HOMEWORK 4. AIYAGARI MODEL

HUAYAN GENG

Q1. Firm's Problem

The firm's problem

$$\begin{aligned}\max_{\{K^d_{t+1},N^d_t\}} \quad & \sum_{t=0}^{\infty} \quad \left(\frac{1}{\prod_{i=0}^{r} r_i}\right)\left(F(K_t,N_t)-\omega_t N_t - r_t K_t + (1-\delta)K_t\right) \\ \text{s.t.} \quad F(K_t,N_t) \quad & = K_t^{\alpha} N_t^{1-\alpha}\end{aligned}$$

The largrange equation is

$$\mathscr{L} \quad = \quad \sum_{t=0}^{\infty}\left(\frac{1}{\prod_{i=0}^{r} r_i}\right)\left(K_t^{\alpha}N_t^{1-\alpha}-\omega_t N_t - r_t K_t + (1-\delta)K_t\right)$$

To find the first-order conditions, take first derivative of $\mathscr{L}$ w.r.t. K_{t+1} and N_t and set as zero

$$\begin{aligned}\frac{\partial \mathscr{L}}{\partial K_{t+1}} &= \frac{1}{\prod_{i=0}^{r} r_i}\left(\alpha K_{t+1}^{\alpha-1}N_{t+1}^{1-\alpha} - r_{t+1} + (1-\delta)\right) \\ \frac{\partial \mathscr{L}}{\partial N_t} &= \frac{1}{\prod_{i=0}^{r} r_i}\left((1-\alpha)K_t^{\alpha}N_t^{-\alpha}-\omega_t\right)\end{aligned}$$

The factor prices given K and N are

$$\begin{aligned}r_{t+1} &= \alpha K_{t+1}^{\alpha-1}N_{t+1}^{1-\alpha} + (1-\delta) \\ \omega_t &= (1-\alpha)K_t^{\alpha}N_t^{-\alpha}\end{aligned}$$

Q2. Household's recursive problem

The households' recursive problem is

$$v(z,a) \quad = \quad \max_{a'\in\Gamma(z,a)}\left\{\frac{\left(z\omega\bar{l}+ra-a'\right)^{1-\sigma}}{1-\sigma}+\beta\mathbb{E}_{s'|s}\left[v(z',a')\right]\right\}$$

where assets bond $a_{t+1} \geq \underline{a}$, $l_t = \bar{l} = 1$, z is the worker's labor productivity.

Q3. Exogenous productivity grids

Given z as $m = 5$ possible values $z = \begin{bmatrix} 0.5002 & 0.7072 & 1.0000 & 1.4140 & 1.9993 \end{bmatrix}$, the corresponding transition matrix over prodctivity is $\begin{bmatrix} 0.1932 & 0.6135 & 0.1886 & 0.0047 & 7.4512\times10^{-6} \\ 0.0416 & 0.4584 & 0.4584 & 0.0414 & 2.6600\times10^{-4} \\ 0.0047 & 0.1886 & 0.6135 & 0.1889 & 0.0047 \\ 2.6600\times10^{-4} & 0.0414 & 0.4584 & 0.4584 & 0.0416 \\ 7.4512\times10^{-6} & 0.0047 & 0.1886 & 0.6135 & 0.1932 \end{bmatrix}$.

The invariant distribution of the Markov process over productivity states $\pi^{inv}(z) = [\ 0.0145 \quad 0.2189 \quad 0.5333 \quad 0$

The aggregate labor supply is $N^s = \int_z a\bar{l}\pi^{inv}dz = 1.0338 = N^d$.

Date: 11/08/2017.

Q4. Endogenous assets grids

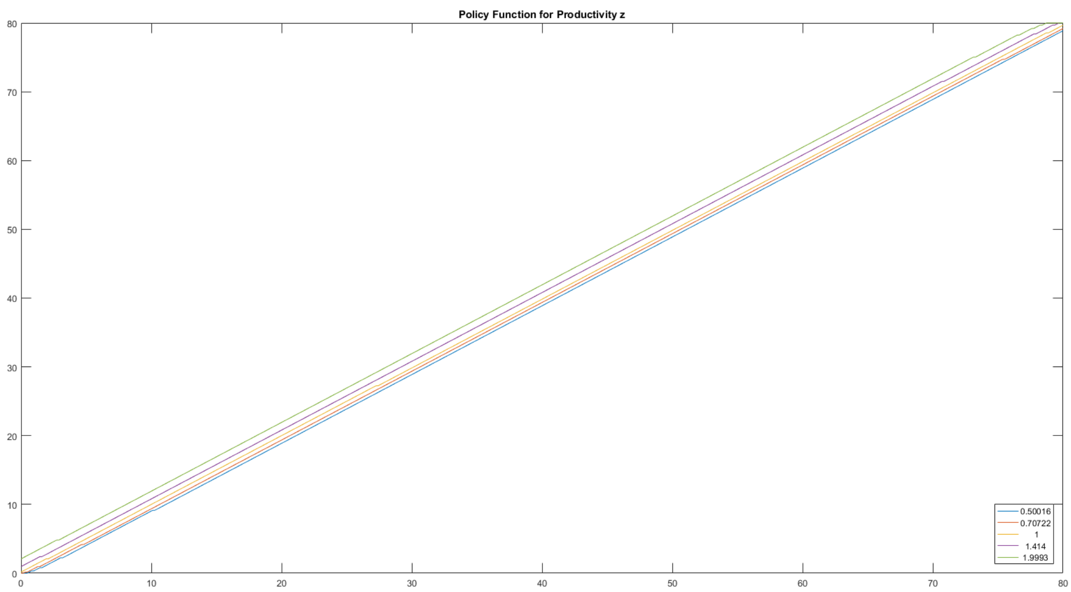
Assume a grid with $n = 500$ points. Given $\underline{a} = 0$, I guess a_{max} and manually adjust $a_{max} = 80$. $K_{min=20}$ and $K_{max} = 50$.

Q6. Analysis the results

The steady state interest rate is 1.0099 (or net 0.99%) which is slightly lower than complete markets steady state interest rate $r^{CM} = 1/\beta = 1/0.9900 = 1.0101$ (or net 1.01%) .

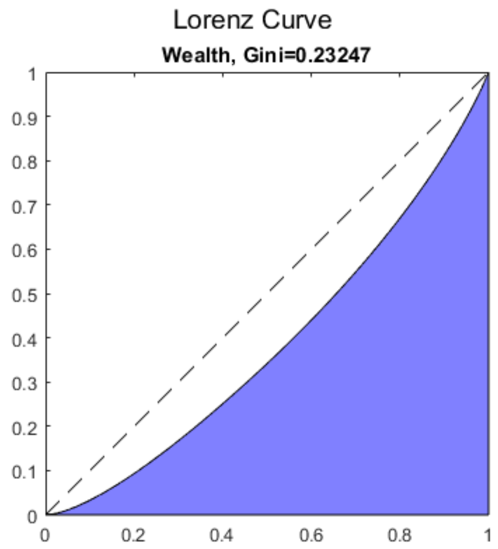
The policy functions for each productivity states are plotted in Figure 1

Figure 1. Policy Function



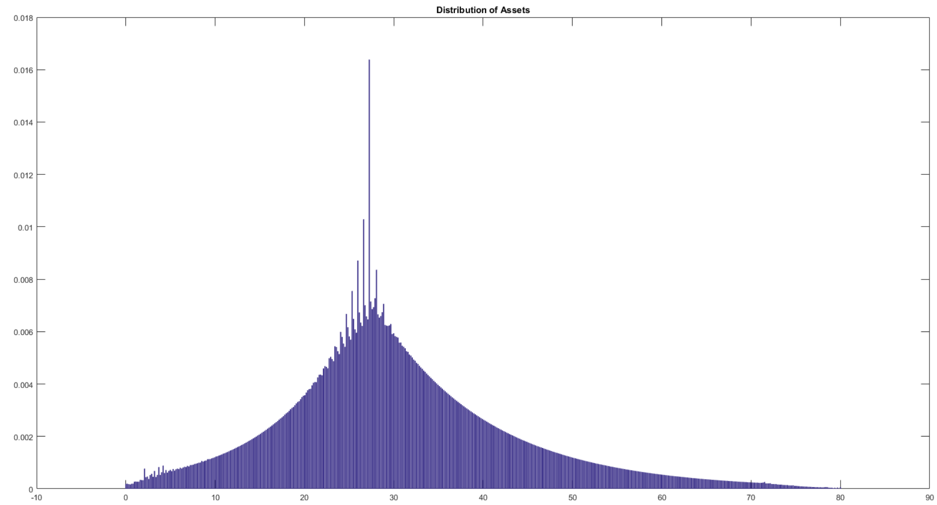
The Gini of Aiyagari model is 0.23247 and Lorenz curve is plotted in Figure 2

Figure 2. Lorenz and Gini

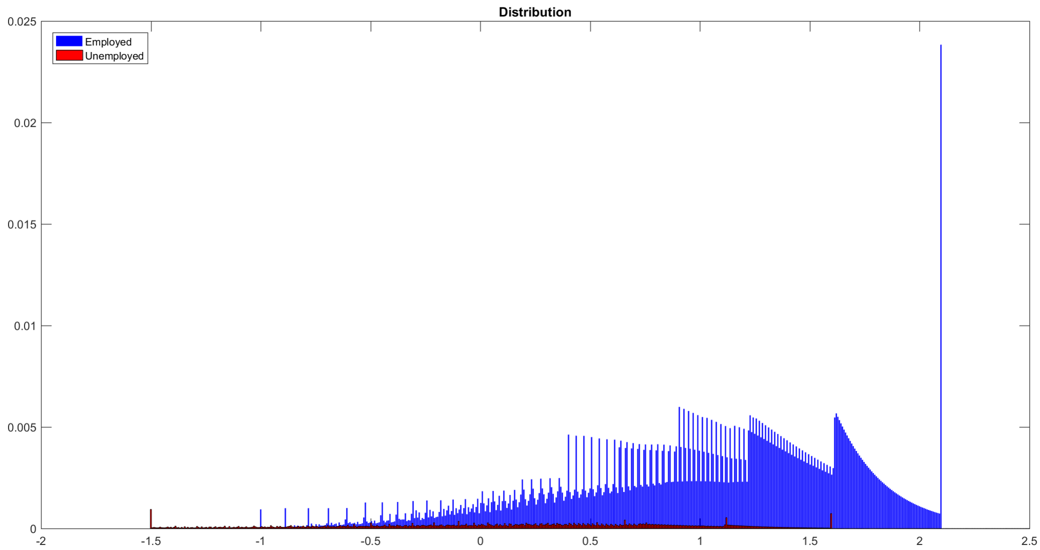


The wealth distribution of assets comparing to the Huggett model, the distribution form Aiyagari is more like the empirical distribution

FIGURE 3. Wealth Distribution



(A) Assets Distribution from Aiyagari Model



(B) Assets Distribution from Huggett Model

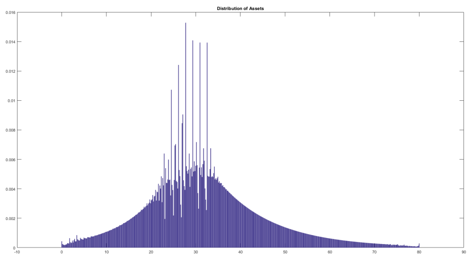
Q7. Alternative ways

Coarse Grid. Using VFI results from 50 grids points of a, the wealth distribution and lorenz curve are little bit different.

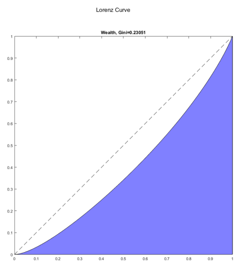
Policy function iteration. Use $k = 30$ policy iterations for each optimization step. The results is the same as grid search.

Linear interpolation. I am still working on questions related to interpolation, I hope I can update it during the weekends.

FIGURE 4. Results from Coarse Grid



(A) Wealth Distribution



(B) Lorenz Curve of Coarse Grid

Time tracking and error. Track the time of value function iteration and policy function iteration

FIGURE 5. Time Tracking

Lines where the most time was spent

Line Number	Code	Calls	Total Time	% Time	Time Plot
54	`value_mat=ret+beta*repmat(perm...`	28559	148.706 s	48.7%	
82	`(PI(z_ind(ii), :)*mass(ii))';`	110315000	76.312 s	25.0%	
81	`MuNew(:, apr_ind) = MuNew(:, a...`	110315000	46.101 s	15.1%	
56	`[vfn, pol_indx] = max(value_ma...`	28559	23.284 s	7.6%	
79	`apr_ind = pol_indx(z_ind(ii),a...`	110315000	4.364 s	1.4%	
All other lines			6.548 s	2.1%	
Totals			305.316 s	100%	

(A) Value Function Iteration (with Euler Equation Error 0.0050)

Lines where the most time was spent

Line Number	Code	Calls	Total Time	% Time	Time Plot
102	`(PI(z_ind(ii), :)*mass(ii))';`	110342500	78.963 s	44.3%	
101	`MuNew(:, apr_ind) = MuNew(:, a...`	110342500	47.221 s	26.5%	
65	`Q = makeQmatrix(pol_indx, PI);`	943	34.486 s	19.3%	
54	`value_mat=ret+beta*repmat(perm...`	943	5.025 s	2.8%	
99	`apr_ind = pol_indx(z_ind(ii),a...`	110342500	4.346 s	2.4%	
All other lines			8.358 s	4.7%	
Totals			178.398 s	100%	

(B) Policy Function Iteration (with Euler Equation Error 0.0050)

Line Number	Code	Calls	Total Time	% Time	Time Plot
98	`(PI(z_ind(ii), :)*mass(ii))';`	105990000	75.258 s	56.2%	
97	`MuNew(:, apr_ind) = MuNew(:, a...`	105990000	46.457 s	34.7%	
95	`apr_ind = pol_indx(z_ind(ii),a...`	105990000	4.349 s	3.2%	
100	`end`	105990000	2.760 s	2.1%	
56	`value_mat=ret+beta*repmat(perm...`	28671	1.191 s	0.9%	
All other lines			3.935 s	2.9%	
Totals			133.950 s	100%	

(C) Coarse Grid