Homework 2

Marcos Cardozo

February 10, 2017

1. Please see code 'main2.m'.
2. From the FOCs we get:

$$\begin{aligned} c_t^{-\sigma} &= \beta E c_{t+1}^{-\sigma}[\alpha z_t {k_t}^{\alpha-1} + (1-\delta)] \\ c_t + k_{t+1} &= z_t {k_t}^{\alpha} + (1-\delta)k_t \\ \log z_{t+1} &= \rho z_t + \epsilon_t \end{aligned}$$

I define $\hat{c_t} = \frac{c_t - c^*}{c^*} \approx \log c_t - \log c^*$, $\hat{k_t} = \frac{k_t - k^*}{k^*}$ and $\hat{z}_t = \frac{z_t - z^*}{z^*}$.

The most difficult equation to log-linearize is the Euler Equation, for the others I omit the calculations. For the procedure, I first take logs at both sides of the equation, and then use a first order Taylor expansion.

$$-\sigma \log c_t = \log\beta + E[-\sigma \log c_{t+1} + \log \alpha z_t {k_t}^{\alpha-1} + (1-\delta)]$$

$$-\sigma(\log c_t - \log c^*) = \log\beta + \log 1/\beta + E[-\sigma \hat{c}_{t+1} + \log \alpha z_t {k_t}^{\alpha-1} + (1-\delta) + \frac{1/\beta - (1-\delta)}{1/\beta}\hat{z}_t + \frac{(\alpha-1)[1/\beta - (1-\delta)]}{1/\beta}\hat{k_t}]$$

$$-\sigma \hat{c}_{t+1} = E[\sigma \hat{c_t} + (1-(1-\delta)\beta)\hat{z_t} + (\alpha-1)[1-(1-\delta)\beta]\hat{k_t}]$$

And we also have, log-linearized using the same procedure,

$$k^*\hat{k}_{t+1} = {k^*}^{\alpha} - c^* c_t + (1/\beta)k^*\hat{k_t}$$

$$\hat{z}_{t+1} = \rho\hat{z_t} + \epsilon_t$$

Assuming $\hat{z^*} = 1$, recalling $y^* = \alpha {k^*}^{\alpha-1}$, $k^* = [\frac{(1-(1-\delta)\beta)}{\alpha\beta}]^{1/(\alpha-1)}$, $c^* = {k^*}^{\alpha} - \delta k^*$ we can write the system in matrix form

$$\begin{bmatrix} 1-(1-\delta)\beta & (\alpha+1)(1-(1-\delta)\beta) & -\sigma \\ 0 & k^* & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{z}_{t+1} \\ \hat{k}_{t+1} \\ \hat{c}_{t+1} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -\sigma \\ y^* & \alpha y^* + (1-\delta)k^* & -c^* \\ \rho & 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{z}_t \\ \hat{k}_t \\ \hat{c}_t \end{bmatrix}$$

I then use the Jordan decomposition technique to solve it noting that the system complies with the Blanchard-Khan condition. For more details please see the code 'main2.m'.

3. Please see code 'linearpf.m'.
4. (a) Idem.

(b) Idem.

Figure 1: Consumption of Value Function Iteration

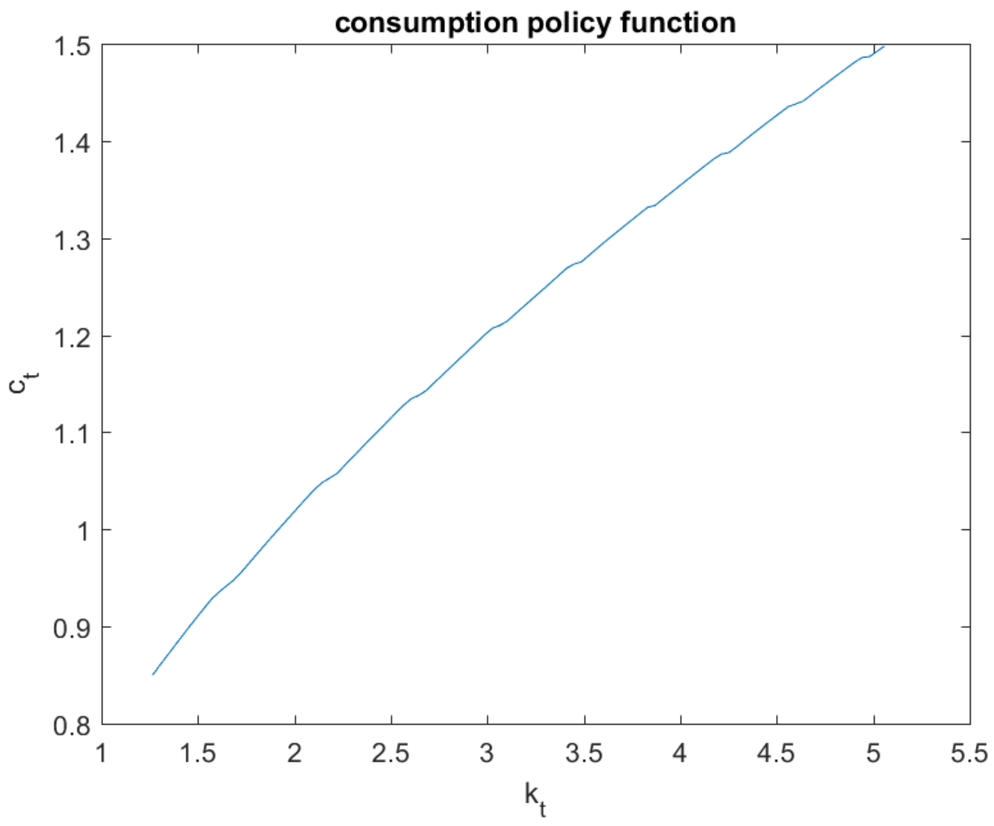


Figure 2: Capital of Value Function Iteration

Figure 3: Value Function

(c)

(d) If I increase N_k, that is, the number of grid points, the algorithm takes much longer to converge. For example, increasing from $N_k = 99$ to 299, made the algorithm take 3 times longer to converge. Increasing κ from 0.3 to 0.8 didn't make a difference regarding resolution speed, however, I noticed the value function and the policy function less concave.

(e) This figures were made with $\kappa = 0.6$ and $N_k = 99$.

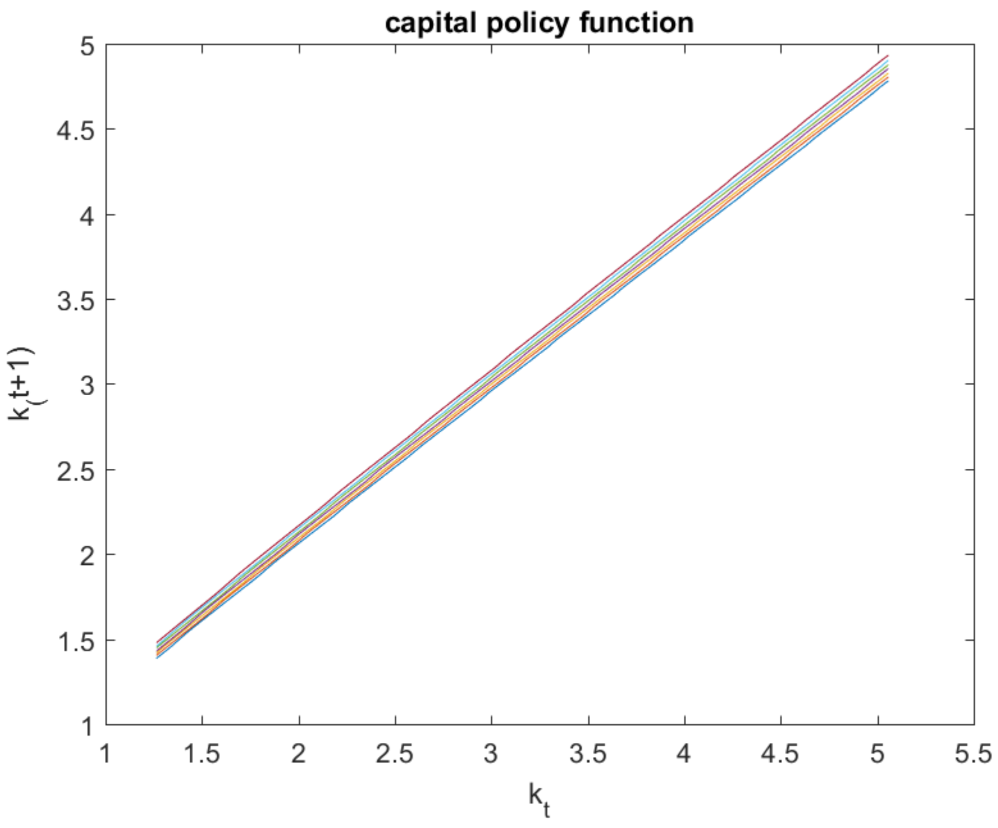
Figure 4: Comparison of consumption

Figure 5: Comparison of capital

(f) This figures were made with $\sigma = 10$. There seems to be an error, since I can see a jump in capital and consumption. I expected the results of the functions to be more concave and thus differ more from the log-linearized solution.

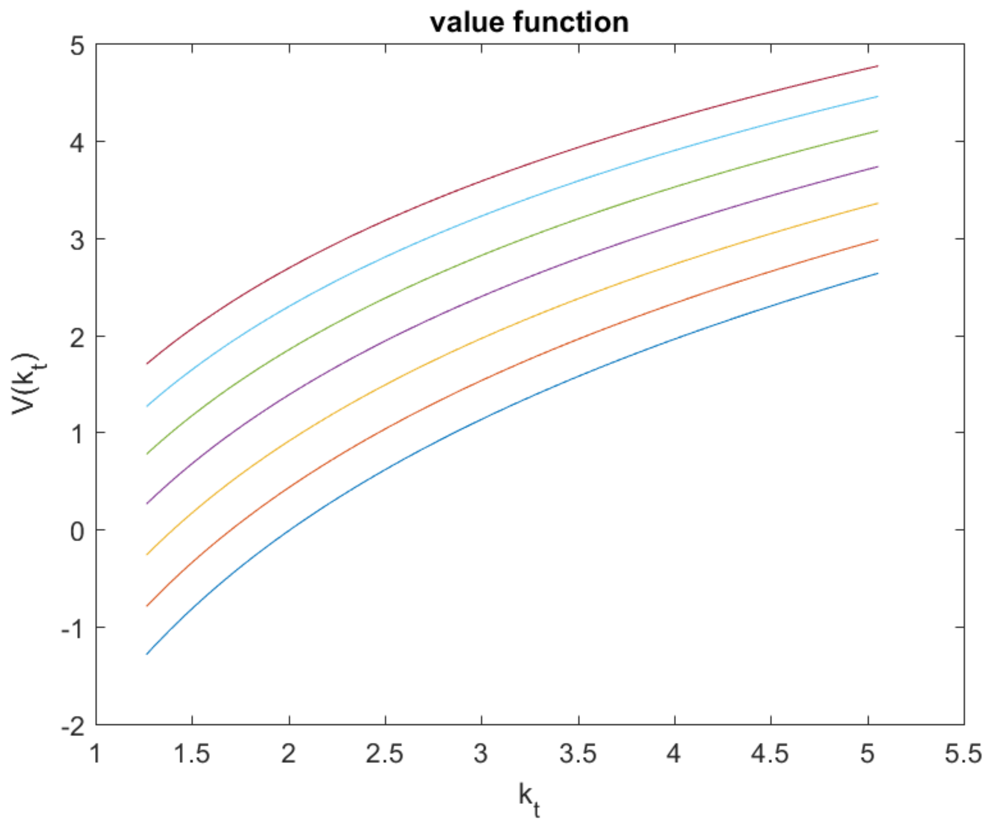
Figure 6: Comparison of consumption sigma=10

Figure 7: Comparison of capital sigma=10

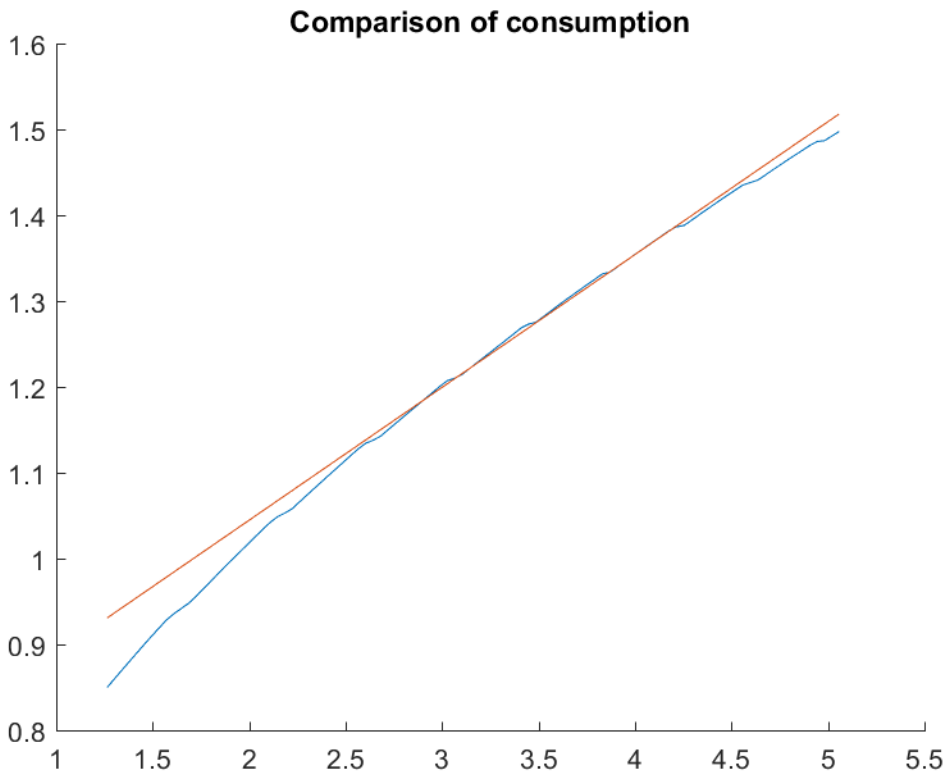


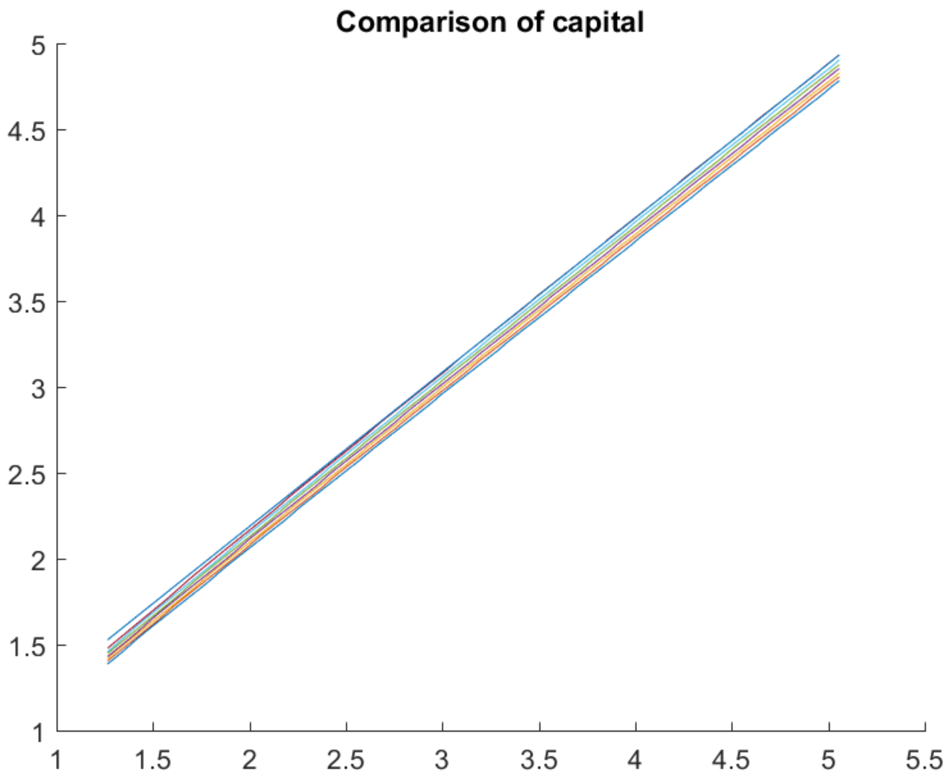
5. I tried to compute the analytical value function, but probably I got the wrong solution, since my analytical value function has a lower value for each capital in the grid.

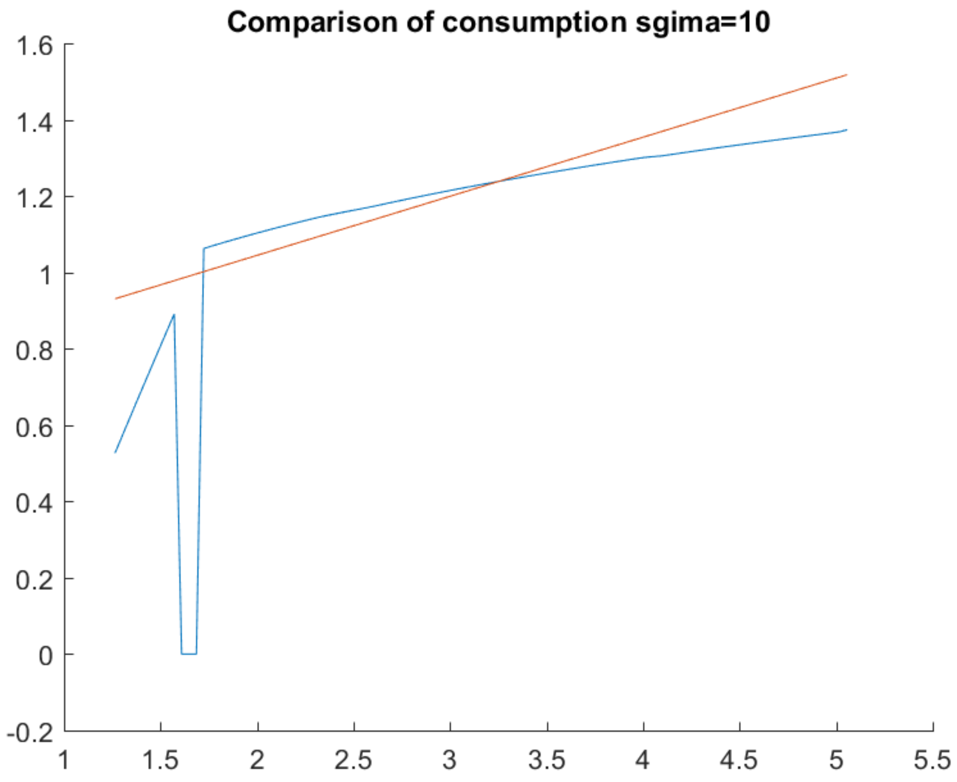
Figure 8: Comparison of value functions

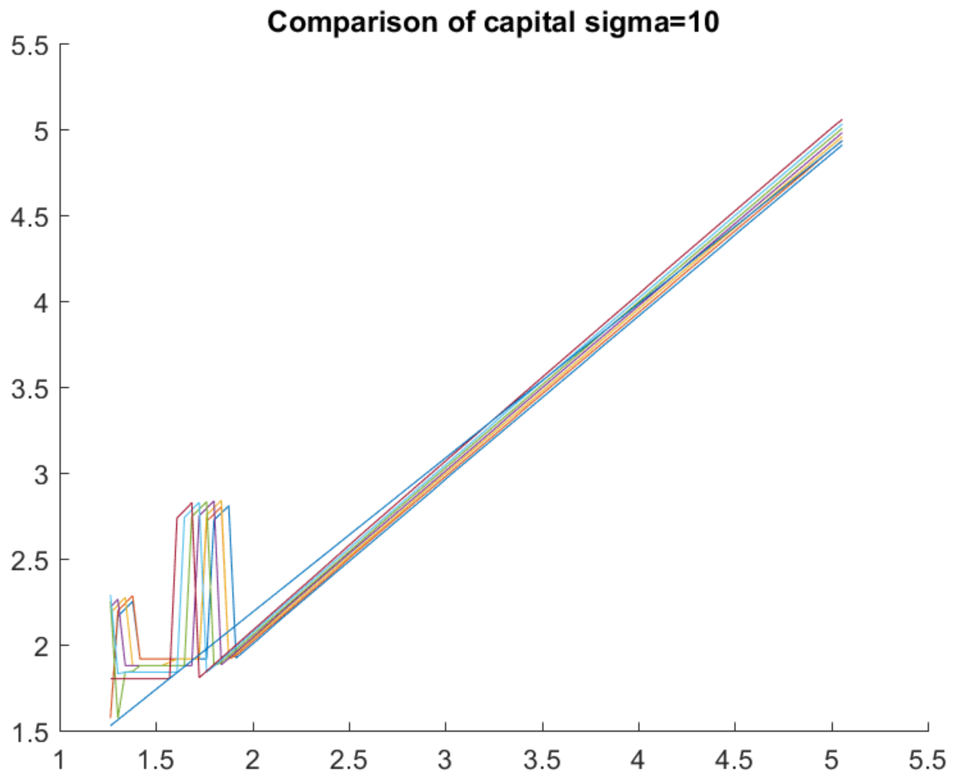


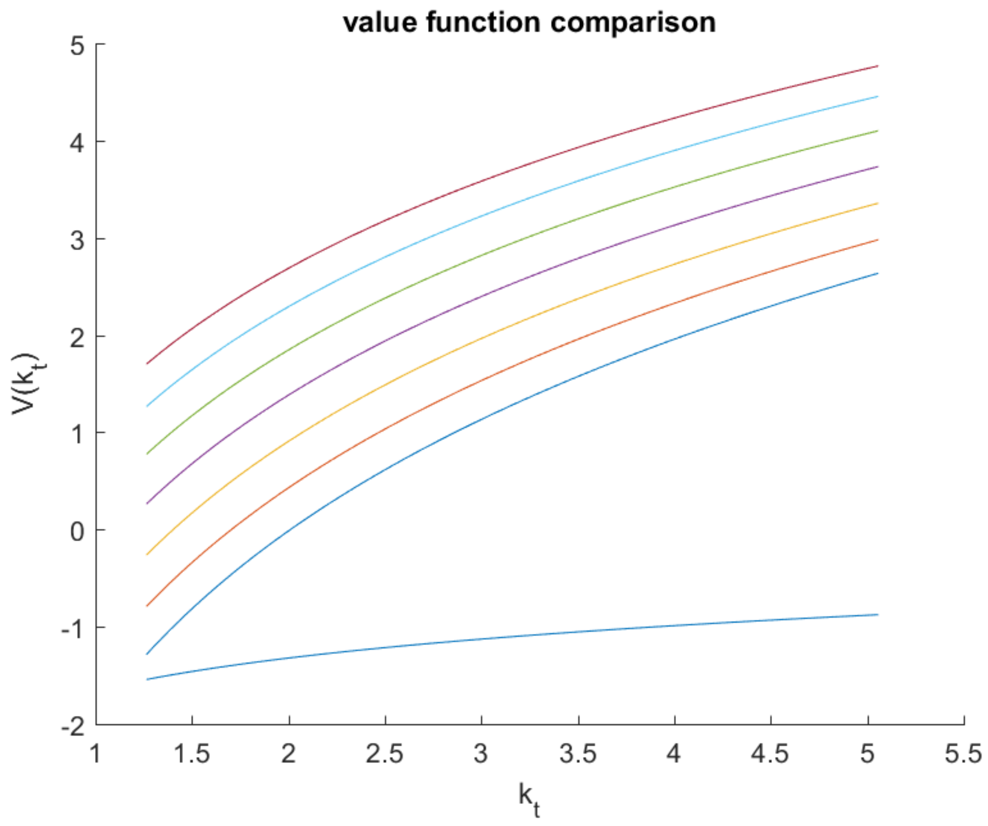
value function
V(k_t)
k_t
5
4
3
2
1
0
-1
-2
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5


Comparison of consumption
1.6
1.5
1.4
1.3
1.2
1.1
1
0.9
0.8
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5


Comparison of capital
5
4.5
4
3.5
3
2.5
2
1.5
1
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5


Comparison of consumption sgima=10
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5


Comparison of capital sigma=10
5.5
5
4.5
4
3.5
3
2.5
2
1.5
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5



value function comparison
V(k_t)
k_t
5
4
3
2
1
0
-1
-2
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5