Discussion Session 2: Analytical and Numerical Solutions for Value Function Iteration

Carlos Góes

September 2021

1 Characterizing the Solution

Recall our original dynamic problem:

$$
\begin{aligned}
\underbrace{v(K)}_{\text{state variables}} \quad &= \quad \underbrace{max_{C,K'}}_{\text{choice variables}} \; \log(C) + \beta V(K') \\
&s.t. \quad C + K' \leq AK^{\alpha}
\end{aligned}
$$

Recall that, by definition, the value function $v(K)$ reflects the optimal choices of C, K', given state variable K. How can we find the optimal points? You already know how to do that. We can set up a Lagrangian and take first order conditions to solve for the optimality conditions:

$$
\begin{aligned}
\mathscr{L} \quad &= \quad \log(C) + \beta v(K') + \Lambda[AK^{\alpha} - C - K'] \\
\textbf{FOCs:} & \\
C \quad &: \quad \frac{1}{C} = \Lambda \\
K' \quad &: \quad \Lambda = \beta v'(K') \\
CS \quad &: \quad \Lambda[AK^{\alpha} - C - K'] = 0
\end{aligned}
$$

In order to derive $v'(K')$ we take advantage of the an application of the Envelope Theorem for macroeconomics. Remember that the envelope theorem tells us that, as a parameter of the optimal value function changes slightly, there will be no indirect re-optimization effect, but rather only a direct effect over the objective function and the constraint. When applying the envelope theorem to dynamic programming problems, we call this the *Benveniste-Scheinkman Condition*.

Theorem 1 (Envelope Theorem). *Let $f(x, \alpha) : \mathbb{R}^2 \to \mathbb{R}$ be a continuously differentiable real-valued concave function, where x is the choice variable and α is parameter. If, for every $\alpha \in \mathbb{R}$ there exists an interior solution $x^*(\alpha) = argmax_x f(x, \alpha)$, then for the optimal value function $v(\alpha) \equiv f(x^*(\alpha), \alpha)$ the following holds true*

$$v'(\alpha) = \left. \frac{\partial f(x, \alpha)}{\partial \alpha} \right|_{x = x^*(\alpha)}$$

Proof. $v'(\alpha) = \frac{\partial f(x^*(\alpha), \alpha)}{\partial \alpha} = \left. \frac{\partial f(x^*(\alpha), \alpha)}{\partial x^*(\alpha)} \frac{\partial x^*(\alpha)}{\partial \alpha} + \frac{\partial f(x, \alpha)}{\partial \alpha} \right|_{x = x^*(\alpha)}$. Since $x^*(\alpha)$ is an interior solution, then, by optimality, $\frac{\partial f(x^*(\alpha), \alpha)}{\partial x^*(\alpha)} = 0$. This cancels out the first term and delivers the desired result. □

Example 1. *Let $f(x, \alpha) \equiv -\alpha x^2 + x$, a strictly concave function. The solution to this program satisfies: $-2\alpha x + 1 = 0 \implies x^*(\alpha) = \frac{1}{2\alpha}$. From this, we can define the value function:*

$$v(\alpha) = -\alpha [x^*(\alpha)]^2 + x^*(\alpha) = -\alpha \left[\frac{1}{2\alpha} \right]^2 + \frac{1}{2\alpha} = \frac{1}{4\alpha}$$

You can immediately see that $v'(\alpha) = -\frac{1}{4\alpha^2} = -x^*(\alpha)^2 = \left. \frac{\partial f(x, \alpha)}{\partial \alpha} \right|_{x = x^*(\alpha)}$.

Our value function, when evaluated at the optimal point $C(K), K'(K)$ is:

$$v(K) = \log[C(K)] + \beta v[K'(K)] = \log[AK^\alpha - K'(K)] + \beta v[K'(K)]$$

By applying the envelope theorem to this particular value function, we can state that the derivative of the value function with respect to K satisfies:

$$v'(K) = \frac{\alpha AK^{\alpha-1}}{AK^\alpha - K'(K)} = \frac{\alpha AK^{\alpha-1}}{C(K)}$$

Therefore, in the first order condition with respect to K'

$$\Lambda = \beta v'(K') = \beta \frac{\alpha A(K')^{\alpha-1}}{C'(K')}$$

By combining the envelope condition and the first-order conditions for C, K' we arrive at our Euler Equation for this recursive problem:

$$\frac{1}{C} = \beta \frac{1}{C'} \alpha A(K')^{\alpha-1} \tag{1}$$

Intuitively, (1) states that, at the optimal, the optimizer will adjust its consumption levels such that the marginal utility of consumption today equals the discounted marginal utility of saving one unit today, producing an extra unit tomorrow, and consuming it tomorrow. The Euler Equation subsumes the dynamics of the model. It shows how consumption tomorrow is related to consumption today.

A solution for the planner's recursive problem is characterized by a value function $v(K)$ and policy functions $C(K), K'(K)$, such that $C(K), K'(K)$ maximize utility and the budget constraint is satisfied $C(K) + K'(K) = AK^\alpha$.

Finally, at the **steady state**, we will look for a solution in which we reach a stationary state: $C'(K) = C$ and $K'(K) = K$. Note that the Euler Equation still holds at the steady state. Therefore, if a steady state exists (as it does in this simple problem), it must be that:

$$\begin{aligned}
\frac{1}{C_{ss}} &= \beta \frac{1}{C_{ss}} \alpha A (K_{ss})^{\alpha-1} \\
(K_{ss})^{\alpha-1} &= \frac{1}{A\beta\alpha} \\
K_{ss} &= (A\beta\alpha)^{\frac{1}{1-\alpha}}
\end{aligned}$$

Using the budget constraint, we can find for C_{ss}:

$$C_{ss} = (A\beta\alpha)^{\frac{\alpha}{1-\alpha}} - (A\beta\alpha)^{\frac{1}{1-\alpha}}$$

2 Numerical Methods

We need to approximate the value function $v(K)$, which is defined over a continuous interval $(0, \infty)$. However, our computers, as a general rule, cannot handle continuous spaces, so we have to **discretize our continuous space.**

- **Step 1: construct a grid**. We do so by constructing a grid —i.e., a set of discrete values of $K \in [\underline{K}, \bar{K}]$ where $\underline{K} > 0$, $\bar{K} < \infty$:

$$G = \{K_1, K_2, \cdots, K_n\}$$

  with $K_1 = \underline{K}$ and $K_n = \bar{K}$; and $K_i - K_{i-1} = c$. Therefore, the grid is a equidistant set of points over the real-line ranging from $\underline{K}$ to $\bar{K}$ for all $i < n$.

- **Step 2: Construct Matrix of Utilities**. Given the grid, we can calculate consumption values $C(K_i, K'_j) = K_i^\alpha - K'_j$ for some values of present K_i and future K'_j capital. We use our grid G in two dimensions, and construct a utility matrix:

$$U = \begin{bmatrix} u(C(K_1, K_1')) & u(C(K_1, K_2')) & \cdots & u(C(K_1, K_n')) \\ \vdots & \ddots & \ddots & \vdots \\ u(C(K_n, K_1')) & u(C(K_n, K_2')) & \cdots & u(C(K_n, K_n')) \end{bmatrix}$$

with a non-restriction $u(\cdot) = -M$ if $C(\cdot, \cdot) < 0$, where M is a large number.

- **Step 3: Have a candidate value function**. We then need a starting guess $v_0 = (v_0(K_1), v_0(K_2), \cdots, v_0(K_n))$ —this can be any guess, including a vector of zeros $v_0 = (0, 0, \cdots, 0)$.

  For a given vector v_m, we can calculate a matrix $\tilde{V}^m$:

$$\tilde{V}^m = \begin{bmatrix} U_{1,1} + \beta v_m(K_1') & U_{1,2} + \beta v_m(K_2') & \cdots & U_{1,n} + \beta v_m(K_n') \\ \vdots & \ddots & \ddots & \vdots \\ U_{n,1} + \beta v_m(K_1') & U_{n,2} + \beta v_m(K_2') & \cdots & U_{n,n} + \beta v_m(K_n') \end{bmatrix}$$

- **Step 4: Update the value function**. Given the results above, update our value function as:

$$v_{m+1}(K_i) = \max_p \tilde{V}^m_{i,p}$$

  resulting in $v_{m+1} = (v_{m+1}(K_1), v_{m+1}(K_2), \cdots, v_{m+1}(K_n))$.

- **Step 5: Calculate update gains**: If $||v_{m+1} - v_m|| = \sup_{K_i} |v_{m+1}(K_i) - v_m(K_i)| < \varepsilon$, where ε is a small error tolerance, we stop the algorithm.

  Otherwise, we go back to **Step 3**, using using v_{m+1} on the right-hand-side of matrix M.

The result of the numerical solution are:

- A value function $v(K)$, which reflects the value of maximized lifetime utility for any starting capital stock K;
- A savings policy function $K'(K)$, which states the optimal choice of future capital for each level of current capital;
- A consumption policy function, derived as $C(K) = K^\alpha - K'(K)$, which states the optimal choice of consumption for each level of current capital.

Given those policy functions and a starting capital level K_0, we can iteratively calculate convergence to a steady-state.