Introduction

Thomas H. Jørgensen

2023

Outline

1 This Course

2 Models

3 Programming in Python

4 Dynamic Programming

Introduction

- **Teacher:** Thomas Jørgensen
  (Associate Professor, University of Copenhagen)

Introduction

- **Teacher:** Thomas Jørgensen
  (Associate Professor, University of Copenhagen)
- **Focus:** "Dynamic labor and family economics"
- **Topics:**
  Labor supply of couples
  Career costs of children
  Family formation and dissolution

Introduction

- **Teacher:** Thomas Jørgensen
  (Associate Professor, University of Copenhagen)
- **Focus:** "Dynamic labor and family economics"
- **Topics:**
  Labor supply of couples
  Career costs of children
  Family formation and dissolution
- **Tools:**
  Dynamic programming in Python
- **Prerequisite:**
  Introduction to Programming and Numerical Analysis

Introduction

- **Teacher:** Thomas Jørgensen
  (Associate Professor, University of Copenhagen)
- **Focus:** "Dynamic labor and family economics"
- **Topics:**
  Labor supply of couples
  Career costs of children
  Family formation and dissolution
- **Tools:**
  Dynamic programming in Python
- **Prerequisite:**
  Introduction to Programming and Numerical Analysis
- Plan for today:
  1. Course description
  2. Programming in Python
  3. Introduction to Dynamic Programming

Teaching Methods

- **Lectures:** Wednesday 12-15
  2 hours lecture
  1 hour of problem-solving (e.g. programming)

Teaching Methods

- **Lectures:** Wednesday 12-15
  2 hours lecture
  1 hour of problem-solving (e.g. programming)
- **Content:**
  1. Discussions of research papers on different *topics* related to *household behavior over the life cycle*
  2. Discussions of simple *models* to investigate central mechanisms
  3. Explanation of computer code and how to solve these models

Teaching Methods

- **Lectures:** Wednesday 12-15
  2 hours lecture
  1 hour of problem-solving (e.g. programming)
- **Content:**
  1. Discussions of research papers on different *topics* related to *household behavior over the life cycle*
  2. Discussions of simple *models* to investigate central mechanisms
  3. Explanation of computer code and how to solve these models
- **Code:**
  I will provide code.
  You will be asked to modify it to answer certain questions.

Teaching Methods

- **Lectures:** Wednesday 12-15
  2 hours lecture
  1 hour of problem-solving (e.g. programming)
- **Content:**
  1. Discussions of research papers on different *topics* related to *household behavior over the life cycle*
  2. Discussions of simple *models* to investigate central mechanisms
  3. Explanation of computer code and how to solve these models
- **Code:**
  I will provide code.
  You will be asked to modify it to answer certain questions.
- **Material:**
  Web: `https://sites.google.com/view/householdbehavior`
  GitHub:
  `https://github.com/ThomasHJorgensen/HouseholdBehaviorCourse`

Course Plan (preliminary)

Introduction

1. Introduction
2. Dynamic Programming and Structural Estimation

Part 1: Labor Supply

3. Static and Dynamic labor supply
4. Dynamic labor supply and learning by doing
5. Career costs of children
6. *No lecture: Work on assignment*
7. Household Labor Supply and Taxes
8. Household Labor Supply and Child-Related Transfers

Part 2: Family Formation and Dissolution

9. Marriage and Divorce.
10. Household Bargaining: Limited Commitment
11. Divorce Law and Labor Supply.
12. Taxes, Transfers and Intra-Household Inequality.
13. Fertility.

Outroduction

14. Buffer

Assignments and Exam

- **Exam** (Portfolio):
  1. 3 individual assignments.
     $+$ peer feedback.
  2. 48 hour individual take-home exam.
     Model formulation, code modification, simulations, economic interpretations.

Assignments and Exam

- **Exam** (Portfolio):
  1. 3 individual assignments.
     + peer feedback.
  2. 48 hour individual take-home exam.
     Model formulation, code modification, simulations, economic interpretations.
- 3 individual **assignments** (hand-in on Absalon)
  1. Based on our dynamic **labor supply** model
     Deadline: March 17 (no lecture that week)
  2. Based on our dynamic **bargaining** model
     Deadline: April 28
  3. **Free**: Formulate a research question + model + data.
     Deadline: May 12
     Deadline for peer feedback: May 19
- All feedback can be used to improve assignments before exam date

Outline

1. This Course
2. Models
3. Programming in Python
4. Dynamic Programming

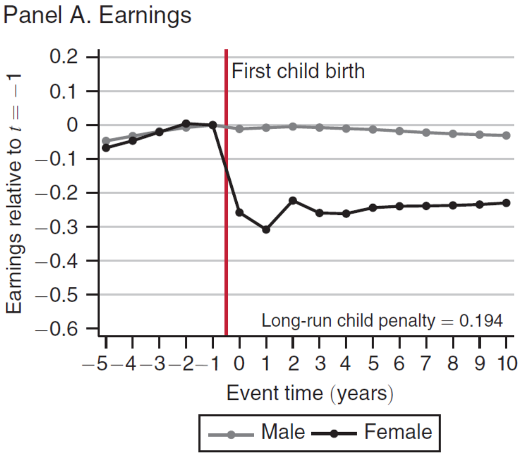
Models! What are they good for!?

- We will use **empirical regularities as motivation**

Models! What are they good for!?

- We will use **empirical regularities as motivation**

Example: "Child penalty" (Kleven, Landais and Søgaard, 2019)



Models! What are they good for!?

- We will use **economic models to understand/quantify behavior**

Models! What are they good for!?

- We will use **economic models to understand/quantify behavior**

  Example: "Career costs of children" (Adda, Dustmann and Stevens, 2017)
  "We estimate a dynamic life cycle model of labor supply, fertility, and savings, incorporating occupational choices, with specific wage paths and skill atrophy that vary over the career. This allows us to understand the trade-off between occupational choice and desired fertility, as well as sorting both into the labor market and across occupations. We **quantify** the life cycle career costs associated with children, how they **decompose** into loss of skills during interruptions, lost earnings opportunities, and selection into more child-friendly occupations. We analyze the **long-run effects of policies** that encourage fertility and show that they are considerably smaller than short-run effects."

Models! What are they good for!?

- I view models as an
  Internally consistent framework to study behavior of interest

Models! What are they good for!?

- I view models as an
  Internally consistent framework to study behavior of interest

- or bluntly...
  Equations that describe a small part of reality...

Models! What are they good for!?

- I view models as an
  Internally consistent framework to study behavior of interest

- or bluntly...
  Equations that describe a small part of reality...

> *"All models are wrong, but some are useful"*
>
> *George E.P. Box*

Models! What are they good for!?

- Why ever use a model then?

Models! What are they good for!?

- Why ever use a model then?

- **The Lucas critique:** *Behavioral rules change with policy*
  $\Rightarrow$ policy advice can not rely on estimated behavioral rules (reduced-form estimates using existing variation)
  $\Rightarrow$ we need to estimate *structural (deep) parameters*

  > *"Invariance of parameters in an economic model is not, of course, a property which can be assured in advance, but it seems reasonable to hope that neither tastes nor technology vary systematically with variations in counter-cyclical policies." (Lucas, 1977)*

Models! What are they good for!?

- **Economic laboratory**
  The usefulness of a model lies in which types of mechanisms are included and thus which *counterfactual* scenarios the model can inform about.

Models! What are they good for!?

- **Economic laboratory**
  The usefulness of a model lies in which types of mechanisms are included and thus which *counterfactual* scenarios the model can inform about.
- **Counterfactual simulations**:
  Implications of not-yet implemented policy reforms (e.g. child subsidies)
  Responses to changes in the economic environment (e.g. wages)
  Quantifying the importance of mechanisms (e.g. risk)

Models! What are they good for!?

- **Economic laboratory**
  The usefulness of a model lies in which types of mechanisms are included and thus which *counterfactual* scenarios the model can inform about.
- **Counterfactual simulations**:
  Implications of not-yet implemented policy reforms (e.g. child subsidies)
  Responses to changes in the economic environment (e.g. wages)
  Quantifying the importance of mechanisms (e.g. risk)
- **Example:**
  A model of *retirement timing of couples* likely cannot inform us about the effects of changing *school fees*.

Models! What are they good for!?

- **Economic laboratory**
  The usefulness of a model lies in which types of mechanisms are included and thus which *counterfactual* scenarios the model can inform about.
- **Counterfactual simulations**:
  Implications of not-yet implemented policy reforms (e.g. child subsidies)
  Responses to changes in the economic environment (e.g. wages)
  Quantifying the importance of mechanisms (e.g. risk)
- **Example:**
  A model of *retirement timing of couples* likely cannot inform us about the effects of changing *school fees*.
  But it might be able to predict how changes in the *early retirement scheme* changes *labor supply at older ages*.

Models! What are they good for!?

- **Economic laboratory**
  The usefulness of a model lies in which types of mechanisms are included and thus which *counterfactual* scenarios the model can inform about.
- **Counterfactual simulations**:
  Implications of not-yet implemented policy reforms (e.g. child subsidies)
  Responses to changes in the economic environment (e.g. wages)
  Quantifying the importance of mechanisms (e.g. risk)
- **Example:**
  A model of *retirement timing of couples* likely cannot inform us about the effects of changing *school fees*.
  But it might be able to predict how changes in the *early retirement scheme* changes *labor supply at older ages*.

→ Tight link between **research question** and model formulation.

Learning Objectives

See all at `https://kurser.ku.dk/course/a%c3%98kk08427u/2022-2023`

- **Knowledge (two):**
  Define, formulate and interpret *models* of household behaviour
  Account for backwards induction and how to *solve* dynamic programming models

Learning Objectives

See all at `https://kurser.ku.dk/course/a%c3%98kk08427u/2022-2023`

- **Knowledge (two):**
  Define, formulate and interpret *models* of household behaviour
  Account for backwards induction and how to *solve* dynamic programming models

- **Skills (one):**
  Analyze *counterfactual* policy reform simulations from simple and more complex models of household behavior

Learning Objectives

See all at `https://kurser.ku.dk/course/a%c3%98kk08427u/2022-2023`

- **Knowledge (two):**
  Define, formulate and interpret *models* of household behaviour
  Account for backwards induction and how to *solve* dynamic programming models

- **Skills (one):**
  Analyze *counterfactual* policy reform simulations from simple and more complex models of household behavior

- **Competences (two):**
  Discuss and evaluate *research* on household behavior over the life cycle
  Modify computer *code* to analyze small changes to simple models

How to read a research paper (in this course)

- Each lecture will be based on 1 mandatory (*) **research paper**
  What is the main *research question*?
  What is the *(empirical) motivation*?
  What are the central *mechanisms in the model*?
  What is the *simplest model* in which we could capture these?
  Challenging: Research frontier.

How to read a research paper (in this course)

- Each lecture will be based on 1 mandatory (*) **research paper**
  What is the main *research question*?
  What is the *(empirical) motivation*?
  What are the central *mechanisms in the model*?
  What is the *simplest model* in which we could capture these?
  Challenging: Research frontier.

- **How to read** a research paper in this course?
  - **Focus** on the questions above
    How do the questions interact and inform each other?
  - Try not to **get stuck** in too many details!
    (we can discuss some in class if you want)
  - Research papers often include many **"bells and whistles"**
  - Read ~**30 min before** each lecture.
    See reading-guide for each lecture

Outline

1. This Course
2. Models
3. Programming in Python
4. Dynamic Programming

Programming in Python

- **Only very simple**/unrealistic models can be solved with pen/paper
  $\rightarrow$ We need numerical methods

Programming in Python

- **Only very simple**/unrealistic models can be solved with pen/paper
  $\rightarrow$ We need numerical methods

- **Purpose** of you programming:
  Learn best by implementing!
  Appreciate bottlenecks and challenges
  Better understanding of frontier research
  Set you free...!

Programming in Python

- **Only very simple**/unrealistic models can be solved with pen/paper
  $\rightarrow$ We need numerical methods

- **Purpose** of you programming:
  Learn best by implementing!
  Appreciate bottlenecks and challenges
  Better understanding of frontier research
  Set you free...!

- **Goal** is to keep things simple!
  No fancy numerical tricks (at least in beginning)
  Code should be "intuitive"
  $\rightarrow$ slow...
  (might have to do more efficient code)

Programming in Python

- **Setup** as
  Introduction to Programming and Numerical Analysis
  `https://sites.google.com/view/numeconcph-introprog/`
  - Jupyter Lab (execution of code)
  - Visual Studio Code (editor, Python modules)
- **Installation** guide:
  `https://sites.google.com/view/numeconcph-introprog/guides/installation`
- **Packages** (all by Jeppe Druedahl):
  EconModel
  consav
  See "`01. Introduction to EconModel and consav.ipynb`"

Outline

1 This Course

2 Models

3 Programming in Python

4 Dynamic Programming

Introduction to Dynamic Programming (DP)

- **Agents maximize** expected discounted sum of utility throughout life
  Maximize wrt. $\{\mathcal{C}_t\}_{t=1}^{T}$
  Forward looking $\rightarrow$ dynamic
  Assume *optimal* behavior in all periods

Introduction to Dynamic Programming (DP)

- **Agents maximize** expected discounted sum of utility throughout life
  Maximize wrt. $\{\mathcal{C}_t\}_{t=1}^{T}$
  Forward looking $\rightarrow$ dynamic
  Assume *optimal* behavior in all periods
- **Bellman Equation**:

$$V_t(\mathcal{S}_t) = \max_{\mathcal{C}_t} U(\mathcal{C}_t, \mathcal{S}_t) + \beta \mathbb{E}[V_{t+1}(\mathcal{S}_{t+1}) | \mathcal{C}_t, \mathcal{S}_t]$$

$$\text{s.t.}$$

$$\mathcal{S}_{t+1} \sim F(\mathcal{C}_t, \mathcal{S}_t)$$

Introduction to Dynamic Programming (DP)

- **Agents maximize** expected discounted sum of utility throughout life
  Maximize wrt. $\{\mathcal{C}_t\}_{t=1}^{T}$
  Forward looking $\rightarrow$ dynamic
  Assume *optimal* behavior in all periods
- **Bellman Equation**:

$$V_t(\mathcal{S}_t) = \max_{\mathcal{C}_t} U(\mathcal{C}_t, \mathcal{S}_t) + \beta \mathbb{E}[V_{t+1}(\mathcal{S}_{t+1})|\mathcal{C}_t, \mathcal{S}_t]$$

$$\text{s.t.}$$

$$\mathcal{S}_{t+1} \sim F(\mathcal{C}_t, \mathcal{S}_t)$$

  - $V_t(\mathcal{S}_t)$: Indirect util, Value today of *states*, $\mathcal{S}_t$ (all relevant knowledge).

Introduction to Dynamic Programming (DP)

- **Agents maximize** expected discounted sum of utility throughout life
  Maximize wrt. $\{\mathcal{C}_t\}_{t=1}^{T}$
  Forward looking $\rightarrow$ dynamic
  Assume *optimal* behavior in all periods
- **Bellman Equation**:

$$V_t(\mathcal{S}_t) = \max_{\mathcal{C}_t} U(\mathcal{C}_t, \mathcal{S}_t) + \beta \mathbb{E}[V_{t+1}(\mathcal{S}_{t+1}) | \mathcal{C}_t, \mathcal{S}_t]$$

s.t.

$$\mathcal{S}_{t+1} \sim F(\mathcal{C}_t, \mathcal{S}_t)$$

  - $V_t(\mathcal{S}_t)$: Indirect util, Value today of *states*, $\mathcal{S}_t$ (all relevant knowledge).
  - $U(\mathcal{C}_t, \mathcal{S}_t)$: flow-utility

Introduction to Dynamic Programming (DP)

- **Agents maximize** expected discounted sum of utility throughout life
  Maximize wrt. $\{\mathcal{C}_t\}_{t=1}^T$
  Forward looking $\rightarrow$ dynamic
  Assume *optimal* behavior in all periods
- **Bellman Equation**:

$$V_t(\mathcal{S}_t) = \max_{\mathcal{C}_t} U(\mathcal{C}_t, \mathcal{S}_t) + \beta \mathbb{E}[V_{t+1}(\mathcal{S}_{t+1})|\mathcal{C}_t, \mathcal{S}_t]$$

s.t.

$$\mathcal{S}_{t+1} \sim F(\mathcal{C}_t, \mathcal{S}_t)$$

  - $V_t(\mathcal{S}_t)$: Indirect util, Value today of *states*, $\mathcal{S}_t$ (all relevant knowledge).
  - $U(\mathcal{C}_t, \mathcal{S}_t)$: flow-utility
  - $\beta \mathbb{E}[V_{t+1}(\mathcal{S}_{t+1})|\mathcal{C}_t, \mathcal{S}_t]$: expected discounted value of next-period

Introduction to Dynamic Programming (DP)

- **Agents maximize** expected discounted sum of utility throughout life
  Maximize wrt. $\{\mathcal{C}_t\}_{t=1}^{T}$
  Forward looking $\rightarrow$ dynamic
  Assume *optimal* behavior in all periods
- **Bellman Equation**:

$$V_t(\mathcal{S}_t) = \max_{\mathcal{C}_t} U(\mathcal{C}_t, \mathcal{S}_t) + \beta \mathbb{E}[V_{t+1}(\mathcal{S}_{t+1})|\mathcal{C}_t, \mathcal{S}_t]$$

s.t.

$$\mathcal{S}_{t+1} \sim F(\mathcal{C}_t, \mathcal{S}_t)$$

  - $V_t(\mathcal{S}_t)$: Indirect util, Value today of *states*, $\mathcal{S}_t$ (all relevant knowledge).
  - $U(\mathcal{C}_t, \mathcal{S}_t)$: flow-utility
  - $\beta \mathbb{E}[V_{t+1}(\mathcal{S}_{t+1})|\mathcal{C}_t, \mathcal{S}_t]$: expected discounted value of next-period
  - $\mathcal{S}_{t+1} \sim F(\mathcal{C}_t, \mathcal{S}_t)$: transition density of states (fcn of $\mathcal{C}_t$!)
    (there might be other constraints)

Backwards Induction

- Solved by **backwards induction**

1. Start with last/terminal period, T (no future)

$$V_T(\mathcal{S}_T) = \max_{\mathcal{C}_T} U(\mathcal{C}_T, \mathcal{S}_T)$$

Backwards Induction

- Solved by **backwards induction**

1. Start with last/terminal period, T (no future)

$$V_T(\mathcal{S}_T) = \max_{\mathcal{C}_T} U(\mathcal{C}_T, \mathcal{S}_T)$$

2. Go to second-to-last (we now know period-T optimal behavior for all $\mathcal{S}_T$)

$$V_{T-1}(\mathcal{S}_{T-1}) = \max_{\mathcal{C}_{T-1}} U(\mathcal{C}_{T-1}, \mathcal{S}_{T-1}) + \beta \mathbb{E}[V_T(\mathcal{S}_T) | \mathcal{C}_{T-1}, \mathcal{S}_{T-1}]$$

s.t.

$$\mathcal{S}_T \sim F(\mathcal{C}_{T-1}, \mathcal{S}_{T-1})$$

Backwards Induction

- Solved by **backwards induction**

1. Start with last/terminal period, T (no future)

$$V_T(\mathcal{S}_T) = \max_{\mathcal{C}_T} U(\mathcal{C}_T, \mathcal{S}_T)$$

2. Go to second-to-last (we now know period-T optimal behavior for all $\mathcal{S}_T$)

$$V_{T-1}(\mathcal{S}_{T-1}) = \max_{\mathcal{C}_{T-1}} U(\mathcal{C}_{T-1}, \mathcal{S}_{T-1}) + \beta\mathbb{E}[V_T(\mathcal{S}_T)|\mathcal{C}_{T-1}, \mathcal{S}_{T-1}]$$
$$\text{s.t.}$$
$$\mathcal{S}_T \sim F(\mathcal{C}_{T-1}, \mathcal{S}_{T-1})$$

3. Continue backwards...

Introduction to Dynamic Programming (DP)

- **On a computer**, everything is discrete $\rightarrow$ arrays + loops

Introduction to Dynamic Programming (DP)

- **On a computer**, everything is discrete $\rightarrow$ arrays + loops
- ***Numerical*** Dynamic Programming
  - Backwards induction on arrays
  - Grids
  - Interpolation
- See "`02. Consumption-Saving Model.ipynb`"

Next Time

- **Next time:**
  Dynamic programming with *uncertainty*
  Structural *estimation*

- **Literature:**
  Gourinchas and Parker (2002): "Consumption Over the Life Cycle"
  - **Read** before lecture
  - **Reading guide:**
    Section 1: Introduction – *Key* (page 50 is not that important)
    Section 2: Model – *Key*, we will discuss. Do not get stuck.
    Section 3: Estimation method (SMM). *Key*, we will discuss.
    Section 4: First stage calibrations. Skim fast.
    Section 5: Data. Skim fast.
    Section 6: Results. Focus on 6.1. Figures 5 and 7 are main results.

References I

Adda, J., C. Dustmann and K. Stevens (2017): "The Career Costs of Children," *Journal of Political Economy*, 125(2), 293–337.

Gourinchas, P.-O. and J. A. Parker (2002): "Consumption Over the Life Cycle," *Econometrica*, 70(1), 47–89.

Kleven, H. J., C. Landais and J. E. Søgaard (2019): "Children and gender inequality: Evidence from Denmark," *American Economic Journal: Applied Economics*, 11, 181–209.