Quantitative Macroeconomics – Homework 4

Itsaso Apezteguia Extramiana

Value Function Iteration (inelastic labor supply)

<u>The model:</u>

- Stationary economy
- Large number of identical infinitely lived households, that maximize:

$$E_0 \left\{ \sum_{t=0}^{\infty} \beta^t u(c, h_t) \right\},$$

over consumption and leisure $u(c_t, 1-h_t) = \ln c_t - \kappa \frac{h_t^{1+\frac{1}{\nu}}}{1+\frac{1}{\nu}}$, subject to:

$$c_t + i_t = y_t$$

$$y_t = k_t^{1-\theta} \ (h_t)^{\theta}$$

$$i_t = k_{t+1} - (1-\delta)\ k_t$$

- Parameter values are: θ=0.679, β=0.988 ,δ=0.013, ĸ =5.24 and ν= 2
- At the beginning h_t=1 (labor is inelastically supplied)
- To compute the steady-state we normalize output to 1.

Value Function Iteration (inelastic labor supply)

In order to solve this problem we write the recursive formulation of the sequential problem as follows:

V(k) = max u(c) + β V(k´) $\longrightarrow$ V(k) = max u(f(k) + (1-δ)k-k´)+ β V(k´)

Our initial guess states that the value function is equal to zero: V(k) = 0

We also need to discretize the state space, that is, we define a grid for the continuous variable k in order to discretize it. We set the size of the grid to be equal to 200. k_{min} is set equal to 1, in oder to avoid potential problems of having zero output and therefore zero consumption. We set k_{max} slightly above the steady state level of capital, since we are interested in dynamics above the steady state. For that we also need to compute the steady state level of capital.

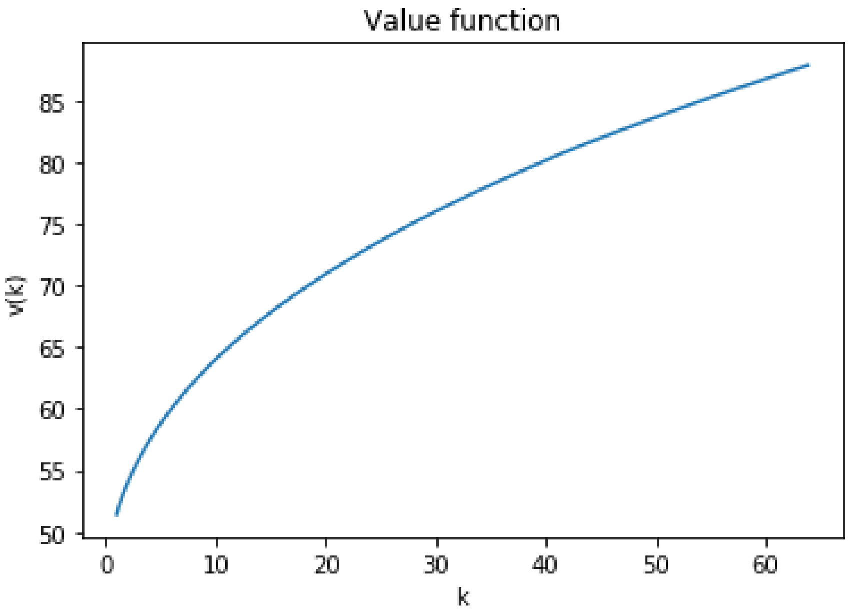
After doing this, we define the return matrix M, which collects the utility of all posible combinations of today and tomorrow capital. In this step we need to make sure that we do not obtain any solution where consuption is zero. For that, we add a constraint.

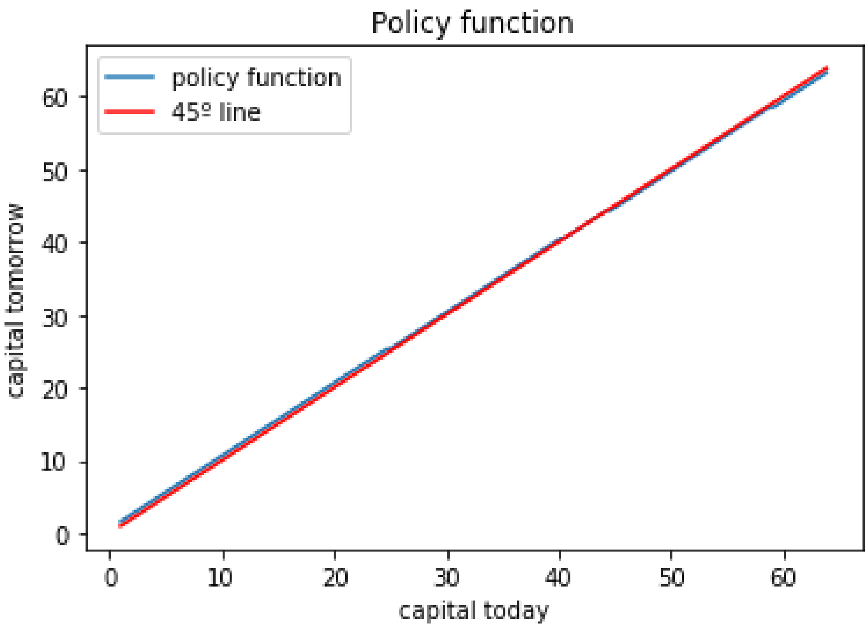
Once we have the return matrix we define a method in order to deliver the higher value function of all posible combinations, taking as input the value of next period. This method also delivers the position in the matrix of this máximum. The method defines a matrix X which collects all values for V, and selects for each k the máximum of them.

Finally, we define a method which takes as input the previous function and iterates until the difference between the obtained last 2 values of V is less than a choosen error, meaning that we have reached the steady state. This method delivers the vector solution V and the policy function for capital.

First we solve the problem with brute force iterations of the value function. After this we try to speed up the algorithm making use of the monotonicity of the optimal decision rule, the concavity of the value function, of monotonicity and concavity together, by local search and by Howard´s policy.

In the following graphs we can see the dynamics of the value function and the policy function.





The table below summarizes the number of iterations and the time (in seconds) the value function needs to converge with different VFI algorithms.

Method	*Time (seconds)*	*Number of iterations*
Brute Force	7,936	238
Monotonicity	5,212	238
Concavity	7,190	238
Local Search	1,096	238
Monotonicity & Concavity	16,42	238

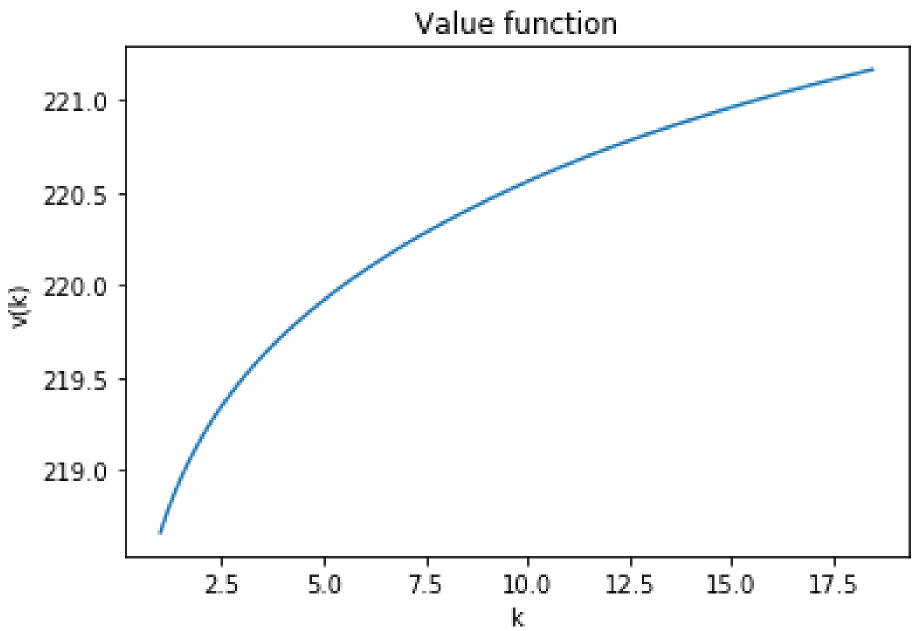
As we can see, the most efficient way to iterate the value function is to use local search into the VFI algorithm. The time needeed to obtain convergence is much lower than with the other methods. However, the number of iterations needeed is always the same, no matter which method we use.

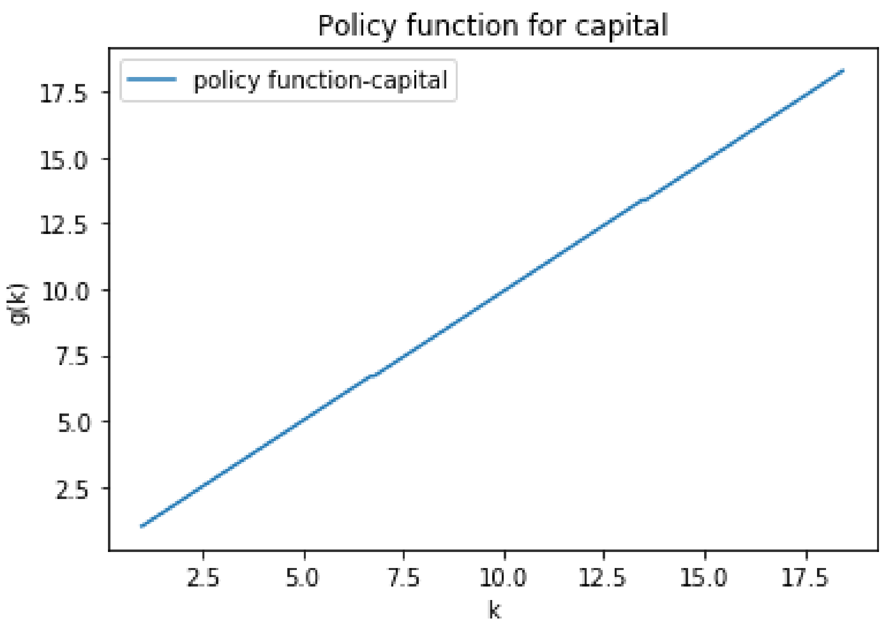
When we use monotonicity and concavity together we expect the time to be lower. However in this case we need more time than with other methos, maybe due to an error in the code.

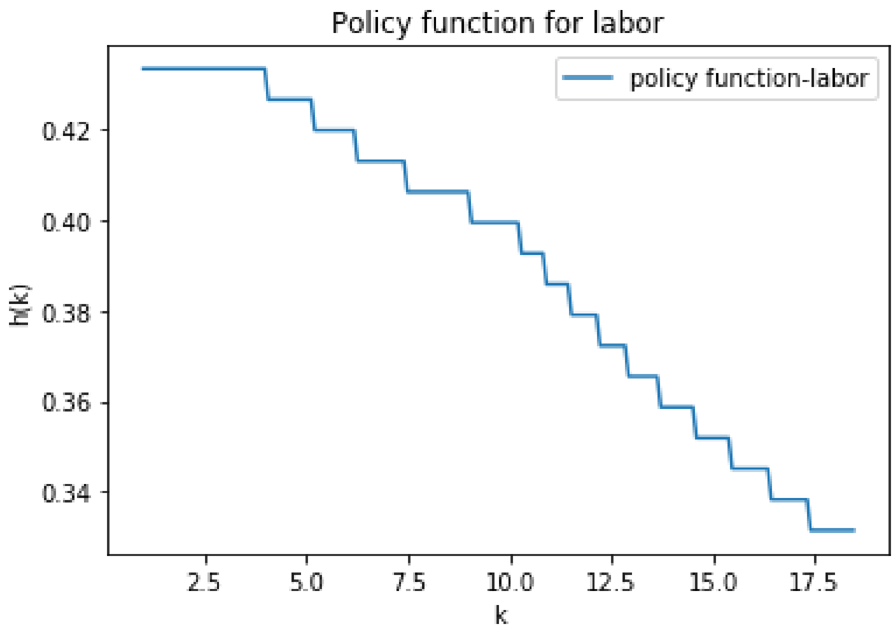
Value Function Iteration (continuous labor supply)

Now agents must choose also the quantity of labor they are going to supply to the market. Therefore, now we need to discretize capital and labor. We choose the grid for k in the same way as before. For h we choose h_{min} = 0.1, h_{max} = 0.434 and the size of the grid equal to 50.

In the following graphs we can see the dynamics of the value function, and the policy functions por capital and labor.







Capital in the next period have positive relation with capital today, whereas labor is decreasing in capital.

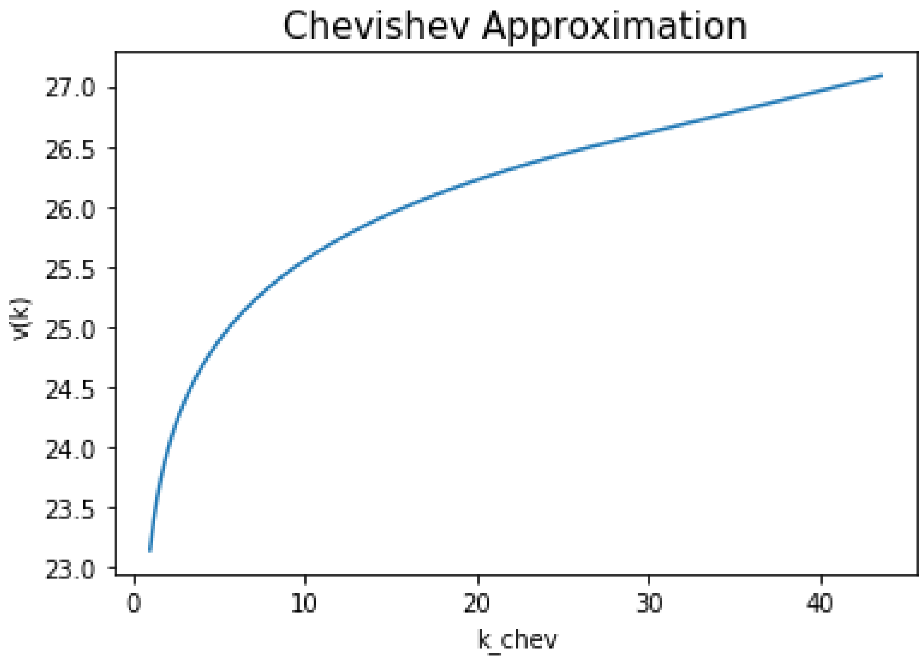
The convergence of the value function takes 654 iterations, in 01:35.20. Therefore, we can conclude that when we add a labor choice that is continuous to the problem, computing the solution by the VFI algorithm becomes harder.

Chebyshev regression algorithm to approximate the value function

Now we use Chebyshec regression algorithm in order to approximate the value function. For this we define methods to commpute the chebyshec roots, chebyshev polynoamials and its coefficients. Once we obtain this, we define a new method in order to compute V taking into account the Chebyshev approximations of the value function.

The time needed to reach the steady state is too much high, so we change the tolerance level to a larger one.

In the following graph we can see the value function:



The time of execution is 64,446 seconds. The number of iterations needed to reach the steady state is 10, therefore with this method we need much less iterations to reach the steady state equilibrium than with other methods used above.