Macroeconomic Theory - Huggett and Aiyagari Models

Darapheak Tin

Research School of Economics,
Australian National University

Sample Teaching Slides III

Table of Contents

Huggett Model

Huggett Model I - Environment

A simple **endowment economy** based on Huggett (1993).

- Unit mass of infinitely lived, ex-ante identical households.
- Households face idiosyncratic income shock y_t every period (no aggregate shocks).
- Incomplete (insurance) markets. Households can save/borrow one-period risk-free bond.
- No firm, no government. Output/income is an endowment.

Huggett Model II - Preferences and Asset

Preferences. time-separable with CRRA utility:

$$U = \sum_{t=0}^{\infty} \beta^t \, u(c_t), \qquad u(c) = \frac{c^{\,1-\sigma}}{1-\sigma}, \quad \beta \in (0,1), \quad \sigma > 0.$$

Asset (bond). Households trade a risk-free one-period bond with gross return R, given that $\beta R < 1$.

Timing (per period).

1. *State*: Observe income y_t and beginning-of-period bonds b_t.
2. *Decision*: Choose consumption c_t and next-period bonds b_{t+1}.
3. Income y_{t+1} is a stochastic component
   $\longrightarrow$ households form expectation $\mathbb{E}_t(.)$ about the future.

Note

R is *exogenous* here (partial equilibrium). With incomplete markets, the only way to smooth consumption is to accumulate b (self-insurance).

Huggett Model III - Idiosyncratic Income (Shock)

In the continuous version, income follows an AR(1) process:

$$y_t = \rho\, y_{t-1} + \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, \sigma_\varepsilon^2).$$

To make the model computable, we discretize this process into a finite-state Markov chain—e.g., using Tauchen method (Tauchen 1989).

1. **Discrete income states:**

$$y_t \in \{y^L, y^H\}, \qquad y^L < y^H$$

Agents can be in either a *low-income* or *high-income* state each period.

2. **Transition probabilities:**

$$\Pi = \begin{bmatrix} \pi_{LL} & \pi_{LH} \\ \pi_{HL} & \pi_{HH} \end{bmatrix}, \qquad \pi_{ij} = \Pr(s_{t+1} = j \mid s_t = i), \quad \text{rows sum to 1.}$$

- If π_{LL} is high, low-income households are likely to stay poor (persistent inequality).
- If π_{LH} is high, mobility from low to high income is frequent.

Huggett Model III - Idiosyncratic Income (Shock)

3. Evolution of income distribution:

At time t, the probability of being in each state is

$$\boldsymbol{\mu}_t = \begin{bmatrix} \mu_{L,t} \\ \mu_{H,t} \end{bmatrix}, \quad \text{where } \mu_{L,t} = \Pr(s_t = L), \quad \mu_{H,t} = \Pr(s_t = H).$$

The transition to $t+1$ follows a first-order Markov process:

$$\underbrace{\begin{bmatrix} \mu_{L,t+1} \\ \mu_{H,t+1} \end{bmatrix}}_{\mu_{t+1}} = \underbrace{\begin{bmatrix} \pi_{LL} & \pi_{HL} \\ \pi_{LH} & \pi_{HH} \end{bmatrix}}_{\Pi^T} \underbrace{\begin{bmatrix} \mu_{L,t} \\ \mu_{H,t} \end{bmatrix}}_{\mu_t} = \begin{bmatrix} \pi_{LL}\mu_{L,t} + \pi_{HL}\mu_{H,t} \\ \pi_{LH}\mu_{L,t} + \pi_{HH}\mu_{H,t} \end{bmatrix}.$$

- The $t+1$ distribution is a weighted average of current states.
- Over time, $\boldsymbol{\mu}_t$ converges to $\boldsymbol{\mu}^*$ satisfying $\boldsymbol{\mu}^* = \Pi^\top \boldsymbol{\mu}^*$.

Household I - Primal Problem

Given current bond and income (*state*: b_t, y_t), households choose consumption and next-period bond (*decision*: c_t, b_{t+1}) to maximize expected lifetime utility:

$$\max_{\{c_t, b_{t+1}\}} \mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t u(c_t)$$

subject to

$$c_t + b_{t+1} = y_t + b_t; \quad y_{t+1} \sim P(\cdot \mid y)$$
$$b' \geq \underline{b}, \quad c_t > 0;$$
$$\lim_{T\to\infty} \frac{b_{T+1}}{R^T} = 0; \quad \text{(TVC rules out Ponzi schemes/explosive debts)}$$

Household II - Recursive formulation (Bellman)

Bellman representation with state (b_t, y_t):

$$V(b_t, y_t) = \max_{c_t, b_{t+1}} \left\{ \frac{c_t^{1-\sigma}}{1-\sigma} + \beta \underbrace{\mathbb{E}_t \left[V(b_{t+1}, y_{t+1}) \mid y_t \right]}_{\sum_{y_{t+1}} \Pi(y_{t+1}|y_t) V(b_{t+1}, y_{t+1})} \right\} \quad (1)$$

subject to

$$c_t + b_{t+1} = y_t + Rb_t, \qquad b_{t+1} \geq \underline{b}_t, \qquad c_t > 0 \quad (2)$$

FOCs lead to decision rules:

$$b_{t+1} = g_b(b_t, y_t), \qquad c_t = g_c(b_t, y_t) = y_t + Rb_t - g_b(b_t, y_t)$$

▸ More on optimality conditions

Stationary Equilibrium I - Definition

A **stationary equilibrium** is a value function $V(b, y)$, decision rules $g_c(b, y)$ and $g_b(b, y)$, and a stationary (time-invariant distribution) $\mu(b, y)$ over states $\mathcal{B} \times \mathcal{Y}$ such that:

1. **Household optimality.** For each (b, y), the policies $g_c(b, y)$ and $g_b(b, y)$ solve the household Bellman problem given R and Π.

$$V(b, y) = \max_{b' \geq \underline{b}} \Big\{ u(c) + \beta \, \mathbb{E}[V(b', y') | y] \Big\},$$

   subject to constraints (2).

2. **Stationarity of the cross-sectional distribution.** Let $\mathcal{T}$ be the Markov operator (law of motion) induced by $b'(\cdot)$ and the income transition Π. Then, the distribution μ is stationary if $\mu(\cdot) \ = \ \mathcal{T} \, \mu(\cdot)$:

$$\mu(y', b') = \sum_{y \in \mathcal{Y}} \int_{\mathcal{B}} \pi(y' \mid y) \, \mathbf{1}\{b' = b(b, y)\} \, \mu(b, y) \, \mathrm{d}b.$$

Stationary Equilibrium II - Definition

3. **Bond market clearing.**

$$\sum_{y \in \mathcal{Y}} \int_{\mathcal{B}} g_b(b, y)\, \mu(b, y)\, \mathrm{d}b \;=\; 0.$$

Remark: In our partial-equilibrium Huggett benchmark, R is taken as exogenous and condition (2) can be omitted. With fixed R, this model is equivalent to an open economy (i.e., r constant at world rate) $\Rightarrow$ there may be net aggregate lending/borrowing due to foreign capital flows.

Calibration

Parameters		
Preferences	discount factor	$\beta = 0.98$
	CRRA coefficient	$\sigma = 2$
Credit limit	borrowing bound	$\underline{b} = -2$
Shocks	endowment levels	$y \in \{y^L, y^H\} = \{0.25,\ 3.0\}$
	transition matrix	$\Pi = \begin{bmatrix} \pi_{LL} & \pi_{LH} \\ \pi_{HL} & \pi_{HH} \end{bmatrix} = \begin{bmatrix} 0.6 & 0.4 \\ 0.3 & 0.7 \end{bmatrix}$

Table: Parameter values for calibration.

- $\underline{b} = -2$ allows limited borrowing (if you tighten it, it will increase the mass at the constraint and precautionary saving).
- In the basic Huggett benchmark, the gross return is fixed at $= 1.02$. In Aiyagari model, we will endogenise it via market clearing.

Huggett Algorithm - Overview I

Goal: Compute the stationary equilibrium of the Huggett economy by iterating on the Bellman equation (household problem) and the distribution of agents.

Step 1: Discretize state space

$$b \in \mathcal{B} = \{\, \underline{b}, \ldots, \bar{b} \,\}, \qquad y \in \{\, y^L, y^H \,\}.$$

For each (b, y), the next-period choice b' must satisfy

$$b' \geq \underline{b}, \quad c = y + Rb - b' > 0.$$

VFI iteration stage (detail in Appendix and lecture slides)

Step 2: Guess an initial value function $V^{(0)}(b, y)$.

Step 3: Iterate on the Bellman operator

$$V^{(k+1)}(b, y) = \max_{b' \geq \underline{b}} \Big\{ u(c) + \beta\, \mathbb{E}\big[V^{(k)}(b', y') | y \big] \Big\}.$$

where $c = y + Rb - b'$. The iteration continues until

$$\| V^{(k+1)} - V^{(k)} \| < \varepsilon_V.$$

Huggett Algorithm - Overview II

Step 4: Extract policy functions.

$$b'^{*} = g_b(b, y) = \arg\max_{b'} \Big\{ V(b, y) \Big\},$$

and define $c^* = g_c(b, y) = y + Rb - g_b(b, y)$.

Stationary distribution stage (detail in Appendix and lecture slides)

Step 5: Compute the stationary distribution $\mu(b, y)$.

- Initialize $\mu^{(0)}$ (e.g. uniform density across all (b, y)).
- For each iteration:

$$\mu^{(t+1)}(b', y') = \sum_{y \in \mathcal{Y}} \int_{\mathcal{B}} \mathbf{1}\{g_b(b, y) = b'\}\, \pi(y'|y)\, \mu^{(t)}(b, y) \mathrm{d}b.$$

- Interpolate when $g_b(b, y)$ lies between grid points.
- Iterate until $\|\mu^{(t+1)} - \mu^{(t)}\| < \varepsilon_\mu$.

Step 6: Market clearing (if R is endogenous): Adjust R until the bond market clears:

$$\sum_{y \in \mathcal{Y}} \int_{\mathcal{B}} g_b(b, y)\, \mu(b, y)\, \mathrm{d}b = 0.$$

Huggett_model_v0.m I - Initialization

```
% ---- Solve Huggett-type model ----
clear all; close all; tic

%% [1] Parameters
Pv     = 0.5*ones(1,2);            % initial income weights
gsigma = 2.0;                       % CRRA coefficient
gbeta  = 0.98;                      % discount factor

% Income shocks (two-state Markov)
yv = [0.25 3];                      % y^L, y^H
P  = [0.6 0.4;                      % [pi_LL, pi_LH;
      0.3 0.7];                     %  pi_HL, pi_HH]
for i = 1:30, Pv = Pv * P; end      % approach stationary weights

% Asset space (grid on bonds)
bmin     = 0;                % NOTE: use -2 for borrowing if desired
bmax     = 5;
grid     = round(100*(bmax - bmin)); % N points
bv       = linspace(bmin, bmax, grid)'; % Nx1 column grid
gridstep = bv(2) - bv(1);
distance = bv(end) - bv(1);
```

Huggett_model_v0.m II - Initialization

```
%% [2.] VFI + Stationary Distribution
% Fix R (Hugget PE)
r    = 0.02;
R    = 1 + r;
Rold = 1;

iter = 0;
error = 100;
errorv = [];

%% [2.1] Value function iteration (decision rule for saving)

% initial guess V^(0) for both shocks (N x 2)
Vnext = [ (bv - bmin + 0.1).^(1-gsigma)/(1-gsigma), ...
         (bv - bmin + 0.1).^(1-gsigma)/(1-gsigma) ];
EV   = (P * Vnext')';    % Nx2 expected value across shocks
Vnow = Vnext; bbv = bv;  % reuse grid as column for choices

% bookkeeping for VFI loop
iter1 = 0;
error1 = 100;
error1v = 100;
```

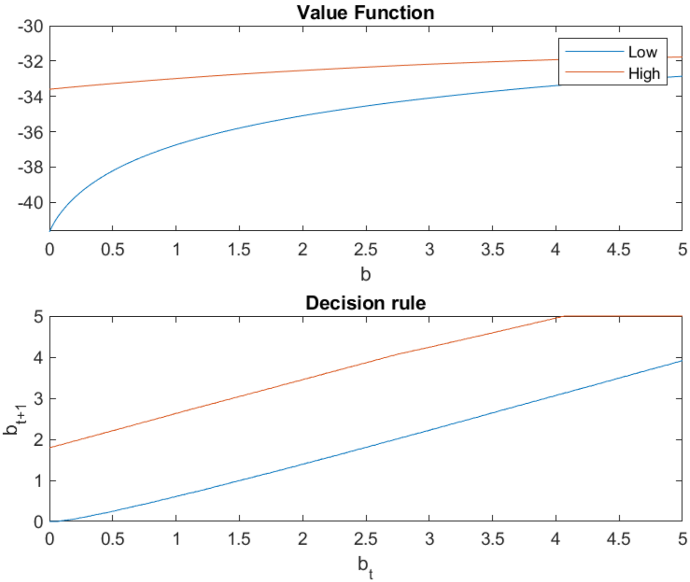
Huggett_model_v0.m III - Value function iteration

```
while (iter1 < 300 && error1 > 1e-4)
    % Update expected continuation values
    EV = (P * Vnext')';      % Nx2, column z is E[V | current state z]

    for i = 1:grid           % current asset b = bv(i)
        for z = 1:2          % current income state (L/H)
            income = yv(z) + R * bv(i);    % cash on hand m_t
            % vector of feasible consumption for all b' in grid
            cv = income - bbv;             % Nx1
            cv = (cv > 0).*cv + (cv <= 0)*1e-10; % positive c only
            % value over all b' choices for this (b_i, z)
            vv = cv.^(1-gsigma)/(1-gsigma) + gbeta * EV(:,z);

            % maximise over b' (store argmax)
            [val, pos] = max(vv);
            Bopt(i,z) = bbv(pos);          % policy b'(b_i,z)
            Copt(i,z) = cv(pos);           % c(b_i,z)
            Vnow(i,z) = val;               % V^{new}(b_i,z)
        end
    end
    % relative error across two consecutive iterations
    error1 = 100 * sum(sum(abs(Vnow - Vnext))) / sum(sum(abs(Vnext)));
    iter1  = iter1 + 1;
    Vnext  = Vnow;                          % update
end
```

Huggett_model_v0.m IV - Value function iteration

```
figure(1)
subplot(2,1,1);
plot(bv, Vnow);
legend('Low','High');
xlabel('b');
title('Value Function');

subplot(2,1,2);
plot(bv, Bopt);
xlabel('b_t');
ylabel('b_{t+1}');
title('Decision rule');

saveas(gcf, 'Tutorial11_Fig1.png');
```

Hugget_model_v0.m - Policy and Value Functions



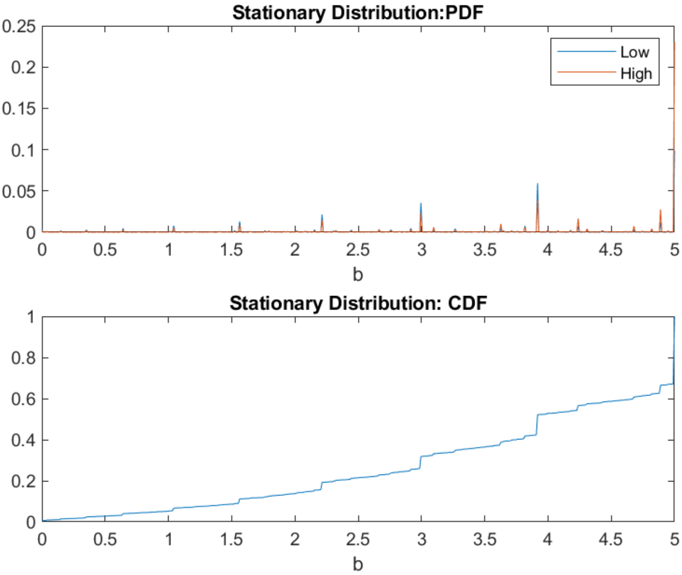
Huggett_model_v0.m V - Stationary distribution

```
%% [2.2] Stationary distribution via linear interpolation
Mnow  = ones(grid,2)/(grid*2);        % uniform initial mass (Nx2)
iter2 = 0; error2 = 10;
while (iter2 < 1000 && error2 > 1e-10)
   Mnext = zeros(grid,2);
   for i = 1:grid
       for z = 1:2
           % locate b'(i,z) on the grid (interpolation)
           posL = min(floor((Bopt(i,z)-bmin)/distance*grid) + 1, grid);
           posL = round(posL);
           if bv(posL) > Bopt(i,z), posL = posL - 1; end
           posH   = min(posL + 1, grid);
           weight = (Bopt(i,z) - bv(posL)) / gridstep; % w_H
           % transition across income states
           transp = Mnow(i,z) * P(z,:); % 1x2 mass moved to (L,H)
           % distribute mass to neighboring b' nodes
           for zz = 1:2
              Mnext(posL,zz) = Mnext(posL,zz) + (1-weight) * transp(zz);
              Mnext(posH,zz) = Mnext(posH,zz) + weight * transp(zz);
           end
       end
   end
   error2 = sum(sum(abs(Mnext - Mnow)));
   Mnow   = Mnext; iter2 = iter2 + 1;
end
```

Huggett_model_v0.m VI - Stationary distribution

```
%% [3.] Reporting results: PDF and CDF of assets
[rs,cs] = size(Mnow);
F       = zeros(rs,cs);
F(1,:)  = Mnow(1,:);
for z = 1:2
    for i = 2:rs, F(i,z) = F(i-1,z) + Mnow(i,z); end
end

figure(2);
subplot(2,1,1);
plot(bv, Mnow); legend('Low','High');
xlabel('b'); title('Stationary Distribution: PDF');

subplot(2,1,2);
plot(bv, sum(F,2)); xlabel('b'); title('Stationary Distribution: CDF');

% Export
saveas(gcf, 'Tutorial11_Fig2.png');

toc
```

Hugget_model_v0.m - Distribution



Aiyagari Model

Huggett vs Aiyagari (what changes?)

	Huggett	**Aiyagari**
Production side	None (endowment economy). No firms.	Representative firm with $Y = AK^{\alpha}H^{1-\alpha}$.
Assets	One risk-free bond b; zero net supply.	Physical capital k accumulated by households
Prices	R exogenous	r, w from firm FOCs at (K, H); GE requires $K^S{=}K^D$.
Market clearing object	None ($R^\star$ to clear bond market if endogenous).	Aggregate capital $K^\star$ (equivalently $r^\star, w^\star$).
Risk/incompleteness	Idiosyncratic income y; no Arrow securities; borrowing limit.	Idiosyncratic productivity z; same incompleteness; borrowing limit on k'.
Resource constraint	Household budget only; no production.	$C + K' = Y + (1-\delta)K$ at the aggregate level.
Algorithm	Fix $R \to$ VFI $\to$ stationary μ.	Guess $K \to$ implied r, w $\to$ VFI $\to$ stationary μ $\to$ implied $K_{\text{new}} \to$ update until $K_{\text{new}} = K$.

Household I

Demography and preferences.

- A continuum of infinitely lived, ex-ante identical households (unit mass).
- Preferences:

$$E_0 \sum_{t=0}^{\infty} \beta^t u(c_t), \qquad u(c) = \frac{c^{1-\sigma}}{1-\sigma}, \quad \beta \in (0,1),\ \sigma > 0.$$

- Each household is endowed with initial assets (physical capital) k_0 and one unit of time per period.
- Leisure (l_t) is not valued in our baseline (inelastic). Extensions can add l_t as a disutility term in preferences.

Timing (per period).

1. *State*: Observe productivity z_t and beginning-of-period assets (capital) k_t.
2. *Decision*: Choose consumption c_t and next-period capital k_{t+1}.
3. Productivity z_{t+1} is a stochastic component $\longrightarrow$ households form expectation $\mathbb{E}_t(.)$ about the future.

Household II

Idiosyncratic risk.

- Two possible productivity states $s \in \{L, H\}$. Labor productivity $z_t \in \mathcal{Z} = \{z^L, z^H\}$ follows a Markov chain with transition matrix

$$\Pi = \{\pi_{ss'}\} \text{ with } \pi_{ss'} = \pi(z_{t+1} = s' \mid z_t = s).$$

  Specifically: $\Pi = \begin{bmatrix} \pi_{LL} & \pi_{LH} \\ \pi_{HL} & \pi_{HH} \end{bmatrix}$

- Labor income is $w_t z_t$, where w_t is the market wage.

Assets and market incompleteness.

- Households save in one risk-free asset (physical capital) k_{t+1}, yielding gross return $R_t \equiv 1 + r_t$.
- No Arrow securities $\Rightarrow$ idiosyncratic risk cannot be fully insured (incomplete markets). Borrowing is limited by $k_{\min}$.

Remark: This is virtually identical to Hugget's household, except the idiosyncratic risk is associated with productivity (due to production economy) rather than endowment. Additionally, instead of savings in bonds b, household savings become future productive capital k'.

Household III: Recursive formulation (Bellman)

Given prices $\{w_t, R_t\}$, and current assets and realized productivity (*state*: k_t, z_t), the household chooses consumption and next-period assets (*decision*: c_t, k_{t+1}) to solve:

$$V(k_t, z_t) = \max_{c_t, k_{t+1}} \left\{ \frac{c_t^{1-\sigma}}{1-\sigma} + \beta \underbrace{\mathbb{E}_t \left[V(k_{t+1}, z_{t+1}) \mid z_t \right]}_{\sum_{z_{t+1}} \Pi(z_{t+1}|z_t) V(k_{t+1}, z_{t+1})} \right\} \tag{3}$$

subject to the budget constraint

$$c_t + k_{t+1} = w_t z_t + R_t k_t, \qquad k_t \geq k_{min}, \qquad c_t > 0. \tag{4}$$

FOCs lead to decision rules:

$$k_{t+1} = g_k(k_t, z_t), \qquad c_t = g_c(k_t, z_t) = w_t z_t + R k_t - k(k_t, z_t)$$

Aiyagari Household Euler Conditions

Firm (Aiyagari production side)

Technology. A representative pcompetitive firm with $Y = AK^{\alpha}H^{1-\alpha}$ and depreciation δ.

The representative firm's problem is:

$$\max_{K_t, H_t} \left\{ AK_t^{\alpha}H_t^{1-\alpha} - w_t H_t - q_t K_t \right\},$$

Factor prices (FOCs):

$$r = \alpha A K^{\alpha-1}H^{1-\alpha} - \delta, \qquad w = (1-\alpha)AK^{\alpha}H^{-\alpha},$$

where aggregate effective labor $H = \sum_z z\,\mu(z)$ under inelastic labor.

Goods resource constraint (no government).

$$C + K' = Y + (1-\delta)K.$$

Stationary Equilibrium I

A **stationary equilibrium** consists of value function $V(k,z)$, decision rules for current consumption $g_c(k,z)$ and next-period assets/capital $g_k(k,z)$, time-invariant prices $\{w,R\}$ for labor and capital, stationary distribution $\mu(k,z)$ and aggregate quantities $\{K,H,C\}$ such that:

1. **Household optimality.** $g_c(k,z)$ and $g_k(k,z)$ solve the household problem, given (r,w) and productivity transition Π:

$$V(k,z) = \max_{c,k'} \left\{ \frac{c^{1-\sigma}}{1-\sigma} + \beta\,\mathbb{E}\big[V(k',z') \mid z\big] \right\}$$

   subject to constraints (4)

2. **Firm optimality.** Firm maximises profits:

$$\max_{K,H} \; AK^{\alpha}H^{1-\alpha} - wH - qK.$$

   Factor prices (r,w) satisfy marginal product conditions:

$$w = (1-\alpha)AK^{\alpha}H^{-\alpha}, \quad q = \alpha AK^{\alpha-1}H^{1-\alpha}, \quad R = 1 + q - \delta.$$

Stationary Equilibrium II

3. **Stationarity of the joint distribution:** The distribution μ over individual state (z, k) is stationary:

$$\mu(k', z') = \sum_z \sum_k \mathbf{1}\{k' = g_k(k, z)\}\pi(z'|z)\,\mu(k, z).$$

where $\mu(k, z)$ is the invariant measure over the state space (k, z).

4. **Factor markets clear, and aggregates are given by**

$$K = \sum_k \sum_z \mu(k, z)\, g_k(k, z), \qquad H = \sum_k \sum_z \mu(k, z)\, z,$$

$$C = \sum_k \sum_z \mu(k, z)\, g_c(k, z).$$

5. **Aggregate resource constraint holds:**

$$C + K' = Y + (1 - \delta)K$$

Calibration

Parameters		
Preferences	discount factor β	0.98
	risk aversion σ	2
Technology	TFP A	1
	(α, δ)	(0.36, 0.05)
Borrowing limit	capital floor k_{min}	0
Shocks	productivity $z \in \{z^L, z^H\}$	$\{0.10,\ 1.00\}$
	transitions Π	$\begin{bmatrix} 0.5 & 0.5 \\ 0.1 & 0.9 \end{bmatrix}$

Table: Baseline Aiyagari calibration (`Aiyagari_model_v2.m`).

Aiyagari Algorithm I - Value function iteration

(**Remark:** *These Aiyagari VFI slides are adapted from the lecture slides.*)

[1] Discretize the current asset space $k = kv$ as a column vector of grid points:

$$kv = \begin{bmatrix} k_{\min}^{1} \\ \vdots \\ k_{\max}^{N} \end{bmatrix}_{N \times 1} \qquad \text{(current capital grid).}$$

[2] Guess an initial value function for each shock state $s \in \{L, H\}$ (two columns for z^L, z^H):

$$VV^{\text{iter}=1} = \begin{bmatrix} 0.1 & 0.1 \\ \vdots & \vdots \\ 0.1 & 0.1 \end{bmatrix}_{N \times 2} \quad .$$

Aiyagari Algorithm II - Value function iteration

[3] Candidate choices for next-period assets k' (same grid as kv):

$$kkv = \begin{bmatrix} k^1_{\min} \\ \vdots \\ k^N_{\max} \end{bmatrix}_{N\times 1} \quad \Rightarrow \quad k' \in \{kkv(j)\}_{j=1}^N.$$

Feasible consumption vectors given current (k', z):

$$\underbrace{cv\big(k, z^L\big)}_{\text{size } N\times 1} = \begin{bmatrix} c^1 \\ \vdots \\ c^N \end{bmatrix} = w\, z^L + (1+r)\, kv(i) - \overbrace{\begin{bmatrix} k^1_{\min} \\ \vdots \\ k^N_{\max} \end{bmatrix}}^{kkv},$$

$$\underbrace{cv\big(k, z^H\big)}_{\text{size } N\times 1} = \begin{bmatrix} c^1 \\ \vdots \\ c^N \end{bmatrix} = w\, z^H + (1+r)\, kv(i) - \begin{bmatrix} k^1_{\min} \\ \vdots \\ k^N_{\max} \end{bmatrix}.$$

Aiyagari Algorithm III - Value function iteration

[4] Iterate on Bellman operator:

For each current state pair $(k^i = kv(i), z)$, form the vector of RHS values across all $k' = kkv(j)$ choices and take a max.

Low state z^L:

$$\underbrace{VV^{L,\text{iter}+1}(k^i)}_{\text{scalar}} = \max_{j=1,\dots,N} \left\{ \frac{\left[wz^L + (1+r)\,kv(i) - kkv(j)\right]^{1-\sigma}}{1-\sigma} \right.$$

$$\left. + \beta\left[\pi_{LL}\, VV^{L,\text{iter}}(kkv(j)) + \pi_{LH}\, VV^{H,\text{iter}}(kkv(j))\right] \right\}$$

High state z^H:

$$\underbrace{VV^{H,\text{iter}+1}(k^i)}_{\text{scalar}} = \max_{j=1,\dots,N} \left\{ \frac{\left[wz^H + (1+r)\,kv(i) - kkv(j)\right]^{1-\sigma}}{1-\sigma} \right.$$

$$\left. + \beta\left[\pi_{HL}\, VV^{L,\text{iter}}(kkv(j)) + \pi_{HH}\, VV^{H,\text{iter}}(kkv(j))\right] \right\}.$$

Store the *argmax* index $j^{L,\max}(i)$ and $j^{H,\max}(i)$ to recover policies $k_L'^{*} = g_k(k^i, z^L)$ and $k_H'^{*} = g_k(k^i, z^H)$, and corresponding values.

Aiyagari Algorithm IV - Value function iteration

[4] Update and repeat.

- After looping $i = 1, \ldots, N$, stack the updated columns to get $VV^{\text{iter}+1} \in \mathbb{R}^{N\times 2}$.
- Compute the error and iterate until convergence:

$$\text{error} \;=\; \left\| \, VV^{\text{iter}+1} - VV^{\text{iter}} \, \right\| < \epsilon.$$

[5] Extract policy functions.

$$g_k(k^i, z^L) = kkv\big(j^{L,\max}(i)\big), \qquad g_k(k^i, z^H) = kkv\big(j^{H,\max}(i)\big),$$

$$g_c(k^i, z) = wz + (1+r)\,k^i - g_k(k^i, z) \qquad \text{for } z \in \{z^L, z^H\}$$

Aiyagari Algorithm I - Stationary Distribution

(**Remark:** *These Aiyagari Stationary Distribution slides are adapted from the lecture slides.*)

[1] Stationarity condition. Given (R, w) and individual policies $g_k(k,z), g_c(k,z)$, the distribution μ over individual states (k,z) is *stationary* if

$$\mu(k', z') \ = \ \sum_z \sum_k \mathbf{1}\{k' = g_k(k,z)\}\pi(z'|z)\,\mu(k,z)$$

[2] Representing distribution at iteration t, $\mu^{(t)}(k,z)$ as an $N \times 2$ matrix:

$$\mu^{(t)}(k,z) = \begin{bmatrix} \mu^{(t)}_{1,1} & \mu^{(t)}_{1,2} \\ \vdots & \vdots \\ \mu^{(t)}_{i,1} & \mu^{(t)}_{i,2} \\ \vdots & \vdots \\ \mu^{(t)}_{N,1} & \mu^{(t)}_{N,2} \end{bmatrix}_{N\times 2}$$

columns: productivity (z^L, z^H), rows: asset grid $k^1, \ldots, k^N$.

Aiyagari Algorithm II - Stationary Distribution

[3] Initialize $\mu^{(0)}$ (e.g., uniform), and then iterate forward.

$$\mu^{(t+1)}(k', z') \; = \; \sum_z \sum_k \mathbf{1}\{k' = g_k(k, z)\}\pi(z'|z)\, \mu^{(t)}(k, z).$$

Vector view:

$$\underbrace{\begin{bmatrix} \mu^{(t)}_{1,1} & \mu^{(t)}_{1,2} \\ \vdots & \vdots \\ \mu^{(t)}_{i,1} & \mu^{(t)}_{i,2} \\ \vdots & \vdots \\ \mu^{(t)}_{N,1} & \mu^{(t)}_{N,2} \end{bmatrix}}_{\mu_{\text{now}}(k,z)} \xrightarrow[\Pi,\; k'=g_k(k,z)]{} \underbrace{\begin{bmatrix} \mu^{(t+1)}_{1,1} & \mu^{(t+1)}_{1,2} \\ \vdots & \vdots \\ \mu^{(t+1)}_{i,1} & \mu^{(t+1)}_{i,2} \\ \vdots & \vdots \\ \mu^{(t+1)}_{N,1} & \mu^{(t+1)}_{N,2} \end{bmatrix}}_{\mu_{\text{next}}(k',z')}$$

Because $k' = g_k(k, z)$ may land *between* grid points, we split mass to neighbors (linear interpolation in code).

Aiyagari Algorithm III - Stationary Distribution

[4] Iterate until the distribution converges (stationary)

$$\|\mu^{(t+1)} - \mu^{(t)}\| \; < \varepsilon_\mu .$$

Note

(*) Like in Hugget model, individual policies $g_k(.), g_c(.)$ obtained via VFI are used to compute the stationary distribution $\mu(.)$, thus completing the **inner iteration**.

(**) However, in Aiyagari, there is an **outer iteration** (**Gauss-Seidel**) (i.e., VFI and stationary distribution blocks are nested inside the Gauss-Seidel loop). Given individual policies, stationary household distribution, and K^i (from guess or previous iteration), we aggregate decisions across households to obtain K^{i+1} and other aggregate terms:

$$K^{i+1} = \sum_k \sum_z \mu(k,z)\, g_k(k,z), \quad H = \sum_k \sum_z \mu(k,z)\, z,$$

$$C = \sum_k \sum_z \mu(k,z)\, g_c(k,z).$$

We continue the outer iteration until $\|K^{i+1} - K^i\| \; < \varepsilon_{GS}$. See **Aiyagari Computational Method** for the full solution algorithm.

Aiyagari_model_v2.m - Initialization

```
% ---- Aiyagari Model ----
clear all; close all; tic

% Preferences
gbeta = 0.98;     gsigma = 2.0;

% Technology
gA     = 1.0;      galpha = 0.36;    gdelta = 0.05;

% Idiosyncratic productivity (two-state Markov)
gPz = [0.5 0.5;
       0.1 0.9];
gzv = 1.0*[0.1, 1]; % z^L, z^H
gzn = 2;            % number of types
gPv = 0.5*ones(1,2);
for i=1:30; gPv=gPv*gPz; end
H = gPv*gzv';       % aggregate labor (given distribution over z)

% Discretize asset states
gkmin  = 0;         gkmax  = 30;             % min and max asset holdings
ggrid  = 10*gkmax;
gkv    = linspace(gkmin,gkmax,ggrid);     % vector of k grid points
ggridstep = gkv(2)-gkv(1);                % gridstep
gdist    = gkmax-gkmin;
```

Aiyagari_model_v2.m - VFI + Gauss-Seidel I

```
% Initialization for outer loop (Gauss-Seidel)
K      = 0.01;   Kold = K;
w      = 1.0;
r      = 0.02;   update = 0.1;

% Bookkeeping for Gauss-Seidel
iiter = 1;  maxiiter = 30;
itoler = 1e-2; ierror = 100;  % error for agg. capital

% [1.] Begin inner loop (Gauss-Seidel) - capital market clearing
% ---------------------------------------------------------
while (ierror > itoler) && (iiter <= maxiiter)
  % ---- Value function iteration (household) ----
  % preallocate
  Vnow  = ones(ggrid,gzn).^(1-gsigma)/(1-gsigma);
  Vnext = Vnow;
  Vopt  = Vnow;
  Sopt  = zeros(ggrid,gzn); % policy k'
  Copt  = zeros(ggrid,gzn); % policy c

  jiter = 1;
  jerror = 100;
```

Aiyagari_model_v2.m - VFI + Gauss-Seidel II

```
% [2.] Begin inner loop (household optimal decisions via VFI)
% -------------------------------------------------------
while (jerror > 1e-3) || (jiter < 200)
  % Continuation values E_z'[V(k',z')] given Vnext guess/update
  EVnext = zeros(ggrid,gzn);
  for i = 1:ggrid
    for z = 1:gzn
      EVnext(i,z) = sum(Vnext(i,:).*gPz(z,:));
    end
  end
  % Bellman equation (maximization over next-period assets k')
  for i = 1:ggrid          % current k = gkv(i)
    for z = 1:gzn          % current productivity state

      incomej = gzv(z)*w + (1+r)*gkv(i); % income (cash-on-hand)

      % Feasible next k' indices
      % (ensure that c>=0 and k' >= k_min constraints are met)
      iimin = 1;
      iimax = min(floor(ggrid*(incomej/gkmax)), ggrid);
      if iimax < iimin, iimax = iimin; end
      sii = zeros(iimax-iimin+1,1);        % feasible s
      cii = 10^-5*ones(iimax-iimin+1,1);    % feasible c
      vii = -1e12*ones(iimax-iimin+1,1);    % feasible V
```

Aiyagari_model_v2.m - VFI + Gauss-Seidel III

```
      % Search for optimal k' decision, implied c and V
      for ii = iimin:iimax
        sii(ii) = gkv(ii);             % candidate for optimal k'
        cii(ii) = incomej - sii(ii);  % implied c
        vii(ii) = cii(ii)^(1-gsigma)/(1-gsigma) + gbeta*EVnext(ii,z);
      end

      % Extract optimal choice
      [val,pos] = max(vii); % find max position
      Sopt(i,z) = sii(pos); % optimal next-period asset holdings k'
      Copt(i,z) = cii(pos); % optimal consumption
      Vopt(i,z) = val;      % value function
    end
  end

  % Error between two iterations
  Vnext  = Vopt;              % Update Vnext for next iteration
  jerror = norm(Vnext - Vnow); % Check convergence
  Vnow   = Vopt;              % Update Vnow for next iteration
  jiter = jiter + 1;
end  % End of VFI for household optimal decision rules
```

Aiyagari_model_v2.m - VFI + Gauss-Seidel IV

```
% [3.] Stationary distribution given policy Sopt(k,z) and gPz
% -------------------------------------------------------
Mu1   = ones(ggrid,gzn)/(ggrid*gzn); % uniform initial mass
iter2 = 0; error2 = 10;

while (iter2 < 300) && (error2 > 10^-10)

  Mu2 = zeros(ggrid,gzn);
  % Find next-period mass (k',z') given (k,z)
  for i = 1:ggrid   % k
    for z = 1:gzn   % z
      % --- Next-period density of households holding k' assets ----
      % locate k' = Sopt(i,z) on the grid (linear interpolation)
      % -> find gkv(posL) and gkv(posH): gkv(posL) <= k' <= gkv(posH)
      idx = (Sopt(i,z)-gkmin)/ggridstep + 1;
      posL= max(1, min(floor(idx), ggrid-1));
      % adjust to correct position
      if gkv(posL) > Sopt(i,z)
          posL = posL-1;
      end
      posH  = min(posL+1, ggrid);
      % Assign weight s.t k' = (1-weight)*gkv(posL) + weight*gkv(posH)
      weight = (Sopt(i,z) - gkv(posL))/ggridstep;
      weight = max(0, min(1, weight));
```

Aiyagari_model_v2.m - VFI + Gauss-Seidel V

```
      % --- Next-period density of households with z' ----
      transp = Mu1(i,z) * gPz(z,:);

      % --- Next-period joint density of households with (k', z') ----
      % (based on density calculation above)
      for zz = 1:gzn
        Mu2(posL,zz) = Mu2(posL,zz) + (1-weight)*transp(zz);
        Mu2(posH,zz) = Mu2(posH,zz) + weight*transp(zz);
      end

    end
  end

  % Check convergence of distribution
  error2 = sum(sum(abs(Mu2 - Mu1)));
  Mu1    = Mu2 / sum(Mu2, 'all'); % update for next iteration
  iter2 = iter2 + 1;

end  % End of iteration for stationary distribution

Mu = Mu1; % Store stationary distribution
```

Aiyagari_model_v2.m - VFI + Gauss-Seidel VI

```
    % [4.] Aggregation, price update, and find fixed point for K
    % -----------------------------------------------------------
    Knew   = sum(sum(Mu .* Sopt)); % aggregate capital from policy
    ierror = abs((Knew - Kold)/max(1e-12,Kold)); % check convergence in K

    % Gauss-Seidel update on K (damps oscillations)
    K     = update*Knew + (1-update)*Kold; % convex updating
    Y     = gA*Knew^galpha * H^(1-galpha); % implied output

    % Update factor prices from firm FOCs
    w     = (1-galpha)*gA*K^galpha*H^(-galpha);
    q     = galpha*gA*K^(galpha-1)*H^(1-galpha);
    r     = q - gdelta;

    % Update/prepare for next iteration if convergence criterion not met
    Kold   = K;
    iiter = iiter + 1;

end % End of Gauss-Seidel iteration (Outer loop)
```

Aiyagari_model_v2.m - VFI + Gauss-Seidel VII

```
% Get cumulative distribution (CDF) and mean values
% --------------------------------------------------

Pjoin = Mu;              % joint distribution over (k,z)
% CDF over k by z type
[rs,cs] = size(Pjoin);
Pcumz   = zeros(rs,cs);
Pcumz(1,:) = Mu(1,:); %
% Create a cumulative distribution by productivity type
for z = 1:2
    for i = 2:ggrid
        Pcumz(i,z) = Pcumz(i-1,z) + Mu(i,z);
    end
end

Pdis   = sum(Mu, 2);      % Marginal distribution over k
% CDF over k
[rs, cs] = size(Pdis);
Pcum    = zeros(rs, cs);
Pcum(1) = Pdis(1);
for i = 2:ggrid,
    Pcum(i) = Pcum(i-1) + Pdis(i);
end
```

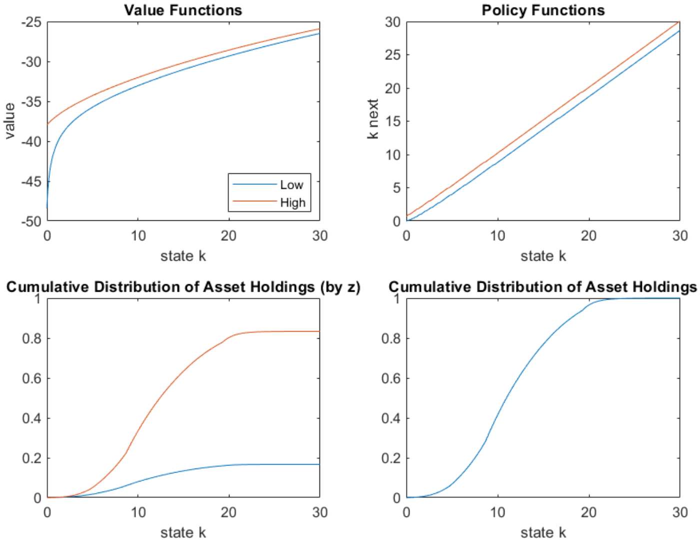
Aiyagari_model_v2.m - VFI + Gauss-Seidel VIII

```
% Mean value of consumption and welfare
C    = sum(sum(Mu .* Copt)); % average consumption
Welf = sum(sum(Mu .* Vopt)); % average value

disp('-------- Results ----------');
disp(['Y = ' num2str(Y)]);
disp(['K = ' num2str(K)]);
disp(['C = ' num2str(C)]);
disp(['w = ' num2str(w)]);
disp(['r = ' num2str(r)]);

% Figures (values, policy, cdf by z, cdf)
figure(1)
subplot(2,2,1);
plot(gkv,Vnow(:,1),'-',gkv,Vnow(:,2));
title('Value Functions');
xlabel('k'); ylabel('V(k,z)'); legend('Low','High');

subplot(2,2,2);
plot(gkv,Sopt(:,1),'-',gkv,Sopt(:,2));
title('Policy k''(k,z)');
xlabel('k'); ylabel('k''');
```

Aiyagari_model_v2.m - VFI + Gauss-Seidel IX

```
subplot(2,2,3);
plot(gkv,sum(Mu,2));
title('Distribution (PDF)');
xlabel('k');

subplot(2,2,4);
plot(gkv,Pcum);
title('Distribution (CDF)');
xlabel('k');

toc
```

Aiyagari - Value Function, Policy and Distributions



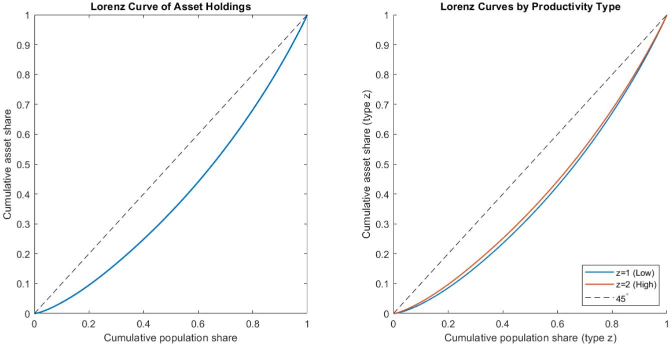
Aiyagari_model_v2.m - VFI + Gauss-Seidel X

```
% PLOT LORENZ CURVES
% Get cumulative population share (sorted by k) (x axis)
Pdis = sum(Mu, 2);
pop_cum = cumsum(Pdis);          % in [0,1]

% Get cumulative share of wealth/assets (y axis)
asset_mass = gkv(:) .* Pdis; % Wealth x mass at each k (grid point)
A_tot = sum(asset_mass);     % Total assets
asset_cum = cumsum(asset_mass) / A_tot;

% Anchor with (0,0)
xL = [0; pop_cum];    yL = [0; asset_cum];

% Lorenz curve (all type)
figure(2);
subplot(1,2,1);
plot(xL, yL, 'LineWidth', 1.5); hold on;
plot([0 1], [0 1], 'k--'); hold off;
xlim([0 1]); ylim([0 1]);
xlabel('Cumulative population share');
ylabel('Cumulative asset share');
title('Lorenz Curve of Asset Holdings');
```

Aiyagari_model_v2.m - VFI + Gauss-Seidel XI

```
% Per-type Lorenz curves (conditional within each z)
subplot(1,2,2);
hold on;
for z = 1:2
    Pz = Mu(:,z);
    if sum(Pz) > 0
        Pz = Pz / sum(Pz);
        xz = [0; cumsum(Pz)]; % Cumulative share of population
        yz = [0; cumsum(gkv(:).*Pz) / sum(gkv(:).*Pz)]; % Cumulative
             share of assets
        plot(xz, yz, 'LineWidth', 1.2);
    end
end
plot([0 1], [0 1], 'k--'); hold off;
xlim([0 1]); ylim([0 1]);
xlabel('Cumulative population share (type z)');
ylabel('Cumulative asset share (type z)');
title('Lorenz Curves by Productivity Type');
legend('z=1 (Low)','z=2 (High)','45^\circ','Location','SouthEast');

% Gini from the Lorenz curve
% (Gini = 1 - 2 * area under Lorenz curve)
area_L = trapz(xL, yL);  % trapezoidal area under Lorenz
Gini   = 1 - 2*area_L;
fprintf('Gini (assets) = %.4f\n', Gini);
```

Aiyagari - Lorenz Curves (Wealth Inequality)



Remark: The wealth Gini implied by our baseline model is 0.225 (really small). Empirical estimates for advanced economies typically range between 0.5 and 0.8.

APPENDIX

Household III: Optimality conditions (1/2)

Assuming an interior solution ($b' > \underline{b}$), the Euler equation is

$$u'(c_t) \;=\; \beta R\, \mathbb{E}_t[u'(c_{t+1})]\,.$$

With CRRA utility $u'(c) = c^{-\sigma}$:

$$c_t^{-\sigma} \;=\; \beta R\, \mathbb{E}_t\left[c_{t+1}^{-\sigma}\right]\,.$$

Complementary slackness (borrowing constraint).

$$h_t \geq 0, \quad b_{t+1} - \underline{b} \geq 0, \quad h_t\,(b_{t+1} - \underline{b}) = 0,$$

and the Euler becomes

$$u'(c_t) = \beta R\, \mathbb{E}_t[u'(c_{t+1})] + h_t$$

▸ Back to Main Section

Household III: Optimality conditions (2/2)

- When the constraint binds ($h_t > 0$) $\Rightarrow u'(c_t) \geq \beta R\, \mathbb{E}_t[u'(c_{t+1})]$: households *want* to borrow to raise consumption but cannot.
- Since $c^{-\sigma}$ is convex for all $c > 0$ when $\sigma > 0$, by Jensen's inequality:

$$\mathbb{E}[c_{t+1}^{-\sigma}] \geq (\mathbb{E}[c_{t+1}])^{-\sigma}$$

  Thus, holding the mean of c_{t+1} fixed, any increase spread (risk) around the mean or an increase in σ makes the expected marginal utility of c_{t+1} larger ($\mathbb{E}_t[c_{t+1}^{-\sigma}] \uparrow$). This leads households to save more (buffer stocks).
- With $\beta R < 1$, the impatience condition prevents explosive saving paths and ensures a stationary policy. To see this, consider a no-risk example

$$u'(c_t) = \beta R u'(c_{t+1})$$

  $\beta R < 1$ mean $u'(c_t) < u'(c_{t+1}) \implies c_t > c_{t+1}$. Thus, the household does not want to keep increasing assets forever (the present is weighted more heavily). If $\beta R \geq 1$, saving is so attractive that there is a tendency to push assets upward without bound, leading to explosive savings path.

Note

In this model, two forces meet: (i) Risk creates precautionary savings motive, pushing wealth up; (ii) Impatience places more weight on today's consumption, pushing wealth down. This tug-of-war yields a finite wealth-income ratio and a stable consumption/saving path.

Huggett Algorithm - Value function iteration I

[1] Discretize asset space (current bonds).

$$b_t \equiv b^i \in \mathcal{B} = \{b_{\min} = b^1, \ldots, b^N = b_{\max}\}, \qquad \mathbf{b} = \begin{bmatrix} b^1 \\ \vdots \\ b^N \end{bmatrix} \in \mathbb{R}^{N\times 1}.$$

Income is two-state: $y \in \{y^L, y^H\}$ with transition

$$\Pi = \begin{bmatrix} \pi_{LL} & \pi_{LH} \\ \pi_{HL} & \pi_{HH} \end{bmatrix}.$$

[2] Guess initial value functions (one vector per y)

$$V_L^{(1)}(\mathbf{b}) = v_0 \mathbf{1}_N, \qquad V_H^{(1)}(\mathbf{b}) = v_0 \mathbf{1}_N, \quad \text{e.g. } v_0 = 0.1.$$

So explicitly

$$V_s^{(1)}(\mathbf{b}) = \begin{bmatrix} 0.1 \\ \vdots \\ 0.1 \end{bmatrix}, \qquad s \in \{L, H\}.$$

Note: Each $V_s^{(k)}(\mathbf{b})$ is an $N \times 1$ column: its ith entry is $V_s^{(k)}(b^i)$.

Huggett Algorithm - Value function iteration II

[3]: Choice set for next-period assets (the b' grid).

$$b_{t+1} \equiv b'^j \in \mathcal{B}' = \mathcal{B}, \qquad \mathbf{b}' = \begin{bmatrix} b'^1 = b_{\text{min}} \\ \vdots \\ b'^N = b_{\text{max}} \end{bmatrix}.$$

Cash on hand and feasible consumption vector for each pair (b^i, y^s).

$$m(b^i, y^s) = y^s + Rb^i, \qquad \mathbf{c}(b^i, y^s) = \underbrace{m(b^i, y^s)}_{\text{scalar}} \mathbf{1}_N - \underbrace{\mathbf{b}'}_{N \times 1}.$$

The consumption vector for (b^i, y^s) is therefore:

$$\mathbf{c}(b^i, y^s) = \begin{bmatrix} c^1 \\ \vdots \\ c^N \end{bmatrix} = \begin{bmatrix} m(b^i, y^s) - b'^1 \\ \vdots \\ m(b^i, y^s) - b'^N \end{bmatrix}, \quad c^j > 0, \ \ b'^j \geq \underline{b}.$$

Note: Entries with $c^j \leq 0$ are ruled out (or given a large negative utility penalty).

Huggett Algorithm - Value function iteration III

[4]: Iterate on Bellman operator.

Calculate utility for every (b^i, y^s) combination:

$$\mathbf{u}(b^i, y^s) = u\big(\mathbf{c}(b^i, y^s)\big) = \begin{bmatrix} u(c^1) \\ \vdots \\ u(c^N) \end{bmatrix}.$$

Expected continuation value vector at iteration k for $s \in \{L, H\}$:

$$\mathbf{EV}_s^{(k)}(\mathbf{b}') \;=\; \pi_{sL}\, V_L^{(k)}(\mathbf{b}') + \pi_{sH}\, V_H^{(k)}(\mathbf{b}') = \begin{bmatrix} \pi_{sL} V_L^{(k)}(b'^1) + \pi_{sH} V_H^{(k)}(b'^1) \\ \vdots \\ \pi_{sL} V_L^{(k)}(b'^N) + \pi_{sH} V_H^{(k)}(b'^N) \end{bmatrix}.$$

Bellman update at (b^i, y^s).

$$\mathbf{v}^{(k+1)}(b^i, y^s) \;=\; \mathbf{u}(b^i, y^s) \;+\; \beta\, \mathbf{EV}_s^{(k)}(\mathbf{b}') \;\in \mathbb{R}^{N\times 1}.$$

Huggett Algorithm - Value function iteration IV

Maximization and argmax index j^*.

$$V_s^{(k+1)}(b^i) = \max_{j \in \{1,\dots,N\}} \mathbf{v}_s^{(k+1)}(b^i)[j], \qquad j^\star = \arg\max_j \mathbf{v}_s^{(k+1)}(b^i)[j].$$

Policies:

$$g_b(b^i, y^s) = b'[j^\star], \qquad g_c(b^i, y^s) = m(b^i, y^s) - b'[j^\star].$$

[5] Convergence of values.

$$\|V^{(k+1)} - V^{(k)}\| \ < \ \varepsilon.$$

Huggett Algorithm - Stationary Distribution I

[1] Stationarity condition. The distribution of the individual state variable (y, b) satisfies

$$\mu(y', b') = \sum_y \sum_b \mathbf{1}\{b' = g_b(b, y)\}\pi(y'|y)\,\mu(y, b),$$

where $\mu(y, b)$ denotes the joint distribution of agents over current income y and asset holdings b.

[2] Representing $\mu(b, y)$ as an $N \times 2$ matrix:

$$\mu(b, y) = \begin{bmatrix} \mu_{1,1} & \mu_{1,2} \\ \vdots & \vdots \\ \mu_{i,1} & \mu_{i,2} \\ \vdots & \vdots \\ \mu_{N,1} & \mu_{N,2} \end{bmatrix}_{N\times 2} ,$$

columns: income (y^L, y^H), rows: asset grid $b^1, \dots, b^N$.
Note: Each cell $\mu_{i,s}$ represents the fraction of agents with income y^s and current bond b^i.

Huggett Algorithm - Stationary Distribution II

[3] Propagate distribution forward.

Let $\mu_{\text{now}}(b, y)$ denote the current density and $\mu_{\text{next}}(b', y')$ the next-period density.

$$\mu_{\text{now}}(b, y) = \begin{bmatrix} \mu_{1,1} & \mu_{1,2} \\ \vdots & \vdots \\ \mu_{i,1} & \mu_{i,2} \\ \vdots & \vdots \\ \mu_{N,1} & \mu_{N,2} \end{bmatrix} \xrightarrow[\text{via } \pi(y'|y) \text{ and } b'=g_b(b,y)]{\text{transition}} \mu_{\text{next}}(b', y') = \begin{bmatrix} \mu'_{1,1} & \mu'_{1,2} \\ \vdots & \vdots \\ \mu'_{i,1} & \mu'_{i,2} \\ \vdots & \vdots \\ \mu'_{N,1} & \mu'_{N,2} \end{bmatrix}$$

[4] Iterate to stationarity. Repeat until the distribution converges:

$$\|\mu^{(t+1)} - \mu^{(t)}\| < \varepsilon_\mu.$$

Remark: Convergence $\Rightarrow$ inflows $=$ outflows for every state $\Rightarrow$ stationary cross-sectional distribution.

Back to Main Section

Aiyagari Household IV: FOCs and envelope

Euler conditions, combining interior (non-binding constraint, $k_{t+1} > k_{min}$) and complementary slackness (binding constraint, $k_{t+1} = k_{min}$):

$$c_t^{-\sigma} \;\geq\; \beta\, R_t\, E\left[c_{t+1}^{-\sigma} \mid z_t\right], \qquad k_{t+1} \geq k_{\min},$$

$$\left(c_t^{-\sigma} - \beta R_t E[c_{t+1}^{-\sigma} \mid z_t]\right)\left(k_{t+1} - k_{\min}\right) = 0.$$

Intuition: interior solution arises when savings are above the borrowing limit; otherwise, the borrowing constraint binds and the Euler inequality is strict.

Envelope.

$$V_k(k_t, z_t) = R_t\, c_t^{-\sigma}.$$

Back to Main Section

Aiyagari Computational Method I - VFI + Gauss-Seidel

Step 1. Discretize the household's state space.

- Set a lower and upper bound for assets/capital $k \in [k_{\min}, k_{\max}]$.
- Discretize state space: $k = \{k_{\min}, \dots, k_{\max}\}$ and idiosyncratic productivity $z \in \{z^L, z^H\}$ with transition matrix Π.

Step 2. Guess aggregate capital $K^{(0)}$.

- Given $K^{(0)}$, compute factor prices using the firm's FOCs:

$$r^0 = \alpha A\left(\frac{H}{K^{(0)}}\right)^{1-\alpha} - \delta, \qquad w^0 = (1-\alpha)A(K^{(0)})^{\alpha} H^{-\alpha}.$$

Remark: H is fixed (inelastic labour) $\rightarrow$ we do not need to guess $H^{(0)}$ here.

Step 3. Given r_0 and w_0, solve the household problem using VFI.

$$V(k,z) = \max_{k' \geq k_{\min}} \left\{u(c) + \beta E[V(k',z') \mid z]\right\}, \quad \text{s.t.} \quad c = w^0 z + (1+r^0)k - k'$$

- iterate the value function until convergence to obtain optimal decision rules $g_k(k,z)$ and $g_c(k,z)$.

Aiyagari Computational Method II - VFI + Gauss-Seidel

Step 4. Compute the stationary distribution

- Given $g_k(k, z)$ and $\Pi(z'|z)$, update the distribution:

$$\mu^{(t+1)}(k', z') = \sum_z \sum_k \mathbf{1}\{k' = g_k(k, z)\}\Pi(z'|z)\mu^{(t)}(k, z).$$

- Iterate until $\mu^{(t+1)} = \mu^{(t)}$.

Step 5. Aggregation.

- Compute aggregate capital, labor, and consumption:

$$K^{(1)} = \sum_k \sum_z \mu(k, z) g_k(k, z), \qquad H = \sum_k \sum_z \mu(k, z) z,$$

$$C = \sum_k \sum_z \mu(k, z) g_c(k, z).$$

- Compute implied prices (r^1, w^1)

Aiyagari Computational Method III - VFI + Gauss-Seidel

Step 6. Check market clearing

- If $|K^{(1)} - K^{(0)}| < \varepsilon_{GS}$, equilibrium found.
- Else, update:

$$K^{(1)} = (1-a)K^{(0)} + aK^{(1)}, \quad 0 < a < 1,$$

Step 7. Repeat **Steps 2-6** for $K^{(2)}, K^{(3)}, \ldots, K^{(iter_{max})}$ until convergence in K (capital market clears).

Back to Main Section