3. Dynamic Programming

Yvan Becard
PUC-Rio

Macroeconomics I, 2023

Solving Models

- To understand large-scale economic phenomena, macroeconomists deal with quantitative data, mostly in the form of time series and panel data
- They then build theoretical and empirical models that aim to fit the data
- Today, most macroeconomic models consist of systems of dynamic equations
- To solve these dynamical systems, macroeconomists use mathematical tools

1. Dynamic programming when time is discrete
2. Optimal control when time is continuous

Dynamic Programming

- Dynamic programming is an optimization method to simplify a complex problem by breaking it down into simpler sub-problems in a recursive way
- Each sub-problem is solved once, and the solution is stored
- Dynamic programming was developed by American mathematician Richard Bellman in the 1950s

The Principle

"Life can only be understood going backwards,
but it must be lived going forwards."

Danish philosopher Søren Kierkegaard, 1843

Lecture Outline

1. A Sequential Problem
2. The Bellman Equation
3. Solution
4. Computational Methods
5. The Euler Equation
6. Stochastic Dynamic Programming
7. The Alternative Method
8. Exercises

Main Reference: Ljungqvist and Sargent, 2018, *Recursive Macroeconomic Theory*, Fourth Edition, Chapter 3

Practical Lecture

"This chapter aims to the reader to start using the methods quickly. We hope to promote demand for further and more rigorous study of the subject."

Ljunqvist and Sargent, 2018, *Recursive Macroeconomic Theory*, Chapter 3

1. A Sequential Problem

Sequential Problem

- The problem is to choose an infinite sequence of controls $\{u_t\}_{t=0}^{\infty}$ to maximize

$$\sum_{t=0}^{\infty} \beta^t r(x_t, u_t) = r(x_0, u_0) + \beta r(x_1, u_1) + \beta^2 r(x_2, u_2) + \beta^3 r(x_3, u_3) + \dots$$

$$\text{subject to} \quad x_{t+1} = g(x_t, u_t), \quad \text{with } x_0 \in \mathbb{R}^n \text{ given}$$

- $\beta \in (0, 1)$ is a constant parameter representing a subjective discount factor
- x_t is a state variable representing the state of the economy today at time t
- u_t is a control variable, also called decision, choice, or policy variable
- $r(x_t, u_t)$ is the objective function we want to maximize
- $x_{t+1} = g(x_t, u_t)$ is a constraint in the optimizing problem, where $g(x_t, u_t)$ is a transition function linking today's state and control to tomorrow's state

Example – A Consumer's Consumption-Savings Problem

- A consumer chooses an infinite sequence of consumption $\{c_t\}_{t=0}^{\infty}$ to maximize utility

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

$$\text{subject to} \quad a_{t+1} = (1+r)a_t - c_t, \quad \text{with } a_0 \text{ given}$$

- Consumption c_t is the control variable
- Asset a_t is the state variable
- $u_t(c_t)$ is a utility function the consumer wants to maximize
- $a_{t+1} = (1+r)a_t - c_t$ is the constraint, an asset accumulation equation
- r is a parameter representing the net rate of return on the asset

Example – A Firm's Investment Problem

- A firm chooses an infinite sequence of investment $\{i_t\}_{t=0}^{\infty}$ to maximize profit

$$\sum_{t=0}^{\infty} \beta^t \{f(k_t) - i_t - \phi(i_t)\}$$

$$\text{subject to} \quad k_{t+1} = (1-\delta)k_t + i_t, \quad \text{with } k_0 \text{ given}$$

- Investment i_t is the control variable
- Capital k_t is the state variable
- $\pi_t \equiv f(k_t) - i_t - \phi(i_t)$ is a profit function the firm wants to maximize
- $f(k_t)$ is a production function, $\phi(i_t)$ is an adjustment cost function
- $\delta \in (0,1)$ is a parameter representing the depreciation rate of capital

State vs Control

- The state variable x_t is a stock variable: it reflects the state of the world at a particular point in time and typically changes gradually over time
- Examples of state variables include wealth, debt, capital stock, money supply, unemployment, a person's level of education or level of skills
- The control variable u_t is a flow variable: it is chosen at every period and need not evolve gradually, ie it can jump, it can be zero, or even negative
- Examples include savings, investment, profit, hours worked, training effort

Back to Our Problem

- We want to solve

$$\max_{\{u_t\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t r(x_t, u_t)$$

$$\text{subject to} \quad x_{t+1} = g(x_t, u_t), \quad \text{with } x_0 \in \mathbb{R}^n \text{ given}$$

- This is an infinite-horizon sequential problem, starting today in period $t = 0$ and going to the infinite future $t = \infty$
- Starting time in period $t = 0$ is a convenient convention
- How do we solve this problem?

Policy Function

- Dynamic programming seeks a time-invariant policy function h mapping the state x_t into the control u_t
- The policy function h, also called decision rule, is such that, starting from an initial condition x_0, the sequence $\{u_t\}_{t=0}^{\infty}$ generated by iterating

$$\begin{aligned} u_t &= h(x_t) && \longleftarrow \text{policy function} \\ x_{t+1} &= g(x_t, u_t) && \longleftarrow \text{original constraint} \end{aligned}$$

  solves the problem
- A solution in the form of these two equations is said to be recursive

2. The Bellman Equation

Value Function

- How do we find the policy function h?
- We take an intermediary step: we define a value function V

$$V(x_0) = \max_{\{u_s\}_{s=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t r(x_t, u_t)$$

$$\text{subject to} \quad x_{t+1} = g(x_t, u_t), \quad \text{with } x_0 \in \mathbb{R}^n \text{ given}$$

- $V(x_0)$ expresses the optimal value of the original problem, starting from an arbitrary initial condition x_0

Bellman's Principle of Optimality

- The principle of optimality formulated by Bellman states

  "An optimal policy has the property that whatever the initial state and initial decision [ie control] are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision."

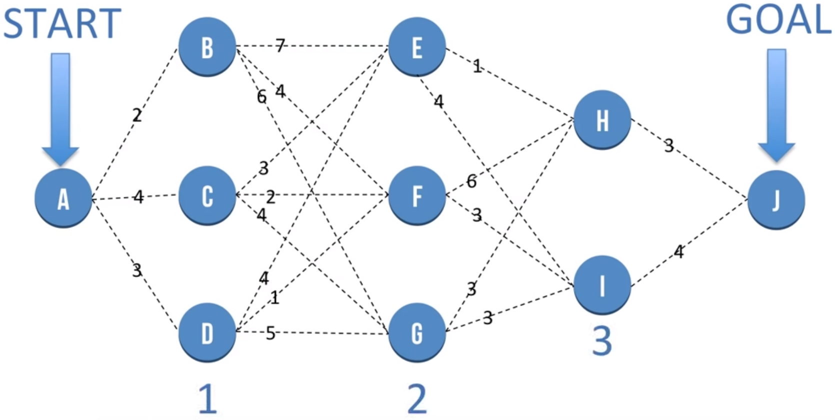
  Richard Bellman, 1957, *Dynamic Programming*

Intuition

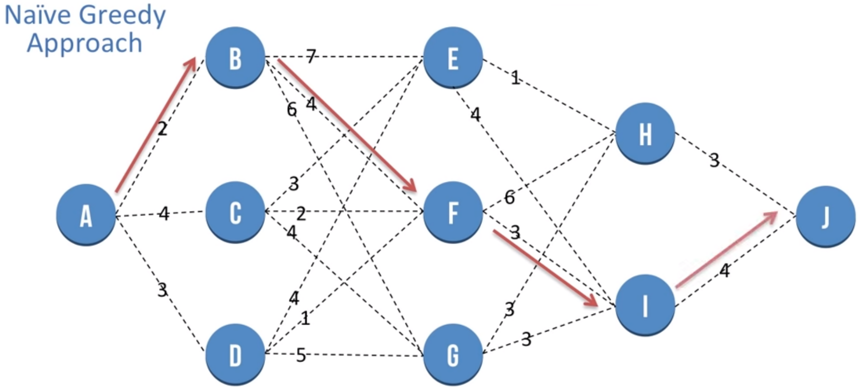
- Intuitively, the principle of optimality says that there is no need to change the last steps of the plan if the target does not change
- For example, suppose the fastest route from Porto Alegre to Rio de Janeiro passes through São Paulo
- The principle of optimality says that the SP to RJ portion of the route is also the fastest route for a trip that starts from São Paulo and ends in Rio
- Let's see a more elaborate example

Example – Shortest Path Problem

- What is the shortest path from node A to node J?



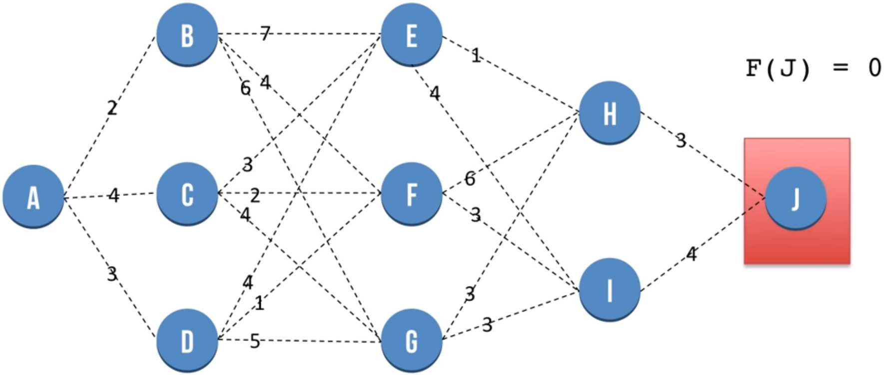
Naive Greedy Approach



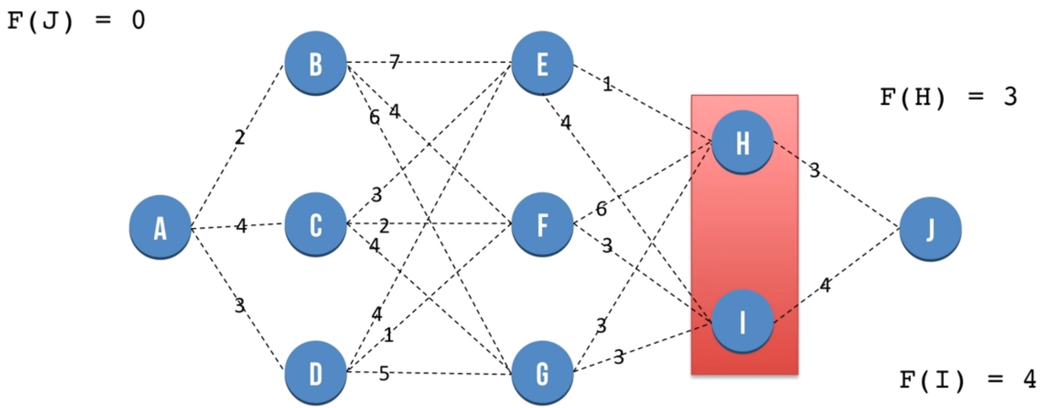
- The naive greedy method builds a solution step by step without backtracking: in each step it picks what is best in the current state
- The naive greedy method yields a total path cost of $2 + 4 + 3 + 4 = 13$

Dynamic Programming Approach

- Let $F(X)$ be the minimum distance required to reach J from a node X



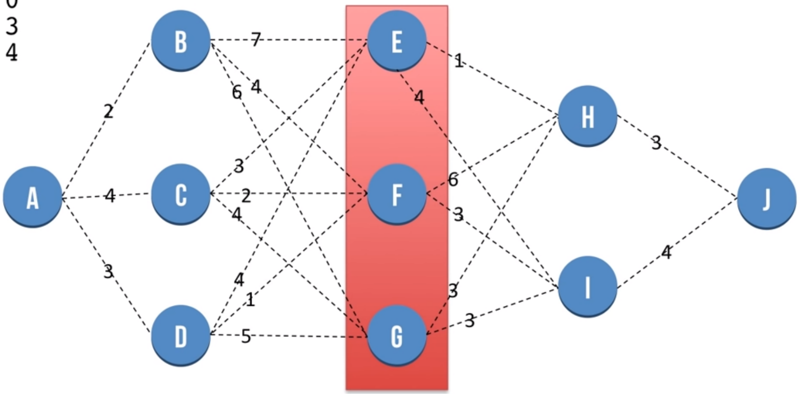
Working Backwards



One Step at a Time

$F(E) = \min\{ 1+F(H), 4+F(I) \}$
$F(E) = 4$

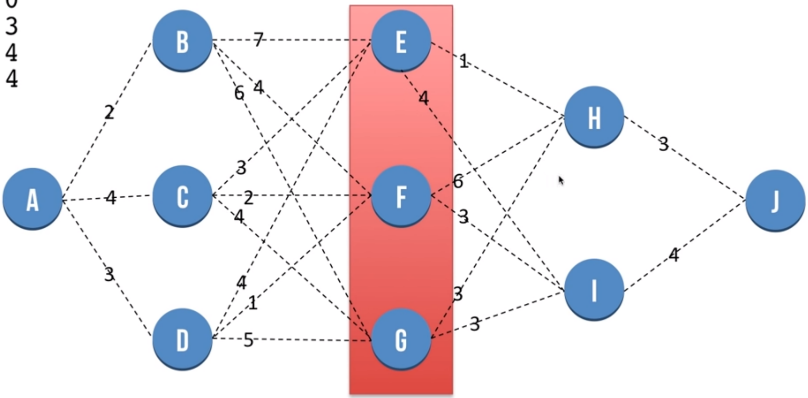
$F(J) = 0$
$F(H) = 3$
$F(I) = 4$



Minimize

F(F) = min{ 6+F(H), 3+F(I)
F(F) = 7

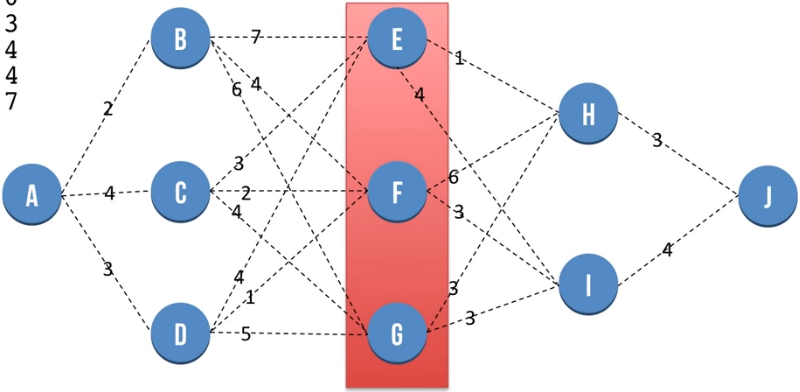
F(J) = 0
F(H) = 3
F(I) = 4
F(E) = 4



Slowly But Surely

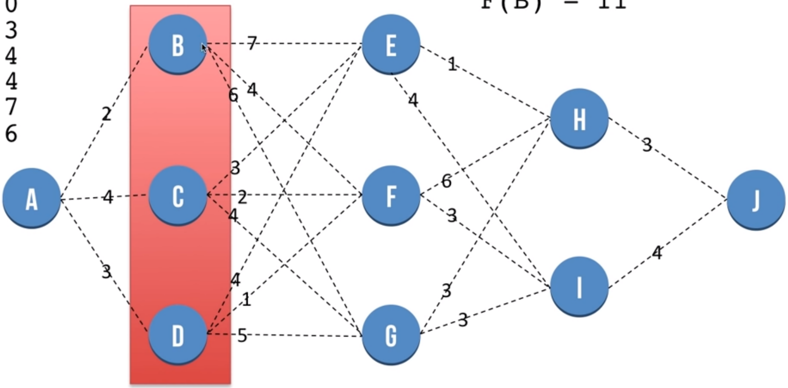
```
F(G) = min{ 3+F(H), 3+F(I)
F(G) = 6
```

```
F(J) = 0
F(H) = 3
F(I) = 4
F(E) = 4
F(F) = 7
```



One More Step

$F(J) = 0$
$F(H) = 3$
$F(I) = 4$
$F(E) = 4$
$F(F) = 7$
$F(G) = 6$



$$F(B) = \min\{\ 7+F(E),\ 4+F(F),\ 6+F(G)\}$$

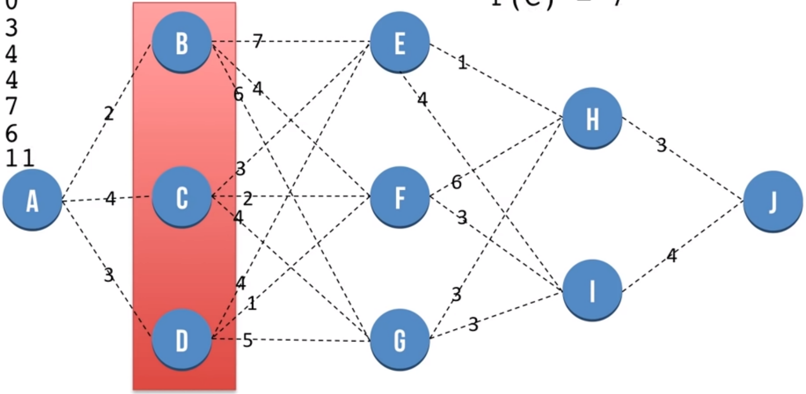
$$F(B) = 11$$

Take it Easy

$$F(C) = \min\{\ 3+F(E),\ 2+F(F),\ 4+F(G)$$

$$F(C) = 7$$

$F(J) = 0$
$F(H) = 3$
$F(I) = 4$
$F(E) = 4$
$F(F) = 7$
$F(G) = 6$
$F(B) = 11$



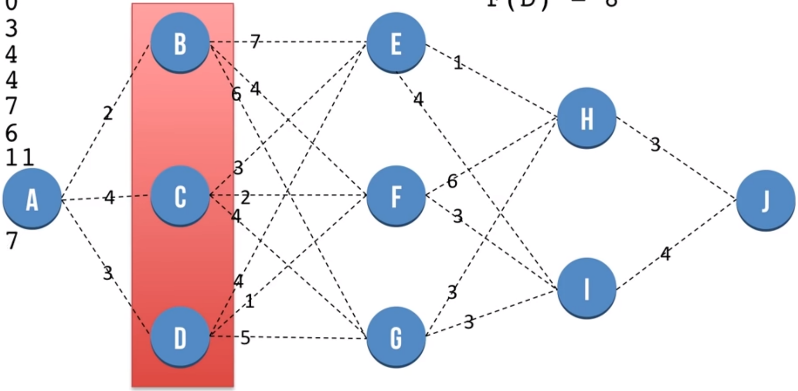
Almost There

$$F(D) = \min\{\ 4+F(E),\ 1+F(F),\ 5+F(G)\}$$

$$F(D) = 8$$

$F(J) = 0$
$F(H) = 3$
$F(I) = 4$
$F(E) = 4$
$F(F) = 7$
$F(G) = 6$
$F(B) = 11$

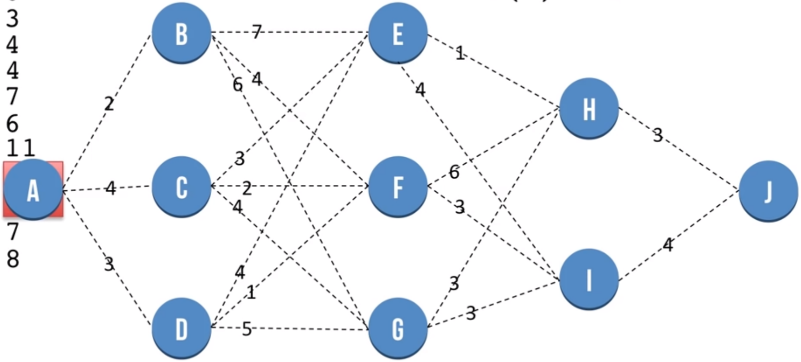
$F(C) = 7$



Last Step

F(J) = 0
F(H) = 3
F(I) = 4
F(E) = 4
F(F) = 7
F(G) = 6
F(B) = 11

F(C) = 7
F(D) = 8



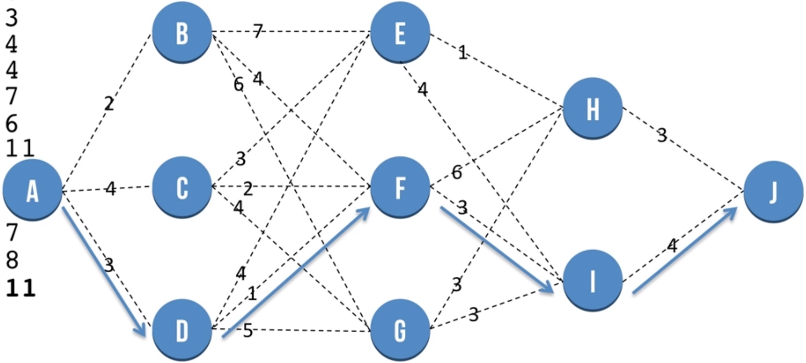
$$F(A) = \min\{\, 2+F(B),\ 4+F(C),\ 3+F(D)\,\}$$

F(A) = 11

And the Winner Is

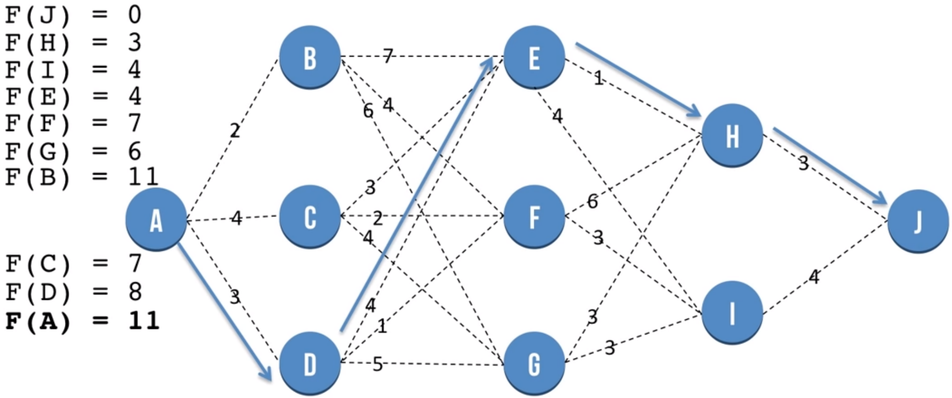
F(J) = 0
F(H) = 3
F(I) = 4
F(E) = 4
F(F) = 7
F(G) = 6
F(B) = 11

F(C) = 7
F(D) = 8
F(A) = 11



- The shortest path costs 11, an improvement over the naive algorithm (13)

Another Solution



- There are actually two solutions to this problem

Example – Recap

- We have built an optimal solution to the problem
  - By breaking the problem into small subproblems
  - And storing and using the optimal solutions to the subproblems
- This example illustrates the principle of optimality
- To see the full video click here (credits to CSBreakdown)

Rewriting the Value Function

- We can express the value function as follows

$$V(x_0) = \max_{\{u_s\}_{s=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t r(x_t, u_t) = \max_{\{u_s\}_{s=0}^{\infty}} \left\{ r(x_0, u_0) + \sum_{t=1}^{\infty} \beta^t r(x_t, u_t) \right\}$$
$$= \max_{\{u_s\}_{s=0}^{\infty}} \left\{ r(x_0, u_0) + \beta \sum_{t=1}^{\infty} \beta^{t-1} r(x_t, u_t) \right\} = \max_{\{u_s\}_{s=0}^{\infty}} \left\{ r(x_0, u_0) + \beta \sum_{t=0}^{\infty} \beta^t r(x_{t+1}, u_{t+1}) \right\}$$

- Using the Bellman principle, we write

$$V(x_0) = \max_{u_0} \{ r(x_0, u_0) + \beta V(x_1) \}$$

Bellman Equation

- Let $\tilde{x}$ denote the state next period
- We can rewrite the problem as

$$V(x) = \max_{u} \{r(x, u) + \beta V(\tilde{x})\}$$
$$\text{subject to} \quad \tilde{x} = g(x, u), \quad \text{with } x \text{ given}$$

- This is the Bellman equation

Bellman Equation

- Remember, the policy function takes the form $h(x) = u$
- If we knew the value function V, we could simply compute h by solving

$$\max_u \{r(x,u) + \beta V(\tilde{x})\}$$
$$\text{subject to} \quad \tilde{x} = g(x,u), \quad \text{with } x \text{ given}$$

- Put differently
$$h(x) = \arg\max_u \{r(x,u) + \beta V[g(x,u)]\}$$
- The policy function $h(x)$ is the maximizer, ie the value of the control u that solves the Bellman equation, the point at which the objective is maximized

Unknown Value Function

- But the problem is, we don't know the value function $V(x)$
- Therefore our task is to jointly solve for $V(x)$ and $h(x)$

Functional Equation

- $V(x)$ and $h(x)$ are linked by the Bellman equation

$$V(x) = \max_u \{r(x,u) + \beta V[g(x,u)]\}$$

- The maximizer of this equation is the policy function $h(x)$ that satisfies

$$V(x) = r[x, h(x)] + \beta V\{g[x, h(x)]\}$$

- This a functional equation to be solved for the pair of unknown functions $V(x), h(x)$

Recap

- So far we have exchanged the problem of finding an infinite sequence of controls

$$\max_{\{u_s\}_{s=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t r(x_t, u_t)$$

$$\text{subject to} \quad x_{t+1} = g(x_t, u_t), \quad \text{with } x_0 \text{ given}$$

- For the problem of finding functions $V(x)$ and $h(x)$ that solve a continuum of maximization problems, one for each x

$$V(x) = \max_u \{r(x, u) + \beta V(\tilde{x})\}$$

$$\text{subject to} \quad \tilde{x} = g(x, u), \quad \text{with } x \text{ given}$$

A Simplified Problem

- Why do we exchange the problem in such way?
- Sequential problems are hard to solve, both analytically and numerically
- So by using the Bellman equation we simplify the problem

$$V(x) = \max_u \{r(x,u) + \beta V[g(x,u)]\}$$

- It becomes easier to prove that a solution exists and is unique

3. Solution

Four Results

- Repeat the Bellman equation in functional form

$$V(x) = \max_u \{r(x,u) + \beta V[g(x,u)]\} \tag{1}$$

- Under particular assumptions about r and g, we can establish four results that allow us solve this recursive problem

Unique V, Unique h, Convergence, Differentiable

1. There is a unique, continuous, bounded, strictly concave value function V that satisfies the functional equation (1)
2. There is a unique and time-invariant optimal policy function of the form $u_t = h(x_t)$ where h maximizes the right side of equation (1)
3. Starting from any bounded and continuous initial V_0, the unique value function V is approached in the limit as $j \to \infty$ by iterating on

$$V_{j+1}(x) = \max_u \{r(x,u) + \beta V_j(\tilde{x})\}$$

subject to $\tilde{x} = g(x,u)$, where x is given and j denotes an iteration, not time

4. The function V is strictly increasing and differentiable off corners

What Are These Assumptions?

- Two assumptions guarantee that a solution exists, though it may not be unique: non-emptiness and compactness
- A third assumption guarantees that V is strictly concave and thus that the solution is unique: convexity
- A fourth and fifth additional assumptions guarantee that V is increasing and differentiable: monotonicity and differentiability

V Exists, Is Unique and Concave, h Is Unique

1. Non-emptiness. $g(x, u)$ exists for all allowed states x; $\lim_{n\to\infty} \sum_{t=0}^{n} \beta^t r(x_t, u_t)$ exists and is finite
   - Prevents currently good-looking path from running into dead ends
   - We don't want unbounded or infinite utility
2. Compactness. The set of all states x is compact, ie closed and bounded; $g(x, u)$ is continuous; and $r(x, u)$ is continuous and bounded
   - Compactness avoids existence issues: without it, there could always be a slightly better x (noncompact set problems are not well behaved)
3. Convexity. $r(x, u)$ is strictly concave and the set of all states x is convex
   - This puts us in a convex optimization problem, ie the constraint is convex and the objective to maximize is concave
   - In such a problem the optimal solution is the global optimum

V Is Strictly Increasing and Differentiable

4. Monotonicity. $r(x, u)$ is strictly increasing in x and $g(x, u)$ is monotone
   - Intuitively, more consumption or wealth is better than less
   - Monotone r and g is needed for monotonicity of the policy function
5. Differentiability. $r(x, u)$ is continously differentiable on the interior of its domain
   - This allows us to derive and work with the first-order conditions

Formal Treatment

- One can prove the preceding conditions
- To see proofs and formal treatment, read Daron Acemoglu, 2009, *Introduction to Modern Economic Growth*, Chapter 6
- Another good reference is Stockey, Lucas, with Prescott, 1989, *Recursive Methods in Economic Dynamics*

4. Computational Methods

Three Computational Methods

- There are three main ways to solve dynamic programs

1. Guess and verify
2. Value function iteration
3. Policy function iteration

Method 1 – Guess and Verify

- Also known as the method of undetermined coefficients
- The guess and verify method comprises three steps

1. Guess a value function $\tilde{V}$
2. Compute h

$$h(x) = \arg\max_u \left\{ r(x,u) + \beta \tilde{V}[g(x,u)] \right\}$$

3. Compute V

$$V(x) = r[x, h(x)] + \beta \tilde{V}\{g[x, h(x)])\}$$

If $V(x) = \tilde{V}(x)$ for all x, we have found the solution

Example – The Neoclassical Growth Model

- A central planner chooses a consumption stream $\{c_t\}_{t=0}^{\infty}$ to maximize

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

$$\text{subject to} \quad k_{t+1} = (1-\delta)k_t + f(k_t) - c_t, \quad \text{with } k_0 \text{ given}$$

- Consumption c_t is the control, capital k_t is the state, $u(c_t)$ is a utility function, $f(k_t)$ is a production function, δ is capital depreciation
- We make the following assumptions
  - Logarithmic preferences: $u(c_t) = \ln c_t$
  - Cobb-Douglas production function: $f(k_t) = Ak_t^{\alpha}$, $A > 0$ and $\alpha \in (0, 1)$
  - Full capital depreciation after use: $\delta = 1$

First Step

0. Preliminary step. Write the problem in recursive form

$$V(k) = \max_{c}\{\ln c + \beta V(\tilde{k})\}$$

$$\text{subject to} \quad \tilde{k} = Ak^{\alpha} - c$$

1. Make the guess

$$\tilde{V}(k) = E + F \ln k$$

where E and F are undetermined constants

Second Step

2. Compute h

$$h(k) = \arg\max_c \{\ln c + \beta[E + F \ln(Ak^\alpha - c)]\}$$

The first-order condition is the solution to $\partial h(k)/\partial c = 0$

$$\frac{1}{h(k)} - \beta F \frac{1}{Ak^\alpha - h(k)} = 0 \quad \Longrightarrow \quad h(k) = \frac{Ak^\alpha}{1 + \beta F}$$

Third Step

3. **Verify the guess, ie verify that** $V(k) = \beta V[Ak^{\alpha} - h(k)] + \ln h(k)$

$$
\begin{aligned}
E + F\ln k &= \ln\left(\frac{Ak^{\alpha}}{1+\beta F}\right) + \beta\left[E + F\ln\left(Ak^{\alpha} - \frac{Ak^{\alpha}}{1+\beta F}\right)\right] \\
&= \underbrace{\ln\left(\frac{A}{1+\beta F}\right) + \beta E + \beta F\ln\left(\frac{A\beta F}{1+\beta F}\right)}_{E} + \underbrace{\alpha(1+\beta F)}_{F}\ln k
\end{aligned}
$$

Third Step, Continued

- Use the method of undetermined coefficients

$$F = \alpha(1 + \beta F) \quad \Longrightarrow \quad F = \frac{\alpha}{1 - \beta\alpha}$$

$$\begin{aligned} E &= \ln\left(\frac{A}{1 + \beta F}\right) + \beta E + \beta F \ln\left(\frac{A\beta F}{1 + \beta F}\right) \\ &= \frac{1}{1 - \beta}\left[\ln\left[(1 - \beta\alpha)A\right] + \beta\frac{\alpha}{1 - \beta\alpha}\ln(\alpha\beta A)\right] \end{aligned}$$

- Our guess was correct

Solution

- With F, we can compute h

$$h(k) = \frac{Ak^\alpha}{1+\beta F} = (1-\beta\alpha)Ak^\alpha$$

- The optimal paths for consumption and capital are therefore

$$\begin{aligned} c_t &= (1-\beta\alpha)Ak_t^\alpha && \longleftarrow \text{policy function} \\ k_{t+1} &= \beta\alpha Ak_t^\alpha && \longleftarrow \text{law of motion of } k \end{aligned}$$

- $Ak_t^\alpha = f(k_t)$ is total income, so the optimal behavior is to devote a constant share $1-\beta\alpha$ of income to consumption and the rest, $\beta\alpha$, to capital (savings)

Convergence

- Remember, α is less than 1 by assumption
- Thus for any positive initial value k_0, k_t converges to a constant k_∞

$$\begin{aligned} k_\infty &= \beta \alpha A k_\infty^\alpha \\ &= (\beta \alpha A)^{\frac{1}{1-\alpha}} \end{aligned}$$

- This point is called the steady state or stationary state

Guess and Verify – Summary

- To sum up, in the optimal neoclassical growth model with particular assumptions: log utility, Cobb-Douglas production, full depreciation
- We are able to derive analytically the optimal paths of consumption (the control) and capital (the state)
- The main drawback of this method is that it depends on a correct guess and thus is not generally available

Method 2 – Value Function Iteration

- Also called iterating on the Bellman equation
- The idea is to construct a sequence of value functions

1. Start from $V_0(x) = 0$ where the subscript denotes an iteration, not time
2. Define
$$V_{j+1}(x) = \max_u \{r(x,u) + \beta V_j(\tilde{x})\}$$
$$\text{subject to } \tilde{x} = g(x,u), \quad x \text{ given}$$
3. Iterate on this equation until V_j converges to the unique V

Example – Neoclassical Growth Model

- Same example as previously

1. Start with $V_0(k) = 0$
2. Define $V_1(k) = \max_c\{\ln c + \beta V_0(\tilde{k})\}$ and solve $\max_c\{\ln c\}$ s.t. $c + \tilde{k} = Ak^\alpha$
   Solution: $c = Ak^\alpha$, $\tilde{k} = 0$, thus $V_1(k) = \ln A + \alpha \ln k$
3. Define $V_2(k) = \max_c\{\ln c + \beta V_1(\tilde{k})\}$ and repeat the procedure
   Solution: $c = \frac{1}{1+\beta\alpha} Ak^\alpha$, $\tilde{k} = \frac{\beta\alpha}{1+\beta\alpha} Ak^\alpha$, thus
   $V_2(k) = \ln \frac{A}{1+\alpha\beta} + \beta \ln A + \alpha\beta \ln \frac{\alpha\beta A}{1+\alpha\beta} + \alpha(1+\alpha\beta) \ln k$
3. Continue, use algebra of geometric series, and find limiting policy functions:
   $c = (1-\beta\alpha)Ak^\alpha$, $\tilde{k} = \beta\alpha Ak^\alpha$,
   $V(k) = \frac{1}{1-\beta}\left(\ln[A(1-\beta\alpha)] + \frac{\beta\alpha}{1-\beta\alpha}\ln(A\beta\alpha)\right) + \frac{\alpha}{1-\beta\alpha}\ln k$

Value Function Iteration

- The main drawback of the value function iteration method is that it is analytically unfeasible in the majority of cases
- In these cases we must turn to numerical methods
- The algorithm in the next slide shows how to approximate the value function numerically

Numerical Approximation

1. Discretize the state variable $x \in [x_1 < x_2 < \cdots < x_n]$; the finer the grid the better the approximation
2. Choose an initial value function V_j for $j = 0$; eg $V_0(x) = 0$ for all x
3. Compute for all $x \in [x_1 < x_2 < \cdots < x_n]$

$$V_{j+1}(x) = \max_u \{r(x,u) + \beta V_j[g(x,u)]\}$$

4. Check convergence: for a given $\varepsilon > 0$
   - If $\max_x |V_{j+1}(x) - V_j(x)| < \varepsilon$, end
   - If $\max_x |V_{j+1}(x) - V_j(x)| > \varepsilon$, iterate and return to step 3

Method 3 – Policy Function Iteration

- Also known as Howard's improvement algorithm
- The policy function iteration manipulates the policy directly, rather than finding it indirectly via the optimal value function
- This method is usually faster than the value function iteration
- The algorithm consists of three steps

Policy Function Iteration

1. Pick an initial feasible policy $u = h_0(x)$ and compute the value associated with operating forever with that policy

$$V_{h_j}(x) = \sum_{t=0}^{\infty} \beta^t r[x_t, h_j(x_t)] \quad \text{where } x_{t+1} = g[x_t, h_j(x_t)], \text{ with } j = 0$$

2. Generate a new policy $u = h_{j+1}(x)$ that solves the two-period problem

$$\max_u \left\{ r(x,u) + \beta V_{h_j}[g(x,u)] \right\} \quad \text{for each } x$$

3. Iterate over j on steps 1 and 2 until convergence, ie until h_j and h_{j+1} are sufficiently close

In Search of an Analytical Solution

- To summarize, a Bellman equation is

$$V(k) = \max_{u}\{r(x,u) + \beta V(\tilde{x})\} \quad \text{subject to } \tilde{x} = g(x,u)$$

- It is usually difficult to find an exact solution to the value function, ie characterize V analytically, but it is easy to approximate V numerically
- Economists like analytical solutions, so is there a way to solve analytically for the optimal path of $\{x_t, u_t\}_{t=0}^{\infty}$ without knowing the value function?
- Yes, using the Euler equation

5. The Euler Equation

A Slightly Modified Problem

- Let us rewrite the problem in such a way that only the control appears in the transition equation

$$\tilde{x} = g(\hat{u}) \quad \text{instead of } \tilde{x} = g(x, u)$$

- The Bellman equation writes

$$V(x) = \max_{\hat{u}} \{ r(x, \hat{u}) + \beta V[g(\hat{u})] \}$$

- Under previous assumptions, V is differentiable

First-Order Condition and Envelope Theorem

- The way to solve this problem by hand is to derive optimality conditions
- The first-order condition (FOC) is such that $\partial V(x)/\partial \hat{u} = 0$

$$r_u(x, \hat{u}) + \beta V'[g(\hat{u})]g'(\hat{u}) = 0 \tag{2}$$

- The envelope theorem (ET), or envelope condition, is

$$V'(x) = r_x(x, \hat{u}) \tag{3}$$

Envelope Theorem

- Where does the envelope condition (3) come from?
- Once we solve our problem and find the FOC, we have the function $h(x)$ that satisfies

$$V(x) = r[x, h(x)] + \beta V\{g[h(x)]\}$$

- Take the total derivative

$$\begin{aligned} V'(x) &= r_x[x, h(x)] + \underbrace{r_{h(x)}[x, h(x)]h'(x) + \beta V'\{g[h(x)]\}g'[h(x)]h'(x)}_{= 0 \text{ by the FOC (2)}} \\ &= r_x[x, h(x)] \end{aligned}$$

where the second lines makes use of the FOC (2) with $\hat{u} = h(x)$

Intuition for the Envelope Theorem

- We know that agents are optimizing
- Therefore, we always evaluate the value function and its derivative at the particular value of the control $\hat{u}$ that satisfies the first-order condition
- It follows that the derivative of a value function with respect to control variables is always zero at the optimum
- In other words, there exists no other value of $\hat{u}_t$ that could raise the value function since optimizing agents have already solved for $\frac{\partial V(x)}{\partial \hat{u}} = 0$

Necessary Conditions

- The first-order condition and the envelope condition are two necessary conditions to characterize the optimal path
- In other words, an optimal sequence $\{x_t, \hat{u}_t\}_{t=0}^{\infty}$ must satisfy

$$\begin{aligned} \textbf{FOC:} \quad & r_u(x_t, \hat{u}_t) + \beta V'[g(\hat{u}_t)]g'(\hat{u}_t) = 0 \\ \textbf{ET:} \quad & V'(x_t) = r_x(x_t, \hat{u}_t) \quad \Longrightarrow \quad V'(x_{t+1}) = r_x(x_{t+1}, \hat{u}_{t+1}) \end{aligned}$$

Euler Equation

- Given that $x_{t+1} = g(\hat{u}_t)$, we combine the FOC and the ET to substitute out $V'(x_{t+1})$
$$r_u(x_t, \hat{u}_t) + \beta r_x(x_{t+1}, \hat{u}_{t+1}) g'(\hat{u}_t) = 0$$
- We obtain the Euler equation

Solution to the Optimal Path

- We have now solved our problem
- Two equations characterize the optimal dynamics of the system

1. Euler equation (FOC + ET): $r_u(x_t, \hat{u}_t) + \beta r_x(x_{t+1}, \hat{u}_{t+1}) g'(\hat{u}_t) = 0$
2. Law of motion of the state variable (constraint): $x_{t+1} = g(\hat{u}_t)$

- The law of motion of the state variable is always part of the solution: if we had N state variables, we would have N laws of motion in the solution

Necessary But Not Sufficient

- The Euler equation and law of motion of the state variable are necessary conditions for the optimal path $\{x_t, \hat{u}_t\}_{t=0}^{\infty}$
- But they are not sufficient
- We need a transversality condition: more on this in an upcoming lecture

Example – Neoclassical Growth Model

- The problem is to choose an infinite sequence of consumption and capital

$$\max_{\{c_t, k_{t+1}\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t u(c_t)$$

$$\text{subject to} \quad k_{t+1} = (1-\delta)k_t + f(k_t) - c_t, \quad \text{with } k_0 \text{ given}$$

- Plug the constraint into the objective function to get rid of c_t

$$\max_{\{k_{t+1}\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t u\left[(1-\delta)k_t + f(k_t) - k_{t+1}\right], \quad k_0 \text{ given}$$

- The problem is now to choose a sequence of controls k_{t+1}, given states k_t

Bellman Equation

- Write a Bellman equation

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

- Note that $\tilde{k}$ is the control variable and k is the state variable
- Once we have chosen $\tilde{k}$ we know exactly what c is by using the constraint

$$c = (1-\delta)k + f(k) - \tilde{k}$$

Necessary Conditions

- Derive the necessary conditions

$$\textbf{FOC:} \quad \frac{\partial V(k)}{\partial \tilde{k}} = 0 \quad \Longrightarrow \quad -u'[(1-\delta)k + f(k) - \tilde{k}] + \beta V'(\tilde{k}) = 0$$

$$\textbf{ET:} \quad \frac{\partial V(k)}{\partial k} + \frac{\partial V(k)}{\partial \tilde{k}} = V'(k) = u'[(1-\delta)k + f(k) - \tilde{k}][(1-\delta + f'(k)]$$

- Combine the two necessary conditions and use $c = (1-\delta)k + f(k) - \tilde{k}$

$$u'[\underbrace{(1-\delta)k + f(k) - \tilde{k}}_{c}] = \beta u'[\underbrace{(1-\delta)\tilde{k} + f(\tilde{k}) - \hat{k}}_{\tilde{c}}][(1-\delta + f'(\tilde{k})]$$

where $\hat{k}$ is the value of capital two periods ahead

Euler Equation

- We thus arrive at the Euler equation

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

- This equation is a building block of macroeconomic and finance models
- Example with log utility, $u'(c_t) = \log c_t$ and let $R_t \equiv 1 - \delta + f'(k_t)$

$$\frac{c_{t+1}}{c_t} = \beta R_{t+1}$$

- The growth rate of consumption equals the return to capital

Intuition for the Euler Equation

- $u'(c_t)$ is the marginal benefit of increasing consumption today, or equivalently the marginal cost of reducing consumption today
- $\beta u'(c_{t+1})[1 - \delta + f'(k_{t+1})]$ is the present-value marginal gain of investing in k_t today to consume more tomorrow, or the marginal loss of disinvesting
- The Euler equation states that, at the optimum, agents choose a level of c_t such that the marginal utility of consumption today equals that of tomorrow
- Increasing c_t to reduce k_t and hence c_{t+1} does not raise utility; conversely, decreasing c_t to increase k_t and c_{t+1} does not raise utility either

6. Stochastic Dynamic Programming

Stochastic Control Problem

- So far the problem was nonstochastic, ie there was no uncertainty
- We introduce an exogenous stochastic variable, often called a shock
- So long as shocks are either independently and identically distributed (iid) or Markov, then dynamic programming methods are appropriate

Stochastic Control Problem

- The problem consists of maximizing

$$E_0 \sum_{t=0}^{\infty} \beta^t r(x_t, u_t)$$

$$\text{subject to} \quad x_{t+1} = g(x_t, u_t, z_t), \quad \text{with } x_0 \in \mathbb{R}^n \text{ given}$$

- z_t is an exogenous state variable, which is either a sequence of iid random variables or a Markov chain
- The history of z_t up to period t is $z^t = [z_0, z_1, \ldots, z_t]$
- $E_t(y)$ denotes the mathematical expectation of a random variable y given information known at time t

Policy Function

- If z_t is iid or Markov, there exists a time-invariant policy function h mapping endogenous state x_t and exogenous state z_t into control u_t
- h is such that, given initial x_0, the sequence $\{u_t\}_{t=0}^{\infty}$ generated by iterating

$$u_t = h(x_t, z_t)$$
$$x_{t+1} = g(x_t, u_t, z_t)$$

solves the problem

Bellman Equation

- The Bellman equation is

$$V(x,z) = \max_u \{r(x,u) + \beta E[V(\tilde{x},\tilde{z})|z]\}$$
$$\text{subject to} \quad \tilde{x} = g(x,u,z)$$

- Under the same previous five assumptions, modified to include the extra state z, everything that we discussed above applies to the stochastic case
- For proofs, see Acemoglu (2009), Chapter 16

Example – Stochastic Neoclassical Growth Model

- Same neoclassical growth model with log utility and full depreciation
- But now the production function is $f(z, k) = zk^{\alpha}$
- z follows a Markov chain with states $[\bar{z}_1, \bar{z}_2, \ldots, \bar{z}_n]$ and transition matrix $\boldsymbol{P}$

Bellman Equation

- Write a Bellman equation

$$V(k,z) = \max_c \left\{ \ln c + \beta E\left[V(\tilde{k},\tilde{z}) | z \right] \right\}$$

$$\text{subject to} \quad \tilde{k} = zk^{\alpha} - c$$

- For example, if $z = \bar{z}_i$

$$E\left[V(\tilde{k},\tilde{z}) | z = \bar{z}_i \right] = \sum_{j=1}^{n} P_{ij} V(\tilde{k},\bar{z}_j)$$

Solution

- We can use the method of undetermined coefficients to determine the optimal path of consumption and capital (see exercise 2)

$$
\begin{aligned}
c_t &= (1 - \beta\alpha) z_t k_t^{\alpha} \\
k_{t+1} &= \beta\alpha z_t k_t^{\alpha}
\end{aligned}
$$

- It turns out the optimal path does not depend on the transition matrix
- We can also derive the necessary conditions and the stochastic Euler equation by following the same steps as above (see exercise 2)

Deterministic vs Stochastic Euler Equation

- To sum up, in a certain world, we derive the Euler equation

$$r_u(x_t, u_t) + \beta r_x(x_{t+1}, u_{t+1})g'(u_t) = 0$$

- In a stochastic world, we have the stochastic Euler equation

$$r_u(x_t, u_t) + \beta E_t[r_x(x_{t+1}, u_{t+1})]g'(u_t) = 0$$

where the expectation is conditional on information available at t

Taking Stock

- We have seen how dynamic programming works
- Dynamic programming solves an optimization problem by breaking it down into simpler subproblems
- Dynamic programming uses the fact that the optimal solution to the overall problem depends on the optimal solution to its subproblems
- It is a powerful technique, yet not the only one we know

7. The Alternative Method

Method of Lagrange Multipliers

- Recall from your undergrad microeconomic studies the method of Lagrange multipliers to solve optimization problems subject to equality constraints

$$\max_{c_1,\, c_2} u(c_1, c_2) \quad \text{subject to} \quad c_1 + c_2 = y$$

- A consumer purchases two goods c_1 and c_2 to maximize utility $u(c_1, c_2)$ subject to a budget constraint $c_1 + c_2 = y$, where y is income
- The Kuhn-Tucker approach generalizes the method of Lagrange multipliers to problems with inequality constraints

$$\max_{c_1,\, c_2} u(c_1, c_2) \quad \text{subject to} \quad c_1 + c_2 \leq y$$

Dynamic Optimization Problem

- We can apply the Lagrange multiplier (or Kuhn-Tucker) method to solve a dynamic optimization problem

$$\max_{\{u_t\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t r(x_t, u_t)$$

$$\text{subject to} \quad x_{t+1} = g(u_t), \quad \text{with } x_0 \in \mathbb{R}^n \text{ given}$$

Lagrangian and Lagrange Multiplier

- To solve this problem, we introduce a Lagrange multiplier λ_t and we form the Lagrangian function $\mathcal{L}(x_t, u_t, \lambda_t)$

$$\mathcal{L}(x_t, u_t, \lambda_t) = \sum_{t=0}^{\infty} \beta^t \left\{ \underbrace{r(x_t, u_t)}_{\text{objective}} + \underbrace{\lambda_t}_{\text{multiplier}} \cdot \underbrace{[x_{t+1} - g(u_t)]}_{\text{constraint}} \right\}$$

Necessary Conditions

- The necessary conditions are, for each t

$$\begin{aligned}
\mathcal{L}_u(x_t, u_t, \lambda_t) = 0 \quad &\Longrightarrow \quad r_u(x_t, u_t) - \lambda_t g'(u_t) = 0 \\
\mathcal{L}_{x+1}(x_t, u_t, \lambda_t) = 0 \quad &\Longrightarrow \quad \beta r_x(x_{t+1}, u_{t+1}) + \lambda_t = 0 \\
\mathcal{L}_\lambda(x_t, u_t, \lambda_t) = 0 \quad &\Longrightarrow \quad x_{t+1} = g(u_t)
\end{aligned}$$

- Combine the first two necessary conditions to substitute out the Lagrange multiplier and obtain the same Euler equation

$$r_u(x_t, u_t) + \beta r_x(x_{t+1}, u_{t+1}) g'(u_t) = 0$$

- Note the constraint is always a necessary condition: $x_{t+1} = g(u_t)$

Dynamic Programming vs Lagrange Multiplier

- What differentiates the two approaches? Mainly the solution method
- Dynamic programming consists of backward induction
  - Identify the optimal action for all possible values of the state variable in the final period, then reason backwards following the state equation
  - In this way get a solution not just for the current path but for all others
- Lagrange multipliers/Kuhn-Tucker basically works the other way around
  - Formulate the necessary and sufficient conditions
  - Solve the resulting difference equation(s) using the initial conditions as starting points

Which Method Should We Use and When?

- In many economic problems we can use either one of the two methods, and both lead to the same optimality conditions
- Dynamic programming provides us with a host of “other solutions” in case we are not currently on the optimal path
- Some problems, however, are much easier cast in recursive form and hence dynamic programming is the way to go (eg next lecture)

Conclusion

- Thanks to dynamic programming and Lagrange multipliers methods, we are now able to solve many kinds of dynamic optimization problems
- And thus virtually any macroeconomic model
- In upcoming lectures we will use these tools to tackle a range of issues
- These include job search, economic growth, risk sharing, asset pricing, business cycles, fiscal policy, labor market dynamics, unemployment

8. Exercises

Exercise 1 – Howard's Policy Iteration Algorithm

Consider the problem to maximize

$$E_0 \sum_{t=0}^{\infty} \beta^t \ln c_t,$$

subject to $c_t + k_{t+1} \leq Ak_t^{\alpha}\theta_t$, k_0 given, $A > 0$, $0 < \alpha < 1$, where $\{\theta_t\}$ is an iid sequence with $\ln \theta_t$ distributed according to a normal distribution with mean zero and variance σ^2.

Consider the following algorithm. Guess at a policy of the form $k_{t+1} = h_0(Ak_t^{\alpha}\theta_t)$ for any constant $h_0 \in (0,1)$. Then form

$$J_0(k_0, \theta_0) = E_0 \sum_{t=0}^{\infty} \beta^t \ln(Ak_t^{\alpha}\theta_t - h_0 Ak_t^{\alpha}\theta_t).$$

Exercise 1 – Continued

Next choose a new policy h_1 by maximizing

$$\ln(Ak^\alpha\theta - k') + \beta E J_0(k', \theta'),$$

where $k' = h_1 A k^\alpha \theta$. Then form

$$J_1(k_0, \theta_0) = E_0 \sum_{t=0}^{\infty} \beta^t \ln(Ak_t^\alpha \theta_t - h_1 A k_t^\alpha \theta_t).$$

Continue iterating on this scheme until successive h_j have converged.

- Show that, for the present example, this algorithm converges to the optimal policy function in one step

Exercise 2 – Stochastic Growth with Log Utility and $\delta = 1$

Consider the growth model studied in class, with log preferences $u(c) = \ln c$, a stochastic production function $f(z, k) = zk^{\alpha}$, and full capital depreciation $\delta = 1$.

1. Using the method of undetermined coefficients, show that for all t

$$\begin{aligned} c_t &= (1 - \beta\alpha) z_t k_t^{\alpha} \\ k_{t+1} &= \beta\alpha z_t k_t^{\alpha} \end{aligned}$$

2. Use dynamic programming to derive the stochastic Euler equation