Optimal Control of Ordinary Differential Equations
SOD 311

Laurent Pfeiffer
Inria and CentraleSupélec, Université Paris-Saclay

Ensta-Paris
Paris-Saclay University
November 4, 2021

Lecture 4:
Numerical resolution of the HJB equation

- *Goal:* constructing a numerical scheme for the resolution of the HJB equation.
- *Issues:* time and space discretization, iterative schemes for the discretized equation, convergence analysis.

Problem formulation

Data:

- A parameter $\lambda > 0$, a compact subset U of $\mathbb{R}^m$.
- Two maps $f\colon (u, y) \in U \times \mathbb{R}^n \to \mathbb{R}^n$ and $\ell\colon (u, y) \in U \times \mathbb{R}^n \to \mathbb{R}$, bounded and Lipschitz continuous.

Problem:

- State equation: for $x \in \mathbb{R}^n$ and $u \in \mathcal{U}_\infty$, there is a unique solution $y[u, x]$ to the ODE

$$\dot{y}(t) = f(u(t), y(t)), \quad y(0) = x.$$

- Cost function W, for $u \in \mathcal{U}_\infty$ and $x \in \mathbb{R}^n$:

$$W(u, x) = \int_0^\infty e^{-\lambda t} \ell(u(t), y[u, x](t)) \, dt.$$

- Optimal control problem and value function V:

$$V(x) = \inf_{u \in \mathcal{U}_\infty} W(u, x). \qquad (P(x))$$

Dynamic programming

Given $\tau > 0$, the "**DP-mapping**"

$$\mathcal{T}\colon v \in BUC(\mathbb{R}^n) \mapsto \mathcal{T}v \in BUC(\mathbb{R}^n),$$

is defined by

$$\mathcal{T}v(x) = \inf_{u \in \mathcal{U}_\tau} \left(\int_0^\tau e^{-\lambda t} \ell(u(t), y(t))\, \mathrm{d}t + e^{-\lambda \tau} v(y[u,x](\tau)) \right).$$

Theorem 1

The DP-mapping is $e^{-\lambda\tau}$-Lipschitz continuous. The value function V is the unique solution to the fixed point equation

$$\mathcal{T}v = v, \quad v \in BUC(\mathbb{R}^n).$$

HJB equation

We define the **pre-Hamiltonian** H and the **Hamiltonian** $\mathcal{H}$ by

$$H(u,x,p) = \ell(u,x) + \langle p, f(u,x) \rangle,$$
$$\mathcal{H}(x,p) = \min_{u \in U} \; H(u,x,p).$$

Theorem 2

The value function is the unique viscosity solution to the HJB equation

$$\lambda V(x) - \mathcal{H}(x, \nabla V(x)) = 0.$$

Remark. The HJB equation can be **heuristically** derived by calculating a first-order Taylor expansion (with respect to τ) of the DP-mapping.

Towards numerics

- *Purpose:* computing a **numerical approximation** of V.
  - Yields a feedback.
  - Can be used to decouple (in time) the optimal control problem.
- *A bad idea:* discretizing the HJB equation by "brute force", e.g. in dimension 1:

$$\lambda V(x) - \mathcal{H}\Big(x, \frac{V(x+\delta x) - V(x)}{\delta x}\Big) = 0.$$

  This is doomed to failure!
- *Key idea:*
  - **discretize the DP-mapping**: $\mathcal{T} \rightsquigarrow \mathcal{T}_{\tau,h}$ in time and space,
  - **solve the fixed point equation**: $v = \mathcal{T}_{\tau,h} v$.

Time-discretization

Recall the definition of $\mathcal{T}$:

$$\mathcal{T}v(x) = \inf_{u \in \mathcal{U}_\tau} \left(\int_0^\tau e^{-\lambda t} \ell(u(t), y(t))\, \mathrm{d}t + e^{-\lambda \tau} v(y[u,x](\tau)) \right).$$

Ingredients for the **time-discretization**, assuming τ **small**:

$$\begin{aligned} \mathcal{U}_\tau &\rightsquigarrow \text{ a constant control on } (0,\tau) \\ \int_0^\tau e^{-\lambda t} \ell(u(t), y(t))\, \mathrm{d}t &\rightsquigarrow \tau \ell(u,x) \\ e^{-\lambda \tau} v(y[u,x](\tau)) &\rightsquigarrow (1-\lambda\tau) v(y[u,x](\tau)). \end{aligned}$$

Remarks:

- at the moment we do note try to simplify $y[u,x](\tau)$
- calculations similar to those for $\dot{\varphi}$.

Time-discretization

We fix now $\tau > 0$ such that $1 - \lambda\tau > 0$ (i.e. $\tau < 1/\lambda$) and define:

$$\mathcal{T}_\tau v(x) = \min_{u \in U} \Big(\tau \ell(u,x) + (1 - \lambda\tau) v\big(y[u,x](\tau)\big) \Big).$$

Remark: notation $y[u,x]$ extended to $u \in U$.

Lemma 3

The map $\mathcal{T}_\tau$ is well-defined from $BUC(\mathbb{R}^n)$ to $BUC(\mathbb{R}^n)$. It is Lipschitz with modulus $(1 - \lambda\tau)$ for the supremum norm.

Proof. Exercise (adapt ideas from the previous lecture).

Corollary 4

There exists a unique $V_\tau \in BUC(\mathbb{R}^n)$ such that $V_\tau = \mathcal{T}_\tau V_\tau$.

Time-discretization

Idea: we give an interpretation of V_τ as value function of a discretized optimal control problem.

Notation: $U^{\mathbb{N}}$ is the set of sequences $u = (u_k)_{k\in\mathbb{N}}$ such that $u_k \in U$, $\forall k \in \mathbb{N}$.

Control set and state equation: given $u \in U^{\mathbb{N}}$, define $y_\tau[u,x] = y[\mathrm{u},x]$, where $\mathrm{u} \in \mathcal{U}_\infty$ is defined by

$$\mathrm{u}(t) = u_k, \quad \text{for a.e. } t \in (k\tau,(k+1)\tau).$$

Cost: $W_\tau(u,x) = \tau \sum_{k=0}^{\infty} (1-\lambda\tau)^k \ell(u_k, y_\tau[u,x](k\tau))$.

Remark. We have "sampled" $\mathcal{U}_\infty$ and discretized $W(x,u)$.

Time-discretization

Theorem 5

Let us consider, for $x \in \mathbb{R}^n$, the optimal control problem

$$\hat{V}_\tau(x) = \inf_{u \in U^{\mathbb{N}}} W_\tau(u,x). \qquad (P_\tau(x))$$

It holds: $V_\tau(x) = \hat{V}_\tau(x)$.

Proof. It suffices to verify that

$$\hat{V}_\tau = \mathcal{T}_\tau \hat{V}_\tau,$$

i.e. to verify that $\hat{V}_\tau$ satisfies an appropriate dynamic programming principle.

Time-discretization

The flow property yields:

$$y_\tau[u,x](k\tau) = y_\tau[\tilde{u}, y_\tau[u_0,x](\tau)]((k-1)\tau),$$

where $\tilde{u} \in U^{\mathbb{N}}$ is defined by $\tilde{u}_k = u_{k+1}$. We have:

$$\begin{aligned} W_\tau(u,x) &= \tau\ell(u_0,x) + \tau\sum_{k=1}^{\infty}(1-\lambda\tau)^k \ell(u_k, y_\tau[u,x](k\tau)) \\ &= \tau\ell(u_0,x) + (1-\lambda\tau)\cdot \\ &\qquad \underbrace{\tau\sum_{k=1}^{\infty}(1-\lambda\tau)^{k-1}\ell\Big(\tilde{u}_{k-1}, y_\tau\big[\tilde{u}, y_\tau[u_0,x](\tau)\big]((k-1)\tau)\Big)}. \end{aligned}$$

Time-discretization

The flow property yields:

$$y_\tau[u,x](k\tau) = y_\tau[\tilde{u}, y_\tau[u_0,x](\tau)]((k-1)\tau),$$

where $\tilde{u} \in U^{\mathbb{N}}$ is defined by $\tilde{u}_k = u_{k+1}$. We have:

$$\begin{aligned} W_\tau(u,x) &= \tau\ell(u_0,x) + \tau\sum_{k=1}^{\infty}(1-\lambda\tau)^k \ell(u_k, y_\tau[u,x](k\tau)) \\ &= \tau\ell(u_0,x) + (1-\lambda\tau)\cdot \\ &\qquad \underbrace{\tau\sum_{k=1}^{\infty}(1-\lambda\tau)^{k-1}\ell\Big(\tilde{u}_{k-1}, y_\tau\big[\tilde{u}, y_\tau[u_0,x](\tau)\big]((k-1)\tau)\Big)}. \end{aligned}$$

Time-discretization

The flow property yields:

$$y_\tau[u,x](k\tau) = y_\tau[\tilde{u}, y_\tau[u_0,x](\tau)]((k-1)\tau),$$

where $\tilde{u} \in U^{\mathbb{N}}$ is defined by $\tilde{u}_k = u_{k+1}$. We have:

$$\begin{aligned} W_\tau(u,x) &= \tau\ell(u_0,x) + \tau\sum_{k=1}^{\infty}(1-\lambda\tau)^k \ell(u_k, y_\tau[u,x](k\tau)) \\ &= \tau\ell(u_0,x) + (1-\lambda\tau)\cdot \\ &\qquad \underbrace{\tau\sum_{k=0}^{\infty}(1-\lambda\tau)^k \ell\Big(\tilde{u}_k, y_\tau\big[\tilde{u}, y_\tau[u_0,x](\tau)\big](k\tau)\Big)}_{=W_\tau(\tilde{u}, y_\tau[u_0,x](\tau))}. \end{aligned}$$

Time-discretization

We obtain:

$$W_\tau(u,x) = \tau\ell(u_0,x) + (1-\lambda\tau)W_\tau(\tilde{u}, y_\tau[u_0,x](\tau)).$$

Proceeding as in the previous lecture, we arrive at:

$$\begin{aligned}
\hat{V}_\tau(x) &= \inf_{u\in U^{\mathbb{N}}} W_\tau(u,x) \\
&= \inf_{u_0\in U} \Big(\tau\ell(u_0,x) + (1-\lambda\tau) \inf_{\tilde{u}\in\mathbb{U}^N} W_\tau(\tilde{u}, y_\tau[u_0,x](\tau)) \Big) \\
&= \inf_{u_0\in U} \Big(\tau\ell(u_0,x) + (1-\lambda\tau)\hat{V}_\tau(y_\tau[u_0,x](\tau)) \Big) \\
&= \mathcal{T}_\tau \hat{V}_\tau(x).
\end{aligned}$$

Time-discretization

The analysis can be summarized with a commutative **diagram**:

Problem $P(x)$

$\downarrow$ Dynamic Prog.

$$V = \mathcal{T} V \xrightarrow{\text{Discretization}} V_\tau = \mathcal{T}_\tau V_\tau$$

Time-discretization

The analysis can be summarized with a commutative **diagram**:

Problem $P(x)$ $\xrightarrow{\text{Discretization}}$ Problem $P_\tau(x)$

Dynamic Prog. $\downarrow$

$$V_\tau = \mathcal{T}_\tau V_\tau$$

Time-discretization

The analysis can be summarized with a commutative **diagram**:

$$
\begin{array}{ccc}
\text{Problem } P(x) & \xrightarrow{\text{Discretization}} & \text{Problem } P_\tau(x) \\
\Big| & & \Big| \\
\text{Dynamic Prog.} & & \text{Dynamic Prog.} \\
\downarrow & & \downarrow \\
V = \mathcal{T} V & \xrightarrow{\text{Discretization}} & V_\tau = \mathcal{T}_\tau V_\tau
\end{array}
$$

The “discretization” and “dynamic programming” phases **commute**.

Space-discretization

We need to further simplify the operator $\mathcal{T}_\tau$.

Difficulties and solutions:

1. Impossible to manipulate (numerically) a function on $\mathbb{R}^n$.

2. Evaluation of $y_\tau[u,x](\tau)$?

Space-discretization

We need to further simplify the operator $\mathcal{T}_\tau$.

Difficulties and solutions:

1. Impossible to manipulate (numerically) a function on $\mathbb{R}^n$.
   - Store $v(x)$ for **finitely many points** x.
   - Value of v is needed at an arbitrary $x \to$ **interpolation**.
2. Evaluation of $y_\tau[u, x](\tau)$?

Space-discretization

We need to further simplify the operator $\mathcal{T}_\tau$.

Difficulties and solutions:

1. Impossible to manipulate (numerically) a function on $\mathbb{R}^n$.
   - Store $v(x)$ for **finitely many points** x.
   - Value of v is needed at an arbitrary $x \to$ **interpolation**.
2. Evaluation of $y_\tau[u,x](\tau)$?
   - Explicit Euler scheme: $y_\tau[u,x](\tau) = x + \tau f(u,x)$.
   - Many other possible schemes.

Space-discretization

Interpolation.

Let $\mathcal{G}$ be a countable subset of $\mathbb{R}^n$, called **grid**. We assume that there exists an **interpolation map**

$$\mu \colon \mathcal{G} \times \mathbb{R}^n \to [0,1]$$

such that for all $x \in \mathbb{R}^n$,

$$x = \sum_{y \in \mathcal{G}} \mu(y,x) y, \quad \sum_{y \in \mathcal{G}} \mu(y,x) = 1.$$

In words: each x is a **convex combination** of some points y of the grid, with weights $\mu(y,x)$.

Space-discretization

Notation: $L^\infty(\mathcal{G})$ is the space of bounded functions from $\mathcal{G}$ to $\mathbb{R}$.

Given $v \in L^\infty(\mathcal{G})$, let the **interpolation** $[v] \in L^\infty(\mathbb{R}^n)$ be defined by

$$[v](x) = \sum_{y \in \mathcal{G}} v(y)\mu(y, x).$$

In words: $[v](x)$ is the **convex combination** of the reals $v(y)$, for the weights $\mu(y, x)$.

Space-discretization

Example of grid and interpolation map.
A natural choice is $\mathcal{G} = \mathbb{Z}^n$. Let us construct a suitable μ_n.

Case $n = 1$. Let $x \in \mathbb{R}$, let $k \in \mathbb{Z}$ be such that $k \leq x < k + 1$. Then,

$$x = (k + 1 - x)k + (x - k)(k + 1).$$

Thus we can define:

$$\mu_1(y, x) = \begin{cases} (k + 1 - x) & \text{if } y = k \\ (x - k) & \text{if } y = k + 1 \\ 0 & \text{otherwise.} \end{cases}$$

Obviously, $\mu_1(y, x) \in [0, 1]$ and $\sum_{y \in \mathbb{Z}} \mu_1(y, x) = 1$.

Space-discretization

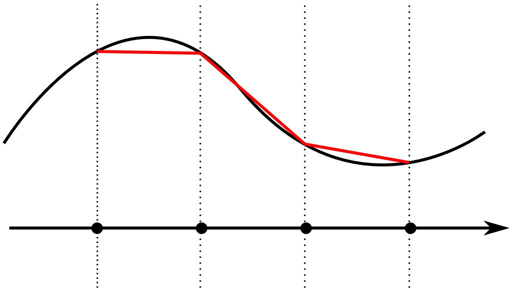

Figure: Interpolation in dimension 1

Space discretization

General case $n > 1$. Let $x = (x_1, ..., x_n) \in \mathbb{R}^n$.
Let $y = (y_1, ..., y_n) \in \mathbb{Z}^n$. Let us define $\mu_n(y, x)$ by

$$\mu_n(y, x) = \prod_{k=1}^{n} \mu_1(y_k, x_k) \in [0, 1].$$

Then we have

$$\begin{aligned} \sum_{y \in \mathbb{Z}^n} \mu_n(y, x) &= \sum_{y \in \mathbb{Z}^n} \left(\prod_{k=1}^{n} \mu_1(y_k, x_k) \right) \\ &= \prod_{k=1}^{n} \underbrace{\left(\sum_{y_k \in \mathbb{Z}} \mu_1(y_k, x_k) \right)}_{=1} = 1. \end{aligned}$$

Space discretization

Moreover,

$$\begin{aligned}
\sum_{y\in\mathbb{Z}^n} \mu_n(y,x)y &= \sum_{y\in\mathbb{Z}^n} \left(\prod_{k=1}^{n} \mu_1(y_k,x_k)(y_1,\dots,y_n)\right) \\
&= \sum_{y_1\in\mathbb{Z}} \cdots \sum_{y_n\in\mathbb{Z}} \Big(\mu_1(y_1,x_1)y_1, \mu_2(y_2,x_2)y_2, \dots, \mu_k(y_k,x_k)y_k\Big) \\
&= \left(\sum_{y_1\in\mathbb{Z}} \mu_1(y_1,x_1)y_1, \sum_{y_2\in\mathbb{Z}} \mu_2(y_2,x_2)y_2, \dots, \sum_{y_n\in\mathbb{Z}} \mu_n(y_n,x_n)y_n \right) \\
&= (x_1,\dots,x_n) = x.
\end{aligned}$$

Space discretization

Some remarks.

- **Many other possibilities** for a grid and for the associated interpolation function. In general, given $x \in \mathbb{R}^n$, the set

$$\{y \in \mathcal{G} \,|\, \mu(y, x) > 0\}$$

should be (ideally) of **small cardinality** and should contain points close to x.

- For the grid $\mathbb{Z}^n$ and the proposed interpolation function μ_n, the evaluation of

$$[v](x) = \sum_{y \in \mathbb{Z}^n} \mu_n(y, x) v(y)$$

requires 2^n operations.

Space discretization

For the grid

$$\mathcal{G}_{n,h} := h\mathbb{Z}^n,$$

one can simply define

$$\mu_{n,h}(y,x) = \mu_n(y/h, x/h).$$

We have, using the change of variable $y = hy'$,

$$\frac{x}{h} = \sum_{y' \in \mathbb{Z}^n} \mu_n(y', x/h) y' = \sum_{y \in \mathcal{G}_{n,h}} \underbrace{\mu_n(y/h, x/h)}_{=\mu_{n,h}(y,x)} \frac{y}{h}.$$

Multiplying by h, we get

$$x = \sum_{y \in \mathcal{G}_{n,h}} \mu_{n,h}(y,x) y.$$

Space discretization

Back to the DP-mapping. We replace the term $v(y_\tau[u,x](\tau))$ by the interpolation

$$[v](x+\tau f(u,x)) = \sum_{y\in\mathcal{G}} \mu(y, x+\tau f(u,x))v(y).$$

The **transition mapping** p is defined by $p(y|u,x) = \mu(y, x+\tau f(u,x))$. Note that

$$p(y|u,x) \in [0,1], \quad \sum_{y\in\mathcal{G}} p(y|u,x) = 1.$$

Thus $p(y|u,x)$ can be interpreted as a **probability transition** from x to y, under the control u.

Space discretization

For $v \in L^\infty(\mathcal{G})$, the discrete DP-mapping is defined by

$$\begin{aligned}\mathcal{T}_{\tau,h} v(x) &= \inf_{u\in U} \Big(\tau \ell(u,x) + (1-\lambda\tau)[v](x+\tau f(u,x)) \Big) \\ &= \inf_{u \in U} \Big(\tau\ell(u,x) + (1-\lambda\tau) \sum_{y\in\mathcal{G}} p(y|u,x) v(y) \Big).\end{aligned}$$

It is still well-defined and Lipschitz with modulus $(1-\lambda\tau)$, for the uniform norm.

Remarks.

- From now on: we only use $p(y|u,x)$, which contains both the interpolation map and the discretization of the ODE.
- The index $h > 0$ will be used to describe the **quality** of the space discretization.

Space-discretization

Further remarks.

- We still need to manipulate elements of $L^\infty(\mathcal{G})$, impossible since $\mathcal{G}$ is infinite. Further **domain restriction** to be applied, we do not discuss this aspect.
- The practical **computation of the infimum** in $\mathcal{T}_{\tau,h}$ may be difficult. Typically, $p(y|u,x)$ is non-differentiable. Extreme solution: **discretization** of U, minimization by enumeration.
- **Curse of dimensionality**.

$$\text{card}\big(B(0,R) \cap h\mathbb{Z}^n\big) = \mathcal{O}\Big(\big(\frac{R}{h}\big)^n\Big).$$

$\rightarrow$ **Exponential** complexity with respect to the dimension n.

Space discretization

Interpretation of the fixed point equation:

$$V_{\tau,h} = \mathcal{T}_{\tau,h} V_{\tau,h}, \quad V_{\tau,h} \in L^\infty(\mathcal{G}).$$

Notation: $L^\infty(\mathbb{N} \times \mathcal{G}; U)$ is the set of functions from $\mathbb{N} \times \mathcal{G}$ to U. Given $u \in L^\infty(\mathbb{N} \times \mathcal{G}; U)$, let $Y[u,x]$ denote the **Markov chain** defined by

$$\mathbb{P}\Big[Y[u,x](k+1) = y' \Big| Y[u,x](k) = y \Big] = p\big(y' | u(k,y), y\big)$$
$$Y[u,x](0) = x.$$

In words:

- At time k, if the Markov chain is equal to y, the control $u(k,y)$ is employed.
- The probability to move to y' is given by $p\big(y'|u(k,y),y\big)$.

Space-discretization

Cost function:

$$W_{\tau,h}(u,x) = \mathbb{E}\Big[\tau \sum_{k=0}^{\infty} (1-\lambda\tau)^k \ell\Big(u(k,Y(k)),Y(k)\Big)\Big],$$

where $Y = Y[u,x]$.

Lemma 6

The unique solution $V_{\tau,h}$ to the fixed-point equation

$$V_{\tau,h} = \mathcal{T}_{\tau,h} V_{\tau,h}$$

is the value function of the following problem:

$$V_{\tau,h}(x) = \inf_{u \in L^\infty(\mathbb{N}\times\mathcal{G};U)} W_{\tau,h}(u,x). \qquad (P_{\tau,h})$$

The analysis can be (again!) summarized with a commutative **diagram**:

$$
\begin{array}{ccc}
\text{Problem } P(x) & \xrightarrow{\text{Discretization}} & \text{Problem } P_{\tau,h}(x) \\
\Big| & & \Big| \\
\text{Dynamic Prog.} & & \text{Dynamic Prog.} \\
\Big\downarrow & & \Big\downarrow \\
V = \mathcal{T} V & \xrightarrow{\text{Discretization}} & V_{\tau,h} = \mathcal{T}_{\tau,h} V_{\tau,h}
\end{array}
$$

The "discretization" and "dynamic programming" phases **commute**.

Value iteration

Value iteration algorithm.

- Input: $v_0 : \mathcal{G} \to \mathbb{R}$.
- For $k = 0, 1, ..., K$, do

$$v_{k+1} = \mathcal{T}_{\tau,h} v_k.$$

- Output: v_K.

Lemma 7

The sequence $(v_k)_{k=0,1,...}$ converges linearly to $V_{\tau,h}$ for the supremum norm. More precisely:

$$\|v_k - V_{\tau,h}\|_{L^\infty(\mathcal{G})} \leq (1 - \lambda\tau)^k \|v_0 - V_{\tau,h}\|.$$

Proof: by induction. Recall that $\mathcal{T}_{\tau,h}$ is $(1 - \lambda\tau)$-Lipschitz.

Policy iteration

Definition 8

Let $L^\infty(\mathcal{G}, U)$ denote the set of mappings from $\mathcal{G}$ to U. We call any element $u \in L^\infty(\mathcal{G}, U)$ a **policy**.

Key idea. **Split** the fixed equation $v = \mathcal{T}_{\tau,h} v$ into a coupled system of equations:

$$\begin{cases} v(x) = \ \tau\ell(u(x), x) + (1-\lambda\tau) \sum_{y\in\mathcal{G}} p(y|u(x), x) v(x) & (i) \\ u(x) \in \ \underset{\alpha\in U}{\operatorname{argmin}}\ \tau\ell(\alpha, x) + (1-\lambda\tau) \sum_{y\in\mathcal{G}} p(y|\alpha, x) v(x) & (ii) \end{cases}$$

involving $v \in L^\infty(\mathcal{G})$ and $u \in L^\infty(\mathcal{G}, U)$.

Policy iteration

Remarks.

- For a given policy $u \in L^\infty(\mathcal{G}, U)$, equation (i) is **a linear fixed-point equation** with respect to v. It can be written in the abstract form
$$v = \mathcal{T}^u_{\tau,h} v,$$
where $\mathcal{T}_{\tau,h} \colon L^\infty(\mathcal{G}) \to L^\infty(\mathcal{G})$ is $(1 - \lambda\tau)$-Lipschitz-continuous for the supremum norm.

- For a given $v \in L^\infty(\mathcal{G})$, there exists a policy $u \in L^\infty(\mathcal{G}, U)$ satisfying (ii).

Policy iteration

Policy iteration method.

- Input: $u_0 \in L^\infty(\mathcal{G}, U)$.
- For $k = 0, 1, ...K$, do
  - Solve $v_{k+1} = \mathcal{T}_{\tau,h}^{u_k} v_{k+1}$.
  - Update the policy: find u_{k+1} such that for all $x \in \mathcal{G}$,

$$u_{k+1}(x) \in \underset{\alpha \in U}{\operatorname{argmin}} \Big(\tau \ell(\alpha, x) + (1 - \lambda\tau) \sum_{y \in \mathcal{G}} p(y|\alpha, x) v_{k+1}(x) \Big).$$

- Output: v_K and u_K.

Goal

Context. Let $V_{\tau,h}$ denote the solution to the fixed point equation

$$V_{\tau,h} = \mathcal{T}_{\tau,h} V_{\tau,h},$$

where

$$\mathcal{T}_{\tau,h} v(x) = \inf_{u \in U} \Big(\tau \ell(u,x) + (1 - \lambda\tau) \sum_{y \in \mathcal{G}} p(y|u,x) v(y) \Big).$$

A specific transition mapping $p : \mathcal{G} \times U \times \mathbb{R}^n \to [0,1]$ has been previously constructed, we consider now a general mapping.

Goal of the section: to **compare** $V_{\tau,h}$ with the value function of the original problem V.

Assumptions

Assumptions: there exists $C > 0$ such that $\forall x \in \mathbb{R}^n$, $\forall u \in U$,

$$\sum_{y \in \mathcal{G}} p(y|u,x) = 1, \tag{A1}$$

$$\left\| \sum_{y \in \mathcal{G}} p(y|u,x) y - (x + f(u,x)\tau) \right\| \leq C\tau^2 \tag{A2}$$

$$\sum_{y \in \mathcal{G}} p(y|u,x) \left\| y - (x + f(u,x)\tau) \right\|^2 \leq Ch^2. \tag{A3}$$

Interpretation:

- Assumption $(A2)$ says that

$$\sum_{y \in \mathcal{G}} p(y|u,x) y \approx x + f(u,x)\tau.$$

- Assumption $(A3)$ says that in this approximation formula, grid points close to $x + f(u,x)\tau$ should be employed...
- ...it is also a bound on the "randomness" of the Markov chain.

Main result

Theorem 9

Assume that V is Lipschitz continuous and that assumptions (A1)-(A3) hold true. Then, there exists a constant $C' > 0$, independent of $(\tau, h, \mathcal{G})$, depending on C, such that

$$|V_{\tau,h}(x) - V(x)| \leq C'\Big(\frac{h^2}{\tau^{3/2}} + \tau^{1/2}\Big).$$

Remarks.

- Lipschitz continuity is guaranteed if $\lambda > L_f$. Extensions of the theorem do exist when V is only Hölderian.
- Appropriate to choose $\tau = h$, bound: $2C'h^{1/2}$.
- In the proof, we make use of a constant C **whose value can be updated from line to line**. It is independent of τ, h, and ε (to appear later).

Proof

Proof. Step 1: decoupling of the variables. Our goal is to find an upper bound of

$$\delta := \sup_{x \in \mathcal{G}} \big(V_{\tau,h}(x) - V(x) \big)$$

and a lower bound of

$$\delta' := \inf_{x \in \mathcal{G}} \big(V_{\tau,h}(x) - V(x) \big).$$

In this proof, we will only explain how to bound (from above) δ.

Proof

The key idea is to start with:

$$\delta = \sup_{x \in \mathcal{G}} \left(V_{\tau,h}(x) - V(x)\right)$$

$$\leq \sup_{\substack{x \in \mathcal{G} \\ y \in \mathbb{R}^n}} \Psi_\varepsilon(x,y) := \left(V_{\tau,h}(x) - V(y) - \frac{\|x-y\|^2}{\varepsilon}\right),$$

where $\varepsilon \in (0,1]$ is arbitrary.

- Proof of the inequality: take $x = y$.
- Small deterioration since for $\varepsilon > 0$ very small, the optimal x and y are close to each other.

Proof

Simplifying assumption: there exists a pair $(x_0, y_0) \in \mathcal{G} \times \mathbb{R}^n$, depending on ε, which **maximizes** Ψ_ε.

[If this was not the case, an arbitrarily small modification of Ψ_ε could be done, so that the assumption holds true; we do not detail this aspect.]

We have:

$$\delta \leq V_{\tau,h}(x_0) - V(y_0) - \frac{\|y_0 - x_0\|^2}{\varepsilon} \leq V_{\tau,h}(x_0) - V(y_0).$$

We look for an **upper bound** of $V_{\tau,h}(x_0)$ and a **lower bound** of $V(y_0)$.

Proof

Step 2: estimate of $\|y_0 - x_0\|$. The inequality

$$\Psi_\varepsilon(x_0, x_0) \leq \Psi_\varepsilon(x_0, y_0),$$

yields

$$V_{\tau,h}(x_0) - V(x_0) - \frac{\|x_0 - x_0\|^2}{\varepsilon} \leq V_{\tau,h}(x_0) - V(y_0) - \frac{\|y_0 - x_0\|^2}{\varepsilon}.$$

Proof

Step 2: estimate of $\|y_0 - x_0\|$. The inequality

$$\Psi_\varepsilon(x_0, x_0) \leq \Psi_\varepsilon(x_0, y_0),$$

yields

$$-V(x_0) \leq -V(y_0) - \frac{\|y_0 - x_0\|^2}{\varepsilon}.$$

Re-arranging:

$$\|y_0 - x_0\|^2 \leq \varepsilon(V(x_0) - V(y_0)) \leq C\varepsilon\|y_0 - x_0\|,$$

since V is Lipschitz. Thus,

$$\|y_0 - x_0\| \leq C\varepsilon.$$

Proof

Step 3: lower bound of $V(y_0)$.
Let $\Phi\colon \mathbb{R}^n \to \mathbb{R}$ be defined by

$$\Phi(y) = -\frac{\|y - x_0\|^2}{\varepsilon}.$$

Since y_0 maximizes $\Psi_\varepsilon(x_0, \cdot)$, we have for any $y \in \mathbb{R}^n$:

$$\Psi_\varepsilon(x_0, y) \leq \Psi_\varepsilon(x_0, y_0)$$

Proof

Step 3: lower bound of $V(y_0)$.
Let $\Phi \colon \mathbb{R}^n \to \mathbb{R}$ be defined by

$$\Phi(y) = -\frac{\|y - x_0\|^2}{\varepsilon}.$$

Since y_0 maximizes $\Psi_\varepsilon(x_0, \cdot)$, we have for any $y \in \mathbb{R}^n$:

$$V_{\tau,h}(x_0) - V(y) - \frac{\|x_0 - y\|^2}{\varepsilon} \leq V_{\tau,h}(x_0) - V(y_0) - \frac{\|x_0 - y_0\|^2}{\varepsilon}$$

Proof

Step 3: lower bound of $V(y_0)$.
Let $\Phi\colon \mathbb{R}^n \to \mathbb{R}$ be defined by

$$\Phi(y) = -\frac{\|y - x_0\|^2}{\varepsilon}.$$

Since y_0 maximizes $\Psi_\varepsilon(x_0, \cdot)$, we have for any $y \in \mathbb{R}^n$:

$$-V(y) + \Phi(y) \leq -V(y_0) + \Phi(y_0)$$

Proof

Step 3: lower bound of $V(y_0)$.
Let $\Phi \colon \mathbb{R}^n \to \mathbb{R}$ be defined by

$$\Phi(y) = -\frac{\|y - x_0\|^2}{\varepsilon}.$$

Since y_0 maximizes $\Psi_\varepsilon(x_0, \cdot)$, we have for any $y \in \mathbb{R}^n$:

$$V(y) - \Phi(y) \geq V(y_0) - \Phi(y_0)$$

Thus $V - \Phi$ has a global minimizer in y_0.

Proof

Let us set

$$p_0 = \nabla\Phi(y_0) = \frac{2(x_0 - y_0)}{\varepsilon}.$$

Since V is a supersolution of the HJB equation, we have

$$\lambda V(y_0) - \mathcal{H}(y_0, p_0) \geq 0.$$

Denote by $u_0 \in U$ the control minimizing the pre-Hamiltonian in $H(\cdot, y_0, p_0)$, we have:

$$\lambda V(y_0) \geq \mathcal{H}(y_0, p_0) = \ell(u_0, y_0) + \langle p_0, f(u_0, y_0)\rangle. \tag{1}$$

Proof

Step 4: upper bound for $V_{\tau,h}(x_0)$. We use the dynamic programming principle. We have:

$$V_{\tau,h}(x_0) \leq \tau \ell(u_0, x_0) + (1 - \lambda\tau) \sum_{y \in \mathcal{G}} p(y|u_0, x_0) V_{\tau,h}(y). \qquad (2)$$

We next bound $V_{\tau,h}(y)$. We have: $\Psi_\varepsilon(y, y_0) \leq \Psi_\varepsilon(x_0, y_0)$, which yields

$$V_{\tau,h}(y) - V(y_0) - \frac{\|y - y_0\|^2}{\varepsilon} \leq V_{\tau,h}(x_0) - V(y_0) - \frac{\|x_0 - y_0\|^2}{\varepsilon}$$

Proof

Step 4: upper bound for $V_{\tau,h}(x_0)$. We use the dynamic programming principle. We have:

$$V_{\tau,h}(x_0) \leq \tau \ell(u_0, x_0) + (1 - \lambda\tau) \sum_{y \in \mathcal{G}} p(y|u_0, x_0) V_{\tau,h}(y). \tag{2}$$

We next bound $V_{\tau,h}(y)$. We have: $\Psi_\varepsilon(y, y_0) \leq \Psi_\varepsilon(x_0, y_0)$, which yields

$$V_{\tau,h}(y) \leq V_{\tau,h}(x_0) + \frac{\|y - y_0\|^2 - \|x_0 - y_0\|^2}{\varepsilon}. \tag{3}$$

We next re-arrange the term $\|y - y_0\|^2 - \|x_0 - y_0\|^2$.

Proof

We have:

$$
\begin{aligned}
\|y - y_0\|^2 - \|x_0 - y_0\|^2 &= 2\langle y - x_0, x_0 - y_0\rangle + \|y - x_0\|^2 \\
&= 2\langle y - (x_0 + f(u_0, x_0)\tau), x_0 - y_0\rangle \\
&\qquad + 2\langle f(u_0, x_0)\tau, x_0 - y_0\rangle \\
&\qquad + \|y - x_0\|^2. \qquad (4)
\end{aligned}
$$

Injecting (4) in (3) and then (3) in (2), we get:

$$
V_{\tau,h}(x_0) \leq \ell(u_0, x_0)\tau + (1 - \lambda\tau)(V_{\tau,h}(x_0) + a_1 + a_2 + a_3), \qquad (5)
$$

where the three terms a_1, a_2, and a_3 are defined and bounded right after.

Proof

Estimate of (a_1). We have

$$
\begin{aligned}
(a_1) &= \frac{2}{\varepsilon} \sum_{y \in \mathcal{G}} \Big(p(y|u_0, x_0) \Big\langle y - (x_0 + f(u_0, x_0)\tau), x_0 - y_0 \Big\rangle \Big) \\
&\leq \frac{2}{\varepsilon} \Big\langle \Big(\sum_{y \in \mathcal{G}} p(y|u_0, x_0) y \Big) - (x_0 + f(u_0, x_0)\tau), x_0 - y_0 \Big\rangle \\
&\leq \frac{2}{\varepsilon} \Big\| \Big(\sum_{y \in \mathcal{G}} p(y|u_0, x_0) y \Big) - (x_0 + f(u_0, x_0)\tau) \Big\| \cdot \|x_0 - y_0\| \\
&\leq \frac{2}{\varepsilon} (C\tau^2)(C\varepsilon) \\
&= C\tau^2,
\end{aligned}
$$

by Assumption (A2).

Proof

Estimate of (a_2). We have

$$
\begin{aligned}
(a_2) &= \frac{2}{\varepsilon} \sum_{y \in \mathcal{G}} p(y|u_0, x_0) \langle f(u_0, x_0)\tau, x_0 - y_0 \rangle \\
&= \frac{2}{\varepsilon} \langle f(u_0, x_0), x_0 - y_0 \rangle \tau \\
&= \langle f(u_0, x_0), p_0 \rangle \tau.
\end{aligned}
$$

Proof

Estimate of (a_3). We have

$$
\begin{aligned}
(a_3) &= \frac{1}{\varepsilon} \sum_{y \in \mathcal{G}} p(y|u_0, x_0) \|y - x_0\|^2 \\
&\le \frac{2}{\varepsilon} \sum_{y \in \mathcal{G}} p(y|u_0, x_0) \Big(\|y - (x_0 + f(u_0, x_0)\tau)\|^2 + \|f(u_0, y_0)\tau\|^2 \Big) \\
&\le C \frac{h^2 + \tau^2}{\varepsilon},
\end{aligned}
$$

by Assumption (A3).

Proof

Let us combine (5) with the three obtained bouds:

$$\begin{aligned} V_{\tau,h}(x_0) \leq \ell(u_0, x_0)\tau + (1-\lambda\tau) V_{\tau,h}(x_0) \\ + (1-\lambda\tau)\langle f(u_0, x_0), p_0 \rangle \tau \\ + (1-\lambda\tau) C\tau^2 \\ + (1-\lambda\tau) C\left(\frac{h^2+\tau^2}{\varepsilon}\right). \end{aligned}$$

Proof

Let us combine (5) with the three obtained bouds:

$$\begin{aligned} V_{\tau,h}(x_0) \leq \ell(u_0, x_0)\tau + (1 - \lambda\tau) V_{\tau,h}(x_0) & \\ + \langle f(u_0, x_0), p_0 \rangle \tau & \\ + C\tau^2 & \\ + C\left(\frac{h^2 + \tau^2}{\varepsilon} \right). & \end{aligned}$$

Re-arranging and dividing by τ:

$$\lambda V_{\tau,h}(x_0) \leq \ell(u_0, x_0) + \langle f(u_0, x_0), p_0 \rangle + C\left(\tau + \frac{h^2 + \tau^2}{\varepsilon\tau} \right). \quad (6)$$

Proof

Step 5. Conclusion.
Let recall the three main inequalities obtained so far:

$$\delta \leq V_{\tau,h}(x_0) - V(y_0),$$

$$\lambda V(y_0) \geq \ell(u_0, y_0) + \langle f(u_0, y_0), p_0 \rangle$$

$$\lambda V_{\tau,h}(x_0) \leq \ell(u_0, x_0) + \langle f(u_0, x_0), p_0 \rangle + C\left(\tau + \frac{h^2 + \tau^2}{\varepsilon \tau}\right).$$

Proof

We deduce that

$$
\begin{aligned}
\lambda V_\tau(x_0) - \lambda V(y_0) &\leq \ell(u_0, x_0) - \ell(u_0, y_0) + \langle f(u_0, x_0) - f(u_0, y_0), p_0 \rangle \\
&\quad + C\left(\tau + \frac{h^2 + \tau^2}{\varepsilon \tau}\right) \\
&\leq C\|x_0 - y_0\| + C\left(\tau + \frac{h^2 + \tau^2}{\varepsilon \tau}\right) \\
&\leq C\left(\varepsilon + \tau + \frac{h^2 + \tau^2}{\varepsilon \tau}\right).
\end{aligned}
$$

Choosing $\varepsilon = \tau^{1/2}$, we finally obtain

$$
\delta \leq V_\tau(x_0) - V(y_0) \leq \frac{C}{\lambda}\left(\tau^{1/2} + \tau + \frac{h^2 + \tau^2}{\tau^{3/2}}\right) \leq C\left(\tau^{1/2} + \frac{h^2}{\tau^{3/2}}\right).
$$

Variants

In this section: two techniques from the **machine-learning** community, in relation with optimal control.

- **Neural networks**
- **Q-learning.**

Reference

D. Bertsekas. Reinforcement learning and optimal control. Athena scientific, July 2019.

Neural networks

A general problem. Let $V : \mathbb{R}^n \to \mathbb{R}$.

- Consider a finite subset $\mathcal{G} = \{y_1, \dots, y_K\}$ of $\mathbb{R}^n$.
- Assume that $V_k := V(y_k)$ is known for all $k = 1, \dots, N$.

Knowing $V_1, \dots, V_k$, can we find a function $\bar{v}$ which "faithfully" **represents** V?

- This question is **not clearly formulated** at a mathematical level... but it arises in the numerical resolution of every problem that involve functions of one or several real numbers (PDEs, infinite-dimensional optimization, etc.)
- Interpolation is an answer.

Neural networks

A general approach. Fix a set $\mathcal{V}$ of "suitable" functions and chose $\bar{v}$ as a solution to the **least-square problem**:

$$\min_{v \in \mathcal{V}} \sum_{k=1}^{K} |v(y_k) - V_k|^2.$$

If $\mathcal{V}$ is convex, then the optimization problem is convex; one can hope to solve it globally.

A typical choice: $\mathcal{V}$ is a finite-dimensional vector space.

Neural networks

Parametric functions. Most of the time, $\mathcal{V}$ is given in a **parametric** form. Let R be a set of **parameters** and let $W\colon \mathbb{R}^n \times R \to \mathbb{R}$ be known explicitly. Then one can define:

$$\mathcal{V} = \{v \,|\, \exists r \in R,\ v(x) = W(x,r)\} = \{W(\cdot, r) \,|\, r \in R\}.$$

If R is convex and W affine with respect to r, then $\mathcal{V}$ is convex.

The least square problem is then equivalent to:

$$\min_{r \in R} \sum_{k=1}^{K} |W(y_k, r) - V_k|^2.$$

For a solution $\bar{r}$, define $\bar{v} = W(\cdot, \bar{r})$.

Neural networks

Example:

$$W(x, r) = \sum_{k=1}^{K} \mu(y_k, x) r_k,$$

where μ is an **interpolation** map.

The trivial solution to the least-square problem is $r_k = V_k$.

Neural network

A **neural network** is a specific **parametric** function, described by:

- Number of layers: l
- Number of hidden units: $d_1, \ldots, d_{l-1}$.
- Activation function $\varphi \colon \mathbb{R} \to \mathbb{R}$.

Many popular choices for φ. We define $d_0 = n$ and $d_l = 1$.

Notation.

- Given k, let $\varphi^k \colon \mathbb{R}^k \to \mathbb{R}^k$ be defined by

$$\varphi^k(x) = (\varphi(x_1), \varphi(x_2), \ldots, \varphi(x_k)).$$

- Given $\beta \in \mathbb{R}^k$ and $w \in \mathbb{R}^{k \times l}$, let $\phi_{\beta,w} \colon \mathbb{R}^l \to \mathbb{R}^k$ be defined by

$$\phi_{\beta,w}(x) = \varphi^k(\beta + wx).$$

Neural network

We consider the parametric function:

$$W(x,r) = \beta_l + w_l\Big(\phi_{\beta_{l-1},w_{l-1}} \circ \ldots \circ \phi_{\beta_2,w_2} \circ \phi_{\beta_1,w_1}(x)\Big),$$

where

$$r = (\beta_1, \beta_2, \ldots, \beta_l, w_1, w_2, \ldots, w_l) \in R,$$

$$\text{where: } R = \Big(\prod_{i=1}^{l} \mathbb{R}^{d_i}\Big) \times \Big(\prod_{i=1}^{l} \mathbb{R}^{d_i \times d_{i-1}}\Big).$$

Q-learning

Recall the (discrete) dynamic programing equation:

$$V(x) = \inf_{u \in U} \Big(\tau \ell(u,x) + \beta \sum_{y \in \mathcal{G}} p(y|u,x) V(y) \Big),$$

with $\beta = (1 - \lambda\tau) \in (0,1)$. We skip the indices τ and h.

A new decoupling, involving $V \colon \mathcal{G} \to \mathbb{R}$ and a **Q-function** $Q \colon U \times \mathcal{G} \to \mathbb{R}$:

$$\begin{cases} Q(u,x) = & \tau \ell(u,x) + \beta \sum_{y \in \mathcal{G}} p(y|u,x) V(y) & (i) \\ V(x) = & \inf_{u \in U} Q(u,x). & (ii) \end{cases}$$

As before, one can design a fixed point mechanism based on:

$$Q \xrightarrow{(ii)} V \xrightarrow{(i)} Q.$$

Q-learning

We focus on the equation (i) and assume now that U is finite. Let U, X, and Y be three random variables in $U \times \mathcal{G} \times \mathcal{G}$. We assume that for all (y, u, x),

$$\mathbb{P}\big[\mathsf{Y} = y \,|\, \mathsf{U} = u,\, \mathsf{X} = x\big] = p(y|u, x).$$

Let $\mu(u, x) = \mathbb{P}\big[(\mathsf{X}, \mathsf{U}) = (x, u)\big]$. Given $\phi\colon U \times \mathcal{G} \times \mathcal{G} \to \mathbb{R}$, we have

$$\mathbb{E}\big[\phi(\mathsf{U}, \mathsf{X})\big] = \sum_{(u,x) \in U \times \mathcal{G}} \phi(u, x)\xi(u, x).$$

Q-learning

For any function $\phi \colon \mathcal{G} \times U \times \mathcal{G} \to \mathbb{R}$, we have:

$$\mathbb{E}\left[\phi(\mathrm{X}, \mathrm{U}, \mathrm{Y})\right] = \sum_{(y,u,x) \in U \times \mathcal{G}} \phi(x, u, y) p(y|u, x) \xi(u, x).$$

Lemma 10

Let $(u, x) \in U \times \mathcal{G}$. Let $v \colon \mathcal{G} \to \mathbb{R}$. The unique solution to the following problem

$$\inf_{w \in \mathbb{R}} \sum_{y \in \mathcal{G}} p(y|u, x)(v(y) - w)^2$$

is given by

$$w = \sum_{y \in \mathcal{G}} p(y|u, x) v(y).$$

Q-learning

We can now reformulate equation (i).

$$
\begin{aligned}
Q(u,x) &= \tau\ell(u,x) + \beta \sum_{y\in\mathcal{G}} p(y|u,x)V(y) \\
&= \sum_{y\in\mathcal{G}} p(y|u,x)\big(\tau\ell(u,x) + \beta V(y)\big) \\
&= \operatorname*{argmin}_{q\in\mathbb{R}} \sum_{y\in\mathcal{G}} p(y|u,x)\Big(\tau\ell(u,x) + \beta V(y) - q\Big)^2.
\end{aligned}
$$

Let $\mathcal{Q}$ denote the set of "suitable" Q-functions. For solving (i), we can consider the optimization problem:

$$
\inf_{Q\in\mathcal{Q}} \sum_{(u,x)\in U\times\mathcal{G}} \sum_{y\in\mathcal{G}} \big(\tau\ell(u,x) + \beta V(y) - Q(u,x)\big)^2 p(y|u,x)\mu(u,x).
$$

Q-learning

Equivalently:

$$\inf_{Q \in \mathcal{Q}} \mathbb{E}\Big[\Big(\tau\ell(U,X) + \beta V(Y) - Q(U,X)\Big)^2\Big].$$

The problem can be **sampled.** Consider a "black box" which can **simulate** K outcomes of the random variable $(\mathrm{Y},\mathrm{U},\mathrm{X})$, denoted $(y_k, u_k, x_k)_{k=1,\ldots,K}$, as well as $\ell_k = \ell(u_k, x_k)$.

An approximation of the problem is:

$$\inf_{Q \in \mathcal{Q}} \sum_{k=1}^{K} \Big[\Big(\tau\ell_k + \beta V(y_k) - Q(u_k, x_k)\Big)^2\Big].$$

Q-learning

Last remarks!

- This is a **model-free approach**: the knowledge of ℓ and p is transfered to the black box.
- In the iterative algorithm, V only needs to be evaluated at the points y_k.
- Recent application: various board games, video games, automotive driving, etc.