Dynamic Programming and Applications

Discrete Time Dynamics and Optimization

Lecture 1

Andreas Schaab

Introduction

- Many (most) economic phenomena are about dynamics
- Dynamics $\sim$ changes in the behavior of economic agents over time
- This is a course about **dynamic optimization**: main approach to study dynamic models in economics
- Goal of this course: hard skills + empower you to do research
  $\implies$ *focus on tools, but with tons of applications*
- The more you can "tool up" during your first two years, the better
- Specifically: **dynamic programming** one of most powerful tools in economic analysis

Dynamic programming will empower you

- Macro: consumption, investment, portfolio allocation, ...
- Labor: search, wage bargaining, ...
- IO: competition and games, pricing, ...
- Finance: asset pricing, dynamic capital structure, portfolio choice, ...
- Growth: technology adoption, poverty traps, firm innovation, ...
- Trade / Urban: migration in spatial models, ...
- Inequality: evolution of income-wealth distribution, ...
- Game theory: dynamic games, ...
- ... much, much more

Acknowledgements

- Course builds on excellent teaching material developed by others
- Huge thanks to David Laibson and Benjamin Moll for making available their incredible teaching material
  `https://projects.iq.harvard.edu/econ2010c/lecturesDLaibson`
  `https://benjaminmoll.com/`
- Grateful to many others whose material I build on: Pablo Kurlat, Gabriel Chodorow-Reich, Gianluca Violante, ...
- Only required textbook: LeVeque on finite difference methods (check syllabus for many other recommendations)

GitHub

- One super useful *hard skill* for research / code development: version control
- You should start using git (via GitHub) in your own work from the start
- Course material will be available via a GitHub organization at:
  `https://github.com/schaab-teaching/DynamicProgramming2024`
- You should create a GitHub account if you don't already have one
- If this is new for you, I also recommend GitKraken
- Get Pro versions of GitHub and GitKraken for free (verify your academic affiliation)
- Great resource: `https://git-scm.com/book/en/v2`
  (Especially chapters 2, 3, 5, and 6)

Admin

- Course co-taught by Kiyea and myself
- Lectures will focus on theory and applications; sections will teach you coding
- Problem sets every week. Two parts in two separate (!) docs: analytical and numerical. They are optional. Final counts for 100% of your grade.
  You are responsible from now on
- **However**: completing the homework earns you extra credit, up to 60%
  To illustrate, suppose you score 75/100 on exam. Without homework, 75%. If you complete half of the problems each week: 30 + 0.7*75 = 82.5%. If you complete all problems: 60 + 0.4*75 = 90%.
- Final exam will be close to the homework: I want to maximally incentivize you to work hard on the homework

Honor Code

Rules of the game:

> *By turning in homework to receive extra credit, I affirm that (i) I did not consult past solution keys and (ii) I intellectually contributed to each problem answer I am submitting.*

Course Overview

Part I: dynamic programming (with examples and applications)

- Lectures 1-2: Dynamic programming in discrete time
- Lectures 3-4: Deterministic dynamic programming in continuous time
- Lectures 5-6: Stochastic dynamic programming in continuous time

Part II: (even more) applications

- Lectures 7-8: Consumption
- Lectures 9: Investment
- Lectures 10-11: Inequality
- Lectures 12-13: Welfare

Outline

Part 1: Neoclassical growth model

1. Environment
2. Planning problem

Part 2: dynamic programming

1. Bellman equation
2. First-order and envelope conditions
3. Solving the Bellman equation with guess-and-verify
4. Bellman operator

Part 3: Competitive equilibrium

Part 1: Neoclassical Growth Model

1. Environment

- Time is discrete with $t = 0, 1, \ldots$ and there is no uncertainty
- Consider a **representative household**
- **Preferences** over consumption c_t given by

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

- $u(\cdot)$ is **flow utility**: units of c_t "dollars" or "apples", units of $u(c_t)$ "utils"

- **Technologies**: final good produced according to

$$y_t = f(k_t)$$

and capital accumulation technology is

$$k_{t+1} = i_t + (1 - \delta)k_t$$

for given $k_0 = \bar{k}$ (**initial condition**)

- **Resource constraint:**

$$y_t = c_t + i_t$$

Definition. Given initial capital stock k_0, a **feasible allocation** consists of sequences $\{y_t, c_t, i_t, k_t\}_{t=0}^{\infty}$ that satisfy the production and capital accumulation technologies, the resource constraint, as well as non-negativity constraints $y_t \geq 0$, $c_t \geq 0$, $i_t \geq 0$ and $k_{t+1} \geq 0$.

2. Planning problem

Definition. The planning problem is to choose a feasible allocation that maximizes the lifetime utility of the representative household.

- Notice: we have not introduced prices, markets, budget constraints, etc.
- Why is planning problem relevant? (i) important in its own right to understand efficiency, and (ii) turns out first welfare theorem applies here, so efficient allocation = competitive equilibrium allocation
- Social welfare = lifetime utility of representative household (why?)
- "choose a feasible allocation" = choose sequences $\{y_t, c_t, i_t, k_t\}_{t=0}^{\infty}$ subject to technologies and resource constraints

- Assume $f(k_t) = k_t^\alpha$ with $\alpha \in (0,1)$ and $\delta = 1$
- Can write the planning problem as

$$V(k_0) = \max_{\{c_t\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t u(c_t)$$

s.t.

$$k_{t+1} = k_t^\alpha - c_t.$$

- This is the **sequence problem** (or problem in sequence form)
- We can substitute in:

$$V(k_0) = \max_{\{k_{t+1}\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t u(k_t^\alpha - k_{t+1})$$

- Given initial condition k_0

- We can tackle the sequence problem directly using tools from constrained dynamic optimization
- Lagrangian after substituting with FOC for k_{t+1}:

$$L(k_0) = \sum_{t=0}^{\infty} \beta^t u(k_t^{\alpha} - k_{t+1})$$

$$0 = -\beta^t u'(c_t) + \beta^{t+1} u'(c_{t+1}) \alpha k_{t+1}^{\alpha-1}$$

- Lagrangian with multiplier before substituting with FOCs for c_t and k_{t+1}:

$$L(k_0) = \sum_{t=0}^{\infty} \beta^t \left[u(c_t) + \lambda_t \left(k_t^{\alpha} - c_t - k_{t+1} \right) \right]$$

$$0 = \beta^t u'(c_t) - \beta^t \lambda_t$$
$$0 = -\beta^t \lambda_t + \alpha \beta^{t+1} \lambda_{t+1} k_{t+1}^{\alpha-1}$$

Proposition. An allocation $\{y_t, c_t, i_t, k_t\}_{t=0}^{\infty}$ is (Pareto) efficient if it is feasible and satisfies

$$u'(c_t) = \beta f'(k_{t+1}) u'(c_{t+1}).$$

In other words, an efficient allocation solves the equations:

$$\begin{aligned} u'(c_t) &= \beta f'(k_{t+1}) u'(c_{t+1}) \\ y_t &= f(k_t) \\ k_{t+1} &= i_t \\ y_t &= c_t + i_t, \end{aligned}$$

given initial condition $k_0 = \bar{k}$.
Always count equations and unknowns!!

Part 2: Dynamic Programming

1. Bellman equation

Definition. The **Bellman equation** characterizes the value function as the sum of the **flow payoff** and the discounted **continuation value**

$$V(k) = \max_{k'} \left\{ u(k^\alpha - k') + \beta V(k') \right\} \quad \text{for all } k$$

- We call $V(k)$ the value function and $u(k^\alpha - k')$ flow payoff or (instantaneous) utility flow
- **Recursive representation** of the planning problem (not sequence form)
- We say that $\mathcal{X} = [0, \bar{k}]$ is the **state space** of the neoclassical growth model. The Bellman equation must hold for all feasible levels of capital $k \in \mathcal{X}$.
- If $V(k)$ solves the above equation, then it is a solution to the Bellman equation. Next: the unique value function that solves sequence problem also solves Bellman equation

- Sequence problem $\longrightarrow$ recursive representation (Bellman equation)

$$
\begin{aligned}
V(k_0) &= \max_{\{k_{t+1}\}_{t=0}^{\infty}} \left\{ \sum_{t=0}^{\infty} \beta^t u(k_t^\alpha - k_{t+1}) \right\} \\
&= \max_{\{k_{t+1}\}_{t=0}^{\infty}} \left\{ u(k_0^\alpha - k_1) + \sum_{t=1}^{\infty} \beta^t u(k_t^\alpha - k_{t+1}) \right\} \\
&= \max_{\{k_{t+1}\}_{t=0}^{\infty}} \left\{ u(k_0^\alpha - k_1) + \beta \sum_{t=1}^{\infty} \beta^{t-1} u(k_t^\alpha - k_{t+1}) \right\} \\
&= \max_{k_1} \left\{ u(k_0^\alpha - k_1) + \beta \max_{\{k_{t+1}\}_{t=1}^{\infty}} \left\{ \sum_{t=0}^{\infty} \beta^t u(k_{t+1}^\alpha - k_{t+2}) \right\} \right\} \\
&= \max_{k_1} \left\{ u(k_0^\alpha - k_1) + \beta V(k_1) \right\}
\end{aligned}
$$

- Recursively, with k capital "today" and k' capital "tomorrow"

$$
V(k) = \max_{k'} \left\{ u(k^\alpha - k') + \beta V(k') \right\}
$$

- A solution to the Bellman equation also solves Sequence Problem

$$
\begin{aligned}
V(k_0) &= \max_{k_1} \left\{ u(k_0^\alpha - k_1) + \beta V(k_1) \right\} \\
&= \max_{k_1} \left\{ u(k_0^\alpha - k_1) + \beta \left(\max_{k_2} \left\{ u(k_1^\alpha - k_2) + \beta V(k_2) \right\} \right) \right\} \\
&= \max_{k_1, k_2} \left\{ u(k_0^\alpha - k_1) + \beta u(k_1^\alpha - k_2) + \beta^2 V(k_2) \right\} \\
&= \max_{k_1, k_2, k_3} \left\{ u(k_0^\alpha - k_1) + \beta u(k_1^\alpha - k_2) + \beta^2 u(k_2^\alpha - k_3) + \beta^3 V(k_3) \right\} \\
&\vdots \\
&= \max_{\{k_{t+1}\}_{t=0}^{\infty}} \left\{ \sum_{t=0}^{\infty} \beta^t u(k_t^\alpha - k_{t+1}) \right\}
\end{aligned}
$$

- Stokey and Lucas Thm 4.3: Sufficient condition is that $\lim_{n \to \infty} \beta^n V(k_n) = 0$ for all feasible sequences of $\{k_t\}$

2. First-Order and Envelope Conditions

- How do we find optimal behavior?
- First-order condition for k' is

$$\frac{\partial u(k^\alpha - k')}{\partial k'} = \beta \frac{\partial V(k')}{\partial k'}$$

- FOC defines **policy function** $k'(k)$, use to plug back into Bellman:

$$V(k) = u(k^\alpha - k'(k)) + \beta V(k'(k))$$

- The implied **consumption policy function** is: $c(k) = k^\alpha - k'(k)$
- Important: we now characterize the solution to the model via **functions** ($V(k)$ and $c(k)$) and no longer via stochastic processes ($\{c_t\}_{t=0}^{\infty}$)

- Envelope theorem (in discrete time):

$$\frac{\partial V(k)}{\partial k} = \frac{\partial u(k^{\alpha} - k')}{\partial k}$$

- Work out at home: (i) prove envelope condition (ii) show resulting Euler equation coincides with solving sequence problem using Lagrangian

3. Solving the Bellman equation with guess-and-verify

- Assume the functional form $u(c_t) = \log(c_t)$
- Guess that the value function takes the form

$$V(k) = A + B \log(k)$$

- Strategy: Plug into Bellman equation, match coefficients
- Oftentimes, value function inherits functional form / shape of utility function

- First-order condition for capital with log utility:

$$\frac{1}{k^\alpha - k'} = \beta B \frac{1}{k'} \implies k' = \frac{\beta B}{1 + \beta B} k^\alpha$$

- Plug guess into Bellman:

$$V(k) = \max_{k'} \left\{ \log(k^\alpha - k') + \beta V(k') \right\}$$

$$A + B \log(k) = \max_{k'} \left\{ \log(k^\alpha - k') + \beta A + \beta B \log(k') \right\}$$

$$A + B \log(k) = \log\left(k^\alpha - \frac{\beta B}{1 + \beta B} k^\alpha \right) + \beta A + \beta B \log\left(\frac{\beta B}{1 + \beta B} k^\alpha \right)$$

- Solve for A and B so that coefficients of $\log(k)$ and constant terms cancel

More general Stokey-Lucas notation: Let $F(x_t, x_{t+1})$ be flow payoff

Sequence problem: Find $V(x)$ such that

$$V(x_0) = \sup_{\{x_{t+1}\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t F(x_t, x_{t+1})$$

subject to x_{t+1} in some feasible set $\Gamma(x_t)$, with x_0 given

Bellman equation: Find $V(x)$ such that

$$V(x) = \sup_{x' \in \Gamma(x)} \left\{ F(x, x') + \beta V(x') \right\}$$

for all x in the state space

4. Bellman operator

- In discrete time, Bellman equation is a **functional equation**
- Define the **Bellman operator** B, operating on function $w(\cdot)$, as

$$(Bw)(x) \equiv \max_{x'} \left\{ F(x, x') + \beta w(x') \right\} \quad \text{for all } x$$

- **Operators** are maps from one function space to another (learn functional analysis!)
- The value function $V(\cdot)$ is a fixed point of the operator B:

$$\begin{aligned} (BV)(x) &= \max_{x'} \left\{ F(x, x') + \beta w(x') \right\} \\ &= V(x) \end{aligned}$$

- This idea is useful in numerical analysis: Suppose we guess V^0 and look for fixed point

$$\lim_{n \to \infty} B^n V^0 = V$$

- The Bellman operator is a **contraction mapping** under some conditions
- This tells us that the Bellman operator converges (and we can use this to construct numerical fixed point algorithms)
- For example: If an operator B maps a complete metric space into itself and is a contraction mapping, then it has a unique fixed point
- This is a very important idea. And you should read about this on your own.

Part 3: Competitive Equilibrium

1. Environment

- Time is discrete with $t = 0, 1, \ldots$ and there is no uncertainty

Households. Consider a **representative household** with preferences

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

- $u(\cdot)$ is **flow utility**: units of c_t "dollars" or "apples", units of $u(c_t)$ "utils"
- Household owns capital and faces budget constraint

$$k_{t+1} = i_t + (1 - \delta)k_t \qquad \text{and} \qquad c_t + i_t = r_t k_t + w_t$$

  where r_t is rental rate of capital, w_t is wage

- Household endowed with 1 unit of time, inelastically supplied as labor

Firms. Consider a **representative firm** that operates technology

$$y_t = f(k_t, \ell_t)$$

and earns profit $\Pi_t = y_t - r_t k_t - w_t \ell_t$

- **Assumptions:** $f(\cdot)$ is constant-returns and firms are perfectly competitive
- Implies 0 profits and

$$r_t = f_k(k_t, \ell_t) \qquad \text{and} \qquad w_t = f_\ell(k_t, \ell_t)$$

Market clearing. How many markets are there?

$$y_t = c_t + i_t$$
$$\ell_t = 1$$

What about capital?

Definition. Given initial capital k_0, a **competitive equilibrium** comprises an allocation $\{y_t, \ell_t, k_{t+1}, c_t, i_t\}$ and prices $\{w_t, r_t\}$ such that

1. Households solve: taking as given $\{r_t, w_t\}$, $\max_{\{c_t\}} \sum_{t=0}^{\infty} \beta^t u(c_t)$ subject to $k_{t+1} = i_t + (1-\delta)k_t$ and $i_t + c_t = r_t k_t + w_t$ as well as $c_t, k_{t+1}, i_t \geq 0$
2. Firms solve: taking as given $\{r_t, w_t\}$, $\max_{k_t, \ell_t} y_t - r_t k_t - w_t \ell_t$ subject to $y_t = f(k_t, \ell_t)$ and $k_t, \ell_t \geq 0$
3. Markets for goods, labor and capital clear