^x)}{(1+\tau^c)}(k' - (1-\delta)k) \right) + \beta v(k') \right\} \\
&= u\left(\frac{f(k)}{(1+\tau^c)} - \frac{(1+\tau^x)}{(1+\tau^c)}(g_v(k) - (1-\delta)k) \right) + \beta v(g_v(k)) \\
&\le u\left(\frac{f(k)}{(1+\tau^c)} - \frac{(1+\tau^x)}{(1+\tau^c)}(g_v(k) - (1-\delta)k) \right) + \beta w(g_v(k)) \\
&\le u\left(\frac{f(k)}{(1+\tau^c)} - \frac{(1+\tau^x)}{(1+\tau^c)}(g_f(k) - (1-\delta)k) \right) + \beta w(g_w(k)) \\
&= \max_{0 \le k' \le \frac{f(k)}{(1+\tau^x)} + (1-\delta)k} \left\{ u\left(\frac{f(k)}{(1+\tau^c)} - \frac{(1+\tau^x)}{(1+\tau^c)}(k' - (1-\delta)k) \right) + \beta w(k') \right\}
\end{aligned}$$

$\Rightarrow (Tv)(k) \le (Tw)(k)$

(ii) Discounting

$$
\begin{aligned}
(Tv + a)(k) &= \max_{0 \le k' \le \frac{f(k)}{(1+\tau^x)} + (1-\delta)k} \left\{ u\left(\frac{f(k)}{(1+\tau^c)} - \frac{(1+\tau^x)}{(1+\tau^c)}(k' - (1-\delta)k) \right) + \beta(v(k') + a) \right\} \\
&= \max_{0 \le k' \le \frac{f(k)}{(1+\tau^x)} + (1-\delta)k} \left\{ u\left(\frac{f(k)}{(1+\tau^c)} - \frac{(1+\tau^x)}{(1+\tau^c)}(k' - (1-\delta)k) \right) + \beta v(k') \right\} + \beta a \\
&= (Tv)(k) + \beta a
\end{aligned}
$$

Hence, T is a contraction.

(4) Figure 1 show the value function around the steady state and Figure 2 shows the policy function.

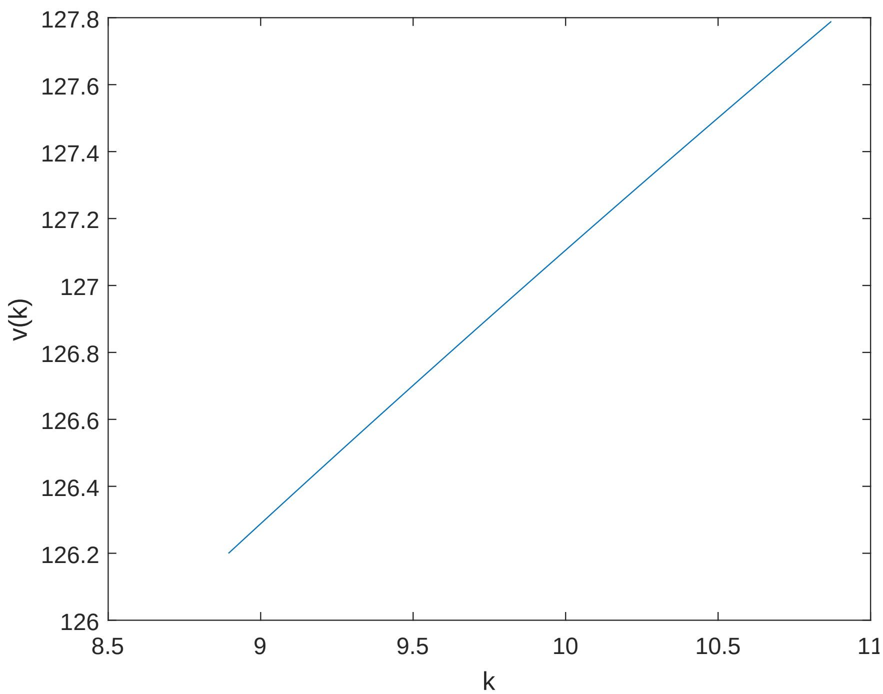

Figure 1: Value function

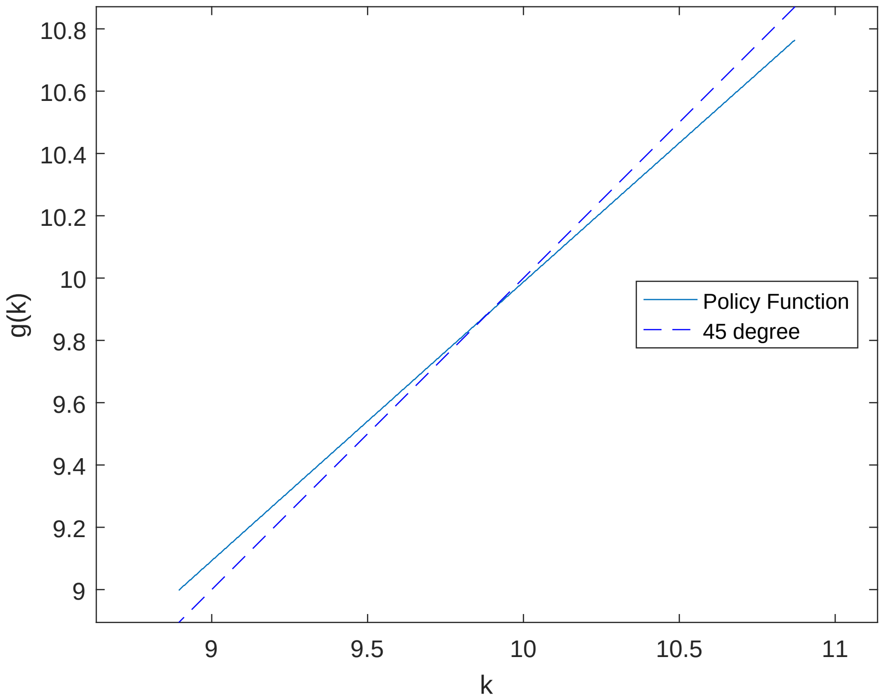

Figure 2: Policy function

(5) Figure 3 show the time path of capital starting from $k_0 = 0.9k^*$.

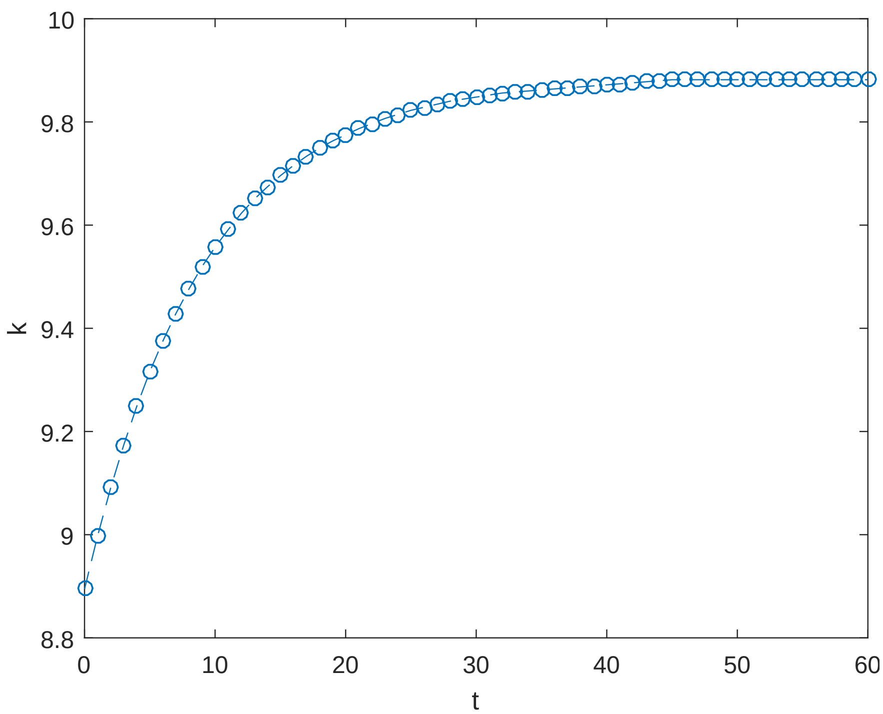

Figure 3: Time path for consumption and capital

(6) We use the shooting algorithm for this question. For sake of clarity, we provide the

dynamic equations:

$$k_{t+1} = \frac{k_t^\alpha}{(1+\tau_t^x)} + (1-\delta)k_t - c_t\frac{(1+\tau_t^c)}{(1+\tau_t^x)}$$

$$c_{t+1} = c_t\left(\frac{(1+\tau_t^c)(1+\tau_{t+1}^x)}{(1+\tau_t^x)(1+\tau_{t+1}^c))}\right)^{\frac{1}{\sigma}}\left(\frac{\alpha\beta k_t^{\alpha-1}}{(1+\tau_{t+1}^x)} + \beta(1-\delta))\right)^{\frac{1}{\sigma}}$$

Figure 4 show the time path of capital for a permanent tax increase. Note that the consumption jumps at $t=1$ to get back to the steady path.

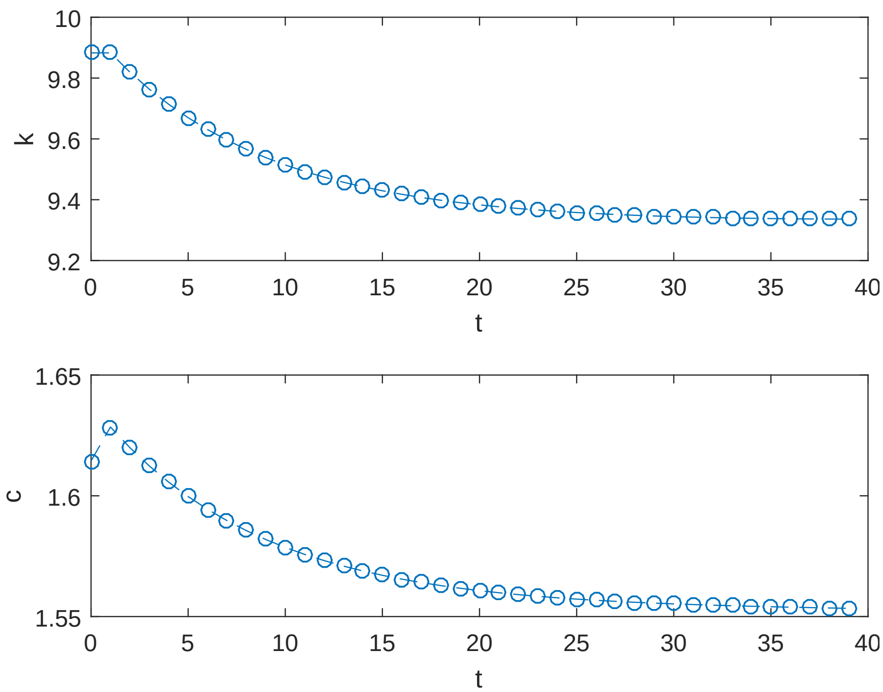

Figure 4: Time path for consumption and capital

(7) Again we use the shooting algorithm for this question. Figure 5 show the time path of capital for a temporary tax increase.

In this setting, the steady path moves both at $t=1$ and $t=10$ explaining the two jumps in consumption to insure that we will reach the steady state in the future.

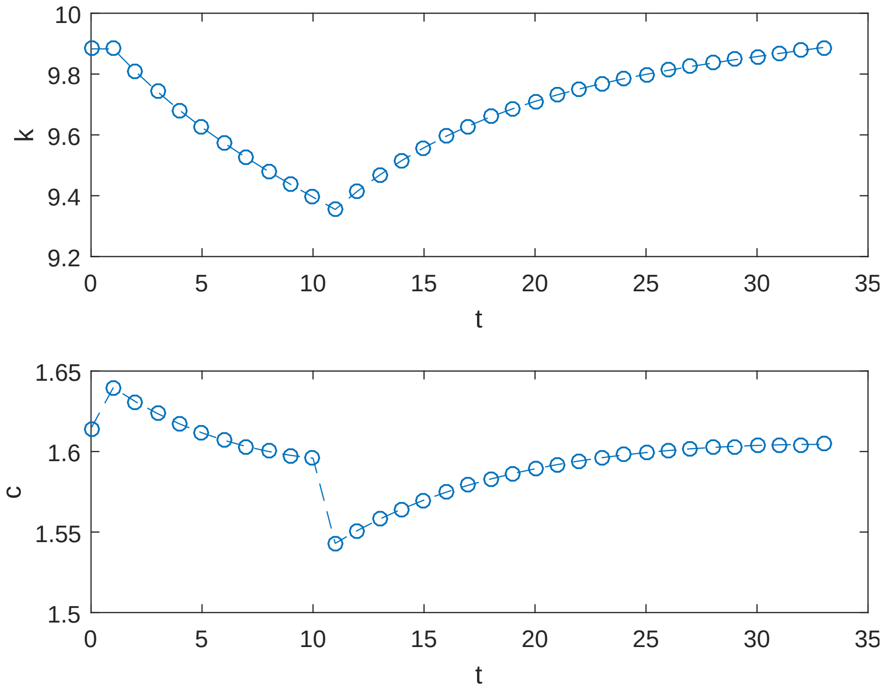

Figure 5: Time path for consumption and capital

Ben-Porath

(1) (i) $\Rightarrow$

Let $Y^* = \max_{s(t),h(t)} \int_0^\infty \exp(-rt)w(t)(1-s(t))h(t)dt$ and $\{c(t), s(t), h(t)\}_{t=0}^\infty$ solve the household problem.

Let's assume that $\{s(t), h(t)\}$ does not maximizes $\int_0^\infty \exp(-rt)w(t)(1-s(t))h(t)dt$, i.e. $\exists Y$ such that

$$\int_0^\infty \exp(-rt)w(t)(1-s(t))h(t)dt = Y < Y^*$$

Let $c'(t) = c(t) + \epsilon$ for some $\epsilon > 0$. Clearly, $c' \succ c$ due to strictly increasing nature of $u(\cdot)$. If, we can find an ϵ such that $c'(t)$ is feasible, then we have a contradiction with $\{c(t), s(t), h(t)\}_{t=0}^\infty$ solving the household problem.

$$\begin{aligned}
\int_0^\infty \exp(-rt)c'(t)dt &= \int_0^\infty \exp(-rt)c(t)dt + \int_0^\infty \exp(-rt)\epsilon dt \\
&\leq \int_0^\infty \exp(-rt)w(t)(1-s(t))h(t)dt + \frac{\epsilon}{r} \\
&\leq Y + \frac{\epsilon}{r}
\end{aligned}$$

Since $Y < Y'$, there exists an $\epsilon > 0$ such that $Y + \frac{\epsilon}{r} \leq Y^*$. Hence, there exists some $\{s'(t), h'(t)\}_{t=0}^\infty$ such that $\int_0^\infty \exp(-rt)w(t)(1-s'(t))h'(t)dt = Y^*$ which implies that $\{c'(t), s'(t), h'(t)\}_{t=0}^\infty$ is feasible, i.e. we have a contradiction.

(ii) $\Leftarrow$

Let $\{s(t), h(t)\}_{t=0}^{\infty}$ solve $\max_{s(t),h(t)} \int_0^\infty \exp(-rt)w(t)(1-s(t))h(t)dt = Y$.
Let $\{c(t)\}_{t=0}^{\infty}$ solve the household problem and $\{c'(t)\}_{t=0}^{\infty}$ be any solution such that $c' \succ c$.
Since $\{c(t)\}_{t=0}^{\infty}$ solves the household problem with the following budget constraint,

$$\int_0^\infty \exp(-rt)c(t)dt \leq Y$$

we have that

$$\int_0^\infty \exp(-rt)c'(t)dt > Y$$

Therefore, any $\{s'(t), h'(t)\}_{t=0}^{\infty}$ that supports $\{c'(t)\}_{t=0}^{\infty}$ it must be the case that $int_0^\infty \exp(-rt)w(t)(1-s'(t))h'(t)dt > Y$ which is a contradiction.
Hence, $\{c(t), s(t), h(t)\}_{t=0}^{\infty}$ solves the household problem.

(2) The current Hamiltonian is given by

$$\mathcal{H}(s, h, \mu) = w(t)(1-s(t))h(t) + \mu(t)(\phi(s(t)h(t)) - \delta_h h(t))$$

where h is the state variable, s the control variable and μ the costate variable.

The FOCs, TVC, and constraint are given by

$$\begin{aligned}
s &: -w(t)h(t) + \mu(t)h(t)\phi'(s(t)h(t)) = 0 \\
h &: w(t)(1-s(t)) + \mu(t)(s(t)\phi'(s(t)h(t)) - \delta_h) = r\mu(t) - \dot{\mu}(t) \\
TVC &: \lim_{t\to\infty} \exp(-rt)\mu(t)h(t) = 0 \\
&: \dot{h} = \phi(s(t)h(t)) - \delta_h h(t)
\end{aligned}$$

(3) First, we differentiate the first FOC

$$\mu(t)\phi''(x(t))\dot{x}(t) + \phi'(x(t))\dot{\mu}(t) = \dot{w}(t)$$

We can then combine the FOCs in the following way

$$\begin{aligned}
-\dot{\mu}(t) &= w(t)(1-s(t)) + \frac{w(t)}{\phi'(x(t))}(s(t)\phi'(x(t)) - \delta_h - r) \\
-\dot{\mu}(t) &= w(t) + \frac{w(t)}{\phi'(x(t))}(-\delta_h - r) \\
-\phi'(x(t))\dot{\mu}(t) &= \phi'(x(t))w(t) + w(t)(-\delta_h - r) \\
\mu(t)\phi''(x(t))\dot{x}(t) - \dot{w}(t) &= \phi'(x(t))w(t) - w(t)(\delta_h + r) \\
\Rightarrow \dot{x}(t) &= \frac{1}{\mu(t)\phi''(x(t))}(\dot{w}(t) + w(t)(\phi'(x(t)) - \delta_h - r))
\end{aligned}$$

(4) In steady state, $\dot{\mu} = \dot{h} = 0$. Hence,

$$\begin{aligned} 0 &= \phi(x^*(t)) - \delta_h h^*(t) \\ \Rightarrow h^*(t) &= \frac{\phi(x^*(t))}{\delta_h} \end{aligned}$$

and

$$\begin{aligned} \phi'(x^*(t)) &= \delta_h + r \\ \Rightarrow x^*(t) &= \phi^{'-1}(\delta_h + r) \end{aligned}$$

This implies that

$$\begin{aligned} \Rightarrow h^*(t) &= \frac{\phi(\phi^{'-1}(\delta_h + r))}{\delta_h} \\ \Rightarrow s^*(t) &= \frac{x^*(t)}{h^*(t)} = \frac{\delta_h \phi^{'-1}(\delta_h + r)}{\phi(\phi^{'-1}(\delta_h + r))} \end{aligned}$$

We have that h^* and s^* are uniquely determined by δ_h, r and the shape of $\phi(\cdot)$. Note that $\phi(\cdot)$ is strictly increasing while $\phi^{'-1}(\cdot)$ is strictly decreasing.

Additionally, $w(t)$ does not influence the steady state human capital accumulation and the schooling decision. This contrasts the fact that $w(t)$ does influence the path of $x(t)$, but not the steady state.

Code

```
%% Value Function Iteration + Shooting Algorithm
% Course: ECON 6140
% Version: 1.0
% Author: Julien Neves
clear, clc;

%% Question 4
n = 1000;          % Size of grid

alpha = 0.33;    % labor share
delta = 0.05;    % depreciation of capital
sigma = 0.5;     % CRRA
beta = 0.98;      % discount factor
tau_x = 0.01;    % technology
tau_c = 0.01;    % technology

crit = 1;          % Initialize convergence criterion
```

```
tol = .001;      % Convergence tolerance

T = 60;

% Grid
kstar = ((alpha*beta)/((1-beta*(1-delta))*(1+tau_x)))^(1/(1-alpha)); % Steady state
cstar = (kstar^alpha-delta*kstar*(1+tau_x))/(1+tau_c);

ub_k = 0.9*kstar;        % Upper bound
lb_k = 1.1*kstar;        % Lower bound
kgrid = linspace(lb_k,ub_k,n)'; % Create grid

k0 = 0.9*kstar;

% Empty
val_temp = zeros(n,1);  % Initialize temporary value function vector
val_fun = zeros(n,1);   % Initialize value function vector
pol_fun = zeros(n,1);   % Initialize policy function vector

ite = 0;    % Initialize iteration counter

% Value function iteration
while crit>tol ;
    % Iterate on k
    for i=1:n
        c = (kgrid(i)^alpha + (1+tau_x)*((1-delta)*kgrid(i) - kgrid))/(1+tau_c); % Compute consumption for kt
        utility_c = c.^(1-sigma)/(1-sigma); % Compute utility for every ct
        utility_c(c<=0) = -Inf; % Set utility to -Inf for c<=0
        [val_fun(i),pol_fun(i)] = max(utility_c + beta*val_temp); % Solve bellman equation
    end
    crit = max(abs(val_fun-val_temp));  % Compute convergence criterion
    val_temp = val_fun; % Update value function
    ite = ite + 1 % update iteration
end

% Value function
figure(1)
plot(kgrid,val_fun)
xlabel('k')
```

```
ylabel('v(k)')
print -depsc fig1.eps

% Policy function
figure(2)
plot(kgrid,kgrid(pol_fun))
axis equal
hold on
plot(kgrid,kgrid, '--b')
xlabel('k')
ylabel('g(k)')
legend('Policy Function','45 degree', 'Location', 'best')
print -depsc fig2.eps


%% Question 5
kpath=zeros(1,T);
[~,kpath(1)]=min(abs(kgrid-k0));

for i = 1:T
    kpath(i+1) = pol_fun(kpath(i));
end
kpath = kgrid(kpath);

% Capital path
figure(3)
plot(0:length(kpath)-1,kpath, '--o');
xlabel('t') % x-axis label
ylabel('k') % y-axis label
print -depsc fig3.eps

%% Question 6
kstar_old = kstar;
cstar_old = cstar;

N = 100000; % grid size
T = 100;     % time periods
tol = .01;       % Convergence tolerance

tau_c = repmat(.03,1,T);    % technology
tau_x = repmat(.05,1,T);    % technology

kstar = ((alpha*beta)/((1-beta*(1-delta))*(1+tau_x(T))))^(1/(1-alpha)); % Steady state
cstar = (kstar^alpha-delta*kstar*(1+tau_x(T)))/(1+tau_c(T));    % c
```

```
steady state

k0 = kstar_old; % starting k value

k = zeros(1,T+1);   % initial k path vector
c = zeros(1,T+1);   % initial c path vector

lb_k = 0.9*kstar;   % lower bound of k axis
ub_k = 1.1*kstar;   % upper bound of k axis

lb_c = 0.9*cstar;   % lower bound of c axis
ub_c = 1.1*cstar;   % lower bound of c axis

kgrid = linspace(lb_k,ub_k,n)';   % k axis
cgrid = linspace(lb_c,ub_c,n)';   % c axis

crit = 1;   % initialize tolerance criteria
ite = 1;    % initialize iteration

while (crit>tol && ite<=N)
    k(1) = k0;  % set starting k0
    c(1) = cgrid(ite); % pick c0
    for t = 1:T-1
        k(t+1) = k(t)^alpha/(1+tau_x(t))+(1-delta)*k(t)-c(t)*(1+tau_c(t))/(1+tau_x(t)); % compute k(t+1)
        A = (1+tau_c(t))*(1+tau_x(t+1))/((1+tau_x(t))*(1+tau_c(t+1))); % tax scaling factor
        c(t+1) = c(t)*(A*(alpha*beta*k(t)^(alpha-1)/(1+tau_x(t+1))+beta*(1-delta)))^(1/sigma); %compute c(t+1)
        crit = max(abs(kstar-k(t+1)),abs(cstar-c(t+1)));     % deviation from steady state
        if crit<=tol
            % if close to steady state stop algorithm
            k = k(1:t+1); % cut path after convergences
            c = c(1:t+1); % cut path after convergences
            break
        else
            continue
        end
    end
    ite = ite + 1 % update iteration
end

k = [kstar_old, k]; % add k0
c = [cstar_old, c]; % add c0
```

```

% plot time path of k and c
figure(4)
subplot(2,1,1)
plot(0:length(k)-1,k, '--o');
xlabel('t') % x-axis label
ylabel('k') % y-axis label
subplot(2,1,2)
plot(0:length(c)-1,c, '--o');
xlabel('t') % x-axis label
ylabel('c') % y-axis label
print -depsc fig4.eps

%% Question 7
tau_c = [repmat(.03,1,10),repmat(.01,1,T-10)];    % technology
tau_x = [repmat(.05,1,10),repmat(.01,1,T-10)];    % technology

kstar = ((alpha*beta)/((1-beta*(1-delta))*(1+tau_x(T))))^(1/(1-alpha)); % Steady state
cstar = (kstar^alpha-delta*kstar*(1+tau_x(T)))/(1+tau_c(T));    % c steady state

k0 = kstar_old; % starting k value

k = zeros(1,T+1);    % initial k path vector
c = zeros(1,T+1);    % initial c path vector

lb_k = 0.9*kstar;    % lower bound of k axis
ub_k = 1.1*kstar;    % upper bound of k axis

lb_c = 0.9*cstar;    % lower bound of c axis
ub_c = 1.1*cstar;    % lower bound of c axis

kgrid = linspace(lb_k,ub_k,n)';    % k axis
cgrid = linspace(lb_c,ub_c,n)';    % c axis

crit = 1;    % initialize tolerance criteria
ite = 1;     % initialize iteration

while (crit>tol && ite<=N)
    k(1) = k0;  % set starting k0
    c(1) = cgrid(ite); % pick c0
    for t = 1:T-1
        k(t+1) = k(t)^alpha/(1+tau_x(t))+(1-delta)*k(t)-c(t)*(1+tau_c(t))/(1+tau_x(t)); % compute k(t+1)
```

```
        A = (1+tau_c(t))*(1+tau_x(t+1))/((1+tau_x(t))*(1+tau_c(t+1))); % tax scaling factor
        c(t+1) = c(t)*(A*(alpha*beta*k(t)^(alpha-1)/(1+tau_x(t+1))+beta*(1-delta)))^(1/sigma); %compute c(t+1)
        crit = max(abs(kstar-k(t+1)),abs(cstar-c(t+1)));     % deviation from steady state
        if crit<=tol
            % if close to steady state stop algorithm
            k = k(1:t+1); % cut path after convergences
            c = c(1:t+1); % cut path after convergences
            break
        else
            continue
        end
    end
    ite = ite + 1 % update iteration
end

k = [kstar_old, k]; % add k0
c = [cstar_old, c]; % add c0

% plot time path of k and c
figure(5)
subplot(2,1,1)
plot(0:length(k)-1,k, '--o');
xlabel('t') % x-axis label
ylabel('k') % y-axis label
subplot(2,1,2)
plot(0:length(c)-1,c, '--o');
xlabel('t') % x-axis label
ylabel('c') % y-axis label
print -depsc fig5.eps
```