# Neural networks for first order HJB equations and application to front propagation with obstacle terms*

Olivier Bokanowski,[†‡] Averil Prost,[§] Xavier Warin[¶]

**Abstract**

We consider a deterministic optimal control problem, focusing on a finite horizon scenario. Our proposal involves employing deep neural network approximations to capture Bellman's dynamic programming principle. This also corresponds to solving first-order Hamilton-Jacobi-Bellman equations. Our work builds upon the research conducted by Huré et al. (SIAM J. Numer. Anal., vol. 59 (1), 2021, pp. 525-557), which primarily focused on stochastic contexts. However, our objective is to develop a completely novel approach specifically designed to address error propagation in the absence of diffusion in the dynamics of the system. Our analysis provides precise error estimates in terms of an average norm. Furthermore, we provide several academic numerical examples that pertain to front propagation models incorporating obstacle constraints, demonstrating the effectiveness of our approach for systems with moderate dimensions (e.g., ranging from 2 to 8) and for nonsmooth value functions.

**Keywords**: neural networks, deterministic optimal control, dynamic programming principle, first order Hamilton-Jacobi-Bellman equation, state constraints

## 1 Introduction

In this work, we are interested by the approximation of a deterministic optimal control problem with finite horizon involving a maximum running cost, defined as

$$v(t,x) = \inf_{a(\cdot)\in\mathbb{A}_{[t,T]}} \max\left( \max_{\theta\in[t,T]} g(y_x^a(\theta)),\ \varphi(y_x^a(T))\right), \tag{1}$$

where the state $x$ belongs to $\mathbb{R}^d$ and $t \in [0,T]$ for some $T \geq 0$. Here the trajectory $y(s) = y_x^a(s)$ obeys the following dynamics

$$\dot{y}(s) = f(y(s), a(s)), \quad a.e.\ s \in [t,T], \tag{2}$$

with initial condition $y(t) = x$, and control $a \in \mathbb{A}_{[t,T]} := L^\infty([t,T], A)$. It is assumed that $A$ is a non-empty convex and compact subset of $\mathbb{R}^\kappa$ ($\kappa \geq 1$) and ($f$, $\varphi$, $g$) are Lipschitz continuous. The value $v$ is the solution of the following Hamilton-Jacobi-Bellman (HJB) partial differential equation, in the viscosity sense (see for instance [14])

$$\min\left( - v_t + \max_{a\in A}(-\nabla_x v \cdot f(x,a)),\ v - g(x)\right) = 0, \quad t \in [0,T] \tag{3a}$$

$$v(T,x) = \max(\varphi(x), g(x)). \tag{3b}$$

*This research benefited from the support of the FMJH Program PGMO and from the support to this program from EDF.

†Université Paris Cité, Laboratoire Jacques-Louis Lions (LJLL), F-75013 Paris, France

‡Sorbonne Université, CNRS, LJLL, F-75005 Paris, France olivier.bokanowski@u-paris.fr

§INSA Rouen Normandie, Normandie Univ, LMI UR 3226, F-76000 Rouen, France. averil.prost@insa-rouen.fr

¶EDF R&D & FiME, 91120 Palaiseau, France xavier.warin@edf.fr

Tremendous numerical efforts have been made to propose efficient algorithms for solving problems related to equation (1) or the corresponding HJB equation (3). Precise numerical methods have been developed using various grid-based approximations, including Markov Chains approximations [33], finite difference schemes (such as monotone schemes [17]), semi-Lagrangian schemes (see e.g., [18, 20]), ENO or WENO higher-order schemes [37, 39], finite element methods [31], discontinuous Galerkin methods [28, 35], and specifically [12, 13] for the HJB equation (3) (see also [41]), as well as max-plus approaches [2].

However, grid-based methods have limitations in high dimensions due to the curse of dimensionality. To address this challenge, various alternative approaches have been studied, such as sparse grid methods [16, 21], tree structure approximation algorithms (see e.g., [3]), tensor decomposition methods [19], and max-plus approaches [36].

In the deterministic context, problem (1) is motivated by deterministic optimal control with state constraints (see e.g. [14] and [4]). In [42], the HJB equation (3) is approximated by deep neural networks (DNN) for solving state constrained reachability control problems of dimension up to $d = 10$. In [15] or in [6], formulation (1) is used to solve an abort landing problem (using different numerical approaches); in [11], equations such as (3) are used to solve an aircraft payload optimization problem; a multi-vehicle safe trajectory planning is considered in [8].

On the other hand, for stochastic control, DNN approximations were already used for gas storage optimization in [9], where the control approximated by a neural network was the amount of gas injected or withdrawn in the storage. This approach has been adapted and popularized recently for the resolution of BSDE in [26] (deep BSDE algorithm). For a convergence study of such algorithms in a more general context, see [27].

In this work, we study some neural network approximations for (1). We are particularly interested for the obtention of a rigorous error analysis of such approximations. We follow the approach of [29] and its companion paper [7], combining neural networks approximations and Bellman's dynamic programming principle. We obtain accurate error estimates in an average norm.

Note that the work of [29] is developed in the stochastic context, where an error analysis is given. However this error analysis somehow relies strongly on a diffusion assumption of the model, (transition probabilities with densities are assumed to exists). In our case, we would need to assume that the deterministic process admits a density, which is overly restrictive (see remark 6.5). Therefore the proof of [29] does not apply to the deterministic context. Here we propose a new approach for the convergence analysis, leading to new error estimates. We chose to present the algorithm on a running cost optimal control problem, but the approach can be generalized to Bolza or Mayer problems (see e.g. [4, 6]).

For sake of completeness, let us notice that the ideas of [29] are related to methods already proposed in [23] and [10] for the resolution of Backward Stochastic Differential Equations (BSDE), where the control function is calculated by regression on a space of some basis functions (the *Hybrid-Now* algorithm is related to [23], and the *performance iteration* algorithm is related to an improved algorithm in [10]). For recent developments, see [24] using classical linear regressions, and [30] and [22] for BSDE approximations using neural networks.

From the numerical point of view, we illustrate our algorithms on some academic front-propagation problems with or without obstacles. We focus on a "Lagrangian scheme" (a deterministic equivalent of the *performance iteration* scheme of [29]), and also compare with other algorithms: a "semi-Lagrangian algorithm" (similar to the *Hybrid-Now* algorithm of [29]) and an hybrid algorithm combining the two previous, involving successive projections of the value function on neural network spaces.

The plan of the paper is the following. In Section 2 we define a semi-discrete value approximation for (1), with controlled error with respect to the continuous value, using piecewise constant controls. In Section 3, equivalent reformulations of the problem are given using feedback controls and dynamic programming. In Section 4, an approximation result of the discrete value function using Lipschitz continuous feedback controls is given. In Section 5 we present three numerical schemes using neural networks approximations (for the approximation of feedback controls and/or for the value), using general Runge Kutta schemes for the approximation of the controlled dynamics. Section 6 contains our main convergence result for one of the proposed scheme, the "Lagrangian" scheme, which involves only approximations of the feedback controls, and Section 7 focuses on the proof of our main result. Section 8 is devoted to some numerical academic examples of front-propagation problems with or without an obstacle term (state constraints), for average dimensions, showing the potential of the proposed algorithms in this context, and also giving comparisons between the different algorithms introduced. An Appendix contains some details for computing reference solutions for some of the considered examples.

**Notations.** Unless otherwise stated, the norm $|.|$ on $\mathbb{R}^q$ ($q \geq 1$) is the max norm $|x| = \|x\|_\infty = \max_{1\leq i\leq q} |x_i|$. The notation $[\![p, q]\!] = \{p, p+1, \ldots, q\}$ is used, for any natural numbers $p \leq q$. For any function $\alpha : \mathbb{R}^p \to \mathbb{R}^q$ for some $p, q \geq 1$, $[\alpha] := \sup_{y\neq x} \frac{|\alpha(y)-\alpha(x)|}{|y-x|}$ denotes the corresponding Lipschitz constant. We also denote $a \vee b := \max(a, b)$ for any $a, b \in \mathbb{R}$. The set of "feedback" controls is defined as $\mathcal{A} := \{a : \mathbb{R}^d \to A,\ a(\cdot) \text{ measurable}\}$.

**Note.** All data generated or analysed during this study are included in this article.

# 2 Semi-discrete approximation with piecewise constant controls

In this section, we first aim to define a semi-discrete approximation of the Dynamic Programming Principle (1) in time. Let the following assumptions hold on the set $A$ and functions $f, g, \varphi$.

**(H0)** *$A$ is a non-empty, convex and compact subset of $\mathbb{R}^\kappa$.*

**(H1)** *$f : \mathbb{R}^d \times A \to \mathbb{R}^d$ is Lipschitz continuous and there exists constants $[f]_1, [f]_2 \geq 0$ such that, for any $(x, x') \in (\mathbb{R}^d)^2$ and $(a, a') \in A^2$:*

$$|f(x, a) - f(x', a')| \ \leq\ [f]_1|x - x'| + [f]_2|a - a'|.$$

**(H2)** *$g : \mathbb{R}^d \to \mathbb{R}$ is Lipschitz continuous.*

**(H3)** *$\varphi : \mathbb{R}^d \to \mathbb{R}$ is Lipschitz continuous.*

Let $T > 0$ be the horizon, $N \in \mathbb{N}^*$ be a number of iterations, and $(t_k)_{k\in[\![0,N]\!]} \subset [0, T]$ be a time mesh with $t_0 = 0$ and $t_N = T$. To simplify the presentation, we restrict ourselves to the uniform mesh $t_k = k\Delta t$ with $\Delta t = \frac{T}{N}$, but the arguments would carry over unchanged with a non-uniform time mesh.

For a given $h > 0$, let us consider $F_h : \mathbb{R}^d \times A \to \mathbb{R}^d$ corresponding to some one time step approximation of $y^a_x(h)$ starting from $y^a_x(0) = x$. For instance, we may consider the Euler scheme $F_h(x, a) = x + hf(x, a)$, or the Heun scheme $F_h(x, a) = x + \frac{h}{2}(f(x, a) + f(x + hf(x, a), a))$, and so on. General Runge Kutta schemes are considered later on in Section 5.2. Assumptions on $F_h$ will be made precise when needed.

Given $n \in [\![0, N-1]\!]$, a sequence $a := (a_n, a_{n+1}, \ldots, a_{N-1}) \in A^{N-n}$ which corresponds to a piecewise constant control approximation, and an integer $p \geq 1$, we define two levels of approximation for the trajectories.

The fine approximation, for a fixed control $a_k$ and $0 \leq j \leq p$ is denoted $Y^{a_k}_{j,x}$. It involves a time step $h = \frac{\Delta t}{p}$, and is defined recursively by

$$Y^{a_k}_{0,x} = x \tag{4a}$$
$$Y^{a_k}_{j+1,x} = F_h(Y^{a_k}_{j,x}, a_k), \quad j = 0, \ldots, p-1, \tag{4b}$$

which also corresponds to $j$ iterates of $y \to F_h(y, a_k)$, starting from $y = x$, with the same control $a_k$. This fine level is designed to obtain approximation of the trajectory at intermediate time steps $t_k + jh$ which lie into $[t_k, t_{k+1}]$.

The coarse approximation with time step $\Delta t$ is denoted $(X^a_{k,x})_{n\le k\le N}$ and is defined recursively by

$$X^a_{n,x} := x \tag{5a}$$

$$X^a_{k+1,x} = Y^{a_k}_{p,X^a_{k,x}}, \quad k = n, \ldots, N-1. \tag{5b}$$

**Notations.** Given $a \in A$, we denote $F(\cdot, a) \equiv F^a(\cdot) := Y^a_{p,\cdot}$ , such that (5b) can also be written

$$X^a_{k+1,x} = F(X^a_{k,x}, a_k), \quad k = n, \ldots, N-1.$$

We can now define the following cost functional, for $x \in \mathbb{R}^d$, $a \in A^{N-n}$ and $n \in [\![0, N]\!]$

$$J_n(x, a) := \max_{n\le k<N} \max_{0\le j<p} g(Y^{a_k}_{j,X^a_{k,x}}) \bigvee (g \vee \varphi)(X^a_{N,x}), \quad x \in \mathbb{R}^d,\ a \in A^{N-n}, \tag{6}$$

and the following semi-discrete version of (1), for $x \in \mathbb{R}^d$ and $n \in [\![0, N]\!]$

$$V_n(x) := \min_{a\in A^{N-n}} J_n(x, a). \tag{7}$$

We also denote, for $a \in A$ and $x \in \mathbb{R}^d$,

$$G^a(x) := \max_{0\le j<p} g(Y^a_{j,x}). \tag{8}$$

The values $(V_n)_{0\le n\le N}$ satisfy therefore $V_N(x) = g(x) \vee \varphi(x)$, and the following dynamic programming principle (DPP) for $n = 0, \ldots, N-1$:

$$V_n(x) = \inf_{a\in A} G^a(x) \vee V_{n+1}(F(x, a)), \quad x \in \mathbb{R}^d. \tag{9}$$

Let us notice that the case $p = 1$ leads to the following simplifications: $h = \Delta t$, $F(x, a) = F_{\Delta t}(x, a)$, $G^a(x) = g(x)$, $J_n(x) = \max_{n\le k\le N} g(X^a_{k,x}) \vee \varphi(X^a_{N,x})$, as well as $V_N(x) = g(x) \vee \varphi(x)$ and the DPP $V_n(x) = \inf_{a\in A} g(x) \vee V_{n+1}(F_{\Delta t}(x, a))$ for all $0 \le n \le N-1$.

The motivation behind the introduction of the finer level of approximation $(Y^{a_k}_{j,x})_{0\le j\le p}$ is first numerical. It enables a better evaluation of the running cost term $g(\cdot)$ along the trajectory, without the computational cost of more intermediate controls. The numerical improvement is illustrated in the examples of Section 8.1. Furthermore, from the theoretical point of view, the convergence analysis in our main result strongly uses the fact that $x \to F_h(x, a)$ is a change of variable for $h$ sufficiently small (i.e., $p$ sufficiently large).

We start by showing some uniform Lipschitz bounds.

**Lemma 2.1.** *Assume (H0)-(H3), and the Lipschitz bound* $\sup_{a\in A}[F_h(\cdot, a)] \le 1 + ch$ *for some constant* $c \ge 0$ *, where we recall that* $[F_h(\cdot, a)]$ *stands for the Lipschitz constant of* $x \mapsto F_h(x, a)$.

$(i)$ *The function* $J_n(\cdot, a)$ *is Lipschitz for all* $a \in A^{N-n}$*, with uniform bound* $[J_n(\cdot, a)] \le [g] \vee [\varphi]e^{cT}$.

$(ii)$ *In particular, the uniform bound* $\max_{0\le n\le N}[V_n] \le [g] \vee [\varphi]e^{cT}$ *holds.*

*Proof.* $(i)$ Notice that $1+ch \le e^{ch}$. Then for $a \in A$ and for the $j$-th iterate $F^{(j)}_h(\cdot, a)$ of $F_h$, we obtain $|Y^a_{j,x} - Y^a_{j,y}| = |F^{(j)}_h(x, a) - F^{(j)}_h(y, a)| \le e^{jch}|x - y| \le e^{c\Delta t}|x - y|$ for any $0 \le j \le p$ (by recursion). In particular with $j = p$ and from the definitions, we obtain also $|F(x, a)-F(y, a)| \le e^{c\Delta t}|x-y|$, for any $a \in A$. From this last inequality, we deduce, by recursion, for any $a = (a_n, \ldots, a_{N-1}) \in A^{N-n}$ and $n \le k \le N$, $|X^a_{k,x} - X^a_{k,y}| \le e^{c(k-n)\Delta t}|x - y| \le e^{cT}|x - y|$. The desired result follows from the definition of $J_n$ and repeated use of $\max(a, b) - \max(c, d) \le \max(a - c, b - d)$.

$(ii)$ As a direct consequence of $(i)$ and the definition of $V_n$. □

The following result shows that $V_n(x)$ is a first order approximation of $v(t_n, x)$ in time.

**Theorem 2.2.** *Assume (H0)-(H3), and that there exists $h_0 > 0$ such that*

- *$F_h$ is consistent with the dynamics $f$ in the following sense: there exists $C \geq 0$, for all $(x, a, h) \in \mathbb{R}^d \times A \times (0, h_0)$*

$$|F_h(x, a) - (x + hf(x, a))| \leq Ch^2(1 + |x|) \tag{10}$$

- *for all $h \in ]0, h_0]$, $\sup_{a \in A}[F_h(\cdot, a)] \leq 1 + ch$ for some constant $c \geq 0$ that may depend on $[f]$*
- *for all $x \in \mathbb{R}^d$, $f(x, A)$ is a convex set.*

*Let $h = \frac{\Delta t}{p} \leq h_0$, with $p \geq 1$. Then there exists a constant $\widetilde{C} \geq 0$, independent of $x, \Delta t, p$ and $N$, such that for all $x \in \mathbb{R}^d$:*

$$\max_{0 \leq n \leq N} |V_n(x) - v(t_n, x)| \leq \widetilde{C} \Delta t (1 + |x|). \tag{11}$$

*Proof.* This follows from the arguments of Theorem B.1. in [15]. □

**Corollary 2.3.** *In particular, if $F_h$ is a consistent RK scheme as in Definition 5.2 and $f(x, A)$ is convex for all $x$, then by Lemma 5.6,* (10) *holds and therefore the error estimate* (11) *also holds.*

Our aim is now to propose numerical schemes for the approximation of $V_n(\cdot)$.

# 3 Reformulation with feedback controls

In this section, equivalent definitions for $V_n$ are given using feedback controls in $\mathcal{A}$ (the set of measurable functions $a : \mathbb{R}^d \to A$). These formulations lead to the numerical schemes detailed in Section 5.

First, for a given $a_k \in \mathcal{A}$, the fine approximation $Y^{a_k}_{x,j}$ with time step $h = \frac{\Delta t}{p}$ is defined by $Y^{a_k}_{x,j} := Y^{a_k(x)}_{x,j}$ using Definition (4). This corresponds also to

$$Y^{a_k}_{0,x} = x \tag{12}$$

$$Y^{a_k}_{j+1,x} = F_h(Y^{a_k}_{j,x}, a_k(x)), \quad j = 0, \dots, p-1 \tag{13}$$

that is, $j$ iterates of $y \to F_h(y, a_k(x))$, starting from $y = x$, with the fixed control $a_k(x)$. Let us also introduce the following notation, for any $a \in \mathcal{A}$ and $x \in \mathbb{R}^d$:

$$F^a(x) := F(x, a(x)).$$

Then, for a given sequence $a := (a_n, \dots, a_{N-1}) \in \mathcal{A}^{N-n}$, the coarse approximation is defined by

$$X^a_{n,x} = x \tag{14a}$$

$$X^a_{k+1} = F^{a_k}(X^a_{k,x}) \equiv F\left(X^a_{k,x}, a_k(X^a_{k,x})\right), \quad k = n, \dots, N-1. \tag{14b}$$

We also extend the definition of $J_n$ in the case of feedback controls, for $x \in \mathbb{R}^d$ and $a \in \mathcal{A}^{N-n}$, as

$$J_n(x, a) := \big(\max_{n \leq k \leq N} G^{a_k}(X^a_{k,x})\big) \bigvee \varphi(X^a_{N,x}) \tag{15}$$

where now, for a given control $a \in \mathcal{A}$, we extend the definition of (8) by

$$G^a(x) := \max_{0 \leq j < p} g(Y^{a(x)}_{j,x}). \tag{16}$$

Note that this also corresponds to define $G^a(x)$ as $G^{a(x)}(x)$. With this definitions, we have the following results.

**Proposition 3.1.** *Assume (H0)-(H3), and let* $V_n$ *be defined as in* (7).

(*i*) $V_n(x)$ *is the minimum of* $J_n(x,\cdot)$ *over feedback controls:*

$$V_n(x) = \min_{a\in\mathcal{A}^{N-n}} J_n(x,a), \quad x\in\mathbb{R}^d.$$

(*ii*) *For all* $0 \le n \le N-1$, $V_n$ *satisfies the following dynamic programming principle over feedback controls*

$$V_n(x) = \min_{a\in\mathcal{A}} G^a(x) \bigvee V_{n+1}(F^a(x)), \quad x\in\mathbb{R}^d,\ n=0,\ldots,N-1 \tag{17}$$

*and in particular, the infimum is reached by some some* $\bar a_n \in \mathcal{A}$.

*Proof.* The problem is to show the existence of a measurable feedback control. Considering the equivalent formulation (9) of (17), that $(x,a)\to G^a(x)$ and $(x,a)\to F^a(x)$ are continuous fonctions on $\mathbb{R}^d\times A$, and that $V_{n+1}$ is also continous on $\mathbb{R}^d$, by using a measurable selection procedure as in [34, Lemmas 2A, 3A p. 161], and since $A$ is compact we may choose $\bar a_n$ measurable in (17). □

The following well-known result links pointwise minimization over open-loop controls $a\in A$ and minimization of an averaged value over feedback controls $a\in\mathcal{A}$.

**Lemma 3.2.** *Let* $X$ *be a random variable with values in* $\mathbb{R}^d$ *which admits a density* $\rho$, *and such that* $\mathbb{E}[|X|]<\infty$. *Then for any measurable* $\Omega\subset\mathbb{R}^d$ *such that* $\rho(x)>0$ *a.e.* $x\in\Omega$, *and* $n\in[\![0,N-1[\![$,

$$\bar a_n(\cdot) \in \operatorname*{argmin}_{a\in\mathcal{A}} \mathbb{E}\Big[\mathbb{1}_\Omega(X)\ G^a(X)\vee V_{n+1}(F^a(X))\Big]$$
$$\iff \quad \Big(\bar a_h(x) \in \operatorname*{argmin}_{a\in A}\big(G^a(x)\vee V_{n+1}(F^a(x))\big),\ \textit{a.e. } x\in\Omega\Big).$$

We now introduce a new assumption on a sequence of sets $\Omega_n$, densities $\rho_n$ supported in $\bar\Omega_n$, and random variables $X_n$ with associated probability densities $\rho_n$. More precisely:

**(H4)** *for all* $k=0,\ldots,N$, $\rho_k\in L^1(\mathbb{R}^d)$, $\Omega_k$ *is an open set of* $\mathbb{R}^d$, *and it holds:*

$$\textit{for all } k=0,\ldots,N\textit{: } \rho_k(x)>0 \textit{ on } \Omega_k, \textit{ and } supp(\rho_k)\subset\overline{\Omega_k} \tag{18a}$$

$$\textit{for all } a\in A \textit{ and } k=0,\ldots,N-1\textit{: } F(\Omega_k,a)\subset\Omega_{k+1} \tag{18b}$$

$$\textit{for all } k=0,\ldots,N-1\textit{: } C_{k,\Delta t} := \sup_{x\in\Omega_k}\sup_{a\in A}\frac{\rho_k(x)}{\rho_{k+1}(F(x,a))}<\infty. \tag{18c}$$

*Furthermore, we consider random variables* $(X_k)_{0\le k\le N}$ *on some probability space, with values in* $\mathbb{R}^d$, *absolutely continuous with respect to Lebesgue's measure and admitting* $(\rho_k)_{0\le k\le N}$ *as associated densities.*

From the definitions we have $\mathbb{E}[\phi(X_k)] = \int_{\Omega_k}\phi(x)\rho_k(x)dx$ for any measurable bounded function $\phi$.

The technical assumption (18c) is not important in this section but will be needed for the main result later on. Before going on, we give some examples where (H4) holds.

- In the case $\Omega_0$ is a bounded subset of $\mathbb{R}^d$, we may consider $\Omega_k = \Omega_0 + \mathscr{B}(0, ck\Delta t\|f\|_\infty)$, where $\mathscr{B}(0,r)$ denotes the ball of radius $r$, $\|f\|_\infty$ is a bound for $|f|$ on $\Omega_N\times A$, assuming $|F_h(x)|\le|x|+ch\|f\|_\infty$ for some constant $c\ge0$ (such a bound holds in the case of a RK scheme as in Definition (5.2)), furthermore where $(\rho_k(\cdot))_k$ are bounded functions such that there exists $\eta>0$, for all $0\le k\le N-1$ and $x\in\Omega_{k+1}$, $\rho_k(x)\ge\eta$.

A useful example is the case of $\Omega_k := \mathscr{B}(0, L_0 + ck\Delta t\|f\|_\infty)$, $\forall k \geq 0$, with uniform densities $\rho_k$ compactly supported on $\Omega_k$. In that case we notice that $C_{k,\Delta t} = \frac{|\Omega_{k+1}|}{|\Omega_k|} = \big(\frac{L_0+c(k+1)\Delta t\|f\|_\infty}{L_0+ck\Delta t\|f\|_\infty}\big)^d$ and also the following uniform estimate holds:

$$\max_{0\leq k<N} \prod_{k=n}^{N-1} C_{k,\Delta t} \leq \left(\frac{L_0 + cT\|f\|_\infty}{L_0}\right)^d. \tag{19}$$

- We may consider the case of $\Omega_k = \mathbb{R}^d$ and $\rho_k(x) \overset{|x|\to\infty}{\sim} e^{-q_k|x|}$ with $q_k > 0$, for all $k$.
- We can consider the case when $\Omega_k = \Omega$ $(\forall k)$ and $\rho_k = \rho$ $(\forall k)$, where $\Omega$ is a bounded set, assuming furthermore that $\Omega$ is invariant by the dynamics, i.e., $F_h^a(\Omega) \subset \Omega$ for all $a \in A$ and $h \geq 0$.

We can now give the following equivalent properties for $V_n$.

**Proposition 3.3.** *Let $n \in [\![0, N-1]\!]$ and $(\Omega_n, \rho_n)$ as in (H4), with associated random variables $X_n$. Then $V_n$ satisfies the following dynamic programming principle*

$$V_n(x) = G^{\bar{a}_n}(x) \bigvee V_{n+1}(F^{\bar{a}_n}(x)), \quad \forall x \in \Omega_n \tag{20}$$

*for any*

$$\bar{a}_n(\cdot) \in \operatorname*{argmin}_{a\in\mathcal{A}} \mathbb{E}\Big[G^a(X_n) \vee V_{n+1}(F^a(X_n))\Big]. \tag{21}$$

*In particular, we have*

$$\mathbb{E}[V_n(X_n)] = \mathbb{E}\Big[G^{\bar{a}_n}(X_n) \bigvee V_{n+1}(F^{\bar{a}_n}(X_n))\Big] = \inf_{a\in\mathcal{A}} \mathbb{E}\Big[G^a(X_n) \vee V_{n+1}(F^a(X_n))\Big]. \tag{22}$$

*Proof.* The proof follows from Lemma 3.2 and the dynamic programming principle of Proposition 3.1. □

The above reformulation with an averaging criteria is motivated by numerical aspects: the problem can then be relaxed with an approximation $\hat{\mathcal{A}}$ of the control space $\mathcal{A}$, for instance neural networks. However, in general, $\bar{a}_n$ is no more than measurable. To circumvent this difficulty, we first approximate problem (22) by more regular feedback controls.

# 4 Approximation by Lipschitz continuous feedback controls

We aim to approximate (22) by using by Lipschitz continuous feedback controls. Note that in Krylov [32], some approximations using feedback controls are given, yet in a different context with stochastic differential equations and for non-degenerate diffusions.

Let $\rho \in \mathcal{C}^1(\mathbb{R}^d, \mathbb{R})$ be a smooth function such that $\text{supp}(\rho) \subset \mathscr{B}(0,1)$, and $\int_{\mathbb{R}^d} \rho(x)dx = 1$. Let $(\rho_\tau)_{\tau>0}$ be the mollifying sequence such that $\rho_\tau(x) := \frac{1}{\tau^d}\rho(\frac{x}{\tau})$. For any sequence $a := (a_0, \ldots, a_{N-1}) \in \mathcal{A}^N$, we associate the regularization by convolution

$$a_k^\tau := \rho_\tau * a_k. \tag{23}$$

By using classical arguments, $a_k^\tau$ is Lipschitz continuous, with the bounds $\|\nabla a_k^\tau\|_{L^\infty} = \|(\nabla\rho_\tau) * a_k\|_{L^\infty} \leq \frac{1}{\tau}\|\nabla\rho\|_{L^1}\|a_k\|_{L^\infty}$. By classical arguments, $\lim_{\tau\to 0} a^\tau(x) = a(x)$ a.e $x \in \mathbb{R}^d$.

In this section the following assumptions on $F_h$ are needed.

**(H5)** *The function $F_h$ satisfies the following two conditions:*

- *there exists a constant $C > 0$ and $h_0 > 0$ such that, for all $0 < h \le h_0$:*

$$\forall x \in \mathbb{R}^d,\ \forall a \in A, \qquad |F_h(x,a)| \le |x| + Ch(1+|x|) \tag{24}$$

- *$F_h$ satisfies a continuity property, for all $0 < h \le h_0$:*

$$\forall x \in \mathbb{R}^d, \quad a \in A \to F_h(x,a) \in \mathbb{R}^d \text{ is continuous.} \tag{25}$$

Such assumptions are naturally satisfied by Euler or Heun schemes already mentioned, and also by more general RK schemes presented later on.

The following result will be used later on in order to obtain a regularized sequence of controls for the approximation of the dynamic programming principle, which becomes more and more precise as $k$ varies from $k = n$ to $k = N-1$.

**Proposition 4.1.** *Let $k \in [\![0, N]\!]$. Assume (H0)-(H4) and (H5). Then*

$$\lim_{\tau \to 0^+} \left| \mathbb{E}[V_k(X_k)] - \mathbb{E}[G^{\bar{a}_k^\tau}(X_k) \bigvee V_{k+1}(F(X_k, \bar{a}_k^\tau))] \right| = 0 \tag{26}$$

*with $\bar{a}_k$ as in (21) and $\bar{a}_k^\tau$ as in (23).*

**Lemma 4.2.** *Assume (24). There exists constants $\alpha, \beta$ independent of $p \ge 1$, such that*

$$\forall (x,a) \in \mathbb{R}^d \times A, \quad |F(x,a)| \le \alpha |x| + \beta.$$

*Proof.* By using the bound $|F_h(x,a)| \le |x| + Ch(1+|x|) \le e^{Ch}|x| + Ch$, by recursion and a discrete Gronwall estimate, we obtain $|F(x,a)| = |F_h^{(p)}(\cdot, a)| \le e^{Chp}(|x| + pCh) \le e^{C\Delta t}(|x| + C\Delta t)$. □

*Proof of Proposition 4.1.* In order to simplify the presentation, we consider the case of $p = 1$ ($G^a(x) = g(x)$), the proof being similar in the general case $p \ge 1$. The optimal control $\bar{a}_n(\cdot)$ satisfies $V_n(x) = g(x) \vee V_{n+1}(F(x, \bar{a}_n(x)))$ a.e. $x \in \Omega_n$, hence $\mathbb{E}[V_n(X_n)] = \mathbb{E}[g(X_n) \vee V_{n+1}(F(X_n, \bar{a}_n))]$. By assumption (25) of (H5), and the pointwise convergence of $\bar{a}_n^\tau(x)$, we deduce that $\lim_{\tau \to 0} F(x, \bar{a}_n^\tau(x)) = F(x, \bar{a}_n(x))$ a.e. $x$. On the other hand, by using Lemma 4.2 and the fact that $g$ is Lipschitz continuous, there exists constants $\alpha', \beta'$ such that $|g(x) \vee F(x, \bar{a}_n^\tau(x))| \le \alpha'|x| + \beta'\ \forall x \in \mathbb{R}^d$. Therefore, the result is obtained by Lebesgue's dominated convergence theorem, the continuity of $(x,y) \to g(x) \vee V_{n+1}(y)$ and the integrability assumption on $X_n$. □

We give also an other approximation result of $V_n$ by $J_n$.

**Proposition 4.3.** *Let $N \ge 1$ be given, assume (H0)-(H4), and (H5). Then*

$$\lim_{\tau \to 0} \max_{0 \le n \le N-1} \mathbb{E}\Big[ |J_n\big(X_n, (\bar{a}_n^\tau, \ldots, \bar{a}_{N-1}^\tau)\big) - V_n(X_n)| \Big] = 0.$$

*Proof.* It suffices to prove the result for a given $n$. Let $a \in \mathcal{A}^{N-n}$ be an arbitrary sequence. By Lemma 4.2 we have, for all $k = n, \ldots, N-1$

$$|X_{k+1,x}^a| \le e^{C\Delta t}(|X_{k,x}^a| + C\Delta t).$$

By similar estimates, we obtain

$$|X_{n,x}^a| \le e^{Cn\Delta t}(|x| + Cn\Delta t) \le e^{CT}(|x| + CT).$$

Hence

$$|J_n(x,a)| \le \max_{n \le k \le N} \left[ |g(0)| + [g] \left| X_{k,x}^a \right| \right] \bigvee \left[ |\varphi(0)| + [\varphi] \left| X_{N,x}^a \right| \right] \le K_0 + K_1(|x| + CT)$$

where $K_0 := |g(0)| \vee |\varphi(0)|$ and $K_1 := ([g] \vee [\varphi])e^{CT}$. In particular $\mathbb{E}(|J_n(X,a)|) < \infty$. In the same way, we can also obtain, for any $k \leq n \in [0,N]$, a Lipschitz bound of the form

$$|G^a(x) \vee G^a(X^a_{k+1,x}) \vee \cdots \vee G^a(X^a_{n,x}) \vee V_{n+1}(X^a_{n+1,x}))| \leq \alpha'|x| + \beta'$$

for some constant $\alpha', \beta'$.

In order to simplify the presentation, we consider again the case of $p = 1$, corresponding to $G^a(x) = g(x)$, the proof being similar in the general case $p \geq 1$. Let $\eta > 0$. Consider the optimal control sequence $\bar{a}$ and its regularization $\bar{a}^\tau$. Let $n \in [0, \ldots, N-1]$. By using the optimality of $\bar{a}_n$, we have $V_n(x) = g(x) \vee V_{n+1}(F^{\bar{a}_n}(x))$, and as in Proposition 4.1, for $\tau > 0$ small enough,

$$\left|\mathbb{E}[V_n(X_n)] - \mathbb{E}[g(X_n) \vee V_{n+1}(F^{\bar{a}^\tau_n}(X_n))]\right| \leq \frac{\eta}{N}.$$

Then we remark that $V_{n+1}(F^{\bar{a}^\tau_n}(x)) = g(F^{\bar{a}^\tau_n}(x)) \vee V_{n+2}(F^{\bar{a}_{n+1}}(F^{\bar{a}^\tau_n}(x)))$. Hence by the same argument as before, for $\tau > 0$ small enough,

$$\left|\mathbb{E}[V_n(X_n)] - \mathbb{E}[g(X_n) \vee g(F^{\bar{a}^\tau_n}(X_n)) \vee V_{n+2}(F^{\bar{a}^\tau_{n+1}}(F^{\bar{a}^\tau_n}(X_n)))]\right| \leq 2\frac{\eta}{N}.$$

Iterating this argument we deduce the existence of Lipschitz continuous controls $\bar{a}^\tau := (\bar{a}^\tau_n, \ldots, \bar{a}^\tau_{N-1})$ such that

$$\left|\mathbb{E}[V_n(X_n)] - \mathbb{E}[J_n(X_n, \bar{a}^\tau)]\right| \leq (N-n)\frac{\eta}{N} \leq \eta. \tag{27}$$

This concludes the proof. □

# 5 Numerical schemes

## 5.1 Dynamic programming schemes

It is natural to consider approximation schemes that mimic the dynamic programming principle (20)-(21). We see that (9) or its equivalent formulation (17) has been relaxed by (21) where we minimize a certain expectation over a set of feedback controls.

We consider three schemes. Two of them may be seen as deterministic counterparts of the "value iteration" scheme (Huré *et al.* [29] or the BSDE scheme of [23]) and the "performance iteration" scheme (Huré *et al.* [29] or the BSDE scheme of [10]), hereafter denoted the "SL-scheme" and the "L-scheme", respectively.

The third one is an hybrid combination of both paradigm, and is hereafter denoted the "H-scheme".

The set of measurable functions $\mathcal{A}$ is approximated by finite dimensional spaces $(\hat{\mathcal{A}}_n)_n$, with $\hat{\mathcal{A}}_n$ typically a neural network space. When needed, neural networks are also used in order to approximate value functions: in this case, we denote by $\hat{\mathcal{V}}_n \subset \mathcal{C}(\Omega, \mathbb{R})$ a finite-dimensional space for the approximation of $\hat{V}_n$.

Let $(\rho_n)_{0\leq n\leq N}$ be a sequence of densities supported in domains $(\Omega_n)_{0\leq n\leq N}$, as in (H4), with associated random variables $(X_n)_{0\leq n\leq N}$. Recall that, for feedback controls $a \in \mathcal{A}$, $F^a(x)$ and $G^a(x)$ are defined at the beginning of Section 3 in terms of the approximate dynamics $F_h(\cdot,\cdot)$, $F^a(x)$ corresponds to $p$ iterates of $y \to F_h(y, a(x))$ starting from $y = x$, and $G^a(x)$ corresponds to the maximum of $g(\cdot)$ taken at the first previous $p$ iterates.

**Semi-Lagrangian scheme (or "SL-scheme")** Let $(\hat{\mathcal{A}}_n)_{n\in[\![0,N-1]\!]}$ and $(\hat{\mathcal{V}}_n)_{n\in[\![0,N-1]\!]}$ be two given sequences of finite-dimensional spaces. Set $\hat{V}_N := g \vee \varphi$. Then, for $n = N-1, \ldots, 0$:
- compute a feedback control $\hat{a}_n$ according to

$$\hat{a}_n \in \operatorname*{argmin}_{a\in\hat{\mathcal{A}}_n} \mathbb{E}\Big[G^a(X_n) \bigvee \hat{V}_{n+1}(F^a(X_n))\Big] \tag{28a}$$

- set

$$\hat{V}_n := \operatorname*{argmin}_{V\in\hat{\mathcal{V}}_n} \mathbb{E}\Big[\Big|V(X_n) - G^{\hat{a}_n}(X_n) \bigvee \hat{V}_{n+1}\big(F^{\hat{a}_n}(X_n)\big)\Big|^2\Big]. \tag{28b}$$

The approximations $(\hat{V}_n)_{n\in[\![0,N-1]\!]}$ are stored, and only $\hat{V}_{n+1}$ is used at iteration $n$. This explains the "semi-Lagrangian" terminology. Owing to these projections, the computational cost is in $O(N)$, where $N$ is the number of time steps.

**Lagrangian scheme (or "L-scheme")** Let $(\hat{\mathcal{A}}_n)_{n\in[\![0,N-1]\!]}$ be a given sequence of finite-dimensional spaces. Set $\hat{V}_N := g \vee \varphi$. Then, for $n = N-1, \ldots, 0$:
- compute a feedback control $\hat{a}_n$ according to

$$\hat{a}_n \in \operatorname*{argmin}_{a\in\hat{\mathcal{A}}_n} \mathbb{E}\Big[G^a(X_n) \bigvee \hat{V}_{n+1}(F^a(X_n))\Big] \tag{29a}$$

- set

$$\hat{V}_n(x) := G^{\hat{a}_n}(x) \bigvee \hat{V}_{n+1}\left(F^{\hat{a}_n}(x)\right) \equiv J_n(x, (\hat{a}_n, \ldots, \hat{a}_{N-1})). \tag{29b}$$

In this algorithm, only the feedback controls $(\hat{a}_k)$ are stored ( the $\hat{V}_n$ are not stored). Each evaluation of the value $\hat{V}_{n+1}(x)$ uses the previous controls $(\hat{a}_{n+1}, \ldots, \hat{a}_{N-1})$ to compute the approximated characteristic, in a fully Lagrangian philosophy. Therefore the overall computational cost is then of order $O(N^2)$. This scheme completely avoids projections of the value on functional subspaces.

**Remark 5.1.** *From a computational point of view, an approximation of the minimum* (29a) *is obtained by using a stochastic gradient algorithm, see numerical Section for details. Hence the optimality of $\hat{a}_k$ in* (29a) *should therefore be replaced by some approximation $\hat{a}_k$ such that*

$$\mathbb{E}\Big[G^{\hat{a}_k}(X_k) \bigvee \hat{V}_{n+1}(F^{\hat{a}_k}(X_k))\Big] \leq \min_{a\in\hat{\mathcal{A}}_k} \mathbb{E}\Big[G^a(X_k) \bigvee \hat{V}_{n+1}(F^a(X_k))\Big] + \gamma_k \tag{30}$$

*for some $\gamma_k \geq 0$ which takes into account some error on the optimal feedback control. Then an error analysis still holds, see Corollary 6.8, showing some robustness of the approach.*

**Hybrid scheme (or "H-scheme")** Let $(\hat{\mathcal{A}}_n)_{n\in[\![0,N-1]\!]}$ and $(\hat{\mathcal{V}}_n)_{n\in[\![0,N-1]\!]}$ be two given sequences of finite-dimensional spaces. Set $\hat{V}_N := g \vee \varphi$ as well as $\hat{V}_N^{[\text{tmp}]} := g \vee \varphi$. Then, for $n = N-1, \ldots, 0$:
- compute a feedback control $\hat{a}_n$ according to

$$\hat{a}_n \in \operatorname*{argmin}_{a\in\hat{\mathcal{A}}_n} \mathbb{E}\Big[G^a(X_n) \bigvee \hat{V}_{n+1}^{[\text{tmp}]}(F^a(X_n))\Big] \tag{31a}$$

- if $n \geq 1$, compute $\hat{V}_n^{[\mathrm{tmp}]} \in \hat{\mathcal{V}}_n$ (in prevision of the computation of $\hat{a}_{n-1}$), such that

$$\hat{V}_n^{[\mathrm{tmp}]} := \underset{V \in \hat{\mathcal{V}}_n}{\operatorname{argmin}} \mathbb{E}\Big[ \Big( V(X_n) - \hat{V}_n(X_n) \Big)^2 \Big], \tag{31b}$$

where $\hat{V}_n$ is such that

$$\hat{V}_n(x) := G^{\hat{a}_n}(x) \bigvee \hat{V}_{n+1}\big(F^{\hat{a}_n}(x)\big) \ \equiv J_n(x, (\hat{a}_n, \dots, \hat{a}_{N-1})). \tag{31c}$$

The sequence of controls $(\hat{a}_0, \dots, \hat{a}_{N-1})$ is the output of the algorithm, and $\hat{V}_n(x)$ can be recovered using (31c). At each iteration $1 \leq n \leq N$, $\hat{V}_n$ is projected on the space $\hat{\mathcal{V}}_n$, and its projection $\hat{V}_n^{[\mathrm{tmp}]}$ is used to compute $\hat{a}_{n-1}$. In this hybrid method, we still avoid some of the projection errors, by computing $\hat{V}_n$ from the feedback controls in (31c). Each evaluation of $\hat{V}_n$ costs $N-n$ evaluations of the control mappings, leading to an overall quadratic cost in $O(N^2)$. However, in the minimization procedure for (31a), we can directly access to the values of $\hat{V}_n^{[\mathrm{tmp}]}(\cdot)$, which is less costly than computing $\hat{V}_n(\cdot)$.

In the present work, only the convergence of the L-scheme is analyzed. However, the three proposed schemes are compared on several examples in the numerical Section, see in particular Sec. 8.1.

## 5.2 Runge Kutta schemes

In this section, we consider a particular class of Runge-Kutta (RK) schemes for the definition of $F_h(x,a)$ which corresponds to some approximation of the characteristics for a given control $a$.

For given $c = (c_i)_{1 \leq j \leq q}$, $B \in \mathbb{R}^{q \times q}$, let us denote

$$|c|_1 := \sum_{j=1}^{q} |c_i|, \quad \|B\|_\infty := \max_i \sum_j |b_{ij}|$$

and let also

$$C_f := |f(0,A)| := \max_{a \in A} |f(0,a)|.$$

**Definition 5.2** (Runge-Kutta scheme). *1. For a given $a \in A$, we say that $x \to F_h(x,a)$ is a Runge-Kutta scheme for $\dot{y} = f(y,a)$ with time step $h > 0$, if there exists $q \in \mathbb{N}^*$, $(b_{ij})_{i,j} \in \mathbb{R}^{q \times q}$ and $(c_i)_i \in \mathbb{R}^q$ such that*

$$\forall i \in [\![1, q]\!], \qquad y_i(x) = x + h \sum_{j=1}^{q} b_{ij} f(y_j(x), a)$$

*and*

$$F_h(x,a) = x + h \sum_{i=1}^{q} c_i f(y_i(x), a).$$

*2. The scheme is said to be* consistent *if $\sum_{i=1}^{q} c_i = 1$.*

*3. The scheme is said to be* explicit *if $b_{ij} = 0$ for all $j \geq i$.*

**Remark 5.3.** *(a) Note that a consistent RK scheme satisfies $F_h(x,a) = x + hf(x,a) + O(h^2)$ for $h$ sufficiently small. This is made precise in Lemma 5.6.*

*(b) Also, for any given value $a \in A$, $y_i := x + h \sum_{i=1}^{q} b_{ij} f(y_j, a)$ can be solved by a fixed point argument in $(\mathbb{R}^d)^q$, as soon as $h\|B\|_\infty [f]_1 < 1$. Hence for $h$ small enough such that $h\|B\|_\infty [f]_1 < 1$, the RK scheme is well defined. Explicit RK schemes are always well defined.*

*(c) Note that in the above definition, the value of the control $a_0 = a(x)$ is frozen at the foot of the characteristic. Hence we consider a RK approximation of $\dot{y}(t) = f(y(t), a(x))$ on $[t_n, t_{n+1}]$ with $y(t_n) = x$ and fixed $a(x)$, rather than an approximation of $\dot{y}(t) = f(y(t), a(y(t)))$.*

We now give some estimates for later use.

The following lemma gives an estimate between two trajectories led by different controls.

**Lemma 5.4.** *We assume (H0) and (H1). Let $F_h$ be a Runge-Kutta scheme, with $h > 0$ small enough such that $h\|B\|_\infty [f]_1 \le \frac{1}{2}$. Let $(a, \bar{a}) \in \mathcal{A}^2$ be any two given feedback controls. Then,*

*(i) For any $x, y \in \mathbb{R}^d \times \mathbb{R}^d$,*

$$|F_h(x, a(x)) - F_h(y, \bar{a}(y))| \le e^{2h|c|_1[f]_1}|x - y| + 2h|c|_1[f]_2|a(x) - \bar{a}(y)|. \tag{32}$$

*(ii) For all $0 \le j \le p$ and $x \in \mathbb{R}^d$,*

$$|(F_h^a)^{(j)}(x) - (F_h^{\bar{a}})^{(j)}(x)| \le C_F \Delta t |a(x) - \bar{a}(x)|, \tag{33}$$

*where $C_F$ is a constant independent of $\Delta t$ and such that*

$$2|c|_1[f]_2 e^{2\Delta t|c|_1[f]_1} \le C_F. \tag{34}$$

*We recall here that $(F_h^a)^{(j)}(x)$ corresponds to $j$ iterates of $y \to F_h(y, a(x))$ starting from $y = x$. In particular, denoting $F^a(x) = F(x, a(x))$, we have $F^a(x) = (F_h^a)^{(p)}(x)$ and therefore also*

$$|F^a(x) - F^{\bar{a}}(x)| \le C_F \Delta t |a(x) - \bar{a}(x)|. \tag{35}$$

*(iii) More generally,*

$$|F^a(x) - F^{\bar{a}}(y)| \le e^{C_1 \Delta t}(|x - y| + C_2 \Delta t |a(x) - \bar{a}(y)|) \tag{36}$$

*with constants $C_1 := 2|c|_1[f]_1$ and $C_2 := 2|c|_1[f]_2$.*

*Proof.* $(i)$ From the definitions, denoting $a_0 = a(x)$ and $\bar{a}_0 = \bar{a}(y)$, we have

$$\begin{aligned} \big|F_h(x, a_0) - F_h(y, \bar{a}_0)\big| &\le |x - y| + h \sum_{1 \le j \le q} |c_j||f(y_j^a(x), a_0) - f(y_j^{\bar{a}}(y), \bar{a}_0)| \\ &\le |x - y| + h|c|_1[f]_1\|Y^a - Y^{\bar{a}}\|_\infty + h|c|_1[f]_2|a_0 - \bar{a}_0| \end{aligned} \tag{37}$$

where the intermediate values of the RK schemes are denoted $Y^a = (y_j^a(x))_{1 \le j \le q}$ and $Y^{\bar{a}} = (y_j^{\bar{a}}(y))_{1 \le j \le q}$, and satisfy $Y^a = X + hBf(Y^a, a_0)$ and $Y^{\bar{a}} = Y + hBf(Y^{\bar{a}}, \bar{a}_0)$, where $X := (x, \dots, x)$ and $Y := (y, \dots, y)$. Hence we also have

$$\|Y^a - Y^{\bar{a}}\|_\infty \le |x - y| + h\|B\|_\infty[f]_1\|Y^a - Y^{\bar{a}}\|_\infty + h\|B\|_\infty[f]_2|a_0 - \bar{a}_0|,$$

from which we deduce, using the assumption of the present Lemma,

$$\|Y^a - Y^{\bar{a}}\|_\infty \le 2\big(|x - y| + h\|B\|_\infty[f]_2|a_0 - \bar{a}_0|\big).$$

Combining with (37), we obtain

$$\big|F_h(x, a_0) - F_h(y, \bar{a}_0)\big| \le e^{2h|c|_1[f]_1}|x - y| + 2h|c|_1[f]_2|a_0 - \bar{a}_0|.$$

$(ii)$ From the previous bound, denoting $e_j := \big|(F_h^{a_0})^{(j)}(x) - (F_h^{\bar{a}_0})^{(j)}(y)\big| \equiv \big|Y_{j,x}^{a_0} - Y_{j,y}^{\bar{a}_0}\big|$, we have

$$e_{j+1} \le e^{2h|c|_1[f]_1} e_j + 2h|c|_1[f]_2|a_0 - \bar{a}_0|,$$

and, by recursion,

$$e_j \le e^{2jh|c|_1[f]_1}(e_0 + 2jh|c|_1[f]_2|a_0 - \bar{a}_0|), \quad 0 \le j \le p. \tag{38}$$

This concludes to $(ii)$ by using $y = x$, $j = p$ and hence $jh = \Delta t$.

The proof of $(iii)$ is obtained from (38) with $j = p$. □

**Lemma 5.5.** *Assume (H0), (H1). Let $F_h$ be a RK scheme with $h>0$ such that $h\|B\|_\infty[f]_1 \le \frac{1}{2}$.*
*$(i)$ For $a \in A$ and $x \in \mathbb{R}^d$:*

$$|F_h(x,a)| \le |x| + 2h|c|_1(C_f \vee [f]_1)(|x|+1).$$

*$(ii)$ Let $a$ in $\mathcal{A}$ and denote $F_h^a(x) \coloneqq F_h(x,a(x))$, then it holds*

$$[F_h^a - i_d] \le 2|c|_1([f]_1 + [f]_2[a])\, h. \tag{39}$$

*Proof.* We start by proving $(ii)$. As in the proof of Lemma 5.4$(ii)$, with $\bar{a}=a$, we obtain

$$|(F^a(x,a(x)) - F^a(y,a(y))) - (x-y)| \le h|c|_1[f]_1(2|x-y| + 2h\|B\|_\infty[f]_2[a]|x-y|) + h|c|_1[f]_2[a]|x-y|.$$

Combining with the assumption that $2h\|B\|_\infty[f]_1 \le \frac{1}{2}$, we obtain the desired bound.

For the proof of $(i)$, for any $a$ in $A$ we have $[f]_a = [f]_1$ and $|F_h^a(x)| \le |F_h^a(0)| + [F_h^a]|x| \le |F_h^a(0)| + (1 + 2h|c|_1[f]_1])|x|$. By direct bounds we have also $|F_h^a(0)| \le 2h|c|_1 C_f$, from which we deduce the desired bound. □

**Lemma 5.6.** *Assume (H0), (H1), and that $F_h^a$ be a consistent RK scheme with $h|B|_\infty[f]_1 \le \frac{1}{2}$. Then $F_h$ is consistent with $f$ in the following sense:*

$$\exists C \ge 0,\ \exists h_0 > 0, \forall (x,a,h) \in \mathbb{R}^d \times A \times (0,h_0), \qquad |F_h(x,a) - (x + hf(x,a))| \le Ch^2(1+|x|) \tag{40}$$

*Proof.* We use $|f(y_j,a)| \le [f]_1|y_j| + C_f$ where $C_f = \max_{a\in A}|f(0,a)|$ to obtain in the RK scheme $|y_i - x| \le h\sum_j |b_{ij}|([f]_1|y_j| + C_f)$, hence

$$\max_i |y_i - x| \le h\|B\|_\infty[f]_1(\max_i |y_i|) + h\|B\|_\infty C_f. \tag{41}$$

By using the assumption we then get $\max_i |y_i| \le 2|x| + 2h\|B\|_\infty C_f$. Let $h \in ]0,h_0]$ with $h_0$ such that $h_0\|B\|_\infty[f]_1 = \frac{1}{2}$. Using (41), and the fact that $h\|B\|_\infty[f]_1 \le \frac{1}{2}$, we obtain

$$\max_i |y_i - x| \le h\|B\|_\infty[f]_1|x| + 2h\|B\|_\infty C_f = Ch(1+|x|) \tag{42}$$

for some constant $C \ge 0$. Then

$$\begin{aligned} F_h(x,a) &= x + h\sum_{j=1}^q c_j f(y_j,a) \;=\; x + h\sum_{j=1}^q c_j f(x + O(h(1+|x|)),a) \\ &= x + h\sum_{j=1}^q c_j\big(f(x,a) + O(h(1+|x|))\big) \\ &= x + hf(x,a) + O(h^2(1+|x|)) \end{aligned}$$

which is the desired result. □

## 5.3 Neural network spaces

Neural networks are functions build by compositions of other "simple" functions. They are widely used for their approximating capabilities, and are known to be dense in the class of continuous multivariate functions under mild hypotheses, see for instance Lemma 16.1 of [25]. We restrict ourselves to so-called *feedforward* neural networks, in the following sense.

**Definition 5.7** (Feedforward neural network). *Let $L \in \mathbb{N}^*$ (the* number of layers*), and $(d_k)_{k\in[\![0,L]\!]} \subset \mathbb{N}^*$ be a sequence of dimensions. A neural network is a function $\mathcal{R} : \mathbb{R}^{d_0} \mapsto \mathbb{R}^{d_L}$ of the form*

$$\mathcal{R}(x) = \sigma_L \circ \mathcal{L}_L \circ \cdots \circ \sigma_1 \circ \mathcal{L}_1(x)$$

*where $\mathcal{L}_k : \mathbb{R}^{d_{k-1}} \mapsto \mathbb{R}^{d_k}$ is an affine transformation, $\sigma_k : \mathbb{R}^{d_k} \mapsto \mathbb{R}^{d_k}$ is a nonlinear* activation *function which acts coordinate by coordinate:*

$$\sigma_k(x) = (\bar\sigma_k(x_1), \bar\sigma_k(x_2), \ldots, \bar\sigma_k(x_{d_k}))$$

*for $x \in \mathbb{R}^{d_k}$ and for a certain $\bar\sigma_k : \mathbb{R} \mapsto \mathbb{R}$.*

Each affine transformation is represented by a *weight* matrix $\omega_k \in \mathbb{R}^{d_k\times(d_{k-1}+1)}$, with $\mathcal{L}_k(x) = \omega_k \begin{pmatrix} x \\ 1 \end{pmatrix}$. Classical examples of activation functions include the sigmoid function $\bar\sigma(x) = \frac{1}{1+e^{-x}}$ or the rectified linear unit (ReLU) $\bar\sigma(x) = \max(0, x)$. The last activation function, $\bar\sigma_L$, may be set to the identity function, or $\sigma_L$ may be a more complex function, so that $\sigma_L(\cdot) \in A$ for the control approximation, depending on the example.

We may now define the sets $\hat{\mathcal{A}}$ and $\hat{\mathcal{V}}$ as (recall that $A \subset \mathbb{R}^\kappa$)

$$\hat{\mathcal{A}} \coloneqq \{\text{feedforward neural networks with } d_0 = d \text{ and } d_L = \kappa\},$$
$$\hat{\mathcal{V}} \coloneqq \{\text{feedforward neural networks with } d_0 = d \text{ and } d_L = 1\}.$$

In our numerical examples, we always choose ReLU for the inner layer activation functions. The last activation function $\sigma_L$ may vary to fit the definition of $A$ for the given problem (see the numerical section). We also choose to set $d_1 = \cdots = d_{L-1} =: N_e$, i.e., the same *number of neurons* for each layer, to simplify the set of parameters.

# 6 Main result

In this section, we focus on proving the convergence of the Lagrangian scheme (29). This algorithm only uses approximations of feedback controls.

Since our estimates are valid on Lipschitz continuous controls, and since the exact optimal solution in general does not involve such regular controls, we first introduce $\eta$-weak approximations as follows.

**Definition 6.1.** *For a given sequence $\eta = (\eta_n, \eta_{n+1}, \ldots, \eta_{N-1}) \in (\mathbb{R}_+^*)^{N-n}$, we say $\hat a = (\hat a_n, \ldots, \hat a_{N-1}) \in \mathcal{A}^{N-n}$ is an $\eta$-weak approximation of $(V_k)_{n\le k\le N}$ if*
*(i) $(\hat a_n, \ldots, \hat a_{N-1})$ are Lipschitz continuous controls,*
*(ii) $\left|\mathbb{E}[V_k(X_k)] - \mathbb{E}[G^{\hat a_k}(X_k) \vee V_{k+1}(F^{\hat a_k}(X_k))]\right| \le \eta_k$, for all $k \in [\![n, N-1]\!]$.*

Notice that by using Prop. 4.1, it is possible to construct $\eta$-weak approximations. By recursion, it is furthermore possible to construct the controls such that, for all $k \in [\![n, N-1]\!]$

$$\left|\mathbb{E}[V_k(X_k)] - \mathbb{E}[G^{\hat a_k}(X_k) \vee V_{k+1}(F^{\hat a_k}(X_k))]\right| \le \eta_k([\bar a_n^{\tau_n}], [\bar a_{n+1}^{\tau_{n+1}}], \ldots, [\bar a_{k-1}^{\tau_{k-1}}]), \tag{43}$$

for given strictly positive functions $\eta_k$ (i.e., $\eta_k > 0$, $\eta_{k+1}(x_k) > 0$, ..., $\eta_{N-1}(x_k, \ldots, x_{N-2}) > 0$).

We now state our last assumption that is needed on the approximate dynamics $F_h$.

**(H6)** *There exist a constant $c > 0$, for any Lipschitz continuous function $a(\cdot) \in \mathcal{A}$, for $h > 0$ small enough, the map $x \to F_h(x, a(x))$ is Lipschitz continuous, with Lipschitz bound*

$$[F_h(\cdot, a(\cdot)) - i_d] \le c\,([f]_1 + [f]_2[a])\,h \tag{44}$$

*(where $i_d(x) \coloneqq x$).*

Hereafter we also use the notation, for any Lipschitz continuous control $a \in \mathcal{A}$:

$$[f]_a \coloneqq [f]_1 + [f]_2[a].$$

In particular it is easy to see that $[f^a]$, the Lipschitz constant of $f^a(x) = f(x, a(x))$, is bounded by $[f]_a$.

Note also that if $\max_{a\in A} |F(0,a)| \le Ch$ for some constant $C$, then (H6) implies the bound (24) of (H5).

Assumption (H6) is needed in order to use a change of variables formula for $y = F^a(x)$, corresponding to $p$ iterates of $y \to F_h(y, a(x))$ starting from $y = x$.

**Remark 6.2.** *Lemma 5.5 shows that (H5)-(H6) are satisfied for Runge Kutta schemes as defined in 5.2. For instance, dynamics $F_h$ such as the Euler scheme $F_h(x,a) = x + hf(x,a)$, or the Heun scheme $F_h(x,a) = x + \frac{h}{2}(f(x,a) + f(x + hf(x,a), a))$ satisfy (H5)-(H6). Higher order schemes, or implicit schemes, could be used as well.*

**Remark 6.3.** *Assumption (H6) is satisfied by the exact characteristics. Indeed, for any regular control $a \in \mathcal{A}$, the map $x \to F_h^a(x) \coloneqq y_x^a(h)$ (where $y(s) = y_x^a(s)$ is the solution of $\dot{y}(s) = f^a(y(s))$ with $y(0) = x$) is one-to-one and onto on $\mathbb{R}^d$ and satisfies $DF_h^a(x) = \exp B_h$ with $B_h = \int_0^h Df^a(y_x^a(s))ds$. Then denoting $\|DF_h^a\|_\infty = \sup_{x\in\mathbb{R}^d} \|DF_h^a(x)\|_\infty$, we have $[F_h^a - i_d] = \|DF_h^a - I\|_\infty \le e^{\|B_h\|_\infty} - 1 \le 2\|B_h\|_\infty$ as soon as for instance $\|B_h\|_\infty \le \frac{1}{2}$, with also $\|B_h\|_\infty \le h\|Df^a\|_\infty \le h[f^a] \le h[f]_a$, hence (44) holds true with $c = 2$ and $h > 0$. When $a$ is only Lipschitz regular, the same bound is obtained by a regularization argument.*

**Lemma 6.4.** *Assumption (44) implies the following bound:*

$$\|DF_h^a(x)\|_\infty \le [F_h^a] \le 1 + c[f]_a h \le e^{c[f]_a h}, \quad a.e.\ x \in \mathbb{R}^d. \tag{45}$$

*From this estimate we can deduce*

$$|\det(DF_h^a(x))| \quad \le \quad e^{dc[f]_a h},\ a.e.\ x \in \mathbb{R}^d. \tag{46}$$

*Furthermore, if $c[f]_a h \le \frac{1}{2}$ then $x \to F_a^h(x)$ is a change of variable, i.e., it is one-to-one and onto on $\mathbb{R}^d$, it inverse mapping $G_h^a \coloneqq (F_a^h)^{-1}$ is well defined, Lipschitz continous and satisfies the following estimates*

$$\|DG_h^a(x)\|_\infty \le 1 + 2c[f]_a h \le e^{2c[f]_a h}, \quad a.e.\ x \in \mathbb{R}^d \tag{47}$$

*and*

$$|\det(DG_h^a(x))| \quad \le \quad e^{2dc[f]_a h},\ a.e.\ x \in \mathbb{R}^d. \tag{48}$$

*Proof of Lemma 6.4.* Estimates (45) and (46) are straightforward. In order to show that $x \to y = F_h^a(x)$ is invertible, notice that $F_h^a(x) = x + \Phi(x)$ where, from the assumptions, $[\Phi] \le c[f]_a h$. As soon as $[\Phi] < 1$ we can see that the map $x \to y - \Phi(x)$ is contractant on $\mathbb{R}^d$, and by the fixed-point theorem we obtain that $F_h^a$ is invertible. Furthermore, if $c[f]_a h < 1$, then from the fact that $|(F_h^a(x) - F_h^a(y)) - (x - y)| \le c[f]_a h|x - y|$, we get $|x - y| \le |F_h^a(x) - F_h^a(y)| \frac{1}{1 - c[f]_a h}$ hence the inverse $G_h^a = (F_h^a)^{-1}$ satisfies the bound $[G_h^a] \le \frac{1}{1 - c[f]_a h}$. If $0 \le x \le \frac{1}{2}$ then we use the (rough) bound $\frac{1}{1-x} \le 1 + \frac{3}{2}x \le 1 + 2x$ which leads to $[G_h^a] \le 1 + 2c[f]_a h \le e^{2c[f]_a h}$. □

**Remark 6.5.** *In [29], the discrete dynamic takes the form of a stochastic process* $(X_k)_k$ *whose evolution is given by a transition kernel* $P^a(x, dx')$*, in the sense that* $\mathbb{P}_{X^a_{k+1}}(\cdot) = \int P^a(x,\cdot)d\mathbb{P}_{X_k}(x)$*. It is assumed that the transition kernel admits a density with respect to a fixed measure* $\mu$*, called the* training measure*, i.e.* $P^a(x, dx') = r(x, a; x')d\mu(x')$*. In the deterministic setting, such a training distribution* $\mu$ *would be a combination of Dirac masses. But then, enforcing the same assumption would lead to unrealistic situations, where the discrete dynamic* $x + \Delta t f(x, a) \to x'$ *would only be allowed to explore the support of* $\mu$*. Instead, assumption (H6) is used in the change of variable formula (Lemma 7.1) in order to get a recursive error bound estimate.*

In the following, we recall that $(\hat{V}_k)_{0\le k\le N}$ corresponds to the Lagrangian scheme (29). Also we have $\hat{V}_k(x) \ge V_k(x)$, and therefore we look for an upper bound of $\hat{V}_k(x) - V_k(x)$.

We introduce a notation for neural network sets that have a given Lipschitz constant bound:

$$\hat{\mathcal{A}}_{k,L} := \{a \in \hat{\mathcal{A}}_k,\ [a] \le L\}. \tag{49}$$

**Remark 6.6.** *Notice that Group Sort neural networks satisfies, for any* $\bar{a}$ *Lipschitz continuous and* $\Omega_n$ *bounded set (see [5, 43]):*

$$\lim_{\Theta\to\infty} \inf_{a\in\hat{\mathcal{A}}^{\Theta}_{n,[\bar{a}]}} \mathbb{E}[|a(X_n) - \bar{a}(X_n)|] = 0. \tag{50}$$

**Theorem 6.7.** *Assume (H0)-(H6),* $N \ge 1$*, and let* $n \in [\![0, N]\!]$*. For* $k = n, \dots, N-1$*, let* $\eta_k : \mathbb{R}^{n-k} \to \mathbb{R}^*_+$ *be given functions (*$\eta_n > 0$ *is a constant,* $\eta_{n+1} > 0$ *is a function of one variable, and so on). Let* $\bar{a} = (\bar{a}_0, \dots, \bar{a}_{N-1})$ *be an* $\eta$*-weak approximation in the sense of* (43)*, and* $L := \max([\bar{a}_0], \dots, [\bar{a}_{N-1}])$*. Assume also that* $p \ge 1$ *is large enough such that*

$$\sup_k \sup_{a\in\hat{\mathcal{A}}_{k,[\bar{a}_k]}} c[f]_a \frac{\Delta t}{p} \le \frac{1}{2} \tag{51}$$

*and* $c \ge 0$ *is as in (H6). Then*

$$\mathbb{E}\left[(\hat{V}_n - V_n)(X_n)\right] \le \inf_{(a_n\dots,a_{N-1})\in\bigotimes_{k=n}^{N-1}\hat{\mathcal{A}}_{k,[\bar{a}_k]}} \sum_{k=n,\dots,N-1} \Big(\prod_{i=n,\dots,k-1} C_i^{a_i}\Big)(\varepsilon_k^{a_k} + \eta_k([\bar{a}_n, \dots, \bar{a}_{k-1}]))$$

*where* $C^a_k \coloneqq C_{k,\Delta t} e^{2dc[f]_a\Delta t}$ *with* $C_{k,\Delta t}$ *is as in (H4),*

$$\varepsilon^a_k \coloneqq C_F([g] + [V_{k+1}])\,\Delta t\,\mathbb{E}_k\Big[|a(X_k) - \bar{a}_k(X_k)|\Big], \tag{52}$$

$[V_{k+1}]$ *is bounded as in Lemma 2.1,* $C_F$ *satisfies* (34).

Note that the consistency of the scheme (with respect to the dynamics $f$, as in (10)) is not needed in the previous Theorem, because the result only focuses on the error between the semi-discrete problem and its approximation by a Lagrangian scheme.

Note also that for condition (51) to hold it is sufficient to have $c([f]_1 + L[f]_2)\frac{\Delta t}{p} \le \frac{1}{2}$.

**Corollary 6.8.** *In the same way, for the perturbed algorithm* (30) *the same error bound holds where each term* $(\varepsilon^{a_k}_k + \eta_k)$ *is replaced by* $(\varepsilon^{a_k}_k + \eta_k + \gamma_k)$*.*

The proof of Theorem 6.7 is postponed to Section 7 (the proof of Corollary 6.8 follows exactly the same lines). We now give two corollaries of the previous theorem.

**Corollary 6.9.** *Assume (H0)-(H6), and $N \geq 1$. Let $\hat{\mathcal{A}}^{\Theta}_{n,L}$ denote the control approximation space at time $t_n$, with explicit dependency over the size $\Theta$ and the Lipschitz bound $L$. We denote by $\Theta \to \infty$ the limit when some parameters go to infinity (for instance the number of neurons of a neural network). We assume that for any $n = 0, \dots, N-1$, any Lipschitz continuous function $\bar{a} \in \mathcal{A}$ can be approximated by some function of $a \in \hat{\mathcal{A}}^{\Theta}_{n,[\bar{a}]}$ up to any arbitrary precision, which we write as*

$$\lim_{\Theta\to\infty} \inf_{a\in\hat{\mathcal{A}}^{\Theta}_{n,[\bar{a}]}} \mathbb{E}[|a(X_n) - \bar{a}(X_n)|] = 0. \tag{53}$$

*Let $(\hat{V}^{\Theta}_n)$ be the corresponding L-scheme values associated with sets $(\mathcal{A}^{\Theta_n}_{n,[\bar{a}_n]}, \mathcal{A}^{\Theta_{n+1}}_{n+1,[\bar{a}_{n+1}]}, \dots, \mathcal{A}^{\Theta_{N-1}}_{N-1,[\bar{a}_{N-1}]})$. Then*

$$\lim_{\Theta\to\infty} \max_{0\leq n\leq N} \mathbb{E}[\hat{V}^{\Theta}_n(X_n) - V_n(X_n)] = 0.$$

*(where $\Theta \to \infty$ means here that $\Theta_k \to \infty$ for all $k = n, \dots, N-1$).*

*Proof of Corollary 6.9.* Let $\varepsilon > 0$. Let $\eta_n := \varepsilon/(2N)$, $\eta_{n+1}([\bar{a}_n]) := \varepsilon/(2NC^{\bar{a}_n}_n)$, $\dots$, and $\eta_{N-1}([\bar{a}_n], \dots, [\bar{a}_{N-2}]) := \varepsilon/(2NC^{\bar{a}_n}_n C^{\bar{a}_{n+1}}_{n+1} \cdots C^{\bar{a}_{N-2}}_{N-2})$. Let $L := \max([\bar{a}_n], \dots, [\bar{a}_{N-1}])$.

By assumption (53),

$$\forall k = n, \dots, N-1, \quad \lim_{\Theta\to\infty} \inf_{a_k\in\hat{\mathcal{A}}^{\Theta}_{k,L}} \mathbb{E}_{k+1}[|a_k - \hat{a}_k|] = 0$$

and therefore $\lim_{\Theta\to\infty} \inf_{a_k\in\hat{\mathcal{A}}^{\Theta}_{k,L}} \varepsilon^{a_k,[\bar{a}_k]}_k = 0$, where $\varepsilon^a_k$ is defined in (52). Hence we can find $a_n \in \hat{\mathcal{A}}^{\Theta_n}_{n,[\bar{a}_n]}$ (for $\Theta_n$, the size of $\hat{\mathcal{A}}_n$, large enough) such that $\varepsilon^{a_n}_n \leq \frac{\varepsilon}{2N}$, and we have

$$\varepsilon^{a_n}_n + \eta_n \leq \frac{\varepsilon}{N}.$$

Then there exists $a_{n+1} \in \hat{\mathcal{A}}^{\Theta_{n+1}}_{n+1,[\bar{a}_{n+1}]}$ (for $\Theta_{n+1}$ large enough) such that

$$C^{a_n}_n(\varepsilon^{a_{n+1}}_{n+1} + \eta_{n+1}([\bar{a}_n])) \leq \frac{\varepsilon}{N},$$

and so on, until we choose $a_{N-1} \in \hat{\mathcal{A}}^{\Theta_{N-1}}_{N-1,[\bar{a}_{N-1}]}$ such that

$$C^{a_n}_n C^{a_{n+1}}_{n+1} \cdots C^{a_{N-2}}_{N-2} \big(\varepsilon^{a_{N-1}}_{N-1} + \eta_{N-1}([\bar{a}_n], \dots, [\bar{a}_{N-2}])\big) \leq \frac{\varepsilon}{N}$$

(we have $N - n$ such bounds). By using the bound of Theorem 6.7, the sum of all error terms is bounded by $(N-n)\frac{\varepsilon}{N}$, and therefore

$$\mathbb{E}[\hat{V}_n(X_n) - V_n(X_n)] \leq \varepsilon.$$

This shows that $\lim_{\Theta\to\infty} \mathbb{E}[\hat{V}^{\Theta}_n(X_n) - V_n(X_n)] = 0$. The desired result follows since we have only a finite number $N$ of such terms. $\square$

Notice that in the previous result $N \geq 1$ is given. This does not give in general a convergence result as $N \to \infty$, because of the uncontrolled Lipschitz constants that appear in the bounds of Theorem 6.7.

However, in the case the optimal controls $(\bar{a}_n)_n$ can be shown to be Lipschitz continuous with a uniform Lipschitz constant, we may improve the result. We suppose that the numerical feedback space $\hat{\mathcal{A}}$ can be restricted to Lipschitz functions with a controlled Lipschitz constant. For instance, if $\hat{\mathcal{A}}$ is a neural network space, one could choose the GroupSort activation function and bound the weights to obtain this estimate (see [5, 43]).

**Corollary 6.10.** *Assume (H0)-(H6), and that there exists a sequence of optimal feedback control (denoted $\bar{a}$) which are Lipschitz continuous: there exists $L \geq 0$ such that for any $N \geq 1$ and $0 \leq k \leq N-1$, $[\bar{a}_k] \leq L$. Assume also $N \geq 1$, $\Delta t = \frac{T}{N}$ and $p$ large enough such that $c([f]_1 + L[f]_2)\frac{\Delta t}{p} \leq \frac{1}{2}$. Then it holds*

$$\max_{0\leq n<N} \mathbb{E}[\hat{V}_n(X_n) - V_n(X_n)]$$
$$\leq \; K_N \inf_{(a_0\ldots,a_{N-1})\in\bigotimes_{k=0}^{N-1}\hat{\mathcal{A}}_{k,[\bar{a}_k]}} \left(\Delta t \sum_{k=0}^{N-1} \mathbb{E}_k\big[|a_k(X_k) - \bar{a}_k(X_k)|\big]\right)$$

*where*

$$K_N := 2C_F([g]\vee[\varphi])e^{(d+1)c([f]_1+L[f]_2)T} \max_{0\leq n<N} \prod_{k=n}^{N-1} C_{k,\Delta t}. \tag{54}$$

*Furthermore, in the case of uniform densities, we can use the estimate* (19) *to deduce a bound for $K_N$ which is independent of $N$ (other situations could also lead to a uniform bound for $K_N$).*

*Proof of Corollary 6.10.* We make use of the bound of Theorem 6.7 with $\eta_k = 0$, $\forall k$. Notice that $[f^{\bar{a}_k}] \leq [f]_1 + L[f]_2$, and also, with $[a_k] \leq [\bar{a}_k] \leq L$, we have $[f^{a_k}] \leq [f]_1 + L[f]_2$. Then

$$\prod_{n\leq k\leq N-1} C^{\bar{a}_k} \leq \Big(\prod_{n\leq k\leq N-1} C_{k,\Delta t}\Big)e^{dc\sum_{k=n}^{N-1}[f^{\bar{a}_k}]\Delta t} \leq \Big(\prod_{n\leq k\leq N-1} C_{k,\Delta t}\Big)e^{dc([f]_1+L[f]_2)T}$$

as well as $[V_{k+1}] \leq ([g]\vee[\varphi])e^{c([f]_1+L[f]_2)T}$ and therefore (using also $[g] \leq [g]\vee[\varphi]$)

$$\varepsilon_k^{a_k} \leq 2C_F([g]\vee[\varphi])e^{c([f]_1+L[f]_2)T}\Delta t \; \mathbb{E}_k\big[|a_k(X_k) - \bar{a}_k(X_k)|\big].$$

The desired result follows. □

# 7 Proof of Theorem 6.7

We first state a change of variable Lemma, giving a statement for either an exact characteristic ($x \to y^a_{t,x}$) or for an approximate one ($x \to F^a(x)$). Only the second statement is used in the convergence analysis.

**Lemma 7.1** (Change of variable)**.** *Let $a : \mathbb{R}^d \to \mathbb{R}^\kappa$ be a given Lipschitz continuous function.*

1. *Let $t \to y^a_{t,x}$ denotes the characteristic associated with dynamics $x \to f^a(x)$ and such that $y^a_{0,x} = x$. We assume the following analogue of (H4) in the continuous case:*

$$y^a_{\Delta t,\Omega_k} \subset \Omega_{k+1},\ \forall k = 0,\ldots,N-1,$$

*and*

$$\widetilde{C}_{k,\Delta t} := \max_{0\leq k\leq N-1} \sup_{x\in\Omega_k} \frac{\rho_k(x)}{\rho_{k+1}(y^a_{\Delta t,x})} < \infty.$$

*Then for any non-negative measurable function $\Phi : \Omega_{k+1} \to \mathbb{R}$,*

$$\mathbb{E}_k[\Phi(y^a_{\Delta t,X_k})] \leq \widetilde{C}_{k,\Delta t}\, e^{d[f]_a\Delta t}\; \mathbb{E}_{k+1}[\Phi(X_{k+1})]. \tag{55}$$

2. *Suppose (H4), (H5) and (H6). Assume $p \geq 1$ is such that $ch[f]_a \leq \frac{1}{2}$, where $h = \frac{\Delta t}{p}$ and $c \geq 0$ is as in (H6). Then for any non-negative measurable function $\Phi : \Omega_{k+1} \to \mathbb{R}$,*

$$\mathbb{E}_k[\Phi(F^a(X_k))] \leq C_{k,\Delta t}\, e^{2dc[f]_a\Delta t}\; \mathbb{E}_{k+1}[\Phi(X_{k+1})] \tag{56}$$

*with $C_{k,\Delta t}$ is as in (H4).*

*Proof of Lemma 7.1.* $(i)$ Let $y_{t,x}$ be the solution at time $t$ of the differential equation $\dot{y}_{t,x} = f^a(y_{t,x})$ for $t \in \mathbb{R}$, with $y_{0,x} = x$. Then for any function $\Phi \geq 0$ and $t \geq 0$, using the change of variable $x \to x' = y_{t,x}$ on $\Omega_k$ (with inverse function $x = y_{-t,x'}$):

$$\mathbb{E}_k[\Phi(y_{t,X_k})] = \int_{\Omega_k} \Phi(y_{t,x})\rho_k(x)\, dx = \int_{y_{t,\Omega_k}} \Phi(x')\frac{\rho_k(y_{-t,x'})}{J(x)}\, dx'$$

where $J(x)$ denotes the Jacobian of $x \to x' = y_{t,x}$ at point $x = y_{-t,x'}$. By Remark 6.3 we have $J(x) = \det(DF^a_t(x))$ with $DF^a_t(x) = \exp(B_t)$ and with the bounds $\|DF^a_t(x)\|_\infty \leq \exp(\|B_t\|_\infty) \leq \exp([f]_a t)$. Hence $J(x) \leq e^{d[f]_a t}$. Here also $1/J(x)$ is the Jacobian of the inverse mapping which satisfies similar estimates, therefore $1/J(x) \leq e^{d[f]_a t}$ and we have

$$\mathbb{E}_k[\Phi(y_{t,X_k})] \leq e^{d[f]_a t} \int_{y_{t,\Omega_k}} \Phi(x')\frac{\rho_k(y_{-t,x'})}{\rho_{k+1}(x')}\rho_{k+1}(x')dx'. \quad (57)$$

Let $t = \Delta t$, we have $y^a_{t,\Omega_k} \subset \Omega_{k+1}$ by assumption (18b). Also, for any $x' = y^a_{\Delta t,x} \in y^a_{\Delta t,\Omega_k}$, we have $\rho_k(y^a_{-\Delta t,x'})/\rho_{k+1}(x') = \rho_k(x)/\rho_{k+1}(y^a_{\Delta t,x}) \leq \widetilde{C}_{k,\Delta t}$ by assumption (18c). Together with (57) this allows to conclude to the desired bound.

$(ii)$ The proof is similar to $(i)$. We use the assumption $c[f]_a h \leq \frac{1}{2}$ in order to get that $x \to F^a_h(x)$ is a change of variable, and, by Lemma 6.4, the bound $1/J_{F^a_h}(x) \leq e^{2dc[f]_a h}$ (where $J_F(x) = \det(DF(x))$). Then $F^a := (F^a_h)^{(p)}$ is also a change of variable and satisfies the bound $1/J_{F^a}(x) \leq e^{2dc[f]_a hp} = e^{2dc[f]_a \Delta t}$. We conclude as in $(i)$. □

We are now in position to prove the main result.

*Proof of Theorem 6.7.* Our aim is to bound recursively the quantity

$$e_n := \mathbb{E}[\hat{V}_n(X_n) - V_n(X_n)].$$

Let $\eta_n > 0$. By Prop. 4.1, there exists $\bar{a}_n \in \mathcal{A}_n$, Lipschitz continuous, such that

$$\left|\mathbb{E}[V_n(X_n)] - \mathbb{E}[G^{\bar{a}_n} \vee V_{n+1}(F^{\bar{a}_n}(X_n))]\right| \leq \eta_n.$$

Recall that $\hat{V}_n$ satisfies

$$\mathbb{E}\left[\hat{V}_n(X_n)\right] = \inf_{a \in \hat{\mathcal{A}}_n} \mathbb{E}\left[G^a(X_n) \bigvee \hat{V}_{n+1}(F^a(X_n))\right],$$

which leads in particular to

$$\mathbb{E}\left[\hat{V}_n(X_n)\right] \leq \inf_{a \in \hat{\mathcal{A}}_{n,[\bar{a}_n]}} \mathbb{E}\left[G^a(X_n) \bigvee \hat{V}_{n+1}(F^a(X_n))\right].$$

Hence

$$\mathbb{E}\left[\hat{V}_n(X_n) - V_n(X_n)\right] \leq \inf_{a \in \hat{\mathcal{A}}_{n,[\bar{a}_n]}} \mathbb{E}\left[G^a(X_n) \bigvee \hat{V}_{n+1}(F^a(X_n)) - G^{\bar{a}_n}(X_n) \bigvee V_{n+1}(F^{\bar{a}_n}(X_n))\right] + \eta_n. \quad (58)$$

Using $\max(a,b) - \max(c,d) \leq \max(a-c, b-d)$, we have

$$\begin{aligned} G^a(x) \bigvee \hat{V}_{n+1}(F^a(x)) - G^{\bar{a}_n}(x) \bigvee V_{n+1}(F^{\bar{a}_n}(x)) &\leq \max_{0 \leq j < p}(g(Y^a_{j,x}) - g(Y^{\bar{a}_n}_{j,x})) \bigvee \left(\hat{V}_{n+1}(F^a(x)) - V_{n+1}(F^{\bar{a}_n}(x))\right) \\ &\leq \left(\max_{0 \leq j < p}[g]|Y^a_{j,x} - Y^{\bar{a}_n}_{j,x}|\right) \bigvee \left(\hat{V}_{n+1}(F^a(x)) - V_{n+1}(F^{\bar{a}_n}(x))\right). \end{aligned}$$

In order to bound the r.h.s of this last equation, we first use the following bound:

$$\begin{aligned}
&\hat{V}_{n+1}(F^a(x)) - V_{n+1}(F^{\bar{a}_n}(x)) \\
&\quad = \Big(\hat{V}_{n+1}(F^a(x)) - V_{n+1}(F^a(x))\Big) + \Big(V_{n+1}(F^a(x)) - V_{n+1}(F^{\bar{a}_n}(x))\Big) \\
&\quad \le \Big(\hat{V}_{n+1}(F^a(x)) - V_{n+1}(F^a(x))\Big) + [V_{n+1}]|F^a(x) - F^{\bar{a}_n}(x)|. \qquad (59)
\end{aligned}$$

By using estimates (58) and (59) we deduce

$$\begin{aligned}
&\mathbb{E}\left[\hat{V}_n(X_n) - V_n(X_n)\right] \\
&\le \inf_{a\in\hat{\mathcal{A}}_{n,[\bar{a}_n]}} \Bigg([g]\ \mathbb{E}\Big[\max_{0\le j\le p}\Big|Y^a_{j,X_n} - Y^{\bar{a}_n}_{j,X_n}\Big|\Big] \\
&\qquad + \mathbb{E}\big[(\hat{V}_{n+1} - V_{n+1})(F^a(X_n))\big] + [V_{n+1}]\ \mathbb{E}\big[|F^a(X_n) - F^{\bar{a}_n}(X_n)|\big]\Bigg) + \eta_n \\
&\le \inf_{a\in\hat{\mathcal{A}}_{n,[\bar{a}_n]}} \Bigg(\mathbb{E}\big[(\hat{V}_{n+1} - V_{n+1})(F^a(X_n))\big] + C_F([g] + [V_{n+1}])\Delta t\mathbb{E}\big[|a(X_n) - \bar{a}_n(X_n)|\big]\Bigg) + \eta_n
\end{aligned}$$

where the estimate of Lemma 5.4$(ii)$ has been used for the last inequality.

Then by using the change of variable Lemma 7.1, we obtain

$$\begin{aligned}
e_n &\le \inf_{a\in\hat{\mathcal{A}}_{n,[\bar{a}_n]}} \Bigg(C_{n,\Delta t}e^{2dc[f]_a\Delta t}\mathbb{E}\big[(\hat{V}_{n+1} - V_{n+1})(X_{n+1})\big] + C_F([g] + [V_{n+1}])\Delta t\mathbb{E}\big[|a(X_n) - \bar{a}_n(X_n)|\big]\Bigg) + \eta_n \\
&\le \inf_{a\in\hat{\mathcal{A}}_{n,[\bar{a}_n]}} C^a_n e_{n+1} + \big(\varepsilon^a_n + \eta_n\big)
\end{aligned}$$

where $\varepsilon^a_n := C_F([g] + [V_{n+1}])\,\Delta t\,\mathbb{E}\Big[|a(X_n) - \bar{a}_n(X_n)|\Big]$ and $C^a_n := C_{n,\Delta t}e^{2dc[f]_a\Delta t}$. By induction, and using the fact that $e_N = 0$ because $\hat{V}_N = V_N$, we obtain (for given coefficients $\eta_k > 0$, $k = n,\dots,N-1$):

$$e_n \le \inf_{(a_n,\dots,a_{N-1})\in\bigotimes_{k=n}^{N-1}\hat{\mathcal{A}}_{k,[\bar{a}_k]}} \big[(\varepsilon^{a_n}_n + \eta_n) + C^{a_n}_n(\varepsilon^{a_{n+1}}_{n+1} + \eta_{n+1}) + \cdots + C^{a_n}_n \cdots C^{a_{N-2}}_{N-2}(\varepsilon^{a_{N-1}}_{N-1} + \eta_{N-1})\big]\,.$$

However, we can improve this bound. We can chose a coefficient $\eta_{n+1} = \eta_{n+1}([\bar{a}_n]) > 0$ (which may have a dependency over $[\bar{a}_n]$), and proceed in the same way. By Prop. 4.1, there exists $\bar{a}_{n+1}\in\mathcal{A}$, Lipschitz continuous, such that

$$\big|\mathbb{E}[V_{n+1}(X_n)] - \mathbb{E}[G^{\bar{a}_{n+1}} \vee V_{n+2}(F^{\bar{a}_{n+1}}(X_{n+1}))]\big| \le \eta_{n+1}([\bar{a}_n]).$$

Then we obtain the bound

$$e_n \le \inf_{(a_n,a_{n+1})\in\hat{\mathcal{A}}_{n,[\bar{a}_n]}\times\hat{\mathcal{A}}_{n+1,[\bar{a}_{n+1}]}} \Bigg(\big(\varepsilon^{a_n}_n + \eta_n\big) + C^{a_n}_n\big(\varepsilon^{a_{n+1}}_{n+1} + \eta_{n+1}([\bar{a}_n])\big) + C^{a_n}_n C^{a_{n+1}}_{n+1} e_{n+2}\Bigg).$$

At the next step, we can chose a coefficient $\eta_{n+2} = \eta_{n+2}([\bar{a}_n],[\bar{a}_{n+1}])$, and so on. By induction, we get the desired bound. □

# 8 Numerical results

In the following $d$-dimensional examples (where $d \ge 2$), two dimensional "local" and "global" errors are computed in the following way. Depending on the example, a two-dimensional plane of reference $\mathcal{P} = \text{Span}(w_1, w_2)$ is set, passing through the origin, where $w_1, w_2$ are chosen vectors of $\mathbb{R}^d$. A uniform grid mesh $x_k = a_i w_1 + b_j w_2$, for $k = (i, j)$, $|a_i|, |b_j| \leq R_{\max}$ for a given $R_{\max} > 0$, is chosen in the plane $\mathcal{P}$ in order to compute the exact solution and to compare with the numerical solution.

Given a threshold $\eta > 0$, the errors are computed at the last iteration by

$$e^{\eta}_{L^1_\eta} := \frac{\sum_{\{x_i \in \Omega_\eta\}} |v(0, x_i) - \hat{V}_0(x_i)|}{\sum_{\{x_i \in \Omega_\eta\}} 1} \quad \text{and} \quad e^{\eta}_{L^\infty_{\text{loc}}} := \max_{\{x_i \in \Omega_\eta\}} |v(0, x_i) - \hat{V}_0(x_i)|$$

where $v(0, \cdot)$ is the analytical solution at time $t = 0$, $\hat{V}_0$ is its approximation by the scheme used, $\Omega_\eta := \{x \in \Omega, |v(0, x)| \leq \eta\}$, and $\Omega$ is the bounded computational domain. Notice that

$$e_{L^1_\eta} \simeq \frac{\|v(0, .)\|_{L^1(\Omega_\eta)}}{\|1\|_{L^1(\Omega_\eta)}}, \quad e_{L^\infty_\eta} \simeq \max_{x \in \Omega_\eta} |v(0, x)|.$$

The global errors, corresponding to the case $\eta = +\infty$, are denoted $e_{L^1}$ and $e_{L^\infty}$ (i.e., $\Omega_\eta = \Omega$). Unless otherwise stated, the local errors are computed with $\eta = 0.1$, and denoted $e_{L^1_{\text{loc}}}$ and $e_{L^\infty_{\text{loc}}}$.

We use feedforward neural networks with ReLu activation function on the inner layers. Some other activation functions were also tested, including the sigmoid and the tanh functions. We found that ReLu was performing better on our examples, and we report only these results. If not otherwise stated, the output activation function is the identity, and the Heun scheme is used for $F_{\Delta t}$, with $p = 5$ substeps, excepted for Example 1 where $p = 1$ and $p = 5$ are compared.

Implementation of neural networks uses python TensorFlow 2, with Adam optimizer (see [1]), and the architecture is an Intel Xeon Gold 6140 Processor with 2 CPUs and a total of 36 cores.

## 8.1 Example 1 : Rotation with obstacle

This first problem is a two-dimensional example. We aim at computing the backward reachable set of a target disk $\mathscr{B}(x_A, r_0)$ before time $T$, while avoiding the region $\mathscr{B}(x_B, r_1)$ with the following parameters

$$x_A = (1, 0), \quad x_B = (0, 1), \quad r_0 = 0.5, \quad r_1 = 0.25, \quad \text{and } T = 0.4$$

(see Fig. 1). The dynamics $f(x, a)$ with controls $a \in [-1, 1]$ is given by

$$f((x_1, x_2), a) := 2\pi a(-x_2, x_1) \text{ with } a \in A := [-1, 1]$$

and corresponds to a clockwise to counter-clockwise rotation. We set

$$\varphi(x) := \|x - x_A\|_2 - r_0 \quad \text{and} \quad g(x) := r_1 - \|x - x_B\|_2.$$

The value $v(t, x)$ of this problem problem (as defined in (1), with $t \in [0, T]$ and $x \in \mathbb{R}^2$) is also solution of the following HJB equation with an obstacle term

$$\min\big( - v_t + 2\pi|x_2 v_{x_1} - x_1 v_{x_2}|,\ v - g(x)\big) = 0, \quad t \in [0, T] \tag{60}$$

$$v(T, x) = \max(\varphi(x), g(x)) \tag{61}$$

where $v_{x_i} = \frac{\partial v}{\partial x_i}$.

Here, the control networks use the sigmoid output activation function, with value in $[0, 1]$, and is converted to $[-1, 1]$ by a linear transformation.

In Fig 1, we compare the SL-scheme, the H-scheme and the L-scheme. Errors are given in Table 1, using here $(w_1, w_2) = (e_1, e_2)$, the canonical basis.

We first investigate the influence of the substeps ($p \geq 1$). We choose $F_h$ as the Heun scheme, with $N = 5$ time steps ($\Delta t = T/N$), and compare the results using $p = 1$ or $p = 5$ (recall that $p$ is the number of substeps in order to approximate the characteristic with a constant control $a$ on a given time interval $[t_k, t_k + \Delta t]$).

The results, for all schemes, are clearly in favor of using $p = 5$ (better characteristic approximation) which benefit from the regions of regularity of the control. Hence, for the forthcoming examples, we always use the Heun scheme with $p = 5$.

Notice that for this low-dimensional example ($d = 2$), only a small number of stochastic gradient iterations is enough to obtain reasonable results, and in particular to observe the contribution of $p$.

We also compare the three schemes for $p = 5$, looking at the relative errors. We observe that the L-scheme gives the best results, the H-scheme gives intermediate results and the SL-scheme is less precise. Here we observe that a local $L^1$ relative error less or equal to $10^{-2}$ corresponds to an almost perfect result to the eye.

| Scheme | Parameters | | | | | Global errors | | Local errors | | CPU time |
|---|---|---|---|---|---|---|---|---|---|---|
| | $N$ | lay. | neur. | $M$ | S.G. it. | $L_\infty$ | $L_1$ rel. | $L_\infty$ | $L_1$ rel. | (s.) |
| SL ($p = 1$) | 5 | 3 | 40 | 1000 | 1000 | 2.23e-01 | 5.23e-02 | 1.07e-01 | 3.14e-02 | 133.00 |
| SL ($p = 5$) | 5 | 3 | 40 | 1000 | 1000 | 1.22e-01 | 1.72e-02 | 1.02e-01 | 1.19e-02 | 182.94 |
| H ($p = 1$) | 5 | 3 | 40 | 1000 | 1000 | 2.12e-01 | 5.39e-02 | 1.13e-01 | 2.58e-02 | 180.18 |
| H ($p = 5$) | 5 | 3 | 40 | 1000 | 1000 | 1.20e-01 | 7.96e-03 | 7.83e-02 | 7.93e-03 | 285.84 |
| L ($p = 1$) | 5 | 3 | 40 | 1000 | 1000 | 5.99e-01 | 4.74e-02 | 5.01e-01 | 2.55e-02 | 54.42 |
| L ($p = 5$) | 5 | 3 | 40 | 1000 | 1000 | 2.10e-01 | 3.22e-03 | 2.00e-01 | 4.08e-03 | 106.46 |

Table 1: Errors for example 1

Finally, on this example, we have also tested a direct method (the DGM approach of [40], see also the PINN approach in [38]). A global space-time DNN is used in order to approximate the value $(t, x) \to v(t, x)$ solution of the PDE (60). However, in our experiments, we found that the DNN in general fails to see the obstacle part of the solution. A typical illustration is given in Figure 2, where 3 simulations with increasing final time $T$ are presented. We considered neural networks with tanh activation function, both in the inner and output layers. In the presented results, the network uses 3 inner layers of 40 neurons. At each iteration of the minimization, the stochastic gradient draws 10,000 points in the space-time domain and 1000 points on the border $t = T$ ($100,000$ iterations of stochastic gradient used).

## 8.2 Example 2 : eikonal equation

Next we consider a $d$-dimensional problem, with no obstacle term, for various dimensions $d = 6, 7, 8$.

More precisely the dynamics is $f(x, a) := a$ with $a \in A := \overline{\mathscr{B}}(0, 1)$, the closed unit ball of $\mathbb{R}^d$ (for the Euclidean norm). The function $\varphi$ is

$$\varphi(x) := \min\Big(\|x - x_A\|_2 - r_0,\ \|x - x_B\|_2 - r_0\Big)$$

with $x_A = (1, 0, \ldots, 0)$ and $x_B = (-1, 0, \ldots, 0)$, and parameters $T = 1.0$ and $r_0 = 0.5$. Hence the value is defined as the solution of (1) (with $g := -\infty$). The analytical solution is known and given by $v(t, x) = \min\big((\|x - x_A\|_2 - (T - t))_+ - r_0,\ (\|x - x_B\|_2 - (T - t))_+ - r_0\big)$.

The corresponding HJB equation (for $x \in \mathbb{R}^d$), using $\max_{a \in A} f(x, a) \cdot \nabla_x v = \|\nabla_x v\|$, is the following eikonal equation

$$-v_t + \|\nabla_x v\| = 0, \quad t \in [0, T] \tag{62}$$

$$v(T, x) = \varphi(x). \tag{63}$$

Here, we choose the control networks to take their values in $\mathbb{R}^d$. The results are then converted from $\mathbb{R}^d$ to the unit ball $\overline{\mathscr{B}}(0, 1)$ of $\mathbb{R}^d$ by using the map $p \mapsto \frac{p}{\max(1, \|p\|)}$. (Numerical tests showed that the choice of the map may affect the results, and the results may deteriorate in particular when using an anisotropic map.) Errors are given in Table 2 for dimensions $d = 6, 7, 8$, and some illustrations are given in Fig. 3 for dimension $d = 8$ (results for $d \in \{6, 7\}$ are indistinguishable to the eye from the case $d = 8$, and they are not included). Errors and figures are computed in the plane $\mathcal{P}$ generated by the first two vectors $e_1, e_2$ of the canonical basis of $\mathbb{R}^d$, which corresponds to the choice $(w_1, w_2) = (e_1, e_2)$ here.

In particular we observe that the $L$-scheme performs well (numerical and exact 0-level sets are indistinguisable to the eye), as long as a sufficient number of SG iterations is used, and that the

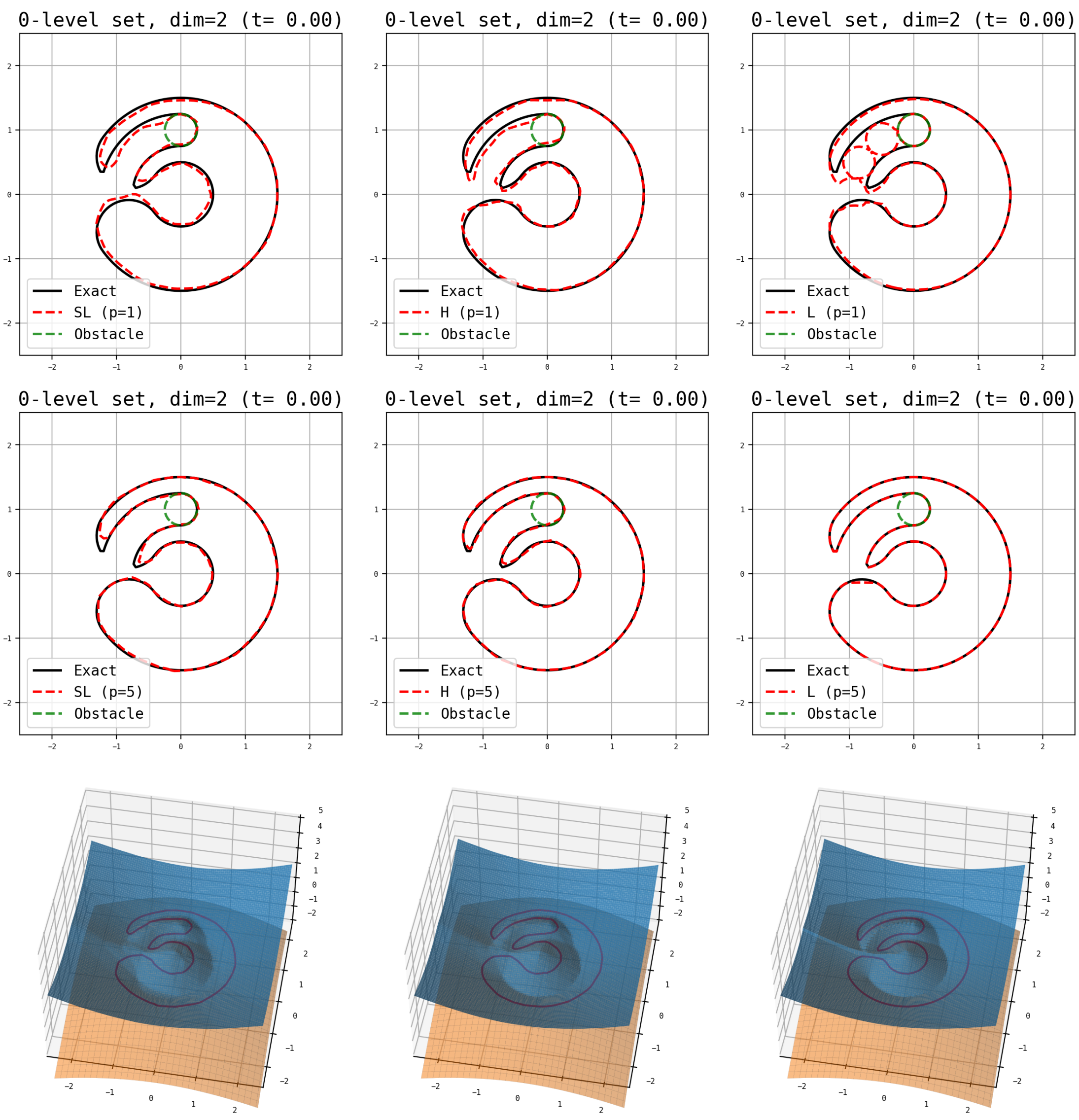


Figure 1: (Example 1) The SL-scheme (left), the H-scheme (middle) and the L-scheme (right) are tested with Euler scheme with $p = 1$ (top) and Heun scheme with $p = 5$ (middle/bottom). The bottom figures corresponds to the surface plots of $z = v(0, x, y)$ (blue), the plot of the obstacle function (orange), and the 0-level set (red line). Networks uses 3 hidden layers, 40 neurons, with $N = 5$ time steps.

control map from $\mathbb{R}^d$ to $\overline{\mathscr{B}}(0,1)$ is well chosen.

| Scheme | Parameters | | | | | | Global errors | | Local errors | | CPU time |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | $d$ | $N$ | layers | neurons | $M$ | S.G it. | $L_\infty$ | $L_1$ rel. | $L_\infty$ | $L_1$ rel. | |
| L | 6 | 4 | 3 | 40 | 1000 | 100000 | 2.16e-02 | 1.96e-03 | 4.06e-04 | 1.58e-04 | 1h26 |
| L | 7 | 4 | 3 | 40 | 1000 | 200000 | 5.00e-02 | 3.41e-03 | 1.51e-02 | 1.26e-04 | 3h55 |
| L | 8 | 4 | 3 | 40 | 1000 | 400000 | 1.99e-01 | 1.81e-02 | 4.39e-04 | 2.19e-04 | 10h31 |

Table 2: (Example 2) $L$-scheme, dimensions $d = 6, 7, 8$

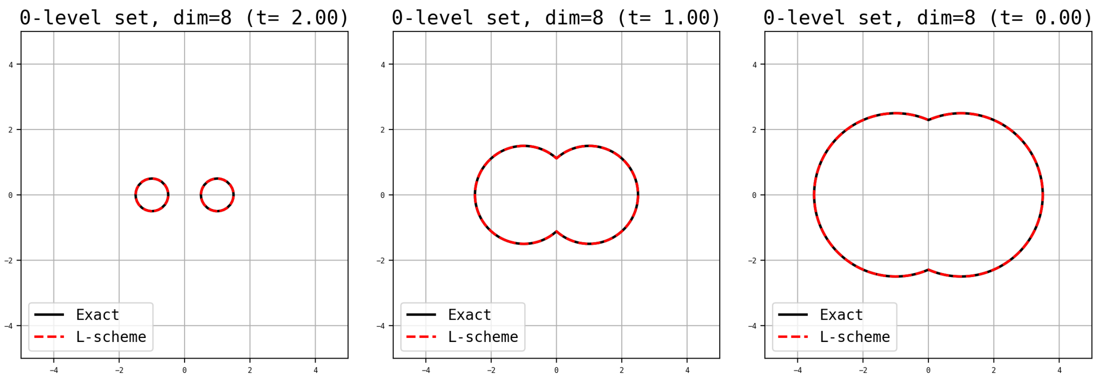


Figure 3: (Example 2) Eikonal equation, L-scheme, dimension $d = 8$, at time $t = T = 2.0$ (left, terminal condition), $t = 1.0$ (center), $t = 0.0$ (right). Networks of 3 hidden layers and 40 neurons; $N = 4$ time steps.

### 8.3 Example 3: $d$-dimensional advection with obstacle

We now consider an elementary $d$-dimensional advection problem with an obstacle term, and compare the SL-scheme, the H-scheme and the L-scheme. The problem is to reach the target $\{\varphi(x) \leq 0\}$, while avoiding an obstacle $\{g(x) \leq 0\}$, with linear dynamics $f(x,a) := -ae$ where $e \in \mathbb{R}^d$ and the control $a$ lies in $A := [0,1]$. Since $\max_{a\in[0,1]}(ae \cdot \nabla v) = \max(0, e \cdot \nabla v)$, the

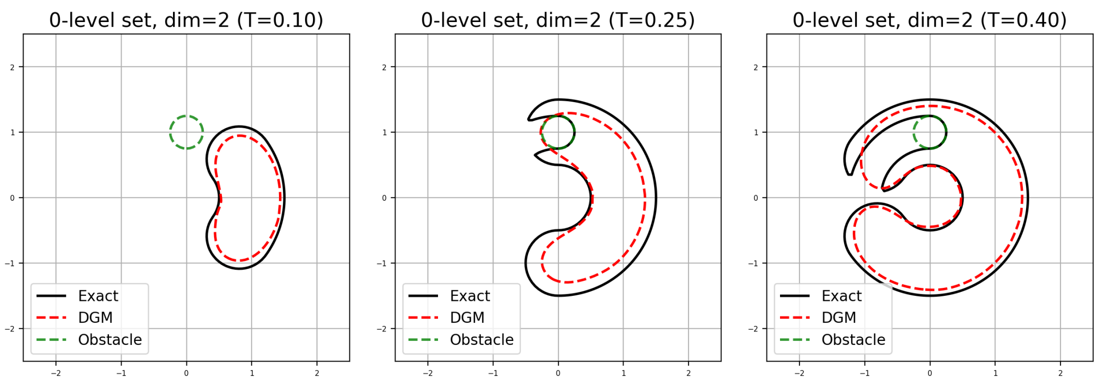


Figure 2: (Example 1) DGM direct method in dimension $d = 2$, the computation is done in dimension $d+1$ to include time. Results are given at $t = 0$. Final time is set to $T = 0.1$ (left), $T = 0.25$ (middle) and $T = 0.4$ (right).

corresponding HJB equation is

$$\min\Big( -v_t + \max(0, e\cdot\nabla v),\ v - g(x)\Big) = 0, \quad t \in [0,T] \tag{64}$$

$$v(T,x) = \max(\varphi(x), g(x)). \tag{65}$$

The reachable set at time $t$ is given by $\{v(t,\cdot) \le 0\}$ (corresponding to the set of points that can reach the target before time $t$). The target function $\varphi$ and the obstacle function $g$ are defined by

$$\varphi(x) \coloneqq \|x - A_0\|_2 - r_0 \quad \text{and} \quad g(x) \coloneqq r_1 - \|x - A_1\|_2$$

so that $\{\varphi(x) \le 0\} = \overline{\mathscr{B}}(A_0, r_0)$, and $\{g(x) \ge 0\} = \overline{\mathscr{B}}(A_1, r_1)$. The following parameters are considered:

$$e = (1,1,...,1)/\sqrt{d}, \quad A_0 = -(1,1,...,1)/\sqrt{d}, \quad A_1 = (0,0,\dots,0), \quad r_0 = 0.5, \quad r_1 = 0.25.$$

Here, the exact solution can be computed as $v(t,x) = \varphi(p(x,t)) \vee g(q(x))$, where

$$p(x,t) \coloneqq x - \max\left(0, \min\left(\langle x - A_0, e\rangle, T-t\right)\right) e \quad \text{and} \quad q(x) = x - \max\left(\langle x - A_1, e\rangle, 0\right) e.$$

For the control networks we use the sigmoid as the output activation function (output in $[0,1]$). For the figure and error computations, we have chosen a grid in the 2-dimensional plane $\mathcal{P} = Vect(w_1, w_2)$ where

$$w_1 = e \equiv (1,1,\dots,1)/\sqrt{d}, \qquad w_2 = (1,-1,0,\dots,0)/\sqrt{2}.$$

Notice that for such parameters the exact 0-level set is the same independently of the dimension $d$. In order to perform the SG iterations, the size of the random batch points is set to $M = 2000$ (as well as for the value approximation by neural networks, step $(ii)$ of SL-scheme). Results are given in Table 3 and in Figure 4, for dimension $d = 6$. The difference between the schemes is clearer when the dimension increases.

The CPU time reflects the computational cost of the projection of the value function that is present in the SL-scheme and the H-scheme. Both the SL-scheme and the H-scheme need to optimize two networks per time step: one for the control, and one for the value, whereas the L-scheme needs only one for the control. Additionnally, the H-scheme computes the whole characteristics, leading to a higher CPU time than the SL-scheme. However, if the number of time steps $N$ grows, the L-scheme may become more expensive than the SL-scheme.

Looking in particular at the figures in Figure 4, this example shows some kind of numerical diffusion that we may encounter with the SL-scheme, as well as with the H-scheme, to a lesser extent.

In this example, we have also numerically observed that an increasing number of stochastic gradient iterations were needed as the dimension increases.

| Scheme | Parameters | | | | | | Global errors | | Local errors | | CPU |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | $d$ | $N$ | lay. | neur. | $M$ | S.G. it. | $L_\infty$ | $L_1$ rel. | $L_\infty$ | $L_1$ rel. | time |
| SL | 6 | 5 | 3 | 40 | 2000 | 200000 | 9.08e-02 | 1.84e-02 | 5.77e-02 | 1.29e-02 | 6h41 |
| H | 6 | 5 | 3 | 40 | 2000 | 200000 | 9.47e-02 | 1.50e-02 | 6.42e-02 | 1.08e-02 | 8h31 |
| L | 6 | 5 | 3 | 40 | 2000 | 200000 | 2.14e-03 | 9.58e-05 | 1.79e-03 | 9.54e-05 | 4h59 |

Table 3: (Example 3) Advection with obstacle, comparison of schemes

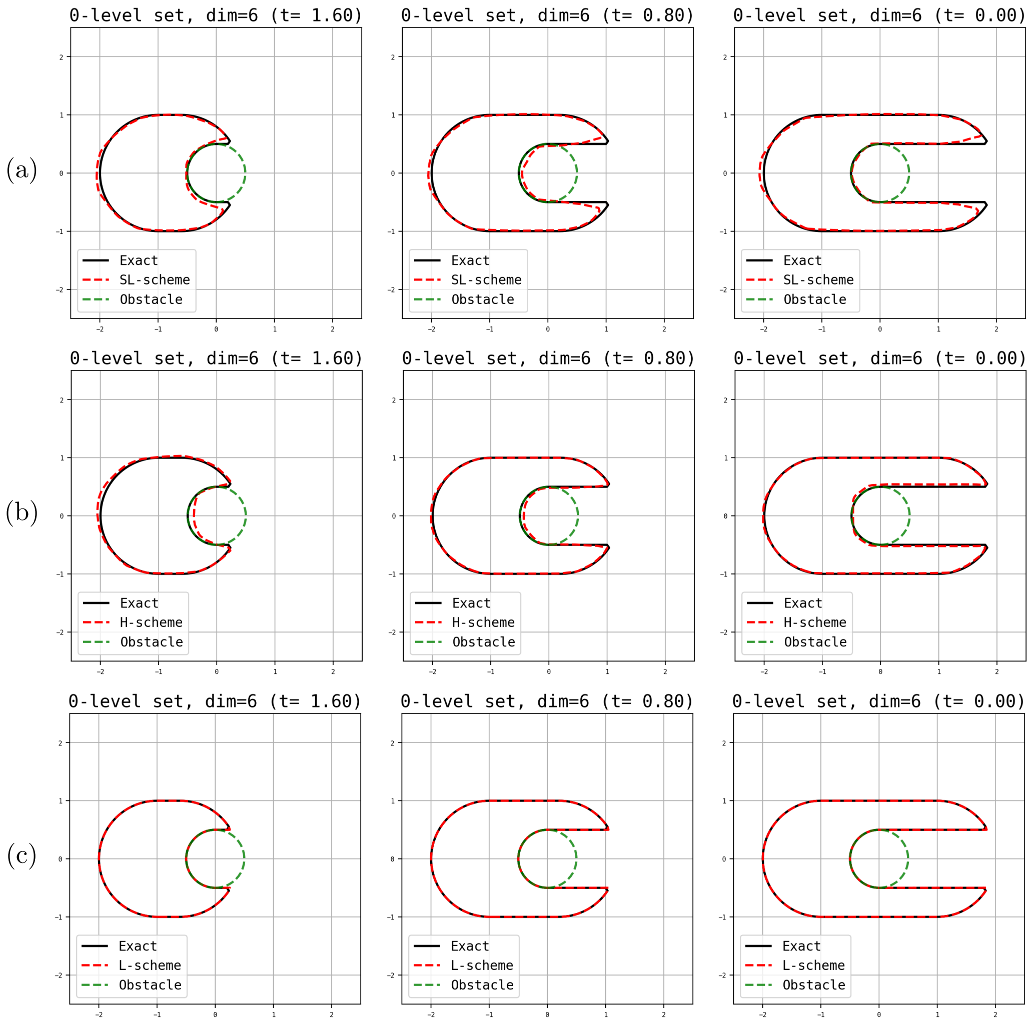


Figure 4: (Example 3) Results obtained with H-scheme (row (a)), SL-scheme (row (b)), and L-scheme (row (c)) respectively. Dimension $d = 6$, $N = 5$ time steps, neural networks of 3 layers and 40 neurons.

## 8.4 Example 4: eikonal advection equation with obstacle, large drift

We consider now a mixed $d$-dimensional eikonal/advection equation with an obstacle term. More precisely, the dynamics is defined by $f(x, a) = -be_1 + ca$, where $e_1 = (1, 0, ..., 0)^t \in \mathbb{R}^d$, the control $a$ belongs to $A := \mathbb{S}^{d-1}$ the unit ball of $\mathbb{R}^d$, $b \in \mathbb{R}$ is a coefficient corresponding to the "drift", and $c \geq 0$ is a speed coefficient for the eikonal part of the equation. Then the corresponding HJB equation is:

$$\min\Big(-v_t + b \cdot v_{x_1} + c\|\nabla_x v\|,\ v - g(x)\Big) = 0, \quad t \in [0, T] \tag{66}$$

$$v(T, x) = \max(\varphi(x), g(x)). \tag{67}$$

The obstacle term and terminal condition are defined as

$$g(x) := \min\left(g_{\max} - c_e\,|x_1 - g_c|\,, c_x\,|x_\perp| + g_{\min}\right), \quad \varphi(x) := \|x\| + \alpha_{\min},$$

where $c_e$, $c_x$, $g_c$, $g_{\max}$ $g_{\min}$ and $\alpha_{\min}$ are coefficients, and, for a given $x = (x_1, \dots, x_d)^t \in \mathbb{R}^d$, $x_\perp := (0, x_2, \dots, x_d)^t$ (the orthogonal projection of $x$ on $\mathrm{Vect}(e_2, \dots, e_d)$). Note that this obstacle term correspond to a wall obstacle with a tube opening centered around the $e_1$ axis (see for instance the green dotted line in Fig. 6).

The exact solution can be computed. Details are given in Appendix A. In this example, more precisely, the following parameters are considered

$$g_{\max} = 2, \quad g_{\min} = -2, \quad c_e = 1, \quad c_x = 1.5, \quad g_c = 4, \quad b = 1, \quad c = 0.5, \quad \alpha_{\min} = -1.$$

Here in particular $|b| > c$: the drift is dominant, which corresponds to a non-controllable situation.

*Comparison of schemes in dimension $d = 4$.* First, the SL-, H- and L-schemes are compared. Neural networks with 3 layers of 60 neurons are chosen, and each simulation uses 100,000 stochastic gradient step. Figure (5) displays the error $|\hat{V}_0(x) - v(0, x)|$ in function of space, with $N \in \{8, 16\}$ number of time steps. Results are shown in Table 4. For this example as well as the next one, the plane $\mathcal{P}$ for error computations and graphics is generated by $(w_1, w_2) = (e_1, e_2)$, the first two vectors of the canonical basis of $\mathbb{R}^d$.

The SL-scheme approximates both the control and the value function by neural networks. The projection of the latter is a source of errors, that accumulates during the simulation. This drawback is avoided with the H-scheme and L-scheme, where the value function is computed as a composition of the (exact) target function $\varphi$ and the approximated controls. Again, for the error, the H-scheme and L-scheme behave better than the SL-scheme.

Furthermore, for the H-scheme and the L-scheme, when $N$ varies from 8 to 16, we observe very roughly that the $L^1$ (global and local) errors are divided by a factor two (this is less clear for the $L^\infty$ errors). This is not the case for the SL-scheme, for which errors have a tendency to accumulate more with the time iterations.

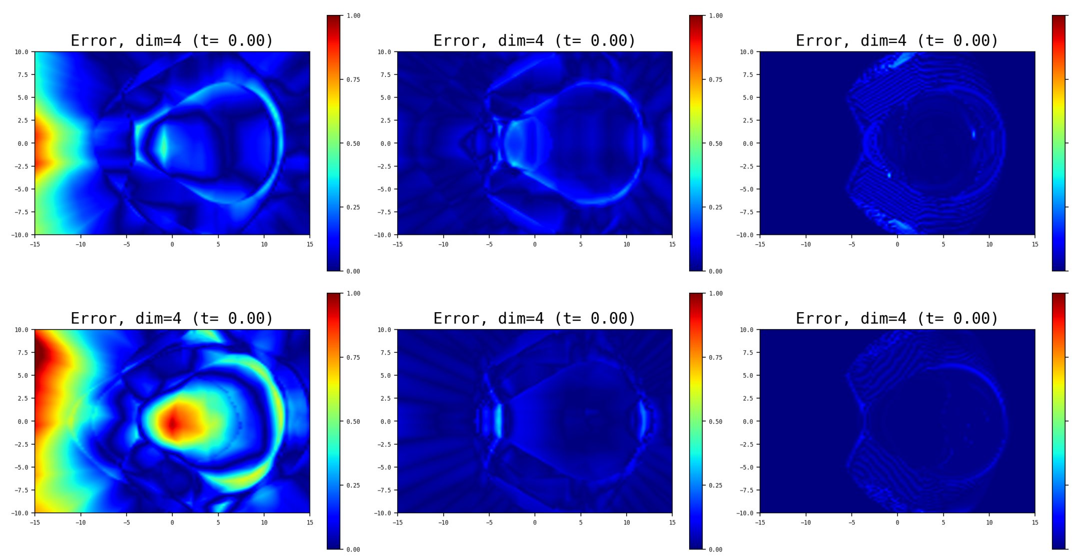


Figure 5: (Example 4) Comparison between SL-scheme (left), H-scheme (middle) and L-scheme (right), for $N = 8$ (top) and $N = 16$ (bottom).

| Scheme | Parameters | | | | | | Global errors | | Local errors | | CPU time |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | $d$ | $N$ | lay. | neur. | $M$ | S.G. it. | $L_\infty$ | $L_1$ rel. | $L_\infty$ | $L_1$ rel. | |
| SL | 4 | 8 | 3 | 60 | 4000 | 100000 | 8.60e-01 | 4.91e-02 | 1.98e-01 | 7.48e-02 | 6h14 |
| H | 4 | 8 | 3 | 60 | 4000 | 100000 | 3.34e-01 | 1.58e-02 | 2.01e-01 | 4.61e-02 | 11h26 |
| L | 4 | 8 | 3 | 60 | 4000 | 100000 | 3.24e-01 | 6.68e-03 | 1.09e-01 | 2.56e-02 | 8h16 |
| SL | 4 | 16 | 3 | 60 | 4000 | 100000 | 1.07e+00 | 9.21e-02 | 2.92e-01 | 1.06e-01 | 12h29 |
| H | 4 | 16 | 3 | 60 | 4000 | 100000 | 3.32e-01 | 9.62e-03 | 1.54e-01 | 2.88e-02 | 34h26 |
| L | 4 | 16 | 3 | 60 | 4000 | 100000 | 1.85e-01 | 3.57e-03 | 8.85e-02 | 1.64e-02 | 28h08 |

Table 4: (Example 4) comparison between schemes

*Test of the L-scheme for increasing dimensions.* Next, the L-scheme is tested for several dimensions $d \in \{2, 4, 6, 8\}$, and results are given in Table 5. The neural network size is kept constant, with 3 layers of 60 neurons, as for the number of time iterations ($N = 8$). In order to reach comparable precision, we have observed that the number of stochastic gradient iterations has to grow with $d$. As the dimension increases, more iterations are needed to explore the whole region of interest. Otherwise, the scheme is relatively robust with respect to the physical dimension of the problem (see Fig. 6).

We observe for dimension $d = 8$ some defects in the numerical solution, and some oscillations appears. Because of CPU time limitations, we did not attempt using more S.G. iterations, although in principle, as observed for lower dimensions, this should enable a better optimization and solve the problem.

| Scheme | Parameters | | | | | | Global errors | | Local errors | | CPU time |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | $d$ | $N$ | lay. | neur. | $M$ | S.G. it. | $L_\infty$ | $L_1$ rel. | $L_\infty$ | $L_1$ rel. | |
| L | 2 | 8 | 3 | 60 | 4000 | 50000 | 2.66e-01 | 5.99e-03 | 1.19e-01 | 4.61e-02 | 3h02 |
| L | 4 | 8 | 3 | 60 | 4000 | 100000 | 3.90e-01 | 6.77e-03 | 1.16e-01 | 2.69e-02 | 8h13 |
| L | 6 | 8 | 3 | 60 | 4000 | 400000 | 9.69e-01 | 1.09e-02 | 1.78e-01 | 2.88e-02 | 35h20 |
| L | 8 | 8 | 3 | 60 | 4000 | 600000 | 1.05e+00 | 3.75e-02 | 1.71e-01 | 2.95e-02 | 45h27 |

Table 5: (Example 4) $L$-scheme, dimensions $d = 2, 4, 6, 8$

## 8.5 Example 5: eikonal advection equation with obstacle, small drift

We now turn on a similar example as in example 4, excepted for the coefficients which are now

$$c = 1, \quad b = 0.5.$$

Results obtained with the L-scheme are given in Table 6 and Fig. 7. Here $|b| < c$, the drift is small, and it corresponds to a controllable situation. We observe that the front has to negociate a sharper angle near the boundary of the tube (see Fig. 7).

As in example 4, the results are rather robust with respect to dimension, provided the number of stochastic gradient iterations is large enough.

| Scheme | Parameters | | | | | | Global errors | | Local errors | | CPU time |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | $d$ | $N$ | lay. | neur. | $M$ | S.G it. | $L_\infty$ | $L_1$ rel. | $L_\infty$ | $L_1$ rel. | |
| L | 2 | 8 | 3 | 60 | 4000 | 50000 | 1.24e-01 | 2.94e-03 | 8.81e-02 | 5.21e-03 | 3h09 |
| L | 4 | 8 | 3 | 60 | 4000 | 100000 | 2.49e-01 | 4.70e-03 | 8.67e-02 | 5.45e-03 | 6h51 |
| L | 6 | 8 | 3 | 60 | 4000 | 400000 | 8.74e-01 | 3.70e-02 | 1.09e-01 | 1.01e-02 | 35h07 |

Table 6: Errors for example 5

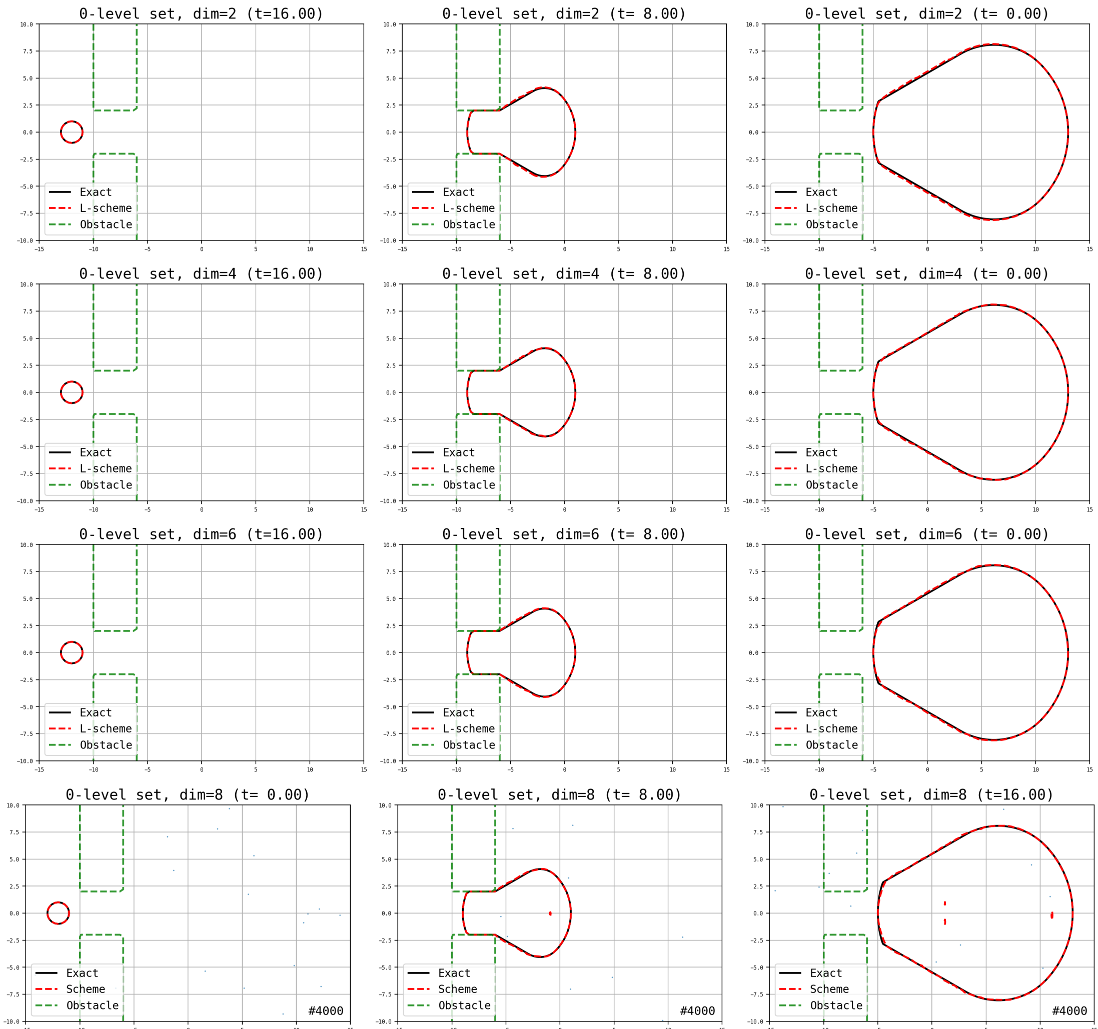


Figure 6: (Example 4) L-scheme, dimensions 2,4,6,8, $N = 8$ time steps, networks : 3 layers of 60 neurons.

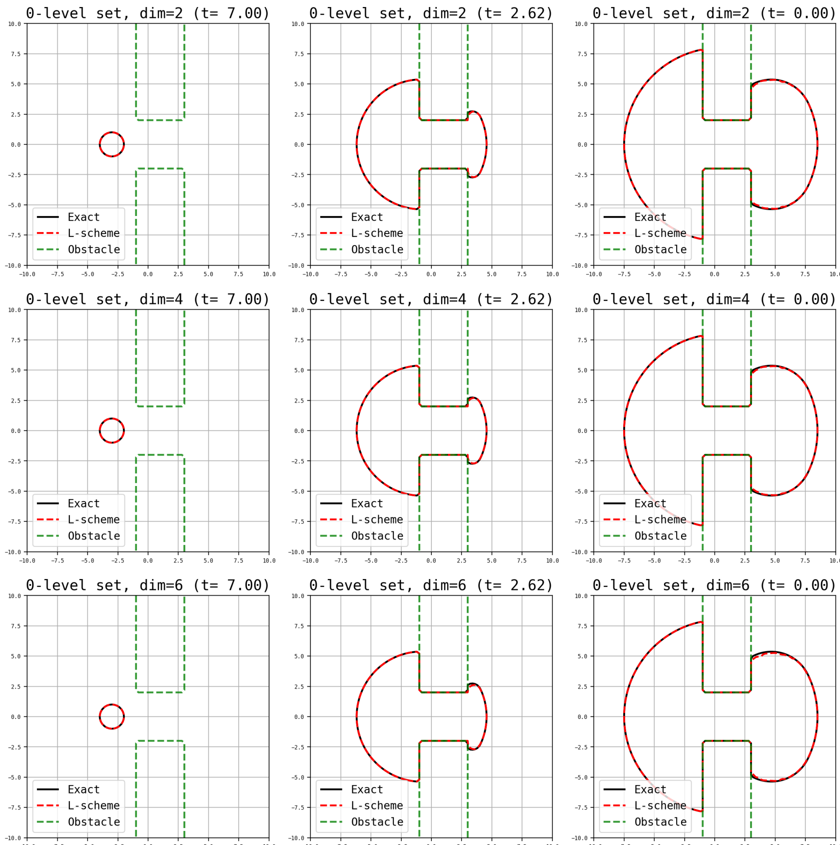


Figure 7: (Example 5) L-scheme, dimensions 2, 4 and 6. Networks with 3 layers of 60 neurons, $N = 8$ time steps.

# A Semi-analytical solution for examples 4 and 5

We briefly describe how to compute the exact values for examples 4 and 5, that is, in order to compute $v = v(t, x)$ for given values $t$ and $x$. We consider the case of data $\varphi(x) = \|x\| + \alpha_{\min}$, for $\alpha_{\min} \in \mathbb{R}$.

For $\alpha \in [g_{\min}, g_{\max}]$, notice that the level set $\{g = \alpha\}$ corresponds to a wall pierced by a square door (see figure 8).

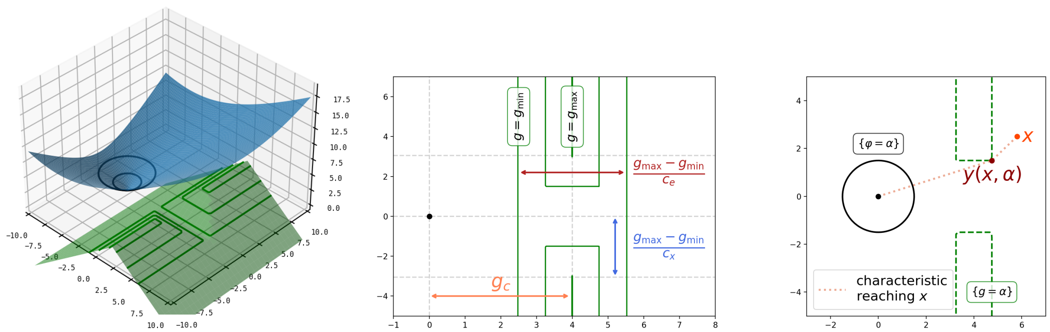


Figure 8: Initial condition (in blue) and obstacle function (in green). Illustration of parameters. Exemple of a characteristics with obstacle in the case $b = 0$.

First, by using the symmetries of the problem around the $e_1$ axis, and an orthogonal axis along $x - x_1 e_1$, it is possible to set back the problem into a 2-dimensional problem.

If no obstacle term is present (or if it does not modify the value $v$), for a given point $x \in \mathbb{R}^d$, and for a given level-set value $v$, it is possible to compute the minimal time to reach $x$ from the initial level set front, corresponding to some point on the level set $\{\varphi(x) = v\}$. More precisely, the optimal trajectory is a straight segment and the value satisfies

$$v(t, x) = (\|x - be_1 t\| - ct)_+ + \alpha_{\min},$$

and $t$ is also the time for the front $\{\varphi(\cdot) = v\}$ to reach the point $x$.

In the more complex situation when the optimal trajectory from $\{\varphi(\cdot) = v\}$ to point $x$ is not a straight segment, it is composed of two segments: one starting from some point on $\{\varphi(\cdot) = v\}$ to reach some point $y$ on the level set $\{g(y) = v\}$, and the other one starting from $y$ to $x$ - as depicted in Fig. 8-right). Then the minimal time $t$, associated to some value $v = v(t, x)$, is the sum of two minimal times $t_1$, $t_2$ such that

$$t = t_1 + t_2 \tag{68a}$$

$$\|y - be_1 t_1\|^2 = (ct_1 + v - \alpha_{\min})^2 \tag{68b}$$

$$\|x - (y + be_1 t_2)\|^2 = (ct_2)^2. \tag{68c}$$

Note that $t_1$ is the time for the level set $\{\varphi(\cdot) = v\}$ to reach $y$ on $\{g(\cdot) = v\}$, $t_2$ is the time to reach $x$ from $y$. Then, for a given value $v$, $y = y(v)$ is known and it has an affine analytic expression in term of $v$. Times $t_1 = t_1(v)$ and $t_2 = t_2(v)$ are obtained as root solutions of (68b) and (68c). Finally, for given $(t, x)$, the value $v$ is obtained through a Newton algorithm for solving system (68). Note that on regions not attained by the front, we consider the complex roots $t_1(v)$ or $t_2(v)$ in the Newton algorithm (in order to always have well-defined and continuous reaching times).

# References


[1] M. Abadin *et al.* TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] M. Akian, S. Gaubert, and A. Lakhoua. The max-plus finite element method for solving deterministic optimal control problems: basic properties and convergence analysis. *SIAM J. Control Optim.*, 47(2):817–848, 2008.

[3] A. Alla, M. Falcone, and L. Saluzzi. A tree structure algorithm for optimal control problems with state constraints. *Rendiconti di Matematica e delle sue Applicazioni. Serie VII*, 41(3-4):193–221, 2020.

[4] A. Altarovici, O. Bokanowski, and H. Zidani. A general Hamilton-Jacobi framework for non-linear state-constrained control problems. *ESAIM: Control, Optimisation and Calculus of Variations*, 19(337–357), 2013.

[5] C. Anil, J. Lucas, and R. Grosse. Sorting out lipschitz function approximation. In *International Conference on Machine Learning*, pages 291–301. PMLR, 2019.

[6] M. Assellaou, O. Bokanowski, A. Desilles, and H. Zidani. Value function and optimal trajectories for a maximum running cost control problem with state constraints. Application to an abort landing problem. *ESAIM Math. Model. Numer. Anal.*, 52(1):305–335, 2018.

[7] A. Bachouch, C. Huré, N. Langrené, and H. Pham. Deep Neural Networks Algorithms for Stochastic Control Problems on Finite Horizon: Numerical Applications. *Methodol. Comput. Appl. Probab.*, 24(1):143–178, 2022.

[8] S. Bansal, M. Chen, K. Tanabe, and C. Tomlin. Provably safe and scalable multi-vehicle trajectory planning. *IEEE Transactions on Control Systems Technology (TCST)*, 2021.

[9] C. Barrera-Esteve, F. Bergeret, C. Dossal, E. Gobet, A. Meziou, R. Munos, and D. Reboul-Salze. Numerical methods for the pricing of swing options: a stochastic control approach. *Methodology and computing in applied probability*, 8(4):517–540, 2006.

[10] C. Bender and R. Denk. A forward scheme for backward sdes. *Stochastic processes and their applications*, 117(12):1793–1812, 2007.

[11] O. Bokanowski, E. Bourgeois, A. Désilles, and H. Zidani. Payload optimization for multi-stage launchers using hjb approach and application to a sso mission. In *Proceedings, 20th IFAC*, 2017.

[12] O. Bokanowski, Y. Cheng, and C.-W. Shu. A discontinuous Galerkin scheme for front propagation with obstacles. *Numer. Math.*, 126(1):1–31, 2014.

[13] O. Bokanowski, Y. Cheng, and C.-W. Shu. Convergence of discontinuous Galerkin schemes for front propagation with obstacles. *Math. Comp.*, 85(301):2131–2159, 2016.

[14] O. Bokanowski, N. Forcadel, and H. Zidani. Reachability and minimal times for state constrained nonlinear problems without any controllability assumption. *SIAM J. Control Optim.*, 48(7):4292–4316, 2010.

[15] O. Bokanowski, N. Gammoudi, and H. Zidani. Optimistic Planning Algorithms For State-Constrained Optimal Control Problems. *Computers & Mathematics with Applications*, 109(1):158–179, 2022.

[16] O. Bokanowski, J. Garcke, M. Griebel, and I. Klompmaker. An adaptive sparse grid semi-Lagrangian scheme for first order Hamilton-Jacobi Bellman equations. *Journal of Scientific Computing*, 55(3):575–605, 2013.

[17] M. G. Crandall and P. L. Lions. Two approximations of solutions of hamilton-jacobi equations. *Mathematics of Computation*, 43(167):1–19, 1984.

[18] K. Debrabant and E. R. Jakobsen. Semi-Lagrangian schemes for linear and fully non-linear Hamilton-Jacobi-Bellman equations. In *Hyperbolic problems: theory, numerics, applications*, volume 8 of *AIMS Ser. Appl. Math.*, pages 483–490. Am. Inst. Math. Sci. (AIMS), Springfield, MO, 2014.

[19] S. Dolgov, D. Kalise, and K. K. Kunisch. Tensor Decomposition Methods for High-dimensional Hamilton–Jacobi–Bellman Equations. *SIAM Journal on Scientific Computing*, 43(3):A1625–A1650, Jan. 2021.

[20] M. Falcone and R. Ferretti. Numerical methods for Hamilton-Jacobi type equations. In *Handbook of numerical methods for hyperbolic problems*, volume 17 of *Handb. Numer. Anal.*, pages 603–626. Elsevier/North-Holland, Amsterdam, 2016.

[21] J. Garcke and A. Kröner. Suboptimal feedback control of PDEs by solving HJB equations on adaptive sparse grids. *J. Sci. Comput.*, 70(1):1–28, 2017.

[22] M. Germain, H. Pham, and X. Warin. Approximation error analysis of some deep backward schemes for nonlinear pdes. *SIAM Journal on Scientific Computing*, 44(1):A28–A56, 2022.

[23] E. Gobet, J.-P. Lemor, and X. Warin. A regression-based monte carlo method to solve backward stochastic differential equations. *The Annals of Applied Probability*, 15(3):2172–2202, 2005.

[24] E. Gobet and P. Turkedjiev. Linear regression mdp scheme for discrete backward stochastic differential equations under general conditions. *Mathematics of Computation*, 85(299):1359–1391, 2016.

[25] L. Györfi, M. Kohler, A. Krzyżak, and H. Walk. *A Distribution-Free Theory of Nonparametric Regression.* Springer Series in Statistics. Springer New York, New York, NY, 2002.

[26] J. Han, A. Jentzen, and W. E. Solving high-dimensional partial differential equations using deep learning. *Proc. Natl. Acad. Sci. USA*, 115(34):8505–8510, 2018.

[27] J. Han and J. Long. Convergence of the Deep BSDE Method for Coupled FBSDEs. *Probability, Uncertainty and Quantitative Risk*, 5(1):5, Dec. 2020. arXiv:1811.01165 [cs, math].

[28] C. Hu and C.-W. Shu. A discontinuous Galerkin finite element method for Hamilton-Jacobi equations. *SIAM Journal on Scientific Computing*, 21:666–690, 1999.

[29] C. Huré, H. Pham, A. Bachouch, and N. Langrené. Deep neural networks algorithms for stochastic control problems on finite horizon: convergence analysis. *SIAM J. Numer. Anal.*, 59(1):525–557, 2021.

[30] C. Huré, H. Pham, and X. Warin. Deep backward schemes for high-dimensional nonlinear pdes. *Mathematics of Computation*, 89(324):1547–1579, 2020.

[31] M. Jensen and I. Smears. On the convergence of finite element methods for Hamilton-Jacobi-Bellman equations. *SIAM J. Numer. Anal.*, 51(1):137–162, 2013.

[32] N. V. Krylov. *Controlled diffusion processes*, volume 14 of *Stochastic Modelling and Applied Probability.* Springer-Verlag, Berlin, 2009. Translated from the 1977 Russian original by A. B. Aries, Reprint of the 1980 edition.

[33] H. J. Kushner and P. G. Dupuis. *Numerical methods for stochastic control problems in continuous time*, volume 24 of *Applications of mathematics.* Springer, New York, 2001. Second edition.

[34] E. B. Lee and L. Markus. *Foundations of Optimal Control Theory.* R.E. Krieger Pub. Co, Malabar, Fla, 1986.

[35] F. Li and C.-W. Shu. Reinterpretation and simplified implementation of a discontinuous Galerkin method for Hamilton-Jacobi equations. *Applied Mathematics Letters*, 18:1204–1209, 2005.

[36] W. M. McEneaney, A. Deshpande, and S. Gaubert. Curse-of-complexity attenuation in the curse-of-dimensionality-free method for hjb pdes. In *2008 American Control Conference*, pages 4684–4690, 2008.

[37] S. Osher and C.-W. Shu. High essentially nonoscillatory schemes for Hamilton-Jacobi equations. *SIAM J. Numer. Anal.*, 28(4):907–922, 1991.

[38] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.*, 378:686–707, 2019.

[39] C.-W. Shu. High order ENO and WENO schemes for computational fluid dynamics. In *High-order methods for computational physics*, volume 9 of *Lect. Notes Comput. Sci. Eng.*, pages 439–582. Springer, Berlin, 1999.

[40] J. Sirignano and K. Spiliopoulos. Dgm: A deep learning algorithm for solving partial differential equations. *Journal of computational physics*, 375:1339–1364, 2018.

[41] I. Smears and E. Süli. Discontinuous Galerkin finite element methods for time-dependent Hamilton-Jacobi-Bellman equations with Cordes coefficients. *Numer. Math.*, 133(1):141–176, 2016.

[42] B. Somil and J. T. Claire. DeepReach: A Deep Learning Approach to High-Dimensional Reachability. In *International Conference on Robotics and Automation (ICRA)*, 2021.

[43] U. Tanielian, M. Sangnier, and G. Biau. Approximating Lipschitz continuous functions with GroupSort neural networks. 2020, 2020.