

# Lecture 1: Introduction

## Dynamic Programming

Thomas Jørgensen



# Outline for today

- ➊ **Introduction** to the course
- ➋ **Example** of what we will be doing
- ➌ The course **plan**



# Why dynamic programming?

- **Policy recommendations** are central for all social science
- Need **models** for **counter-factual analysis**
- **Realistic models** → **no *analytical* solution**
  - ① dynamic
  - ② multi-dimensional
  - ③ uncertainty
  - ④ heterogeneity
  - ⑤ learning
- Solving such models require **dynamic programming**
- Estimation gives you **laboratory economies**



# Dynamic programming

- ① You should expect a **steep learning curve**
- ② But it is an **extremely versatile tool**: Micro, macro, labor, industrial organization, household finance, partial and general equilibrium, games, etc. etc.
- ③ And **increasingly important** – better algorithms, more powerful computers



# What am I using dynamic programming for?

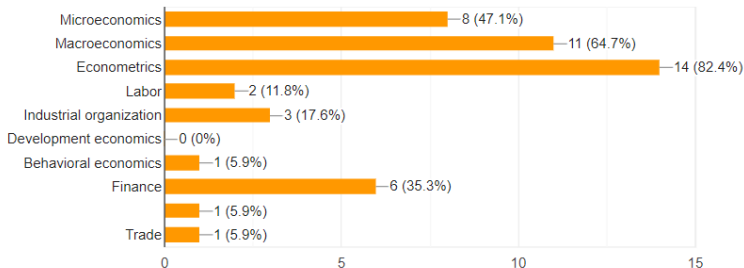
- I am an Assistant Professor here at the Department (PhD from here too)
- My **research** is in the intersection of
  - ① Household behavior
  - ② Micro data
  - ③ Computational methods
- **Examples:**
  - ① Consumption and saving over the life cycle
  - ② Fertility behavior and planning
  - ③ Consumers information set and learning
  - ④ Computational methods for solving and estimating life-cycle models
    - as fast as possible



# What are your interests in this course?

What are your core fields of interest? (in economics...)

17 responses



# Organization

- **Setup:**
  - **Lectures:** Tuesday (8-10 in CSS 2-0-36) and Wednesday (10-12 in CSS 2-2-30)
  - **Classes:** Tuesday (10-12)
- **Your suggested work flow:**
  - ① Prepare for the lectures (at first max 15 min.)  
slides will often be available evening before
  - ② Participate actively in the lectures
  - ③ Re-read the slides and texts after the lecture  
and ask a question on **Padlet**  
([https://padlet.com/thomas\\_jorgensen1/DP](https://padlet.com/thomas_jorgensen1/DP))
  - ④ Read the exercises beforehand
  - ⑤ Work on the exercises in class by yourself in groups  
*with help from Johan*



# Exam

- **Oral exam in group project**
  - ① April 9, 8–12: Presentation of project (descriptions)
  - ② April 16: **Submit project description**
  - ③ May 1, 9–15: 2. Supervision day
  - ④ May 24: **Submit project**
  - ⑤ Week 25: **Oral exam** (25 min)
- **Examples include:**
  - ① **Replication** of (selected) results from some paper
  - ② **Tests** (e.g. Monte Carlo) of various computational solution and estimation methods
  - ③ Presentation and implementation of a **new method** not taught in the course
  - ④ **New model, new data**
- **Formalities:** Around 8-12 pages per person (font size 12, double spacing and 3 cm margins)





# Books

- ① **Adda and Cooper (2003), Dynamic Economics:**
  - Chapter 2-3  $\approx$  lecture 1-5
  - Chapter 4  $\approx$  lecture 6
  - Chapter 5-10 no equivalent lectures
    - read about the topics that interests you
- ② **Judd (1998), Numerical Methods:** Good for general reference and tools

## Mathematical background

- ① Stockey and Lucas (1989), Recursive Methods in Economics
- ② Bertsekas (2005, 2012), Dynamic Programming and Optimal Control Vol. I-II
- ③ Puterman (2008), Markov Decision Processes: Discrete Stochastic Dynamic Programming



# MATLAB

- **Download version 2016b or later**
  - ① Go to **kunet.ku.dk**
  - ② Under Systemadgange click on **Softwarebibliotek**
  - ③ Follow the instructions
- **Online MATLAB Course for Student of Economics:**  
See link on *Absalon*
- **Video** with Matlab essentials for this course  
10mins, link on *Absalon*



# The Simplest Consumption Model

- **Model:**

- ① You have  $M$  goods for consumption
- ② You have  $T$  years back of your life
- ③ Your per year utility is  $\sqrt{C}$
- ④  $C \in \{0, 1, 2, \dots\} = \mathbb{N}$  is consumption.
- ⑤ You discount future utility by a factor of  $\beta < 1$

- **Formally:**

$$V^*(M) = \max_{C_1, C_2, \dots, C_T} \left\{ \sqrt{C_1} + \beta \sqrt{C_2} + \beta^2 \sqrt{C_3} + \dots + \beta^{T-1} \sqrt{C_T} \right\}$$

s.t.

$$M = C_1 + C_2 + \dots + C_T$$

$$C_t \in \mathbb{N}$$

- **Solution:**

- $T = 1$ :  $C_T = M$ .
- $T = 2$ : Try all  $C_{T-1} \leq M$  where  $C_T = M - C_{T-1}$ .
- $T = 3$ : Try all  $C_{T-2} \leq M$  and  $C_{T-1} \leq M - C_{T-2}$  where  $C_T = M - C_{T-1} - C_{T-2}$ .



---

**Algorithm 1:** Algorithm for  $T = 1$ 


---

**input :**  $M$  (goods)**output:**  $C_T^*$  (optimal choice)

```

1  $V^* = -\infty$ 
2 for  $C_T = 0$  to  $M$  do
3    $V = \sqrt{C_T}$ 
4   if  $V > V^*$  then
5      $V^* = V$ 
6    $C_T^* = C_T$ 

```

---

$$V^*(M) = \max_{C_1, C_2, \dots, C_T} \sqrt{C_t} + \beta \sqrt{C_{t+1}} + \beta^2 \sqrt{C_{t+2}} + \dots + \beta^T \sqrt{C_T}$$

s.t.

$$M = C_1 + C_2 + \dots + C_T$$



---

**Algorithm 2:** Algorithm for  $T = 2$ 


---

**input :**  $M$  (goods)  
**output:**  $C_T^*, C_{T-1}^*$  (optimal choices)

```

1  $V^* = -\infty$ 
2 for  $C_{T-1} = 0$  to  $M$  do
3    $V = \sqrt{C_{T-1}} + \beta\sqrt{M - C_{T-1}}$ 
4   if  $V > V^*$  then
5      $V^* = V$ 
6      $C_{T-1}^* = C_{T-1}$ 
7      $C_T^* = M - C_{T-1}$ 

```

---

$$\begin{aligned}
 V^*(M) &= \max_{C_1, C_2, \dots, C_T} \sqrt{C_t} + \beta\sqrt{C_{t+1}} + \beta^2\sqrt{C_{t+2}} + \dots + \beta^T\sqrt{C_T} \\
 &\text{s.t.} \\
 M &= C_1 + C_2 + \dots + C_T
 \end{aligned}$$



---

**Algorithm 3:** Algorithm for  $T = 3$ 


---

**input :**  $M$  (goods)**output:**  $C_T^*, C_{T-1}^*, C_{T-2}^*$  (optimal choices)

```

1  $V^* = -\infty$ 
2 for  $C_{T-2} = 0$  to  $M$  do
3   for  $C_{T-1} = 0$  to  $M - C_{T-2}$  do
4      $V = \sqrt{C_{T-2}} + \beta \sqrt{C_{T-1}} + \beta^2 \sqrt{M - C_{T-1} - C_{T-2}}$ 
5     if  $V > V^*$  then
6        $V^* = V$ 
7        $C_{T-1}^* = C_{T-1}$ 
8        $C_{T-2}^* = C_{T-2}$ 
9        $C_T^* = M - C_{T-1} - C_{T-2}$ 

```

---

- Write it on a poster and put on the white board for  $T = 4$  (5 mins)



# Backwards induction - intuition

- **Can we do something smarter?**
- **Period  $T$ :** Value of having  $M$  goods? Consume everything

$$V_T^*(M) = \sqrt{M}$$

- **Period  $T - 1$ :** Value of having  $M$  goods?
  - today:  $\sqrt{C_{T-1}}$
  - having  $M - C_{T-1}$  left tomorrow:  $V_T^*(M - C_{T-1})$
  - $\rightarrow$  seek to maximize the sum:

$$V_{T-1}^*(M) = \max_{C_{T-1}} \left\{ \sqrt{C_{T-1}} + \beta V_T^*(M - C_{T-1}) \right\}$$

- **Period  $T - 2$ :** Value of having  $M$  goods?
  - today:  $\sqrt{C_{T-2}}$
  - having  $M - C_{T-2}$  left tomorrow:  $V_{T-1}^*(M - C_{T-2})$
  - $\rightarrow$  seek to maximize the sum:

$$V_{T-2}^*(M) = \max_{C_{T-2}} \left\{ \sqrt{C_{T-2}} + \beta V_{T-1}^*(M - C_{T-2}) \right\}$$



# Backwards induction - formula for all $t$

$$\begin{aligned}V_T^*(M) &= \sqrt{M} \\V_{T-1}^*(M) &= \max_{C_{T-1}} \left\{ \sqrt{C_{T-1}} + \beta V_T^*(M - C_{T-1}) \right\} \\V_{T-2}^*(M) &= \max_{C_{T-2}} \left\{ \sqrt{C_{T-2}} + \beta V_{T-1}^*(M - C_{T-2}) \right\} \\V_{T-3}^*(M) &= \max_{C_{T-3}} \left\{ \sqrt{C_{T-3}} + \beta V_{T-2}^*(M - C_{T-3}) \right\} \\&\dots \\V_t^*(M) &= \max_{C_t} \left\{ \sqrt{C_t} + \beta V_{t+1}^*(M - C_t) \right\}\end{aligned}$$





---

**Algorithm 4:** Find  $V_t^*$  given  $V_{t+1}^*$  [called "find\_V" later]
 

---

**input :**  $M$  (goods)  
            $V_{t+1}^*[\bullet]$  (value next period)  
**output:**  $V_t^*[\bullet]$  (value of optimal choice)  
            $C_t^*[\bullet]$  (optimal choice)

```

1  for  $m = 0$  to  $M$  do
2     $i_M = m + 1$ 
3     $V_t^*[i_M] = -\infty$ 
4    for  $C_t = 0$  to  $m$  do
5       $i_{next} = (m - C_t) + 1$ 
6       $V = \sqrt{C_t} + \beta V_{t+1}^*[i_{next}]$ 
7      if  $V > V_t^*[i_M]$  then
8         $V_t^*[i_M] = V$ 
9         $C_t^*[i_M] = C_t$ 
  
```

---

$$V_t^*(M) = \max_{C_t} \left\{ \sqrt{C_t} + \beta V_{t+1}^*(M - C_t) \right\}$$



- On a computer, we only have discrete arrays of data:

$i$	$M_T$	$V_T^*(M_T)$
1	0	$\sqrt{0}$
2	1	$\sqrt{1}$
3	2	$\sqrt{2}$
4	3	$\sqrt{3}$
$\vdots$	$\vdots$	$\vdots$

$i$	$M_{T-1}$	$V_{T-1}^*(M_{T-1})$
1	0	$\max_{C_{T-1}} \sqrt{C_{T-1}} + \beta V_T^*[0 - C_{T-1} + 1]$
2	1	$\max_{C_{T-1}} \sqrt{C_{T-1}} + \beta V_T^*[1 - C_{T-1} + 1]$
3	2	$\max_{C_{T-1}} \sqrt{C_{T-1}} + \beta V_T^*[2 - C_{T-1} + 1]$
4	3	$\max_{C_{T-1}} \sqrt{C_{T-1}} + \beta V_T^*[3 - C_{T-1} + 1]$
$\vdots$	$\vdots$	$\vdots$



- On a computer, we only have discrete arrays of data:

$i$	$M_T$	$V_T^*(M_T)$
1	0	$\sqrt{0}$
2	1	$\sqrt{1}$
3	2	$\sqrt{2}$
4	3	$\sqrt{3}$
$\vdots$	$\vdots$	$\vdots$

$i$	$M_{T-1}$	$V_{T-1}^*(M_{T-1})$
1	0	$\max_{C_{T-1}} \sqrt{C_{T-1}} + \beta V_T^*[0 - C_{T-1} + 1]$
2	1	$\max_{C_{T-1}} \sqrt{C_{T-1}} + \beta V_T^*[1 - C_{T-1} + 1]$
3	2	$C_{T-1} = 0 \rightarrow \sqrt{0} + \beta V_T^*[2 - 0 + 1]$
4	3	$\max_{C_{T-1}} \sqrt{C_{T-1}} + \beta V_T^*[3 - C_{T-1} + 1]$
$\vdots$	$\vdots$	$\vdots$



- On a computer, we only have discrete arrays of data:

$i$	$M_T$	$V_T^*(M_T)$
1	0	$\sqrt{0}$
2	1	$\sqrt{1}$
3	2	$\sqrt{2}$
4	3	$\sqrt{3}$
$\vdots$	$\vdots$	$\vdots$

$i$	$M_{T-1}$	$V_{T-1}^*(M_{T-1})$
1	0	$\max_{C_{T-1}} \sqrt{C_{T-1}} + \beta V_T^*[0 - C_{T-1} + 1]$
2	1	$\max_{C_{T-1}} \sqrt{C_{T-1}} + \beta V_T^*[1 - C_{T-1} + 1]$
3	2	$C_{T-1} = 1 \rightarrow \sqrt{1} + \beta V_T^*[2 - 1 + 1]$
4	3	$\max_{C_{T-1}} \sqrt{C_{T-1}} + \beta V_T^*[3 - C_{T-1} + 1]$
$\vdots$	$\vdots$	$\vdots$



- On a computer, we only have discrete arrays of data:

$i$	$M_T$	$V_T^*(M_T)$
1	0	$\sqrt{0}$
2	1	$\sqrt{1}$
3	2	$\sqrt{2}$
4	3	$\sqrt{3}$
$\vdots$	$\vdots$	$\vdots$

$i$	$M_{T-1}$	$V_{T-1}^*(M_{T-1})$
1	0	$\max_{C_{T-1}} \sqrt{C_{T-1}} + \beta V_T^*[0 - C_{T-1} + 1]$
2	1	$\max_{C_{T-1}} \sqrt{C_{T-1}} + \beta V_T^*[1 - C_{T-1} + 1]$
3	2	$C_{T-1} = 2 \rightarrow \sqrt{2} + \beta V_T^*[2 - 2 + 1]$
4	3	$\max_{C_{T-1}} \sqrt{C_{T-1}} + \beta V_T^*[3 - C_{T-1} + 1]$
$\vdots$	$\vdots$	$\vdots$



---

**Algorithm 5:** Find all  $V_t^*$ 


---

**input :**  $M$  (goods)

**output:**  $V_t^*[\bullet]$  for all  $t$   
 $C_t^*[\bullet]$  for all  $t$

- 1  $V_T^*[m+1] = \sqrt{m}$  for all  $m = 0$  to  $M$
  - 2 **for**  $t = T - 1$  **to** 1 **do**
  - 3      $V_t^*[\bullet], C_t^*[\bullet] = \text{find\_V}(V_{t+1}^*[\bullet])$
- 

• **Simulate from period  $t$  with *any*  $M$ :**

- $C_t = C_t^*(M)$
- $C_{t+1} = C_{t+1}^*(M - C_t)$
- $C_{t+3} = C_{t+2}^*(M - C_t - C_{t+1})$
- etc.



# Vocabulary

- ① *States* = number of goods,  $M_t$
- ② *Choices* = consumption,  $C_t \in \mathcal{C}(M_t) = \{0, 1, 2, \dots, M_t\}$
- ③ *Payoff function* = utility,  $\sqrt{C_t}$
- ④ *Transition function* = next-period states,  
 $M_{t+1} = \Gamma(M, C_t) = M_t - C_t$
- ⑤ *Value function* = value today,  $V_t^*(M_t)$
- ⑥ *Continuation value* = value *after* today,  
 $\beta V_{t+1}^*(M_{t+1}) = \beta V_{t+1}^*(\Gamma(M_t, C_t))$
- ⑦ *Policy function* = optimal choice,  $C_t^*(M_t)$



# Extensions - the rest of the course

- ① Stochastic elements?
- ② Continuous states and/or choices?
- ③ Multiple states and/or choices?
- ④ Infinitely many periods?
- ⑤ Estimation?
- ⑥ Multiple agents? (equilibrium and games)





# Plan

- ① **Theory and tools** (lecture 1-6).
- ② **Consumption-saving models** (lecture 7-11, with JD as guest in week 9).
- ③ **Estimation of dynamic discrete choice models** (lecture 12-17)

And you should as soon as possible figure out a group and topic for a term paper!



# Until next time

- **Ensure that you understand:**
  - ① Algorithm 4 and 5
  - ② The vocabulary
- Go to **PadLet** and ask *or answer* a question ([https://padlet.com/thomas\\_jorgensen1/DP](https://padlet.com/thomas_jorgensen1/DP))
- **Think about what to do if:**
  - ① New goods arrived **stochastically** so that

$$M_{t+1} = \begin{cases} M_t - C_t + 1 & \text{with probability } \pi \in (0, 1) \\ M_t - C_t & \text{else} \end{cases}$$

- ② Consumption was **continuous**,  
i.e.  $C_t \in \mathbb{R}$  and not just  $C_t \in \mathbb{N}$
- **Now:** go to **b.socratic.com** room name  
“**DynamicProgramming**” and answer a few questions

