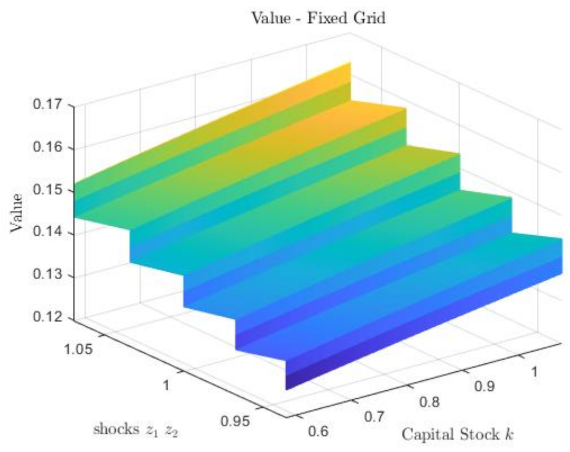
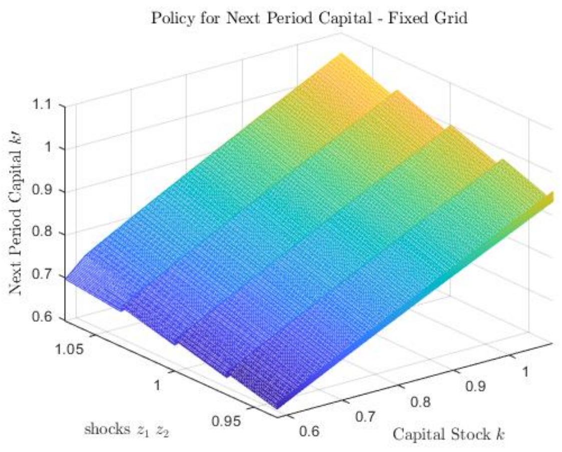
Contents

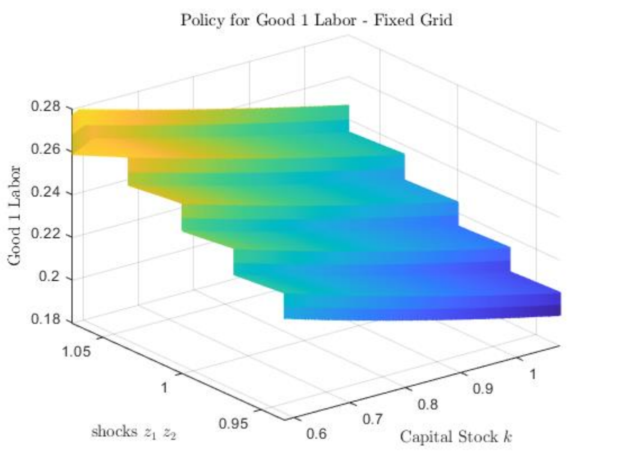
Value Function Iteration with a Fixed Grid

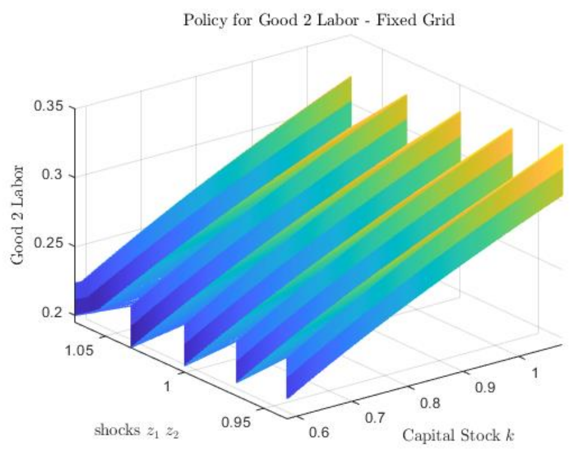
Fix a grid of 250 points of capital, centered around kss with a coverage of30% of kss and equally spaced. Iterate on the Value function implied by the Social Planner's Problem using linear interpolation until the change in the sup norm between to iterations is less than 106: Compute the Policy function. Describe the responses of the economy to a technology shock.

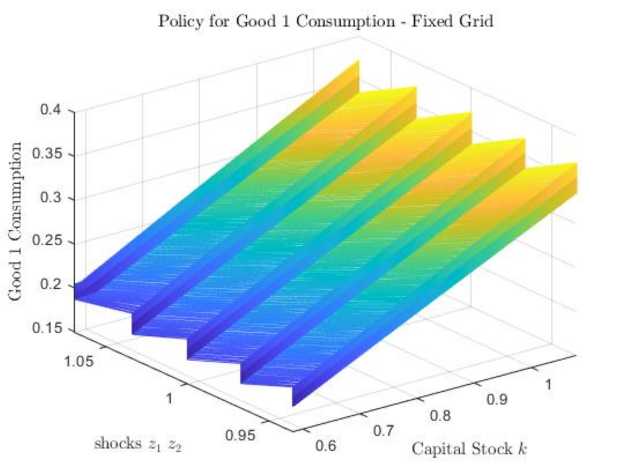
I did the value function iteration (VFI henthforth) in two steps. First I set the labor level to steady state and iterate till convergence. Then I use the value function got in the first step as the initial guess for the regular VFI where I make optimal decisions of both capital and labor. The result seems normal, but the running time is prohibitively slow – more than 11 hours on my computer, and more than 4 hours on the lab computer. Below are the results. Please note that the axis for shocks are integrated so that we can see how the value changes with the shock to employment while holding the productivity shock constant.

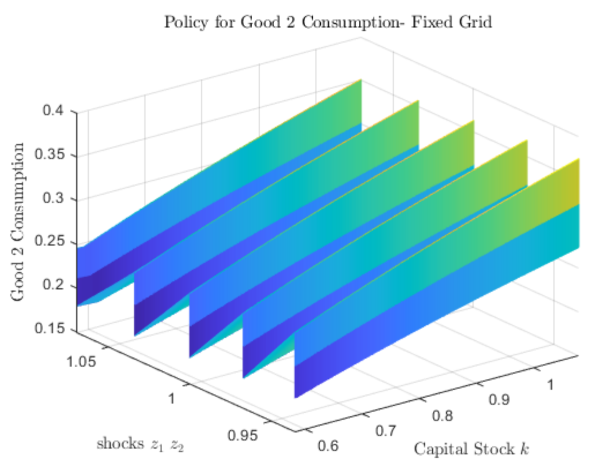






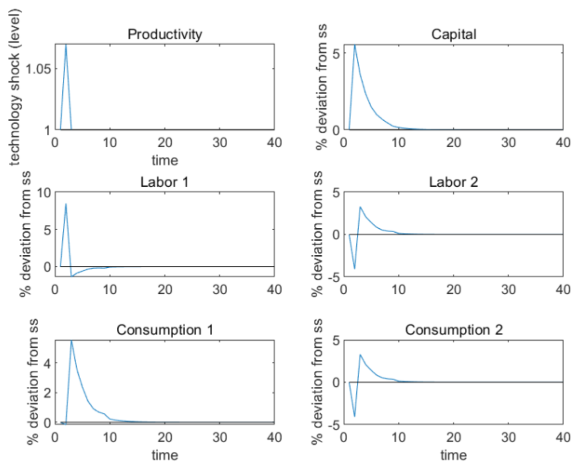


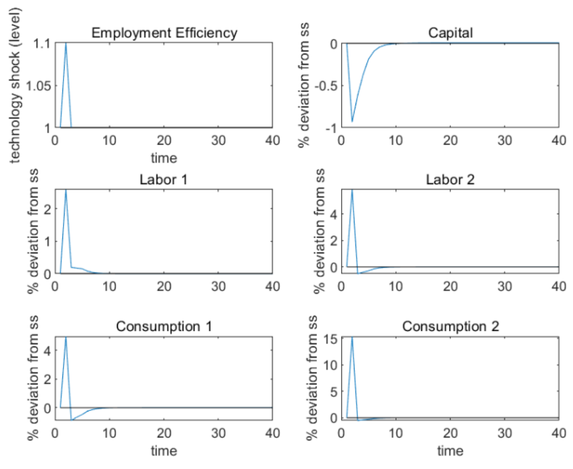




Since the value function and policy functions looks almost the same for every algorithm, I will not copy paste any more of them herein.

Below depict the impulse response to different shocks.





We can see from the impulse response that a productivity shock boosts up the capital but an employment shock lowers capital stock. The employment shock increases the return of the production of good 2, so people participate more in the production of good 2 than in good 1, but the production of good 1 is the one that gets to accumulate capitals.

In terms of labor participation, both productivity and employment shock boost up labor force participation for good 1, but the former decreases labor for good 1 and increases labor for good 2. Such case is plausible, since productivity shock encourages labor participation in good 1, whose production is directly affected by productivity shocks, and discourages labor for good 2, since production of good 1 now yields higher marginal returns than of good 2. But employment shock affects both labor 1 and labor 2 in the positive direction.

Value Function Iteration with an Endogenous Grid

Repeat previous exercise with an endogenous grid.

I spent over 8 hours on this part alone, and still couldn't get the iteration to converge, even with a really good guess, which I got from VFI with no interpolation, which is really fast. Then using it as the initial guess, I set both types of labor to steady state to do endogenous grid scheme for capital. But it won't converge. I also tried to set the level of labor according to the policy function I got from the previous no-interpolation VFI, but still it won't converge. If I were given some more time I would have a more thorough grasp on the scheme.

Comparison of Grids

Compare the solutions in 2) and 3) in terms of: 1) accuracy, 2) computing time, and 3) complexity of implementation. Present evidence to support your claims.

I did not come up with the scheme of endogenous grid, so I won't compare between fixed grid and the endogenous grid scheme. But at the end I'll compare among fixed grid and other scheme I've tried.

I don't know about the accuracy and computer time, but the implementation of endogenous grid scheme is much more complicated than the regular VFI.

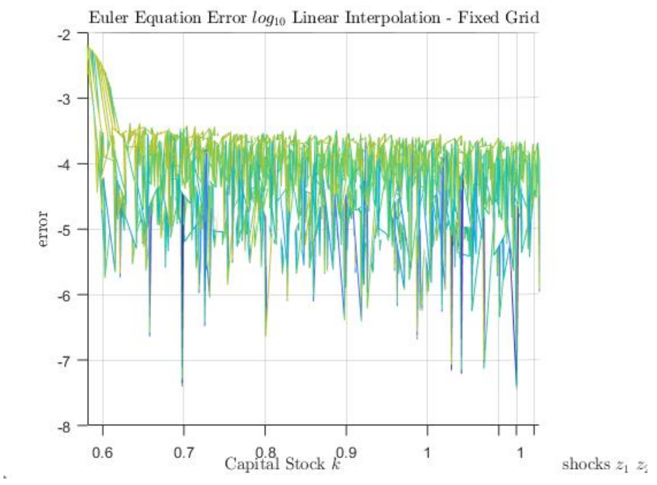
Accelerator

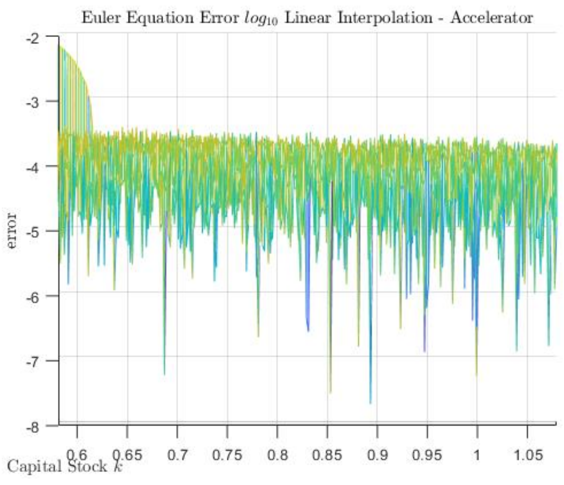
Recompute your solution to 2) using an accelerator, i.e., skipping the max operator in the Bellman equation 9 out of each 10 times. Compare accuracy and computing time between the simple grid scheme implemented in 2) and the results from the accelerator scheme. Present evidence to support your claims.

The accelerator scheme is easy to code, but we should make sure that the last iteration is the one that does the maximization.

In terms of computing time, for the fixed grid scheme, it runs on the lab computer for about 4 hours, while the accelerator runs about 41 minutes. Both are done with two steps – first set labor to steady state level, and then do the regular VFI.

In terms of accuracy, when comparing between the Euler Equation errors of these two, I find no significant difference between these two, but the accelerator scheme is slightly better.



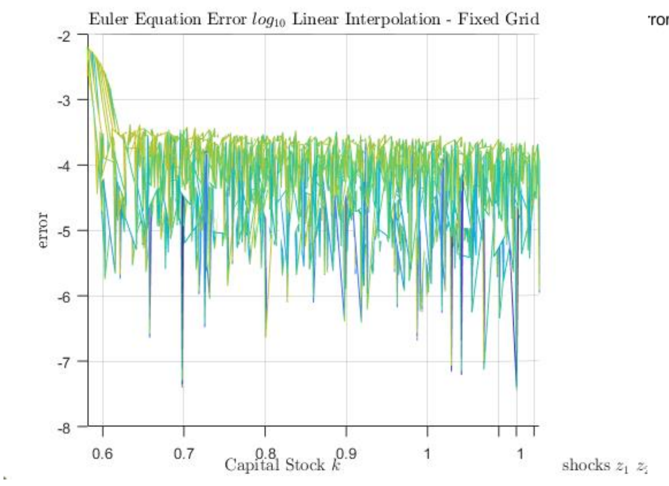


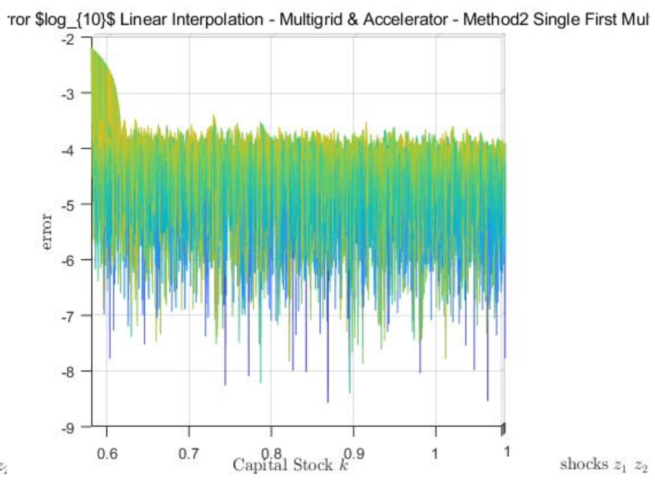
Multigrid

Implement a multigrid scheme (Chow and Tsitsiklis, 1991) for a Value function iteration, with the grid centered around k_{ss} with a coverage of $\pm$30% of k_{ss} and equally spaced.
You will have 100 capital grid points in the first grid, 500 capital grid points in the second, and 5000 capital grid points in the third.
Compare accuracy and computing time between the simple grid scheme implemented in 2) and the results from the multigrid scheme. Present evidence to support your claims.

The multigrid scheme is easy to implement. On a lab computer, it runs about 2.7 hours with 5000 grid points, and compared to the simple grid scheme with 250 grid points which runs about 4 hours, it's extremely fast. Also note that I integrate the accelerator scheme into the multigrid scheme.

And when comparing between the simple scheme and the multigrid scheme in terms of Euler equation error, we can see the latter is significantly better.



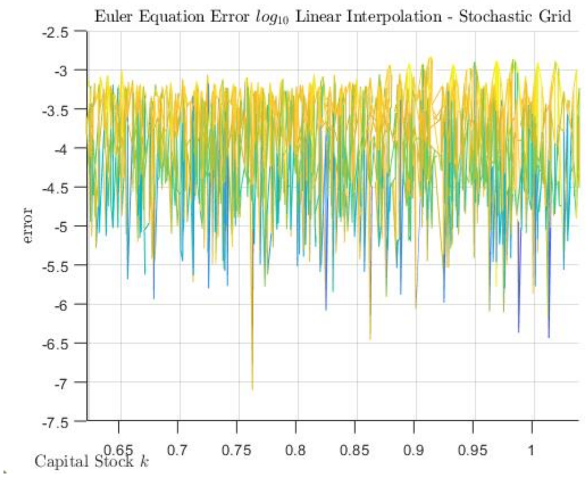


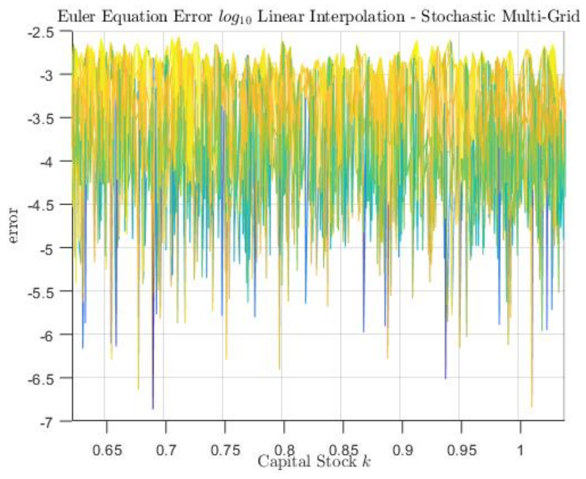
Stochastic Grid

Implement a stochastic grid scheme (Rust, 1997) for a Value function iteration, with 500 vertex points with a coverage of 25% of kss (you can keep the grid of investment fixed). Compare accuracy and computing time between the simple grid scheme implemented in 2) and the results from the multigrid scheme. Present evidence to support your claims

In terms of computing time, both are really fast (i.e., less than 5 seconds), since I used grid search instead of linear interpolation for the VFI. By precomputing a big matrix that gives me all the current utility entry for every k' chosen, the computation is really fast.

In terms of accuracy, I don't see much a difference.


Euler Equation Error log_{10} Linear Interpolation - Stochastic Grid
error
Capital Stock k


Euler Equation Error log_{10} Linear Interpolation - Stochastic Multi-Grid
error
Capital Stock k