Advanced Macro

Assignment 2

Hans Martinez

Sep 30, 2020

Competitive equilibrium

Arrow-Debreu competitive equilibrium consists of prices $\{p_t, w_t, r_t\}_{t=0}^{\infty}$, allocations for the firm $\{y_t, k_t^d, l_t^d\}_{t=0}^{\infty}$ and the allocations for household $\{c_t, k_t^s, l_t^s\}_{t=0}^{\infty}$ such that,

(i) Given a sequence of prices $\{p_t, w_t, r_t\}_{t=0}^{\infty}$, the firm allocation $\{y_t, k_t^d, l_t^d\}_{t=0}^{\infty}$ solves the firm problem,

$$\begin{aligned} &\max_{\{y_t,k_t,l_t\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} p_t(y_t - r_t k_t - w_t l_t) \\ &\text{s.t. } y_t = z k_t^{\alpha} l_t^{1-\alpha}, \forall t \geq 0; \\ &\quad y_t, k_t, l_t \geq 0, \forall t \geq 0. \end{aligned} \tag{1}$$

(ii) Given a sequence of prices $\{p_t, w_t, r_t\}_{t=0}^{\infty}$, the household allocation $\{c_t, k_t^s, l_t^s\}_{t=0}^{\infty}$ solves the household problem,

$$\begin{aligned} &\max_{\{c_t,k_{t+1},l_t\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t \left(\frac{c_t^{1-\sigma}}{1-\sigma} - \chi \frac{l_t^{1+\eta}}{1+\eta}\right) \\ &\text{s.t. } \sum_{t=0}^{\infty} p_t(c_t + K_{t+1} - (1-\delta)k_t) \leq \sum_{t=0}^{\infty} p_t(r_t k_t + w_t l_t); \\ &\quad 0 \leq l_t \leq 1, 0 \leq k_t \leq k_0, c_t \geq 0, k_{t+1} \geq 0, \forall t \geq 0; \\ &\quad k_0 \text{ given.} \end{aligned} \tag{2}$$

(iii) The market clear conditions,

$$\begin{aligned} z k_t^{\alpha} l_t^{1-\alpha} + (1-\delta)k_t &= c_t + k_{t+1}; \\ l_t^d &= l_t^s; \\ k_t^d &= k_t^s. \end{aligned}$$

Steady state

For firm problem,

$$\begin{aligned} r_t =& z\alpha k_t^{\alpha-1} l_t^{1-\alpha} \\ w_t =& z(1-\alpha) k_t^{\alpha} l_t^{-\alpha} \\ r_t k_t + w_t l_t =& z k_t^{\alpha} l_t^{1-\alpha} \end{aligned} \tag{3}$$

Then for household problem,

$$\begin{aligned}
\mathcal{L}(\{c_t, k_{t+1}, l_t\}_{t=0}^{\infty}; \lambda_t) = \sum_{t=0}^{\infty} \beta^t (\frac{c_t^{1-\sigma}}{1-\sigma} - \chi \frac{l_t^{1+\eta}}{1+\eta}) + \lambda_t (\sum_{t=0}^{\infty} p_t (z k_t^{\alpha} l_t^{1-\alpha} - c_t - k_{t+1} + (1-\delta) k_t) \\
\frac{\partial \mathcal{L}}{\partial c_t} =& \beta^t c_t^{-\sigma} - \lambda_t p_t = 0, \\
\frac{\partial \mathcal{L}}{\partial l_t} =& -\beta^t \chi l_t^{\eta} + \lambda_t p_t z (1-\alpha) k_t^{\alpha} l_t^{-\alpha} = 0, \\
\frac{\partial \mathcal{L}}{\partial k_t} =& \lambda_t p_t (z \alpha k_t^{\alpha-1} l_t^{1-\alpha} + 1 - \delta) - \lambda_{t-1} p_{t-1} = 0, \\
c_t =& z k_t^{\alpha} l_t^{1-\alpha} - k_{t+1} + (1-\delta) k_t
\end{aligned} \tag{4}$$

For Steady state, $p_0 = 1$, then,

$$\begin{aligned}
& c^{\sigma} l^{\eta} = z(1-\alpha) k^{\alpha} l^{-\alpha} / \chi \\
& z \alpha k^{\alpha-1} l^{1-\alpha} = 1/\beta - 1 + \delta \\
& c = z k^{\alpha} l^{1-\alpha} - \delta k
\end{aligned} \tag{5}$$

$$M = \frac{k}{l} = \left(\frac{z \alpha \beta}{1 - \beta + \beta \delta} \right)^{\frac{1}{1-\alpha}},$$

$$N = \frac{c}{l} = z (\frac{k}{l})^{\alpha} - \delta \frac{k}{l} = z M^{\alpha} - \delta M,$$

$$\begin{aligned}
(lN)^{\sigma} l^{\eta} =& z(1-\alpha) (\frac{k}{l})^{\alpha} / \chi \Rightarrow \\
l^{\sigma+\eta} =& \frac{z(1-\alpha) M^{\alpha}}{\chi N^{\sigma}} \Rightarrow \\
l_{ss} =& \left(\frac{z(1-\alpha) M^{\alpha}}{\chi N^{\sigma}} \right)^{\frac{1}{\sigma+\eta}},
\end{aligned}$$

Then,

$$\begin{aligned}
k_{ss} =& Ml \\
c_{ss} =& Nl \\
y_{ss} =& z k^{\alpha} l^{1-\alpha} = z M^{\alpha} l \\
r_{ss} =& \alpha z k^{\alpha-1} l^{1-\alpha} = \alpha z M^{\alpha-1} \\
w_{ss} =& (1-\alpha) z k^{\alpha} l^{\alpha} = (1-\alpha) z M^{\alpha}.
\end{aligned}$$

Social planner problem

The problem of the social planner is that, given the initial capital k_0,

$$\begin{aligned}
w(k_0) = \max_{\{c_t, k_t, l_t\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t (\frac{c_t^{1-\sigma}}{1-\sigma} - \chi \frac{l_t^{1+\eta}}{1+\eta}) \\
s.t. \;\; z k_t^{\alpha} l_t^{1-\alpha} = c_t + k_{t+1} - (1-\delta) k_t, \;\; \forall t \geq 0 \\
c_t \geq 0, \; k_t \geq 0, \; 0 \leq l_t \leq 1, \;\; \forall t \geq 0 \\
k_0 \text{ is given.}
\end{aligned} \tag{6}$$

Bellman equation,

$$V(k) = \max_{\substack{0 \le l \le 1 \\ 0 \le k' \le zk^{\alpha}l^{1-\alpha} + (1-\delta)k}} \{ \frac{(zk^{\alpha}l^{1-\alpha} + (1-\delta)k - k')^{1-\sigma}}{1-\sigma} - \chi \frac{l^{1+\eta}}{1+\eta} + \beta \mathbb{E} V(k') \} \tag{7}$$

Obtain the path of {c,k,r,w,y} in the following cases

Let $\alpha = \frac{1}{3}, z = 1$. Use planner's solution, starting from steady state:

A. Capital decreases to 80% of its steady state.

```
# Julia code
# setting working directory
cd("/Volumes/SSD Hans/Github/MacroFall2020/AdvMacro")

#Defining vars
a = 1/3
b = 0.6
k_star = (a*b)^(1/(1-a))
r = 1/b

# Initializing array for loop
m, n = 100,2
K = fill(0.0, (m, n))

#defining policy fn
g = x -> a*b*(x^a)

# loop for capital
k = 0.8*k_star
for i in 1:m
    K[i, 1] = k
    K[i, 2] = g(K[i,1])
    global k = K[i, 2]
end

# defining fn's for the rest of the variables
w = x -> (1-a)x^a
y = x -> x^a
c = (p,q) -> y(p)-q
r = x -> a*(x^(a-1))
int = r.(K[1:100,2])
wage = w.(K[1:100,1])
output = y.(K[1:100,1])
cons = c.(K[1:100,1],K[1:100,2])

# importing packages for plotting
import Pkg; Pkg.add("Plots")
using Plots
gr()

# plotting capital
plot(1:10,fill(k_star,10), label="capital SS")
```

```
plot!(1:10,K[1:10,1], label="capital path", linestyle = :dot, legend = :bottomright)
xlabel!("t")
title!("Capital")
savefig("graphs/capital")

# plotting output

plot(1:10,fill(y(k_star),10), label="output SS")
plot!(1:10,output[1:10], label="output path", linestyle = :dot, legend = :bottomright)
xlabel!("t")
title!("Output")
savefig("graphs/output")

# plotting consumption

plot(1:10,fill(c(k_star,k_star),10), label="consumption SS")
plot!(1:10,cons[1:10], label="consumption path", linestyle = :dot, legend = :bottomright)
xlabel!("t")
title!("Consumption")
savefig("graphs/consumption")

# plotting wage

plot(1:10,fill(w(k_star),10), label="wage SS")
plot!(1:10,wage[1:10], label="wage path", linestyle = :dot, legend = :bottomright)
xlabel!("t")
title!("Wage")
savefig("graphs/wage")

# plotting capital rental rate
plot(1:10,fill(r(k_star),10), label="capital rental rate SS")
plot!(1:10,int[1:10], label="rate path", linestyle = :dot, legend = :topright)
xlabel!("t")
title!("Capital rental rate")
savefig("graphs/capitalr")
```

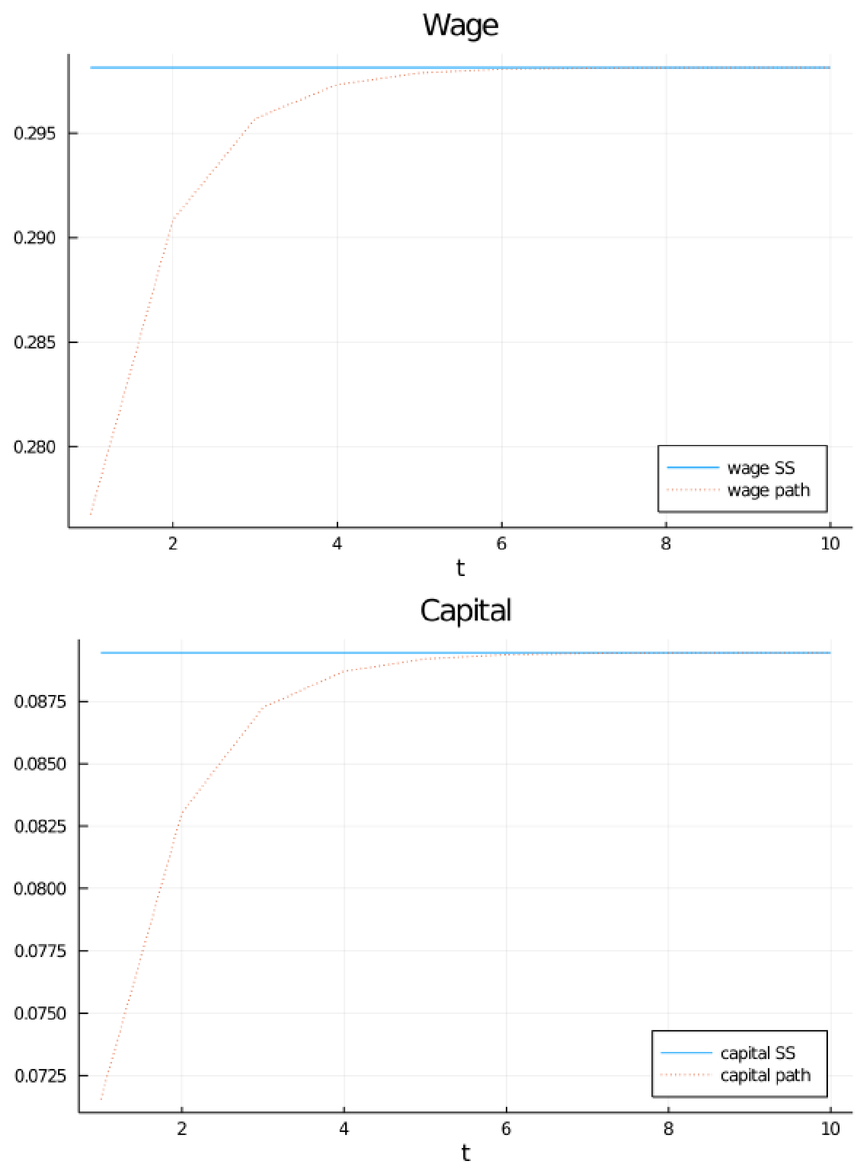

Figure 1: Wage and capital

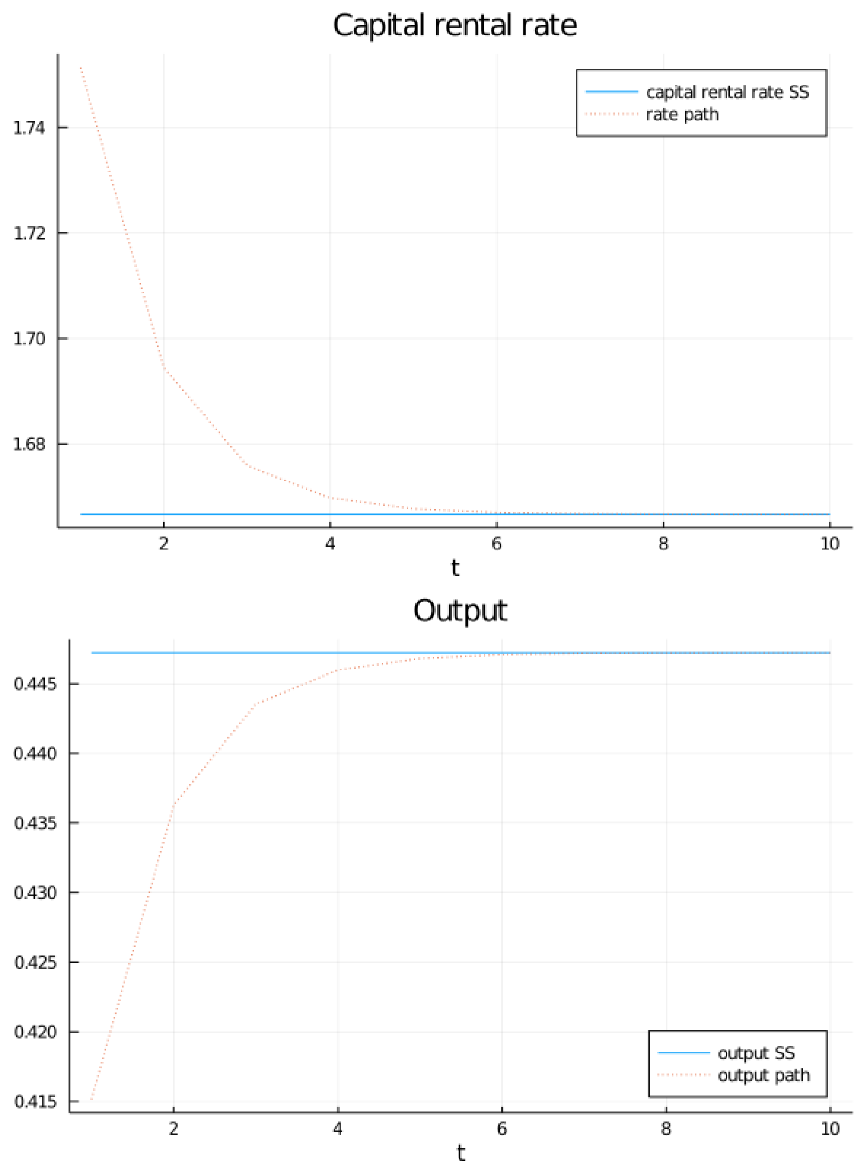

Figure 2: Capital rate and output

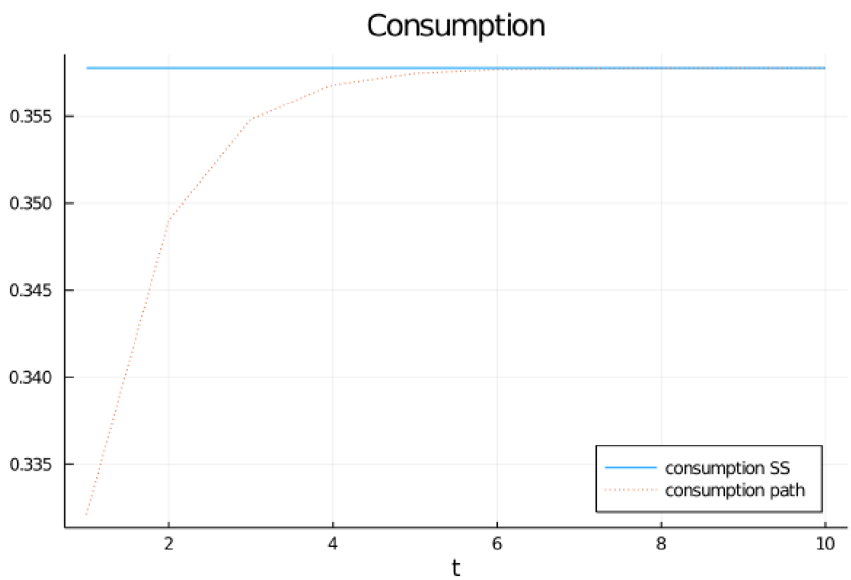

Figure 3: Consumption

B. Productivity increases permanently by 5%

```
# Julia code
# Increase in productivity permanently by 5%

k_starp = (a*b*1.5)^(1/(1-a))
#defining policy fn with productivity increases
g1 = x -> a*b*1.05*(x^a)

# Initializing array for loop
K1 = fill(0.0, (m, n))

# loop for capital
k1 = k_starp
for i in 1:m
    K1[i, 1] = k1
    K1[i, 2] = g(K[i,1])
    global k1 = K[i, 2]
end

# defining fn's for the rest of the variables
w1 = x -> (1-a)*(x^a) #does not change
y1 = x -> 1.5*(x^a)
c1 = (p,q) -> y(p)-q #does not change
r1 = x -> a*(1.5*x^(a-1))
int_p = r1.(K1[1:100,2])
wage_p = w1.(K1[1:100,1])
output_p = y1.(K1[1:100,1])
cons_p = c1.(K1[1:100,1],K1[1:100,2])


# plotting capital
```

```julia
plot(1:10,fill(k_star,10), label="capital SS")
plot!(1:10,K1[1:10,1], label="capital path", linestyle = :dot, legend = :bottomright)
xlabel!("t")
title!("Capital (productivity increase)")
savefig("graphs/capitalp")

# plotting output

plot(1:10,fill(y1(k_star),10), label="output SS")
plot!(1:10,output_p[1:10], label="output path", linestyle = :dot, legend = :bottomright)
xlabel!("t")
title!("Output (productivity increase)")
savefig("graphs/outputp")

# plotting consumption

plot(1:10,fill(c1(k_star,k_star),10), label="consumption SS")
plot!(1:10,cons_p[1:10], label="consumption path", linestyle = :dot, legend = :bottomright)
xlabel!("t")
title!("Consumption (productivity increase)")
savefig("graphs/consumptionp")

# plotting wage

plot(1:10,fill(w1(k_star),10), label="wage SS")
plot!(1:10,wage_p[1:10], label="wage path", linestyle = :dot, legend = :bottomright)
xlabel!("t")
title!("Wage (productivity increase)")
savefig("graphs/wagep")

# plotting capital rental rate
plot(1:10,fill(r1(k_star),10), label="capital rental rate SS")
plot!(1:10,int_p[1:10], label="rate path", linestyle = :dot, legend = :topright)
xlabel!("t")
title!("Capital rental rate (productivity increase)")
savefig("graphs/capitalrp")
```

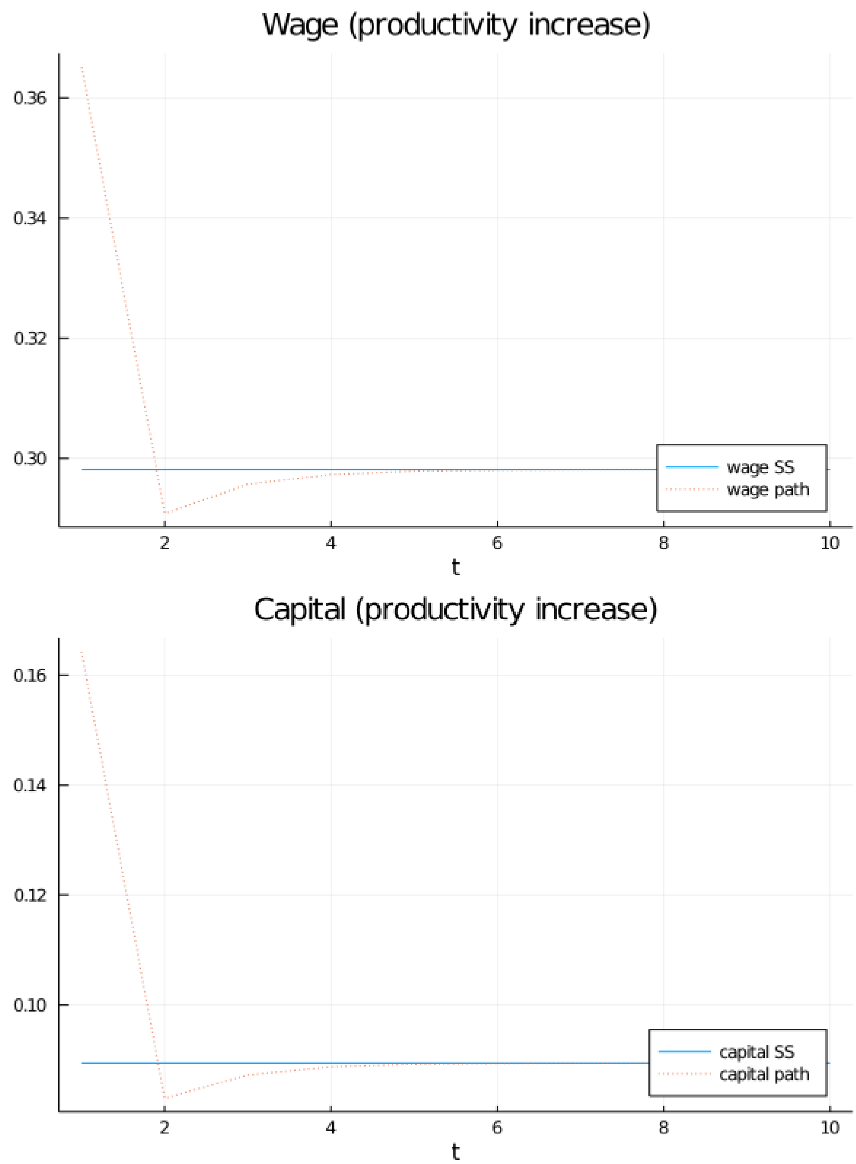

Figure 4: Wage and capital (productiviy increase)

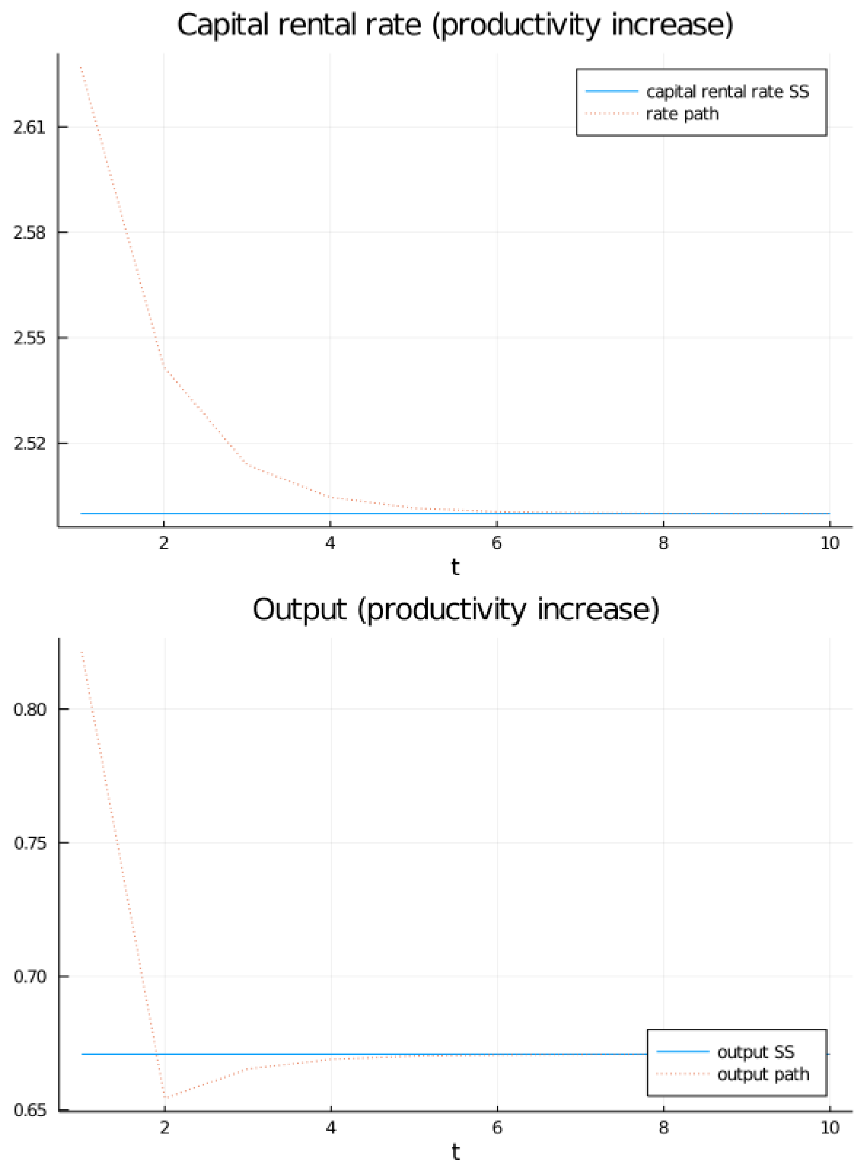

Figure 5: Capital rate and output (productivity increase)

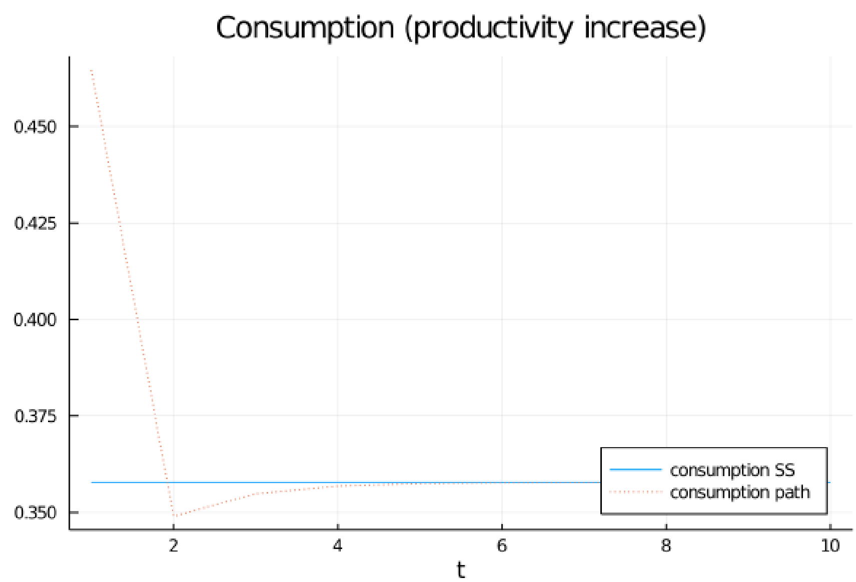

Figure 6: Consumption (productivity increase)