Lecture 4
Policy Function Iteration

Peifan Wu

UBC

Income Fluctuation Problem

$$\begin{aligned} V(a,y) &= \max_{c,a'} u(c) + \beta \sum_{y' \in Y} \pi(y'|y) V(a',y') \\ c + a' &\leqslant Ra + y \\ a' &\geqslant \underline{a} \end{aligned}$$

Euler equation reads

$$u_c(Ra + y - a') - \beta R \sum_{y' \in Y} \pi(y'|y) u_c(Ra' + y' - a'') \geqslant 0$$

where the strict inequality holds when the constraint is binding

Policy function iteration (and improvements):

- PFI with linear interpolation
- Endogenous grid method
- Envelope condition method

Policy Function Iteration with Linear Interpolation

1. Construct a grid on the asset space $\{a_0, a_1, \ldots, a_m\}$ with $a_0 = \underline{a}$
2. Guess an initial vector of decision rules for a' on the grid points, call it $\hat{a}_0(a_i, y)$
3. For each point (a_i, y) on the grid, check whether the borrowing constraint binds, i.e.

$$u_c\left(Ra_i + y - \underline{a}\right) - \beta R \sum_{y' \in Y} \pi\left(y'|y\right) u_c\left(R\underline{a} + y' - \hat{a}_0\left(\underline{a}, y'\right)\right) > 0$$

4. If the inequality holds then the borrowing constraint binds. Set $\hat{a}_0(a_i, y) = \underline{a}$ in this case. Otherwise we have an interior solution and we proceed to the next step.

Policy Function Iteration with Linear Interpolation

5. For each point (a_i, y) on the grid, use a nonlinear equation solver to find the solution a^* of the nonlinear equation

$$u_c\left(Ra_i + y - a^*\right) - \beta R \sum_{y' \in Y} \pi\left(y'|y\right) u_c\left(Ra^* + y' - \hat{a}_0\left(a^*, y'\right)\right) = 0$$

   We need to evaluate the function $\hat{a}_0$ outside grid points: assume it is piecewise linear. Find the pair of adjacent grid points such that $a_i < a^* < a_{i+1}$ and then compute

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

Policy Function Iteration with Linear Interpolation

6. Check convergence by comparing $a_0' - \hat{a}_0$ through some pre-specified norm, e.g.,

$$\max\left\{\left|a_n'(a_i, y) - \hat{a}_n(a_i, y)\right|\right\} < \epsilon$$

7. Stop if convergence is achieved. Otherwise take new guess $\hat{a}_1 = a_0'$ and go back to 3.

The most time-consuming step is 5., the root-finding problem. By avoiding this step we can accelerate the process.

Endogenous Grid Method (EGM)

- EGM does not require the use of a nonlinear equation solver and hence much faster
- Idea: construct a grid on a', next period's asset holdings, rather than on a
- This method requires the policy function to be weakly monotonic in the state

EGM Algorithm

1. Construct a grid for (a, y) and guess a policy function $\hat{c}_0(a_i, y_j)$. If y is persistent, a good initial guess if $\hat{c}_0(a_i, y_j) = ra_i + y_j$ which is the solution under quadratic utility if income follows a random walk
2. Fix y_j. Instead of iterating over a_i we iterate over a'_i. For any pair $\{a'_i, y_j\}$ on the mesh, construct the RHS of the Euler equation

$$B\left(a'_i, y_j\right) \equiv \beta R \sum_{y' \in Y} \pi\left(y' | y_j\right) u_c\left(\hat{c}_0\left(a'_i, y'\right)\right)$$

   where the RHS of this equation uses the guess $\hat{c}_0$
3. Use the Euler equation to solve for the value $\tilde{c}\left(a'_i, y_j\right)$ that satisfies

$$u_c\left(\tilde{c}\left(a'_i, y_j\right)\right) = B\left(a'_i, y_j\right)$$

   Note that it can be done analytically: for $u_c(c) = c^{-\gamma}$, $\tilde{c}\left(a'_i, y_j\right) = \left[B\left(a'_i, y_j\right)\right]^{-\frac{1}{\gamma}}$, hence does not require a nonlinear solver

EGM Algorithm

4. From the budget constraint

$$\tilde{c}\left(a_i', y_j\right) + a_i' = Ra_i^* + y_j$$

   solve for a^* – the value of assets today that would lead to consumer to have a_i' assets tomorrow if her income shock was y_j today. This is the endogenous grid on a and it changes on each iteration

5. Let a_0^* be the value of asset holdings that induces the borrowing constraint to bind next period. Now we need to update our guess defined on the original grid by interpolation

Envelope Condition Method (ECM)

- An alternative method that, in some cases, avoids the use of nonlinear solvers
- Approximate the value function by polynomial interpolation, then we can evaluate the derivative of the value function
- Use envelope condition to pin down the policy functions

Envelope Condition Method (ECM)

1. Construct a grid on a with $n+1$ points, call it $\mathcal{A}^V$. Guess a value function $\hat{V}_0\left(a, y_j\right) = \sum_{k=0}^{n} w_{kj}^0 B_k\left(a\right)$ where B_k are piecewise linear splines

2. Define another grid on a of size $n+1$, call it $\mathcal{A}^D$. Compute the derivative of the value function on the nodes of $\mathcal{A}^D$

$$\tilde{V}_0'\left(a, y_j\right) = \sum_{k=0}^{n} w_{kj}^0 B_k'\left(a\right)$$

   The value function is interpolated by polynomials, and we can evaluate the derivative of the value function using the derivatives of interpolants

   We need to use a different grid because $\tilde{V}_0$ is not differentiable on the nodes of the original grid $\mathcal{A}^V$

Envelope Condition Method (ECM)

3. For any pair of (a, y) on $\mathcal{A}^D \times \mathcal{Y}$, apply envelope condition

$$\tilde{V}_0' \left(a_i, y_j\right) = u_c \left(Ra_i + y_j - \hat{a}_0 \left(a_i, y_j\right)\right) R$$

   from which we obtain

$$\hat{a}_0 \left(a_i, y_j\right) = Ra_i + y_j - u_c^{-1} \left(\frac{\hat{V}_0' \left(a_i, y_j\right)}{R} \right)$$

4. Update the value function from the Bellman equation with the new policy function. For each point of the grid $\mathcal{A}^V$, compute

$$\tilde{V}_1 \left(a_i, y_j\right) = u \left(Ra_i + y_j - \tilde{a}_0 \left(a_i, y_j\right)\right) + \beta \sum_{y_k \in Y} \pi_{jk} \tilde{V}_0 \left(\tilde{a}_0 \left(a_i, y_j\right), y_k'\right)$$

5. Check convergence