Lecture 2

Solving the Income Fluctuation Problem
Numerical Dynamic Programming

Macroeconomics EC442

Benjamin Moll

London School of Economics, Fall 2020

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 expectations
6. rational
7. partial equilibrium
8. general equilibrium
9. ... what else?

Plan for remaining lectures before break

1. Income fluctuation problem a.k.a. consumption-saving problem with idiosyncratic labor income risk in partial equilibrium
   - Ethan already covered this
2. Numerical dynamic programming a.k.a. numerical solution of Bellman equations
   - application: numerical solution of income fluctuation problem
3. Textbook heterogeneous agent model: Aiyagari-Bewley-Huggett
   - income fluctuation problem, embedded in general equilibrium
4. Perpetual youth model
5. Further directions
   - business cycles with heterogeneous agents (idiosyncratic + aggregate risk): Den Haan & Krusell-Smith
   - Heterogeneous Agent New Keynesian (HANK) models
   - Why is the wealth distribution so skewed?

Useful references & resources – see syllabus for more

- Key papers in literature
  - Aiyagari (1994)
  - Huggett (1993)
- Textbook treatment: Ljungqvist-Sargent "Recursive Macro Theory"
  - Part IV "Savings Problems and Bewley Models"
- Matlab, Python & Julia codes: `http://benjaminmoll.com/ha_codes/`
  (Note: .zip file, my Google Chrome tries to block download)
  - written by Greg Kaplan in Matlab
  - translated to Python & Julia by Tom Sweeney
- Other computational resources
  - `http://quant-econ.net/`, particularly Aiyagari model codes
    Python: `http://quant-econ.net/py/aiyagari.html`
    Julia: `http://quant-econ.net/jl/aiyagari.html`

The Income Fluctuation Problem

Income fluctuation problem: Overview

1. Individuals are subject to exogenous income shocks. These shocks are not fully insurable because of the lack of a complete set of Arrow-Debreu contingent claims

2. There is only a risk-free asset (i.e., and asset with non-state contingent rate of return) in which the individual can save/borrow, and that the individual faces a borrowing (liquidity) constraint

3. A continuum of such agents subject to different shocks will give rise to a wealth distribution

4. Integrating wealth holdings across all agents will give rise to an aggregate supply of capital

- Have already seen 1. and 2. in Ethan's part of the course

Liquidity Constraints

- Modify PIH so that household maximizes

$$\max_{a_{t+1}} E_0 \sum_{t=0}^{\infty} \beta^t u(c_t)$$
$$\text{s.t. } c_t + a_{t+1} \leq y_t + Ra_t$$
$$\text{and } a_{t+1} \geq \phi,$$

where ϕ is a liquidity constraint.

- Modified Euler equation

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

or

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

with strict inequality only if $a_{t+1} = \phi$.

The Income Fluctuation Problem

- Will change notation very slightly: $\underline{a}$ instead of ϕ

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

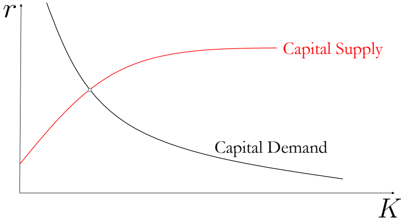
$$c_t + a_{t+1} \leq y_t + R a_t$$

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

- Will also interchangeably use R and $1 + r$ to denote gross return

What we'll do next

1. In general, no analytic solution $\Rightarrow$ learn how to solve income fluctuation problem on a computer
   - Bellman equation
   - wealth distribution generated by optimal saving behavior
2. "Close the model" i.e. embed the income fluctuation problem in general equilibrium, thereby endogenizing r
   - Different ways of doing this $\Leftrightarrow$ different assumptions on where capital demand comes from
   - Aiyagari: K demand of rep firm, Huggett: bonds w $K = 0$, ...



Deterministic Saving Problem
Deterministic Dynamic Programming

Saving Problem with Deterministic Income

- 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
- Arguments of value function are called state variables
- Solution is
  - Value function: $V(a)$
  - Policy functions: $c(a), \; a'(a)$

Euler Equation from Bellman Equation

- Form Lagrangean:

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

- 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 right until 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 2 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

$$V(a_t) = \max_{c_t, a_{t+1}} u(c_t) + \beta V(a_{t+1}) \quad \text{s.t.}$$

$$c_t + a_{t+1} \leq y + R a_t$$

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

- Please don't do this!!!

- Why not?

Finite Horizon Dynamic Programming

- Value function depends on time t

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

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

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

subject to

$$c_t + a_{t+1} \leq y_t + (1+r)\, a_t$$
$$a_{t+1} \geq \underline{a}$$
$$a_0 \text{ given}$$

- Assume y_t is a Markov Process: CDF F satisfies

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

where $y^t := \{y_0, y_1, \ldots, y_t\}$ denotes history of income realizations

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

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

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 $\Pr(y_{t+1} = y_{j'} | y_t = y_j) = p_{jj'}$
  - $\forall\, j, j' \;\; p_{jj'} \in [0, 1]$
  - $\forall j, \quad \sum_{j'=1}^{J} p_{jj'} = 1$
- 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 $\mathbf{A}\mathbf{v} = \lambda \mathbf{v}$ 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

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

subject to

$$c + a' \leq y_j + \left(1 + r\right) a$$
$$a' \geq \underline{a}$$

- Euler Equation is

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

- 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

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

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

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

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

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 2 with $\ell := \ell + 1$
  - If $\epsilon_\ell < \bar{\epsilon}$, then

- Step 5: Extract optimal policy functions
  - $a'(a, y) = a_{\ell+1}(a, y)$
  - $V(a, y) = V_{\ell+1}(a, y)$
  - $c(a, y) = y_i + (1 + r)a - a'(a, y)$
- 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 **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 **B**
- Step 3: Stat dist g, a $K \times 1$ vector w entries $g(a_i, y_j)$, solves

$$g = \mathbf{B}^\top g \qquad \text{or} \qquad g \approx (\mathbf{B}^\top)^N g_0 \quad \text{for large } N$$

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. `https://benjaminmoll.com/huggett_partialeq/`

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 $a_i > a^*(y_j)$, $a_i \in \mathcal{A}$

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

- 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

- Step 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 $s_1 = \underline{a}$
- Step 2: Set $\ell = 0$. Guess $c_0(x_i)$, $\forall x_i \in \mathcal{X}$. A good first guess is

$$c_0(x_i) = r x_i$$

- Step 3: Compute (via interpolation of $c(x)$ or $\mathrm{MUC}(x) \equiv u'(c(x))$)

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

- Step 4: Using EE at equality

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

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