Computational Methods for Macroeconomics (Part II)
Lecture 1: A heterogeneous household model with incomplete market

Hanbaek Lee

University of Cambridge

February 28, 2022

Intro

- Instructor: Hanbaek Lee
- Field: Macroeconomics and Finance
- Email: hl610@cam.ac.uk
- Office hour: Monday 10:00 - 11:00 am @ Robinson #93
  - If it conflicts with your schedule, please shoot me an email.
- Coverage:
  - A heterogenous household model with incomplete market: **Aiyagari (1994)**
  - A heterogenous household model with incomplete market and aggregate risk: **Krusell and Smith (1998)**
  - A heterogeneous household model in continuous time: **Moll et al. (2021)**
  - Heterogeneous firm models: **Khan and Thomas (2008) and Winberry (2021)**

Goal of this lecture

- Precise understanding of the model's mechanism
  - more focus on the mechanism + computation, but less on the economics
  - Each lecture starts from a model introduction
  - The first lecture more on the model (as it is shared in other papers)
- Guiding you to replicate the main computational results of key papers
- Understanding strengths and weaknesses of each computational method
- Learning techniques to boost the speed and accuracy of your computation
- Application to your model
- All the lecture materials will be based on Matlab.

Background: Heterogeneous agent model

- Irrelevance of micro-level heterogeneity in macro
  - Caplin and Spulber (1987)
- In the end, it is all about (non-)linearity in the demand or supply curve.
  - If the aggregate outcome is not different from the representative-agent model, there is no reason for using HA.
  - We will come back to this point in the second lecture.
- Computational hurdle: an infinite-dimensional object as a state variable
- Rise of non-linear HA models
  - Fernandez-Villaverde et al. (2011), Bloom et al. (2018), Fernandez-Villaverde et al. (2019), Carvalho and Grassi (2019), Winberry (2021), Lee (2021a)
  - How to solve? Very tricky question: Lee (2021b)
- Rise of rich micro-level data
  - Rich data disciplines HA models to match diverse micro-level patterns.
- Rise of HANK model
  - THANK model (Bilbiie, 2021) develops an analytically tractable HANK.

Background: Aiyagari (1994)

- Arrow-Debreu complete market: perfect insurance through a state-contingent claim
- A stark wealth dispersion under incomplete market: Bewley (1977, 1983) and Huggett (1993)
- Aiyagari (1994) studies heterogeneous agents under the incomplete market in general equilibrium.
- Basic ingredients:
  - Continuum of households
  - Time is discrete and lasts forever
  - No aggregate uncertainty
  - Idiosyncratic labor income shock: not insurable (incomplete market)
  - A borrowing constraint
  - A representative firm

Model

Environment

- Preference:

$$\mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t u(c_{it})$$

- Budget constraint:

$$c_{it} + a_{it+1} = w_t z_{it} + (1 + r_t) a_{it}$$

- Initial condition:

$$z_{i0}, a_{i0} \geq 0.$$

- Borrowing constraint (precautionary motivation $\uparrow$):

$$a_{it+1} \geq \underline{a}$$

Idiosyncratic income process

- Stochastic income process for each household i: $\{z_{it}\}_{t=0}^{\infty}$

$$z_{it} \in \mathcal{Z} = \{z_1, z_2, \ldots, z_N\}$$

- A Markov process with transitions: $\Gamma_{z,z'} \geq 0$
- Depending on the transition matrix $\Gamma = [\Gamma_{z,z'}]$, the income process can have a stationary distribution or not.
  - The stationary distribution Φ_z satisfies $\Gamma' \Phi_z = \Phi_z$.
  - Φ_z is the eigenvector of Γ with the largest eigenvalue (=1).
  - If a Markov chain is irreducible and aperiodic, then there is a unique stationary distribution.
  - A non-stochastic simulation method of Young (2010): endogenous policies can be expressed in a transition matrix. So it's all about the eigenvector! We will come back to this method.
- Aggregate labor supply: $L^S = \sum_{i=1}^{N} \Phi_z(z_i) z_i$ (No endogenous labor supply decision)

A representative firm

- A representative firm (perfect competition + frictionless) with Cobb-Douglas production function:

$$\max_{K_t, L_t} \quad F(K_t, L_t; A_t) - w_t L_t - (r_t + \delta) K_t$$

$$\text{where } F(K_t, L_t; A_t) = A_t K_t^{\alpha} L_t^{1-\alpha}$$

  - Bewley-Huggett models are abstract from the production sector.
  - In Aiyagari (1994), $A_t = 1$.
  - Krusell and Smith (1998), A_t follows a Markov chain.
  - Zero profit (dividend) to be distributed.
- There is no state-contingent claim:
  - Household's wealth (fixed from the last period) is aggregated as a current capital.
  - Incomplete market
- Aggregate resource constraint: $C_t + K_{t+1} - (1 - \delta) K_t = F(K_t, L_t)$

Recursive form

A household begins a period with wealth a and labor productvity z given.

$$v(a,z;\Phi) = \max_{c,a'} u(c) + \beta \sum_{z'} \Gamma_{z,z'} v(a',z';\Phi')$$

$$\begin{aligned} \text{s.t.} \quad & c + a' = w(\Phi)z + a(1+r(\Phi)) \\ & a' \geq \underline{a} \\ & z' \sim \Gamma(z'|z) \quad \text{(Markov chain)} \\ & \Phi' = G(\Phi) \end{aligned}$$

- Q) Why do households need to understand Φ?
- There is no state-contigent contract available: incomplete market.
- cf.) Arrow-Debreu economy:

$$c + \sum_{z'} \Gamma_{z,z'} q(z',\Phi') a'(a,z;z',\Phi') = w(\Phi)z + a(1+r(\Phi))$$

The optimality conditions

$$\mathcal{L} = u(c) + \beta \mathbb{E} v(a', z'; \Phi') + \lambda(w(\Phi)z + a(1 + r(\Phi)) - c - a') + \mu(a' - \underline{a})$$

First-order condition :

$$[c] : u'(c(a, z; \Phi)) = \lambda$$

$$[a'] : \lambda = \beta \mathbb{E} v_1(a'(a, z), z'; \Phi') + \mu$$

Envelope condition :

$$[a] : v_1(a, z; \Phi) = \lambda(1 + r(\Phi))$$

The role of borrowing constraint I

$$\begin{aligned} u'(c(a,z;\Phi)) = \lambda &= \beta\mathbb{E}v_1(a'(a,z),z';\Phi') + \mu \\ &= \beta\mathbb{E}\lambda'(1+r(\Phi')) + \mu \\ &= \beta\mathbb{E}(\beta\mathbb{E}\lambda''(1+r(\Phi'')) + \underbrace{\mu'}_{\text{Possibe future binding state}})(1+r(\Phi')) + \mu \end{aligned}$$

- $\mu \geq 0$ shifts up $\lambda \iff$ as $u'' < 0$, μ shifts down c
- Even if $\mu = 0$, possible future state of binding constraint ($\mu' \geq 0$) shifts down c.
- Therefore, the borrowing constraint increases the saving of households through precautionary motivation.

Natural borrowing limit

- How do we know $\mu > 0$ happens?
- If $\underline{a}$ is extremely low, the borrowing constraint might be meaningless (never binding).
- Non-negative consumption $c_t \geq 0$ and $\lim_{t\to\infty} a_t/(1+r)^t < \infty$ gives a natural borrowing limit, $\underline{a}^N$. which sharply identifies the lowest meaningful constraint:

$$c_t + a_{t+1} = wz_t + a_t(1+r)$$

$$\frac{1}{(1+r)}(c_{t+1} + a_{t+2}) = \frac{1}{(1+r)}(wz_{t+1} + a_{t+1}(1+r))$$

$$\dots$$

$$\implies c_t + \frac{c_{t+1}}{(1+r)} + \frac{c_{t+2}}{(1+r)^2} + \dots = a_t(1+r) + w\left(z_t + \frac{z_{t+1}}{(1+r)} + \frac{z_{t+2}}{(1+r)^2} + \dots\right)$$

$$0 \leq a_t(1+r) + w\left(z_{min} + \frac{z_{min}}{(1+r)} + \frac{z_{min}}{(1+r)^2} + \dots\right)$$

$$\underline{a}^N := -\frac{wz_{min}}{r} \leq a_t$$

- If $\underline{a} < \underline{a}^N$, then $\mu = 0$. Given $l_{min} \geq 0$ and $r > 0$, $a_t \geq \underline{a} = 0$ is always a relevant constraint.

The role of borrowing constraint II

Define $\overline{z}(a;\Phi)$ as the level of income shock where

$$a'(a,\overline{z}(a;\Phi);\Phi) = \underline{a} \text{ and } \mu(a,\overline{z}(a;\Phi);\Phi) = 0$$

- This is the nife-edge case (uncontrained optimal choice of $\underline{a}$).
- For $\forall z < \overline{z}(a;\Phi)$, $a'(a,z;\Phi) = \underline{a}$
- Consider two household 1 and 2 with $z_1 < z_2 < \overline{z}(a;\Phi)$ and the same wealth a.
  - Income gap $= (z_2 - z_1)w(\Phi)$
  - Consumption gap $= c(a,z_2;\Phi) - c(a,z_1;\Phi) = (z_2 - z_1)w(\Phi) =$ Income gap
  - MPC $= 1$ for constraint-binding households: poor hand-to-mouth
  - Kaplan and Violante (2014) points out wealthy hand-to-mouth: Two individual endogenous states (liquid and illiquid assets)

Money in hand and allocations

Aiyagari (1994) defines money in hand z_t and modified asset holding $\widehat{a}_t$ as follows:

$$\widehat{a}_t := a_t - \underline{a} \geq 0$$

$$z_t := wl_t + \widehat{a}_t(1+r) + r\underline{a} \quad \text{(Sorry about the notation)}$$

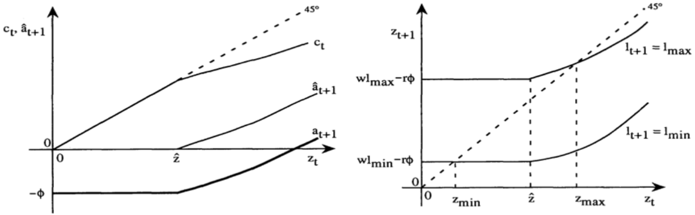


FIGURE Ia
Consumption and Assets as Functions of Total Resources

FIGURE Ib
Evolution of Total Resources

Notes: The figure is from Aiyagari (1994).

The role of incomplete market

- Consider an economy where you can trade a state-contingent asset $a'(z';\Phi')$. What would be the difference?
- In the Arrow-Debreu economy, the perfect insurance is guaranteed:

$$c_t^i = \overline{c} > 0. \text{ for } \forall t, i$$
$$\implies \Delta a' = w(\Phi)\Delta z$$

- In incomplete market model,

$$\Delta c + \Delta a' = w(\Phi)\Delta z.$$

  - Strong prudence: tendency to save more from precautionary motivation
  - Strong risk aversion: tendency to save less due to uninsured risk
- The Arrow-Debreu economy is not empirically supported (Mace, 1991).
- The Arrow-Debreu model with uniform initial wealth = A representative-agent model

Recursive competitive equilibrium

$(g_a, g_c, v, G, g_K, g_{a,L}, r, w)$ are recursive competitive equilibrium if

- (g_a, g_c, v) solves household's problem.
- $(g_K, g_{a,L})$ solves a representative firms' problem.
- (r, w) clears capital and labor markets.

$$\text{(Capital market)} \quad g_K(\Phi) = \int a d\Phi \quad \text{at } r = r(\Phi)$$

$$\text{(Labor market)} \quad g_{a,L}(\Phi) = \sum_{i=1}^{N} \Phi_z(z_i) z_i \quad \text{at } w = w(\Phi)$$

- $\int (g_c(a, z; \Phi) + g_a(a, z; \Phi)) \, d\Phi = F(g_K(\Phi), g_{a,L}(\Phi)) + (1 - \delta) g_K(\Phi)$
- $G(\Phi) = \Phi'$ holds, where for $\forall \mathcal{A}' \subseteq \mathbb{A}$, $\Phi'(\mathcal{A}', z') = \int \Gamma_{z,z'} \mathbb{I}\{g_a(a, z) \in \mathcal{A}\} d\Phi$ where $\mathbb{A}$ is the Borel σ-algebra generated from $[\underline{a}, \infty)$.

Note: The equilibrium allows $\Phi' \neq \Phi$: perfect foresight transition paths included. If the productivity is stochastic, all the equilibrium allocations are stochastic (KS, 1998).

Stationary recursive competitive equilibrium

$(g_a, g_c, v, G, g_K, g_{a,L}, r, w, \Phi)$ are stationary recursive competitive equilibrium if

- (g_a, g_c, v) solves household's problem given Φ (or given r, w).
- $(g_K, g_{a,L})$ solves a representative firms' problem given Φ (or given r, w).
- (r, w) clears capital and labor markets.

$$\text{(Capital market)} \quad g_K(\Phi) = \int a d\Phi \quad \text{at } r$$

$$\text{(Labor market)} \quad g_{a,L}(\Phi) = \sum_{i=1}^{N} \Phi_z(z_i) z_i \quad \text{at } w$$

- $\int (g_c(a,z) + g_a(a,z))\, d\Phi = F(g_K, g_{a,L}) + (1-\delta) g_K$
- $G(\Phi) = \Phi$ holds, where for $\forall \mathcal{A} \subseteq \mathbb{A}$, $\Phi(\mathcal{A}, z') = \int \Gamma_{z,z'} \mathbb{I}\{g_a(a,z) \in \mathcal{A}\} d\Phi$, where $\mathbb{A}$ is the Borel σ-algebra generated from $[\underline{a}, \infty)$.

Note: The equilibrium requires $\Phi' = \Phi$: The fixed point in terms of the distribution.

Price characterization in SRCE (RCE)

In the competitive equilibrium, the marginal productivity of an input is equal to the price:

$$r(\Phi) = MPK(\Phi) - \delta = \alpha \left(\frac{K(\Phi)}{L} \right)^{\alpha - 1} - \delta$$

$$w(\Phi) = MLK(\Phi) = (1 - \alpha) \left(\frac{K(\Phi)}{L} \right)^{\alpha}$$

where $L = \sum_{i=1}^{N} \Phi_z(z_i) z_i$ is exogenously given. From the first order conditions above we get:

$$w(\Phi) = (r(\Phi) + \delta)^{\frac{\alpha}{\alpha - 1}} \alpha^{\frac{\alpha}{1 - \alpha}} (1 - \alpha)$$

Therefore, once $r(\Phi)$ is known, all the equilibrium allocations are obtained (Walras' law). Similarly, we can say once $K(\Phi)$ is known, all the prices (w, r) are obtained.

- Initial guess on either K or r (or w) until the market clears: a single external loop in SRCE. Then, how about in RCE?

Existence and uniqueness in SRCE I

- From the previous characterization, we have found that for given K, the corresponding prices (w, r) are obtained, and, thus, allocations are obtained (partial equilibrium).
- In general equilibrium, the allocations should be consistent with K.

$$K_1 \to g_a(a, z; K_1) \to K_2 \to g_a(a, z; K_2) \to \cdots \to K^* \tag{1}$$

- 1) Will there be such K^* always? 2) If so, will it be unique? 3) Why do we care?
  - 3-1) It will take forever (and fail) for your computer if the solution does not exist.
  - 3-2) Depending on the initial guess, your computation outcomes are different equilibria.
  - 3) Practically, if it is well-known that a representative (similar) model *has unique* equilibrium, the similar nature is inherited to your model, so it is usually fine.
  - In 1) and 2), strong monotonicity is the key for the proof (in sequence (1)).

Existence and uniqueness in SRCE II

- There are two separate issues in existence:
  (a) The existence of stationary distribution: A fixed point of measure
  (b) The existence of stationary recursive competitive equilibrium: A fixed point of price
- (b) cannot be achieved without (a).
- (a) with continuous wealth and productivity is studied in Hopenhayn and Prescott (1988).
- (a) with discretized space as an approximation is guaranteed when the discretized policy matrix (transition matrix) is i) irreducible and ii) aperiodic.
- The intuition behind (b) is as follows:
  - $\int ad\Phi$ is continuous and strictly decreasing (increasing) in K (in r) in a moderate range.
  - There is $\overline{K}$ such that $\overline{K} > \int ad\Phi$.
  - There is $\underline{K}$ such that $\underline{K} < \int ad\Phi$. $\implies$ IVT gives the unique equilibrium K^*!
- Within a moderate range of K, the strong monotonicity of $\int ad\Phi$ gives uniqueness. However, this is not guaranteed for parameters in extreme ranges when equilibrium price is also in an extreme range.

Computation - SRCE

Roadmap

- Now let's suppose we are guaranteed existence and uniqueness.
- Computation is about getting the equilibrium allocations in numbers.
- We can compute SRCE, but we cannot fully compute RCE (functions): RCE is mostly somehow approximated, or we only get equilibrium outcomes (not equilibrium itself).
- The following is the roadmap for the computation section.
  1. Price (aggregate allocation) guess
  2. Solution (optimization)
     - VFI/PFI/EGM/Projection method
     - VFI with acceleration
     - Interpolation
  3. Simulation
     - Stochastic simulation
     - Non-stochastic simulation: Iteration method
     - Non-stochastic simulation: Eigenvector method
  4. Aggregation
  5. Price update

Basic setup

All the codes start from the following steps.

- Set directories
- Set parameters (might be a function argument)

Then, two important steps follow.

- Setting grid points
  - Two sets of wealth grid points: before/after interpolation
  - Finer grids for smaller wealth (Maliar, Maliar, and Valli, 2010)
- Discretizing idiosyncratic shock process (Markov chain)
  - Tauchen method
  - Rouwenhorst method

Grid setup

```
%wealth grid
pNumGrida  = 50;
pNumGrida2 = 300; %finer grid for interpolation

%finer grid near lower wealth (Maliar, Maliar, and Valli, 2010)
vGridamin  = 0;
vGridamax  = 30;
x          = linspace(0,0.5,pNumGrida);
x2         = linspace(0,0.5,pNumGrida2);
y          = x.^5/max(x.^5);
y2         = x2.^5/max(x2.^5);
vGrida     = vGridamin+(vGridamax-vGridamin)*y;
vGrida2    = vGridamin+(vGridamax-vGridamin)*y2;
```

Idiosyncratic income process I

For $ln(z_{t+1}) = \rho ln(z_t) + \sigma\epsilon_{t+1}, \quad \epsilon_{t+1} \sim N(0,1)$,

```
pNumGridz  = 13;
[mTransz, vGridz] = fnTauchen(pRrho, 0, pSsigma^2, pNumGridz, 5);
vGridz = exp(vGridz');
```

Or we can manually assume a Markov chain matrix:

```
%idiosyncratic income shock
%simply 10% separation and 66% labor participation
%no distiction between unemp and non-participation
pNumGridz  = 2;
vGridz     = [0.0001,1];
mTransz    = [0.80,0.20;...
              0.10,0.90];
```

Idiosyncratic income process II

The stationary distribution of labor income shock, *vL*:

```
[vL,~] = eigs(mTransz',1);
vL = vL/sum(vL);
```

Then, aggregate labor supply is exogenously given as:

```
supplyL = vGridz*vL;
```

Price (aggregate allocation) guess

- First, we need to set what are **sufficient statistics** to characterize the equilibrium.
- This often comes from some analytical derivations from the model.
- In Aiyagari (1994), K. We can choose r or w as well.
- It could be multiple allocations (total externality, total credit amount, etc.). It depends on the models. Usual candidates are the first moments (or median) of a dimension.
- SRCE needs to specify only a ceratin number in $\mathbb{R}$ for a guess.
- RCE needs to specify some structure (functional form) on the allocations.
- In Aiyagari (1994), we only care SRCE.
- Instead of constantly updating the price guess, one can consider a grid of price candidates and compute the excess demand at each price level.
  - Computationally extremely costly as too many irrelevant price candidates.
  - If the price grid is fine enough, the equilibrium can be sharply pinned down after interpolation.
  - Easy to parallelize. Still too costly, so recommended for the first check for existence. (crossing zero?)

Computation - GE: Solution

Solution (optimization)

This step is not different from any representative agent models with some stochastic shocks.

- VFI: utilize monotonicity + accelerator (Howard improvement)
  - Policy functions converge faster than value functions
- PFI (Time iteration): fast and accurate (next lecture for Krusell and Smith (1998))
- EGM: fast and accurate
  - Limited when a value function has non-differentiable points; Difficult to accommodate aggregate uncertainty
- Projection method: accurate and Smolyak algorithm makes codes run fast

Value function iteration with accelerator

The pseudo code is as follows:

1. Guess $V^{(n)}$
2. Solve for the policy function, $g_a^{(n)}$ (using monotonicity: $g_a^{(n)} \geq g_{\widetilde{a}}^{(n)}$ for $a \geq \widetilde{a}$).
   - Interpolation is needed.
3. Update $V^{(n+1)}$ using the policy function, $g_a^{(n)}$.
4. Update $V^{(n+2)}$ using the policy function, $g_a^{(n)}$.

   ...
5. Update $V^{(n+m)}$ using the policy function, $g_a^{(n)}$.
6. Check if $||V^{(n+m)} - V^{(n+m-1)}||_p < Tol$
   - If yes, the solution converged.
   - If no, go back to step 1.

The key is less frequently updating the policy function. The typical bottleneck is in the step 2, and accelerator visits step 2 less frequently while giving a convergence (c.f., profile on; profile viewer in Matlab): Let me do a demonstration.

Optimizer selection

- If we use the numerical derivative for the unconstrained optimum, the computation is more efficient:
  $$u'(a(1+r)+wl-a') = \beta\mathbb{E}_z V_1(a',z')$$
- However, if V features high curvature, the accuracy is not high (for low a).
- A conventional maximum (minimum) finder works better in terms of accuracy for low a (golden section search).
- Therefore, we can increase the computation speed by letting the code runs golden-section search for low a and using the numerical derivative for high a.

```
%optimal saving decision
if ia < thrhdOptimizer || minWealth == vGridamin
  aprime = fnOptFAST(pBbeta,budget,vGrida,expVal,pNumGrida,minWealth);
else
  aprime = fnOptFASTEuler(pBbeta,budget,vGrida,expVal,pNumGrida);
end
```

Interpolation

- There are two necessary interpolation steps in the entire codes:
  - Solution
    - Your future wealth stock a' may not necessarily be on the grid you chose.
    - Then, value function (policy function) needs to be computed for all continuous $a' \in (0, a_{max}]$
  - Simulation (next section)
- There are multiple options depending on the number of dimensions and the nature of the function.
  - Manual linear interpolation (extrapolation) (Fast!)
  - Unidimension: *interp*1 (linear, cubic, spline, etc.)
  - Multidimension + rectangular grid: *interpn* (linear, cubic, spline, etc.) [c.f., Khan and Thomas (2013), Kaplan and Violante (2014)]
  - 2D (or 3D) + flexible grid: Delaunay Triangulation (linear, cubic, spline, etc.) (e.g. capital (illiquid) + liquidity)

Interpolation: Delaunay Triangulation

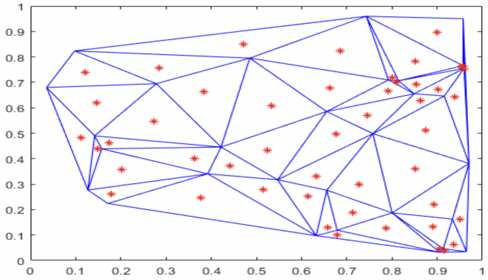


Figure: Delaunay Triangulation

Notes: The figure is from MathWorks.

- Compared to the interpolation based on rectangular, DT is computationally more efficient.

Interpolation: Manual interpolation

- Find the closest grid locations (*aLow* and *aHigh*) to the target point (*aprime*).
- Caculate the weight on each grid based on the distance from the target point.
- Obtain the linear combination of the values at each grid point.

```
%linear interpolation for off-the-grid aprime
aLow = sum(vGrida<aprime);
aLow(aLow<=1) = 1;
aLow(aLow>=pNumGrida) = pNumGrida-1;
aHigh = aLow+1;
weightLow = (vGrida(aHigh) - aprime)/(vGrida(aHigh)-vGrida(aLow));
value = weightLow*expVal(aLow)+(1-weightLow)*expVal(aHigh);
```

Computation - GE: Simulation

Stochastic simulation

For the aggregate allocations (macroeconomic outcomes), micro-level agents are simulated based on the policy functions obtained from the solution section.

1. Set the number of agents N and length of history T.
2. Set the initial distribution of individual state variables.
   - Thanks to the law of large numbers, after burn-in periods, the distribution would converge to the stationary distribution.
   - But time till convergence sharply depends on the initial distribution setup.
3. Simulate the idiosyncratic shock path of T periods for N agents. (Monte Carlo simulation)
4. Given $s_{it} = (a_{it}, z_{it})$, using $g_a(a_{it}, z_{it})$ determine a_{it+1}, and combine it with the simulated z_{it+1} to get $s_{it+1} = (a_{it+1}, z_{it+1})$.
5. As a result, we have an $N \times T$ wealth history matrix and an $N \times T$ shock history matrix.
6. Any other allocations (e.g., c_{it}) can be obtained by using these state history matrices.

Interpolation (revisited)

In the stochastic simulation, there are two steps where interpolations are necessary.

5. Given $s_{it} = (a_{it}, z_{it})$, using $g_a(a_{it}, z_{it})$ determine a_{it+1}, and combine it with the simulated z_{it+1} to get $s_{it+1} = (a_{it+1}, z_{it+1})$.
   - Issue: $g_a(a_{it}, z_{it}) = a_{it+1}$ does not have to be on the grid of wealth. Then, for period $t+2$, how can we determine $a_{it+2} = g_a(a_{it+1}, z_{it+1})$?
   - $g_a(a_{it+1}, z_{it+1})$ is obtained by interpolating $g_a(aLow, z_{it+1})$ and $g_a(aHigh, z_{it+1})$,

     where $aLow$ and $aHigh$ are the closest locations on the grid to the target point (a_{t+1}).

7. Any other allocations (e.g., c_{it}) can be obtained by using these state history matrices.
   - Issue: $g_a(a_{it}, z_{it}) = a_{it+1}$ does not have to be on the grid of wealth. Then, for period $t+1$, how can we determine $c_{it+1} = g_c(a_{it+1}, z_{it+1})$?
   - $g_c(a_{it+1}, z_{it+1})$ is obtained by interpolating $g_c(aLow, z_{it+1})$ and $g_c(aHigh, z_{it+1})$

Intro to non-stochastic simulation

- Stochastic simulation is subject to a sampling error.
- Young (2010) suggests a histogram-based non-stochastic simulation.
- Histogram is an object, $\mathcal{H}_t(a,z)$ such that $\sum_{a,z} \mathcal{H}_t(a,z) = 1$.
- No index in (a,z): the grids are fixed, and the histogram moves around (indexed by t).
- Non-stochastic simulation is often computationally more efficient:
  - In stochastic simulation, agents i and j with the same state (a,z) are taking two rows in simulation.
  - In non-stochastic simulation, agents i and j with the same state (a,z) are counted by measure and taking one row in the simulation.
- For tracking the life-cycle aspect of an agent (firm), stochastic simulation is better. In non-stochastic simulation, there is no individual agent. Only the states and the weight (histogram) are used.
- Two approaches are available:
  - Iteration method: For GE problem, fast (advantage of initial guess)
  - Eigenvector method: Accurate and fast

Non-stochastic simulation: Iteration method

1. Set the initial weights (histogram) on the individual states.
   - Thanks to the law of large numbers, after burn-in periods, the distribution would converge to the stationary distribution.
   - But time till convergence sharply depends on the initial distribution setup.
   - The previous GE loop's converged histogram is a very good initial guess for the current loop's stationary distribution.
2. Given $\mathcal{H}_t$, using g_a and Γ determine $\mathcal{H}_{t+1}(a,z)$ for each (a,z) as follows:

$$\mathcal{H}_{t+1}(a,z) = \sum_{\widetilde{z}} \sum_{\widetilde{a}} \mathbb{I}\{g(\widetilde{a},\widetilde{z}) = a\}\Gamma_{\widetilde{z},z}\mathcal{H}_t(\widetilde{a},\widetilde{z})$$

   - Q) How do we deal with the case where $g(\widetilde{a},\widetilde{z})$ is not on the grid? Linearly split the weight (next slide).
3. Repeat step 1-2, until $||\mathcal{H}_{t+1} - \mathcal{H}_t||_p < Tol$
4. Any other allocations are already computed in the previous step for (a,z).

Non-stochastic simulation: Iteration method - interpolation

The unresolved question is

- Q) How do we deal with the case where $g(\tilde{a}, \tilde{z})$ not on the grid? Linearly split the weight:

$$\mathcal{H}_{t+1}(a,z) = \sum_{\tilde{z}} \sum_{\tilde{a}} \mathbb{I}\{g_{a,L}(\tilde{a},\tilde{z}) = a\} \Gamma_{\tilde{z},z} \mathcal{H}_t(\tilde{a},\tilde{z}) \omega_L(\tilde{a},\tilde{z})$$
$$+ \sum_{\tilde{z}} \sum_{\tilde{a}} \mathbb{I}\{g_{a,H}(\tilde{a},\tilde{z}) = a\} \Gamma_{\tilde{z},z} \mathcal{H}_t(\tilde{a},\tilde{z}) (1 - \omega_L(\tilde{a},\tilde{z}))$$

where $g_{a,L}(\tilde{a}, \tilde{z})$ is the closest on-the-grid point smaller than $g_a(\tilde{a}, \tilde{z})$, and $g_{a,H}(\tilde{a}, \tilde{z})$ is the closest on-the-grid point greater than $g_a(\tilde{a}, \tilde{z})$.

$$w_L(\tilde{a},\tilde{z}) = \frac{g_{a,H}(\tilde{a},\tilde{z}) - a}{g_{a,H}(\tilde{a},\tilde{z}) - g_{a,L}(\tilde{a},\tilde{z})}$$

$w_L(\tilde{a}, \tilde{z})$ is the linear interpolation weight on $g_{a,L}(\tilde{a}, \tilde{z})$.

Non-stochastic simulation: Iteration method - interpolation (cont'd)

```
a = vGrida2(ia);
nexta = mPolaprime2(ia,iz);
LB = sum(vGrida2<nexta);
LB(LB<=0) = 1; LB(LB>=pNumGrida2) = pNumGrida2-1;
UB = LB+1;
weightLB = (vGrida2(UB) - nexta)/(vGrida2(UB)-vGrida2(LB));
weightLB(weightLB<0) = 0;
weightLB(weightLB>1) = 1;
weightUB = 1-weightLB;
mass = currentDist(ia,iz);
for futureiz = 1:pNumGridz
    nextDist(LB,futureiz) = nextDist(LB,futureiz)...
        +mass*mTransz(iz,futureiz)*weightLB;
    nextDist(UB,futureiz) = nextDist(UB,futureiz)...
        +mass*mTransz(iz,futureiz)*weightUB;
end
```

Non-stochastic simulation: Eigenvector method

- This method utilizes that policy functions are large Markov transition matrices.
- The stationary distribution is an eigenvector associated with the eigenvalue of unity.
- The key is to construct a large transition matrix combining g_a and Γ.
- The pseudo code is as follows:
  1. Generate a combined grid of all individual states (a, z) using kronecker product, $\otimes$ (vectorize). Let's denote $s = (a, z)$.
  2. Construct a generalized transition matrix $G(s, s')$ as follows:

$$
\begin{aligned}
G(s, s') = \mathbb{I}\{g_{a,L}(s) = a'\}\Gamma_{z,z'}\omega_L(s) \\
+ \mathbb{I}\{g_{a,H}(s) = a'\}\Gamma_{z,z'}(1 - \omega_L(s))
\end{aligned}
$$

  3. Obtain the eigenvector Φ such that $1 * \Phi = G' * \Phi$
  4. Obtain the normalized distribution $\Phi^* = \Phi / \sum_s \Phi$ to satisfy $\sum_s \Phi^* = 1$.
  5. Rearrange Φ^* to a histogram $\mathcal{H}$ on the rectangular grid.
- Computing eigenvector takes non-trivial time. In GE loop, iteration method might be faster due to an advantage of good initial guess.

Non-stochastic simulation: Eigenvector method (cont'd)

```
a = vGrida2(ia);
nexta = mPolaprime2(ia,iz);
LB = sum(vGrida2<nexta);
LB(LB<=0) = 1; LB(LB>=pNumGrida2) = pNumGrida2-1;
UB = LB+1;
weightLB = (vGrida2(UB) - nexta)/(vGrida2(UB)-vGrida2(LB));
weightLB(weightLB<0) = 0; weightLB(weightLB>1) = 1;
weightUB = 1-weightLB;
for izprime = 1:pNumGridz
    mPolicy(iLocation,:) =
        mPolicy(iLocation,:)+(vLocCombineda==LB).*(vLocCombinedz==izprime) *
        weightLB * mTransz(iz,izprime);
    mPolicy(iLocation,:) =
        mPolicy(iLocation,:)+(vLocCombineda==UB).*(vLocCombinedz==izprime) *
        weightUB * mTransz(iz,izprime);
end
```

Computation - GE: Aggregation and price update

Aggregation

- In the stochastic simulation,

$$K^{update} = K_T = \sum_{i=1}^{N} a_{iT}/N$$

  - Even if T is large enough, $K_T, K_{T-1}, K_{T-2}, \ldots$ might be slightly different from each other: Sampling error.
  - Make sure to normalize by N: the total measure of the agents is unity.
- In the non-stochastic simulation,

$$K^{update} = \sum_a a \sum_z \mathcal{H}(a,z)$$

- K^{update} might be different from the K we started from at the begining of the current iteration: Update! (next slide)

Update the price (outer loop)

- Bisection method:
  - Set a large enough upper bar $\overline{K}$ and a low enough lower bar $\underline{K}$.
  - Solve the problem using the guess, $K^{guess} = \frac{\overline{K}+\underline{K}}{2}$. From the aggregation get K^{update}.
  - Update the boundaries $(\overline{K}, \underline{K})$:
    - If $K^{update} > K^{guess}$, update $\overline{K} = K^{update}$.
    - If $K^{update} < K^{guess}$, update $\underline{K} = K^{update}$.
    - If $K^{update} = K^{guess}$, GE is solved.
- Linear combination:
  - $K^{new} = \omega K^{old} + (1-\omega)K^{update}$
  - Or, $log(K^{new}) = \omega log(K^{old}) + (1-\omega)log(K^{update})$
  - $\omega = 0.9$ gives slow but solid convergence.
  - $\omega = 0.6$ gives fast but unstable convergence.
  - High ω gets extra speed boost as the initial guess is closer to the solution for each iteration.

Aggregate demand and supply

- The aggregate capital supply (wealth) is greater in Aiyagari (1994) than in Arrow-Debreu economy: stronger precautionary saving motivation.
- Thus, $1 + r < 1/\beta$.

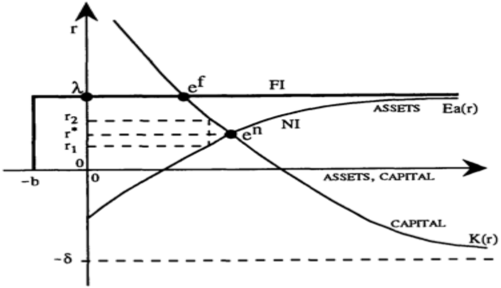


FIGURE IIb
Steady-State Determination

Notes: The figure is from Aiyagari (1994).

Parallelization

- Parallelization is a technique to let a computer use multiple cores (brains) to compute multiple tasks simultaneously to boost up the speed.
- Task independence is the basic necessary condition:
  - Value function iteration cannot be parallelized.
  - Stochastic simulation can be parallelized.
  - For each price, a partial equilibrium algorithm can be parallelized.
  - The method of simulated moments can be parallelized: Particle swarm algorithm
- A cheat key in Matlab: for→parfor. Let me do a demonstration.
- It is subject to an overhead computing cost: Not a free lunch.
- GPU is another type of parallelization: Use a graphic card (use gpuArray in Matlab)
- Recent processors have dramatically increased the number of cores (M1 chip)
- In each academic institution, there are server computers which allow a great number of cores: ask the IT team, or Amazon Web Services (AWS) is an alternative.
- In C/C++/Fortran, CUDA is the platform for parallelization.

Transitional Dynamics

Transitional dynamics (perfect foresight)

- Many heterogeneous agent models study how an economic environment change (parameter) affects the growth or transition of the economy (Buera and Shin, 2013)
  - A permanent change in productivity/elasticity of substitution/constraint
  - The change is unexpected for each agent: all stay in the steady-state just before the change (MIT shock).
- There are three large steps:
  1. Computation of the original steady-state in GE
  2. Computation of the new steady-state in GE
  3. Computation of the transitional dynamics in GE
- Along with the transitional dynamics, each agent fully understands where the economy is headed: perfect foresight.
- Thus, the algorithm repeatedly computes the backward solution and the forward simulation.
- Let me first define the transitional competitive equilibrium.

Transitional competitive equilibrium (model)

A household begins a period with wealth a and labor productvity z given.

$$v_t(a, z; \Phi_t) = \max_{c,a'} u(c) + \beta \sum_{z'} \Gamma_{z,z'} v_{t+1}(a, z'; \Phi_{t+1})$$

$$\begin{aligned} \text{s.t.} \quad & c + a' = w(\Phi_t)z + a(1 + r(\Phi_t)) \\ & a' \geq \underline{a} \\ & z' \sim \Gamma(z'|z) \quad \text{(Markov chain)} \\ & \Phi_{t+1} = G_t(\Phi_t) \end{aligned}$$

- Now, value function and the individual state distribution is indexed by the time, t.
- The policy functions are also indexed by t: $g_{a,t}$ and $g_{c,t}$.
- The production sector's policy does not have to be indexed by t: No history dependence in the production side: g_K and $g_{a,L}$.
- SRCE is a special case of TCE, where all the policy functions stay invariant over time.

Transitional competitive equilibrium

Given Φ_0 (the original SRCE distribution) and v_{T+1} (the new SRCE value function), $(g_{a,t}, g_{c,t}, v_t, G_t, r_t, w_t, \Phi_t)_{t=1}^{T}$ and $(g_K, g_{a,L})$ are transitional competitive equilibrium if

- $(g_{a,t}, g_{c,t}, v_t)_{t=1}^{T}$ solves household's problem given $(\Phi_t)_{t=1}^{T}$ (or given $(r_t, w_t)_{t=1}^{T}$).
- $(g_K, g_{a,L})$ solves a representative firms' problem given $(\Phi_t)_{t=1}^{T}$ (or given $(r_t, w_t)_{t=1}^{T}$).
- $(r_t, w_t)_{t=1}^{T}$ clears capital and labor markets of time $\forall t$.

$$\text{(Capital market)} \quad g_K(\Phi_t) = \int a d\Phi_t$$

$$\text{(Labor market)} \quad g_{a,L}(\Phi_t) = \sum_{i=1}^{N} \Phi_z(z_i) z_i$$

- $\int (g_{c,t}(a,z) + g_{a.t}(a,z))\, d\Phi_t = F(g_K, g_{a,L}) + (1-\delta) g_K$
- $G_t(\Phi_t) = \Phi_{t+1}$ holds, where for $\forall \mathcal{A} \subseteq \mathbb{A}$, $\Phi_{t+1}(\mathcal{A}, z') = \int \Gamma_{z,z'} \mathbb{I}\{g_{a,t}(a,z) \in \mathcal{A}\} d\Phi_t$, where $\mathbb{A}$ is the Borel σ-algebra generated from $[\underline{a}, \infty)$.

Computation - Transitional competitive equilibrium

1. Compute the original SRCE (Φ_0)
2. Compute the new SRCE (v_{T+1})
3. Compute TCE
   - 3.1 Set a long enough time T (e.g., 50 years).
   - 3.2 Guess the price path (aggregate allocation path): $(K_t^{guess})_{t=1}^T$.
   - 3.3 Given v_{T+1} and $(K_t^{guess})_{t=1}^T$, solve $(v_t, g_{a,t}, g_{c,t})_{t=1}^T$ using the finite backward iteration.
   - 3.4 Given Φ_0, simulate forward using $(g_{a,t})_{t=1}^T$ to obtain $(\Phi_t)_{t=1}^T$.
     - Either stochastic or non-stochastic methods work.
     - Eigenvalue method does not work (non-stationarity).
   - 3.5 Aggregate $(\Phi_t)_{t=1}^T$ to obtain $(K_t^{update})_{t=1}^T$.
   - 3.6 Check if $||K^{update} - K^{guess}||_p < Tol$
     - If yes, the solution converged.
     - If no, go back to step 3.2.

Note: From the original SRCE, we only need Φ_0. From the new SRCE, we only need v_{T+1}. After the computation, we have $(\Phi_t)_{t=0}^T$: rich dynamics of heterogeneous agents.

Impulse response analysis (perfect foresight)

- A conventional method utilizes (S)VAR on the simulated data.
  - (Log-)Linearity is usually imposed due to the estimation specification.
- Instead, an impulse response function (IRF) can be directly computed to an unexpected aggregate shock (MIT shock).
  - pros: possible non-linearity can be well-captured.
  - cons: certainty equivalence is imposed due to an MIT shock + perfect foresight.
- In the end, the IRF algorithm is identical to the TCE algorithm:
  - There is no new steady-state (the aggregate shock is not permanent). Replace the new SRCE with the original SRCE in the TCE algorithm.
  - In the backward solution step, make sure to impose the aggregate shock at $t = 1$.
  - A persistent shock can be considered (e.g., $log(A_{t+1}) = \rho log(A_t)$, $A_1 = \sigma$).
- An extremely useful tool for analyzing how heterogeneity affects the business cycle (Micro to Macro) and how the business cycle heterogeneously affects agents (Macro to Micro).
- Boppart et al. (2018) develops an algorithm for solving HA model with aggregate uncertainty based on this algorithm. We will come back later.

Computation - Impulse response function

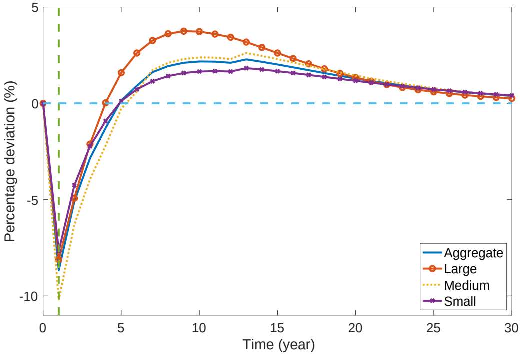
1. Compute the SRCE (Φ_0, v_{T+1})
2. Compute TCE
   2.1 Set a long enough time T (e.g., 50 years).
   2.2 Guess the price path (aggregate allocation path): $(K_t^{guess})_{t=1}^T$.
   2.3 Given v_{T+1} and $(K_t^{guess})_{t=1}^T$, solve $(v_t, g_{a,t}, g_{c,t})_{t=1}^T$ using the finite backward iteration.
      - Make sure to properly apply the shock level at each time t.
   2.4 Given Φ_0, simulate forward using $(g_{a,t})_{t=1}^T$ to obtain $(\Phi_t)_{t=1}^T$.
      - Either stochastic or non-stochastic methods work.
      - Eigenvalue method does not work (non-stationarity).
   2.5 Aggregate $(\Phi_t)_{t=1}^T$ to obtain $(K_t^{update})_{t=1}^T$.
   2.6 Check if $||K^{update} - K^{guess}||_p < Tol$
      - If yes, the solution converged.
      - If no, go back to step 3.2.

Note: From the SRCE, we obtain both $\Phi_0 = \Phi^{SRCE}$ and $v_{T+1} = v^{SRCE}$.
After the computation, we have $(\Phi_t)_{t=0}^T$: rich dynamics of heterogeneous agents.

An example of IRF

The figure plots the impulse responses of the firm-level and the aggregate spike ratios.

- The spike ratio is defined as the portion of firms making investment greater than 20%.



Notes: The figure is from Lee (2021).

Concluding remark

Summary

- We have studied the model in Aiyagari (1994) and how to compute the SRCE.
- Incomplete market and the borrowing constraint lead to a rich wealth dynamics of households.
- The computation of this model is a crucial step to jump into the HA world. Other models are computed in very similar steps except for some details.
- We have studied the algorithm for the transitional dynamics under perfect foresight.
- An impulse response is a particular type of transitional dynamics.
- What if there is an aggregate uncertainty? (perfect foresight breaks down)
  - Next lecture on Krusell and Smith (1998)

Homework

Do it yourself, and compare it with your colleagues. The sample replication code will be provided in the following week (not an answer key).

1. Replicate the Aiyagari economy with productivity process discretized by Tauchen method under the following parametrization:
   - log utility and the discount factor of 0.96
   - log income persistence 0.9 and volatility 0.1 (AR(1) process)
   - Tauchen: 7 grid points with a three-standard deviation range covered.
   - Production: capital share 0.36 and depreciation 0.08
   - Borrowing constraint at 0
2. Compute the wealth portion of top 0.1%/1%/10%/20%/50% in GE.
3. Compare the magnitude of $1 + r(\Phi)$ with $1/\beta$ and explain it.
4. Draw supply and demand curves in the capital market.
5. Let's assume aggregate productivity unexpectedly permanently jumps up by 5% $(1 \to 1.05)$. Compute the transitional dynamics and compare the wealth portion dynamics of top wealth groups on the transition path (x-axis: time, y-axis: portions).

Definitions ▸ Back

Definition 1 (Irreducibility)

A Markov chain is irreducible if all the states are reachable from each other (a single communicating class).

Definition 2 (Aperiodicity)

A Markov chain $\{x_1, x_2, \dots\}$ is aperiodic if all of the states' period is 1, where the period of state i, d_i is defined as follows:

$$d_i = gcd\{n | P(x_n = i | x_0 = i) > 0\}$$

$$P_1 = \begin{bmatrix} 0.9 & 0.1 & 0 \\ 0.1 & 0.9 & 0 \\ 0 & 0 & 1 \end{bmatrix} : \text{ reducible} \quad P_2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} : \text{ periodic (period=3)}$$