QM1 – Tutorial Session 3

Grégoire Sempé
gregoire.sempe@psemail.eu

Paris School of Economics, Université Paris 1 Panthéon-Sorbonne

October 14, 2024

Today's Agenda

1. **Theory**: Bellman Equation, example of a consumption-saving program
   - Recursive form of the deterministic problem
   - Markov chains and stochastic dynamic programming
   - Contraction mapping theorem and backward iteration

2. **Computational**: Value Function Iteration
   - On-grid Value Function Iteration
   - Off-grid VFI & Euler errors

   $\Rightarrow$ Pseudo-code of the algorithm (whiteboard)

A consumption saving program, without uncertainty

For the PS, you will be asked to solve a <u>stochastic</u> neoclassical growth model. In this tutorial, we will take the example of a consumption (c_t) saving (a_{t+1}) program in partial eq. You will have to think carefully about the differences between the two models for the problem set...

A consumption saving program, without uncertainty

For the PS, you will be asked to solve a stochastic neoclassical growth model. In this tutorial, we will take the example of a consumption (c_t) saving (a_{t+1}) program in partial eq. You will have to think carefully about the differences between the two models for the problem set...

Challenge: Find the sequence of $\{c_t, a_{t+1}\}_{\forall s \geq t}$ that solves:

A consumption saving program, without uncertainty

For the PS, you will be asked to solve a stochastic neoclassical growth model. In this tutorial, we will take the example of a consumption (c_t) saving (a_{t+1}) program in partial eq. You will have to think carefully about the differences between the two models for the problem set…

Challenge: Find the sequence of $\{c_t, a_{t+1}\}_{\forall s \geq t}$ that solves:

$$V_t(a_t) \equiv \max_{\{c_s, a_{s+1}\}_{\forall s \geq t}} \sum_{s \geq t} \beta^{s-t} u(c_s) \quad \text{s.t.} \quad \begin{cases} c_s + a_{s+1} = (1 + r_s)a_s + \bar{y} & \forall s \geq t \\ a_s \geq 0 & \forall s \geq t \\ a_t \quad \text{is given} & \end{cases}$$

A consumption saving program, without uncertainty

For the PS, you will be asked to solve a stochastic neoclassical growth model. In this tutorial, we will take the example of a consumption (c_t) saving (a_{t+1}) program in partial eq. You will have to think carefully about the differences between the two models for the problem set...

Challenge: Find the sequence of $\{c_t, a_{t+1}\}_{\forall s \geq t}$ that solves:

$$V_t(a_t) \equiv \max_{\{c_s, a_{s+1}\}_{\forall s \geq t}} \sum_{s \geq t} \beta^{s-t} u(c_s) \quad \text{s.t.} \quad \begin{cases} c_s + a_{s+1} = (1 + r_s) a_s + \bar{y} & \forall s \geq t \\ a_s \geq 0 & \forall s \geq t \\ a_t \quad \text{is given} & \end{cases}$$

Issue: This problem is subject to the curse of dimensionality...

$\rightarrow$ How would you reduce its dimensionality to find a global solution?

Dimensionality reduction of the problem

1/ Algebra: reduce the number of control variables using the budget constraint

$$V_t(a_t) = \max_{\{a_{s+1}\}_{\forall s \geq t}} \sum_{s \geq t} \beta^{s-t} u\left((1 + r_s)a_s + \bar{y} - a_{s+1}\right) \quad \text{s.t.} \quad a_{s+1} \geq 0 \quad \forall s \geq t$$

Dimensionality reduction of the problem

1/ Algebra: reduce the number of control variables using the budget constraint

$$V_t(a_t) = \max_{\{a_{s+1}\}_{\forall s \geq t}} \sum_{s \geq t} \beta^{s-t} u\left((1+r_s)a_s + \bar{y} - a_{s+1}\right) \quad \text{s.t.} \quad a_{s+1} \geq 0 \quad \forall s \geq t$$

2/ We can write the problem in the state space (recursive form)

$$V_t(a_t) = \max_{a_{t+1}} \left\{ u(c_t) + \beta \underbrace{\left[\max_{\{a_{s+1}\}_{\forall s \geq t+1}} u(c_{t+1}) + \sum_{s \geq t+1} \beta^{s-(t+1)} u(c_s) \right]}_{V_{t+1}(a_{t+1})} \right\}$$

$$\text{s.t.} \quad c_t = (1+r_t)a_t + \bar{y} - a_{t+1} \qquad \text{and} \quad a_{t+1} \geq 0$$

The deterministic Bellman Equation

Bellman Equation reduces the problem to ”today’s choice” given ”tomorrow” optimal

$$V_t(a_t) = \underbrace{\mathcal{T}\{V_{t+1}\}(a_t)}_{\text{Bellman operator}} = \max_{a_{t+1}} u\left((1+r)a_t + \bar{y} - a_{t+1}\right) + \beta V_{t+1}(a_{t+1}) \qquad \text{s.t.} \quad a_{t+1} \geq 0$$

The deterministic Bellman Equation

Bellman Equation reduces the problem to ”today’s choice” given ”tomorrow” optimal

$$V_t(a_t) = \underbrace{\mathcal{T}\{V_{t+1}\}(a_t)}_{\text{Bellman operator}} = \max_{a_{t+1}} u\left((1+r)a_t + \bar{y} - a_{t+1}\right) + \beta V_{t+1}(a_{t+1}) \qquad \text{s.t.} \quad a_{t+1} \geq 0$$

1. Problem is to find the optimal **policy function** $a_{t+1} = g_t(a_t)$, instead of a sequence

   $\Rightarrow$ Solution to the curse of dimensionality!

The deterministic Bellman Equation

Bellman Equation reduces the problem to ”today’s choice” given ”tomorrow” optimal

$$V_t(a_t) = \underbrace{\mathcal{T}\{V_{t+1}\}(a_t)}_{\text{Bellman operator}} = \max_{a_{t+1}} u\left((1+r)a_t + \bar{y} - a_{t+1}\right) + \beta V_{t+1}(a_{t+1}) \qquad \text{s.t.} \quad a_{t+1} \geq 0$$

1. Problem is to find the optimal **policy <u>function</u>** $a_{t+1} = g_t(a_t)$, instead of a sequence

   $\Rightarrow$ Solution to the curse of dimensionality!

2. $V_t(a_t)$ is the lifetime discounted sum of utility, given states and with optimal policies

The deterministic Bellman Equation

Bellman Equation reduces the problem to ”today’s choice” given ”tomorrow” optimal

$$V_t(a_t) = \underbrace{\mathcal{T}\{V_{t+1}\}(a_t)}_{\text{Bellman operator}} = \max_{a_{t+1}} u\left((1+r)a_t + \bar{y} - a_{t+1}\right) + \beta V_{t+1}(a_{t+1}) \qquad \text{s.t.} \quad a_{t+1} \geq 0$$

1. Problem is to find the optimal **policy function** $a_{t+1} = g_t(a_t)$, instead of a sequence
   $\Rightarrow$ Solution to the curse of dimensionality!
2. $V_t(a_t)$ is the lifetime discounted sum of utility, given states and with optimal policies
3. $a_{t+1} \in \Gamma(a_t)$ is the **Choice correspondence**

The deterministic Bellman Equation

Bellman Equation reduces the problem to "today's choice" given "tomorrow" optimal

$$V_t(a_t) = \underbrace{\mathcal{T}\{V_{t+1}\}(a_t)}_{\text{Bellman operator}} = \max_{a_{t+1}} u\left((1+r)a_t + \bar{y} - a_{t+1}\right) + \beta V_{t+1}(a_{t+1}) \qquad \text{s.t.} \quad a_{t+1} \geq 0$$

1. Problem is to find the optimal **policy <u>function</u>** $a_{t+1} = g_t(a_t)$, instead of a sequence

   $\Rightarrow$ Solution to the curse of dimensionality!

2. $V_t(a_t)$ is the lifetime discounted sum of utility, given states and with optimal policies

3. $a_{t+1} \in \Gamma(a_t)$ is the **Choice correspondence**

4. The Bellman Equation maps a function into a function. It is a *functional equation*

Adding uncertainty: Markov chains

Uncertainty is a key feature of economic behavior, often modeled using **Markov processes**

$\hookrightarrow$ In our example, take households' earning, subject to uncertainty $\omega \sim AR(1)$

Adding uncertainty: Markov chains

Uncertainty is a key feature of economic behavior, often modeled using **Markov processes**

$\hookrightarrow$ In our example, take households' earning, subject to uncertainty $\omega \sim AR(1)$

1/ Markov processes have desirable properties for dynamic programming

   (a) **Memoryless**: "Future states depend only on the current state"

   (b) **Stationary transitions**: Transition probabilities are time-invariant

   (c) **Discretization**: Approximate continuous processes by markov chains (Tauchen, Rowenhorst)

Adding uncertainty: Markov chains

Uncertainty is a key feature of economic behavior, often modeled using **Markov processes**

$\hookrightarrow$ In our example, take households' earning, subject to uncertainty $\omega \sim AR(1)$

1/ Markov processes have desirable properties for dynamic programming

(a) **Memoryless**: "Future states depend only on the current state"

(b) **Stationary transitions**: Transition probabilities are time-invariant

(c) **Discretization**: Approximate continuous processes by markov chains (Tauchen, Rowenhorst)

2/ Markov chains are defined by

- A discrete set of states Ω, with a probability transition matrix $\Pi = (\pi_{\omega,\omega'})_{\forall \omega,\omega' \in \Omega^2}$
- An initial distribution μ_0

Stochastic Bellman Equation

Earnings are stochastic $\rightarrow$ replace $\bar{y}$ by ω. The Bellman Equation becomes

$$
\begin{aligned}
V_t(\omega_t, a_t) &= \mathcal{T}\{V_{t+1}\}(\omega_t, a_t) \\
&= \max_{a_{t+1}} u\left((1+r)a_t + \omega_t - a_{t+1}\right) + \beta \mathbb{E}_{\omega_{t+1}|\omega_t} V_{t+1}(\omega_{t+1}, a_{t+1}) \\
&= \max_{a_{t+1}} u\left((1+r)a_t + \omega_t - a_{t+1}\right) + \beta \sum_{\omega_{t+1}} \pi_{\omega_t, \omega_{t+1}} V_{t+1}(\omega_{t+1}, a_{t+1}) \\
\text{s.t.} \quad & a_{t+1} \geq 0 \qquad \forall\ \omega_t, a_t \in \Omega \times \mathbb{R}_+
\end{aligned}
$$

Two useful applications of the Bellman Equation

1. **At steady state**, the Value Function is the unique[1] fixed point to the Bellman operator

$$\text{CMT} : V^* \text{ s.t. } V^*(\omega, a) = \mathcal{T}\{V^*\}(\omega, a) = \max_{a'} u\left((1+r)a + \omega - a'\right) + \beta \mathbb{E}_{\omega'|\omega} V^*(\omega', a')$$

[1] By the application of the contraction mapping theorem. See the lecture slides.

Two useful applications of the Bellman Equation

1. **At steady state**, the Value Function is the unique[1] fixed point to the Bellman operator

$$\text{CMT}: V^* \text{ s.t. } V^*(\omega, a) = \mathcal{T}\{V^*\}(\omega, a) = \max_{a'} u\left((1+r)a + \omega - a'\right) + \beta \mathbb{E}_{\omega'|\omega} V^*(\omega', a')$$

2. **Backward induction** (in finite time):

   1/ Start from a terminal condition $V_{T+1}(a_{T+1})$ e.g $V_{T+1}(a_{T+1}) = 0$ if T is large enough

   2/ Given a sequence of ω_t, the Value function at time $t < T+1$ is obtained by applying the Bellman Operator

$$V_t(\omega_t, a_t) = \mathcal{T}\{V_{t+1}\}(\omega_t, a_t) \qquad \forall a_t$$

[1] By the application of the contraction mapping theorem. See the lecture slides.

Two useful applications of the Bellman Equation

1. **At steady state**, the Value Function is the unique[1] fixed point to the Bellman operator

$$\text{CMT}: V^* \text{ s.t. } V^*(\omega, a) = \mathcal{T}\{V^*\}(\omega, a) = \max_{a'} u\left((1+r)a + \omega - a'\right) + \beta \mathbb{E}_{\omega'|\omega} V^*(\omega', a')$$

2. **Backward induction** (in finite time):

   1/ Start from a terminal condition $V_{T+1}(a_{T+1})$ e.g $V_{T+1}(a_{T+1}) = 0$ if T is large enough

   2/ Given a sequence of ω_t, the Value function at time $t < T+1$ is obtained by applying the Bellman Operator

$$V_t(\omega_t, a_t) = \mathcal{T}\{V_{t+1}\}(\omega_t, a_t) \qquad \forall a_t$$

$\Rightarrow$ We can find the sequence of optimal policies functions $\{a'_t(a_t)\}_{t=0}^{T}$

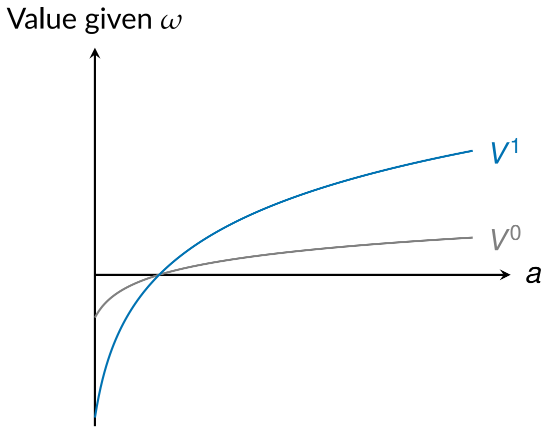
[1] By the application of the contraction mapping theorem. See the lecture slides.

Value Function Iteration

How to find the value function and policy functions at steady state?

Value Function Iteration

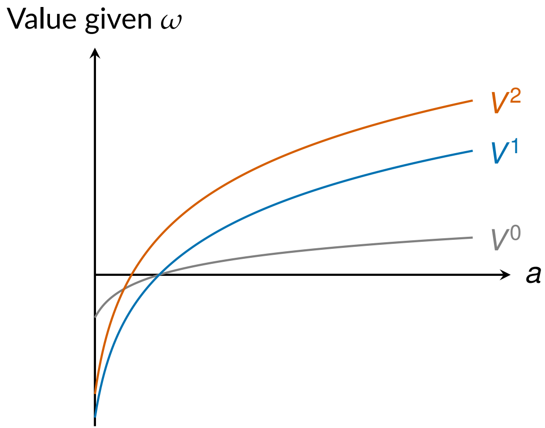
How to find the value function and policy functions at steady state?



- Guess an initial value function $V^0(\omega, a)$
- New guess: $V^1(\omega, a) = \mathcal{T}\{V^0\}(\omega, a)$

Value Function Iteration

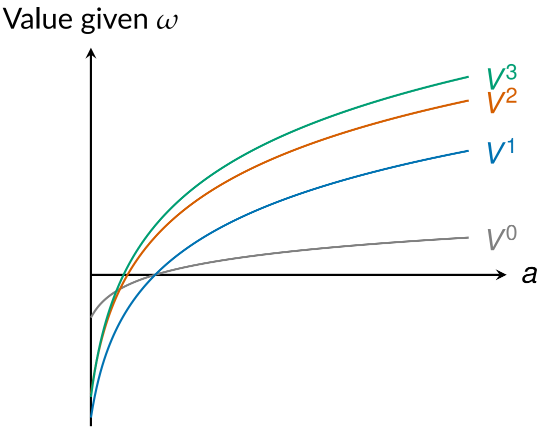
How to find the value function and policy functions at steady state?



- Guess an initial value function $V^0(\omega, a)$
- New guess: $V^1(\omega, a) = \mathcal{T}\{V^0\}(\omega, a)$
- Continue: $V^2(\omega, a) = \mathcal{T}\{V^1\}(\omega, a)$

Value Function Iteration

How to find the value function and policy functions at steady state?

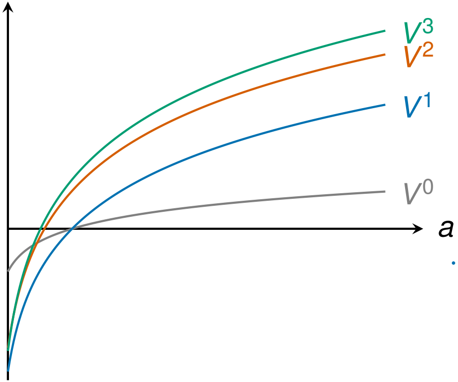


- Guess an initial value function $V^0(\omega, a)$
- New guess: $V^1(\omega, a) = \mathcal{T}\{V^0\}(\omega, a)$
- Continue: $V^2(\omega, a) = \mathcal{T}\{V^1\}(\omega, a)$
- Again: $V^3(\omega, a) = \mathcal{T}\{V^2\}(\omega, a)$

Value Function Iteration

How to find the value function and policy functions at steady state?

Value given ω



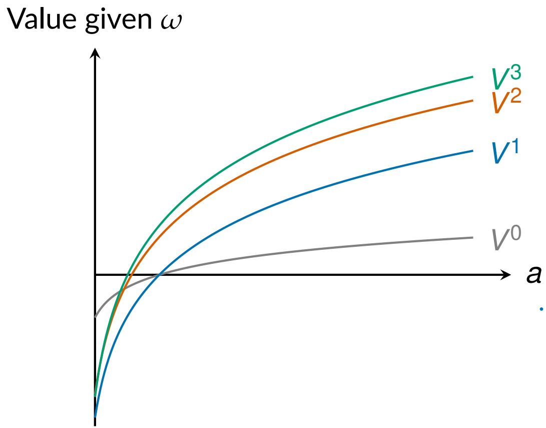
- Guess an initial value function $V^0(\omega, a)$
- New guess: $V^1(\omega, a) = \mathcal{T}\{V^0\}(\omega, a)$
- Continue: $V^2(\omega, a) = \mathcal{T}\{V^1\}(\omega, a)$
- Again: $V^3(\omega, a) = \mathcal{T}\{V^2\}(\omega, a)$

... Continue until "convergence" btw. functions

$$\| V^n - V^{(n-1)} \| \leq \varepsilon$$

Value Function Iteration

How to find the value function and policy functions at steady state?



- Guess an initial value function $V^0(\omega, a)$
- New guess: $V^1(\omega, a) = \mathcal{T}\{V^0\}(\omega, a)$
- Continue: $V^2(\omega, a) = \mathcal{T}\{V^1\}(\omega, a)$
- Again: $V^3(\omega, a) = \mathcal{T}\{V^2\}(\omega, a)$

... Continue until "convergence" btw. functions

$$\| V^n - V^{(n-1)} \| \leq \varepsilon$$

$\Longrightarrow$ Let's do a pseudo-code of the algorithm together!

How backward iteration and steady state are connected?

On the board explanation...