Week 9 — Dynamic Optimisation II

Dynamic Programming & the Bellman Equation

Dr Juan Zurita

Optimisation & Mathematical Methods for Economics
The University of Edinburgh · School of Economics

The Bellman Equation

$$V(k) = \max_{c} \{u(c) + \beta V(k')\}$$

subject to $k' = f(k) - c$.

Principle of Optimality (Bellman, 1957)

An optimal policy has the property that, regardless of the initial state, the remaining decisions must constitute an optimal policy.

Contraction Mapping

Blackwell's Sufficient Conditions

The Bellman operator T is a contraction if it satisfies **monotonicity** and **discounting**.

Consequence: iterating $V_{n+1} = TV_n$ converges to the unique fixed point V^* from any initial guess.

This is the theoretical foundation of **value function iteration**.

Value Function Iteration

Algorithm:

1. Discretise the state space
2. Guess $V_0(k)$
3. For each k: $V_{n+1}(k) = \max_c\{u(c) + \beta V_n(k')\}$
4. Repeat until $\|V_{n+1} - V_n\| < \varepsilon$

This is the algorithm you implement in PNM Week 8.

Analytical Benchmark

With $u(c) = \ln c$ and $f(k) = k^\alpha$:

Optimal savings rate: $s = \alpha\beta$

Policy function: $c^*(k) = (1 - \alpha\beta)k^\alpha$

Use this to verify your numerical code!

Summary

- Bellman equation: recursive formulation of dynamic problems
- Contraction mapping $\Rightarrow$ VFI converges
- Policy function: the optimal decision rule
- Connects directly to PNM Weeks 8–10
- **Next week:** frontiers — stochastic DP, SGD, and beyond