

Exercise class 11

Introduction to Programming
and Numerical Analysis

Class 3 and 6

Annasofie Marckstrøm Olesen
Spring 2024

UNIVERSITY OF COPENHAGEN



Data project feedback

Problem set 7

Dynamic programming

Feedback on Data Project

In general: Good job!

I've approved your projects if you:

- Do at least *some* cleaning/processing of the data
- Do at least *some* analysis (calculations or plots)

Common feedback points:

- It's a bit too easy to just plot raw data - you should at the minimum compute some descriptive statistics. Preferably some split-apply-combine.
- Good idea to use modules (py-file) for data cleaning.
- Clean up your repositories: Update README, delete files you are not using.

Plan for today: Problem set 7

1. Now-15.50: Work on optimization problems
2. 15.55-16.00: We talk about optimization in class
3. 16.00-16.15: Break
4. 16.15-16.35: Intro to dynamic programming
5. 16.35-17.00: Work on dynamic programming problems

Tip: Focus on the optimization tasks - If you get stuck with plotting, move on to the next task.

Terminology...

State variable: A variable determining the *state of the world* as it is when I make my decision. State variables evolve over time according to a *transition process*.

Example: Cash-on-hand in consumption-savings model.

Terminology...

State variable: A variable determining the *state of the world* as it is when I make my decision. State variables evolve over time according to a *transition process*.

Example: Cash-on-hand in consumption-savings model.

Choice variable: The variable I *choose* to maximize utility (*also control variable*).

Example: Consumption in consumption-savings model.

Terminology...

State variable: A variable determining the *state of the world* as it is when I make my decision. State variables evolve over time according to a *transition process*.

Example: Cash-on-hand in consumption-savings model.

Choice variable: The variable I *choose* to maximize utility (*also control variable*).

Example: Consumption in consumption-savings model.

Policy function: The optimal *choice* as a function of *state*.

Terminology...

State variable: A variable determining the *state of the world* as it is when I make my decision. State variables evolve over time according to a *transition process*.

Example: Cash-on-hand in consumption-savings model.

Choice variable: The variable I *choose* to maximize utility (*also control variable*).

Example: Consumption in consumption-savings model.

Policy function: The optimal *choice* as a function of *state*.

Value function: The total value of today's *state*: Utility today + discounted value of tomorrow's state.

Terminology...

State variable: A variable determining the *state of the world* as it is when I make my decision. State variables evolve over time according to a *transition process*.

Example: Cash-on-hand in consumption-savings model.

Choice variable: The variable I *choose* to maximize utility (*also control variable*).

Example: Consumption in consumption-savings model.

Policy function: The optimal *choice* as a function of *state*.

Value function: The total value of today's *state*: Utility today + discounted value of tomorrow's state.

Solution: Set of (approximate) policy and value functions describing optimal behavior as a function of the state.

Two period consumption-savings model

Period 1:

$$V_1(m_1) = \max_{c_1} u(c_1) + \beta \mathbb{E} [V_2(m_2)]$$

$$s.t. \quad m_2 = R(m_1 - c_1) + y_2$$

$$y_2 \sim U(0, 1)$$

Two period consumption-savings model

Period 1:

$$\begin{aligned} V_1(m_1) &= \max_{c_1} u(c_1) + \beta \mathbb{E}[V_2(m_2)] \\ s.t. \quad m_2 &= R(m_1 - c_1) + y_2 \\ y_2 &\sim U(0, 1) \end{aligned}$$

Period 2:

$$\begin{aligned} V_2(m_2) &= \max_{c_2} u(c_2) \\ s.t. \quad m_2 &\geq c_2 \end{aligned}$$

Two period consumption-savings model

State variable(s)?

Choice variable(s)?

Policy function(s)?

Two period consumption-savings model

State variable(s)? m_1, m_2

Choice variable(s)?

Policy function(s)?

Two period consumption-savings model

State variable(s)? m_1, m_2

Choice variable(s)? c_1, c_2

Policy function(s)?

Two period consumption-savings model

State variable(s)? m_1, m_2

Choice variable(s)? c_1, c_2

Policy function(s)? $c_1^*(m_1), c_2^*(m_2)$

Two period consumption-savings model - Solution

Period 1:

$$V_1(m_1) = \max_{c_1} u(c_1) + \beta \mathbb{E} [V_2(m_2)]$$

$$s.t. \quad m_2 = R(m_1 - c_1) + y_2$$

$$y_2 \sim U(0, 1)$$

Two period consumption-savings model - Solution

Period 1:

$$\begin{aligned} V_1(m_1) &= \max_{c_1} u(c_1) + \beta \mathbb{E} [V_2(m_2)] \\ \text{s.t. } m_2 &= R(m_1 - c_1) + y_2 \\ y_2 &\sim U(0, 1) \end{aligned}$$

The optimal choice today depends on the **value function tomorrow** - which in turn depends on the optimal choice tomorrow.

So we need to solve tomorrow's problem first - **backwards induction**.

Two period consumption-savings model - Solution

Period 2:

$$\begin{aligned} V_2(m_2) &= \max_{c_2} u(c_2) \\ \text{s.t. } m_2 &\geq c_2 \end{aligned}$$

This is just a standard constrained optimization problem. We can solve this for any value of m_2 , we like. This gives us consumption $c_2^*(m_2)$ and a value $V_2(m_2)$ for any given input m_2 .

Two period consumption-savings model - Solution

Period 2:

$$\begin{aligned} V_2(m_2) &= \max_{c_2} u(c_2) \\ \text{s.t. } m_2 &\geq c_2 \end{aligned}$$

This is just a standard constrained optimization problem. We can solve this for any value of m_2 , we like. This gives us consumption $c_2^*(m_2)$ and a value $V_2(m_2)$ for any given input m_2 .

Problem: Evaluating $V_2(m_2)$ is **costly** because it involves an optimization.

Two period consumption-savings model - Solution

Period 2:

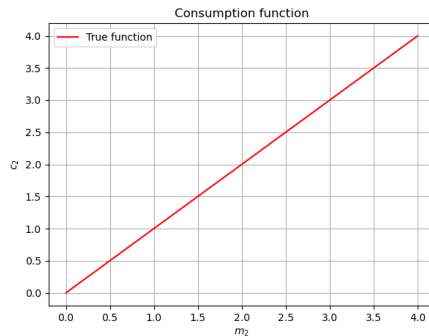
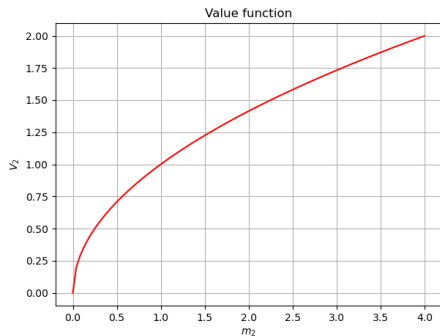
$$\begin{aligned} V_2(m_2) &= \max_{c_2} u(c_2) \\ \text{s.t. } m_2 &\geq c_2 \end{aligned}$$

This is just a standard constrained optimization problem. We can solve this for any value of m_2 , we like. This gives us consumption $c_2^*(m_2)$ and a value $V_2(m_2)$ for any given input m_2 .

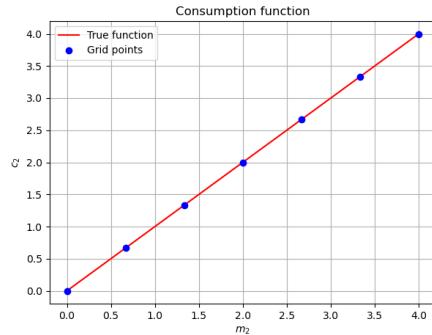
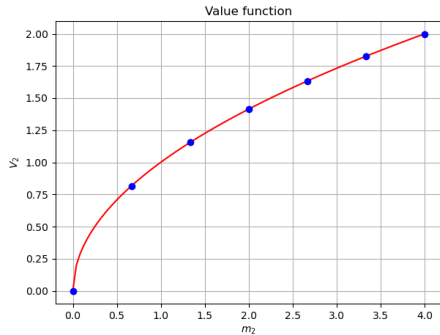
Problem: Evaluating $V_2(m_2)$ is **costly** because it involves an optimization.

Solution: We evaluate V_2 and c_2^* on a **grid** over m_2 and approximate the rest of the functions by **interpolation**.

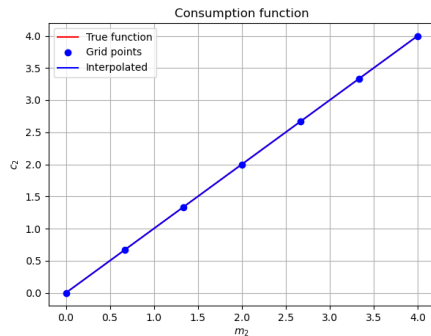
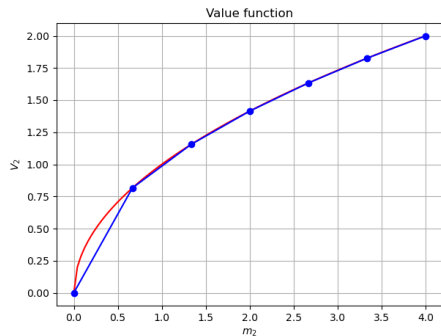
Two period consumption-savings model - Solution



Two period consumption-savings model - Solution



Two period consumption-savings model



Two period consumption-savings model - Simulation

Back to period 1...

$$\begin{aligned} V_1(m_1) &= \max_{c_1} u(c_1) + \beta \mathbb{E} [\check{V}_2(m_2)] \\ s.t. \quad m_2 &= R(m_1 - c_1) + y_2 \\ y_2 &\sim U(0, 1) \end{aligned}$$

Now we know the (approximate) period 2 value function, so we can solve the problem in period 1.

We still need to take expectations with respect to $\check{V}_2(m_2)$, where m_2 is stochastic...

Two period consumption-savings model - Simulation

Back to period 1...

$$\begin{aligned} V_1(m_1) &= \max_{c_1} u(c_1) + \beta \mathbb{E} [\check{V}_2(m_2)] \\ s.t. \quad m_2 &= R(m_1 - c_1) + y_2 \\ y_2 &\sim U(0, 1) \end{aligned}$$

Now we know the (approximate) period 2 value function, so we can solve the problem in period 1.

We still need to take expectations with respect to $\check{V}_2(m_2)$, where m_2 is stochastic... \rightarrow for instance Monte Carlo Integration for continuous m_2 , weighted average for discrete m_2 ! 1

Two period consumption-savings model - Solution

We can solve this problem over a grid of m_1 . After doing so we have:

- Grid over m_1 : \vec{m}_1
- Value $V_1(\vec{m}_1)$ and $c_1^*(\vec{m}_1) \rightarrow$ Interpolators $\check{V}_1(m_1)$ and $\check{c}_1^*(m_1)$
- Transition rule for state variable: $m_2 = R(m_1 - c_1) + y_2$
- Grid over m_2 : \vec{m}_2
- Value $V_2(\vec{m}_2)$ and $c_2^*(\vec{m}_2) \rightarrow$ Interpolators $\check{V}_2(m_2)$ and $\check{c}_2^*(m_2)$

So for any given initial state m_1 , we can **simulate** behavior.

Two period consumption-savings model - Simulation

We simulate forwards:

Two period consumption-savings model - Simulation

We simulate forwards:

1. Set initial state, m_1 .

Two period consumption-savings model - Simulation

We simulate forwards:

1. Set initial state, m_1 .
2. Interpolate period 1 consumption, $c_1^* = \check{c}_1^*(m_1)$

Two period consumption-savings model - Simulation

We simulate forwards:

1. Set initial state, m_1 .
2. Interpolate period 1 consumption, $c_1^* = \check{c}_1^*(m_1)$
3. Compute next period state:

Two period consumption-savings model - Simulation

We simulate forwards:

1. Set initial state, m_1 .
2. Interpolate period 1 consumption, $c_1^* = \check{c}_1^*(m_1)$
3. Compute next period state:
 - 3.1 Draw income shock $y_2 \sim U(0, 1)$

Two period consumption-savings model - Simulation

We simulate forwards:

1. Set initial state, m_1 .
2. Interpolate period 1 consumption, $c_1^* = \check{c}_1^*(m_1)$
3. Compute next period state:
 - 3.1 Draw income shock $y_2 \sim U(0, 1)$
 - 3.2 Compute $m_2 = R(m_1 - c_1^*) + y_2$

Two period consumption-savings model - Simulation

We simulate forwards:

1. Set initial state, m_1 .
2. Interpolate period 1 consumption, $c_1^* = \check{c}_1^*(m_1)$
3. Compute next period state:
 - 3.1 Draw income shock $y_2 \sim U(0, 1)$
 - 3.2 Compute $m_2 = R(m_1 - c_1^*) + y_2$
4. Interpolate period 2 consumption, $c_2 = \check{c}_2^*(m_2)$

Two period consumption-savings model - Simulation

We simulate forwards:

1. Set initial state, m_1 .
2. Interpolate period 1 consumption, $c_1^* = \check{c}_1^*(m_1)$
3. Compute next period state:
 - 3.1 Draw income shock $y_2 \sim U(0, 1)$
 - 3.2 Compute $m_2 = R(m_1 - c_1^*) + y_2$
4. Interpolate period 2 consumption, $c_2 = \check{c}_2^*(m_2)$
5. Repeat for N consumers

Two period consumption-savings model

Summing up:

- Solve **backwards** to find interpolators of c_1^* and c_2^*
- Simulate **forwards** using interpolators of c_1^* and c_2^*

Approach generalizes to an arbitrary number of periods - start at the last period and iterate backwards.

Next time...

Physical lecture:

- Model project

Video lectures:

- Structural estimation
- OLG and Ramsey models

Exercises

- Work on data project