Hamilton-Jacobi via neural networks

Master's thesis defense

Averil PROST

directed by

Olivier BOKANOWSKI

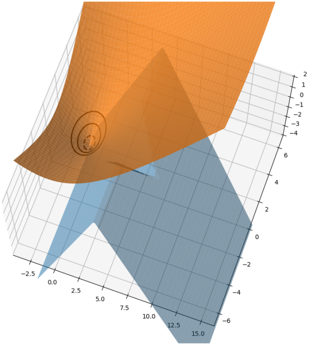
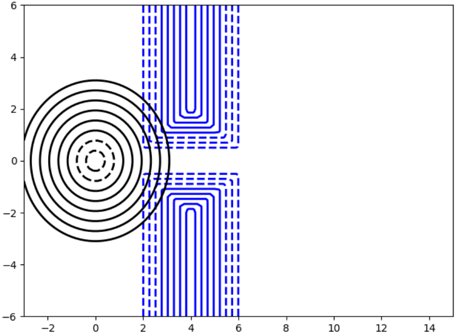
INSTITUT NATIONAL DES SCIENCES APPLIQUÉES ROUEN NORMANDIE

Table of Contents

Deterministic control problems

Level set formulation

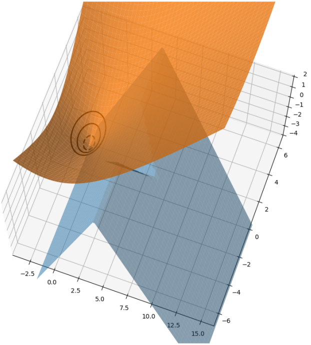
- target region : $\varphi \leqslant 0$

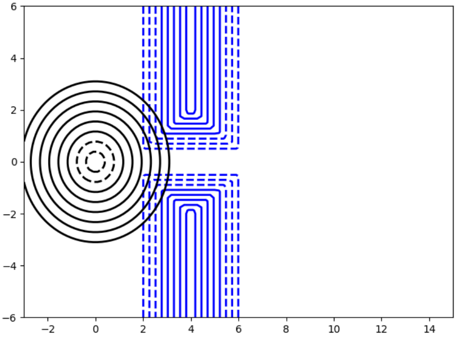



Deterministic control problems

Level set formulation

- target region : $\varphi \leqslant 0$
- state constraints (obstacle) : $g \leqslant 0$

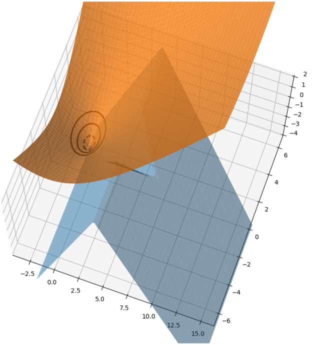


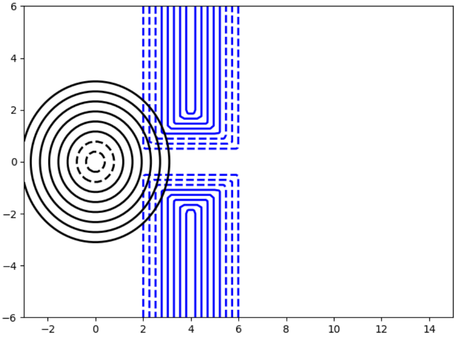


Deterministic control problems

Level set formulation

- target region : $\varphi \leqslant 0$
- state constraints (obstacle) : $g \leqslant 0$
- backward reachable sets t : $u(t,\cdot) \leqslant 0$

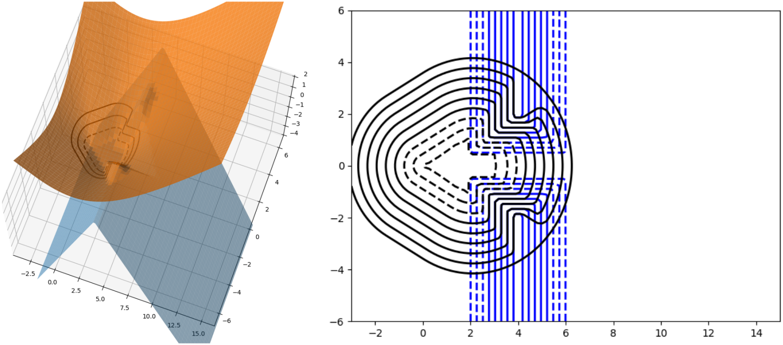




Deterministic control problems

Level set formulation

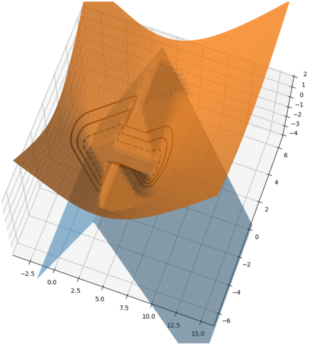
- target region : $\varphi \leqslant 0$
- state constraints (obstacle) : $g \leqslant 0$
- backward reachable sets t : $u(t, \cdot) \leqslant 0$

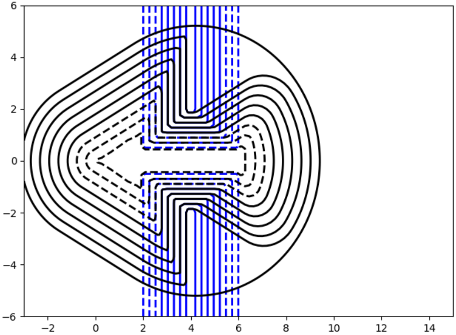


Deterministic control problems

Level set formulation

- target region : $\varphi \leqslant 0$
- state constraints (obstacle) : $g \leqslant 0$
- backward reachable sets t : $u(t, \cdot) \leqslant 0$

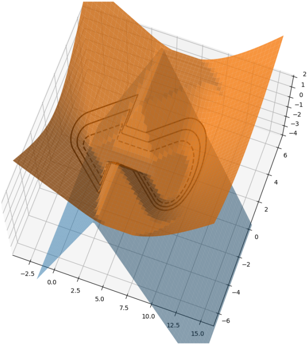


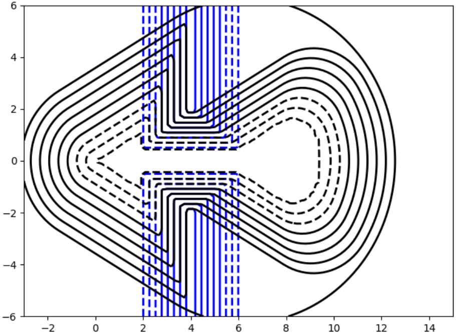


Deterministic control problems

Level set formulation

- target region : $\varphi \leqslant 0$
- state constraints (obstacle) : $g \leqslant 0$
- backward reachable sets t : $u(t, \cdot) \leqslant 0$

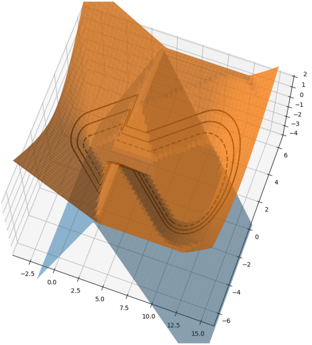


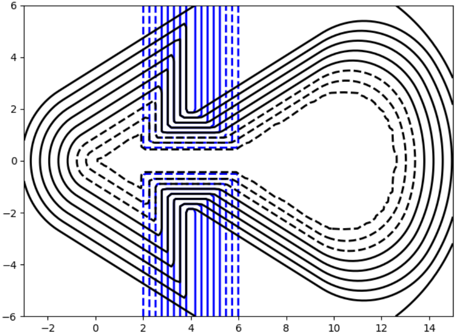


Deterministic control problems

Level set formulation

- target region : $\varphi \leqslant 0$
- state constraints (obstacle) : $g \leqslant 0$
- backward reachable sets t : $u(t, \cdot) \leqslant 0$





Deterministic control problems

Level set formulation

- target region : $\varphi \leqslant 0$
- state constraints (obstacle) : $g \leqslant 0$
- backward reachable sets t : $u(t, \cdot) \leqslant 0$

$$\min\left(\partial_t u + H\left(\nabla u\right), u - g\right) = 0, \quad u(T, \cdot) = \varphi \vee g.$$

Deterministic control problems

Level set formulation

- target region : $\varphi \leqslant 0$
- state constraints (obstacle) : $g \leqslant 0$
- backward reachable sets t : $u(t, \cdot) \leqslant 0$

$$\min\left(\partial_t u + H\left(\nabla u\right), u - g\right) = 0, \quad u(T, \cdot) = \varphi \vee g.$$

Numerical methods :

- Mesh-based methods (FD, FV, FE, DG...)
  Curse of dimensionality.

Deterministic control problems

Level set formulation

- target region : $\varphi \leqslant 0$
- state constraints (obstacle) : $g \leqslant 0$
- backward reachable sets t : $u(t, \cdot) \leqslant 0$

$$\min\left(\partial_t u + H\left(\nabla u\right), u - g\right) = 0, \quad u(T, \cdot) = \varphi \vee g.$$

Numerical methods :

- Mesh-based methods (FD, FV, FE, DG...)
  Curse of dimensionality.
- Reinforcement/learning methods.

Case of study

Obstacle problem ([ABZ13]) Find $u = u(t,x)$ s.t.

$$\begin{cases} \min\left(\partial_t u + \max_{a \in A} \nabla u \cdot f(x,a), u - g(x)\right) = 0, \\ u(T,x) = \varphi(x) \vee g(x). \end{cases}$$

Case of study

Obstacle problem ([ABZ13]) Find $u = u(t,x)$ s.t.

$$\begin{cases} \min\left(\partial_t u + \max_{a\in A} \nabla u \cdot f(x,a), u - g(x)\right) = 0, \\ u(T,x) = \varphi(x) \vee g(x). \end{cases}$$

Noticeable examples :

- $f(x,a) = b \in \mathbb{R}^n$ constant. Advection with obstacle.

Case of study

Obstacle problem ([ABZ13]) Find $u = u(t,x)$ s.t.

$$\begin{cases} \min\left(\partial_t u + \max_{a\in A} \nabla u \cdot f(x,a), u - g(x)\right) = 0, \\ u(T,x) = \varphi(x) \vee g(x). \end{cases}$$

Noticeable examples :

- $f(x,a) = b \in \mathbb{R}^n$ constant. Advection with obstacle.
- $f(x,a) = b(x)$. Let $\dot{y}_x^a(\theta) = f(y_x^a(\theta), a(\theta))$, $y_x^a(0) := x$. Then,

$$u(t,x) = \varphi(y_x(T-t)) \bigvee \max_{\theta\in[t,T]} g(y_x(\theta)).$$

Case of study

Obstacle problem ([ABZ13]) Find $u = u(t,x)$ s.t.

$$\begin{cases} \min\left(\partial_t u + \max_{a\in A} \nabla u \cdot f(x,a), u - g(x)\right) = 0, \\ u(T,x) = \varphi(x) \vee g(x). \end{cases}$$

Noticeable examples :

- $f(x,a) = b \in \mathbb{R}^n$ constant. Advection with obstacle.
- $f(x,a) = b(x)$. Let $\dot{y}_x^a(\theta) = f(y_x^a(\theta), a(\theta))$, $y_x^a(0) := x$. Then,

$$u(t,x) = \varphi(y_x(T-t)) \bigvee \max_{\theta\in[t,T]} g(y_x(\theta)).$$

- $f(x,a) = a$, $g \equiv -\infty$: $u(t,x) = \min_{b\in\mathscr{B}(0,t)} \varphi(x - tb)$.

Table of Contents

DPP

We consider the integral version of our model :

Dynamical programming principle $\forall h \in [0, T-t]$,

$$u(t,x) = \inf_{a(\cdot)\in\mathbb{A}_{[t,t+h]}} u(t+h, y_x^a(h)) \bigvee \max_{\theta\in[0,h]} g(y_x^a(\theta))$$

DPP

We consider the integral version of our model :

Dynamical programming principle $\forall h \in [0, T-t]$,

$$u(t,x) = \inf_{a(\cdot)\in\mathbb{A}_{[t,t+h]}} u(t+h, y_x^a(h)) \bigvee \max_{\theta\in[0,h]} g(y_x^a(\theta))$$

First, let us introduce *feedback* controls :

$\alpha(\theta, x) := a(\theta)$, $\quad a$ optimal for the problem issued from x in $t = \theta$.

DPP

We consider the integral version of our model :

Dynamical programming principle $\forall h \in [0, T-t]$,

$$u(t,x) = \inf_{a(\cdot)\in\mathbb{A}_{[t,t+h]}} u(t+h, y_x^a(h)) \bigvee \max_{\theta\in[0,h]} g(y_x^a(\theta))$$

First, let us introduce *feedback* controls :

$\alpha(\theta, x) := a(\theta)$, $\quad a$ optimal for the problem issued from x in $t = \theta$.

- May lack regularity (measurable and bounded if A compact).

DPP

We consider the integral version of our model :

Dynamical programming principle $\forall h \in [0, T-t]$,

$$u(t,x) = \inf_{a(\cdot)\in\mathbb{A}_{[t,t+h]}} u(t+h, y_x^a(h)) \bigvee \max_{\theta\in[0,h]} g(y_x^a(\theta))$$

First, let us introduce *feedback* controls :

$\alpha(\theta, x) := a(\theta)$, $\quad a$ optimal for the problem issued from x in $t = \theta$.

- May lack regularity (measurable and bounded if A compact).
- May not be defined everywhere (no uniqueness).

DPP

We consider the integral version of our model :

Dynamical programming principle $\forall h \in [0, T-t]$,

$$u(t,x) = \inf_{a(\cdot)\in\mathbb{A}_{[t,t+h]}} u(t+h, y_x^a(h)) \bigvee \max_{\theta\in[0,h]} g(y_x^a(\theta))$$

First, let us introduce *feedback* controls :

$\alpha(\theta, x) := a(\theta)$, $\quad a$ optimal for the problem issued from x in $t = \theta$.

- May lack regularity (measurable and bounded if A compact).
- May not be defined everywhere (no uniqueness).
- If $\alpha(\cdot,\cdot) \in \mathcal{A}_{[t,t+h]} := L^\infty([0,T], A)$, equivalent formulation.

Functional formulation

Let μ a positive measure such that $\mu(\Omega) < \infty$, with Ω the space domain. New equivalent formulation :

$$\begin{cases} J(t,x,\alpha) := u(t+h, y_x^a(h)) \bigvee \max_{\theta \in [0,h]} g(y_x^a(\theta)) \\ \alpha^* \in \operatorname{argmin}_{\alpha \in \mathcal{A}_{[t,t+h]}} \int_{x \in \Omega} J(t,x,\alpha)\mu(dx), \\ u(t,x) := J(t,x,\alpha^*). \end{cases}$$

Functional formulation

Let μ a positive measure such that $\mu(\Omega) < \infty$, with Ω the space domain. New equivalent formulation :

$$\begin{cases} J(t,x,\alpha) := u(t+h, y_x^a(h)) \bigvee \max_{\theta\in[0,h]} g(y_x^a(\theta)) \\ \alpha^* \in \operatorname{argmin}_{\alpha\in\mathcal{A}_{[t,t+h]}} \int_{x\in\Omega} J(t,x,\alpha)\mu(dx), \\ u(t,x) := J(t,x,\alpha^*). \end{cases}$$

Two-step discretization :

- Fonction space : finite-dimensional space $\hat{A} \subset \mathcal{A}$ for the controls.

Functional formulation

Let μ a positive measure such that $\mu(\Omega) < \infty$, with Ω the space domain. New equivalent formulation :

$$\begin{cases} J(t,x,\alpha) := u(t+h, y_x^a(h)) \bigvee \max_{\theta\in[0,h]} g(y_x^a(\theta)) \\ \alpha^* \in \operatorname{argmin}_{\alpha\in\mathcal{A}_{[t,t+h]}} \int_{x\in\Omega} J(t,x,\alpha)\mu(dx), \\ u(t,x) := J(t,x,\alpha^*). \end{cases}$$

Two-step discretization :

- Fonction space : finite-dimensional space $\hat{\mathcal{A}} \subset \mathcal{A}$ for the controls.
- Time : let $h = \Delta t$, computation of $u^n(\cdot) \simeq u(n\Delta t, \cdot)$ with u^{n+1} by induction.

Lagrangian scheme

Let $\hat{A}$ a finite-dimensional space (FE, spectral basis, neural networks...), $N \coloneqq T/\Delta t$, and $M \in \mathbb{N}^*$. Let $\hat{y}_x^\alpha(\Delta t) \simeq y_x^\alpha(\Delta t)$.

Lagrangian scheme

Let $\hat{\mathcal{A}}$ a finite-dimensional space (FE, spectral basis, neural networks...), $N \coloneqq T/\Delta t$, and $M \in \mathbb{N}^*$. Let $\hat{y}_x^\alpha(\Delta t) \simeq y_x^\alpha(\Delta t)$.

Choose $\hat{u}^N \coloneqq \varphi \vee g$.
for $n \in [\![N-1, 0]\!]$ **do**
    Draw $(X_k)_k$ M iid samples of $X \sim \mu$.
    Pick $\hat{\alpha}^n \in \operatorname{argmin}_{\hat{\alpha} \in \hat{\mathcal{A}}} \sum_{i=1}^M \hat{u}^{n+1}(\hat{y}_{X_i}^{\hat{\alpha}}(\Delta t)) \vee g(X_i)$.
    Define $\hat{u}^n(x) \coloneqq \hat{u}^{n+1}(\hat{y}_x^{\hat{\alpha}^n}(\Delta t)) \vee g(x)$.

Lagrangian scheme

Let $\hat{\mathcal{A}}$ a finite-dimensional space (FE, spectral basis, neural networks...), $N := T/\Delta t$, and $M \in \mathbb{N}^*$. Let $\hat{y}_x^\alpha(\Delta t) \simeq y_x^\alpha(\Delta t)$.

---

1 Choose $\hat{u}^N := \varphi \vee g$.
2 **for** $n \in [\![N-1, 0]\!]$ **do**
3 Draw $(X_k)_k$ M iid samples of $X \sim \mu$.
4 Pick $\hat{\alpha}^n \in \operatorname{argmin}_{\hat{\alpha} \in \hat{\mathcal{A}}} \sum_{i=1}^M \hat{u}^{n+1}(\hat{y}_{X_i}^{\hat{\alpha}}(\Delta t)) \vee g(X_i)$.
5 Define $\hat{u}^n(x) := \hat{u}^{n+1}(\hat{y}_x^{\hat{\alpha}^n}(\Delta t)) \vee g(x)$.

- The variable u^n is computed with$(\hat{\alpha}^k)_{k \geqslant n}$.

Lagrangian scheme

Let $\hat{\mathcal{A}}$ a finite-dimensional space (FE, spectral basis, neural networks...), $N \coloneqq T/\Delta t$, and $M \in \mathbb{N}^*$. Let $\hat{y}_x^\alpha(\Delta t) \simeq y_x^\alpha(\Delta t)$.

Choose $\hat{u}^N \coloneqq \varphi \vee g$.
for $n \in [\![N-1, 0]\!]$ **do**
    Draw $(X_k)_k$ M iid samples of $X \sim \mu$.
    Pick $\hat{\alpha}^n \in \operatorname{argmin}_{\hat{\alpha} \in \hat{\mathcal{A}}} \sum_{i=1}^M \hat{u}^{n+1}(\hat{y}_{X_i}^{\hat{\alpha}}(\Delta t)) \vee g(X_i)$.
    Define $\hat{u}^n(x) \coloneqq \hat{u}^{n+1}(\hat{y}_x^{\hat{\alpha}^n}(\Delta t)) \vee g(x)$.

- The variable u^n is computed with$(\hat{\alpha}^k)_{k \geqslant n}$.
- Refinement : consider $\max_{\theta \in \Theta} g(\hat{y}_{X_i}^{\hat{\alpha}}(\theta))$, $\Theta \subset [0, \Delta t]$.

Lagrangian scheme

Let $\hat{\mathcal{A}}$ a finite-dimensional space (FE, spectral basis, neural networks...), $N := T/\Delta t$, and $M \in \mathbb{N}^*$. Let $\hat{y}_x^{\alpha}(\Delta t) \simeq y_x^{\alpha}(\Delta t)$.

Choose $\hat{u}^N := \varphi \vee g$.
for $n \in [\![N-1, 0]\!]$ **do**
  Draw $(X_k)_k$ M iid samples of $X \sim \mu$.
  Pick $\hat{\alpha}^n \in \operatorname{argmin}_{\hat{\alpha} \in \hat{\mathcal{A}}} \sum_{i=1}^{M} \hat{u}^{n+1}(\hat{y}_{X_i}^{\hat{\alpha}}(\Delta t)) \vee g(X_i)$.
  Define $\hat{u}^n(x) := \hat{u}^{n+1}(\hat{y}_x^{\hat{\alpha}^n}(\Delta t)) \vee g(x)$.

- The variable u^n is computed with$(\hat{\alpha}^k)_{k \geqslant n}$.
- Refinement : consider $\max_{\theta \in \Theta} g(\hat{y}_{X_i}^{\hat{\alpha}}(\theta))$, $\Theta \subset [0, \Delta t]$.
- Quadratic complexity.

Semi-Lagrangian scheme

Additional variable $\hat{U}^n$ for the minimization step. Let $\hat{\mathcal{U}}$ be a finite-dimensional space.

Semi-Lagrangian scheme

Additional variable $\hat{U}^n$ for the minimization step. Let $\hat{\mathcal{U}}$ be a finite-dimensional space.

Let $\hat{U}^N := \mathbb{P}_{\hat{\mathcal{U}}}(\varphi \vee g)$.
for $n \in [\![N-1, 0]\!]$ **do**
  Draw $(X_k)_k$ M iid samples of $X \sim \mu$.
  Pick $\hat{\alpha}^n \in \operatorname{argmin}_{\hat{\alpha} \in \hat{\mathcal{A}}} \sum_{i=1}^{M} \hat{U}^{n+1}(\hat{y}_{X_i}^{\hat{\alpha}}(\Delta t)) \vee g(X_i)$.
  Define $\hat{U}^n(x) := \mathbb{P}_{\hat{\mathcal{U}}} \left\{ \hat{U}^{n+1}(\hat{y}_x^{\hat{\alpha}^n}(\Delta t)) \vee g(x) \right\}$.

Semi-Lagrangian scheme

Additional variable $\hat{U}^n$ for the minimization step. Let $\hat{\mathcal{U}}$ be a finite-dimensional space.

Let $\hat{U}^N := \mathbb{P}_{\hat{\mathcal{U}}}(\varphi \vee g)$.

for $n \in [\![N-1, 0]\!]$ **do**

- Draw $(X_k)_k$ M iid samples of $X \sim \mu$.
- Pick $\hat{\alpha}^n \in \operatorname{argmin}_{\hat{\alpha} \in \hat{\mathcal{A}}} \sum_{i=1}^{M} \hat{U}^{n+1}(\hat{y}^{\hat{\alpha}}_{X_i}(\Delta t)) \vee g(X_i)$.
- Define $\hat{U}^n(x) := \mathbb{P}_{\hat{\mathcal{U}}} \left\{ \hat{U}^{n+1}(\hat{y}^{\hat{\alpha}^n}_x(\Delta t)) \vee g(x) \right\}$.

- Linear complexity (with double computational/memory cost).

Semi-Lagrangian scheme

Additional variable $\hat{U}^n$ for the minimization step. Let $\hat{\mathcal{U}}$ be a finite-dimensional space.

Let $\hat{U}^N := \mathbb{P}_{\hat{\mathcal{U}}}(\varphi \vee g)$.
for $n \in [\![N-1, 0]\!]$ **do**
    Draw $(X_k)_k$ M iid samples of $X \sim \mu$.
    Pick $\hat{\alpha}^n \in \operatorname{argmin}_{\hat{\alpha} \in \hat{\mathcal{A}}} \sum_{i=1}^{M} \hat{U}^{n+1}(\hat{y}^{\hat{\alpha}}_{X_i}(\Delta t)) \vee g(X_i)$.
    Define $\hat{U}^n(x) := \mathbb{P}_{\hat{\mathcal{U}}} \left\{ \hat{U}^{n+1}(\hat{y}^{\hat{\alpha}^n}_x(\Delta t)) \vee g(x) \right\}$.

- Linear complexity (with double computational/memory cost).
- Projection step induces diffusion.

Semi-Lagrangian scheme

Additional variable $\hat{U}^n$ for the minimization step. Let $\hat{\mathcal{U}}$ be a finite-dimensional space.

Let $\hat{U}^N := \mathbb{P}_{\hat{\mathcal{U}}}(\varphi \vee g)$.

for $n \in [\![N-1, 0]\!]$ **do**

  Draw $(X_k)_k$ M iid samples of $X \sim \mu$.

  Pick $\hat{\alpha}^n \in \operatorname{argmin}_{\hat{\alpha} \in \hat{\mathcal{A}}} \sum_{i=1}^{M} \hat{U}^{n+1}(\hat{y}_{X_i}^{\hat{\alpha}}(\Delta t)) \vee g(X_i)$.

  Define $\hat{U}^n(x) := \mathbb{P}_{\hat{\mathcal{U}}} \left\{ \hat{U}^{n+1}(\hat{y}_x^{\hat{\alpha}^n}(\Delta t)) \vee g(x) \right\}$.

- Linear complexity (with double computational/memory cost).
- Projection step induces diffusion.
- Hybrid algorithm if the evaluation step is fully Lagrangian.

Table of Contents

Definition

$$\mathscr{R}(x) = \sigma_1 \circ L_1 \circ \cdots \circ \sigma_p \circ L_p, \quad L_i \text{ affine, } \sigma_i \text{ nonlinear.}$$

- Historically, $\sigma \simeq \frac{1}{1+\exp(-x)}$ the logistic function : classifiers.

Definition

$$\mathscr{R}(x) = \sigma_1 \circ L_1 \circ \cdots \circ \sigma_p \circ L_p, \quad L_i \text{ affine, } \sigma_i \text{ nonlinear.}$$

- Historically, $\sigma \simeq \frac{1}{1+\exp(-x)}$ the logistic function : classifiers.
- Activation $\sigma(x) = \max(0, x)$ (ReLu) often used for function approximation.

Definition

$$\mathscr{R}(x) = \sigma_1 \circ L_1 \circ \cdots \circ \sigma_p \circ L_p, \quad L_i \text{ affine, } \sigma_i \text{ nonlinear.}$$

- Historically, $\sigma \simeq \frac{1}{1+\exp(-x)}$ the logistic function : classifiers.
- Activation $\sigma(x) = \max(0, x)$ (ReLu) often used for function approximation.
- The regularity of $\mathbb{R}^m \mapsto \mathbb{R}^p$ stems from that of σ (continuity/Lipschitz-continuity with GroupSort).

Density

Let $\Omega \subset \mathbb{R}$ be compact.

Squasher $\lim\limits_{x\to-\infty} \sigma(x) = 0$, σ nondecreasing and $\lim\limits_{x\to\infty} \sigma(x) = 1$.

Density

Let $\Omega \subset \mathbb{R}$ be compact.

Squasher $\lim_{x \to -\infty} \sigma(x) = 0$, σ nondecreasing and $\lim_{x \to \infty} \sigma(x) = 1$.

Théorème – Universal approximatin theorem (Lemma 16.1 of [GKKW02]) The space of neural network is dense in $(\mathcal{C}(\Omega, \mathbb{R}^p), |\cdot|_\infty)$.

Density

Let $\Omega \subset \mathbb{R}$ be compact.

Squasher $\lim_{x \to -\infty} \sigma(x) = 0$, σ nondecreasing and $\lim_{x \to \infty} \sigma(x) = 1$.

Théorème – Universal approximatin theorem (Lemma 16.1 of [GKKW02]) The space of neural network is dense in $(\mathcal{C}(\Omega, \mathbb{R}^p), |\cdot|_\infty)$.

- Estimates in $|\cdot|_\infty$ norm for dim. 1, in $|\cdot|_{L^2}$ norm otherwise (see Table 1. of [TSB20]).

Density

Let $\Omega \subset \mathbb{R}$ be compact.

Squasher $\lim_{x \to -\infty} \sigma(x) = 0$, σ nondecreasing and $\lim_{x \to \infty} \sigma(x) = 1$.

Théorème – Universal approximatin theorem (Lemma 16.1 of [GKKW02]) The space of neural network is dense in $(\mathcal{C}(\Omega, \mathbb{R}^p), |\cdot|_\infty)$.

- Estimates in $|\cdot|_\infty$ norm for dim. 1, in $|\cdot|_{L^2}$ norm otherwise (see Table 1. of [TSB20]).
- Gap between the estimates and the numerical efficiency of networks.

Table of Contents

Assumptions

Let Ω bounded. We assume some regularity of the data :

Assumptions

Let Ω bounded. We assume some regularity of the data :

- Controls are valued in a convex compact $A \subset \mathbb{R}^p$.

Assumptions

Let Ω bounded. We assume some regularity of the data :

- Controls are valued in a convex compact $A \subset \mathbb{R}^p$.
- The functions f, g and φ are Lipschitz-continuous.

Assumptions

Let Ω bounded. We assume some regularity of the data :

- Controls are valued in a convex compact $A \subset \mathbb{R}^p$.
- The functions f, g and φ are Lipschitz-continuous.
- The training densities μ_n are bounded by $0 < \nu \leqslant \mu_n \leqslant C$.

Assumptions

Let Ω bounded. We assume some regularity of the data :

- Controls are valued in a convex compact $A \subset \mathbb{R}^p$.
- The functions f, g and φ are Lipschitz-continuous.
- The training densities μ_n are bounded by $0 < \nu \leqslant \mu_n \leqslant C$.

Moreover, we assume that the support of the densities "follows" the dynamics, i.e. $\forall \omega \subset \Omega$ open, $y_\omega^{a^*}(T - t_n) \in \{\varphi \leqslant 0\}$ implies $\mu_n(\omega) > 0$.

Result

Let $N \in \mathbb{N}$ be fixed. Let Θ the dimension of the control space.

Proposition Let $(X_n)_n$ a random variable sequence following the laws μ_n.

$$\lim_{\Theta \to \infty} \inf_{\hat{a} \in \hat{\mathcal{A}}_\Theta} \mathbb{E}[|\hat{V}_n(X_n) - V_n(X_n)|] = 0$$

Idea of the proof

Lack of smoothness on the control : let a_n^ε be the regularization of a_n^* by convolution. We decompose $\hat{V}_n - V_n$ in

$$0 \leqslant \mathbb{E}(\hat{V}_n - V_n) \leqslant C_1 \inf_{\hat{a} \in \hat{\mathcal{A}}} \mathbb{E}\,|a_n^\varepsilon - a_n^*| \quad \text{regularization error}$$

$$+ \dots$$

Idea of the proof

Lack of smoothness on the control : let a_n^ε be the regularization of a_n^* by convolution. We decompose $\hat{V}_n - V_n$ in

$$
\begin{aligned}
0 \leqslant \mathbb{E}(\hat{V}_n - V_n) \leqslant\ & C_1 \inf_{\hat{a} \in \hat{\mathcal{A}}} \mathbb{E}\,|a_n^\varepsilon - a_n^*| \quad \text{regularization error} \\
& + C_2 \inf_{\hat{a} \in \hat{\mathcal{A}}} \mathbb{E}\,|\hat{a} - a_n^\varepsilon| \quad \text{approximation of } \mathcal{A} \text{ by } \hat{\mathcal{A}} \\
& + \dots
\end{aligned}
$$

Idea of the proof

Lack of smoothness on the control : let a_n^ε be the regularization of a_n^* by convolution. We decompose $\hat{V}_n - V_n$ in

$$\begin{aligned} 0 \leqslant \mathbb{E}(\hat{V}_n - V_n) \leqslant C_1 \inf_{\hat{a}\in\hat{\mathcal{A}}} \mathbb{E}\,|a_n^\varepsilon - a_n^*| & \quad \text{regularization error} \\ + C_2 \inf_{\hat{a}\in\hat{\mathcal{A}}} \mathbb{E}\,|\hat{a} - a_n^\varepsilon| & \quad \text{approximation of } \mathcal{A} \text{ by } \hat{\mathcal{A}} \\ + C_3\,\mathbb{E}\left(\hat{V}_{n+1} - V_{n+1}\right) & \quad \text{induction term} \end{aligned}$$

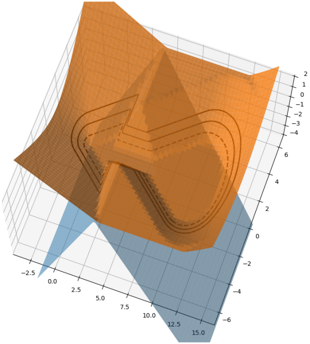
Letting $\Theta \to \infty$, we may control the growth of C_3 when $\varepsilon \searrow 0$. Hence convergence for fixed N.

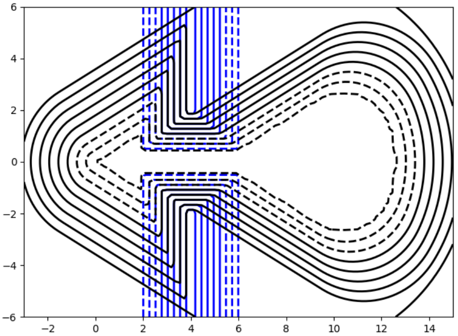
Table of Contents

Advection-eikonal equation with obstacle term

Let $A := \mathscr{B}_{\mathbb{R}^n}(0,1)$, $b \in \mathbb{R}^n$, $c \geqslant 0$. Find $u = u(t,x)$ s. t.

$$\min(-\partial_t u + \max_{a \in A} \nabla u \cdot [b + ca], u - g) = 0$$





One simulation example

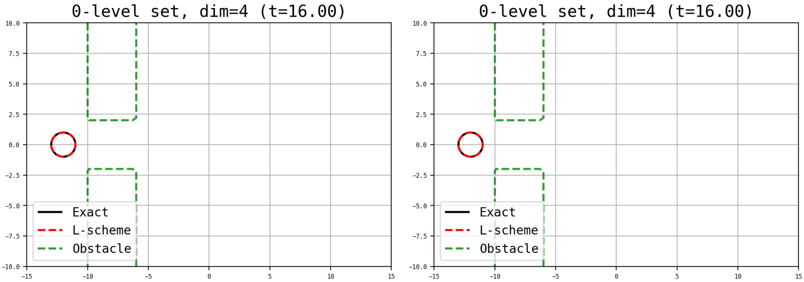


Figure – Left : $N_{\text{it}} = 8$, right : $N_{\text{it}} = 16$, Lagrangian scheme.

Parameters : $b = e_1$, $c = 1/2$, 3 layers of 60 neurons, ReLu activation, 10^5 iterations of S.G. with 4000 points.

One simulation example

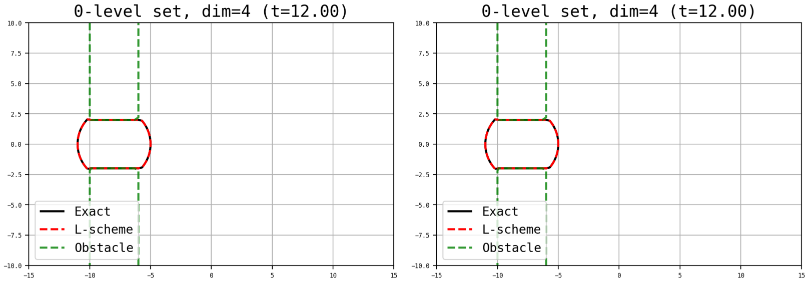


Figure – Left : $N_{\text{it}} = 8$, right : $N_{\text{it}} = 16$, Lagrangian scheme.

Parameters : $b = e_1$, $c = 1/2$, *3 layers of 60 neurons, ReLu activation,* 10^5 *iterations of S.G. with 4000 points.*

One simulation example

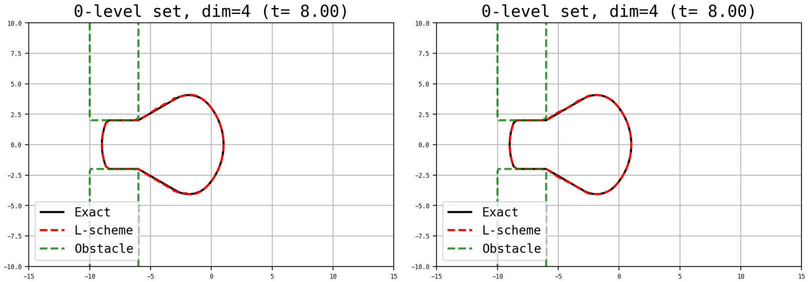


Figure – Left : $N_{\text{it}} = 8$, right : $N_{\text{it}} = 16$, Lagrangian scheme.

Parameters : $b = e_1$, $c = 1/2$, *3 layers of 60 neurons, ReLu activation,* 10^5 *iterations of S.G. with 4000 points.*

One simulation example

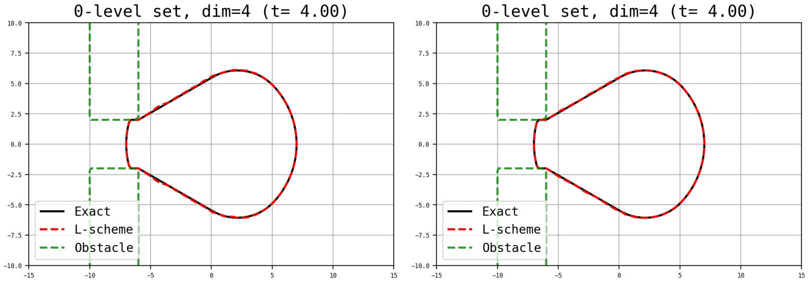


Figure – Left : $N_{\text{it}} = 8$, right : $N_{\text{it}} = 16$, Lagrangian scheme.

Parameters : $b = e_1$, $c = 1/2$, 3 layers of 60 neurons, ReLu activation, 10^5 iterations of S.G. with 4000 points.

One simulation example

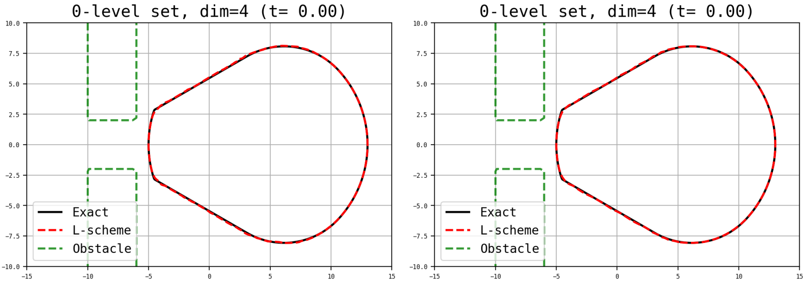


Figure – Left : $N_{\text{it}} = 8$, right : $N_{\text{it}} = 16$, Lagrangian scheme.

Parameters : $b = e_1$, $c = 1/2$, *3 layers of 60 neurons, ReLu activation,* 10^5 *iterations of S.G. with 4000 points.*

Error

d	Global errors		Local errors		Time
	L_∞	L_1 rel.	L_∞	L_1 rel.	
2	2.66e-01	5.99e-03	1.19e-01	4.61e-02	3h02
4	3.90e-01	6.77e-03	1.16e-01	2.69e-02	8h13
6	9.69e-01	1.09e-02	1.78e-01	2.88e-02	35h20

Table – Errors when the dimension d increases.

Parameters : $b = e_1$, $c = 1/2$, *3 layers of 60 neurons, ReLu activation,* 10^5 *iterations of S.G. with 4000 points.*

Conclusion & future work

Done :

- Lagrangian scheme for obstacle problems in high dimension.

Conclusion & future work

Done :

- Lagrangian scheme for obstacle problems in high dimension.
- On one hand, convergence result for fixed N.

Conclusion & future work

Done :

- Lagrangian scheme for obstacle problems in high dimension.
- On one hand, convergence result for fixed N.
- On the other hand, hope in numerical results.

Conclusion & future work

Done :

- Lagrangian scheme for obstacle problems in high dimension.
- On one hand, convergence result for fixed N.
- On the other hand, hope in numerical results.

To do :

- Influence of the training measure μ.

Conclusion & future work

Done :

- Lagrangian scheme for obstacle problems in high dimension.
- On one hand, convergence result for fixed N.
- On the other hand, hope in numerical results.

To do :

- Influence of the training measure μ.
- Scheme of order > 1 for discontinuous right hand-side ODE !

Thank you

Albert Altarovici, Olivier Bokanowski, and Hasnaa Zidani.
A general Hamilton-Jacobi framework for non-linear state-constrained control problems.
ESAIM : Control, Optimisation and Calculus of Variations, 19(2) :337–357, April 2013.

László Györfi, Michael Kohler, Adam Krzyżak, and Harro Walk.
A Distribution-Free Theory of Nonparametric Regression.
Springer Series in Statistics. Springer New York, New York, NY, 2002.

Ugo Tanielian, Maxime Sangnier, and Gerard Biau.
Approximating Lipschitz continuous functions with GroupSort neural networks.
2020.