Dynamic Programming: Value Function Iteration

Based on: The ABCs of RBCs, George McCandless (2008)

Carlos Rondón-Moreno

April 8, 2025

Value Function Iteration

- Infinite horizon optimization problems.
- Recursive techniques.
- State variables: determined variables in period t.
- Control variables: chosen variables to maximize an objective.

Robinson Crusoe Model

Sequential problem formulation:

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

subject to

$$K_{t+1} = f(K_t) + (1-\delta)K_t - c_t$$

Value Function (Recursive form):

$$V(K_t) = \max_{K_{t+1}} \left[u(f(K_t) + (1-\delta)K_t - K_{t+1}) + \beta V(K_{t+1})\right]$$

Derivation of the Recursive Form

Starting from the sequential problem:

$$V(K_t) = \max_{\{K_s\}_{s=t+1}^{\infty}} \sum_{i=0}^{\infty} \beta^i u(f(K_{t+i}) - K_{t+1+i} + (1-\delta)K_{t+i})$$

$$\begin{aligned} V(K_t) &= \max_{\{K_s\}_{s=t+1}^{\infty}} [u(f(K_t) + (1-\delta)K_t - K_{t+1}) \\ &\quad + \beta u(f(K_{t+1}) + (1-\delta)K_{t+1} - K_{t+2}) + \dots] \\ &= \max_{K_{t+1}} [u(f(K_t) + (1-\delta)K_t - K_{t+1}) + \beta V(K_{t+1})] \end{aligned}$$

Derivation of the Recursive Form

Starting from the sequential problem:

$$V(K_t) = \max_{K_{t+1}} [u(f(K_t) + (1-\delta)K_t - K_{t+1}) + \beta V(K_{t+1})]$$

First-order necessary condition (FONC):

$$-u'(c_t) + \beta V'(K_{t+1}) = 0$$

But, we have a problem...we don't know $V'(K_{t+1})$

Derivation of the Recursive Form

Idea: Apply the Envelope theorem:

Derivation of the Recursive Form

Idea: Apply the Envelope theorem:

$$V'(K_t) = u_{K_t}(f(K_t) + (1 - \delta)K_t - K_{t+1})$$

$$V'(K_t) = u'(c_t)[f'(K_t) + (1 - \delta)]$$

Euler Equation and Steady State

Using envelope result into FONC:

$$\frac{u'(c_t)}{u'(c_{t+1})} = \beta[f'(K_{t+1}) + (1-\delta)]$$

Steady State: $c_t = c_{t+1}$

$$\frac{1}{\beta} - (1-\delta) = f'(K^*)$$

General Case

Optimization Problem:

$$\max_{\{Y_s\}_{s=t+1}^{\infty}} \sum_{s=t}^{\infty} F(X_s, Y_s) \quad \text{s.t.} \quad X_{s+1} = G(X_s, Y_s)$$

General Case

Optimization Problem:

$$\max_{\{Y_s\}_{s=t+1}^{\infty}} \sum_{s=t}^{\infty} F(X_s, Y_s) \quad \text{s.t.} \quad X_{s+1} = G(X_s, Y_s)$$

Bellman Equation:

$$V(X_t) = \max_{Y_t} [F(X_t, Y_t) + \beta V(G(X_t, Y_t))]$$

General Case

Optimization Problem:

$$\max_{\{Y_s\}_{s=t+1}^{\infty}} \sum_{s=t}^{\infty} F(X_s, Y_s) \quad \text{s.t.} \quad X_{s+1} = G(X_s, Y_s)$$

Bellman Equation:

$$V(X_t) = \max_{Y_t}[F(X_t, Y_t) + \beta V(G(X_t, Y_t))]$$

First-order necessary condition (FONC):

$$F_Y(X_t, Y_t) + \beta V'(G(X_t, Y_t))G_Y(X_t, Y_t) = 0$$

General Case

Optimization Problem:

$$\max_{\{Y_s\}_{s=t+1}^{\infty}} \sum_{s=t}^{\infty} F(X_s, Y_s) \quad \text{s.t.} \quad X_{s+1} = G(X_s, Y_s)$$

Bellman Equation:

$$V(X_t) = \max_{Y_t} [F(X_t, Y_t) + \beta V(G(X_t, Y_t))]$$

First-order necessary condition (FONC):

$$F_Y(X_t, Y_t) + \beta V'(G(X_t, Y_t)) G_Y(X_t, Y_t) = 0$$

Policy function:

$$Y_t = H(X_t)$$

Benveniste-Sheinkman Conditions

The value function derivative is given by:

$$V'(X_t) = F_X(X_t, Y_t) + \beta V'(G(X_t, Y_t))G_X(X_t, Y_t)$$

Under certain conditions, $G_X(X_t, Y_t) = 0$, then:

$$V'(X_t) = F_X(X_t, Y_t)$$

Iterating forward and combining with FONC:

$$F_Y(X_t, Y_t) + \beta F_X(G(X_t, Y_t), Y_{t+1})G_Y(X_t, Y_t) = 0$$

General Version Under Uncertainty

Bellman Equation:

$$V(X_t, Z_t) = \max_{Y_t} [F(X_t, Y_t, Z_t) + \beta E_t V(X_{t+1}, Z_{t+1})]$$
$$\text{s.t.} \quad X_{t+1} = G(X_t, Y_t, Z_t)$$

Stochastic Euler Equation:

$$F_Y(X_t, Y_t, Z_t) + \beta E_t[F_X(X_{t+1}, Y_{t+1}, Z_{t+1}) G_Y(X_t, Y_t, Z_t)] = 0$$

Steps to approximate the value function numerically:

1. Guess an initial value $V^0(X_t)$.
2. Iterate using:

$$V^{i+1}(X_t) = \max_{Y_t}\left[F(X_t, Y_t) + \beta V^i(G(X_t, Y_t))\right]$$

3. Repeat until convergence.

Approximates the policy function $Y_t = H(X_t)$.

Envelope Theorem - General Definition

General definition:

Consider the optimization problem:

$$V(x) = \max_{y \in Y(x)} f(x, y)$$

If $y^*(x)$ is the solution to this problem, the Envelope Theorem states that:

$$\frac{dV(x)}{dx} = \frac{\partial f(x, y^*(x))}{\partial x}$$

In other words, when differentiating the optimized value function, the indirect effects through changes in the optimal choice $y^*(x)$ vanish.