Lecture 8

Rep vs Heterog Households: Key Differences
The Income Fluctuation Problem

Macroeconomics EC417

Jane Olmstead-Rumsey

London School of Economics, Fall 2022

Just so everyone is on board: vocabulary

What do the following words mean (when used in economics)?

1. deterministic
2. stochastic
3. idiosyncratic
4. i.i.d.
5. rational
6. rational expectations
7. partial equilibrium
8. general equilibrium
9. ... what else?

Plan for remaining lectures

1. Income fluctuation problem a.k.a. consumption-saving problem with idiosyncratic labor income risk in partial equilibrium
2. Numerical dynamic programming a.k.a. numerical solution of Bellman equations
   - numerical solution of income fluctuation problem
3. Textbook heterogeneous agent model: Aiyagari-Bewley-Huggett
   - income fluctuation problem, embedded in general equilibrium
4. Further directions
   - business cycles with heterogeneous agents (idiosyncratic + aggregate risk): Den Haan & Krusell-Smith
   - Heterogeneous Agent New Keynesian (HANK) models
   - Firm heterogeneity in macroeconomics

Useful references & resources – see syllabus for more

- Key papers in literature
  - Aiyagari (1994)
  - Huggett (1993)

- Textbook treatment: Ljungqvist-Sargent "Recursive Macroeconomic Theory"
  - Part IV "Savings Problems and Bewley Models"

- Other computational resources
  - Matlab codes we will go over in lectures/posted on moodle
  - `http://quantecon.org/`, esp. Aiyagari model codes:
    Python: `https://python.quantecon.org/aiyagari.html`
    Julia: `https://julia.quantecon.org/multi_agent_models/aiyagari.html`

Plan for Today

1. Quick summary of workhorse representative agent model: the growth model

2. Key differences between representative and heterogeneous agent models

3. Deterministic consumption-saving problem

4. Tools: Bellman equations (dynamic programming)

5. The income fluctuation problem = key building block of workhorse het agent model

Summary of Growth Model

Growth Model in Discrete Time

- Preferences: representative household with utility function

$$\sum_{t=0}^{\infty} \beta^t u(c_t)$$

- Technology:

$$y_t = f(k_t), \quad c_t + i_t = y_t$$
$$k_{t+1} = i_t + (1-\delta)k_t, \quad c_t \geq 0$$

- Endowments: $\hat{k}_0$ units of capital at $t = 0$

- Pareto optimal allocation solves

$$V(\hat{k}_0) = \max_{\{c_t\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t u(c_t) \quad \text{s.t.}$$
$$k_{t+1} = f(k_t) + (1-\delta)k_t - c_t, \quad k_0 = \hat{k}_0$$

Growth Model in Continuous Time

- Preferences: representative household with utility function

$$\int_0^\infty e^{-\rho t} u(c(t))dt$$

$\rho \geq 0$ = discount *rate* (as opposed to β = discount *factor*)

- Technology:

$$y(t) = f(k(t)), \quad c(t) + i(t) = y(t)$$
$$\dot{k}(t) = i(t) - \delta k(t), \quad c(t) \geq 0, \quad k(t) \geq 0$$

- Endowments: $\hat{k}_0$ of capital at $t = 0$
- Pareto optimal allocation solves

$$V(\hat{k}_0) = \max_{\{c(t)\}_{t\geq 0}} \int_0^\infty e^{-\rho t} u(c(t))dt \quad \text{s.t.}$$
$$\dot{k}(t) = f(k(t)) - \delta k(t) - c(t), \quad k(0) = \hat{k}_0$$

Optimality Condition: Euler Equation

- Discrete time

$$\lambda_t = \beta\lambda_{t+1}(f'(k_{t+1}) + 1 - \delta) \quad \text{where} \quad \lambda_t = u'(c_t)$$

or equivalently

$$u'(c_t) = \beta u'(c_{t+1})(f'(k_{t+1}) + 1 - \delta)$$

- Continuous time

$$\dot{\lambda}(t) = (\rho + \delta - f'(k(t)))\lambda(t) \quad \text{where} \quad \lambda(t) = u'(c(t))$$

Steady State

- Steady state: "if you start there you stay there"
  - look for k^*, c^*, λ^* such that this is true, e.g. if $k_t = k^*$ then also $k_{t+1} = k^*$
  - in particular, in Euler equation set $\lambda_t = \lambda_{t+1}$ or $\dot{\lambda}(t) = 0$
- Discrete time: steady state capital stock solves

$$1 = \beta(f'(k^*) + 1 - \delta) \tag{DSS}$$

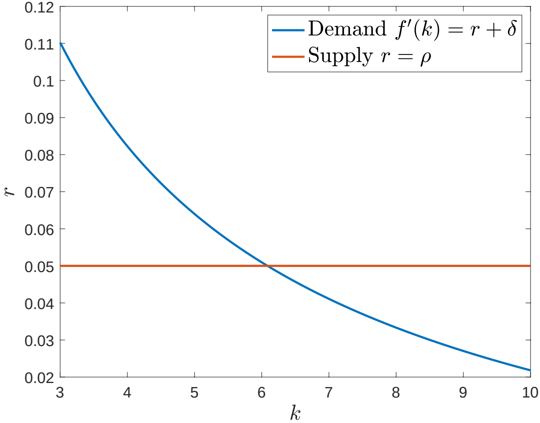
- Continuous time: steady state capital stock solves

$$f'(k^*) = \rho + \delta \tag{CSS}$$

- Note: this is the same equation
  - define discrete-time discount rate $\rho = 1/\beta - 1$
  - then (DSS) reduces to (CSS)

Infinitely-elastic steady state capital supply

- Important property of growth model
- See end of lecture notes 5 for explanation

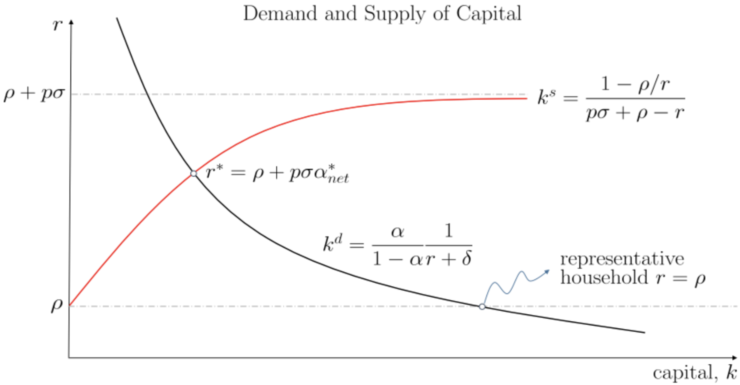


Rep vs Heterog Households: Key Differences

Four key differences between RA and HA models

1. Wealth distribution
   - RA: degenerate or indeterminate stationary distribution
   - HA: non-degenerate stationary distribution
2. Long-run capital supply
   - RA: infinite elasticity
   - HA: finite elasticity
3. Borrowing constraints, marginal propensity to consume (MPC)
   - RA: low MPCs
   - HA: potentially high MPCs
4. Welfare theorems
   - RA (for this point = growth model): typically hold
   - HA: typically do not hold

Key difference 2: long-run capital supply in HA models



Warmup:
Deterministic Consumption-Saving Problem

Deterministic Consumption-Saving Problem

- Consumption-saving decision of a single individual with a potentially time-varying income stream $\{y_t\}_{t=0}^{\infty}$

$$\max_{\{a_{t+1}\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t u(c_t) \quad \text{s.t.}$$
$$c_t + a_{t+1} \leq y_t + Ra_t$$
$$a_{t+1} \geq \underline{a}$$
$$0 = \lim_{T \to \infty} R^{-T} a_{T+1}$$

- Notation: $R = 1 + r$, will sometimes use the two interchangeably.

- Later: income fluctuation problem = same problem but with stochastic income y_t

Euler Equation Without Borrowing Constraint

- Ignore borrowing constraints: only impose No Ponzi condition
- Form Lagrangian:

$$\mathcal{L} = \sum_{t=0}^{\infty} \beta^t u(c_t) + \sum_{t=0}^{\infty} \beta^t \lambda_t [y_t + Ra_t - c_t - a_{t+1}]$$

- Note: scale Lagrange multipliers λ_t by β^t – can always do this
- First-order conditions:

$$u'(c_t) = \lambda_t \qquad [c_t]$$

$$\lambda_t = \beta R \lambda_{t+1} \qquad [a_{t+1}]$$

Implications of Euler Equation

- Standard form of Euler Equation

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

- If utility function strictly concave: $u''(c_t) < 0$

$$c_{t+1} = c_t \text{ if } \beta R = 1$$
$$c_{t+1} > c_t \text{ if } \beta R > 1$$
$$c_{t+1} < c_t \text{ if } \beta R < 1$$

- Intertemporal motive: when $\beta R \neq 1$

- Smoothing motive: when $y_t \neq y$

- Optimal solution also requires transversality condition

$$\lim_{T \to \infty} \beta^T u'(c_T) a_{T+1} = 0$$

CRRA Example

- Risk aversion coefficients:

$$\frac{-cu''(c)}{u'(c)} = \gamma(c)$$

- Constant Relative Risk Aversion (CRRA) utility

$$u(c) = \begin{cases} \frac{c^{1-\gamma}-1}{1-\gamma} & \text{if } \gamma \in (0,1), \gamma > 1 \\ \log c & \text{if } \gamma = 1 \end{cases}$$

$$u'(c) = c^{-\gamma}$$

- Without borrowing constraint, Euler equation implies

$$\begin{aligned} c_t^{-\gamma} &= \beta R c_{t+1}^{-\gamma} \\ c_t &= (\beta R)^{-\frac{1}{\gamma}} c_{t+1} \\ c_t &= (\beta R)^{\frac{t}{\gamma}} c_0 \end{aligned}$$

CRRA Example

- Combining budget constraints (BC)

$$\sum_{t=0}^{\infty} R^{-t} c_t = R a_0 + \sum_{t=0}^{\infty} R^{-t} y_t + \lim_{t\to\infty} R^{-T} a_{T+1}$$

- No-Ponzi condition implies last term = 0
- Substituting in for c_t

$$\sum_{t=0}^{\infty} \left(R^{\frac{1-\gamma}{\gamma}} \beta^{\frac{1}{\gamma}} \right)^{-t} c_0 = R a_0 + \sum_{t=0}^{\infty} R^{-t} y_t$$

$$c_0 = \mathfrak{m}\left(\beta, R, \gamma\right) \left(R a_0 + \sum_{t=0}^{\infty} R^{-t} y_t \right), \quad \mathfrak{m}\left(\beta, R, \gamma\right) = 1 - R^{\frac{1-\gamma}{\gamma}} \beta^{\frac{1}{\gamma}}$$

- where $\mathfrak{m}\left(\beta, R, \gamma\right)$ is called the Marginal Propensity to Consume (MPC)

CRRA Example: insights

- If $\beta R = 1$:

$$\mathfrak{m}(\beta, R, \gamma) = 1 - \beta$$

- If $\gamma = 1$:

$$\mathfrak{m}(\beta, R, \gamma) = 1 - \beta$$

- Without borrowing constraints only PDV of income matters: "permanent income hypothesis," basis of Ricardian equivalence

- Smoothing motive: β_i, R_i basic driver of wealth inequality

Euler Equation with Borrowing Constraint

- Sequence of multipliers $\mu_t \geq 0$ for borrowing constraint

$$\mathcal{L} = \sum_{t=0}^{\infty} \beta^t u(c_t) + \sum_{t=0}^{\infty} \beta^t \lambda_t [y_t + Ra_t - c_t - a_{t+1}] + \sum_{t=0}^{\infty} \beta^t \mu_t [a_{t+1} - \underline{a}]$$

- First order conditions become

$$u'(c_t) = \lambda_t \qquad [c_t]$$

$$\lambda_t = \beta R \lambda_{t+1} + \mu_t \qquad [a_{t+1}]$$

- Substituting we get Euler Equation

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

Euler Equation with Borrowing Constraint

- Since $\mu_t \geq 0$, we rewrite as

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

- Two cases:

    1. Borrowing constraint does not bind $a_{t+1} > \underline{a}$, $\mu_t = 0$ so get Euler Equation holds with equality

    2. Borrowing constraint binds $a_{t+1} = \underline{a}$, Euler equation holds with strict inequality

Tools: Bellman Equations

Bellman Equation: Plan

- Abstract formulation using generic optimization problem from Lecture 2
- Application 1: deterministic consumption-saving problem
- Application 2: growth model
- Numerical solution
- See syllabus for more rigorous, abstract treatment
- Best treatment: Stokey-Lucas-Prescott (1989) "Recursive Methods in Economic Dynamics"

Bellman Equation: Some “History”

William Hamilton

Carl Jacobi

Richard Bellman

- Why called “dynamic programming”?
- Bellman: *“Try thinking of some combination that will possibly give it a pejorative meaning. It’s impossible. Thus, I thought dynamic programming was a good name. It was something not even a Congressman could object to. So I used it as an umbrella for my activities.”* `http://en.wikipedia.org/wiki/Dynamic_programming#History`

Dynamic Optimization: Control-State Formulation

- Recall from Lecture 2: pretty much all deterministic optimal control problems in discrete time can be written as

$$V(\hat{x}_0) = \max_{\{\alpha_t\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t r(x_t, \alpha_t)$$

subject to the law of motion for the state

$$x_{t+1} = g(x_t, \alpha_t) \text{ and } \alpha_t \in A, \quad x_0 = \hat{x}_0.$$

- $\beta \in (0,1)$: discount factor
- $x \in X \subseteq \mathbb{R}^m$: state vector
- $\alpha \in A \subseteq \mathbb{R}^k$: control vector
- $r : X \times A \to \mathbb{R}$: instantaneous return function

Bellman Equation

- Claim: the value function $V(\hat{x}_0)$ satisfies the Bellman equation

$$V(x) = \max_{\alpha} \left\{ r(x,\alpha) + \beta V(x') \quad \text{s.t.} \quad x' = g(x,\alpha) \right\}$$

- Notation: x' denotes tomorrow's state

- Important: calendar time has disappeared – "recursive notation"

- Proof sketch: consider value of optimal strategy $\{\alpha_t^*\}_{t=0}^{\infty}$

$$\begin{aligned} V(x_0) &= \sum_{t=0}^{\infty} \beta^t r(x_t, \alpha_t^*) \\ &= r(x_0, \alpha_0^*) + \sum_{t=1}^{\infty} \beta^t r(x_t, \alpha_t^*) \\ &= r(x_0, \alpha_0^*) + \beta \sum_{t=0}^{\infty} \beta^t r\left(x_{t+1}, \alpha_{t+1}^*\right) \\ &= r(x_0, \alpha_0^*) + \beta V(x_1) \end{aligned}$$

Application 1: Consumption-Saving Problem

- Assume that income is deterministic and constant $y_t = y$

$$\max_{\{a_{t+1}\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t u(c_t) \quad \text{s.t.}$$

$$c_t + a_{t+1} \leq y + Ra_t$$

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

- Recursive formulation of household problem: Bellman equation

$$V(a) = \max_{c,a'} \; u(c) + \beta V(a') \quad \text{s.t.}$$

$$c + a' \leq y + Ra$$

$$a' \geq \underline{a}$$

- Functional equation: solve for unknown function $V(a)$
- Arguments of value function are called state variables
- Solution is
  - Value function: $V(a)$
  - Policy functions: $c(a)$, $a'(a)$

Application 2: Growth Model

$$\max_{\{k_{t+1}\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t u(c_t) \quad \text{s.t.}$$

$$c_t + k_{t+1} \leq f(k_t) + (1-\delta)k_t$$

- How do you write the Bellman equation?

Application 1: Euler Equation from Bellman Equation

- Form Lagrangean:

$$\mathcal{L} = u(c) + \beta V\left(a'\right) + \lambda[y + (1+r)a - c - a'] + \mu[a' - \underline{a}]$$

- First order conditions with respect to c and a':

$$u'(c) = \lambda$$

$$\beta V'(a') = \lambda - \mu$$

- Envelope condition:

$$V'(a) = \lambda(1+r) \quad \Rightarrow \quad V'(a') = \lambda'(1+r)$$

- Substitute into FOC for a'

$$\lambda - \mu = \beta(1-r)\lambda'$$

- Using FOC for c

$$u'(c) = \beta(1+r)u'(c') + \mu$$

- Since $\mu \geq 0$ this is typically written as

$$u'(c) \geq \beta(1+r)u'(c')$$

Value Function Iteration

- Easiest method to numerically solve Bellman equation for $V(a)$
- Guess value function on RHS of Bellman equation then maximize to get value function on LHS
- Update guess and iterate to convergence
- Contraction Mapping Theorem: guaranteed to converge if $\beta < 1$
- We will learn other methods later, but this is simplest (and slowest)

Value Function Iteration – see `vfi_deterministic.m`

- Step 1: Discretized asset space $\mathcal{A} = \{a_1, a_2, \ldots, a_N\}$. Set $a_1 = \underline{a}$
- Step 2: Guess initial $V_0(a)$. Good guess is

$$V_0(a) = \sum_{t=0}^{\infty} \beta^t u(ra + y) = \frac{u(ra + y)}{1 - \beta}$$

- Step 3: Set $\ell = 1$. Loop over all $\mathcal{A}$ and solve

$$\begin{aligned} a'_{\ell+1}(a_i) &= \arg\max_{a' \in \mathcal{A}} u\left(y + (1+r)\, a_i - a'\right) + \beta V_\ell\left(a'\right) \\ V_{\ell+1}(a_i) &= \max_{a' \in \mathcal{A}} u\left(y + (1+r)\, a_i - a'\right) + \beta V_\ell\left(a'\right) \\ &= u\left(y + (1+r)\, a_i - a'_{\ell+1}(a_i)\right) + \beta V_\ell\left(a'_{\ell+1}(a_i)\right) \end{aligned}$$

Value Function Iteration – see vfi_deterministic.m

- Step 4: Check for convergence $\epsilon_\ell < \bar{\epsilon}$

$$\epsilon_\ell = \max_i |V_{\ell+1}(a_i) - V_\ell(a_i)|$$

  - if $\epsilon_\ell \geq \bar{\epsilon}$, go to Step 3 with $\ell := \ell + 1$
  - If $\epsilon_\ell < \bar{\epsilon}$, then
- Step 5: Extract optimal policy functions
  - $a'(a) = a_{\ell+1}(a)$
  - $V(a) = V_{\ell+1}(a)$
  - $c(a) = y + (1+r)a - a'(a)$
- Consumption function restricted to implied grid so not very accurate.

Time Subscripts on State Variable in Bellman Equation

- Sometimes people write

$$
\begin{aligned}
V(a_t) = \max_{c_t, a_{t+1}} \; & u(c_t) + \beta V(a_{t+1}) \quad \text{s.t.} \\
c_t + a_{t+1} &\leq y + R a_t \\
a_{t+1} &\geq \underline{a}
\end{aligned}
$$

- Kind of defeats the purpose

- Point is to remove calendar time and focus on where we are in the state space regardless of time period

Finite Horizon Dynamic Programming

- Value function depends on time t

$$V_t(a) = \max_{c,a'} u(c) + \beta V_{t+1}(a')$$

$$\text{subject to}$$

$$c + a' \leq y_t + (1+r)a$$

$$a' \geq \underline{a}$$

- Solution consists of sequence of value functions $\{V_t(a)\}_{t=0}^{T}$ and sequence of policy functions $\{c_t(a), a'_t(a)\}_{t=0}^{T}$
- Solve by backward induction. Last period:

$$a'_T(a) = 0$$

$$c_T(a) = y_T + (1+r)a$$

$$V_T(a) = u(y_T + (1+r)a)$$

- Why does the state variable a still not have a time subscript?
- Code: `vfi_deterministic_finite.m`

Income Fluctuation Problem
Stochastic Dynamic Programming

Sequence Formulation

- Sequence Formulation of household problem

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

$$\text{subject to}$$

$$c_t + a_{t+1} \leq y_t + (1+r)\, a_t$$

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

$$a_0 \text{ given}$$

where $c_t(y^t), a_t(y^t)$ are endogenous choices with $y^t \equiv \{y_0, y_1, \dots, y_t\}$

- Assume y_t is a Markov Process: CDF F satisfies

$$F(y_{t+1}|y^t) = F(y_{t+1}|y_t)$$

Recursive Formulation

- Bellman equation for household problem

$$\begin{aligned} V(a, y) &= \max_{c, a'} u(c) + \beta \mathbb{E}\left[V(a', y') | y\right] \\ &\quad \text{subject to} \\ c + a' &\leq y + Ra \\ a' &\geq \underline{a} \end{aligned}$$

- Solution consists of
  - Value function: $V(a, y)$
  - Policy functions: $c(a, y)$, $a'(a, y)$

Cash-on-hand State Variable

- When y is IID, can define cash-on-hand x

$$x = y + Ra$$

- Bellman equation becomes

$$V(x) = \max_{c,s} u(c) + \beta \mathbb{E}\left[V\left(Rs + y'\right)\right]$$

$$\text{subject to}$$

$$c + s \leq x$$

$$s \geq \underline{a}$$

- Solution consists of
  - Value function: $V(x)$
  - Policy functions: $c(x)$, $a'(x)$

Stochastic Euler Equation

- We form Lagrangian

$$V(a,y) = \max_{c,a'} u(c) + \beta \mathbb{E}\left[V\left(a', y'\right)|y\right] + \lambda\left[y + (1+r)a - c - a'\right] + \mu\left[a' - \underline{a}\right]$$

$$s.t.\ \mu \geq 0,\ \ \lambda \geq 0$$

- FOC are

$$u'(c) = \lambda \qquad [c]$$

$$\beta \mathbb{E}\left[V_a\left(a', y'\right)|y\right] = \lambda - \mu \qquad [a']$$

- Envelope condition

$$V_a(a,y) = \lambda(1+r)$$

$$V_a\left(a', y'\right) = \lambda'(1+r)$$

Stochastic Euler Equation

- Using FOC for a' and envelope condition

$$\lambda - \mu = \beta (1 + r) \mathbb{E} \left[\lambda' | y \right]$$

- Using FOC for c

$$u' (c) = \beta (1 + r) \mathbb{E} \left[u' \left(c' \right) | y \right] + \mu$$

- Since $\mu \geq 0$, Euler Equation (EE) is

$$u' (c) \geq \beta (1 + r) \mathbb{E} \left[u' \left(c' \right) | y \right] \qquad \text{[EE]:}$$

- Notes:
  - Expectation is conditional on all information at t
  - Borrowing constraint binds $\implies$ EE strict inequality
  - Borrowing constraint not binding $\implies$ EE equality

Discrete-State Markov Process for Income

- Finite number of income realizations: $y \in \{y_1, \cdots, y_J\}$
- $\mathbf{P}$ is Markov transition matrix where
  - (j, j')th element of $\mathbf{P}$ is $\Pr(y_{t+1} = y_{j'} | y_t = y_j) = p_{jj'}$
  - $\forall\ j, j' \quad p_{jj'} \in [0, 1]$
  - $\forall j, \quad \sum_{j'=1}^{J} p_{jj'} = 1$
- Stationary distribution is vector π with elements π_j
  - solves
    $$\pi = \mathbf{P}^\top \pi, \qquad \mathbf{P}^\top = \text{transpose of } \mathbf{P}$$
    (Eigenvalue problem = same form as $\mathbf{Av} = \lambda \mathbf{v}$ with $\lambda = 1$; Equivalently row vector $\tilde{\pi}$ s.t. $\tilde{\pi} = \tilde{\pi}\mathbf{P}$)
  - easy method for finding π in practice: take N large, some π_0
    $$\pi \approx (\mathbf{P}^\top)^N \pi_0$$
  - Logic: $\pi_{t+1} = \mathbf{P}^\top \pi_t$ and hence $\pi \approx \pi_N = (\mathbf{P}^\top)^N \pi_0$

Bellman Equation with Discrete-State Markov Process

$$V\left(a, y_j\right) = \max_{c, a'} u\left(c\right) + \beta \sum_{j'=1}^{J} V\left(a', y_{j'}\right) p_{jj'}$$

$$\text{subject to}$$

$$c + a' \leq y_j + \left(1 + r\right) a$$

$$a' \geq \underline{a}$$

- Euler Equation is

$$u'\left(c\left(a, y_j\right)\right) = \beta\left(1 + r\right) \sum_{j'=1}^{J} u'\left(c\left(a', y_{j'}\right)\right) p_{jj'}$$

with $a' = y_j + (1 + r)a - c(a, y_j)$

- Solution is set of J functions $c(a, y_j)$

Value Function Iteration – see `vfi_IID.m`

- Step 1: Discretized asset space $\mathcal{A} = \{a_1, a_2, \ldots, a_N\}$. Set $a_1 = \underline{a}$
- Step 2: Guess initial $V_0(a, y_j)$. Reasonable first guess is

$$V_0\left(a, y\right) = \sum_{t=0}^{\infty} \beta^t u\left(ra + y\right) = \frac{u\left(ra + y\right)}{1 - \beta}$$

- Step 3: Set $\ell = 1$. Loop over all $a_i \in \mathcal{A}$ and solve

$$a'_{\ell+1}\left(a_i, y_j\right) = \arg\max_{a' \in \mathcal{A}} u\left(y_j + \left(1 + r\right) a_i - a'\right) + \beta \sum_{j'=1}^{J} V_\ell\left(a', y_{j'}\right) p_{jj'}$$

$$V_{\ell+1}\left(a_i, y_j\right) = \max_{a' \in \mathcal{A}} u\left(y_j + \left(1 + r\right) a_i - a'\right) + \beta \sum_{j'=1}^{J} V_\ell\left(a', y_{j'}\right) p_{jj'}$$

$$= u\left(y_j + \left(1 + r\right) a_i - a'_{\ell+1}\left(a_i, y_j\right)\right) + \beta \sum_{j'=1}^{J} V_\ell\left(a'_{\ell+1}\left(a_i, y_j\right)\right.$$

Value Function Iteration – see `vfi_IID.m`

- Step 4: Check for convergence $\epsilon_\ell < \bar{\epsilon}$

$$\epsilon_\ell = \max_{i,j} \left| V_{\ell+1}\left(a_i, y_j\right) - V_\ell\left(a_i, y_j\right) \right|$$

  - If $\epsilon_\ell \geq \bar{\epsilon}$, go to Step 3 with $\ell := \ell + 1$
  - If $\epsilon_\ell < \bar{\epsilon}$, then

- Step 5: Extract optimal policy functions
  - $a'(a, y) = a_{\ell+1}(a, y)$
  - $V(a, y) = V_{\ell+1}(a, y)$
  - $c(a, y) = y + (1 + r)a - a'(a, y)$
- Consumption function restricted to implied grid so not very accurate

Finding the Stationary Distribution

Method 1: Stationary Distribution via Simulation

- Step 1: Set seed of random number generator
- Step 2: Initialize array to hold consumption c_{it} and assets a_{it} for large number I of individuals and time periods T
- Step 3: Loop over agents i, draw y_{i0} from stationary distribution. Set $a_{i0} = 0$
- Step 4: Loop over all time periods t. Use policy function $a'(a, y)$ to compute next period assets $a_{i,t+1}$ for each agent. Use budget constraint to get implied c_{it}. Draw $y_{i,t+1}$ using Markov chain P.
- Step 5: Compute mean asset holdings as

$$A_t = \frac{1}{I} \sum_{i=1}^{I} a_{it}$$

  and check that A_t has converged
- Code: see 2nd part of `vfi_IID.m`

Method 2: Stationary Distribution via Transition Matrix

- Simulation often bad idea bc slow and introduces numerical error
- Now: preferred method that avoids simulation
- Recall: stationary distribution π of income process y solves

$$\pi = \mathbf{P}^\top \pi \qquad \text{or} \qquad \pi \approx (\mathbf{P}^\top)^N \pi_0 \quad \text{for large } N$$

- Idea of method 2: form big transition matrix of joint (a, y) process, let's call it $\mathbf{B}$, and use same strategy
- Step 1: Fix point in grid (a_i, y_j). For all possible grid points $a_{i'}, y_{j'}$ (important: all $a_{i'}$ forced to be on grid $\mathcal{A} = \{a_1, ..., a_N\}$) compute

$$\Pr(a_{t+1} = a_{i'}, y_{t+1} = y_{j'} | a_t = a_i, y_t = y_j)$$

- Can do this by interpolation of policy function $a'(a_i, y_j)$
- Step 2: Stack! 1. Stack grids for a (dim $= N$) and y (dim $= J$) into large $K = N \times J$ grid. Stack Pr's into big matrix $K \times K$ matrix $\mathbf{B}$

Something useful to think about

- We solved for wealth dist of economy with large number of people (say simulation with $N = 100,000$ to approximate continuum)
- How many Bellman equations did we solve?
- Why?

More Advanced Methods and Useful Tricks

More Advanced Methods and Useful Tricks

1. Euler equation iteration
   - see `eei_IID.m`
2. Power-spaced grids
   - used in all our codes I shared with you
3. Endogenous Grid Method
   - see `egp_IID.m`
   - if possible, always use this
4. Continuous-time methods: will teach this in my 2nd-year course
   - see codes here `https://benjaminmoll.com/codes/`, e.g. `http://www.princeton.edu/~moll/HACTproject/huggett_partialeq.m`

Euler Equation Iteration

- Step 1: Construct finite grid $\mathcal{A}$, $a_1 = \underline{a}$
- Step 2: Set $\ell = 0$. Guess initial $c_0(a_i y_j)$. Good first guess is

$$c_0(a_i, y_j) = ra + y$$

- Step 3: Loop over $\mathcal{A}$, solve for c by calculating LHS and RHS

$$u'(c) \geq \beta R \sum_{j'=1}^{J} u'\left(c_\ell\left[y_j + Ra_i - c, y_{j'}\right]\right) p_{jj'}$$

1. At borrowing constraint $a' = \underline{a} \implies c = Ra_i + y_j - \underline{a}$

$$\text{LHS} = u'\left(Ra_i + y_j - \underline{a}\right)$$

$$\text{RHS} = \beta R \sum_{j'=1}^{J} u'\left(c_\ell\left[\underline{a}, y_{j'}\right]\right) p_{jj'}$$

2. LHS $\leq$ RHS $\implies c_{\ell+1}\left(a_i, y_j\right) := Ra_i + y_j - \underline{a}$. Go to Step 4.
3. LHS $>$ RHS $\implies$ solve non-linear equation.

Euler Equation Iteration

- Step 3 (continued):
  - Construct interpolation function

$$EMUC\left(a', y_j\right) = \sum_{j'=1}^{J} u'\left(c\left(a', y_{j'}\right)\right) p_{jj'}$$

    which depends only on today's income. At (a_i, y_j) nonlinear equation becomes

$$u'(c) = \beta(1+r)\, EMUC\left((1+r)\, a_i + y_j - c, y_j\right)$$

  - Solve with non-linear solver: Matlab: `fzero` or `fsolve`, Python: `scipy.optimize.root` or `scipy.optimize.fsolve`

- Step 4: Stop if $\epsilon_\ell < \bar{\epsilon}$ and return policy functions, where

$$\epsilon_\ell = \max_{i,j} \left| c_{\ell+1}\left(a_i, y_j\right) - c_\ell\left(a_i, y_j\right) \right|$$

  If $\epsilon_\ell \geq \bar{\epsilon}$, go to Step 3 with $\ell := \ell + 1$

Power-spaced grids

- Policy functions are typically very non-linear close to the borrowing constraint
- Accurate linear interpolation with more grid points close to the constraint
- Let $[\underline{a}, \bar{a}]$ be the possible range of asset holdings.
- Let $\mathcal{Z}$ be an equi-spaced grid on $[0, 1]$.
- For each grid point $z \in \mathcal{Z}$, define $x = z^{\alpha}$ for some $\alpha \in (1, \infty)$ to create a non-linear spaced grid $\mathcal{X}$ on $[0, 1]$. Notice that as $\alpha \to \infty$, $\mathcal{X}$ has more and more points closer to 0.
- Construct asset grid $\mathcal{A}$ by rescaling each $x \in \mathcal{X}$

$$a = \underline{a} + (\bar{a} - \underline{a})x$$

Endogenous Grid Method

- Step 1: Construct grid $\mathcal{A}$ and set $a_1 = \underline{a}$
- Step 2: Set $\ell = 0$. Guess initial $c_0(a_i, y_j)$. A good first guess is

$$c_0\left(a_i, y_j\right) = ra + y$$

- Step 3: Construct implicit $c_\ell(a_i', y_{j'})$ via interpolating

$$\text{EMUC}_\ell\left(a_i', y_j\right) = \sum_{j'=1}^{J} u'\left(c_\ell\left(a_i', y_{j'}\right)\right) p_{jj'}$$

Use Euler equation at equality to get MUC today and c, a

$$\begin{aligned} \text{MUC}_\ell\left(a_i', y_j\right) &= \beta R \times \text{EMUC}_\ell\left(a_i', y_j\right) \\ \implies c_\ell\left(a_i', y_j\right) &= u'^{-1}\left(\text{MUC}_\ell\left(a_i', y_j\right)\right) \\ a_\ell\left(a_i', y_j\right) &= \frac{c_\ell\left(a_i', y_j\right) + a_i' - y_j}{1 + r} \end{aligned}$$

Invert $a_\ell(a_i', y_j) \implies a'(a, y_j)$ on an endogenous grid
Interpolate on $\mathcal{A}$ to get $a_{\ell+1}(a_i, y_i)$. Use BC to calculate $c_{\ell+1}$

Endogenous Grid Method

- Step 4: Deal with borrowing constraints: define $a^*\left(y_j\right) = a_\ell$. Then for $a_i > a^*(y_j)$, $a_i \in \mathcal{A}$

$$\begin{aligned} a_{\ell+1}\left(a_i, y_j\right) &:= \underline{a} \\ a_{j+1}\left(a_i, y_j\right) &:= (1+r)\, a_i + y_j - \underline{a} \end{aligned}$$

- Step 5: Stop if $\epsilon_\ell < \bar{\epsilon}$ and return policy functions, where

$$\epsilon_\ell = \max_{i,j} \left| c_{\ell+1}\left(a_i, y_j\right) - c_\ell\left(a_i, y_j\right) \right|$$

  If $\epsilon_\ell \geq \bar{\epsilon}$, go to Step 3 with $\ell := \ell + 1$

Endogenous Grid Points with Cash-on-Hand

- When income y is IID, single state variable is x
- Individual chooses consumption c, savings s s.t.

$$\begin{aligned} c + s &\leq x \\ s &\geq \underline{a} \end{aligned}$$

- Cash-on-hand x evolves as

$$x' = (1 + r)\, s + y'$$

Endogenous Grid Points with Cash-on-Hand

1. Discretize $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$, set $x_1 = R\underline{a} + y_{\min}$
   - Step 1.1: Discretize savings $\mathcal{S} = \{s_1, s_2, \ldots, s_N\}$, set $s_1 = \underline{a}$
2. Set $\ell = 0$. Guess $c_0(x_i),\ \forall x_i \in \mathcal{X}$. A good first guess is

$$c_0(x_i) = r x_i$$

3. Compute (via interpolation of $c(x)$ or $\mathrm{MUC}\,(x) \equiv u'\,(c\,(x))$)

$$\mathrm{EMUC}_\ell\,(s_i) = \sum_{j'=1}^{J} u'\left(c_\ell\left((1+r)\,s_i + y_j\right)\right) p_{j'}, \quad \forall s_i \in \mathcal{S}$$

4. Using EE at equality

$$\begin{aligned} \mathrm{MUC}_\ell\,(s_i) &= \beta R \times \mathrm{EMUC}_\ell\,(s_i) \\ \implies c_\ell\,(s_i) &= u'^{-1}\left(\mathrm{MUC}_\ell\,(s_i)\right) \\ x_\ell\,(s_i) &= s_i + c_\ell\,(s_i) \end{aligned}$$

5. Invert $x_\ell(s_i)$ by interpolating on $\mathcal{X}$, checking borr constraint
   Gives $s_{\ell+1}(x_i)$ which gives $c_{\ell+1} := x_i + s_{\ell+1}(x_i)$
6. Check for convergence. If fails, go to step 3