Modern macroeconomic model

Kazu Matsuda

Motivation

- What’s the effect of government policy?
- We want to experiment government policy but can’t…
- Let’s build a model imitating Japan on which we can simulate.

Model ingredients

- What should we have in the model?
  - Many households (including you)
  - Many firms
  - A government (we don’t have it this time for simplicity)
- And we have time in the model to dynamic effects of policy.

Households

- Continuum (measure 1) of households indexed by i.
- They work (supply 1 unit of labor), consume c_{it}, and save as assets a_{it+1}.
- Each household lives infinitely and maximizes the following utility

$$E_0 \sum_{t=0}^{\infty} \beta^t u(c_{it}),\ u' > 0,\ u'' < 0,\ \beta \in (0,1)$$

Income shocks

- Ex-ante identical but faced with idiosyncratic income shocks.
- Stochastic idiosyncratic endowments of efficiency units $h_{it} \in \mathscr{H} = \{h^1, \cdots, h^{N_H}\}$
- The Markov process: $\pi(h' \mid h)$
  - π^* is the invariant distribution associated with π.
- Aggregate endowment of skills

$$H_t = \sum_{i=1}^{N_H} h_i \pi^*(h_i).$$

Household constraints

- Interest rates on assets r_t and wages w_t.
- Budget constraint

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

- Borrowing constraint

$$a_{it+1} \geq -\underline{B}$$

- For simplicity, we assume that households must choose asset levels from $\mathscr{A} = \{a^1, \cdots, a^{N_A}\}$.

Max problem

$$\max_{\{c_{it}\},\{a_{it+1}\}} E_0 \sum_{t=0}^{\infty} \beta^t u(c_{it}) \text{ s.t.}$$

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

$$a_{it+1} \geq -\underline{B},\ c_{it} \geq 0,\ a_{i0} \text{ given}$$

- How to solve this dynamic optimization problem?

Max problem

$$\max_{\{a_{it+1}\}} E_0 \sum_{t=0}^{\infty} \beta^t u((1 + r_t)a_{it} + w_t h_{it} - a_{it+1}) \text{ s.t.}$$

$$a_{it+1} \geq -\underline{B}, (1 + r_t)a_{it} + w_t h_{it} - a_{it+1} \geq 0, a_{i0} \text{ given}$$

- How to solve this dynamic optimization problem?

General form

$$\max_{\{s_t\}} \sum_{t=0}^{T} \beta^t F(s_t, s_{t+1}) \text{ s.t. } s_{t+1} \in \Gamma(s_t),\ s_0 \text{ given.}$$

- For the earlier problem, set $s_t = a_{it}$ and
  - $T \to \infty$,
  - $F(s_t, s_{t+1}) = u(w_t h_{it} + (1 + r_t)a_{it} - a_{it+1})$
  - $s_{t+1} \in \Gamma(s_t) = [-\underline{B}, w_t h_{it} + (1 + r_t)a_{it}]$

Two methods for solving this

- Method 1: take FOC's for $s_1, \cdots, s_{T+1}$.

$$\beta^t F_1(s_t, s_{t+1}) + \beta^{t-1} F_2(s_{t-1}, s_t) = 0.$$

- Method 2:
  - In period 0, s_0 given, choose s_1 (but s_1 also influences future payoffs and choices), get $F(s_0, s_1)$ today.
  - In period 1, s_1 given, choose $s_2 \in \Gamma(s_1)$ and get $F(s_1, s_2)$ today.
  - Balance current F and future consequences.

Basic idea

- Time itself is not important. If the economy starts again from s_1^*, the optimal path doesn't change.

- Optimal solution has the property that the optimal choice for s tomorrow only depends upon s today, and not the actual period.

- Two ways to think about solving this problem

  1. Find $s_0^*, s_1^*, s_2^* \ldots$

  2. $g(s)$ "optimal policy function" $s_0, g(s_0), g(g(s_0)), g(g(g(s_0))), \ldots$ (optimal way to respond)

- Jump ahead to period T. s_T will be given from earlier choices.

$$V_T(s_T) = \max_{s_{T+1}} F(s_T, s_{T+1}) \text{ s.t. } s_{T+1} \in \Gamma(s_T).$$

- $g_T(s_T)$ is the set of maximizers.

- Go to period $T-1$.

$$V_{T-1}(s_{T-1}) = \max_{s_T \in \Gamma(s_{T-1})} F(s_{T-1}, s_T) + \beta V_T(s_T) \text{ and } g_{T-1}(s_{T-1}) \text{ is maximizers.}$$

$$V_0(s_0) = \max_{s_1 \in \Gamma(s_0)} F(s_0, s_1) + \beta V_1(s_1) \text{ and } g_0(s_0) \text{ is maximizers.}$$

- What if we let $T \to \infty$?

- Intuitively, there is always an infinite number of periods after the current period so we would think that all of Vs are the same.

$$V(s_t) = \max_{s_{t+1} \in \Gamma(s_t)} \{F(s_t, s_{t+1}) + \beta V(s_{t+1})\}.$$

This equation of functions is called Bellman equation.

How to find V?

$$V(s_t) = \max_{s_{t+1} \in \Gamma(s_t)} \{F(s_t, s_{t+1}) + \beta V(s_{t+1})\}.$$

- Heuristically, suppose I give you some function $\hat{V}(s)$. Solve the following problem,
- $\forall s$, solve $\max_{s' \in \Gamma(s)} \{F(s, s') + \beta \hat{V}(s')\}$ and call the max $\tilde{V}(s)$.
- RHS can be used to map the function $\hat{V}(s)$ into another function $\tilde{V}(s)$.
- Repeat this process until $\hat{V}(s)$ converges and the point will be $V(s)$.

Household problem

$$V(a,h) = \max_{c,a'} u((1+r)a + wh - a') + \beta \sum_{h'} V(a',h')\pi(h' \mid h) \text{ s.t.}$$

$$-\underline{B} \leq a' \leq (1+r)a + wh.$$

- Solutions are policy functions $g_a(a,h)$.

Firms

- All the firms have production function

$$Y_t = F(K_t, H_t).$$

- Profit: $F(K_t, H_t) - (r_t + \delta)K_t - w_t H_t$

- Capital depreciates at δ and FOC:

$$w_t = F_H(K_t, H_t),$$

$$r_t + \delta = F_K(K_t, H_t).$$

Markets

- How wage w_t and rent r_t are determined?
- Prices clear the 3 markets
  - Labor: w_t
  - Assets: r_t
  - Goods: normalize 1 in steady state

Aggregate state

- For $t \geq 0$ the state of the economy is a distribution of households $\mu_t(a, h)$ over (a, h)

- The state space is $\mathscr{H} \times \mathscr{A}$

$$\mathscr{H} = \{h^1, \cdots, h^{N_H}\}, \mathscr{A} = \{a^1, \cdots, a^{N_A}\}.$$

- How does μ_t evolve over time?

$$\mu_{t+1}(a', h') = \sum_a \sum_h \mathbf{1}\{a : g_a(a, h) \in a'\}\pi(h' | h)\mu_t(a, h)$$

Stationary equilibrium

- We focus on stationary eq where the income distribution μ_t is constant.
  - Though stationary, agents move within the income distribution.
  - In stationary eq, prices (r_t and w_t) should be constant.

Stationary competitive equilibrium

- A stationary CE is a list of functions $V(a,h)$, $g_a(a,h)$, K, H, r, w, $\mu(a,h)$, s.t.

1. (Household optimization) Taking r and w as given, $V(a,h)$ solves

$$V(a,h) = \max_{a'} u((1+r)a + wh - a') + \beta \sum_{h'} V(a',h')\pi(h'|h) \text{ s.t.}$$

$$-\underline{B} \leq a' \leq (1+r)a + wh \text{ and } g_a(a,h) \text{ is an optimal decision rule.}$$

2. (Firm optimization) Taking r and w as given, K and H solve firms problem

$$\max_{k,h} F(k,h) - (r+\delta)k - wh \text{ such that } k \geq 0, h \geq 0.$$

3. (Market clearing)

(1) Labor $H = \sum_h h\pi^*(h)$, (2) Assets $K = \sum_a \sum_h g_a(a,h)\mu(a,h)$,

(3) Goods $F(K,H) = \sum_a \sum_h ((1+r)a + wh - g_a(a,h))\mu(a,h) + \delta K$

4. (Aggregate law of motion) Distribution of agents over states μ is stationary

$$\mu(a',h') = \sum_a \sum_h \mathbf{1}\{a : g_a(a,h) \in a'\}\pi(h' \,|\, h)\mu(a,h)$$

Setting $\pi(h' \mid h)$

- Assume that efficiency units follow an AR1

$$\ln h' = \rho \ln h + \epsilon, \;\; \epsilon \sim N(0, \sigma_\epsilon^2)$$

- Discretize using Tauchen's method.

1. Make $\mathcal{H}$ evenly spaced from -1*$stdd(\ln h)$ to +1*$stdd(\ln h)$. d will be the distance between grids.

2. Assume h_j goes to $h_{j'}$ if $\rho \ln h_j + \epsilon$ is in $[\ln h_{j'} - d/2, \ln h_{j'} + d/2]$.

$$\pi(h_{j'} \mid h_j) = N(\ln h_{j'} + d/2 - \rho \ln h_j) - N(\ln h_{j'} - d/2 - \rho \ln h_j)$$

Computing aggregate labor H

- Start with initial $\pi^{*0}(h_j) = 1/N_H$. Solve forward.
- First, set initially $\pi^{*1}(h_j) = 0$ for each h_j. Then for each h_j on grid $\mathscr{H}$,

$$\pi^{*1}(h_{j'}) \leftarrow \pi^{*1}(h_{j'}) + \pi(h_{j'} \mid h_j)\pi^{*0}(h_j) \text{ for each } h_{j'} \text{ on grid } \mathscr{H}.$$

(Note that they are not equal signs. “Accumulate” in the code).

- Repeat this until $d(\pi^{*1}, \pi^{*0}) < tol$. After finishing this, get $H = \sum_{i=1}^{N_H} h_i \pi^*(h_i)$

Setting functions

- Utility $u(c) = \dfrac{c^{1-\sigma}}{1-\sigma}$.

- Production function $F(K,H) = K^{\alpha}H^{1-\alpha}$. Then firm's FOC

$$r + \delta = F_K(K,H) = \alpha \left(\frac{K}{H}\right)^{1-\alpha}, w = F_H(K,H) = (1-\alpha)\left(\frac{K}{H}\right)^{\alpha-1}$$

- Define $R \equiv 1 + r$.

Computation

1. Guess K^0 Calculate r^0 and w^0 using firm's FOC.

2. Given (r^0, w^0), solve household's problem to get $g_a^0(a, h)$.

3. Use policy function g_a^0 and transition $\pi(h' \mid h)$ to compute $\mu^0(a, h)$.

4. Use invariant distribution $\mu^0(a, h)$ to compute $\tilde{K}^0 = \sum_a \sum_h g_a(a, h)\mu^0(a, h)$.

5. Stop if $|\tilde{K}^0 - K^0| < tol$. Otherwise, update $K^{j+1} = \phi K^j + (1 - \phi)\tilde{K}^j$ and go to step 2.

Discretized value function iteration

1. Choose grids on state variables: $\mathscr{A} \times \mathscr{H}$

2. Initial guess of value function (a matrix): $V_0(a_i, h_j)$

3. Given V_0, for each $(a_i, h_j) \in \mathscr{A} \times \mathscr{H}$,

    1. Find $a' \in \mathscr{A}$ on the grid such that

    $$g_a(a_i, h_j) = a' \in \arg\max_{a' \in \mathscr{A}} u(wh_j + Ra_i - a') + \beta \sum_{y' \in \mathscr{Y}} V_0(a', h')\pi(h' \mid h_j)$$

    2. Update $V_1(a_i, h_j) = u(wh_j + Ra_i - g_a(a_i, h_j)) + \beta \sum_{h' \in \mathscr{H}} V_0(g_a(a_i, h_j), h')\pi(h' \mid h_j)$

4. If $d(V_0, V_1) < tol$, done. Otherwise return to 3 with new guess V_1.

Discretization of density function μ

- Approximate the density by a probability distribution function defined over discretized version of the state space $\mathscr{A} \times \mathscr{H}$.

- Start with initial $\mu^0(a_i, h_j) = \dfrac{1}{N_A N_H}$

- N_A and N_H are the number of grids for $\mathscr{A}$ and $\mathscr{H}$.

Updating measures μ

- First, set initially $\mu^1(a_i, h_j) = 0$ for each (a_i, h_j).

- Then for each (a_i, h_j) on grid $\mathscr{A} \times \mathscr{H}$, for each j'

$$\mu^1(g_a(a_i, h_j), h_{j'}) \leftarrow \mu^1(g_a(a_i, h_j), h_{j'}) + \pi(h_{j'} | h_j)\mu^0(a_i, h_j)$$

for each j. (Note that they are not equal signs. Accumulate in the code)

- Repeat this until $d(\mu^1, \mu^0) < tol$.

Tax

- Now we assume that the government introduces labor income tax with rate τ (exogenous) and rebate it as lump-sum transfer T (endogenous).

$$\max_{\{c_{it}\},\{a_{it+1}\}} E_0 \sum_{t=0}^{\infty} \beta^t u(c_{it}) \text{ s.t.}$$

$$c_{it} + a_{it+1} = (1 + r_t)a_{it} + (1 - \tau)w_t h_{it} + T$$

$$a_{it+1} \geq -\underline{B},\ a_{i0} \text{ given}$$

Stationary competitive equilibrium

- A stationary CE with government policy is a list of functions $V(a, h)$, $g_a(a, h)$, K, H, r, w, $\mu(a, h)$, T s.t.

1. (Household optimization) Taking r and w as given, $V(a, h)$ solves

$$V(a, h) = \max_{a'} u((1 + r)a + (1 - \tau)wh + T - a') + \beta \sum_{h'} V(a', h')\pi(h' \mid h) \text{ s.t.}$$

$$-\underline{B} \leq a' \leq (1 + r)a + (1 - \tau)wh + T \text{ and } g_a(a, h) \text{ is an optimal decision rule.}$$

2. (Firm optimization) Taking r and w as given, K and H solve firms problem

$$\max_{k,h} F(k, h) - (r + \delta)k - wh \text{ such that } k \geq 0, h \geq 0.$$

3. (Government) $\tau w H = T$

4. (Market clearing)

(1) Labor $H = \sum_h h\pi^*(h)$, (2) Assets $K = \sum_a \sum_h g_a(a,h)\mu(a,h)$,

(3) Goods $F(K,H) = \sum_a \sum_h ((1+r)a + (1-\tau)wh + T - g_a(a,h))\mu(a,h) + \delta K$

5. (Aggregate law of motion) Distribution of agents over states μ is stationary

$$\mu(a',h') = \sum_a \sum_h \mathbf{1}\{a : g_a(a,h) \in a'\}\pi(h'|h)\mu(a,h)$$

Computation

1. Guess K^0 Calculate r^0 and w^0 using firm's FOC. Calculate $T^0 = \tau w^0 H$.
2. Given (r^0, w^0, T^0), solve household's problem to get $g_a^0(a, h)$.
3. Use policy function g_a^0 and transition $\pi(h' \mid h)$ to compute $\mu^0(a, h)$.
4. Use invariant distribution $\mu^0(a, h)$ to compute $\tilde{K}^0 = \sum_a \sum_h g_a(a, h)\mu^0(a, h)$.
5. Stop if $|\tilde{K}^0 - K^0| < tol$. Otherwise, update $K^{j+1} = \phi K^j + (1 - \phi)\tilde{K}^j$ and go to step 2.

Interpolated value function iteration

1. Choose grids on state variables: $\mathscr{A} \times \mathscr{H}$

2. Initial guess of value function (a matrix): $V_0(a_i, h_j)$

3. Given V_0, for each $(a_i, h_j) \in \mathscr{A} \times \mathscr{H}$,

    1. Find $a' \geq \bar{a}$ using an optimization routine with interpolated $\hat{V}_0$ such that

    $$g_a(a_i, h_j) = a' \in \arg\max_{a' \geq \bar{a}} u(wh_j + Ra_i - a') + \beta \sum_{h' \in \mathscr{H}} \hat{V}_0(a', h')\pi(h' \mid h_j)$$

    2. Update $V_1(a_i, h_j) = u(wh_j + Ra_i - g_a(a_i, h_j)) + \beta \sum_{h' \in \mathscr{H}} V_0(g_a(a_i, h_j), h')\pi(h' \mid h_j)$

4. If $d(V_0, V_1) < tol$, done. Otherwise return to 3 with new guess V_1.

- Not search finite grids anymore. Find a solution from an infinite set such as real numbers.

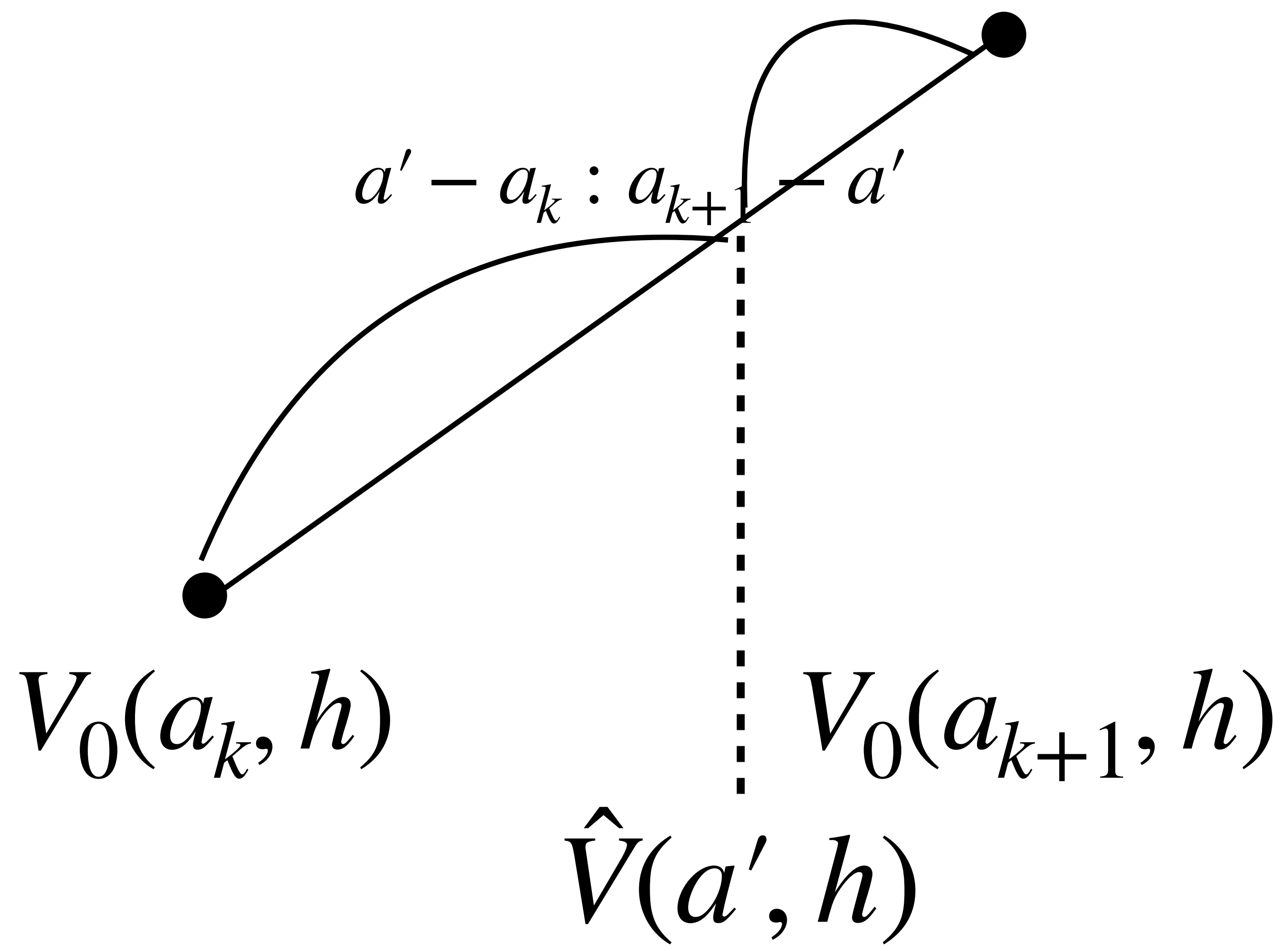
- Theoretically, we have to find V_0 for any real number a'. But we have only V_0 on finite grids. What if we need the value for a' NOT on the grids?

- Interpolation: functional approximation: they must have same values on the grids.

Linear interpolation

- We know the values only on the grids $V_0(a_i, h_j)$ on $\mathscr{A} \times \mathscr{H}$.
- Suppose that we need a value for $a' \notin \mathscr{A}$.
- Find the left grid $a_k \in \mathscr{A}$ such that $a_k \leq a' \leq a_{k+1}$.
- Then the value for a' is approximated as

$$\hat{V}_0(a', h) = \frac{a_{k+1} - a'}{a_{k+1} - a_k} V_0(a_k, h) + \frac{a' - a_k}{a_{k+1} - a_k} V_0(a_{k+1}, h)$$



Optimization routine: Bracketing method

- (Assume we want to minimize something. If you want to maximizing, just multiply by -1.)
- Most reliable for one dimensional problems
- Initialization: find $a < b < c$ such that $f(a), f(c) > f(b)$

1. Choose $d \in (a, c)$ and compute $f(d)$

2. Choose new (a, b, c) triplet. If $d < b$ and $f(d) > f(b)$, then there is a minimum in $[d, c]$. Update the triple (a, b, c) with (d, b, c). If $d < b$ and $f(d) < f(b)$, then the minimum is in $[a, b]$. Update the triple (a, b, c) with (b, d, c). Otherwise, update the triple (a, b, c) with (a, b, d).

- Stop if $c - a < \delta$. If not, go back to step 1

- Golden search: more sophisticated version that features an optimal way to segment intervals

Policy function iteration with linear interpolation

- Construct a grid on the asset space A.

- Guess an initial matrix of decision rules for a'' on the grid points, call it $\hat{a}_0(a_i, h_j)$, where subscript 0 denotes the initial iteration. Choose an interpolant. (e.g., piecewise linear)

- For each point (a_i, h_j) on the grid, check whether the borrowing constraint binds, i.e., check whether:

$$u'(Ra_i + wh_j - a_0) - \beta R \sum_{h' \in H} \pi(h' \mid h_j) u'(Ra_0 + wh' - \hat{a}_0(a_0, h')) > 0$$

- Two cases: (a) if this inequality holds, the borrowing constraint binds. Then, set $a_0'(a_i, h_j) = -\underline{B}$ and repeat this check for the next grid points (b) if the equation instead holds with the $\leq$ inequality, we have an interior solution (it is optimal to save) and we proceed to the next step.

- For each point (a_i, h_j) on the grid, use a nonlinear equation solver to find the solution $a*$ of the nonlinear equation

$$u'(Ra_i + h_j - a*) - \beta R \sum_{h' \in H} \pi(h' \mid h_j) u'(Ra* + wh' - \hat{a}_0(a*, h')) = 0$$

- (a) need to evaluate the function $\hat{a}_0(a, h')$ outside grid points.

- When the solver calls an $a*$ which lines between grid points, your interpolating should do as follows. First, find pair of adjacent grid points $\{a_i, a_{i+1}\}$ such that $a_i < a* < a_{i+1}$, and then compute

$$\hat{a}_0(a*, h') = \hat{a}_0(a_i, h') + (a* - a_i)\left(\frac{\hat{a}_0(a_{i+1}, h') - \hat{a}_0(a_i, h')}{a_{i+1} - a_i}\right)$$

- (c) if the solution of the nonlinear equation is $a*$≤ then set $a_0'(a_i, h_j) = a*$ and iterate on the next grid point.

- Check convergence by comparing $a_0'(a_i, h_j) - \hat{a}_0(a_i, h_j)$ through some pre-specified norm. For example, declare convergence at iteration n when

- $\max_{i,j}\{\,|\,a_n'(a_i, h_j) - \hat{a}_n(a_i, h_j)\,|\,\} < \epsilon$ for some small number ϵ which determined the degree of tolerance in the solution algorithm.

- If convergence is achieved, stop. Otherwise, go back to point 3 with the new guess $\hat{a}_1(a_i, h_j) = a_0'(a_i, h_j)$.

- Note that the most time-consuming step in this procedure is 5, the root-finding problem. Next class, we discuss how to avoid it.

If continuous

- Then for each (a_i, h_j) on grid $\mathscr{A} \times \mathscr{H}$, find the left grid $a_k \in \mathscr{A}$ such that $a_k \leq g_a(a_i, h_j) \leq a_{k+1}$.

- Then the value for a' is approximated as, for each j'

$$\mu^1(a_k, h_{j'}) \leftarrow \mu^1(a_k, h_j') + \pi(h_{j'} | h_j) \frac{a_{k+1} - g_a(a_i, h_j)}{a_{k+1} - a_k} \mu^0(a_i, h_j)$$

$$\mu^1(a_{k+1}, h_{j'}) \leftarrow \mu^1(a_k, h_{j'}) + \pi(h_{j'} | h_j) \frac{g_a(a_i, h_j) - a_k}{a_{k+1} - a_k} \mu^0(a_i, h_j)$$

for each j. (Note that they are not equal signs. Accumulate in the code)

- An interpretation is like a lottery. Suppose that a continuum of people with mass $\mu^0(a_i, h_j)$ are at (a_i, h_j) today. Then

- a fraction $\pi(h_1 \mid h_i)\dfrac{a_{k+1} - a}{a_{k+1} - a_k}$ goes to (a_k, h_1) at tomorrow.

- a fraction $\pi(h_1 \mid h_i)\dfrac{a - a_k}{a_{k+1} - a_k}$ goes to (a_{k+1}, h_1) at tomorrow.

- a fraction $\pi(h_2 \mid h_i)\dfrac{a_{k+1} - a}{a_{k+1} - a_k}$ goes to (a_k, h_2) at tomorrow and so on…

Monte-Carlo simulation

1. Generate a large sample of households and track them.

2. Choose a sample size I. Each sample is indexed by i.

3. Give initial value a_i^0 and h_i^0 for sample i.

4. Then calculate $a_i^1 = g_a(a_i^0, h_i^0)$ and draw h_i^1 from $\pi(\,\cdot \mid h_i^0)$ using a random number generator.

5. Do this for all i. Calculate a moment M^1 using the sample. Continue until convergence $d(M^0, M^1) < tol$. If not, go back to step 4 with initial a_i^1 and h_i^1.

6. $A(r^0)$ is just the mean of a_i^t in the final sample.

Which one is better

- Monte Carlo is time and memory consuming. Not recommended for low dimensional problems.

- Monte Carlo is good for high dimension problems.