Homework 1

Juan C. Mendez-Vizcaino*

October 31st,2024

1 Github

The repository for this homework can be found in: https://github.com/jcmendev/HW1

It contains 7 routines described in the sections of this document

1. HW1_1.m: Integration
2. HW1_2.m: Optimization
3. HW1_3.m: Computing Pareto Efficient Allocations
4. HW1_4.m: Computing Equilibrium Allocations
5. HW1_5_MGSI.m: Contains the code for Value Function Iteration
   (a) Mainly solves 6.2 and 6.3 in the homework
   (b) Allows for doing multigrid (6.7 in homework)
   (c) Allows for alternating between value function (6.6 in homework)
   (d) Allows for computing IRFs (6.3 in homework)
6. HW1_5_EGM.m: Contains the code for Endogenous Grid Method (incomplete)
7. Bilinear_Interpolation.m: Self code for bilinear interpolation

2 Integration

Consider

$$\int_0^T e^{-\rho t} u\left(1-e^{-\lambda t}\right) dt$$

for $T = 100$, $\rho = 0.04$, $\lambda = 0.02$ and $u(.) = -e^{-c}$. Using quadrature methods and Monte Carlo for integration, for $n = \{10, 100, 100, 10000\}$ number of nodes, we obtain the following results.

*jmendevi@sas.upenn.edu

First, given the parameterization, we can find the analytic solution to this integral (using symbolic integration package in Matlab),

$$AI = \int_0^T e^{-\rho t} u \left(1 - e^{-\lambda t}\right) dt = \frac{1}{\lambda} e^{\left(e^{-\lambda t} - 1\right)} \left(e^{-\lambda t} - 1\right) |_0^T = -18.209525399981501$$

We now compare the accuracy, x, of the method, y, relative to this analytic solution, AI as

$$x = 10000 * \left(\frac{y}{AI} - 1\right)$$

we report the results in Table 1.

Table 1: Integration using quadrature methods and Monte Carlo integration Relative to Analytic Solution

Nodes	**Midpoint**	**Trapezoid**	**Simpson**	**Monte Carlo**
10	-1.346028532	3.337663095	0.083010534	42.23117188135
100	-0.013651721	0.027859491	0.000009002	-13.37230355433
1000	-0.000136536	0.000273620	0.000000000	-3.94697117003
10000	-0.000001365	0.000002731	0.000000000	1.11876926336
100000	-0.000000013	0.000000027	0.000000000	-0.22851063592

The results show that Simpson integration is consistently closer to the analytic solution for any number of evaluation points, while Monte Carlo performs consistently worse. Midpoint performs better than Trapezoid while is also in average 10 times faster (.001092 vs .011987) and around 2 times faster than Simpson. The computational performance of Monte Carlo is roughly the same as Midpoint.

3 Optimization

Consider the Rosenbrock function

$$\min_{x,y} 100 \left(y - x^2\right)^2 + (1 - x)^2$$

With global minimum at $(x, y) = (1, 1)$.

To find the minimum, we use 4 methods: Newton-Raphson (NR), BFGS, steepest descent (SD) and conjugate descent (CD).

The initial guess for all the algorithms is $\left(x^0, y^0\right) = (0, 0)$, we compute the gradient and the Hessian analytically (Matlab symbolic toolbox), and for the steepest and conjugate descent we use a simple non-adaptive search parameter $\alpha = 0.01$.

Table 2: Optimization methods

	$\left(x^0, y^0\right) = (0,0)$		$\left(x^0, y^0\right) = (2,2)$	
	Time (sec)	**Accuracy**	**Time (sec)**	**Accuracy**
NR	0.002	10^{-14}	0.0035	10^{-12}
BFGS	0.008	10^{-10}	0.0062	10^{-6}
SD	0.010	10^{-4}	0.015	10^{-5}
CD	0.230	10^{-15}	0.37	10^{-15}

4 Computing Pareto Efficient Allocations

Consider an endowment economy with m different goods and n agents. Each agent $i = 1, ..., n$ has an endowment $e_j^i > 0$ for every $j = 1, ..., m$ and a utility function of the form

$$u^i(x) = \sum_{j=1}^{m} \alpha_j^i \frac{\left(x_j^i\right)^{1+\omega_j^i}}{1+\omega_j^i}$$

where $\alpha_j^i > 0 > \omega_j^i$ are agent specific parameters

Consider the social problem given by

$$\max_{\left\{x_j^i\right\}_{i,j}} \sum_{i=1}^{n} \lambda^i \sum_{j=1}^{m} \alpha_j^i \frac{\left(x_j^i\right)^{1+\omega_j^i}}{1+\omega_j^i}$$

s.t.

$$\sum_{i=1}^{n} \left(x_j^i\right) \leq \sum_{i=1}^{n} \left(e_j^i\right) \qquad \text{for each } j = 1, ...m$$

where $\lambda^i > 0$ are the planner's social weights. The first order conditions for this problem are given by

$$\left[x_j^i\right] = \lambda^i \alpha_j^i \left(x_j^i\right)^{\omega_j^i} = \psi_j$$

for all $i = 1, ..., n$ and all $j = 1, ..., m$, and ψ_j is the associated Lagrange multiplier. Note that the relative consumption between two goods for one agent is given by

$$\lambda^i \alpha_k^i \left(x_k^i\right)^{\omega_k^i} = \lambda^i \alpha_j^i \left(x_j^i\right)^{\omega_j^i}$$

or alternatively

$$\frac{\left(x_k^i\right)^{\omega_k^i}}{\left(x_j^i\right)^{\omega_j^i}} = \frac{\alpha_j^i}{\alpha_k^i}$$

for $k \neq j$. Similarly the relative consumption between two agents for the same good is given by

$$\frac{\left(x_j^l\right)^{\omega_k^l}}{\left(x_j^i\right)^{\omega_j^i}} = \frac{\lambda^i \alpha_j^i}{\lambda^l \alpha_j^l}$$

for $l \neq i$. Suppose $\sum_{i=1}^{n} \left(e_j^i\right) = 1$ for all j, so we can interpret the holdings of the endowment as a share of the total endowment. Suppose in this economy $m = n = 3$. The endowments are randomized across agents and goods. and are given as follows

Table 3: Endowments in Social Planner's Problem

	Good 1	**Good 2**	**Good 3**	
Agent 1	0.57585	0.32547	0.45691	1.35
Agent 2	0.23658	0.42474	0.31904	0.98
Agent 3	0.18756	0.24978	0.22405	0.66
Total	1	1	1	

In this section we will consider three cases: (i) No heterogeneity, (ii) Different weights, (iii) Different coefficient of relative risk-aversion.

The solution for such kind of problem is straight-forward. We have $m * n$ first order conditions, one for each good for each agent. and m resource constraints. then in total we have $m(n+1)$ equations. On the other hand we have $m * n$ goods $\left\{x_j^i\right\}_{i,j}$ and m Lagrange multipliers, $\{\psi_j\}_j$. Then this problem could be solve either as solving a non-linear system of equations or as an constrained optimizing problem, by directly maximizing the objective function with the resource constraint.

No heterogeneity

This exercise is the least interesting one, but its purpose is to set a reference for the other exercises. In this exercise, we set $\lambda^i = \frac{1}{n}$, $\alpha_j^i = \frac{1}{m}$ and a standard $\omega_j^i = -2$. In Table 4 we show the allocation between agents and goods from this exercise. As expected, since there is no heterogeneity, all the agents shall consume the same amount of each good.

Table 4: Social Planner's Solution: No Heterogeneity

	Good 1	**Good 2**	**Good 3**	
Agent 1	0.333	0.333	0.333	1
Agent 2	0.333	0.333	0.333	1
Agent 3	0.333	0.333	0.333	1
Total	1	1	1	

Different Pareto Weights: The Favourite

In this exercise we set $\lambda^1 = 0.5$ and $\lambda^2 = \lambda^3 = 0.25$, the remaining parameters are unchanged $\alpha_j^i = \frac{1}{m}$ and a standard $\omega_j^i = -2$. Agent one has a higher Pareto weight in the planner's solution, which makes him consume more of every good than the other two agents. However, he's consuming less than half of the total resources of the economy even though he has double Pareto weight.

Table 5: Social Planner's Solution: "The Favourite"

	Good 1	**Good 2**	**Good 3**	
Agent 1	0.41421	0.41421	0.41421	1.24
Agent 2	0.29289	0.29289	0.29289	0.879
Agent 3	0.29289	0.29289	0.29289	0.879
Total	1	1	1	

Different Preference for Goods

In this exercise we set $\lambda^i = \frac{1}{n}$, and $\omega_j^i = -2$. For this exercise, we set the preference for goods to be

$$\alpha_j^i = \begin{cases} 1.25 & \text{if } j = i \\ 0.25 & \text{Otherwise} \end{cases}$$

Which means that agents derive more utility for the good indexed by their own. The solution of the planner's problem implies that each agent consumes more of that specific good, and the planner allocates almost the double of this good for each agent.

Table 6: Social Planner's Solution: Preference for Goods

	Good 1	**Good 2**	**Good 3**	
Agent 1	0.52786	0.23607	0.23607	1
Agent 2	0.23607	0.52786	0.23607	1
Agent 3	0.23607	0.29289	0.23607	1
Total	1	1	1	

Different CRRA parameter

In this exercise, we set $\lambda^i = \frac{1}{n}$, $\alpha_j^i = \frac{1}{m}$ and a $\omega_j^i = -2$ for $i = 1, 2$ and $\omega_j^i = -4$ for $i = 3$ which makes agent 3 more risk averse than the other two agents.

As expected, social planner compensates this risk aversion by allocating more of every good to agent 3. Note that in this exercise, agent 3 consumes even more than in the exercise of Pareto weights.

Table 7: Social Planner's Solution: "The Favourite"

	Good 1	**Good 2**	**Good 3**	
Agent 1	0.5	0.5	0.5	1.5
Agent 2	0.25	0.25	0.25	0.75
Agent 3	0.25	0.25	0.25	0.75
Total	1	1	1	

All exercises together

In this last experiment we combine the three exercises. As we distinguished before, the agent with highest risk aversion is the one that consumes the most of the total endowment, followed by the preferred agent in the Pareto weights. The agent that has a lower consumption of the three is agent 2.

Table 8: Social Planner's Solution: "The Favourite"

	Good 1	**Good 2**	**Good 3**	
Agent 1	0.4676	0.23088	0.23143	0.9299
Agent 2	0.14787	0.36506	0.16365	0.6766
Agent 3	0.38454	0.40405	0.60492	1.3935
Total	1	1	1	

The algorithm is robust up to $m = n = 8$ but has convergence problem for $m = n = 10$.

5 Computing Equilibrium Allocations

Consider now the decentralized equilibrium of this endowment economy with m different goods and n agents given by,

Each agent solves

$$\max_{\{x_j^i\}_j} \sum_{j=1}^{m} \alpha_j^i \frac{\left(x_j^i\right)^{1+\omega_j^i}}{1+\omega_j^i}$$

s.t

$$\sum_{j=1}^{m} p_j \left(x_j^i\right) \leq \sum_{j=1}^{m} p_j \left(e_j^i\right) \qquad [DP]$$

The first order conditions of this problem read,

$$[x_j^i]\alpha_j^i \left(x_j^i\right)^{\omega_j^i} = \phi_i p_j$$

As in the Social planner, for the decentralized equilibrium, the relative consumption between two goods for one agent is given by

$$\frac{\alpha_k^i}{p_k}\left(x_k^i\right)^{\omega_k^i} = \frac{\alpha_j^i}{p_j}\left(x_j^i\right)^{\omega_j^i}$$

or alternatively

$$\frac{\alpha_k^i\left(x_k^i\right)^{\omega_k^i}}{\alpha_j^i\left(x_j^i\right)^{\omega_j^i}} = \frac{p_k}{p_j}$$

for $k \neq j$. Note that we can draw a equivalence between social planner and decentralized equilibrium with

$$p_k = \frac{1}{\psi_j}$$

but we will require the adequate weights, following the first and second welfare theorems. Relative consumption between two agents for the same good is given by

$$\frac{\left(x_j^l\right)^{\omega_k^l}}{\left(x_j^i\right)^{\omega_j^i}} = \frac{\alpha_j^i}{\alpha_j^l}$$

The decentralized equilibrium consists on allocations $\left\{x_j^i\right\}_{i,j}$ and prices $\{p_j\}_j$such that given prices $\{p_j\}_j$, agents solve their optimization problem $[DP]$ and the resource constraint holds

$$\sum_{i=1}^{n}\left(x_j^i\right) \leq \sum_{i=1}^{n}\left(e_j^i\right) \qquad \text{for each } j = 1, ...m$$

No heterogeneity

In this exercise, unlike the Social Planner, the agents consumption will depend on their initial endowments. In this exercise, we set $\alpha_j^i = \frac{1}{m}$ and a standard $\omega_j^i = -2$. The allocation between agents and goods from this exercise is different to the one of no heterogeneity in the social planner's solution. As expected, without any further heterogeneity than the one imposed by initial endowments, the agent with the highest endowment (agent 1) shall consume more of all goods.

Table 9: Decentralized Solution: No Heterogeneity

	Good 1	**Good 2**	**Good 3**
Agent 1	0.45274	0.45274	0.45274
Agent 2	0.32679	0.32679	0.32679
Agent 3	0.22047	0.22047	0.22047
Total	1	1	1

And no internal heterogeneity (preferences) implies that prices are the same and given by

Table 10: Decentralized Prices: No Heterogeneity

	Good 1	**Good 2**	**Good 3**
Price	0.57701	0.57701	0.57701

Different Preference for goods

$\omega_j^i = -2$. For this exercise, we set the preference for goods to be

$$\alpha_j^i = \begin{cases} 1.25 & \text{if } j = i \\ 0.25 & \text{Otherwise} \end{cases}$$

As expected, when we include heterogeneity in the preferences for goods, agents consume more of that good they prefer. However, as in the no-heterogeneity case, agent 1 is the one that consumes the most of good 1 (relative to the social planner's solution), but it also implies she pays a higher price for good 1.

Table 11: Decentralized Solution: No Heterogeneity

	Good 1	**Good 2**	**Good 3**
Agent 1	0.64799	0.32266	0.35315
Agent 2	0.20738	0.51631	0.25272
Agent 3	0.14463	0.16104	0.39412
Total	1	1	1

And prices are given by

Table 12: Decentralized Prices: No Heterogeneity

	Good 1	**Good 2**	**Good 3**
Price	0.559318	0.47848	0.39941

6 Value Function Iteration

6.1 Social Planner

6.1.1 Choosing expenditure with no taxes

In this first formulation, we consider government expenditure as part of the physical environment of the problem. Particularly, the planner's problem considers allocations $\{c, i, l, k', g\}$. Consider the recursive formulation of the social planner's problem given by,

$$V\left(k,i_{-},z\right)=\max_{c,i,g,l,k'}\left\{u\left(c,g,l\right)+\beta\sum_{z'|z}\pi\left(z'|z\right)V\left(k',i,z'\right)\right\}$$

$$\begin{aligned} &\text{s.t.} && \text{LM}\\ &c+i+g=\tilde{z}k^{\alpha}l^{1-\alpha} && \Psi\\ &k'=\left(1-\delta\right)k+\Phi\left(i,i_{-}\right)i && \mu \end{aligned}$$

where $\tilde{z}=e^{z}$, the utility function

$$u\left(c,g,l\right)=\log\left(c\right)+\chi\log\left(g\right)-\frac{l^{1+\gamma}}{1+\gamma}$$

and the investment net of the adjustment costs

$$\Phi\left(i,i_{-}\right)i=\left(1-\frac{\phi}{2}\left(\frac{i}{i_{-}}-1\right)^{2}\right)i$$

In this environment, there is no role for taxes, which can be reached by combining budget constraint of a decentralized household, with aggregate resource constraint and a government budget constraint.

The first order conditions for this problem are given by

$$\begin{aligned} \left[c\right]:&u_{c}-\Psi=0\\ \left[l\right]:&u_{l}+\Psi\left[\left(1-\alpha\right)\tilde{z}k^{\alpha}l^{-\alpha}\right]=0\\ \left[g\right]:&u_{g}-\Psi=0\\ \left[k'\right]:&\beta\sum_{z'|z}\pi\left(z'|z\right)V_{k}\left(k',i,z'\right)-\mu=0\\ \left[i\right]:&\beta\sum_{z'|z}\pi\left(z'|z\right)V_{i}\left(k',i,z'\right)+\frac{\partial\Phi\left(i,i_{-}\right)i}{\partial i}\mu-\Psi=0 \end{aligned}$$

and the envelope conditions

$$\begin{aligned} \left[k\right]:V_{k}\left(k,i_{-},z\right)&=\Psi\left[\alpha\tilde{z}k^{\alpha-1}l^{1-\alpha}\right]+\left(1-\delta\right)\mu\\ \left[i_{-}\right]:V_{i}\left(k,i_{-},z\right)&=\mu\frac{\partial\Phi\left(i,i_{-}\right)i}{\partial i_{-}} \end{aligned}$$

The social planner's steady state is then given by the following system of equations,

$$\begin{aligned}
[c] &: u_{c,ss} - \Psi_{ss} = 0 \\
[l] &: u_{l,ss} + \Psi_{ss}\left[(1-\alpha)\tilde{z}_{ss}k_{ss}^{\alpha}l_{ss}^{-\alpha}\right] = 0 \\
[g] &: u_{g,ss} - \Psi_{ss} = 0 \\
[k'] &: \Psi_{ss}\left[\alpha\tilde{z}_{ss}k_{ss}^{\alpha-1}l_{ss}^{1-\alpha}\right] + (1-\delta)\mu_{ss} = \mu_{ss} \\
[i] &: \mu_{ss} - \Psi_{ss} = 0 \\
[RC] &: c_{ss} + i_{ss} + g_{ss} - \tilde{z}_{ss}k_{ss}^{\alpha}l_{ss}^{1-\alpha} = 0 \\
[LoM] &: \delta k_{ss} - i_{ss} = 0
\end{aligned}$$

6.1.2 Choosing labor without government expenditure

An alternative formulation for this problem considers government expenditure to be produced with labor and with a productivity shock given by τ.

$$g = \tau(1-\alpha)\tilde{z}k^{\alpha}l^{1-\alpha}$$

Consider the recursive formulation of the social planner's problem given by,

$$\begin{aligned}
V(k, i_-, z, \tau) = \max_{c,i,l,k'} & \left\{u(c,g,l) + \beta\sum_{z'|z}\pi(z'|z)\sum_{\tau'|\tau}\pi(\tau'|\tau)V(k', i, z', \tau')\right\} \\
\text{s.t.} & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{LM} \\
& c + i = \hat{\tau}\tilde{z}k^{\alpha}l^{1-\alpha} \qquad\qquad\qquad\qquad\qquad\qquad \Psi \\
& k' = (1-\delta)k + \Phi(i, i_-)i \qquad\qquad\qquad\qquad\qquad \mu
\end{aligned}$$

where $\tilde{z} = e^{z}$, and $\hat{\tau} = (1-\tau(1-\alpha))$, the utility function

$$u(c,g,l) = \log(c) + \chi\log\left(\tau(1-\alpha)\tilde{z}k^{\alpha}l^{1-\alpha}\right) - \frac{l^{1+\gamma}}{1+\gamma}$$

and the investment net of the adjustment costs

$$\Phi(i, i_-)i = \left(1 - \frac{\phi}{2}\left(\frac{i}{i_-} - 1\right)^2\right)i$$

6.2 Recursive Competitive Equilibrium (RCE) and Steady State

A RCE consists on allocations $\{c, i, l, k'\}$ and prices $\{r, w\}$ and policies $\{g\}$, and aggregate states $S = \{K, I_-, z, \tau\}$ such that,

1. Given prices $\{r(S), w(S)\}$ and policies $\{g\}$, households solve

$$\tilde{V}(k, i_-, z, \tau; S) = \max_{c,i,g,l,k'} \left\{ u(c,g,l) + \beta \sum_{z'\tau'|z,\tau} \pi\left(z', \tau|z, \tau\right) \tilde{V}\left(k', i, z', \tau'; S'\right) \right\}$$

$$\begin{aligned} &\text{s.t.} && \text{LM} \\ &c + i = (1-\tau) w(S) l^s + r(S) k && \Psi \\ &k' = (1-\delta) k + \Phi(i, i_-) i && \mu \end{aligned}$$

2. Given prices $\{r(S), w(S)\}$ and policies $\{g\}$, firms solve

$$\max \tilde{z} K^{\alpha} L^{1-\alpha} - w(S) L^d - r(S) K$$

which pins down prices as

$$\begin{aligned} w(S) &= (1-\alpha) \tilde{z} K^{\alpha} L^{-\alpha} \\ r(S) &= \alpha \tilde{z} K^{\alpha-1} L^{1-\alpha} \end{aligned}$$

3. Government balances its budget

$$g = \tau w(S) l$$

4. Markets clear

$$\begin{aligned} c + i + g &= \tilde{z} k^{\alpha} l^{1-\alpha} \\ l^s &= L^d \end{aligned}$$

5. Law of Motion and consistency

$$\begin{aligned} S' &= H(S) \\ \phi_k^{ra}(K, I_-, z, \tau; S) &= \phi_k^A(z, \tau; S) \\ \phi_i^{ra}(K, I_-, z, \tau; S) &= \phi_i^A(z, \tau; S) \end{aligned}$$

where the first refers to the law of motion of the aggregate states given the operator $H(.)$, $\phi_i^{ra}(.)$ for $i = \{k, i\}$are the policy functions of capital and investment of the representative agent, and ϕ_k^A the ones of an aggregate decision maker.

In this problem, the representative households FOC are given by

$$\begin{aligned} [c] &: u_c - \Psi = 0 \\ [l] &: u_l + \Psi [(1-\tau) w(S)] = 0 \\ [k'] &: \beta \sum_{z'\tau'|z,\tau} \pi\left(z', \tau|z, \tau\right) V_k\left(k', i, z'\right) - \mu = 0 \\ [i] &: \beta \sum_{z'\tau'|z,\tau} \pi\left(z', \tau|z, \tau\right) V_i\left(k', i, z'\right) + \frac{\partial \Phi(i, i_-) i}{\partial i} \mu - \Psi = 0 \end{aligned}$$

and the envelope conditions

$$\begin{aligned}
[k] : V_k\left(k, i_-, z\right) &= \Psi\left[r\left(S\right)\right] + \left(1-\delta\right)\mu \\
[i_-] : V_i\left(k, i_-, z\right) &= \mu \frac{\partial \Phi\left(i, i_-\right) i}{\partial i_-}
\end{aligned}$$

then the Euler equation for capital next period

$$\mu = \beta \sum_{z'\tau'|z,\tau} \pi\left(z', \tau | z, \tau\right)\left[\Psi' r\left(S'\right) + \left(1-\delta\right)\mu'\right]$$

and for investment

$$\Psi - \frac{\partial \Phi\left(i, i_-\right) i}{\partial i}\mu = \beta \sum_{z'\tau'|z,\tau} \pi\left(z', \tau | z, \tau\right)\left[\mu' \frac{\partial \Phi\left(i', i\right) i'}{\partial i}\right]$$

The steady state is given by the following system of equations,

$$\begin{aligned}
[c] :& u_{c,ss} - \Psi_{ss} = 0 \\
[l] :& u_{l,ss} + \Psi_{ss}\left[\left(1-\tau_{ss}\right) w\left(S\right)\right] = 0 \\
\left[k'\right] :& \Psi_{ss}\left[r\left(S\right)\right] + \left(1-\delta\right)\mu_{ss} = \mu_{ss} \\
[i] :& \mu_{ss} - \Psi_{ss} = 0 \\
[RC] :& c_{ss} + i_{ss} + g_{ss} - \tilde{z}_{ss} k_{ss}^{\alpha} l_{ss}^{1-\alpha} = 0 \\
[LoM] :& \delta k_{ss} - i_{ss} = 0 \\
[w] :& w_{ss}\left(S\right) = \left(1-\alpha\right)\tilde{z}_{ss} k_{ss}^{\alpha} l_{ss}^{-\alpha} \\
[r] :& r_{ss}\left(S\right) = \alpha \tilde{z}_{ss} k_{ss}^{\alpha-1} l_{ss}^{1-\alpha} \\
[g] :& g_{ss} = \tau_{ss} w_{ss}\left(S\right) l_{ss}^{s}
\end{aligned}$$

For the numerical implementation, consider $\tau_{ss} = 0.25$ and $z_{ss} = 0$, $\beta = 0.97$, $\chi = 0.2$, $\gamma = 1$, $\alpha = 0.33$, $\phi = 0.1$, $\delta = 0.1$. Note, that in the social planner solution we are not considering the exogenous distortionary effect of a tax, the undistorted allocations will generate higher consumption and output than in the decentralized equilibrium. The numerical steady state is given by:

6.3 Value Function Iteration with a fixed grid

In this problem we define a fix grid for capital k and lagged investment i_-. Note that when we are looking for a grid-search solution, if we allow for the investment i (control) to take values on the grid of lagged investment, then, by the law of motion of capital

$$k' = \left(1-\delta\right) k + \Phi\left(i, i_-\right) i$$

we are also implying that capital may not necessarily take values on the grid point, since the implied capital by the law of motion does not necessarily coincide with the fixed grid of capital.

Table 13: Steady State

		RCE	Unc. Social Planner
Consumption	c_{ss}	0.85	1.06
Labor	l_{ss}	0.93	1.07
Capital	k_{ss}	3.69	4.27
Government Expenditure	g_{ss}	0.24	0.21
Investment	i_{ss}	0.37	0.42
Wages/MPL	w_{ss}	1.05	1.05
Return on capital/MPK	r_{ss}	0.13	0.13

Then, instead of having just a grid for k' we will have a matrix of dimensions nxn^i. Finally, since k' might not be on the gridpoints of k, for value function iteration, we will need to interpolate to find $\tilde{V}\left(k', i, z', \tau'; S'\right)$.

Consider then the following algorithm

1. Guess prices, r_j, w_j, labor, l_j, and value function $\tilde{V}^j\left(k, i_-, z, \tau; S\right)$ where j indexes the iteration.

2. For each node in the state space (k, i_-, z, τ), compute $\sum_{z'\tau'|z,\tau} \pi\left(z', \tau | z, \tau\right) \tilde{V}^j\left(k', i, z', \tau'; S'\right)$ by bilinearly interpolating the guess $\tilde{V}^j\left(k, i_-, z, \tau; S\right)$ on (k', i) since k' might not take values on the grid

3. Compute consumption c_j using the household budget constraint, guesses and government expenditure in steady state
$$c_j = \left(1 - \tau\right) w_j l_j + r_j k - i$$

4. Compute labor with the marginal disutility of labor
$$\hat{l}_j = \left(\frac{1}{c_j}\left(1 - \tau\right) w_j\right)^{\frac{1}{\gamma}}$$

5. Compute instant utility, , using government expenditure in steady state
$$u\left(c_j, g_{ss}, \hat{l}_j\right) = \log\left(c_j\right) + \chi \log\left(g_{ss}\right) - \frac{\hat{l}_j^{1+\gamma}}{1+\gamma}$$

6. Compute the objective function, F as
$$F\left(s, s', S, S'\right) = u\left(c_j, g_{ss}, \hat{l}_j\right) + \beta \sum_{z'\tau'|z,\tau} \pi\left(z', \tau | z, \tau\right) \tilde{V^j}\left(k', i, z', \tau'; S'\right)$$

   where $F\left(.\right)$ is described by a $n = nk * ni * ntau * nz$ by $m = ni$ matrix, since k' is linked to i grid

7. Linearly search the maximum of F, to find $\tilde{V}^{j+1}(k, i_-, z, \tau; S)$

$$\tilde{V}^{j+1}(k, i_-, z, \tau; S) = \max_{k',i} F(s, s', S, S')$$

and obtain the policies

$$\left(g_i^j, g_k^j\right) = \arg\max_{k',i} F(s, s', S, S')$$

8. With the policies compute consumption c_{j+1}, labor l_{j+1} for the next iteration

$$c_{j+1} = (1-\tau) w_j l_j + r_j k - g_i^j$$

$$l_{j+1} = \left(\frac{1}{c_{j+1}}(1-\tau) w_j\right)^{\frac{1}{\gamma}}$$

update wages and rental rate with firms FOC

$$w_{j+1} = (1-\alpha)\tilde{z}k^{\alpha} l_{j+1}^{-\alpha}$$
$$r_{j+1} = \alpha\tilde{z}k^{\alpha-1} l_{j+1}^{1-\alpha}$$

and compute government consumption

$$g_{j+1} = \tau w_{j+1} l_{j+1}$$

9. Check convergence

$$\max\begin{pmatrix} \left|\left|\tilde{V}^{j+1} - \tilde{V}^{j}\right|\right| \\ \left|\left|w^{j+1} - w^{j}\right|\right| \\ \left|\left|r^{j+1} - r^{j}\right|\right| \\ \left|\left|w^{j+1} - w^{j}\right|\right| \end{pmatrix} \le tol$$

if the criterion holds, store the policies, values and allocations. If the criterion does not hold, repeat the steps by setting

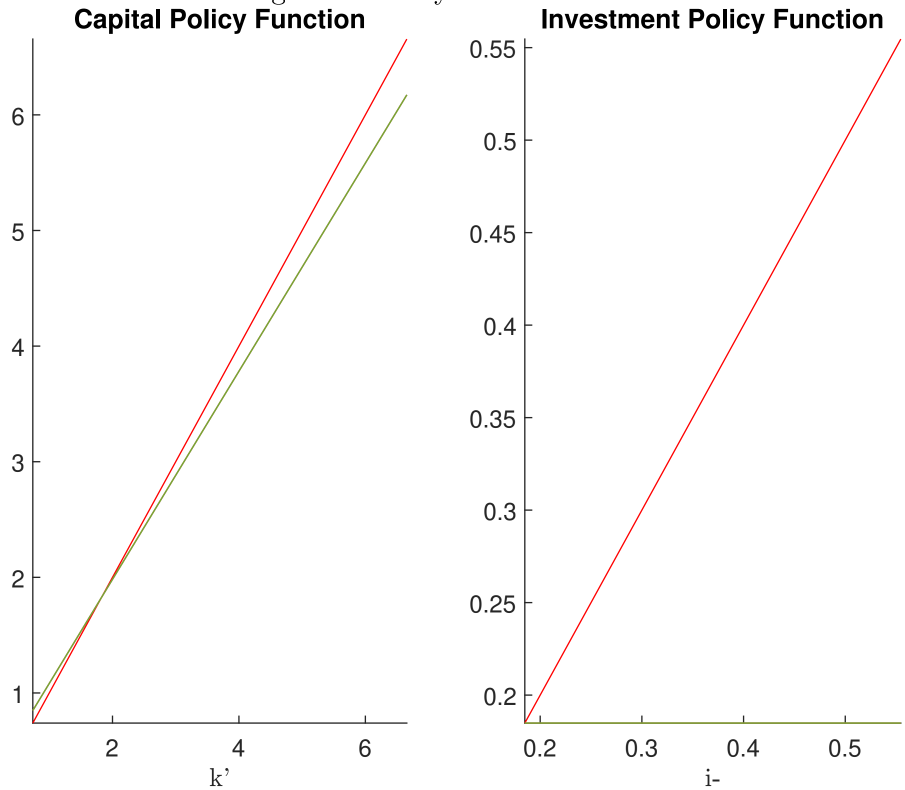
$$\tilde{V}^{j+1} = \rho\tilde{V}^{j}$$
$$w^{j+1} = \rho w^{j}$$
$$r^{j+1} = \rho r^{j}$$
$$w^{j+1} = \rho w^{j}$$

6.3.1 Results

For computational reasons, we implement $n^i = n^k = 25$ nodes. After obtaining convergence, policy function for investment takes the minimum value for every realization of the states (k, i_-, z, τ).

And the impulse responses for both shocks (either negative productivity or increase in taxes), behave accordingly, decreasing in impact investment, which raises consumption in impact but in the dynamics decrease consumption. After investment reaches the minimum value, the dynamics of the remaining variables converge to a new value.

Figure 1: Policy Functions



6.4 Switching between value and policy

For this exercise we compute value function iteration for $n^i = n^k = 20$ and the exogenous states, and compare the required time to find the solution between our baseline VFI algorithm and one in which we switch between value and policy iteration.

Table 14: Baseline VFI vs. Switching

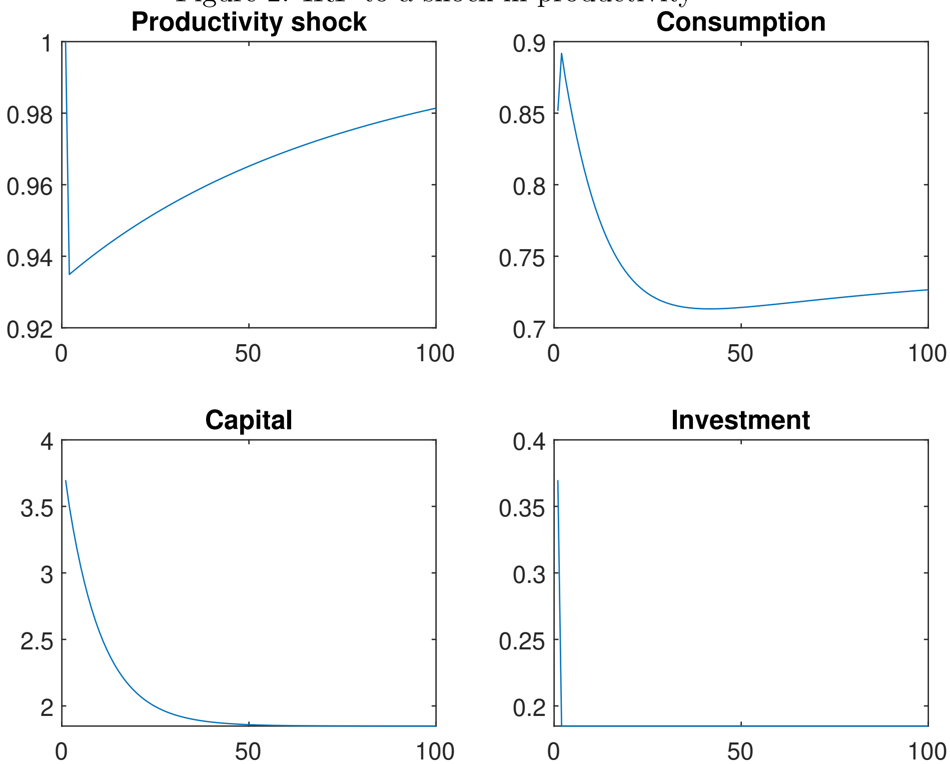
	Baseline	**Switching**
Time (secs)	126	2.44
Iterations	1705	41

The gains in computational speed are notorious. The switching algorithm is n times faster than our standard VFI.

6.5 Multigrid

For this exercise we compute value function iteration for $n^i = 25, n^k = 200$ and the exogenous states, and compare the required time to find the solution between multigrid for $n^k = \{10, 25, 50, 100, 200\}$ and our baseline VFI algorithm with $n^k = 200$ and a guess of value function equal to zero for

Figure 2: IRF to a shock in productivity



all states $\tilde{V}^0(k, i_-, z, \tau; S) = 0$. We compute both results alternating between VFI and PFI. We compare the time to build the exercise of $n^k = 200$ and the total computation time

Table 15: Baseline VFI vs. Multigrid

	Baseline	**Multigrid**
Time $n^k = 200$ **(secs)**	15	29
Total time (secs)	29	29

The computational gain is not considerable using all the nodes, however, as we increase the number of nodes, the multigrid algorithm outpaces the baseline VFI.

Figure 3: IRF to a shock in Taxes