1 Notes on Endogenous Grid Method

This note outlines the endogenous grid algorithm used to solve the income fluctuation problem. The essential idea of the method is to construct a grid on a', next period's asset holdings, rather than on a, as is done in the standard algorithm. The method also requires the policy function for consumption to be at least weakly monotonic in current asset holdings.

Consider the following income fluctuation problem, with states (a, y) in its recursive formulation

$$\begin{aligned} V(a,y) &= \max_{a'} \{u(c) + \beta E[V(a',y')]\} \\ &\quad s.t. \\ c + a' &\leq Ra + y \\ a' &\geq a_{\min} \end{aligned}$$

Suppose that we have already discretized the income process so that we can write the Euler equation of the income fluctuation problem as follows:

$$\begin{aligned} u_c(c(a,y)) &\geq \beta R \sum_{y' \in Y} \pi(y'|y)\, u_c(c(a',y')) \\ &\quad s.t. \\ c + a' &= Ra + y \\ a' &\geq a_{\min} \end{aligned}$$

Equality holds if $a' > a_{\min}$. To solve the consumer problem means to find a decision rule for consumption $c(a, y)$, which is an invariant function of the states that satisfies the Euler Equation and that does not violate the borrowing constraint. We are going to start from a guess $c_0(a, y)$ and will iterate on the Euler Equation until the decision rule for consumption that we solve for is essentially identical to the one in the previous iteration. The algorithm involves the following steps:

1. Construct a grid on (a', y) where $a' \in G_A = \{a_1, ..., \bar{a}\}$ with $a_1 = a_{\min}$ and $y \in G_y = \{y_1, ..., y_N\}$. Notice here that we are defining our grid over assets tomorrow.
2. Guess a policy function $c_0(a_i, y_j)$. A good initial guess is to set $c_0(a_i, y_j) = Ra_i + y_j$
3. For any pair $\{a'_i, y_j\}$ on the mesh $G_A \times G_y$ construct the RHS of the Euler equation [call it $B(a'_i, z_j)$]
$$B(a'_i, y_j) = \beta(1+r) \sum_{y' \in Y} \pi(y'|y_j)\, u_c(c_0(a'_i, y'))$$
4. Use the Euler equation to solve for the value $\tilde{c}(a'_i, y_j)$ that satisfies
$$u_c(\tilde{c}(a'_i, y_j)) = B(a'_i, y_j)$$
and note that it can be done analytically, i.e. for $u_c(c) = c^{-\gamma}$ we have $\tilde{c}(a'_i, y_j) = [B(a'_i, y_j)]^{-\frac{1}{\gamma}}$. This is the key novel step that makes the algorithm much more efficient than the traditional one.

First, we do not require a nonlinear equation solver which takes a lot of computing time and which could introduce numerical instabilities. Also, here we only compute the expectation in step 3 once. In the traditional algorithm this is done by the nonlinear equation solver multiple times and it requires linear interpolation.

5. From the budget constraint, solve for $a^*(a_i', y_j)$ such that

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

which implicitly gives the function $c(a_i^*, y_j) = \tilde{c}(a_i', y_j)$. $a^*(a_i', y_j)$ is the value of assets today that would lead the consumer to have a_i' assets tomorrow if his income shock was y_j today. This is the endogenous grid and it changes on each iteration. Note that this function is not necessarily defined on the grid points of G_A. Let a_1^* be the value for current assets that induces the borrowing constraint to bind next period, i.e. the value a^* that solves that equation for a_1'.

6. Now we need to update our guess. To get the new guess $c_1(a_i, y_j)$ on grid points $a_i > a_1^*$ we can use simple interpolation methods using values for $c(a_i^*, y_j)$ on the two most adjacent values $\{a_n^*, a_{n+1}^*\}$ that include the given grid point a_i. To define the new guess of the consumption policy function on grid values $a_i < a_1^*$, then we use the budget constraint

$$c(a_i, y_j) = Ra_i + y_j - a_1'$$

since we cannot use the Euler equation as the borrowing constraint is binding.

7. Check convergence by comparing $c_{n+1}(a_i', y_j)$ to $c_n(a_i', y_j)$. For example declare convergence at iteration $n+1$ when

$$\max_{i,j} \{|c_{n+1}(a_i', y_j) - c_n(a_i', y_j)|\} < \varepsilon$$

for some small ε which determines the degree of tolerance in the solution algorithm.