

Economics 714 - Computational Methods in Economics

University of Pennsylvania

Fall 2021

Problem Set 2

Javier Tasso

Click on [this link](#) to access the repository.

1 Steady State

With $z = 0$ the resource constraint of the economy reads as follows.

$$\begin{aligned} c + i &= k^\alpha l^{1-\alpha} \\ c + k' - (1 - \delta)k &= k^\alpha l^{1-\alpha} \\ c &= k^\alpha l^{1-\alpha} - k' + (1 - \delta)k \end{aligned} \tag{1}$$

Given the state k , the social planner chooses l and k' (and c) to solve the following problem.

$$V(k) = \max_{k', l} \left\{ (1 - \beta) \left(\ln[k^\alpha l^{1-\alpha} - k' + (1 - \delta)k] - \frac{l^2}{2} \right) + \beta V(k') \right\}$$

Take first order condition with respect to labor to get the following intratemporal equation.

$$\frac{1 - \alpha}{c} \cdot \left(\frac{k}{l} \right)^\alpha = l \tag{2}$$

Take first order condition with respect to capital in next period and use the envelope theorem to get the following intertemporal equation.

$$\frac{1}{c} = \frac{\beta}{c'} \cdot \left[\alpha \left(\frac{l'}{k'} \right)^{1-\alpha} + 1 - \delta \right] \tag{3}$$

Set $c = c' = c_{SS}$, $l = l' = l_{SS}$ and $k = k' = k_{SS}$ and use equations (1), (2) and (3) to find the steady state values. For $\alpha = 0.33$, $\beta = 0.97$ and $\delta = 0.1$. Table (1) shows the values in the steady state.

Table 1: Steady State Values

Variable	Value
Consumption	1.1161
Capital	3.7612
Labor	0.9465

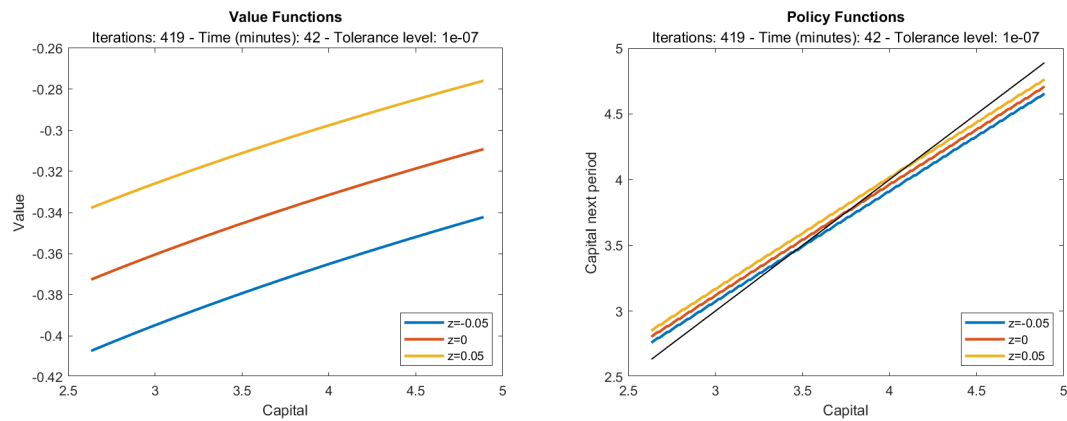


Figure 1: Value and Policy Functions - Fixed Grid

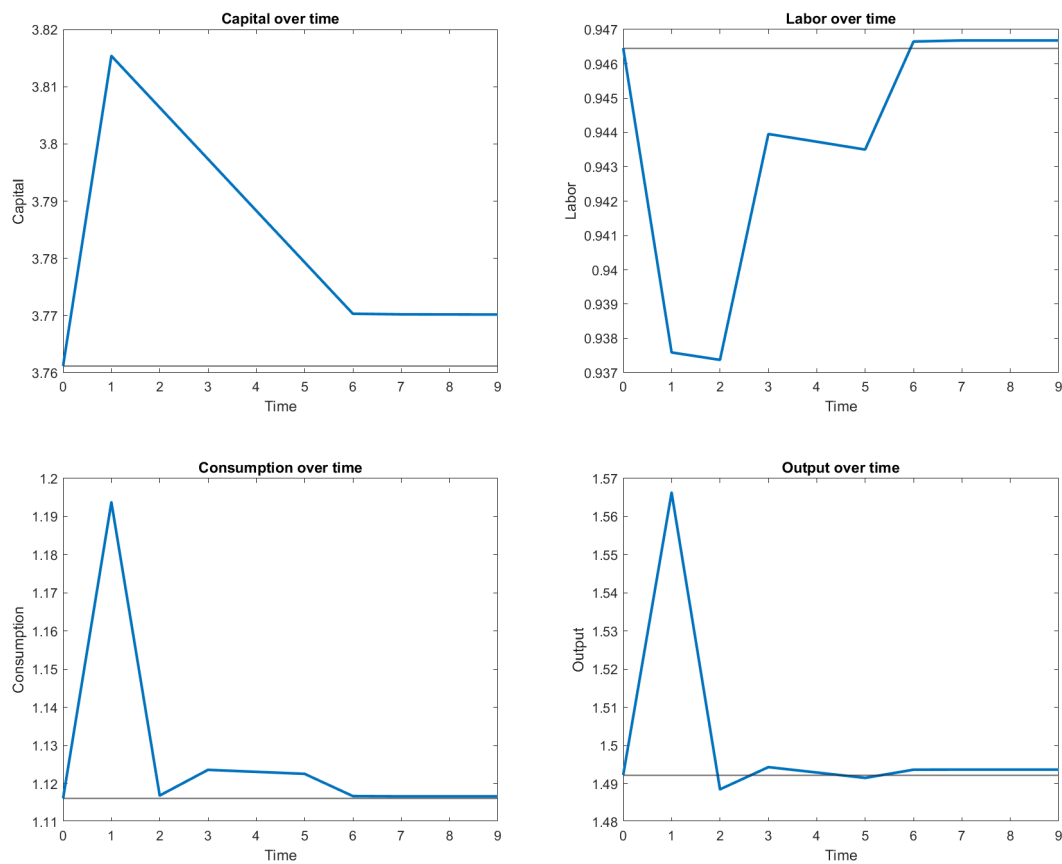


Figure 2: Responses to a Technology Shock

2 Value Function Iteration with a Fixed Grid

Figure (1) plots the value and policy functions and it includes information about the number of iterations, time elapsed and tolerance level. Figure (2) plots trajectories after the following shock: initially we are in the steady state when $z = 0$, suddenly there is a positive shock, that is, $z_1 = 0.05$ and after that shock $z_t = 0$ forever for $t > 1$.

3 Accelerator

It does not work. I tried different things. I think I understand how the procedure is, so let me try to describe it briefly. Script *Econ714_hw2_1_VFI.m* has one section where I tried to implement the accelerator.

- First the algorithm starts as a simple value iteration during the first 50 iterations. This is not necessary but maybe this is good to get a better guess and to start accelerating when you are close enough.
- After that, every 10 iterations do exactly as before.
- On the other 9 iterations implement something similar to equation (3) of Heer, B., & Maussner, A. (2008) that we describe as follows.

$$\mathbf{v} = \mathbf{u} + \beta Q \bar{\mathbf{v}} \quad (4)$$

- Vector \mathbf{v} is a vector of one column that stacks together all the values of the grid for the three levels of productivity. With a grid of length 100 this vector has a length of 300. Vector \mathbf{u} does the same thing but for the period utility level given (k, z) . Matrix Q has zeros everywhere except on those points where the grid of capital is equal to the value of the policy function. Finally vector $\bar{\mathbf{v}}$ is a linear combination of \mathbf{v} of the same size. What this vector does is that it calculates the expected value function using the transition probabilities.
- What we want to do in these steps is to solve (4) for \mathbf{v} and use that as our value function in those iteration. This should speed up things relative to the previous question.

4 Multigrid

In the last grid the initial guess that came from the previous grid was so good that it only took one iteration. Figure (3) plots the results using multi-grid. It took three hours to run. Figure (4) compares the value function using the multi-grid to the one we got in question 2. They seem to be pretty similar. The multi-grid method seems to work very well: it would have been too costly to run simple value function iteration on a grid that long.

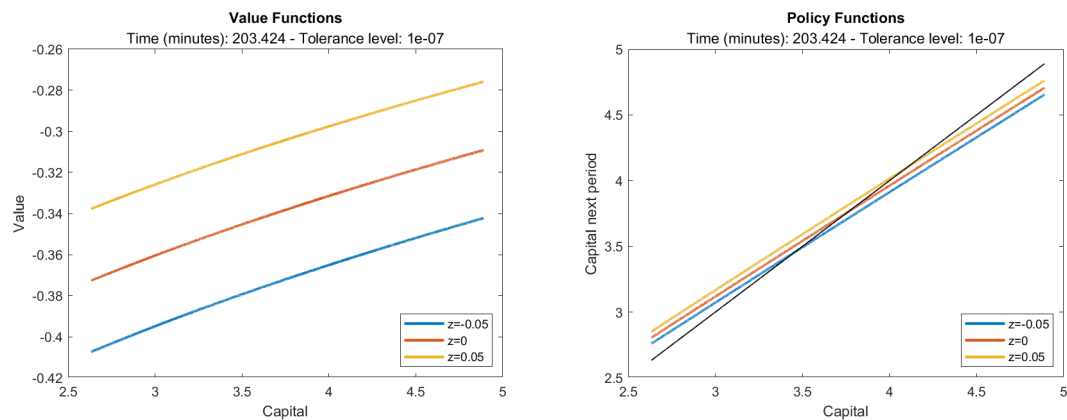


Figure 3: Value and Policy Functions - Multigrid

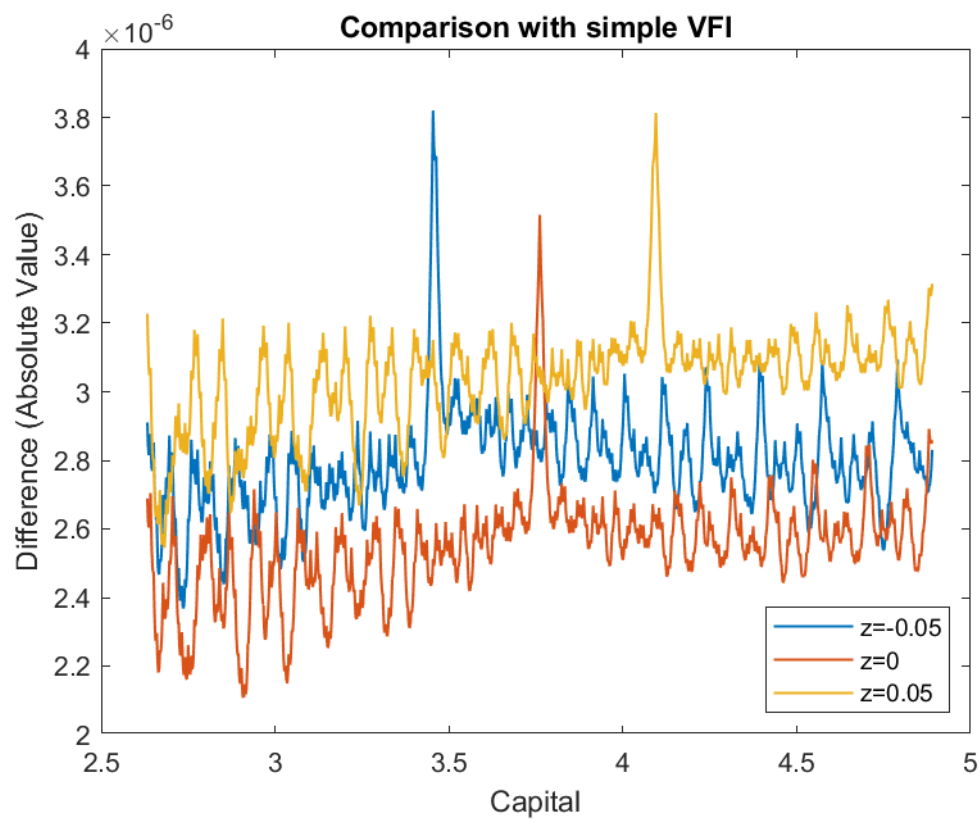


Figure 4: Multigrid vs Simple VFI

5 Chebyshev

Just in case here is the problem and the first order conditions.

$$V(k, z) = \max_{k', l, c} \left\{ (1 - \beta) \left[\ln(c) - \frac{l^2}{2} \right] + \beta \sum_{z'} V(k', z') \mathbb{P}[z'|z] \right\}$$

$$c = e^z k^\alpha l^{1-\alpha} + (1 - \delta)k - k'$$

First order conditions are given as follows. We will use these equations to compute several variables of interest. In particular given (k, l) (and production) the intratemporal condition (first equation of (5)) is going to help us figure out the value of consumption and the Euler equation (second equation of (5)) is what we use to find the appropriate coefficients of the Chebyshev approximation.

$$\begin{aligned} (\text{FOC})_l \quad & \frac{1 - \alpha}{c} \cdot e^z \left(\frac{k}{l} \right)^\alpha = l \\ (\text{FOC})_{k'} \quad & \frac{1}{c} = \beta \cdot \mathbb{E} \left[\frac{1}{c'} \cdot \left(\alpha e^z \left(\frac{l'}{k'} \right)^{1-\alpha} + 1 - \delta \right) \middle| z \right] \end{aligned} \tag{5}$$

Figure (5) shows the value and policy functions, figure (6) show the Euler equation errors of the approximation and finally figure (7) compares this solution with the one we got in question 2. We do linear interpolation to do the comparison.

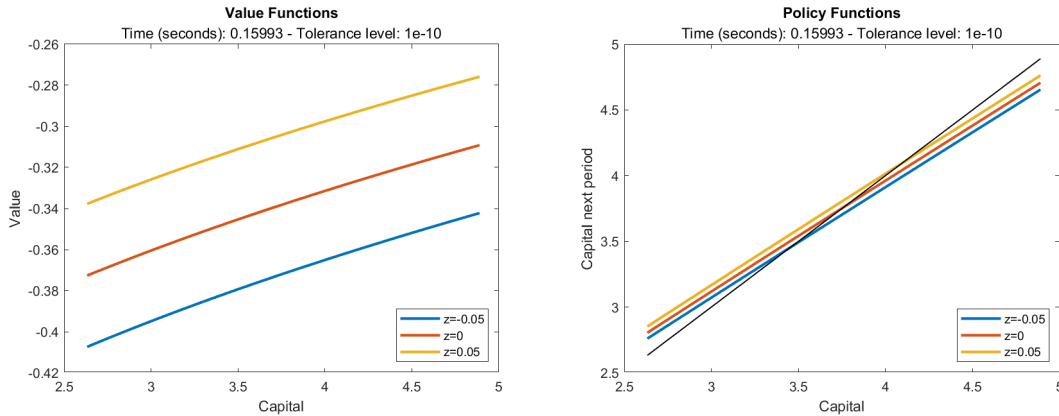


Figure 5: Value and Policy Functions - Chebyshev Polynomials

Here we briefly describe the methodology.

- Script *Econ714_hw2_2_Chebyshev.m* does the computation and generates the plots.
- Function *residual_fcn_2.m* does most of the job. We tried to modify the original code *residual_fcn.m* to adapt it to this problem. Given a set of coefficients stored in vector ρ this function calculates the policy function for labor, takes advantage of the first order condition to get consumption and capital next period and repeat the process one more time to construct the residuals. Remember that for the residuals we need information about the current period and next period. After that it calculates the residuals which will depend on ρ .

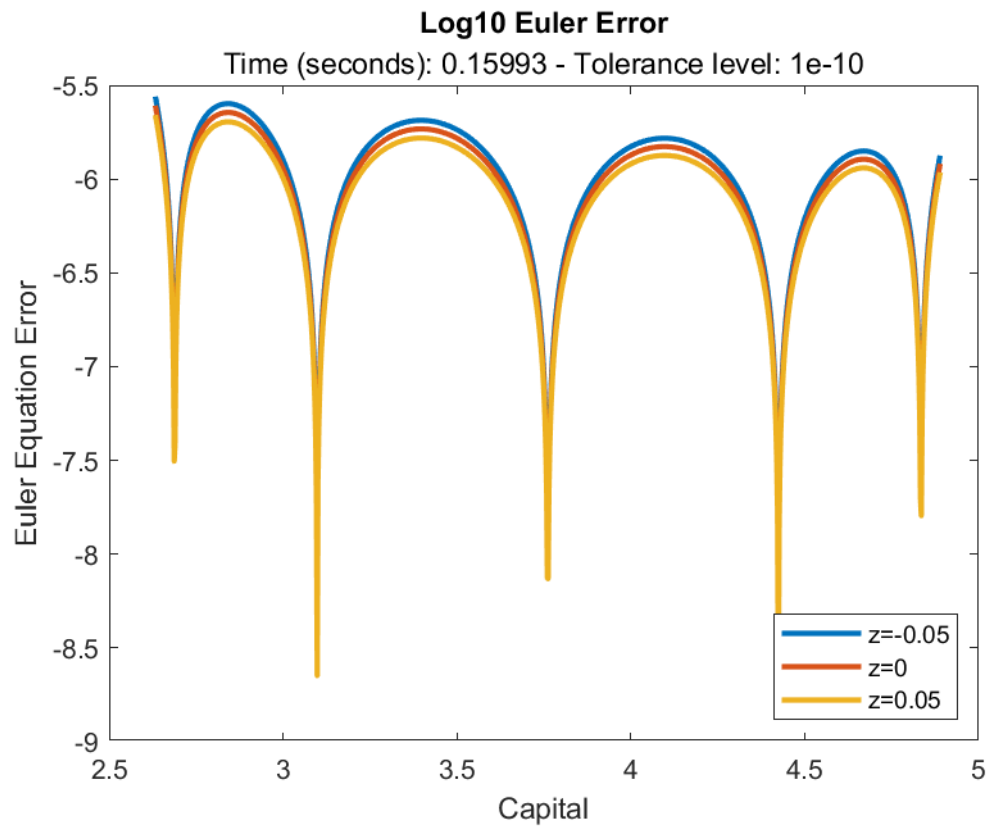


Figure 6: Euler Equation Errors - Chebyshev Polynomials

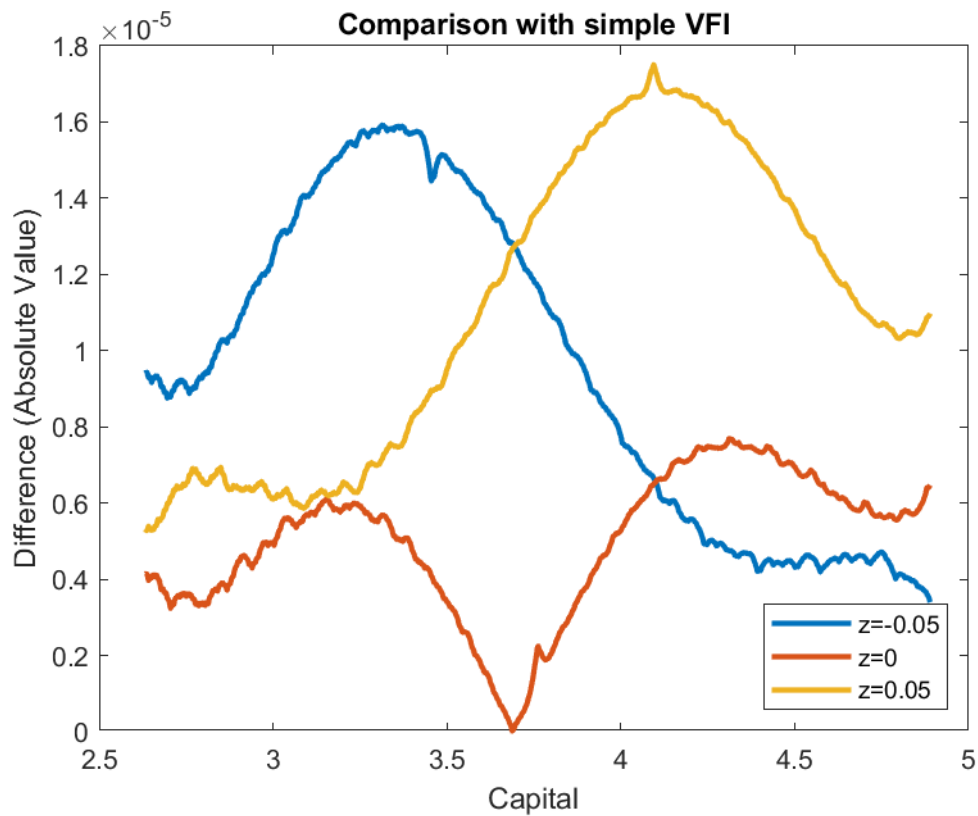


Figure 7: Chebyshev vs Simple VFI

- Finally the main script finds ρ such that the residuals are zero and with this choice ρ^* we can compute the policy and value functions.

6 Finite Elements

I try to follow McGrattan, E. R. (1996). My idea is to solve the system given by equation (15) in that paper. Let E be the number of elements and z the productivity shock (remember there are three of them).

$$\sum_{j=1}^8 \int_{k_j}^{k_{j+1}} w(\vec{k})_E \cdot R(k, z, \theta) dk = 0 \quad (6)$$

- Function *weight_fun_capital.m* has inputs $(k, \mathbf{k}_{\text{ELEMENTS}})$ where k is a state of capital and $\mathbf{k}_{\text{ELEMENTS}}$ is a vector that stores the elements of capital. It spits up a vector of weights that add up to one. For a given value of capital at most two elements of the vector are positive. If only one entry of the vector is positive (and equal to 1) that means we are exactly at a value of the element. This function gives me vector $w(\vec{k})$.
- Function *residual_fun.m* calculates the residuals. Its inputs are the current states (k, z) and a vector of 24 parameters that we want to find θ . It calculates the policy function for labor today and tomorrow using the vector θ as well as other variables (consumption, output, capital tomorrow) that are used to construct the residual. Then it calculates the residuals using the Euler equation where we average out the productivity levels tomorrow. Here we have 8 residuals (at least 6 of them are zero) and this is the output of this function which corresponds to what is inside the integral before weighting it. We will have to solve 24 integrals because we apply *residual_fun.m* three times (one for each value of the productivity shock) and then we want to integrate out capital.
- Function *residuals_averaged_out.m* approximates the integral of the weighted residuals within each element. Then add them together to form all the equations that we need to solve for θ given by system (6). Its inputs are parameters, number of elements, vector of elements and number of evaluations. I calculate the integral using the simplest method a Riemann sum on an equally spaced grid within the elements. For each (k, z) we only need to solve only two integrals because the other weights are zero.
- Finally the script *Econ714_hw2_3_Finite_Elements.m* implements everything. We define 8 elements of unequal size (more precise near the steady state and larger when we are away from the steady state). We pass *residuals_averaged_out.m* into a linear solver to find the value of θ that makes the residual equal to the zero vector. With this vector θ^* then we can compute the policy function for labor, consumption and capital next period and also the value function. The script does not work because I cannot find a solution to the system. Figure (8) shows the results. These value functions do not make sense: we plot them here to show our work.

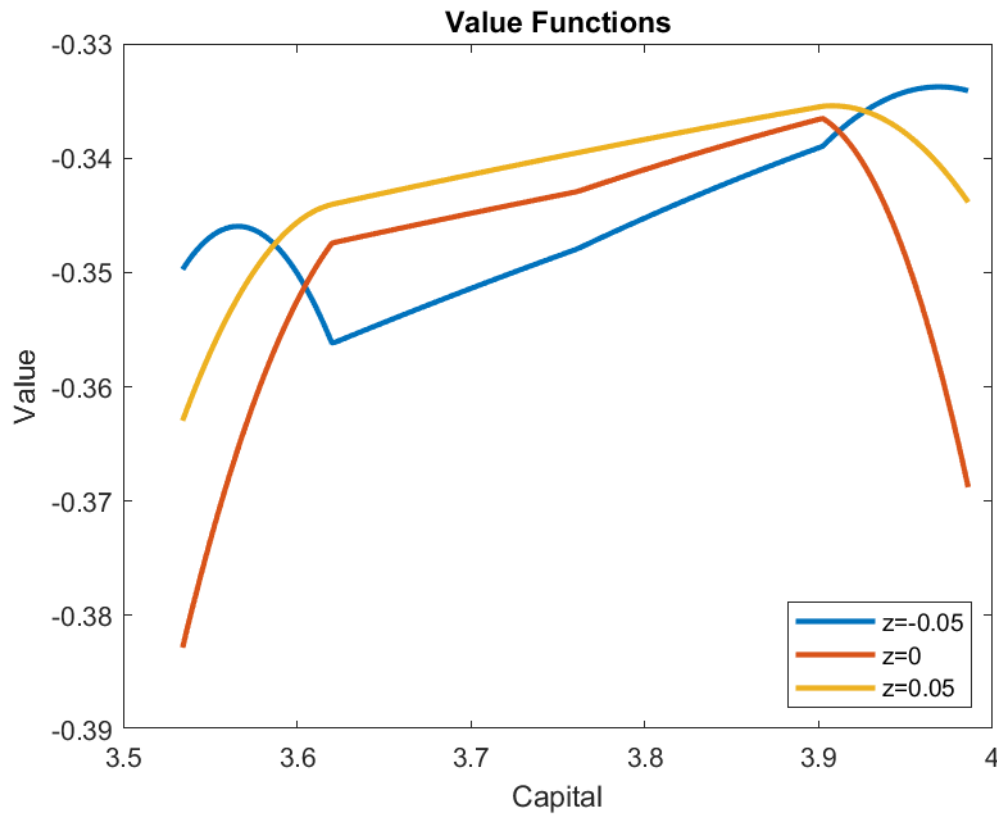


Figure 8: Value functions using finite elements

7 Deep Learning

I couldn't solve this one. I'm really sorry. I tried to follow Fernandez-Villaverde, J., Nuno, G., Sorg-Langhans, G., & Vogler, M. (2020) and the Julia code, but I couldn't even replicate the results for the continuous time model.

References

- Fernandez-Villaverde, J., Nuno, G., Sorg-Langhans, G., & Vogler, M. (2020). Solving high-dimensional dynamic programming problems using deep learning. *Unpublished working paper*.
- Heer, B., & Maussner, A. (2008). Value function iteration as a Solution Method for the Ramsey Model.
- McGrattan, E. R. (1996). Solving the stochastic growth model with a finite element method. *Journal of Economic Dynamics and Control*, 20(1-3), 19-42.