Advanced Macroeconomics II

Handout 8 - Heterogeneous Agent Models

Sergio Ocampo

Western University

November 18, 2020

Heterogeneous agent models

- We want an economy with a continuum of agents

Heterogeneous agent models

- We want an economy with a continuum of agents
- Agents differ in their states:
  - Some are in debt, some have savings
  - Some have high income, some have low income
  - Some are young, some are old
  - Some own housing, some rent (some houses are larger than others)

Heterogeneous agent models

- We want an economy with a continuum of agents
- Agents differ in their states:
  - Some are in debt, some have savings
  - Some have high income, some have low income
  - Some are young, some are old
  - Some own housing, some rent (some houses are larger than others)
- Differences among agents are endogenous:
  - Arise from endogenous reaction to realization of shocks

Heterogeneous agent models

- We want an economy with a continuum of agents
- Agents differ in their states:
  - Some are in debt, some have savings
  - Some have high income, some have low income
  - Some are young, some are old
  - Some own housing, some rent (some houses are larger than others)
- Differences among agents are endogenous:
  - Arise from endogenous reaction to realization of shocks
- Agents interact in markets: Prices are endogenous
  - Labor market, capital market, housing market, etc
  - Demand and supply come from aggregating individual actions

What constitutes a solution?

1. Individual decision rules (policy functions)
    - How do agents behave across different states
    - We get these by solving agents' DP problem (VFI+EGM)

What constitutes a solution?

1. Individual decision rules (policy functions)
   - How do agents behave across different states
   - We get these by solving agents' DP problem (VFI+EGM)
2. Distribution of agents across states
   - How many rich agents are there? How many high-income agents?
   - This is new... and time consuming... more on it later

What constitutes a solution?

1. Individual decision rules (policy functions)
    - How do agents behave across different states
    - We get these by solving agents' DP problem (VFI+EGM)
2. Distribution of agents across states
    - How many rich agents are there? How many high-income agents?
    - This is new... and time consuming... more on it later
3. Prices
    - Need to be consistent with market clearing
    - Demand and supply come from aggregating individual actions (1) with respect to the distribution (2)
    - Solution depends on market structure

Stationary Recursive Competitive Equilibrium

- We will focus on models without aggregate uncertainty

Stationary Recursive Competitive Equilibrium

- We will focus on models without aggregate uncertainty
- All shocks are idiosyncratic and uncorrelated across agents
  - Agents' lives change period by period
  - Each agent gets different realization of shocks, causing change
    - A given agent can have high income one period and low the next
  - But there are no coordinated changes, there is always the same measure of agents in each states

Stationary Recursive Competitive Equilibrium

- We will focus on models without aggregate uncertainty
- All shocks are idiosyncratic and uncorrelated across agents
  - Agents' lives change period by period
  - Each agent gets different realization of shocks, causing change
    - A given agent can have high income one period and low the next
  - But there are no coordinated changes, there is always the same measure of agents in each states
- This means that aggregates do not change!
  - In particular prices are time invariant
  - More importantly: Distribution of agents is time invariant (stationary)

Hugget (1993) Economy

The individual problem (simplest problem)

- Agents live forever and choose consumption/savings
- Income (ϵ) is stochastic and follows a Markov process P:
  - This is an endowment economy, or a Lucas tree economy
  - This is the only shock in the economy
  - Realizations of the shock are independent across agents

The individual problem (simplest problem)

- Agents live forever and choose consumption/savings
- Income (ϵ) is stochastic and follows a Markov process P:
  - This is an endowment economy, or a Lucas tree economy
  - This is the only shock in the economy
  - Realizations of the shock are independent across agents

Agent's Problem:

$$V(\epsilon, a) = \max_{\{c,a'\}} u(c) + \beta E\left[V\left(\epsilon', a'\right)|\epsilon\right]$$

$$\text{s.t. } c + a' = (1+r)a + \epsilon \qquad a' \geq \underline{a}$$

Euler equation:

$$u'(c) = \beta E\left[V_a\left(\epsilon', a'\right)|\epsilon\right] = \beta(1+r)E\left[u'\left((1+r)a' + \epsilon' - g_a\left(\epsilon', a'\right)\right)\right]$$

The natural borrowing constraint

The problem requires a borrowing constraint:

The natural borrowing constraint

The problem requires a borrowing constraint:

1. If $r < 0$ the problem is ill-defined without a constraint
   - Agent could get infinite debt and obtain infinite consumption
   - The borrowing constraint serves a no-ponzi condition

The natural borrowing constraint

The problem requires a borrowing constraint:

1. If $r < 0$ the problem is ill-defined without a constraint
   - Agent could get infinite debt and obtain infinite consumption
   - The borrowing constraint serves a no-ponzi condition
2. If $r > 0$ the non-negativity of consumption (or budget balance) imply a borrowing constraint
   - Future income is stochastic: could be very low
   - To ensure non-negative consumption it must be that $a \geq {}^{-\epsilon_{min}}/_{r}$
   - ${}^{-\epsilon_{min}}/_{r}$ is the natural borrowing constraint

The natural borrowing constraint

The problem requires a borrowing constraint:

1. If $r < 0$ the problem is ill-defined without a constraint
   - Agent could get infinite debt and obtain infinite consumption
   - The borrowing constraint serves a no-ponzi condition
2. If $r > 0$ the non-negativity of consumption (or budget balance) imply a borrowing constraint
   - Future income is stochastic: could be very low
   - To ensure non-negative consumption it must be that $a \geq {}^{-\epsilon_{min}}/_{r}$
   - ${}^{-\epsilon_{min}}/_{r}$ is the natural borrowing constraint

The constraint must satisfy:

$$\underline{a} \geq {}^{-\epsilon_{min}}/_{r}$$

What happens with savings?

This is a Hugget (1993) economy:

- Agents save in bonds
- There is no storage technology... only contracts

What happens with savings?

This is a Hugget (1993) economy:

- Agents save in bonds
- There is no storage technology... only contracts
- Bonds are in zero net supply!
  - One agent's savings is another agent's debt
  - If we had a representative agent there would be no trade!

What happens with savings?

This is a Hugget (1993) economy:

- Agents save in bonds
- There is no storage technology... only contracts
- Bonds are in zero net supply!
  - One agent's savings is another agent's debt
  - If we had a representative agent there would be no trade!
- Agents use bonds to smooth consumption:
  - Save if ϵ is high
  - Borrow if ϵ is low

What happens with savings?

This is a Hugget (1993) economy:

- Agents save in bonds
- There is no storage technology... only contracts
- Bonds are in zero net supply!
  - One agent's savings is another agent's debt
  - If we had a representative agent there would be no trade!
- Agents use bonds to smooth consumption:
  - Save if ϵ is high
  - Borrow if ϵ is low

Equilibrium: Integral over assets is zero

The distribution of agents

- Let $\overline{S}$ be the set of states and $[\underline{a}, \overline{a}]$ be the domain of assets
  - Easy to show that there exists a value $\overline{a} > 0$ such that $a \leq \overline{a}$ for all agents (in the limit)

The distribution of agents

- Let $\overline{S}$ be the set of states and $[\underline{a}, \overline{a}]$ be the domain of assets
  - Easy to show that there exists a value $\overline{a} > 0$ such that $a \leq \overline{a}$ for all agents (in the limit)
- Let $\mathcal{S}$ and $\mathcal{A}$ be σ-algebras over $\overline{S}$ and $[\underline{a}, \overline{a}]$ respectively
- The distribution of agents is a function $\Gamma : \mathcal{S} \times A \to [0, 1]$
  - Γ is measurable with respect to $\mathcal{S}$ and $\mathcal{A}$
  - Γ integrates to 1

The distribution of agents

- Let $\overline{S}$ be the set of states and $[\underline{a}, \overline{a}]$ be the domain of assets
  - Easy to show that there exists a value $\overline{a} > 0$ such that $a \leq \overline{a}$ for all agents (in the limit)
- Let $\mathcal{S}$ and $\mathcal{A}$ be σ-algebras over $\overline{S}$ and $[\underline{a}, \overline{a}]$ respectively
- The distribution of agents is a function $\Gamma : \mathcal{S} \times A \to [0, 1]$
  - Γ is measurable with respect to $\mathcal{S}$ and $\mathcal{A}$
  - Γ integrates to 1
- The distribution $\Gamma\left(S, A\right)$ answers the question:
  - What measure of agents have a state $s \in S \in \mathcal{S}$ and assets $a \in A \in \mathcal{A}$?

Updating the distribution

We can update the distribution by following the actions of agents

- Let $S \times A \in \mathcal{S} \times \mathcal{A}$ be a set in the σ-algebra
  - We want to know if there are agents coming into the set $S \times A$

Updating the distribution

We can update the distribution by following the actions of agents

- Let $S \times A \in \mathcal{S} \times \mathcal{A}$ be a set in the σ-algebra
  - We want to know if there are agents coming into the set $S \times A$

1. Define an indicator function to know if $a^{'} \in A$

$$g\left(\epsilon, a, A\right) = \begin{cases} 1 & \text{if } a^{'}\left(\epsilon, z\right) \in A \\ 0 & \text{otw} \end{cases}$$

Updating the distribution

We can update the distribution by following the actions of agents

- Let $S \times A \in \mathcal{S} \times \mathcal{A}$ be a set in the σ-algebra
  - We want to know if there are agents coming into the set $S \times A$

1. Define an indicator function to know if $a^{'} \in A$

$$g\left(\epsilon, a, A\right) = \begin{cases} 1 & \text{if } a^{'}\left(\epsilon, z\right) \in A \\ 0 & \text{otw} \end{cases}$$

2. From the Markov kernel of ϵ we also have the probability that $\epsilon^{'} \in S$:

$$\Pr\left(\epsilon^{'} \in S|\epsilon\right) = \int_{\epsilon^{'} \in S} P\left(\epsilon^{'}|\epsilon\right) d\epsilon$$

- Both functions depend on the initial state of the agent (ϵ, a)

Updating the distribution

Updating function

$$\Gamma^{'}\left(S,A\right)=\int_{S^{-}\in\mathcal{S}}\int_{A^{-}\in\mathcal{A}}\underbrace{g\left(\epsilon,a,A\right)\cdot\Pr\left(\epsilon^{'}\in S|\epsilon\right)}_{\text{Markov Kernel:}Q\left(\epsilon^{'}a^{'}|\epsilon,a\right)}\cdot\Gamma\left(\epsilon,a\right)da d\epsilon$$

- The updating of the distribution takes the form of the **adjoint Markov operator of Q**

Updating the distribution

Updating function

$$\Gamma^{'}(S,A)=\int_{S^{-}\in\mathcal{S}}\int_{A^{-}\in\mathcal{A}}\underbrace{g(\epsilon,a,A)\cdot\Pr\left(\epsilon^{'}\in S|\epsilon\right)}_{\text{Markov Kernel:}Q\left(\epsilon^{'}a^{'}|\epsilon,a\right)}\cdot\Gamma(\epsilon,a)\,dad\epsilon$$

- The updating of the distribution takes the form of the **adjoint Markov operator of Q**
- We are looking for a fixed point of the adjoint operator
  - Γ such that $\Gamma^{'}=\Gamma$
- If you are interested in the properties that guarantee a unique stationary distribution look at SLP or section 24 of the math camp notes

Updating the distribution

- When exogenous state is discrete updating is simpler
- This is often the case in practice
- Without loss we can have the σ-algebra be:
  $\mathcal{S} = \left\{ \{\epsilon_1\}, \ldots, \{\epsilon_n\}, \ldots, \{\epsilon_N\} \right\}$

Updating function

$$\Gamma^{'}\left(\epsilon^{'}, A\right) = \int_{S^{-} \in \mathcal{S}} \int_{A^{-} \in \mathcal{A}} g\left(\epsilon, a, A\right) \cdot P\left(\epsilon^{'} | \epsilon\right) \cdot \Gamma\left(\epsilon, a\right) da d\epsilon$$

Stationary RCE

A S-RCE is a set of a value function (V), policy function $\left(a^{'}\right)$, distribution (Γ), and price (r) such that:

1. Given r the value and policy functions solve the agent's problem
2. Given the policy function, Γ is a fixed point of the adjoint operator
3. Given the distribution and policy functions the market for bonds clears:

$$\underbrace{\int\int a^{'}(\epsilon, a)\cdot\Gamma(\epsilon, a)\, dadz}_{\text{Net Supply}} = 0 \longleftrightarrow \underbrace{\int\int c(\epsilon, a)\cdot\Gamma(\epsilon, a)\, dadz}_{\text{Demand for Goods}} = \underbrace{\int\int \epsilon\cdot\Gamma(\epsilon, a)\, dadz}_{\text{Supply of Goods}}$$

Algorithm: VFI with EGM

Algorithm 1: EGM for S-RCE problem

Function `EGM(`$V(\epsilon, a), \vec{a}, \vec{\epsilon}, r$*,parameters*`)`:

for *i=1:*n_e *# Income state (exogenous, current period)* **do**

for *j=1:*n_a *# Savings (endogenous, future period)* **do**

1. Expected value: $\mathbb{V} = \beta E\left[V\left(\epsilon', \vec{a}_j\right)|\vec{\epsilon}_i\right]$
2. Consumption from Euler: $u_c\left(\tilde{c}_{ij}\right)\right) = \mathbb{V}_a$

   Solve analytically for $\tilde{c}_{ij}$
3. Endo. assets: $\hat{a}_{ij} = \left(\tilde{c}_{ij} + \vec{a}_j - \vec{\epsilon}_i\right)/1 + r$
4. Update value at endogenous grid: $\hat{V}(\vec{\epsilon}_i, \hat{a}_{ij}) = u\left(\tilde{c}_{ij}\right)\right) + \mathbb{V}$

5. Interpolate to exogenous grid: V_new[i,:] = Interp($\hat{a}$, $\hat{V}$, $\vec{a}$)

Some comments

- This is the easiest DP problem we have had
  - No non-linear equations, no interpolation inside loop
  - We have to check for the borrowing constraint! And adjust accordingly

Some comments

- This is the easiest DP problem we have had
  - No non-linear equations, no interpolation inside loop
  - We have to check for the borrowing constraint! And adjust accordingly
- As discussed in Lecture 7 (RCE) adding labor choice does not complicate the algorithm much

Some comments

- This is the easiest DP problem we have had
  - No non-linear equations, no interpolation inside loop
  - We have to check for the borrowing constraint! And adjust accordingly
- As discussed in Lecture 7 (RCE) adding labor choice does not complicate the algorithm much
- You can also use the ECM of Maliar and Maliar, try it out!

Some comments

- This is the easiest DP problem we have had
  - No non-linear equations, no interpolation inside loop
  - We have to check for the borrowing constraint! And adjust accordingly
- As discussed in Lecture 7 (RCE) adding labor choice does not complicate the algorithm much
- You can also use the ECM of Maliar and Maliar, try it out!
- We pay for this simplicity with having to find the stationary distribution

Alternative: PFI with EGM

Algorithm 2: EGM for S-RCE problem

Function $\mathtt{EGM}(g_a(\epsilon, a), \vec{a}, \vec{\epsilon}, r, parameters)$:

for *i=1:*n_e *# Income state (exogenous, current period)* **do**

for *j=1:*n_a *# Savings (endogenous, future period)* **do**

1. RHS of Euler:
$RHS = \beta E\left[U\left((1+r)\vec{a}_j + w\vec{\epsilon}_i - g_a(epsilon, \vec{a}_j)\right) | \vec{\epsilon}_i\right]$
2. Consumption from Euler: $u_c\left(\tilde{c}_{ij}\right) = RHS$
Solve analytically for $\tilde{c}_{ij}$
3. Endo. assets: $\hat{a}_{ij} = \left(\tilde{c}_{ij} + \vec{a}_j - \vec{\epsilon}_i\right)/1 + r$

4. Check borrowing constraint: If $min(\hat{a}(\vec{\epsilon}_i)) > \underline{a}$ then agent chooses $a' = \underline{a}$ for $a \in [\underline{a}, min(\hat{a}(\vec{\epsilon}_i))]$
5. Interpolate consumption to exogenous grid and obtain implied a' from budget constraint.

Algorithm: The histogram method (Young, 2011)

- **Objective:** Find the stationary distribution

Algorithm: The histogram method (Young, 2011)

- **Objective:** Find the stationary distribution
- **Method:** Discretize the distribution
  - Instead of looking for a continuous density we look for frequencies in a histogram

Algorithm: The histogram method (Young, 2011)

- **Objective:** Find the stationary distribution
- **Method:** Discretize the distribution
  - Instead of looking for a continuous density we look for frequencies in a histogram
- **Inputs:** Transition matrix for exogenous states and policy function

Algorithm: The histogram method (Young, 2011)

- **Objective:** Find the stationary distribution
- **Method:** Discretize the distribution
  - Instead of looking for a continuous density we look for frequencies in a histogram
- **Inputs:** Transition matrix for exogenous states and policy function

Steps:

1. Get a fine grid for assets
2. Use (inverse) linear interpolation to map $a'(\epsilon, a)$ to the grid

Algorithm: The histogram method (Young, 2011)

Algorithm 3: Histogram Method - Transition matrix

Function $\texttt{EGM}(a^{'}(\epsilon, a), \vec{a}, \vec{\epsilon})$:

0. Generate fine grid for assets and interpolate policy function $a^{'}$ to it.

for *i=1:n_e # Income state* **do**

for *j=1:n_a # Assets* **do**

1. Identify $h^{\star}$ s.t. $a^{'}(\vec{\epsilon}_i, \vec{a}_j) \in [\vec{a}_{j^{\star}}, \vec{a}_{h^{\star}+1}]$
   1.1. Save the index for future use: $h(\vec{\epsilon}_i, \vec{a}_j) = h^{\star}$
2. Define weight on lower grid node: $\omega = \frac{a^{'}(\vec{\epsilon}_i, \vec{a}_j) - \vec{a}_{h^{\star}}}{\vec{a}_{h^{\star}+1} - \vec{a}_{h^{\star}}}$
3. Assign weights to transition function for assets:
   $$g(\vec{\epsilon}_i, \vec{a}_j, \vec{a}_{h^{\star}}) = \omega \qquad g(\vec{\epsilon}_i, \vec{a}_j, \vec{a}_{h^{\star}+1}) = 1 - \omega$$
4. All other entries for $(\vec{\epsilon}_i, \vec{a}_j)$ are zero so **g is sparse**.

5. Check that $g \in [0, 1]$ for all states.

Algorithm: The histogram method (Young, 2011)

- We now have a way to get the Transition matrix for assets on our grid
  - To get the index of assets on the grid we use inverse interpolation:
    - $h^{\star} = \text{floor}\left(\left(\frac{a^{'}(\epsilon,a)-\min(\vec{a})}{\max(\vec{a})-\min(\vec{a})}\right)^{\frac{1}{\theta_a}}(n_a-1)+1\right)$ if $a^{'}(\epsilon,a) \in [\min(\vec{a}), \max(\vec{a})]$
    - If not then either $h^{\star} = 1$ or $h^{\star} = n_a$

Algorithm: The histogram method (Young, 2011)

- We now have a way to get the Transition matrix for assets on our grid
  - To get the index of assets on the grid we use inverse interpolation:
    - $h^{\star} = \text{floor}\left(\left(\frac{a^{'}(\epsilon,a)-\min(\vec{a})}{\max(\vec{a})-\min(\vec{a})}\right)^{\frac{1}{\theta_a}}(n_a-1)+1\right)$ if $a^{'}(\epsilon,a) \in [\min(\vec{a}), \max(\vec{a})]$
    - If not then either $h^{\star} = 1$ or $h^{\star} = n_a$
- We have at least three options to get the stationary distribution:
  1. Loops
  2. Matrices
  3. Smart matrices

Algorithm: Stationary distribution with loops

Algorithm 4: Histogram Method - Updating distributions

Function `EGM`($\Gamma, g(\epsilon, a, a^{'}), P(\epsilon, \epsilon^{'})$):

- **for** *i=1:n_e # Income state* **do**
  - **for** *j=1:n_a # Assets* **do**
    - **for** *k=1:n_e # Future income state* **do**
      - $\Gamma^{'}(k, h(i,j)) = \Gamma^{'}(k, h(i,j)) + g(\vec{\epsilon_i}, \vec{a_j}) \cdot P(\vec{\epsilon_i}, \vec{\epsilon_k}) \cdot \Gamma(i,j)$
      - $\Gamma^{'}(k, h(i,j)+1) =$
        $\Gamma^{'}(k, h(i,j)+1) + (1 - g(\vec{\epsilon_i}, \vec{a_j})) \cdot P(\vec{\epsilon_i}, \vec{\epsilon_k}) \cdot \Gamma(i,j)$
- Repeat until convergence

Algorithm: Stationary distribution with matrices

The state space is now effectively discrete, so we can list them:

$$\vec{s} = \begin{bmatrix}(\vec{\epsilon}_1, \vec{a}_1) & \dots & (\vec{\epsilon}_{n_\epsilon}, \vec{a}_1) & (\vec{\epsilon}_1, \vec{a}_2) & \dots & (\vec{\epsilon}_{n_\epsilon}, \vec{a}_2) & \dots & (\vec{\epsilon}_1, \vec{a}_{n_a}) & \dots & (\vec{\epsilon}_{n_\epsilon}, \vec{a}_{n_a})\end{bmatrix}^T$$

can then get a transition matrix from $\vec{s}$ to $\vec{s}'$:

$$T = (P \otimes 1_{n_a \times n_a}) \odot G$$

where P is the transition matrix for the exogenous income process:

$$P = \begin{bmatrix} \Pr\left(\epsilon_1' | \epsilon_1\right) & \dots & \Pr\left(\epsilon_{n_e}' | \epsilon_1\right) \\ \vdots & \ddots & \\ \Pr\left(\epsilon_1' | \epsilon_{n_e}\right) & \dots & \Pr\left(\epsilon_{n_e}' | \epsilon_{n_e}\right) \end{bmatrix}$$

and $G = \underbrace{\left[g\left(\underbrace{\epsilon, a}_{s_i}, a'\right) \right]}_{\text{Size: } (n_\epsilon n_a) \times n_a} \otimes 1_{1 \times n_e}$ is the transition matrix for assets.

Algorithm: Stationary distribution with matrices

- Armed with a transition matrix T we can either:
  - Iterate on $\Gamma' = T^T \Gamma$ many times
  - Solve the eigenvalue problem

Algorithm: Stationary distribution with matrices

- Armed with a transition matrix T we can either:
  - Iterate on $\Gamma' = T^T \Gamma$ many times
  - Solve the eigenvalue problem
- Iteration wins because this matrix is likely to be too large to manage

Algorithm: Stationary distribution with matrices

- Armed with a transition matrix T we can either:
  - Iterate on $\Gamma' = T^T\Gamma$ many times
  - Solve the eigenvalue problem
- Iteration wins because this matrix is likely to be too large to manage
- However the matrix G (and thus T) is very sparse, you can use that to your advantage!

Algorithm: Stationary distribution with matrices

- Armed with a transition matrix T we can either:
  - Iterate on $\Gamma' = T^T \Gamma$ many times
  - Solve the eigenvalue problem
- Iteration wins because this matrix is likely to be too large to manage
- However the matrix G (and thus T) is very sparse, you can use that to your advantage!
- Despite sparseness these matrices are expensive to store and use
  - Lots of wasteful storage because of repeated versions of G and P

Algorithm: Smarter matrices (Tan, 2020)

Key idea: Take the distribution forward one dimension at a time

Algorithm: Smarter matrices (Tan, 2020)

Key idea: Take the distribution forward one dimension at a time

1. Define $G_{n_\epsilon n_a \times n_\epsilon n_a}$ that maps $(\epsilon, a) \to \left(\epsilon, a^{'}\right)$ (we will construct in in next slide)
2. Start from a distribution Γ, an $n_\epsilon n_a$ vector
3. Update only the savings decision: $\hat{\Gamma} = G^{'}\Gamma$
4. Reshape $\hat{\Gamma}$ into a $n_\epsilon \times n_a$ matrix (This matrix is interpreted as $\left(\epsilon, a^{'}\right)$)
5. Update exogenous state: $\Gamma^{'} = P^{'}\hat{\Gamma}$ (result is a $n_\epsilon \times n_a$ matrix)
6. Reshape into new distribution $\Gamma^{'}$

Algorithm: The histogram method (Tan, 2020)

Algorithm 5: Histogram Method - Transition matrix

Function $\texttt{EGM}(a^{'}(\epsilon, a), \vec{a}, \vec{\epsilon})$:

0. Generate fine grid for assets and interpolate policy function $a^{'}$ to it.

for *i=1:n_e # Income state* **do**

for *j=1:n_a # Assets* **do**

1. Identify $h^{\star}$ s.t. $a^{'}(\vec{\epsilon}_i, \vec{a}_j) \in [\vec{a}_{j^{\star}}, \vec{a}_{h^{\star}+1}]$
2. Define weight on lower grid node: $\omega = \frac{a^{'}(\vec{\epsilon}_i, \vec{a}_j) - \vec{a}_{h^{\star}}}{\vec{a}_{h^{\star}+1} - \vec{a}_{h^{\star}}}$
3. Assign weights to transition function for assets:
$$g(\vec{\epsilon}_i, \vec{a}_j, \vec{\epsilon}_i, \vec{a}_{h^{\star}}) = \omega \qquad g(\vec{\epsilon}_i, \vec{a}_j, \vec{\epsilon}_i, \vec{a}_{h^{\star}+1}) = 1 - \omega$$
The second ϵ is just a repeat of the same state
4. All other entries for $(\vec{\epsilon}_i, \vec{a}_j)$ are zero so **g is sparse**.

5. Check that $g \in [0, 1]$ for all states.

Algorithm: S-RCE

Algorithm 6: S-RCE Algorithm

input : Guess for price (r)
output: $V, a^{'}, \Gamma, r$

1. Solve the DP problem of the agent given r:
   $(V, a^{'}) = T(V; r)$ (a fixed point problem) ;
2. Find stationary distribution with histogram method ;
3. Check market clearing: $\sum_i \sum_j a^{'}(\vec{\epsilon}_i, \vec{a}_j) \cdot \Gamma(i, j)$;
4. Update prices to clear market
   Manually by tatonnement or with a Root finder ;
5. Repeat (1)-(4) until market clears ;

Some comments

- You can improve the code by using the insight of Howard's policy iteration algorithm

Some comments

- You can improve the code by using the insight of Howard's policy iteration algorithm
- You don't need to iterate on the distribution until convergence!
  - Waste to iterate until convergence without the "true" policy function

Some comments

- You can improve the code by using the insight of Howard's policy iteration algorithm
- You don't need to iterate on the distribution until convergence!
  - Waste to iterate until convergence without the "true" policy function
- Alternative:

  2'. Iterate forward the distribution N times

  - After you iterate N times you check market clearing and update prices, then get new policy functions
  - Doing this means you have to change the convergence criteria

  4'. Repeat (1),(2'),(3),(4) until convergence in Γ and market clearing

  - You should also dampen the updating of the distribution (no need to update fully if you are not using the right policy function)

Aiyagari (1994) Economy

The Aiyagari economy

Key difference: Move from an endowment to a production economy

The Aiyagari economy

Key difference: Move from an endowment to a production economy

- Aiyagari reconciles the heterogeneous agent framework of Bewley-Hugget with the workhorse model of macro (the NGM)

The Aiyagari economy

Key difference: Move from an endowment to a production economy

- Aiyagari reconciles the heterogeneous agent framework of Bewley-Hugget with the workhorse model of macro (the NGM)
- Production is just as in NGM:
  - Representative firm with CRS technology
  - This will greatly simplify market clearing!

The Aiyagari economy

Key difference: Move from an endowment to a production economy

- Aiyagari reconciles the heterogeneous agent framework of Bewley-Hugget with the workhorse model of macro (the NGM)
- Production is just as in NGM:
  - Representative firm with CRS technology
  - This will greatly simplify market clearing!
- Reinterpretation of the shock to individual agents
  - Shock is now labor efficiency instead of income endowment
  - This does not really change the problem

Individual problem

- Income depends on labor efficiency (ϵ) and the market wage (w)
  - ϵ is stochastic and follows a Markov process P
  - This is the only shock in the economy
  - Realizations of the shock are independent across agents

Individual problem

- Income depends on labor efficiency (ϵ) and the market wage (w)
  - ϵ is stochastic and follows a Markov process P
  - This is the only shock in the economy
  - Realizations of the shock are independent across agents

Agent's Problem:

$$V(\epsilon, a) = \max_{\{c,a'\}} u(c) + \beta E\left[V\left(\epsilon', a'\right)|\epsilon\right]$$

$$\text{s.t. } c + a' = (1+r)a + w\epsilon \qquad a' \geq \underline{a}$$

Euler equation:

$$u'((1+r)a + w\epsilon) = \beta E\left[V_a\left(\epsilon', a'\right)|\epsilon\right]$$

Production and savings

Output is produced from capital and labor: $Y = F(K, L)$

Production and savings

Output is produced from capital and labor: $Y = F(K, L)$

- **Key:** F has constant returns to scale
  - Demand for inputs is perfectly elastic at equilibrium prices
  - The only way to clear the market is to make producers indifferent

Production and savings

Output is produced from capital and labor: $Y = F(K, L)$

- **Key:** F has constant returns to scale
  - Demand for inputs is perfectly elastic at equilibrium prices
  - The only way to clear the market is to make producers indifferent
- **Equilibrium prices**

$$r = F_k(K, L) - \delta \qquad w = F_L(K, L)$$

where the aggregate capital and labor come from distribution:

$$K = \int\int a \cdot \Gamma(\epsilon, a)\, da d\epsilon \qquad L = \int\int \epsilon \cdot \Gamma(\epsilon, a)\, da d\epsilon$$

But L supply is exogenous! So we can say: $L = \int \epsilon \Gamma_\epsilon(\epsilon)\, d\epsilon = 1$

Changes to RCE

Algorithm 7: S-RCE Algorithm

input : Guess for price (r)
output: $V, a^{'}, \Gamma, r$

1. Solve the DP problem of the agent given (r, w):
   $(V, a^{'}) = T(V; r)$ (a fixed point problem) ;
2. Find stationary distribution with histogram method
Alternatively update distribution N times ;
3. Update prices to ensure market clearing:
   $K = \sum_i \sum_j a \cdot \Gamma(i, j) \quad \longrightarrow \quad r = F_k(K, 1) - \delta \quad w = F_L(K, 1)$;
   3.1. Dampen updating of prices if necessary ;
4. Repeat (1)-(3) until prices converge ;

OLG Aiyagari Economy

Life cycle: OLG

- Many applications in macro-labor require life cycle dynamics with heterogeneity

Life cycle: OLG

- Many applications in macro-labor require life cycle dynamics with heterogeneity
- That implies two changes:
  1. A new state: Age
  2. A new solution method for the agent's problem: Backward induction

Life cycle: OLG

- Many applications in macro-labor require life cycle dynamics with heterogeneity
- That implies two changes:
  1. A new state: Age
  2. A new solution method for the agent's problem: Backward induction
- Age is a tricky state. It will increase the dimension of the problem. There is no way around it.

Life cycle: OLG

- Many applications in macro-labor require life cycle dynamics with heterogeneity
- That implies two changes:
  1. A new state: Age
  2. A new solution method for the agent's problem: Backward induction
- Age is a tricky state. It will increase the dimension of the problem. There is no way around it.
- Backward induction is a blessing! No need for value functions!
  - Main benefit: No more approximation of the derivative V_a!
  - Other benefit: No fixed point to solve agent's problem

Individual problem

Agent's Problem:

$$V^{h}\left(\epsilon,a\right)=\max_{\left\{c_{h},a_{h}^{'}\right\}}\ u\left(c_{h}\right)+\beta E\left[V^{h+1}\left(\epsilon^{'},a^{'}\right)|\epsilon\right]$$

$$\text{s.t. }\ c_{h}+a_{h}^{'}=\left(1+r\right)a+w\epsilon \qquad a_{h}^{'}\geq\underline{a}$$

- Age is a state (h), but we write it as an index

Individual problem

Agent's Problem:

$$V^h(\epsilon, a) = \max_{\left\{c_h, a_h^{'}\right\}} u(c_h) + \beta E\left[V^{h+1}\left(\epsilon^{'}, a^{'}\right)|\epsilon\right]$$

$$\text{s.t. } c_h + a_h^{'} = (1+r)\,a + w\epsilon \qquad a_h^{'} \geq \underline{a}$$

- Age is a state (h), but we write it as an index

Euler equation:

$$u^{'}(c_h(\epsilon, a)) = \beta E\left[V_a^{h+1}\left(\epsilon^{'}, a^{'}\right)|\epsilon\right]$$

- Pair with envelope condition to get rid of value function:

$$u^{'}(c_h(\epsilon, a)) = \beta E\left[u^{'}\left(c^{h+1}\left(\epsilon^{'}, a^{'}\right)\right)(1+r)\,|\epsilon\right]$$

OLG-RCE

A S-RCE is a set of a value function (V), policy function $\left(a^{'}\right)$, distribution (Γ), and prices (r, w) such that:

1. Given (r, w) the value and policy functions solve the agent's problem
   - Recall that both value and policy functions are indexed by age
2. Given policy functions, Γ is a fixed point of the adjoint operator
3. Given the distribution the input markets clear:

$$K = \sum_{h=1}^{H} \int \int a \cdot \Gamma^{h}(\epsilon, a)\, da d\epsilon \qquad L = \sum_{h=1}^{H} \int \int \epsilon \cdot \Gamma^{h}(\epsilon, a)\, da d\epsilon$$

$$r = F_{k}(K, L) - \delta \qquad w = F_{L}(K, L)$$

Algorithm: Backward induction

Algorithm 8: Backward Induction EGM for S-RCE problem

Function $\mathtt{EGM}(\vec{a}, \vec{\epsilon}, r, w, parameters)$:

0. Last period we apply terminal conditions

$$a^H(\epsilon, a) = 0 \text{ and } c^H = (1+r)a + w\epsilon$$

for *h=H-1:1 # Age, counting backwards* **do**

for *i=1:*n_e *# Income state (exogenous, current period)* **do**

for *j=1:*n_a *# Savings (endogenous, future period)* **do**

1. Consumption from Euler (Solve analytically for $\tilde{c}_{ij}^h$):

$$u_c\left(\tilde{c}_{ij}^h\right) = \beta E\left[u_c\left(\tilde{c}^{h+1}(\epsilon', \vec{a}_j)\right) | \vec{\epsilon}_i\right]$$

2. Endo. assets: $\hat{a}_{ij}^h = \left(\tilde{c}_{ij}^h + \vec{a}_j - w\vec{\epsilon}_i\right)/1 + r$

3. Interpolate **policy functions** to exogenous grid:

$$c^h[\mathrm{i,:}] = \mathrm{Ip}(\hat{a}^h, \tilde{c}^h, \vec{a}) \text{ and } a^h[i,:] = (1+r)\vec{a} + w\vec{\epsilon}_i - c^h[i,:]$$

Some comments

- The solution is found in one iteration of the algorithm
  - No fixed point involved

Some comments

- The solution is found in one iteration of the algorithm
  - No fixed point involved
- Cost comes with age! Another loop to go through

Some comments

- The solution is found in one iteration of the algorithm
  - No fixed point involved
- Cost comes with age! Another loop to go through
- After interpolation we need to check for borrowing constraint or negative consumption

Some comments

- The solution is found in one iteration of the algorithm
  - No fixed point involved
- Cost comes with age! Another loop to go through
- After interpolation we need to check for borrowing constraint or negative consumption
- The problem is quite flexible
  - Terminal conditions can involve bequests
  - The agent's problem can change with age (working/retirement)

Stationary distribution

- Finding the distribution is not conceptually different than before
- Practical difference: We need to index transition function g by age
- Implementation requires additional outer loop that goes through age
  - It is better to have this loop go backwards in age
  - Distribution of agents of age H maps into new-borns:
    $\Gamma^{H}(\epsilon, a) \longrightarrow \Gamma^{1}\left(\epsilon^{'}, a^{'}\right)$
  - Distribution of agents of age $H-1$ maps into age H:
    $\Gamma^{H-1}(\epsilon, a) \longrightarrow \Gamma^{H}\left(\epsilon^{'}, a^{'}\right)$

Algorithm: S-RCE

Algorithm 9: S-RCE Algorithm

input : Guess for price (r)
output: $V, a^{'}, \Gamma, r$

1. Solve the DP problem of the agent given (r, w):
   Single iteration of Backwards Induction ;
2. Find stationary distribution with histogram method
Alternatively update distribution N times ;
3. Update prices to ensure market clearing:
   $$K = \sum_h \sum_i \sum_j a \cdot \Gamma^h(i,j) \quad \longrightarrow \quad r = F_k(K,1) - \delta \quad w = F_L(K,1) \;;$$
   3.1. Dampen updating of prices if necessary ;
4. Repeat (1)-(3) until prices converge ;

Final comments

Parallel programming and simulation

- Parallel programming is necessary for large-scale models

Parallel programming and simulation

- Parallel programming is necessary for large-scale models
- Julia has built-in parallel options
  - Multithreading is the way to start
  - Multithreading distributes tasks across processors
  - Easiest to implement for loops
  - Loops are the bottleneck of the code

Parallel programming and simulation

- Parallel programming is necessary for large-scale models
- Julia has built-in parallel options
  - Multithreading is the way to start
  - Multithreading distributes tasks across processors
  - Easiest to implement for loops
  - Loops are the bottleneck of the code
- Many results come from simulation
  - Need to simulate draws from the exogenous Markov Process
  - Check code for simulation of MP in Lecture 5
  - Simulation is parallelizable!