Deep backward schemes for high-dimensional nonlinear PDEs

Côme Huré, Huyên Pham, Xavier Warin

▶ **To cite this version:**

Côme Huré, Huyên Pham, Xavier Warin. Deep backward schemes for high-dimensional nonlinear PDEs. Mathematics of Computation, 2020, 89 (324), pp.1547-1579. 10.1090/mcom/3514 . hal-02005362v2

HAL Id: hal-02005362
https://hal.science/hal-02005362v2

Submitted on 4 Jun 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Deep backward schemes for high-dimensional nonlinear PDEs *

Côme Huré† Huyên Pham‡ Xavier Warin §

First version: february 4, 2019
This final accepted version: November 8, 2019

Abstract

We propose new machine learning schemes for solving high dimensional nonlinear partial differential equations (PDEs). Relying on the classical backward stochastic differential equation (BSDE) representation of PDEs, our algorithms estimate simultaneously the solution and its gradient by deep neural networks. These approximations are performed at each time step from the minimization of loss functions defined recursively by backward induction. The methodology is extended to variational inequalities arising in optimal stopping problems. We analyze the convergence of the deep learning schemes and provide error estimates in terms of the universal approximation of neural networks. Numerical results show that our algorithms give very good results till dimension 50 (and certainly above), for both PDEs and variational inequalities problems. For the PDEs resolution, our results are very similar to those obtained by the recent method in [EHJ17] when the latter converges to the right solution or does not diverge. Numerical tests indicate that the proposed methods are not stuck in poor local minima as it can be the case with the algorithm designed in [EHJ17], and no divergence is experienced. The only limitation seems to be due to the inability of the considered deep neural networks to represent a solution with a too complex structure in high dimension.

Key words: Deep neural networks, nonlinear PDEs in high dimension, optimal stopping problem, backward stochastic differential equations.

MSC Classification: 60H35, 65C20, 65M12.

*This work is supported by FiME, Laboratoire de Finance des Marchés de l'Energie, and the "Finance and Sustainable Development" EDF - CACIB Chair.

†LPSM, Paris-Diderot University hure at lpsm.paris

‡LPSM, Paris-Diderot University, CREST-ENSAE & FiME pham at lpsm.paris

§EDF R&D & FiME xavier.warin at edf.fr

1 Introduction

This paper is devoted to the resolution in high dimension of nonlinear parabolic partial differential equations (PDEs) of the form

$$\begin{cases} \partial_t u + \mathcal{L}u + f(.,.,u,\sigma^\intercal D_x u) = 0, & \text{on } [0,T)\times\mathbb{R}^d, \\ u(T,.) = g, & \text{on } \mathbb{R}^d, \end{cases} \tag{1.1}$$

with a non-linearity in the solution and its gradient via the function $f(t,x,y,z)$ defined on $[0,T]\times\mathbb{R}^d\times\mathbb{R}\times\mathbb{R}^d$, a terminal condition g, and a second-order generator $\mathcal{L}$ defined by

$$\mathcal{L}u := \frac{1}{2}\operatorname{Tr}\left(\sigma\sigma^\intercal D_x^2 u\right) + \mu.D_x u.$$

Here μ is a function defined on $[0,T]\times\mathbb{R}^d$ with values in $\mathbb{R}^d$, σ is a function defined on $[0,T]\times\mathbb{R}^d$ with values in $\mathbb{M}^d$ the set of $d\times d$ matrices, and $\mathcal{L}$ is the generator associated to the forward diffusion process:

$$\mathcal{X}_t = x_0 + \int_0^t \mu(s,\mathcal{X}_s)\,\mathrm{d}s + \int_0^t \sigma(s,\mathcal{X}_s)\,\mathrm{d}W_s, \quad 0\le t\le T, \tag{1.2}$$

with W a d-dimensional Brownian motion on some probability space $(\Omega,\mathcal{F},\mathbb{P})$ equipped with a filtration $\mathbb{F} = (\mathcal{F}_t)_{0\le t\le T}$ satisfying the usual conditions.

Due to the so called "curse of dimensionality", the resolution of nonlinear PDEs in high dimension has always been a challenge for scientists. Until recently, only the BSDE (Backward Stochastic Differential Equation) approach first developed in [PP90] was available to tackle this problem: using the time discretization scheme proposed in [BT04], some effective algorithms based on regressions manage to solve non linear PDEs in dimension above 4 (see [GLW05; LGW06]). However this approach is still not implementable in dimension above 6 or 7 : the number of basis functions used for the regression still explodes with the dimension.

Quite recently some new methods have been developed for this problem, and several methodologies have emerged:

- Some are based on the Feyman-Kac representation of the PDE. Branching techniques [HL+16] have been studied and shown to be convergent but only for small maturities and some small nonlinearities. Some effective techniques based on nesting Monte Carlo have been studied in [War18b; War18a]: the convergence is proved for semi-linear equations. Still based on this Feyman-Kac representation some machine learning techniques permitting to solve a fixed point problem have been used recently in [CWNMW19]: numerical results show that it is efficient and some partial demonstrations justify why it is effective.

- Multilevel Picard methods have been developed in [E+18] and [Hut+18] with algorithms based on Picard iterations, multi-level techniques and automatic differentiation. These methods permit to handle some high dimensional PDEs with non linearity in u and its gradient $D_x u$, with convergence results as well as numerous numerical examples showing their efficiency in high dimension.

- Another class of methods is based on the BSDE approach and the curse of dimensionality issue is partially avoided by using some machine learning techniques. The

pioneering papers [HJE18; EHJ17] propose a neural-networks based technique called *Deep BSDE*, which was the first serious attempt for using machine learning methods to solve high dimensional PDEs. Based on an Euler discretization of the forward underlying SDE $\mathcal{X}_t$, the idea is to view the BSDE as a forward SDE, and the algorithm tries to learn the values u and $z = \sigma^\intercal Du$ at each time step of the Euler scheme by minimizing a global loss function between the forward simulation of u till maturity T and the target $g(\mathcal{X}_T)$. This deep learning approximation has been extended to the case of fully nonlinear PDE and second order BSDE in [BEJ19].

- At last, using some machine learning representation of the solution, [SS18] proposes with the so-called Deep Galerkin Method to use the automatic numerical differentiation of the solution to solve the PDE on a finite domain. The authors prove the convergence of their method but without information on the rate of convergence.

Like the second methodology, our approach relies on BSDE representation of the PDE and deep learning approximations: we first discretize the BSDE associated to the PDE by an Euler scheme, but in contrast with [EHJ17], we adopt a classical backward resolution technique. On each time step, we propose to use some machine learning techniques to estimate simultaneously the solution and its gradient by minimizing a loss function defined recursively by backward induction, and solving this local problem by a stochastic gradient algorithm. Two different schemes are designed to deal with the local problems:

(1) The first one tries the estimate the solution and its gradient by a neural network.

(2) The second one tries only to approximate the solution by a neural network while its gradient is estimated directly with some numerical differentiation techniques.

The proposed methodology is then extended to solve some variational inequalities, i.e., free boundary problems related to optimal stopping problems. We mention that the related recent paper [BCJ19] also proposes deep learning method for solving optimal stopping problems, but differently from our method, it relies on the approximation of (randomised) stopping decisions with a sequence of multilayer feedforward neural networks.

Convergence analysis of the two schemes for PDEs and variational inequalities is provided and shows that the approximation error goes to zero as we increase the number of time steps and the number of neurons/layers whenever the gradient descent method used to solve the local problems is not trapped in a local minimum. Notice that similar convergence result for the deep BSDE method has been also obtained in [HL18] with a posteriori error estimation of the solution in terms of the universal approximation capability of global neural networks.

In the last part of the paper, we test our algorithms on different examples. When the solution is easy to represent by a neural network, we can solve the problem in quite high dimension (at least 50 in our numerical tests). We show that the proposed methodology improves the algorithm proposed in [HJE18] that sometimes does not converge or is trapped in a local minimum far away from the true solution. We then show that when the solution has a very complex structure, we can still solve the problem but only in moderate dimension: the neural network used is not anymore able to represent the solution accurately in very high dimension. Finally, we illustrate numerically that the method is effective to solve some system of variational inequalities: we consider the problem of American options and show that it can be solved very accurately in high dimension (we tested until 40).

The outline of the paper is organized as follows. In Section 2, we give a brief and useful reminder for neural networks. We describe in Section 3 our two numerical schemes and compare with the algorithm in [HJE18]. Section 4 is devoted to the convergence analysis of our machine learning algorithms, and we present in Section 5 several numerical tests.

2 Neural networks as function approximators

Multilayer (also called deep) neural networks are designed to approximate unknown or large class of functions. In contrast to additive approximation theory with weighted sum over basis functions, e.g. polynomials, neural networks rely on the composition of simple functions, and appear to provide an efficient way to handle high-dimensional approximation problems, in particular thanks to the increase in computer power for finding the "optimal" parameters by (stochastic) gradient descent methods.

We shall consider feedforward (or artificial) neural networks, which represent the basic type of deep neural networks. Let us recall some notation and basic definitions that will be useful in our context. We fix the input dimension $d_0 = d$ (here the dimension of the state variable x), the output dimension d_1 (here $d_1 = 1$ for approximating the real-valued solution to the PDE, or $d_1 = d$ for approximating the vector-valued gradient function), the global number $L + 1 \in \mathbb{N} \setminus \{1, 2\}$ of layers with m_ℓ, $\ell = 0, \ldots, L$, the number of neurons (units or nodes) on each layer: the first layer is the input layer with $m_0 = d$, the last layer is the output layer with $m_L = d_1$, and the $L - 1$ layers between are called hidden layers, where we choose for simplicity the same dimension $m_\ell = m$, $\ell = 1, \ldots, L - 1$.

A feedforward neural network is a function from $\mathbb{R}^d$ to $\mathbb{R}^{d_1}$ defined as the composition

$$x \in \mathbb{R}^d \longmapsto A_L \circ \varrho \circ A_{L-1} \circ \ldots \circ \varrho \circ A_1(x) \in \mathbb{R}^{d_1}. \tag{2.1}$$

Here A_ℓ, $\ell = 1, \ldots, L$ are affine transformations: A_1 maps from $\mathbb{R}^d$ to $\mathbb{R}^m$, $A_2, \ldots, A_{L-1}$ map from $\mathbb{R}^m$ to $\mathbb{R}^m$, and A_L maps from $\mathbb{R}^m$ to $\mathbb{R}^{d_1}$, represented by

$$A_\ell(x) = \mathcal{W}_\ell x + \beta_\ell,$$

for a matrix $\mathcal{W}_\ell$ called weight, and a vector β_ℓ called bias term, $\varrho : \mathbb{R} \to \mathbb{R}$ is a nonlinear function, called activation function, and applied component-wise on the outputs of A_ℓ, i.e., $\varrho(x_1, \ldots, x_m) = (\varrho(x_1), \ldots, \varrho(x_m))$. Standard examples of activation functions are the sigmoid, the ReLu, the Elu, tanh.

All these matrices $\mathcal{W}_\ell$ and vectors β_ℓ, $\ell = 1, \ldots, L$, are the parameters of the neural network, and can be identified with an element $\theta \in \mathbb{R}^{N_m}$, where $N_m = \sum_{\ell=0}^{L-1} m_\ell(1 + m_{\ell+1})$ $= d(1+m)+m(1+m)(L-2)+m(1+d_1)$ is the number of parameters, where we fix d_0, d_1, L, but allow growing number m of hidden neurons. We denote by Θ_m the set of possible parameters: in the sequel, we shall consider either the case when there are no constraints on parameters, i.e., $\Theta_m = \mathbb{R}^{N_m}$, or when the total variation norm of the neural networks is smaller than γ_m, i.e.,

$$\Theta_m = \Theta_m^\gamma := \big\{\theta = (\mathcal{W}_\ell, \beta_\ell)_\ell : |\mathcal{W}_l| \leq \gamma_m, \ \ \ell = 1, \ldots, L\big\}, \quad \text{with } \gamma_m \nearrow \infty, \text{ as } m \to \infty.$$

We denote by $\Phi_{_m}(.;\theta)$ the neural network function defined in (2.1), and by $\mathcal{NN}^\varrho_{d,d_1,L,m}(\Theta_m)$ the set of all such neural networks $\Phi_{_m}(.;\theta)$ for $\theta \in \Theta_m$, and set

$$\mathcal{NN}^\varrho_{d,d_1,L} \ = \ \bigcup_{m\in\mathbb{N}} \mathcal{NN}^\varrho_{d,d_1,L,m}(\Theta_m) \ = \ \bigcup_{m\in\mathbb{N}} \mathcal{NN}^\varrho_{d,d_1,L,m}(\mathbb{R}^{N_m}),$$

as the class of all neural networks within a fixed structure given by d, d_1, L and ϱ.

The fundamental result of Hornick et al. [HSW89] justifies the use of neural networks as function approximators:

Universal approximation theorem (I): $\mathcal{NN}^{\varrho}_{d,d_1,L}$ is dense in $L^2(\nu)$ for any finite measure ν on $\mathbb{R}^d$, whenever ϱ is continuous and non-constant.

Moreover, we have a universal approximation result for the derivatives in the case of a single hidden layer, i.e. $L = 2$, and when the activation function is a smooth function, see [HSW90].

Universal approximation theorem (II): Assume that ϱ is a (non constant) C^k function. Then, $\mathcal{NN}^{\varrho}_{d,d_1,2}$ approximates any function and its derivatives up to order k, arbitrary well on any compact set of $\mathbb{R}^d$.

3 Deep learning-based schemes for semi-linear PDEs

The starting point for our probabilistic numerical schemes to the PDE (1.1) is the well-known (see [PP90]) nonlinear Feynman-Kac formula via the pair (Y, Z) of $\mathbb{F}$-adapted processes valued in $\mathbb{R} \times \mathbb{R}^d$, solution to the BSDE

$$Y_t = g(\mathcal{X}_T) + \int_t^T f(s, \mathcal{X}_s, Y_s, Z_s)\,\mathrm{d}s - \int_t^T Z_s^\intercal\,\mathrm{d}W_s, \quad 0 \le t \le T, \tag{3.1}$$

related to the solution u of (1.1) via

$$Y_t = u(t, \mathcal{X}_t), \quad 0 \le t \le T,$$

and when u is smooth:

$$Z_t = \sigma^\intercal(t, \mathcal{X}_t) D_x u(t, \mathcal{X}_t), \quad 0 \le t \le T.$$

3.1 The deep BSDE scheme of [HJE18]

The DBSDE algorithm proposed in [HJE18; EHJ17] starts from the BSDE representation (3.1) of the solution to (1.1), but rewritten in forward form as:

$$\begin{aligned} u(t, \mathcal{X}_t) = \; & u(0, x_0) - \int_0^t f(s, \mathcal{X}_s, u(s, \mathcal{X}_s), \sigma^\intercal(s, \mathcal{X}_s) D_x u(s, \mathcal{X}_s))\,\mathrm{d}s \\ & + \int_0^t D_x u(s, \mathcal{X}_s)^\intercal \sigma(s, \mathcal{X}_s)\,\mathrm{d}W_s, \quad 0 \le t \le T. \end{aligned} \tag{3.2}$$

The forward process $\mathcal{X}$ in equation (1.2), when it is not simulatable, is numerically approximated by an Euler scheme $X = X^\pi$ on a time grid: $\pi = \{t_0 = 0 < t_1 < \ldots < t_N = T\}$, with modulus $|\pi| = \max_{i=0,\ldots,N-1} \Delta t_i$, $\Delta t_i := t_{i+1} - t_i$, and defined as

$$X_{t_{i+1}} \; = \; X_{t_i} + \mu(t_i, X_{t_i})\Delta t_i + \sigma(t_i, X_{t_i})\Delta W_{t_i}, \quad i = 0, \ldots, N-1, \; X_0 = x_0, \tag{3.3}$$

where we set $\Delta W_{t_i} := W_{t_{i+1}} - W_{t_i}$. To alleviate notations, we omit the dependence of $X = X^\pi$ on the time grid π as there is no ambiguity (recall that we use the notation $\mathcal{X}$ for

the forward diffusion process). The approximation of equation (1.1) is then given formally from the Euler scheme associated to the forward representation (3.2) by

$$u(t_{i+1}, X_{t_{i+1}}) \approx F(t_i, X_{t_i}, u(t_i, X_{t_i}), \sigma^\intercal(t_i, X_{t_i}) D_x u(t_i, X_{t_i}), \Delta t_i, \Delta W_{t_i}) \tag{3.4}$$

with

$$F(t, x, y, z, h, \Delta) := y - f(t, x, y, z)h + z^\intercal \Delta.$$

In [HJE18; EHJ17], the numerical approximation of $u(t_i, X_{t_i})$ is designed as follows: starting from an estimation $\mathcal{U}_0$ of $u(0, X_0)$, and then using at each time step t_i, $i = 0, \dots, N-1$, a multilayer neural network $x \in \mathbb{R}^d \mapsto \mathcal{Z}_i(x; \theta_i)$ with parameter θ_i for the approximation of $x \mapsto \sigma^\intercal(t_i, x) D_x u(t_i, x)$:

$$\mathcal{Z}_i(x; \theta_i) \approx \sigma^\intercal(t_i, x) D_x u(t_i, x), \tag{3.5}$$

one computes estimations $\mathcal{U}_i$ of $u(t_i; X_{t_i})$ by forward induction via:

$$\mathcal{U}_{i+1} = F(t_i, X_{t_i}, \mathcal{U}_i, \mathcal{Z}_i(X_{t_i}; \theta_i), \Delta t_i, \Delta W_{t_i}),$$

for $i = 0, \dots, N-1$. This algorithm forms a global deep neural network composed of the neural networks (3.5) of each period, by taking as input data (in machine learning language) the paths of $(X_{t_i})_{i=0,\dots,N}$ and $(W_{t_i})_{i=0,\dots,N}$, and giving as output $\mathcal{U}_N = \mathcal{U}_N(\theta)$, which is a function of the input and of the total set of parameters $\theta = (\mathcal{U}_0, \theta_0, \dots, \theta_{N-1})$. The output aims to match the terminal condition $g(X_{t_N})$ of the BSDE, and one then optimizes over the parameter θ the expected square loss function:

$$\theta \mapsto \mathbb{E}\big|g(X_{t_N}) - \mathcal{U}_N(\theta)\big|^2.$$

This is obtained by stochastic gradient descent-type (SGD) algorithms relying on training input data.

3.2 New schemes: DBDP1 and DBDP2

The proposed scheme is defined from a backward dynamic programming type relation, and has two versions:

(1) First version:

- Initialize from an estimation $\widehat{\mathcal{U}}_N^{(1)}$ of $u(t_N, .)$ with $\widehat{\mathcal{U}}_N^{(1)} = g$
- For $i = N-1, \dots, 0$, given $\widehat{\mathcal{U}}_{i+1}^{(1)}$, use a pair of deep neural networks $(\mathcal{U}_i(.;\theta), \mathcal{Z}_i(.;\theta))$ $\in \mathcal{NN}_{d,1,L,m}^{\varrho}(\mathbb{R}^{N_m}) \times \mathcal{NN}_{d,d,L,m}^{\varrho}(\mathbb{R}^{N_m})$ for the approximation of $(u(t_i, .), \sigma^\intercal(t_i, .) D_x u(t_i, .))$, and compute (by SGD) the minimizer of the expected quadratic loss function

$$\begin{cases} \hat{L}_i^{(1)}(\theta) := \mathbb{E}\Big|\widehat{\mathcal{U}}_{i+1}^{(1)}(X_{t_{i+1}}) - F(t_i, X_{t_i}, \mathcal{U}_i(X_{t_i}; \theta), \mathcal{Z}_i(X_{t_i}; \theta), \Delta t_i, \Delta W_{t_i})\Big|^2 \\ \qquad \theta_i^* \in \arg\min_{\theta \in \mathbb{R}^{N_m}} \hat{L}_i^1(\theta). \end{cases} \tag{3.6}$$

  Then, update: $\widehat{\mathcal{U}}_i^{(1)} = \mathcal{U}_i(.; \theta_i^*)$, and set $\widehat{\mathcal{Z}}_i^{(1)} = \mathcal{Z}_i(.; \theta_i^*)$.

(2) Second version:

– Initialize with $\widehat{\mathcal{U}}_N^{(2)} = g$

– For $i = N-1, \dots, 0$, given $\widehat{\mathcal{U}}_{i+1}^{(2)}$, use a deep neural network $\mathcal{U}_i(.;\theta) \in \mathcal{NN}_{d,1,L,m}^{\varrho}(\Theta_m)$, and compute (by SGD) the minimizer of the expected quadratic loss function

$$\begin{cases} \hat{L}_i^{(2)}(\theta) := \mathbb{E}\Big|\widehat{\mathcal{U}}_{i+1}^{(2)}(X_{t_{i+1}}) - \\ \qquad\qquad F(t_i, X_{t_i}, \mathcal{U}_i(X_{t_i};\theta), \sigma^\intercal(t_i, X_{t_i})\hat{D}_x\mathcal{U}_i(X_{t_i};\theta), \Delta t_i, \Delta W_{t_i})\Big|^2 \\ \theta_i^* \in \arg\min_{\theta\in\Theta_m} \hat{L}_i^2(\theta), \end{cases} \tag{3.7}$$

where $\hat{D}_x\mathcal{U}_i(.;\theta)$ is the numerical differentiation of $\mathcal{U}_i(.;\theta)$. Then, update: $\widehat{\mathcal{U}}_i^{(2)} = \mathcal{U}_i(.;\theta_i^*)$, and set $\widehat{\mathcal{Z}}_i^{(2)} = \sigma^\intercal(t_i,.)\hat{D}_x\mathcal{U}_i(.;\theta_i^*)$.

Remark 3.1. For the first version of the scheme, one can use independent neural networks, respectively for the approximation of $u(t_i,.)$ and for the approximation of $\sigma^\intercal(t_i,.)D_xu(t_i,.)$. In other words, the parameters are divided into a pair $\theta = (\xi,\eta)$ and we consider neural networks $\mathcal{U}_i(.;\xi)$ and $\mathcal{Z}_i(.;\eta)$. □

In the sequel, we refer to the first and second version of the new scheme above as DBDP1 and DBDP2, where the acronym DBDP stands for deep learning backward dynamic programming.

The intuition behind DBDP1 and DBDP2 is the following. For simplicity, take $f = 0$, so that $F(t,x,y,z,h,\Delta) = y + z^\intercal\Delta$. The solution u to the PDE (1.1) should then approximately satisfy (see (3.4))

$$u(t_{i+1}, X_{t_{i+1}}) \approx\ u(t_i, X_{t_i}) + D_xu(t_i, X_{t_i})^\intercal\sigma(t_i, X_{t_i})\Delta W_{t_i}.$$

Consider the first scheme DBDP1, and suppose that at time $i+1$, $\widehat{\mathcal{U}}_{i+1}^{(1)}$ is an estimation of $u(t_{i+1,.})$. The quadratic loss function at time i is then approximately equal to

$$\begin{aligned} \hat{L}_i^{(1)}(\theta) &\approx\ \mathbb{E}\Big|u(t_{i+1}, X_{t_{i+1}}) - \mathcal{U}_i(X_{t_i};\theta) - \mathcal{Z}_i(X_{t_i};\theta)^\intercal\Delta W_{t_i}\Big|^2 \\ &\approx\ \mathbb{E}\Big[\big|u(t_i, X_{t_i}) - \mathcal{U}_i(X_{t_i};\theta)\big|^2 + \Delta t_i\big|\sigma^\intercal(t_i, X_{t_i})D_xu(t_i, X_{t_i}) - \mathcal{Z}_i(X_{t_i};\theta)\big|^2\Big]. \end{aligned}$$

Therefore, by minimizing over θ this quadratic loss function, via SGD based on simulations of $(X_{t_i}, X_{t_{i+1}}, \Delta W_{t_i})$ (called training data in the machine learning language), one expects the neural networks $\mathcal{U}_i$ and $\mathcal{Z}_i$ to learn/approximate better and better the functions $u(t_i,.)$ and $\sigma^\intercal(t_i,)D_xu(t_i,)$ in view of the universal approximation theorem [HSW90]. Similarly, the second scheme DPDP2, which uses only neural network on the value functions, learns $u(t_i,.)$ by means of the neural network $\mathcal{U}_i$, and $\sigma^\intercal(t_i,)D_xu(t_i,)$ via $\sigma^\intercal(t_i,)\hat{D}_x\mathcal{U}_i$. The rigorous arguments for the convergence of these schemes will be derived in the next section.

The advantages of our two schemes, compared to the Deep BSDE algorithm, are the following:

- by decomposing the global problem into smaller ones, we may expect to help the gradient descent method to provide estimations closer to the real solution. The memory needed in [HJE18] can be a problem when taking too many time steps.

- at each time step, we initialize the weights and bias of the neural network to the weights and bias of the previous time step treated : this trick is commonly used in iterative solvers of PDE, and allows us to start with a value close to the solution, hence avoiding local minima which are too far away from the true solution. Besides the number of gradient iterations to achieve is rather small after the first resolution step.

The small disadvantage is due to the Tensorflow structure. As it is done in python, the global graph creation takes much time as it is repeated for each time step and the global resolution is a little bit time consuming : as the dimension of the problem increases, the time difference decreases and it becomes hard to compare the computational time for a given accuracy when the dimension is above 5.

3.3 Extension to variational inequalities: scheme RDBDP

Let us consider a variational inequality in the form

$$\begin{cases} \min\big[-\partial_t u - \mathcal{L}u - f(t,x,u,\sigma^\intercal D_x u), u-g\big] = 0, \qquad t\in[0,T),\ x\in\mathbb{R}^d, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad u(T,x) = g(x), \quad x\in\mathbb{R}^d. \end{cases} \tag{3.8}$$

which arises, e.g., in optimal stopping problem and American option pricing in finance. It is known, see e.g. [EK+97], that such variational inequality is related to reflected BSDE of the form

$$\begin{aligned} Y_t =&\ g(\mathcal{X}_T) + \int_t^T f(s,\mathcal{X}_s,Y_s,Z_s)\,\mathrm{d}s - \int_t^T Z_s^\intercal\,\mathrm{d}W_s + K_T - K_t, \\ Y_t \geq&\ g(X_t), \quad 0\leq t\leq T, \end{aligned} \tag{3.9}$$

where K is an adapted non-decreasing process satisfying

$$\int_0^T \big(Y_t - g(X_t)\big)dK_t = 0.$$

The extension of our DBDP1 scheme for such variational inequality, and refereed to as RDBDP scheme, becomes

- Initialize $\widehat{\mathcal{U}}_N = g$
- For $i = N-1,\dots,0$, given $\widehat{\mathcal{U}}_{i+1}$, use a pair of (multilayer) neural network $(\mathcal{U}_i(.;\theta),\mathcal{Z}_i(.;\theta)) \in \mathcal{NN}^{\varrho}_{d,1,L,m}(\mathbb{R}^{N_m}) \times \mathcal{NN}^{\varrho}_{d,d,L,m}(\mathbb{R}^{N_m})$, and compute (by SGD) the minimizer of the expected quadratic loss function

$$\begin{cases} \hat{L}_i(\theta) := \mathbb{E}\big|\widehat{\mathcal{U}}_{i+1}(X_{t_{i+1}}) - F(t_i,X_{t_i},\mathcal{U}_i(X_{t_i};\theta),\mathcal{Z}_i(X_{t_i};\theta),\Delta t_i,\Delta W_{t_i})\big|^2 \\ \quad \theta_i^* \in \arg\min_{\theta\in\mathbb{R}^{N_m}} \hat{L}_i(\theta). \end{cases} \tag{3.10}$$

  Then, update: $\widehat{\mathcal{U}}_i = \max\big[\mathcal{U}_i(.;\theta_i^*), g\big]$, and set $\hat{\mathcal{Z}}_i = \mathcal{Z}(.;\theta_i^*)$.

4 Convergence analysis

The main goal of this section is to prove convergence of the DBDP schemes towards the solution (Y,Z) to the BSDE (3.1) (or reflected BSDE (3.9) for variational inequalities), and to provide a rate of convergence that depends on the approximation errors by neural networks.

4.1 Convergence of DBDP1

We assume the standard Lipschitz conditions on μ and σ, which ensures the existence and uniqueness of an adapted solution $\mathcal{X}$ to the forward SDE (1.2) satisfying for any $p > 1$,

$$\mathbb{E}\big[\sup_{0\le t\le T}|\mathcal{X}_t|^p\big] < C_p(1+|x_0|^p), \tag{4.1}$$

for some constant C_p depending only on p, b, σ and T. Moreover, we have the well-known error estimate with the Euler scheme $X = X^\pi$ defined in (3.3) with a time grid $\pi = \{t_0 = 0 < t_1 < \ldots < t_N = T\}$, with modulus $|\pi|$ s.t. $N|\pi|$ is bounded by a constant depending only on T (hence independent of N):

$$\max_{i=0,\ldots,N-1}\mathbb{E}\Big[|\mathcal{X}_{t_{i+1}} - X_{t_{i+1}}|^2 + \sup_{t\in[t_i,t_{i+1}]}|\mathcal{X}_t - X_{t_i}|^2\Big] = O(|\pi|). \tag{4.2}$$

Here, the standard notation $O(|\pi|)$ means that $\limsup_{|\pi|\to 0}\ |\pi|^{-1}O(|\pi|) < \infty$.

We shall make the standing usual assumptions on the driver f and the terminal data g.

(H1) (i) There exists a constant $[f]_{_L} > 0$ such that the driver f satisfies:

$$|f(t_2,x_2,y_2,z_2) - f(t_1,x_1,y_1,z_1)| \le [f]_{_L}\Big(|t_2-t_1|^{1/2} + |x_2-x_1| + |y_2-y_1| + |z_2-z_1|\Big),$$

for all (t_1,x_1,y_1,z_1) and $(t_2,x_2,y_2,z_2) \in [0,T]\times\mathbb{R}^d\times\mathbb{R}\times\mathbb{R}^d$. Moreover,

$$\sup_{0\le t\le T}|f(t,0,0,0)| < \infty.$$

(ii) The function g satisfies a linear growth condition.

Recall that Assumption **(H1)** ensures the existence and uniqueness of an adapted solution (Y,Z) to (3.1) satisfying

$$\mathbb{E}\Big[\sup_{0\le t\le T}|Y_t|^2 + \int_0^T |Z_t|^2\,\mathrm{d}t\Big] < \infty.$$

From the linear growth condition on f in **(H1)**, and (4.1), we also see that

$$\mathbb{E}\Big[\int_0^T |f(t,\mathcal{X}_t,Y_t,Z_t)|^2\,\mathrm{d}t\Big] < \infty. \tag{4.3}$$

Moreover, we have the standard L^2-regularity result on Y:

$$\max_{i=0,\ldots,N-1}\mathbb{E}\Big[\sup_{t\in[t_i,t_{i+1}]}|Y_t - Y_{t_i}|^2\Big] = O(|\pi|). \tag{4.4}$$

Let us also introduce the L^2-regularity of Z:

$$\varepsilon^Z(\pi) := \mathbb{E}\Big[\sum_{i=0}^{N-1}\int_{t_i}^{t_{i+1}}|Z_t - \bar{Z}_{t_i}|^2 dt\Big], \quad \text{with } \bar{Z}_{t_i} := \frac{1}{\Delta t_i}\mathbb{E}_i\Big[\int_{t_i}^{t_{i+1}} Z_t dt\Big],$$

where $\mathbb{E}_i$ denotes the conditional expectation given $\mathcal{F}_{t_i}$. Since $\bar{Z}$ is a L^2-projection of Z, we know that $\varepsilon^Z(\pi)$ converges to zero when $|\pi|$ goes to zero. Moreover, as shown in [Zha04], when the terminal condition g is also Lipschitz, we have

$$\varepsilon^Z(\pi) = O(|\pi|).$$

Let us first investigate the convergence of the scheme DBDP1 in (3.6), and define (implicitly)

$$\left\{\begin{array}{rcl} \widehat{\mathcal{V}}_{t_i} & := & \mathbb{E}_i\big[\widehat{\mathcal{U}}^{(1)}_{i+1}(X_{t_{i+1}})\big] + f(t_i, X_{t_i}, \widehat{\mathcal{V}}_{t_i}, \overline{\widehat{Z}_{t_i}})\Delta t_i \\ \overline{\widehat{Z}_{t_i}} & := & \frac{1}{\Delta t_i}\mathbb{E}_i\Big[\widehat{\mathcal{U}}^{(1)}_{i+1}(X_{t_{i+1}})\Delta W_{t_i}\Big], \end{array}\right. \tag{4.5}$$

for $i = 0, \dots, N-1$. Notice that $\widehat{\mathcal{V}}_{t_i}$ is well-defined for $|\pi|$ small enough (recall that f is Lipschitz) by a fixed point argument. By the Markov property of the discretized forward process $(X_{t_i})_{i=0,\dots,N}$, we note that there exists some deterministic functions $\hat{v}_i$ and $\overline{\hat{z}_i}$ s.t.

$$\widehat{\mathcal{V}}_{t_i} \;=\; \hat{v}_i(X_{t_i}), \text{ and } \;\; \overline{\widehat{Z}_{t_i}} \;=\; \overline{\hat{z}}_i(X_{t_i}), \qquad i = 0, \dots, N-1. \tag{4.6}$$

Moreover, by the martingale representation theorem, there exists an $\mathbb{R}^d$-valued square integrable process $(\widehat{Z}_t)_t$ such that

$$\widehat{\mathcal{U}}^{(1)}_{i+1}(X_{t_{i+1}}) = \; \widehat{\mathcal{V}}_{t_i} - f(t_i, X_{t_i}, \widehat{\mathcal{V}}_{t_i}, \overline{\widehat{Z}_{t_i}})\Delta t_i + \int_{t_i}^{t_{i+1}} \widehat{Z}_s^{\intercal}\, \mathrm{d}W_s, \tag{4.7}$$

and by Itô isometry, we have

$$\overline{\widehat{Z}_{t_i}} = \; \frac{1}{\Delta t_i}\mathbb{E}_i\Big[\int_{t_i}^{t_{i+1}} \widehat{Z}_s\, \mathrm{d}s\Big], \qquad i = 0, \dots, N-1.$$

Let us now define a measure of the (squared) error for the DBDP1 scheme by

$$\mathcal{E}\big[(\widehat{\mathcal{U}}^{(1)}, \widehat{\mathcal{Z}}^{(1)}), (Y, Z)\big] := \max_{i=0,\dots,N-1} \mathbb{E}\big|Y_{t_i} - \widehat{\mathcal{U}}^{(1)}_i(X_{t_i})\big|^2 + \mathbb{E}\Big[\sum_{i=0}^{N-1}\int_{t_i}^{t_{i+1}} \big|Z_t - \widehat{\mathcal{Z}}^{(1)}_i(X_{t_i})\big|^2 dt\Big].$$

Our first main result gives an error estimate of the DBDP1 scheme in terms of the L^2-approximation errors of $\hat{v}_i$ and $\overline{\hat{z}_i}$ by neural networks $\mathcal{U}_i$ and $\mathcal{Z}_i$, $i = 0, \dots, N-1$, assumed to be independent (see Remark 3.1), and defined as

$$\varepsilon^{\mathcal{N},v}_i \;:=\; \inf_{\xi} \mathbb{E}\big|\hat{v}_i(X_{t_i}) - \mathcal{U}_i(X_{t_i}; \xi)\big|^2, \qquad \varepsilon^{\mathcal{N},z}_i \;:=\; \inf_{\eta} \mathbb{E}\big|\overline{\hat{z}}_i(X_{t_i}) - \mathcal{Z}_i(X_{t_i}; \eta)\big|^2.$$

Here, we fix the structure of the neural networks with input dimension d, output dimension $d_1 = 1$ for $\mathcal{U}_i$, and $d_1 = d$ for $\mathcal{Z}_i$, number of layers L, and m neurons for the hidden layers, and the parameters vary in the whole set $\mathbb{R}^{N_m}$ where N_m is the number of parameters. From the universal approximation theorem (I) ([HSW89]), we know that $\varepsilon^{NN,v}_i$ and $\varepsilon^{NN,z}_i$ converge to zero as m goes to infinity, hence can be made arbitrary small for sufficiently large number of neurons.

Theorem 4.1. (Consistency of DBDP1) *Under* **(H1)**, *there exists a constant $C > 0$, independent of π, such that*

$$\begin{aligned} \mathcal{E}\big[(\widehat{\mathcal{U}}^{(1)}, \widehat{\mathcal{Z}}^{(1)}), (Y, Z)\big] \le\; C\Big(&\mathbb{E}\big|g(\mathcal{X}_T) - g(X_T)\big|^2 + |\pi| + \varepsilon^Z(\pi) \\ &+ \sum_{i=0}^{N-1}\big(N\varepsilon^{\mathcal{N},v}_i + \varepsilon^{\mathcal{N},z}_i\big)\Big). \end{aligned} \tag{4.8}$$

Remark 4.1. The error contributions for the DBDP1 scheme in the r.h.s. of estimation (4.8) consists of four terms. The first three terms correspond to the time discretization of BSDE, similarly as in [BT04], [GLW05], namely (i) the strong approximation of the terminal condition (depending on the forward scheme and the terminal data g), and converging to zero, as $|\pi|$ goes to zero, with a rate $|\pi|$ when g is Lipschitz by (4.2) (see [Avi09] for irregular g), (ii) the strong approximation of the forward Euler scheme, and the L^2-regularity of Y, which gives a convergence of order $|\pi|$, (iii) the L^2-regularity of Z, which converges to zero, as $|\pi|$ goes to zero, with a rate $|\pi|$ when g is Lipschitz. Finally, the better the neural networks are able to approximate/learn the functions $\hat{v}_i$ and $\overline{\hat{z}_i}$ at each time $i = 0, \ldots, N-1$, the smaller is the last term in the error estimation. Moreover, given a prescribed accuracy for the neural network approximation error, the number of parameters of the employed deep neural networks grows at most polynomially in the PDE dimension, as recently proved in [Hut+19] in the case of semi-linear heat equations. □

Proof of Theorem 4.1.
In the following, C will denote a positive generic constant independent of π, and that may take different values from line to line.

Step 1. Fix $i \in \{0, \ldots, N-1\}$, and observe by (3.1), (4.5) that

$$Y_{t_i} - \widehat{\mathcal{V}}_{t_i} = \mathbb{E}_i\big[Y_{t_{i+1}} - \widehat{\mathcal{U}}^{(1)}_{i+1}(X_{t_{i+1}})\big] + \mathbb{E}_i\Big[\int_{t_i}^{t_{i+1}} f(t, \mathcal{X}_t, Y_t, Z_t) - f(t_i, X_{t_i}, \widehat{\mathcal{V}}_{t_i}, \overline{\widehat{Z}_{t_i}})\,\mathrm{d}t\Big].$$

By using Young inequality: $(a+b)^2 \le (1+\gamma\Delta t_i)a^2 + (1+\frac{1}{\gamma\Delta t_i})b^2$ for some $\gamma > 0$ to be chosen later, Cauchy-Schwarz inequality, the Lipschitz condition on f in **(H1)**, and the estimation (4.2) on the forward process, we then have

$$\begin{aligned}
\mathbb{E}\big|Y_{t_i} - \widehat{\mathcal{V}}_{t_i}\big|^2 \le\ & (1+\gamma\Delta t_i)\mathbb{E}\Big|\mathbb{E}_i\big[Y_{t_{i+1}} - \widehat{\mathcal{U}}^{(1)}_{i+1}(X_{t_{i+1}})\big]\Big|^2 \\
& + 4[f]_L^2\Delta t_i\big(1+\frac{1}{\gamma\Delta t_i}\big)\Big\{|\Delta t_i|^2 + \mathbb{E}\Big[\int_{t_i}^{t_{i+1}}\big|Y_t - \widehat{\mathcal{V}}_{t_i}\big|^2\,\mathrm{d}t\Big] \\
& \qquad + \mathbb{E}\Big[\int_{t_i}^{t_{i+1}}\big|Z_t - \overline{\widehat{Z}_{t_i}}\big|^2\,\mathrm{d}t\Big]\Big\} \\
\le\ & (1+\gamma\Delta t_i)\mathbb{E}\Big|\mathbb{E}_i\big[Y_{t_{i+1}} - \hat{\mathcal{U}}^{(1)}_{i+1}(X_{t_{i+1}})\big]\Big|^2 \qquad (4.9)\\
& + 4\frac{[f]_L^2}{\gamma}(1+\gamma\Delta t_i)\Big\{C|\pi|^2 + 2\Delta t_i\mathbb{E}\big|Y_{t_i} - \widehat{\mathcal{V}}_{t_i}\big|^2 + \mathbb{E}\Big[\int_{t_i}^{t_{i+1}}\big|Z_t - \overline{\widehat{Z}_{t_i}}\big|^2\,\mathrm{d}t\Big]\Big\},
\end{aligned}$$

where we use in the last inequality the L^2-regularity (4.4) of Y.

Recalling the definition of $\bar{Z}$ as a L^2-projection of Z, we observe that

$$\mathbb{E}\Big[\int_{t_i}^{t_{i+1}}\big|Z_t - \overline{\widehat{Z}_{t_i}}\big|^2\,\mathrm{d}t\Big] = \mathbb{E}\Big[\int_{t_i}^{t_{i+1}}\big|Z_t - \bar{Z}_{t_i}\big|^2\,\mathrm{d}t\Big] + \Delta t_i\mathbb{E}\big|\bar{Z}_{t_i} - \overline{\widehat{Z}_{t_i}}\big|^2. \qquad (4.10)$$

By multiplying equation (3.1) between t_i and t_{i+1} by ΔW_{t_i}, and using Itô isometry, we

have together with (4.5)

$$\begin{aligned}\Delta t_i\big(\bar{Z}_{t_i}-\overline{\widehat{Z}_{t_i}}\big) &= \mathbb{E}_i\big[\Delta W_{t_i}\big(Y_{t_{i+1}}-\widehat{\mathcal{U}}^{(1)}_{i+1}(X_{t_{i+1}})\big)\big]+\mathbb{E}_i\Big[\Delta W_{t_i}\int_{t_i}^{t_{i+1}} f(t,\mathcal{X}_t,Y_t,Z_t)\,\mathrm{d}t\Big]\\ &= \mathbb{E}_i\Big[\Delta W_{t_i}\Big(Y_{t_{i+1}}-\widehat{\mathcal{U}}^{(1)}_{i+1}(X_{t_{i+1}})-\mathbb{E}_i\big[Y_{t_{i+1}}-\widehat{\mathcal{U}}^{(1)}_{i+1}(X_{t_{i+1}})\big]\Big)\Big]\\ &\quad+\ \mathbb{E}_i\Big[\Delta W_{t_i}\int_{t_i}^{t_{i+1}} f(t,\mathcal{X}_t,Y_t,Z_t)\,\mathrm{d}t\Big].\end{aligned}$$

By Cauchy-Schwarz inequality, and law of iterated conditional expectations, this implies

$$\begin{aligned}\Delta t_i\mathbb{E}\big|\bar{Z}_{t_i}-\overline{\widehat{Z}_{t_i}}\big|^2 \le\ & 2d\Big(\mathbb{E}\big|Y_{t_{i+1}}-\widehat{\mathcal{U}}^{(1)}_{i+1}(X_{t_{i+1}})\big|^2-\mathbb{E}\Big|\mathbb{E}_i\big[Y_{t_{i+1}}-\widehat{\mathcal{U}}^{(1)}_{i+1}(X_{t_{i+1}})\big]\Big|^2\Big)\\ &+2d\Delta t_i\mathbb{E}\Big[\int_{t_i}^{t_{i+1}}|f(t,\mathcal{X}_t,Y_t,Z_t)|^2\,\mathrm{d}t\Big].\end{aligned}\tag{4.11}$$

Then, by plugging (4.10) and (4.11) into (4.9), and choosing $\gamma = 8d[f]_L^2$, we have

$$\begin{aligned}\mathbb{E}\big|Y_{t_i}-\widehat{\mathcal{V}}_{t_i}\big|^2 \le\ & C\Delta t_i\mathbb{E}\big|Y_{t_i}-\widehat{\mathcal{V}}_{t_i}\big|^2+(1+\gamma\Delta t_i)\mathbb{E}\big|Y_{t_{i+1}}-\widehat{\mathcal{U}}^{(1)}_{i+1}(X_{t_{i+1}})\big|^2+C|\pi|^2\\ &+\ C\mathbb{E}\Big[\int_{t_i}^{t_{i+1}}\big|Z_t-\bar{Z}_{t_i}\big|^2\,\mathrm{d}t\Big]+C\Delta t_i\mathbb{E}\Big[\int_{t_i}^{t_{i+1}}|f(t,\mathcal{X}_t,Y_t,Z_t)|^2\,\mathrm{d}t\Big],\end{aligned}$$

and thus for $|\pi|$ small enough:

$$\begin{aligned}\mathbb{E}\big|Y_{t_i}-\widehat{\mathcal{V}}_{t_i}\big|^2 \le\ & (1+C|\pi|)\mathbb{E}\big|Y_{t_{i+1}}-\widehat{\mathcal{U}}^{(1)}_{i+1}(X_{t_{i+1}})\big|^2+C|\pi|^2\\ &+C\mathbb{E}\Big[\int_{t_i}^{t_{i+1}}\big|Z_t-\bar{Z}_{t_i}\big|^2\,\mathrm{d}t\Big]+C|\pi|\mathbb{E}\Big[\int_{t_i}^{t_{i+1}}|f(t,\mathcal{X}_t,Y_t,Z_t)|^2\,\mathrm{d}t\Big].\end{aligned}\tag{4.12}$$

Step 2. By using Young inequality in the form: $(a+b)^2 \ge (1-|\pi|)a^2+(1-\frac{1}{|\pi|})b^2 \ge (1-|\pi|)a^2-\frac{1}{|\pi|}b^2$, we have

$$\begin{aligned}\mathbb{E}\big|Y_{t_i}-\widehat{\mathcal{V}}_{t_i}\big|^2 &= \mathbb{E}\big|Y_{t_i}-\widehat{\mathcal{U}}^{(1)}_i(X_{t_i})+\widehat{\mathcal{U}}^{(1)}_i(X_{t_i})-\widehat{\mathcal{V}}_{t_i}\big|^2\\ &\ge\ (1-|\pi|)\mathbb{E}\big|Y_{t_i}-\widehat{\mathcal{U}}^{(1)}_i(X_{t_i})\big|^2-\frac{1}{|\pi|}\mathbb{E}\big|\widehat{\mathcal{U}}^{(1)}_i(X_{t_i})-\widehat{\mathcal{V}}_{t_i}\big|^2.\end{aligned}\tag{4.13}$$

By plugging this last inequality into (4.12), we then get for $|\pi|$ small enough

$$\begin{aligned}\mathbb{E}\big|Y_{t_i}-\widehat{\mathcal{U}}^{(1)}_i(X_{t_i})\big|^2 \le\ & (1+C|\pi|)\mathbb{E}\big|Y_{t_{i+1}}-\widehat{\mathcal{U}}^{(1)}_{i+1}(X_{t_{i+1}})\big|^2+C|\pi|^2\\ &+\ C\mathbb{E}\Big[\int_{t_i}^{t_{i+1}}\big|Z_t-\bar{Z}_{t_i}\big|^2\,\mathrm{d}t\Big]+C|\pi|\mathbb{E}\Big[\int_{t_i}^{t_{i+1}}|f(t,\mathcal{X}_t,Y_t,Z_t)|^2\,\mathrm{d}t\Big]\\ &+\ CN\mathbb{E}\big|\widehat{\mathcal{V}}_{t_i}-\widehat{\mathcal{U}}^{(1)}_i(X_{t_i})\big|^2.\end{aligned}$$

From discrete Gronwall's lemma (or by induction), and recalling the terminal condition $Y_{t_N} = g(\mathcal{X}_T)$, $\widehat{\mathcal{U}}^{(1)}_i(X_{t_N}) = g(X_T)$, the definition $\varepsilon^Z(\pi)$ of the L^2-regularity of Z, and (4.3), this yields

$$\begin{aligned}\max_{i=0,\dots,N-1}\mathbb{E}\big|Y_{t_i}-\widehat{\mathcal{U}}^{(1)}_i(X_{t_i})\big|^2 \le\ & C\mathbb{E}\big|g(\mathcal{X}_T)-g(X_T)\big|^2+C|\pi|+C\varepsilon^Z(\pi)\\ &+\ CN\sum_{i=0}^{N-1}\mathbb{E}\big|\widehat{\mathcal{V}}_{t_i}-\widehat{\mathcal{U}}^{(1)}_i(X_{t_i})\big|^2.\end{aligned}\tag{4.14}$$

Step 3. Fix $i \in \{0, \ldots, N-1\}$. By using relation (4.7) in the expression of the expected quadratic loss function in (3.6), and recalling the definition of $\overline{\widehat{Z}_{t_i}}$ as a L^2-projection of $\widehat{Z}_t$, we have for all parameters $\theta = (\xi, \eta)$ of the neural networks $\mathcal{U}_i(.;\xi)$ and $\mathcal{Z}_i(.;\eta)$

$$\hat{L}_i^{(1)}(\theta) = \tilde{L}_i(\theta) + \mathbb{E}\Big[\int_{t_i}^{t_{i+1}} \big|\widehat{Z}_t - \overline{\widehat{Z}_{t_i}}\big|^2 \,\mathrm{d}t\Big] \tag{4.15}$$

with

$$\begin{aligned}\tilde{L}_i(\theta) := \; & \mathbb{E}\Big|\widehat{\mathcal{V}}_{t_i} - \mathcal{U}_i(X_{t_i};\xi) + \big(f(t_i, X_{t_i}, \mathcal{U}_i(X_{t_i};\xi), \mathcal{Z}_i(X_{t_i};\eta)) - f(t_i, X_{t_i}, \widehat{\mathcal{V}}_{t_i}, \overline{\widehat{Z}_{t_i}})\big)\Delta t_i\Big|^2 \\ & + \; \Delta t_i \mathbb{E}\big|\overline{\widehat{Z}_{t_i}} - \mathcal{Z}_i(X_{t_i};\eta)\big|^2.\end{aligned}$$

By using Young inequality: $(a+b)^2 \leq (1+\gamma\Delta t_i)a^2 + (1+\frac{1}{\gamma\Delta t_i})b^2$, together with the Lipschitz condition on f in **(H1)**, we clearly see that

$$\tilde{L}_i(\theta) \leq \; (1 + C\Delta t_i)\mathbb{E}\big|\widehat{\mathcal{V}}_{t_i} - \mathcal{U}_i(X_{t_i};\xi)\big|^2 + C\Delta t_i \mathbb{E}\big|\overline{\widehat{Z}_{t_i}} - \mathcal{Z}_i(X_{t_i};\eta)\big|^2. \tag{4.16}$$

On the other hand, using Young inequality in the form: $(a+b)^2 \geq (1-\gamma\Delta t_i)a^2 + (1-\frac{1}{\gamma\Delta t_i})b^2$ $\geq (1-\gamma\Delta t_i)a^2 - \frac{1}{\gamma\Delta t_i}b^2$, together with the Lipschitz condition on f, we have

$$\begin{aligned}\tilde{L}_i(\theta) \geq \; & (1-\gamma\Delta t_i)\mathbb{E}\big|\widehat{\mathcal{V}}_{t_i} - \mathcal{U}_i(X_{t_i};\xi)\big|^2 - \frac{2\Delta t_i [f]_L^2}{\gamma}\Big(\mathbb{E}\big|\widehat{\mathcal{V}}_{t_i} - \mathcal{U}_i(X_{t_i};\xi)\big|^2 + \mathbb{E}\big|\overline{\widehat{Z}_{t_i}} - \mathcal{Z}_i(X_{t_i};\eta)\big|^2\Big) \\ & + \; \Delta t_i \mathbb{E}\big|\overline{\widehat{Z}_{t_i}} - \mathcal{Z}_i(X_{t_i};\eta)\big|^2.\end{aligned}$$

By choosing $\gamma = 4[f]_L^2$, this yields

$$\tilde{L}_i(\theta) \geq \; (1 - C\Delta t_i)\mathbb{E}\big|\widehat{\mathcal{V}}_{t_i} - \mathcal{U}_i(X_{t_i};\xi)\big|^2 + \frac{\Delta t_i}{2}\mathbb{E}\big|\overline{\widehat{Z}_{t_i}} - \mathcal{Z}_i(X_{t_i};\eta)\big|^2. \tag{4.17}$$

Step 4. Fix $i \in \{0, \ldots, N-1\}$, and take $\theta_i^* = (\xi_i^*, \eta_i^*) \in \arg\min_\theta \hat{L}_i^{(1)}(\theta)$ so that $\widehat{\mathcal{U}}_i^{(1)} = \mathcal{U}_i(.;\xi_i^*)$, and $\widehat{\mathcal{Z}}_i^{(1)} = \mathcal{Z}_i(.;\eta_i^*)$. By (4.15), notice that $\theta_i^* \in \arg\min_\theta \tilde{L}_i(\theta)$. From (4.17) and (4.16), we then have for all $\theta = (\xi, \eta)$

$$\begin{aligned}& (1 - C\Delta t_i)\mathbb{E}\big|\widehat{\mathcal{V}}_{t_i} - \widehat{\mathcal{U}}_i^{(1)}(X_{t_i})\big|^2 + \frac{\Delta t_i}{2}\mathbb{E}\big|\overline{\widehat{Z}_{t_i}} - \widehat{\mathcal{Z}}_i^{(1)}(X_{t_i})\big|^2 \\ & \qquad \leq \tilde{L}_i(\theta_i^*) \; \leq \tilde{L}_i(\theta) \; \leq \; (1 + C\Delta t_i)\mathbb{E}\big|\widehat{\mathcal{V}}_{t_i} - \mathcal{U}_i(X_{t_i};\xi)\big|^2 + C\Delta t_i \mathbb{E}\big|\overline{\widehat{Z}_{t_i}} - \mathcal{Z}_i(X_{t_i};\eta)\big|^2.\end{aligned}$$

For $|\pi|$ small enough, and recalling (4.6), this implies

$$\mathbb{E}\big|\widehat{\mathcal{V}}_{t_i} - \widehat{\mathcal{U}}_i^{(1)}(X_{t_i})\big|^2 + \Delta t_i \mathbb{E}\big|\overline{\widehat{Z}_{t_i}} - \widehat{\mathcal{Z}}_i^{(1)}(X_{t_i})\big|^2 \leq \; C\varepsilon_i^{\mathcal{N},v} + C\Delta t_i \varepsilon_i^{\mathcal{N},z}. \tag{4.18}$$

Plugging this last inequality into (4.14), we obtain

$$\begin{aligned}\max_{i=0,\ldots,N-1} \mathbb{E}\big|Y_{t_i} - \widehat{\mathcal{U}}_i^{(1)}(X_{t_i})\big|^2 \leq \; & C\mathbb{E}\big|g(\mathcal{X}_T) - g(X_T)\big|^2 + C|\pi| + C\varepsilon^Z(\pi) \\ & + \; C\sum_{i=0}^{N-1}\big(N\varepsilon_i^{\mathcal{N},v} + \varepsilon_i^{\mathcal{N},z}\big),\end{aligned} \tag{4.19}$$

which proves the consistency of the Y-component in (4.8).

Step 5. Let us finally prove the consistency of the Z-component. From (4.10) and (4.11), we have for any $i = 0, \dots, N-1$:

$$\mathbb{E}\Big[\int_{t_i}^{t_{i+1}} \big|Z_t - \overline{\widehat{Z}_{t_i}}\big|^2 \,\mathrm{d}t\Big] \le \mathbb{E}\Big[\int_{t_i}^{t_{i+1}} \big|Z_t - \bar{Z}_{t_i}\big|^2 \,\mathrm{d}t\Big] + 2d|\pi|\mathbb{E}\Big[\int_{t_i}^{t_{i+1}} |f(t, \mathcal{X}_t, Y_t, Z_t)|^2 \,\mathrm{d}t\Big] + 2d\Big(\mathbb{E}\big|Y_{t_{i+1}} - \widehat{\mathcal{U}}_{i+1}^{(1)}(X_{t_{i+1}})\big|^2 - \mathbb{E}\Big|\mathbb{E}_i\big[Y_{t_{i+1}} - \widehat{\mathcal{U}}_{i+1}^{(1)}(X_{t_{i+1}})\big]\Big|^2\Big)$$

By summing over $i = 0, \dots, N-1$, we get (recall (4.3))

$$\mathbb{E}\Big[\sum_{i=0}^{N-1}\int_{t_i}^{t_{i+1}} \big|Z_t - \overline{\widehat{Z}_{t_i}}\big|^2 \,\mathrm{d}t\Big] \le \varepsilon^Z(\pi) + C|\pi| + 2d\mathbb{E}\big|g(\mathcal{X}_T) - g(X_T)\big|^2 + 2d\sum_{i=0}^{N-1}\Big(\mathbb{E}\big|Y_{t_i} - \widehat{\mathcal{U}}_i^{(1)}(X_{t_i})\big|^2 - \mathbb{E}\Big|\mathbb{E}_i\big[Y_{t_{i+1}} - \widehat{\mathcal{U}}_{i+1}^{(1)}(X_{t_{i+1}})\big]\Big|^2\Big) \tag{4.20}$$

where we change the indices in the last summation. Now, from (4.9), (4.13), we have

$$\begin{aligned}
&2d\Big(\mathbb{E}\big|Y_{t_i} - \widehat{\mathcal{U}}_i^{(1)}(X_{t_i})\big|^2 - \mathbb{E}\Big|\mathbb{E}_i\big[Y_{t_{i+1}} - \widehat{\mathcal{U}}_{i+1}^{(1)}(X_{t_{i+1}})\big]\Big|^2\Big)\\
&\quad\le \Big(\frac{1+\gamma|\pi|}{1-|\pi|} - 1\Big)\mathbb{E}\Big|\mathbb{E}_i\big[Y_{t_{i+1}} - \hat{\mathcal{U}}_{i+1}^{(1)}(X_{t_{i+1}})\big]\Big|^2\\
&\qquad+ \frac{8d[f]_L^2}{\gamma}\frac{1+\gamma|\pi|}{1-|\pi|}\Big\{C|\pi|^2 + |\pi|\mathbb{E}\big|Y_{t_i} - \widehat{\mathcal{V}}_{t_i}\big|^2 + \mathbb{E}\Big[\int_{t_i}^{t_{i+1}}\big|Z_t - \overline{\widehat{Z}_{t_i}}\big|^2\,\mathrm{d}t\Big]\Big\}\\
&\qquad+ \frac{2d}{|\pi|(1-|\pi|)}\mathbb{E}\big|\widehat{\mathcal{U}}_i^{(1)}(X_{t_i}) - \widehat{\mathcal{V}}_{t_i}\big|^2.
\end{aligned}$$

We now choose $\gamma = 24d[f]_L^2$ so that $\frac{8d[f]_L^2}{\gamma}(1+\gamma|\pi|)/(1-|\pi|) \le 1/2$ for $|\pi|$ small enough, and by plugging into (4.20), we obtain (note also that $\big[(1+\gamma|\pi|)/(1-|\pi|) - 1\big] = O(|\pi|)$):

$$\begin{aligned}
\frac{1}{2}\mathbb{E}\Big[\sum_{i=0}^{N-1}\int_{t_i}^{t_{i+1}} \big|Z_t - \overline{\widehat{Z}_{t_i}}\big|^2 \,\mathrm{d}t\Big] &\le \varepsilon^Z(\pi) + C|\pi| + C\max_{i=0,\dots,N}\mathbb{E}\big|Y_{t_i} - \widehat{\mathcal{U}}_i^{(1)}(X_{t_i})\big|^2\\
&\quad+ \frac{1}{2}|\pi|\sum_{i=0}^{N-1}\mathbb{E}\big|Y_{t_i} - \widehat{\mathcal{V}}_{t_i}\big|^2 + CN\sum_{i=0}^{N-1}\mathbb{E}\big|\widehat{\mathcal{U}}_i^{(1)}(X_{t_i}) - \widehat{\mathcal{V}}_{t_i}\big|^2\\
&\le C\varepsilon^Z(\pi) + C|\pi| + C\max_{i=0,\dots,N}\mathbb{E}\big|Y_{t_i} - \widehat{\mathcal{U}}_i^{(1)}(X_{t_i})\big|^2\\
&\quad+ CN\sum_{i=0}^{N-1}\mathbb{E}\big|\widehat{\mathcal{U}}_i^{(1)}(X_{t_i}) - \widehat{\mathcal{V}}_{t_i}\big|^2\\
&\le C\mathbb{E}\big|g(\mathcal{X}_T) - g(X_T)\big|^2 + C|\pi| + C\varepsilon^Z(\pi)\\
&\quad+ C\sum_{i=0}^{N-1}\big(N\varepsilon_i^{\mathcal{N},v} + \varepsilon_i^{\mathcal{N},z}\big),
\end{aligned} \tag{4.21}$$

where we used (4.12) and (4.3) in the second inequality, and (4.18) and (4.19) in the last inequality.

By writing that

$$\mathbb{E}\Big[\int_{t_i}^{t_{i+1}} \big|Z_t - \widehat{\mathcal{Z}}_i^{(1)}(X_{t_i})\big|^2 \,\mathrm{d}t\Big] \le 2\mathbb{E}\Big[\int_{t_i}^{t_{i+1}} \big|Z_t - \overline{\widehat{Z}_{t_i}}\big|^2 \,\mathrm{d}t\Big] + 2\Delta t_i \mathbb{E}\big|\overline{\widehat{Z}_{t_i}} - \widehat{\mathcal{Z}}_i^{(1)}(X_{t_i})\big|^2,$$

and using (4.18), (4.21), we obtain after summation over $i = 0, \ldots, N-1$, the required error estimate for the Z-component as in (4.19), and this ends the proof. □

4.2 Convergence of DBDP2

We shall consider neural networks with one hidden layer, m neurons with total variation smaller than γ_m (see Section 2), a C^3 activation function ϱ with linear growth condition, and bounded derivatives, e.g., a sigmoid activation function, or a tanh function: this class of neural networks is then represented by the parametric set of functions

$$\mathcal{NN}^{\varrho}_{d,1,2,m}(\Theta^{\gamma}_m) := \left\{ x \in \mathbb{R}^d \mapsto \mathcal{U}(x;\theta) = \sum_{i=1}^m c_i \varrho(a_i.x + b_i) + b_0,\ \theta = (a_i, b_i, c_i, b_0)_{i=1}^m \ \in\ \Theta^{\gamma}_m \right\},$$

with

$$\Theta^{\gamma}_m := \left\{ \theta = (a_i, b_i, c_i, b_0)_{i=1}^m : \max_{i=1,\ldots,m} |a_i| \le \gamma_m,\ \sum_{i=1}^m |c_i| \le \gamma_m \right\},$$

for some sequence $(\gamma_m)_m$ converging to ∞, as m goes to infinity, and such that

$$\tfrac{\gamma_m^6}{N} \xrightarrow[m,N\to\infty]{} 0. \tag{4.22}$$

Notice that the neural networks in $\mathcal{NN}^{\varrho}_{d,1,2,m}(\Theta^{\gamma}_m)$ have their first, second and third derivatives uniformly bounded w.r.t. the state variable x. More precisely, there exists some constant C depending only on d and the derivatives of ϱ s.t. for any $\mathcal{U} \in \mathcal{NN}^{\varrho}_{d,1,2,m}(\Theta^{\gamma}_m)$,

$$\begin{cases} \sup_{x\in\mathbb{R}^d, \theta\in\Theta^{\gamma}_m} \Big|D_x \mathcal{U}(x;\theta)\Big| \ \le\ C\gamma_m^2, \quad \sup_{x\in\mathbb{R}^d, \theta\in\Theta^{\gamma}_m} \Big|D_x^2 \mathcal{U}(x;\theta)\Big| \ \le\ C\gamma_m^3, \\ \qquad\qquad\qquad\qquad \text{and} \quad \sup_{x\in\mathbb{R}^d, \theta\in\Theta^{\gamma}_m} \Big|D_x^3 \mathcal{U}(x;\theta)\Big| \ \le\ C\gamma_m^4. \end{cases} \tag{4.23}$$

Let us investigate the convergence of the scheme DBDP2 in (3.7) with neural networks in $\mathcal{NN}^{\varrho}_{d,1,2,m}(\Theta^{\gamma}_m)$, and define for $i = 0, \ldots, N-1$:

$$\begin{cases} \widehat{\mathcal{V}}_{t_i} &:=\ \mathbb{E}_i\big[\widehat{\mathcal{U}}^{(2)}_{i+1}(X_{t_{i+1}})\big] + f(t_i, X_{t_i}, \widehat{\mathcal{V}}_{t_i}, \overline{\widehat{Z}_{t_i}})\Delta t_i \ =\ \hat{v}_i(X_{t_i}), \\ \overline{\widehat{Z}_{t_i}} &:=\ \frac{1}{\Delta t_i}\mathbb{E}_i\Big[\widehat{\mathcal{U}}^{(2)}_{i+1}(X_{t_{i+1}})\Delta W_{t_i}\Big] \ =\ \overline{\hat{z}}_i(X_{t_i}). \end{cases} \tag{4.24}$$

A measure of the (squared) error for the DBDP2 scheme is defined similarly as in DBDP1 scheme:

$$\mathcal{E}\big[(\widehat{\mathcal{U}}^{(2)}, \widehat{\mathcal{Z}}^{(2)}), (Y,Z)\big] := \max_{i=0,\ldots,N-1} \mathbb{E}\big|Y_{t_i} - \widehat{\mathcal{U}}_i^{(2)}(X_{t_i})\big|^2 + \mathbb{E}\Big[\sum_{i=0}^{N-1}\int_{t_i}^{t_{i+1}} \big|Z_t - \widehat{\mathcal{Z}}_i^{(2)}(X_{t_i})\big|^2 dt\Big].$$

Our second main result gives an error estimate of the DBDP2 scheme in terms of the L^2-approximation errors of $\hat{v}_i$ and its derivative (which exists under assumption detailed below) by neural networks $\mathcal{U}_i \in \mathcal{NN}^{\varrho}_{d,1,2,m}(\Theta^{\gamma}_m)$, $i = 0, \ldots, N-1$, and defined as

$$\varepsilon_i^{\mathcal{N},m} := \inf_{\theta\in\Theta^{\gamma}_m} \Big\{ \mathbb{E}\big|\hat{v}_i(X_{t_i}) - \mathcal{U}_i(X_{t_i};\theta)\big|^2 + \Delta t_i \mathbb{E}\big|\sigma^{\intercal}(t_i, X_{t_i})\big(D_x\hat{v}_i(X_{t_i}) - D_x\mathcal{U}_i(X_{t_i};\theta)\big)\big|^2 \Big\},$$

which are expected to be small in view of the universal approximation theorem (II), see discussion in Remark 4.2.

We also require the additional conditions on the coefficients:

(H2) (i) The functions $x \mapsto \mu(t,.)$, $\sigma(t,.)$ are C^1 with bounded derivatives uniformly w.r.t. $(t,x) \in [0,T] \times \mathbb{R}^d$.

(ii) The function $(x,y,z) \mapsto f(t,.)$ is C^1 with bounded derivatives uniformly w.r.t. (t,x,y,z) in $[0,T] \times \mathbb{R}^d \times \mathbb{R} \times \mathbb{R}^d$.

Theorem 4.2. (Consistency of DBDP2) *Under* **(H1)**-**(H2)**, *there exists a constant $C > 0$, independent of π, such that*

$$\mathcal{E}\big[(\widehat{\mathcal{U}}^{(2)}, \widehat{\mathcal{Z}}^{(2)}), (Y,Z)\big] \le C\Big(\mathbb{E}\big|g(\mathcal{X}_T) - g(X_T)\big|^2 + \frac{\gamma_m^6}{N} + \varepsilon^Z(\pi) + N\sum_{i=0}^{N-1} \varepsilon_i^{\mathcal{N},m}\Big). \quad (4.25)$$

Proof. For simplicity of notations, we assume $d = 1$, and only detail the arguments that differ from the proof of Theorem 4.8. From (4.24), and the Euler scheme (3.3), we have

$$\begin{aligned} \hat{v}_i(x) &= \tilde{v}_i(x) + \Delta t_i f(t_i, x, \hat{v}_i(x), \overline{\hat{z}}_i(x)), \quad \tilde{v}_i(x) := \mathbb{E}\big[\hat{u}_{i+1}(X^x_{t_{i+1}})\big], \quad x \in \mathbb{R}^d, \\ \overline{\hat{z}}_i(x) &= \frac{1}{\Delta t_i}\mathbb{E}\big[\hat{u}_{i+1}(X^x_{t_{i+1}})\Delta W_{t_i}\big], \quad X^x_{t_{i+1}} = x + \mu(t_i,x)\Delta t_i + \sigma(t_i,x)\Delta W_{t_i}. \end{aligned}$$

Under assumption **(H2)**(i), and recalling that $\hat{u}_{i+1} = \mathcal{U}_{i+1}(.;\theta^*_{i+1})$ is C^2 with bounded derivatives, we see that $\tilde{v}_i$ is C^1 with

$$\begin{aligned} D_x\tilde{v}_i(x) &= \mathbb{E}\Big[\big(1 + D_x\mu(t_i,x)\Delta t_i + D_x\sigma(t_i,x)\Delta W_{t_i}\big)D_x\hat{u}_{i+1}(X^x_{t_{i+1}})\Big] \\ &= \mathbb{E}\big[D_x\hat{u}_{i+1}(X^x_{t_{i+1}})\big] + \Delta t_i\, R_i(x) \qquad (4.26) \\ R_i(x) &:= D_x\mu(t_i,x)\mathbb{E}\big[D_x\hat{u}_{i+1}(X^x_{t_{i+1}})\big] + \sigma(t_i,x)D_x\sigma(t_i,x)\mathbb{E}\big[D^2_x\hat{u}_{i+1}(X^x_{t_{i+1}})\big], \end{aligned}$$

where we use integration by parts in the second equality. Similarly, we have

$$\begin{cases} \overline{\hat{z}}_i(x) = \sigma(t_i,x)\mathbb{E}\big[D_x\hat{u}_{i+1}(X^x_{t_{i+1}})\big], \\ D_x\overline{\hat{z}}_i(x) = D_x\sigma(t_i,x)\mathbb{E}\big[D_x\hat{u}_{i+1}(X^x_{t_{i+1}})\big] + \sigma(t_i,x)\mathbb{E}\big[D^2_x\hat{u}_{i+1}(X^x_{t_{i+1}})\big] + \Delta t_i\, \sigma(t_i,x)G_i(x) \\ G_i(x) := D_x\mu(t_i,x)\mathbb{E}\big[D^2_x\hat{u}_{i+1}(X^x_{t_{i+1}})\big] + \sigma(t_i,x)D_x\sigma(t_i,x)\mathbb{E}\big[D^3_x\hat{u}_{i+1}(X^x_{t_{i+1}})\big]. \end{cases} \quad (4.27)$$

Denoting by $\hat{f}_i(x) = f(t_i,x,\hat{v}_i(x),\overline{\hat{z}}_i(x))$, it follows by the implicit function theorem, and for $|\pi|$ small enough, that $\hat{v}_i$ is C^1 with derivative given by

$$D_x\hat{v}_i(x) = D_x\tilde{v}_i(x) + \Delta t_i\Big(D_x\hat{f}_i(x) + D_y\hat{f}_i(x)D_x\hat{v}_i(x) + D_z\hat{f}_i(x)D_x\overline{\hat{z}}_i(x)\Big)$$

and thus by (4.26)-(4.27)

$$\big(1-\Delta t_i D_y\hat{f}_i(x)\big)\sigma(t_i,x)D_x\hat{v}_i(x) = \overline{\hat{z}_i}(x) + \Delta t_i\sigma(t_i,x)\Big(R_i(x) + D_x\hat{f}_i(x) + D_z\hat{f}_i(x)D_x\overline{\hat{z}_i}(x)\Big).$$

Under **(H2)**, by the linear growth condition on σ, and using the bounds on the derivatives of the neural networks in $\mathcal{NN}^{\varrho}_{d,1,2,m}(\Theta^{\gamma}_m)$ in (4.23), we then have

$$\mathbb{E}\Big|\sigma(t_i,X_{t_i})D_x\hat{v}_i(X_{t_i}) - \overline{\widehat{Z}_{t_i}}\Big|^2 \le C(\gamma_m^6 + |\pi|^2\gamma_m^8)|\pi|^2. \tag{4.28}$$

Next, by the same arguments as in Steps 3 and 4 in the proof of Theorem 4.1 (see in particular (4.18)), we have for $|\pi|$ small enough,

$$\begin{aligned}\mathbb{E}\big|\widehat{\mathcal{V}}_{t_i} - \widehat{\mathcal{U}}_i^{(2)}(X_{t_i})\big|^2 + \Delta t_i\mathbb{E}\big|\overline{\widehat{Z}_{t_i}} - \widehat{\mathcal{Z}}_i^{(2)}(X_{t_i})\big|^2 \\ \le C\mathbb{E}\big[\big|\hat{v}_i(X_{t_i}) - \mathcal{U}_i(X_{t_i};\theta)\big|^2\big] + C\Delta t_i\mathbb{E}\big|\overline{\widehat{Z}_{t_i}} - \sigma(t_i,X_{t_i})\hat{D}_x\mathcal{U}_i(X_{t_i};\theta)\big|^2,\end{aligned}$$

for all $\theta\in\Theta^N$, and then with (4.28), and by definition of $\varepsilon_i^{NN,v,2}$:

$$\mathbb{E}\big|\widehat{\mathcal{V}}_{t_i} - \widehat{\mathcal{U}}_i^{(2)}(X_{t_i})\big|^2 + \Delta t_i\mathbb{E}\big|\overline{\widehat{Z}_{t_i}} - \widehat{\mathcal{Z}}_i^{(2)}(X_{t_i})\big|^2 \le C\varepsilon_i^{NN,v,2} + C(\gamma_m^6 + |\pi|^2\gamma_m^8)|\pi|^3 \tag{4.29}$$

On the other hand, by the same arguments as in Steps 1 and 2 in the proof of Theorem 4.1 (see in particular (4.14)), we have

$$\begin{aligned}\max_{i=0,\dots,N-1}\mathbb{E}\big|Y_{t_i} - \widehat{\mathcal{U}}_i^{(2)}(X_{t_i})\big|^2 \le\ & C\mathbb{E}\big|g(\mathcal{X}_T) - g(X_T)\big|^2 + C|\pi| + C\varepsilon^Z(\pi) \\ &+\ CN\sum_{i=0}^{N-1}\mathbb{E}\big|\widehat{\mathcal{V}}_{t_i} - \widehat{\mathcal{U}}_i^{(2)}(X_{t_i})\big|^2.\end{aligned}$$

Plugging (4.29) into this last inequality, together with (4.22), gives the required estimation (4.25) for the Y-component. Finally, by following the same arguments as in Step 5 in the proof of (4.1), we obtain the estimation (4.25) for the Z-component. □

Remark 4.2. The universal approximation theorem (II) [HSW90] is valid on compact sets, and one cannot conclude *a priori* that the error of network approximation $\varepsilon_i^{\mathcal{N},m}$ converge to zero as m goes to infinity. Instead, we have to proceed into two steps:

(i) Localize the error by considering

$$\varepsilon_i^{\mathcal{N},m,K} := \inf_{\theta\in\Theta_m^{\gamma}}\mathbb{E}\big[\Delta_i(X_{t_i};\theta)1_{|X_{t_i}|\le K}\big],$$

where we set $\Delta_i(x;\theta) := |\hat{v}_i(x) - \mathcal{U}_i(x;\theta)|^2 + \Delta t_i\big|\sigma^{\intercal}(t_i,x)\big(D_x\hat{v}_i(x) - D_x\mathcal{U}_i(x;\theta)\big)\big|^2$.

(ii) Consider an increasing family of neural networks $\Theta_m^{\gamma^{N-1}} \subset \ldots \subset \Theta_m^{\gamma^i} \subset \ldots \subset \Theta_m^{\gamma^0}$ on which to minimize the approximation errors by backward induction at times t_i, $i = N-1,\dots,0$, and where, γ_m^i is defined by

$$\gamma_m^i := \gamma_{\varphi^{N-1-i}(m)},$$

with $\varphi : \mathbb{N} \to \mathbb{N}$ an increasing function, and where we use the notation $\varphi^k := \varphi \circ ... \circ \varphi$ (composition k times).

The localized approximation error at time t_i, for $0 \le i \le N-1$, should then be rewritten as

$$\varepsilon_{i,N}^{\mathcal{N},m,K} := \inf_{\theta \in \Theta_m^{\gamma^i}} \mathbb{E}\big[\Delta_i(X_{t_i};\theta) 1_{|X_{t_i}| \le K}\big],$$

and the non-localized one as

$$\varepsilon_{i,N}^{\mathcal{N},m} := \inf_{\theta \in \Theta_m^{\gamma^i}} \mathbb{E}\big[\Delta_i(X_{t_i};\theta)\big].$$

Note that $\varepsilon_{i,N}^{\mathcal{N},m,K}$ converges to zero, as m goes to infinity, for any $K > 0$, as claimed by the universal approximation theorem (II) [HSW90]. On the other hand, from the expressions of $\hat{v}_i$, $D_x\hat{v}_i$ in the above proof of Theorem 4.2, we see under **(H1)-(H2)**, and from (4.23) that for all $x \in \mathbb{R}^d$, $\theta \in \Theta_m^{\gamma^i}$, $i = 0, \dots, N-1$:

$$|\Delta_i(x;\theta)| \le C(1+|x|^2)\gamma^4_{\varphi^{N-1}(m)},$$

for some positive constant C independent of m, π. We deduce by Cauchy-Schwarz and Chebyshev's inequalities that for all $K > 0$, and $\theta \in \Theta_m^{\gamma^i}$, $i = 0, \dots, N-1$,

$$\mathbb{E}\big[\Delta_i(X_{t_i};\theta) 1_{|X_{t_i}|>K}\big] \le \Big\|\Delta_i(X_{t_i};\theta)\Big\|_2 \frac{\big\|X_{t_i}\big\|_2}{K} \le C(1+|x_0|^3)\frac{\gamma^4_{\varphi^{N-1}(m)}}{K},$$

where we used (4.1) in the last inequality. This shows that

$$\varepsilon_{i,N}^{\mathcal{N},m} \le \varepsilon_{i,N}^{\mathcal{N},m,K} + C\frac{\gamma^4_{\varphi^{N-1}(m)}}{K}, \quad \forall K > 0,$$

and thus, in theory, the error $\varepsilon_{i,N}^{\mathcal{N},m}$ can be made arbitrary small by suitable choices of large m and K. □

4.3 Convergence of RDBDP

In this paragraph, we study the convergence of machine learning schemes for the variational inequality (3.8).

We first consider the case when f does not depend on z, so that the component $Y_t = u(t, \mathcal{X}_t)$ solution to the reflected BSDE (3.9) admits a Snell envelope representation, and we shall focus on the error on Y by proposing an alternative to scheme (3.10), refereed to as RDBDPbis scheme, which only uses neural network for learning the function u:

- Initialize $\widehat{\mathcal{U}}_N = g$
- For $i = N-1, \dots, 0$, given $\widehat{\mathcal{U}}_{i+1}$, use a deep neural network $\mathcal{U}_i(.;\theta) \in \mathcal{NN}^{\varrho}_{d,1,L,m}(\mathbb{R}^{N_m})$, and compute (by SGD) the minimizer of the expected quadratic loss function

$$\begin{cases} \bar{L}_i(\theta) := \mathbb{E}\big|\widehat{\mathcal{U}}_{i+1}(X_{t_{i+1}}) - \mathcal{U}_i(X_{t_i};\theta) + f(t_i, X_{t_i}, \mathcal{U}_i(X_{t_i};\theta))\Delta t_i\big|^2 \\ \quad \theta_i^* \in \arg\min_{\theta \in \mathbb{R}^{N_m}} \bar{L}_i(\theta). \end{cases} \tag{4.30}$$

  Then, update: $\widehat{\mathcal{U}}_i = \max\big[\mathcal{U}_i(.;\theta_i^*), g\big]$.

Let us also define from the scheme (4.30)

$$\begin{cases} \tilde{\mathcal{V}}_{t_i} := \mathbb{E}_i\big[\widehat{\mathcal{U}}_{i+1}(X_{t_{i+1}})\big] + f(t_i, X_{t_i}, \tilde{\mathcal{V}}_{t_i})\Delta t_i \;=\; \tilde{v}_i(X_{t_i}), \\ \widehat{\mathcal{V}}_{t_i} := \max[\tilde{\mathcal{V}}_{t_i}; g(X_{t_i})], \quad i = 0, \dots, N-1. \end{cases} \tag{4.31}$$

Our next result gives an error estimate of the scheme (4.30) in terms of the L^2-approximation errors of $\tilde{v}_i$ by neural networks $\mathcal{U}_i$, $i = 0, \dots, N-1$, and defined as

$$\tilde{\varepsilon}_i^{\mathcal{N}} := \inf_{\theta \in \mathbb{R}^{N_m}} \mathbb{E}\big|\tilde{v}_i(X_{t_i}) - \mathcal{U}_i(X_{t_i}; \theta)\big|^2.$$

Theorem 4.3. (Case f independent of z: Consistency of RDBDPbis) *Let Assumption* **(H1)** *hold, with g Lipschitz. Then, there exists a constant $C > 0$, independent of π, such that*

$$\max_{i=0,\dots,N-1} \big\|Y_{t_i} - \hat{\mathcal{U}}_i(X_{t_i})\big\|_2 \;\le\; C\Big(|\pi|^{\frac{1}{2}} + \sum_{i=0}^{N-1} \sqrt{\tilde{\varepsilon}_i^{\mathcal{N}}}\Big), \tag{4.32}$$

where $\|.\|_2$ is the L^2-norm on $(\Omega, \mathcal{F}, \mathbb{P})$.

Remark 4.3. The estimation (4.32) implies that

$$\max_{i=0,\dots,N-1} \mathbb{E}\big|Y_{t_i} - \hat{\mathcal{U}}_i(X_{t_i})\big|^2 \;\le\; C\Big(|\pi| + N \sum_{i=0}^{N-1} \tilde{\varepsilon}_i^{\mathcal{N}}\Big),$$

which is of the same order than the error estimate in Theorem 4.1 when g is Lipschitz. □

Proof. Let us introduce the discrete-time approximation of the reflected BSDE

$$\begin{cases} Y_{t_N}^{\pi} \;=\; g(X_{t_N}) \\ \tilde{Y}_{t_i}^{\pi} \;=\; \mathbb{E}_i[Y_{t_{i+1}}^{\pi}] + f(t_i, X_{t_i}, \tilde{Y}_{t_i}^{\pi})\Delta t_i \\ Y_{t_i}^{\pi} \;=\; \max\big[\tilde{Y}_{t_i}^{\pi}; g(X_{t_i})\big], \quad i = 0, \dots, N-1. \end{cases} \tag{4.33}$$

It is known, see [BP03], [BT04] that

$$\max_{i=0,\dots,N-1} \big\|Y_{t_i} - Y_{t_i}^{\pi}\big\|_2 \;=\; O(|\pi|^{\frac{1}{2}}). \tag{4.34}$$

Fix $i = 0, \dots, N-1$. From (4.31), (4.33), we have

$$\begin{aligned} |\tilde{Y}_{t_i}^{\pi} - \tilde{\mathcal{V}}_{t_i}| &\le \mathbb{E}_i\big|Y_{t_{i+1}}^{\pi} - \widehat{\mathcal{U}}_{i+1}(X_{t_{i+1}})\big| + \Delta t_i\big|f(t_i, X_{t_i}, \tilde{Y}_{t_i}^{\pi}) - f(t_i, X_{t_i}, \tilde{\mathcal{V}}_{t_i})\big| \\ &\le \mathbb{E}_i\big|Y_{t_{i+1}}^{\pi} - \widehat{\mathcal{U}}_{i+1}(X_{t_{i+1}})\big| + [f]_L \Delta t_i |\tilde{Y}_{t_i}^{\pi} - \tilde{\mathcal{V}}_{t_i}|, \end{aligned}$$

from the Lipschitz condition on f in **(H1)**, and then for $|\pi|$ small enough

$$\big\|\tilde{Y}_{t_i}^{\pi} - \tilde{\mathcal{V}}_{t_i}\big\|_2 \;\le\; (1 + C|\pi|)\big\|Y_{t_{i+1}}^{\pi} - \widehat{\mathcal{U}}_{i+1}(X_{t_{i+1}})\big\|_2.$$

By Minkowski inequality, this yields for all θ

$$\big\|\tilde{Y}_{t_i}^{\pi} - \mathcal{U}_i(X_{t_i}; \theta)\big\|_2 \;\le\; (1 + C|\pi|)\big\|Y_{t_{i+1}}^{\pi} - \widehat{\mathcal{U}}_{i+1}(X_{t_{i+1}})\big\|_2 + \big\|\tilde{\mathcal{V}}_{t_i} - \mathcal{U}_i(X_{t_i}; \theta)\big\|_2. \tag{4.35}$$

On the other hand, by the martingale representation theorem, there exists an $\mathbb{R}^d$-valued square integrable process $(\tilde{Z}_t)_t$ such that

$$\widehat{\mathcal{U}}_{i+1}(X_{t_{i+1}}) = \tilde{\mathcal{V}}_{t_i} - f(t_i, X_{t_i}, \tilde{\mathcal{V}}_{t_i})\Delta t_i + \int_{t_i}^{t_{i+1}} \tilde{Z}_s^\intercal \, \mathrm{d}W_s,$$

and the expected squared loss function of the DBDP3 scheme can be written as

$$\bar{L}_i(\theta) = \tilde{L}_i(\theta) + \mathbb{E}\Big[\int_{t_i}^{t_{i+1}} \big|\tilde{Z}_t\big|^2 \, \mathrm{d}t \Big]$$

with

$$\sqrt{\tilde{L}_i(\theta)} := \Big\| \tilde{\mathcal{V}}_{t_i} - \mathcal{U}_i(X_{t_i};\theta) + \big(f(t_i, X_{t_i}, \mathcal{U}_i(X_{t_i};\theta)) - f(t_i, X_{t_i}, \tilde{\mathcal{V}}_{t_i}) \big)\Delta t_i \Big\|_2.$$

From the Lipschitz condition on f, and by Minkowski inequality, we have for all θ

$$(1 - [f]_L \Delta t_i) \big\| \tilde{\mathcal{V}}_{t_i} - \mathcal{U}_i(X_{t_i};\theta) \big\|_2 \;\le\; \sqrt{\tilde{L}_i(\theta)} \;\le\; (1 + [f]_L \Delta t_i) \big\| \tilde{\mathcal{V}}_{t_i} - \mathcal{U}_i(X_{t_i};\theta) \big\|_2.$$

Take now $\theta_i^* \in \arg\min_\theta \bar{L}_i(\theta) = \arg\min_\theta \tilde{L}_i(\theta)$. Then, from the above relations, we have

$$(1 - [f]_L \Delta t_i) \big\| \tilde{\mathcal{V}}_{t_i} - \mathcal{U}_i(X_{t_i};\theta_i^*) \big\|_2 \;\le\; (1 + [f]_L \Delta t_i) \big\| \tilde{\mathcal{V}}_{t_i} - \mathcal{U}_i(X_{t_i};\theta) \big\|_2,$$

for all θ, and so

$$\big\| \tilde{\mathcal{V}}_{t_i} - \mathcal{U}_i(X_{t_i};\xi_i^*) \big\|_2 \le \; (1 + C|\pi|) \sqrt{\tilde{\varepsilon}_i^{\mathcal{N}}}. \tag{4.36}$$

By taking $\theta = \theta_i^*$ in (4.35), recalling that $\widehat{\mathcal{U}}_i(X_{t_i}) = \max[\mathcal{U}_i(X_{t_i};\theta_i^*); g(X_{t_i})]$, $Y_{t_i}^\pi = \max[\tilde{Y}_{t_i}^\pi; g(X_{t_i})]$, and since $|\max(a,c) - \max(b,c)| \le |a-b|$, we obtain by using (4.36)

$$\big\| Y_{t_i}^\pi - \widehat{\mathcal{U}}_i(X_{t_i}) \big\|_2 \le \; (1 + C|\pi|) \Big(\big\| Y_{t_{i+1}}^\pi - \widehat{\mathcal{U}}_{i+1}(X_{t_{i+1}}) \big\|_2 + \sqrt{\tilde{\varepsilon}_i^{\mathcal{N}}} \Big).$$

By induction, this yields

$$\max_{i=0,\ldots,N-1} \big\| Y_{t_i}^\pi - \widehat{\mathcal{U}}_i(X_{t_i}) \big\|_2 \le \; C \sum_{i=0}^{N-1} \sqrt{\tilde{\varepsilon}_i^{\mathcal{N}}},$$

and we conclude with (4.34). □

We finally turn to the general case when f may depend on z, and study the convergence of the RDBDP scheme (3.10) towards the variational inequality (3.8) related to the solution (Y, Z) of the reflected BSDE (3.9) by showing an error estimate for

$$\mathcal{E}\big[(\widehat{\mathcal{U}}, \widehat{\mathcal{Z}}), (Y, Z)\big] := \max_{i=0,\ldots,N-1} \mathbb{E}\big|Y_{t_i} - \widehat{\mathcal{U}}_i(X_{t_i})\big|^2 + \mathbb{E}\Big[\sum_{i=0}^{N-1} \int_{t_i}^{t_{i+1}} \big|Z_t - \widehat{\mathcal{Z}}_i(X_{t_i})\big|^2 dt \Big].$$

Let us define from the scheme (3.10)

$$\begin{cases} \tilde{\mathcal{V}}_{t_i} := \; \mathbb{E}_i\big[\widehat{\mathcal{U}}_{i+1}(X_{t_{i+1}})\big] + f(t_i, X_{t_i}, \tilde{\mathcal{V}}_{t_i}, \overline{\tilde{Z}_{t_i}})\Delta t_i \;=\; \tilde{v}_i(X_{t_i}), \\ \overline{\tilde{Z}_{t_i}} := \; \dfrac{1}{\Delta t_i} \mathbb{E}_i \Big[\widehat{\mathcal{U}}_{i+1}(X_{t_{i+1}}) \Delta W_{t_i} \Big] \;=\; \tilde{z}_i(X_{t_i}), \\ \widehat{\mathcal{V}}_{t_i} := \; \max[\tilde{\mathcal{V}}_{t_i}; g(X_{t_i})], \quad i = 0, \ldots, N-1. \end{cases} \tag{4.37}$$

Our final main result gives an error estimate of the RDBDP scheme in terms of the L^2-approximation errors of $\tilde{v}_i$ and $\tilde{z}_i$ by neural networks $\mathcal{U}_i$ and $\mathcal{Z}_i$, $i = 0, \dots, N-1$, assumed to be independent (see Remark 3.1), and defined as

$$\varepsilon_i^{\mathcal{N},\tilde{v}} := \inf_{\xi} \mathbb{E}\big|\tilde{v}_i(X_{t_i}) - \mathcal{U}_i(X_{t_i};\xi)\big|^2, \qquad \varepsilon_i^{\mathcal{N},\tilde{z}} := \inf_{\eta} \mathbb{E}\big|\tilde{z}_i(X_{t_i}) - \mathcal{Z}_i(X_{t_i};\eta)\big|^2.$$

The result is obtained under one of the following additional assumptions

(H3) g is C^1, and g, $D_x g$ are Lipschitz.

or

(H4) σ is C^1, with σ, $D_x\sigma$ both Lipschitz, and g is C^2, with g, $D_x g$, $D_x^2 g$ all Lipschitz.

Theorem 4.4. (Consistency of RDBDP) *Let Assumption* **(H1)** *hold. There exists a constant $C > 0$, independent of π, such that*

$$\mathcal{E}\big[(\widehat{\mathcal{U}}, \widehat{\mathcal{Z}}), (Y, Z)\big] \le C\Big(\varepsilon(\pi) + \sum_{i=0}^{N-1} \big(N\varepsilon_i^{\mathcal{N},\tilde{v}} + \varepsilon_i^{\mathcal{N},\tilde{z}}\big)\Big), \tag{4.38}$$

with $\varepsilon(\pi) = O(|\pi|^{\frac{1}{2}})$ under **(H3)**, *and $\varepsilon(\pi) = O(|\pi|)$ under* **(H4)**.

Proof. Let us introduce the discrete-time approximation of the reflected BSDE

$$\begin{cases} Y_{t_N}^\pi = g(X_{t_N}) \\ Z_{t_i}^\pi = \dfrac{1}{\Delta t_i}\mathbb{E}_i\big[Y_{t_{i+1}}^\pi \Delta W_{t_i}\big], \\ \tilde{Y}_{t_i}^\pi = \mathbb{E}_i[Y_{t_{i+1}}^\pi] + f(t_i, X_{t_i}, \tilde{Y}_{t_i}^\pi, Z_{t_i}^\pi)\Delta t_i \\ Y_{t_i}^\pi = \max\big[\tilde{Y}_{t_i}^\pi; g(X_{t_i})\big], \quad i = 0, \dots, N-1. \end{cases} \tag{4.39}$$

It is known from [BC08] that

$$\begin{cases} \max\limits_{i=0,\dots,N-1} \mathbb{E}\big|Y_{t_i} - Y_{t_i}^\pi\big|^2 = \varepsilon(\pi) \\ \mathbb{E}\Big[\sum\limits_{i=0}^{N-1} \int_{t_i}^{t_{i+1}} \big|Z_t - Z_{t_i}^\pi\big|^2 dt\Big] = O(|\pi|^{\frac{1}{2}}), \end{cases} \tag{4.40}$$

with $\varepsilon(\pi) = O(|\pi|^{\frac{1}{2}})$ under **(H3)**, and $\varepsilon(\pi) = O(|\pi|)$ under **(H4)**.

Fix $i = 0, \dots, N-1$. By writing that

$$\tilde{Y}_{t_i}^\pi - \tilde{\mathcal{V}}_{t_i} = \mathbb{E}_i\big[Y_{t_{i+1}} - \widehat{\mathcal{U}}_{i+1}(X_{t_{i+1}})\big] + \Delta t_i\Big(f(t_i, X_{t_i}, \tilde{Y}_{t_i}^\pi, Z_{t_i}^\pi) - f(t_i, X_{t_i}, \tilde{\mathcal{V}}_{t_i}, \overline{\tilde{Z}_{t_i}})\Big),$$

and proceeding similarly as in Step 1 in the proof of Theorem 4.1, we have by Young inequality and Lipschitz condition on f

$$\begin{aligned} \mathbb{E}\big|\tilde{Y}_{t_i}^\pi - \tilde{\mathcal{V}}_{t_i}\big|^2 \le\ & (1 + \gamma\Delta t_i)\mathbb{E}\Big|\mathbb{E}_i\big[Y_{t_{i+1}}^\pi - \widehat{\mathcal{U}}_{i+1}(X_{t_{i+1}})\big]\Big|^2 \\ & + 2\frac{[f]_L^2}{\gamma}\big(1 + \gamma\Delta t_i\big)\Big\{\Delta t_i \mathbb{E}\big|\tilde{Y}_{t_i}^\pi - \tilde{\mathcal{V}}_{t_i}\big|^2 + \Delta t_i \mathbb{E}\big|Z_{t_i}^\pi - \overline{\tilde{Z}_{t_i}}\big|^2\Big\}. \end{aligned} \tag{4.41}$$

From (4.37), (4.39), Cauchy-Schwarz inequality, and law of iterated conditional expectations, we have similarly as in Step 1 in the proof of Theorem 4.1:

$$\Delta t_i \mathbb{E}\big|Z^\pi_{t_i} - \overline{\tilde{Z}_{t_i}}\big|^2 \le\ 2d\Big(\mathbb{E}\big|Y^\pi_{t_{i+1}} - \widehat{\mathcal{U}}_{i+1}(X_{t_{i+1}})\big|^2 - \mathbb{E}\Big|\mathbb{E}_i\big[Y^\pi_{t_{i+1}} - \widehat{\mathcal{U}}_{i+1}(X_{t_{i+1}})\big]\Big|^2\Big).$$

Then, by plugging into (4.41) and choosing $\gamma = 4d[f]_L^2$, we have for $|\pi|$ small enough:

$$\mathbb{E}\big|\tilde{Y}^\pi_{t_i} - \tilde{\mathcal{V}}_{t_i}\big|^2 \le\ (1 + C|\pi|)\mathbb{E}\big|Y^\pi_{t_{i+1}} - \widehat{\mathcal{U}}_{i+1}(X_{t_{i+1}})\big|^2.$$

Next, by using Young inequality as in Step 2 in the proof of Theorem 4.1, we obtain for all $\theta = (\xi, \zeta)$:

$$\mathbb{E}\big|\tilde{Y}^\pi_{t_i} - \mathcal{U}_i(X_{t_i};\xi)\big|^2 \le\ (1 + C|\pi|)\mathbb{E}\big|Y^\pi_{t_{i+1}} - \widehat{\mathcal{U}}_{i+1}(X_{t_{i+1}})\big|^2 + CN\mathbb{E}\big|\tilde{\mathcal{V}}_{t_i} - \mathcal{U}_i(X_{t_i};\xi)\big|^2. \quad (4.42)$$

On the other hand, by the martingale representation theorem, there exists an $\mathbb{R}^d$-valued square integrable process $(\tilde{Z}_t)_t$ such that

$$\widehat{\mathcal{U}}_{i+1}(X_{t_{i+1}}) =\ \tilde{\mathcal{V}}_{t_i} - f(t_i, X_{t_i}, \tilde{\mathcal{V}}_{t_i}, \overline{\tilde{Z}_{t_i}})\Delta t_i + \int_{t_i}^{t_{i+1}} \tilde{Z}_s^\intercal\, \mathrm{d}W_s,$$

and the expected squared loss function of the RDBDP scheme can be written as

$$\hat{L}_i(\theta) =\ \tilde{L}_i(\theta) + \mathbb{E}\Big[\int_{t_i}^{t_{i+1}} \big|\tilde{Z}_t - \overline{\tilde{Z}_{t_i}}\big|^2\, \mathrm{d}t\Big],$$

where we notice by Itô isometry that $\overline{\tilde{Z}_{t_i}} = \frac{1}{\Delta t_i}\mathbb{E}_i\Big[\int_{t_i}^{t_{i+1}} \tilde{Z}_t dt\Big]$, and

$$\begin{aligned}\tilde{L}_i(\theta) :=\ & \mathbb{E}\Big|\tilde{\mathcal{V}}_{t_i} - \mathcal{U}_i(X_{t_i};\xi) + \big(f(t_i, X_{t_i}, \mathcal{U}_i(X_{t_i};\xi), \mathcal{Z}_i(X_{t_i};\eta)) - f(t_i, X_{t_i}, \tilde{\mathcal{V}}_{t_i}, \overline{\tilde{Z}_{t_i}})\big)\Delta t_i\Big|^2 \\ &+\ \Delta t_i \mathbb{E}\big|\overline{\tilde{Z}_{t_i}} - \mathcal{Z}_i(X_{t_i};\eta)\big|^2.\end{aligned}$$

By the same arguments as in Step 3 in the proof of Theorem 4.1, using Lipschitz condition on f and Young inequality, we show that for all $\theta = (\xi, \eta)$

$$\begin{aligned}&(1 - C\Delta t_i)\mathbb{E}\big|\tilde{\mathcal{V}}_{t_i} - \mathcal{U}_i(X_{t_i};\xi)\big|^2 + \frac{\Delta t_i}{2}\mathbb{E}\big|\overline{\tilde{Z}_{t_i}} - \mathcal{Z}_i(X_{t_i};\eta)\big|^2 \\ &\qquad \le \tilde{L}_i(\theta)\ \le\ (1 + C\Delta t_i)\mathbb{E}\big|\tilde{\mathcal{V}}_{t_i} - \mathcal{U}_i(X_{t_i};\xi)\big|^2 + C\Delta t_i \mathbb{E}\big|\overline{\tilde{Z}_{t_i}} - \mathcal{Z}_i(X_{t_i};\eta)\big|^2.\end{aligned}$$

By taking $\theta_i^* = (\xi_i^*, \eta_i^*) \in \arg\min_\theta \hat{L}_i(\theta) = \arg\min_\theta \tilde{L}_i(\theta)$, it follows that for $|\pi|$ small enough

$$\mathbb{E}\big|\tilde{\mathcal{V}}_{t_i} - \mathcal{U}_i(X_{t_i};\xi_i^*)\big|^2 + \Delta t_i \mathbb{E}\big|\overline{\tilde{Z}_{t_i}} - \mathcal{Z}_i(X_{t_i};\eta_i^*)\big|^2 \le\ C\varepsilon_i^{\mathcal{N},\tilde{v}} + C\Delta t_i \varepsilon_i^{\mathcal{N},\tilde{z}}.$$

By plugging into (4.42), recalling that $\widehat{\mathcal{U}}_i(X_{t_i}) = \max[\mathcal{U}_i(X_{t_i};\xi_i^*); g(X_{t_i})]$, $Y^\pi_{t_i} = \max[\tilde{Y}^\pi_{t_i}; g(X_{t_i})]$, and since $|\max(a,c) - \max(b,c)| \le |a-b|$, we obtain

$$\mathbb{E}\big|Y^\pi_{t_i} - \widehat{\mathcal{U}}_i(X_{t_i})\big|^2 \le\ (1 + C|\pi|)\mathbb{E}\big|Y^\pi_{t_{i+1}} - \widehat{\mathcal{U}}_{i+1}(X_{t_{i+1}})\big|^2 + CN\big(\varepsilon_i^{\mathcal{N},\tilde{v}} + \Delta t_i \varepsilon_i^{\mathcal{N},\tilde{z}}\big),$$

and then by induction

$$\max_{i=0,\dots,N-1} \mathbb{E}\big|Y^\pi_{t_i} - \widehat{\mathcal{U}}_i(X_{t_i})\big|^2 \le\ C\sum_{i=0}^{N-1}\big(N\varepsilon_i^{\mathcal{N},\tilde{v}} + \varepsilon_i^{\mathcal{N},\tilde{z}}\big).$$

Combining with (4.40), this proves the error estimate (4.38) for the Y-component. The error estimate (4.38) for the Z-component is proved along the same arguments as in Step 5 in the proof of Theorem 4.1, and is omitted here. □

5 Numerical results

In the first two subsections, we compare our schemes DBDP1 (3.6), DBDP2 (3.7) and the scheme proposed by [HJE18] on some examples of PDEs and BSDEs.

We first test our algorithms on some PDEs with bounded solutions and quite a simple structure (see section 5.1), and then try to solve some PDEs with unbounded solutions and more complex structures (see section 5.2). Our goal is to emphasize that solutions with simple structure easily represented by a neural network can be evaluated by our method even in very high-dimension, whereas the solution with complex structure can only be evaluated in moderate dimension.

Finally, we apply the scheme described in section 3.3 to an American option problem and show its accuracy in high dimension (see section 5.3).

If not specified, we use in the sequel a fully connected feedforward network with two hidden layers, and $d+10$ neurons on each hidden layer, to implement our schemes (3.6) and (3.7). We choose tanh as activation function for the hidden layers in order to avoid some explosion while calculating the numerical gradient Z in scheme (3.7) and choose identity function as activation function for the output layer. We renormalize the data before entering the network. We use Adam Optimizer, implemented in TensorFlow and mini-batch with 1000 trajectories for the stochastic gradient descent.

5.1 PDEs with bounded solution and simple structure

We begin with a simple example in dimension one. It is not hard to find test cases where the scheme proposed in [HJE18] fails even in dimension one. In fact the latter scheme works well for small maturities and with a starting point close to the solution.

It is always interesting to start by testing schemes in dimension one as one can easily compare graphically the numerical results to the theoretical solution. Then we take some examples in higher dimensions and show that our method seems to work well when the dimension increases higher.

5.1.1 An example in 1D

We take the following parameters for the BSDE problem defined by (1.2) and (3.1):

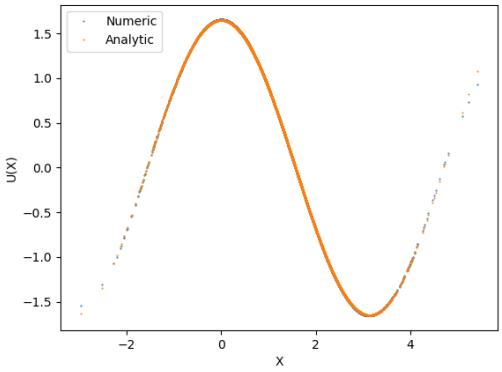
$$\sigma = 1,\ \mu = 0.2,\ T = 2,\ d = 1, \tag{5.1}$$

$$\begin{aligned} f(t,x,y,z) &= (\cos(x)(e^{\frac{T-t}{2}} + \tfrac{\sigma^2}{2}) + \mu\sin(x))e^{\frac{T-t}{2}} - \tfrac{1}{2}\left(\sin(x)\cos(x)e^{T-t}\right)^2 + \tfrac{1}{2}(yz)^2 \\ g(x) &= \cos(x). \end{aligned}$$

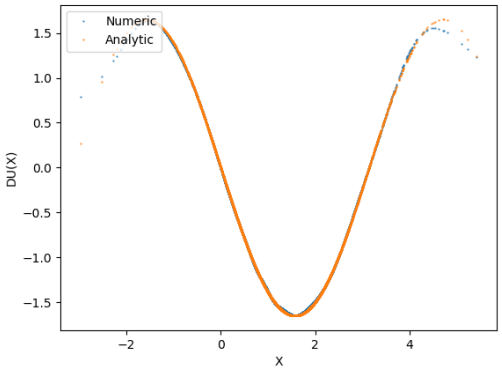
for which, the explicit analytic solution is equal to $u(t,x) = e^{\frac{T-t}{2}}\cos(x)$.

We want to estimate the solution u and its gradient $D_x u$ from our schemes. This example is interesting, because with $T = 1$, the method proposed in [HJE18], initializing $u(0,.)$ as the solution of the associated linear problem associated ($f = 0$) and randomly initializing $D_x u(0,.)$ works very well. However, for $T = 2$, the method in [HJE18] always fails on our test whatever the choice of the initialization: the algorithm is either trapped in a local minimum when the initial learning rate associated to the gradient method is too small or explodes when the learning rate is taken higher. This numerical failure is not dependent on the considered network: using some LSTM networks as in [CWNMW19] gives the same result.

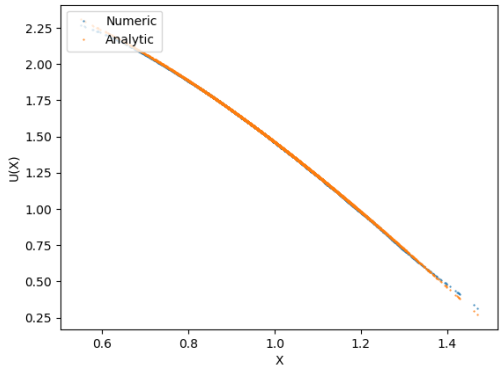
Because of the high non-linearity, we discretize the BSDE using $N = 240$ time steps, and implemented hidden layers with $d + 10 = 11$ neurons. Figure 1 (resp. Figure 2) depicts the estimated functions $u(t, .)$ and $D_x u(t, .)$ estimated from DBDP1 (resp. DBDP2) scheme.



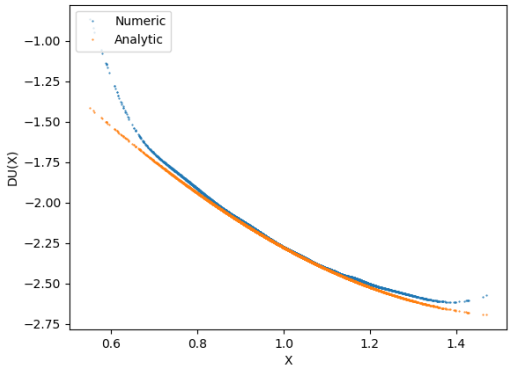
$u(t, .)$ and its estimate at time $t = 1$.



Z and its estimate at time $t = 1$.



$u(t, .)$ and its estimate at time $t = 0.0091$.

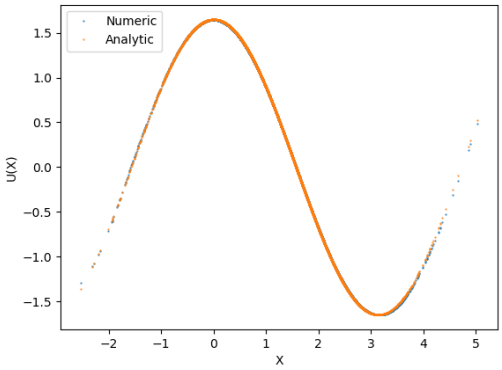


Z and its estimate at time $t = 0.0091$.

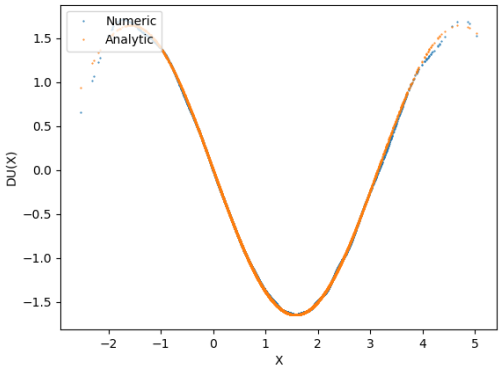
Figure 1: Estimates of u and Z using DBDP1. We took the parameters defined in (5.1) and set $x_0 = 1$.

	Averaged value	Standard deviation
DBDP1	1.46332	0.01434
DBDP2	1.4387982	0.01354

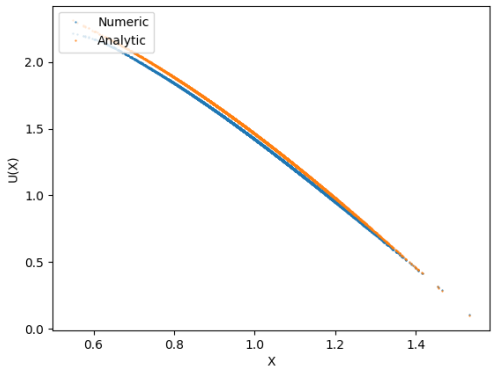
Table 1: Estimate of $u(0, x_0)$ where $d = 1$ and $x_0 = 1$. Average and standard deviation observed over 10 independent runs are reported. The theoretical solution is 1.4686938.



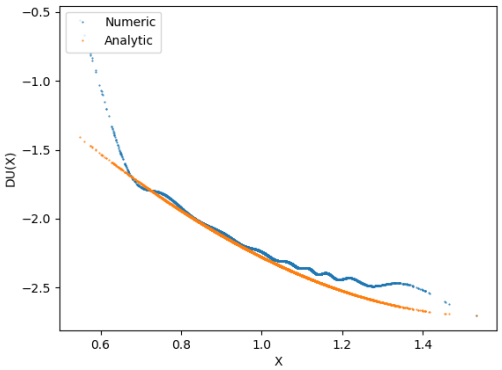
$u(t,.)$ and its estimate at time $t = 1$.



Z and its estimate at time $t = 1$.



$u(t,.)$ and its estimate at time $t = 0.0091$.



Z and its estimate at time $t = 0.0091$.

Figure 2: Estimates of u and Z using DBDP2. We took the parameters defined in (5.1) and set $x_0 = 1$.

5.1.2 Increasing the dimension

We extend the example from the previous section to the following d-dimensional problem:

$$d \geq 1, \quad \sigma = \frac{1}{\sqrt{d}} \mathbf{I}_d, \quad \mu = \frac{0.2}{d} \mathbb{1}_d, \quad T = 1,$$

$$\begin{aligned} f(t,x,y,z) &= (\cos(\bar{x})(e^{\frac{T-t}{2}} + \tfrac{1}{2}) + 0.2\sin(\bar{x}))e^{\frac{T-t}{2}} - \tfrac{1}{2}\left(\sin(\bar{x})\cos(\bar{x})e^{T-t}\right)^2 + \tfrac{1}{2d}(u(\mathbb{1}_d.z))^2, \\ g(x) &= \cos(\bar{x}), \end{aligned}$$

with $\bar{x} = \sum_{i=1}^{d} x_i$.

We take $N = 120$ in the Euler scheme, and $d+10$ neurons for each hidden layer. We take 1000 trajectories in mini batch, use data renormalization, and check the loss convergence every 50 iterations. For this small maturity, the scheme [HJE18] generally converges, and we give the results obtained with the same network and initializing the scheme with the linear solution of the problem. Results in dimension 5 to 50 are given in Tables 2, 3, 4 and 5. Both schemes (3.6) and (3.7) work well with results very close to the solution and close to the results calculated by the scheme [HJE18]. As the dimension increases, scheme (3.6) seems to be the most accurate.

Remark 5.1. In dimension 50, the initial learning rate in scheme [HJE18] is taken small in order to avoid a divergence of the method. In fact, running the test 3 times (with 10 runs each time), we observed convergence of the algorithm two times, and in the last test: one of the ten run exploded, and another one clearly converged to a wrong solution. □

	Averaged value	Standard deviation
DBDP1	0.4637038	0.004253
DBDP2	0.46335	0.00137
Scheme [HJE18]	0.46562	0.0035

Table 2: Estimate of $u(0, x_0)$ where $d = 5$ and $x_0 = \mathbb{1}_5$. Average and standard deviation observed over 10 independent runs are reported. The theoretical solution is 0.46768.

	Averaged value	Standard deviation
DBDP1	- 1.3895	0.00148
DBDP2	-1.3913	0.000583
Scheme [HJE18]	-1.3880	0.00155

Table 3: Estimate of $u(0, x_0)$ where $d = 10$ and $x_0 = \mathbb{1}_{10}$. Average and standard deviation observed over 10 independent runs are reported. The theoretical solution is -1.383395.

	Averaged value	Standard deviation
DBDP1	0.6760	0.00274
DBDP2	0.67102	0.00559
Scheme [HJE18]	0.68686	0.002402

Table 4: Estimate of $u(0, x_0)$ where $d = 20$ and $x_0 = \mathbb{1}_{20}$. Average and standard deviation observed over 10 independent runs are reported. The theoretical solution is 0.6728135.

	Averaged value	Standard deviation
DBDP1	1.5903	0.006276
DBDP2	1.58762	0.00679
Scheme [HJE18]	1.583023	0.0361

Table 5: Estimate of $u(0, x_0)$ where $d = 50$ and $x_0 = \mathbb{1}_{50}$. Average and standard deviation observed over 10 independent runs are reported. The theoretical solution is 1.5909.

5.2 PDEs with unbounded solution and more complex structure

In this section with take the following parameters

$$\begin{aligned} \sigma = & \frac{1}{\sqrt{d}}\mathbf{I}_d, \quad \mu = 0, \quad T = 1, \\ f(x, y, z) = & \ k(x) + \frac{1}{2\sqrt{d}} y(\mathbb{1}_d.z) + \frac{y^2}{2} \end{aligned} \tag{5.2}$$

where the function k is chosen such that the solution to the PDE is equal to

$$u(t, x) = \frac{T - t}{d} \sum_{i=1}^{d} (\sin(x_i) 1_{x_i < 0} + x_i 1_{x_1 \geq 0}) + \cos\left(\sum_{i=1}^{d} i x_i \right).$$

Notice that the structure of the solution is more complex than in the first example. We aim at evaluating the solution at $x = 0.5\mathbb{1}_d$. We take 120 time steps for the Euler time discretization and $d + 10$ neurons in each hidden layers. As shown in Figures 3 and 4 as well as in Table 6, the three schemes provide accurate and stable results in dimension $d = 1$.

	Averaged value	Standard deviation
DBDP1	1.3720	0.00301
DBDP2	1.37357	0.0022
Scheme [HJE18]	1.37238	0.00045

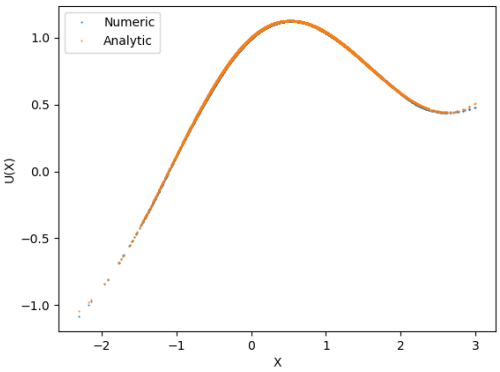
Table 6: Estimate of $u(0, x_0)$, where $d = 1$ and $x_0 = 0.5$. Average and standard deviation observed over 10 independent runs are reported. The theoretical solution is 1.37758.

In dimension 2, the three schemes provide very accurate and stable results, as shown in Figures 5 and 6, as well as in Table 7.

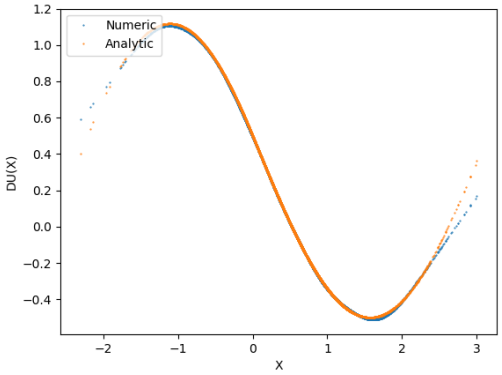
	Averaged value	Standard deviation
DBDP1	0.5715359	0.0038
DBDP2	0.5707974	0.00235
Scheme [HJE18]	0.57145	0.0006

Table 7: Estimate of $u(0, x_0)$, where $d = 2$ and $x_0 = 0.5\mathbb{1}_2$. Average and standard deviation observed over 10 independent runs are reported. The theoretical solution is 0.570737.

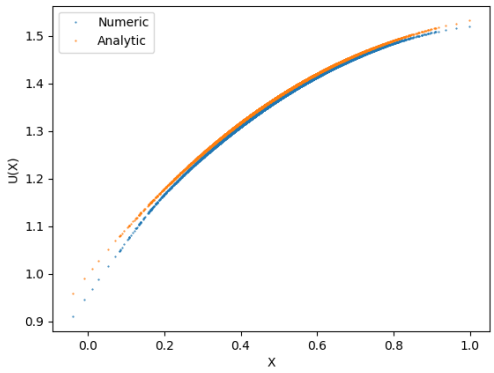
Above dimension 3, the scheme [HJE18] always explodes no matter the chosen initial learning rate and the activation function for the hidden layers (among the tanh, ELU, ReLu and sigmoid ones). Besides, taking 3 or 4 hidden layers does not improve the results.



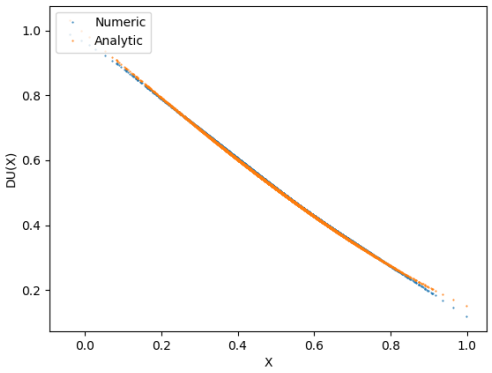
$u(t,.)$ and its estimate at time $t = 0.5$.



Z and its estimate at time $t = 0.5$



$u(t,.)$ and its estimate at time $t = 0.0085$.



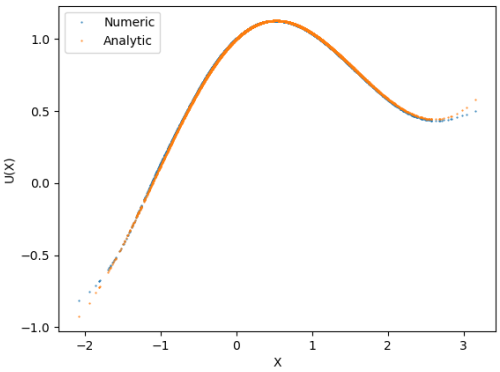
Z and its estimate at time $t = 0.0085$.

Figure 3: Estimates of u and Z using DBDP1. We took the parameters defined in (5.2), with $d = 1$, and set $x_0 = 0,5$.

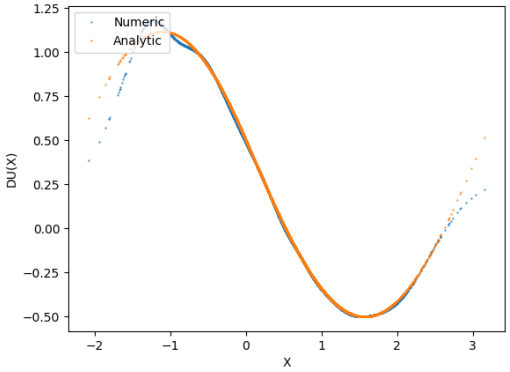
We reported the results obtained in dimension $d = 5$ and 8 in Table 8 and 9. Scheme (3.6) seems to work better than scheme (3.7) as the dimension increases. Note that the standard deviation increases with the dimension of the problem.

	Averaged value	Standard deviation
DBDP1	0.8666	0.013
DBDP2	0.83646	0.00453
Scheme [HJE18]	NC	NC

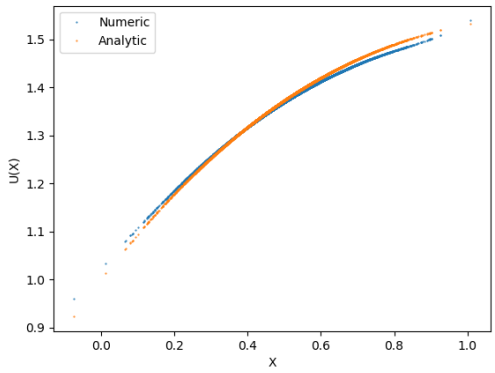
Table 8: Estimate of $u(0, x_0)$, where $d = 5$ and $x_0 = 0.5\mathbb{1}_5$. Average and standard deviation observed over 10 independent runs are reported. The theoretical solution is 0.87715.



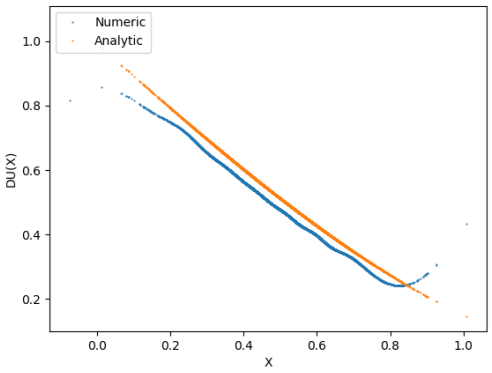
$u(t,.)$ and its estimate at time $t = 0.5$.



Z and its estimate at time $t = 0.5$



$u(t,.)$ and its estimate at time $t = 0.0085$.



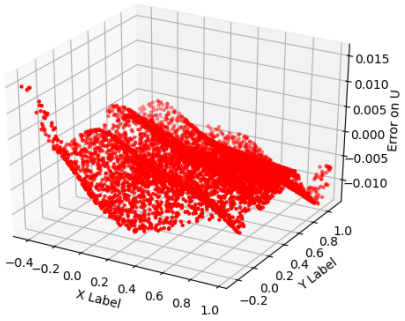
Z and its estimate at time $t = 0.0085$.

Figure 4: Estimates of u and Z using DBDP2. We took the parameters defined in (5.2), with $d = 1$, and set $x_0 = 0.5$.

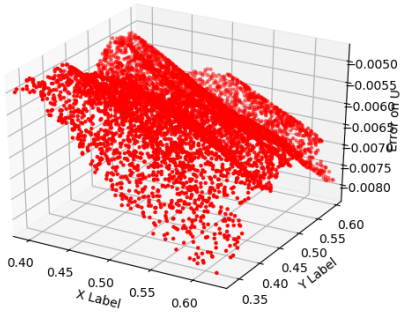
	Averaged value	Standard deviation
DBDP1	1.169441	0.02537
DBDP2	1.0758344	0.00780
Scheme [HJE18]	NC	NC

Table 9: Estimate of $u(0, x_0)$, where $d = 8$ and $x_0 = 0.5\mathbb{1}_8$. Average and standard deviation observed over 10 independent runs are reported. The theoretical solution is 1.1603167.

When $d \geq 10$, schemes (3.6) and (3.7) both fail at providing correct estimates of the solution, as shown in Table 10. Increasing the number of layers or neurons does not improve the result.

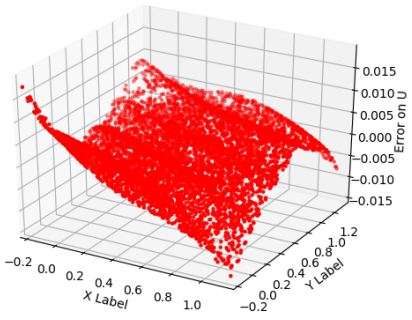


Error on solution at date $t = 0.5$.

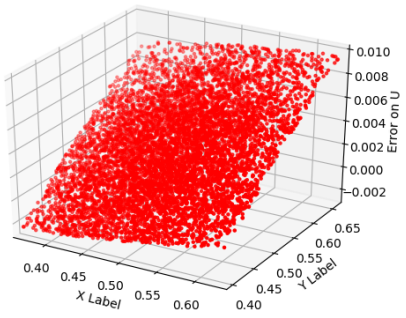


Error on solution at date $t = 0.0085$.

Figure 5: Algebric error of the estimate of u using DBDP1. We took the parameters in (5.2) and set $d = 2$ and $x_0 = 0.5\mathbb{1}_d$.



Error on solution at date $t = 0.5$.



Error on solution at date $t = 0.0085$.

Figure 6: Algebric error of the estimate of u using scheme (3.7). We took the parameters in (5.2) and set $d = 2$ and $x_0 = 0.5\mathbb{1}_d$.

	Averaged value	Standard deviation
DBDP1	-0.3105	0.02296
DBDP2	-0.3961	0.0139
Scheme [HJE18]	NC	NC

Table 10: Estimate of $u(0, x_0)$, where $d = 10$ and $x_0 = 0.5\mathbb{1}_{10}$. Average and standard deviation observed over 10 independent runs are reported. The theoretical solution is -0.2148861.

5.3 Application to American options

Consider the stock price $X_t = (X_t^1, \dots, X_t^d)$ of d assets with the following dynamics under the risk neutral probability measure:

$$dX_t^i = rX_t^i dt + \sigma_i X_t^i dW_t^i,$$

where $W_\cdot = (W_\cdot^1, \dots, W_\cdot^d)$ is a d-dimensional Brownian Motion, $\sigma = (\sigma_1, \dots, \sigma_d) \in \mathbb{R}^d$, and r is the risk-free rate.

The value at time t of an American option with payoff g and maturity T is given by:

$$u(t,x) = \sup_{\tau \in \mathcal{T}_{t,T}} \mathbb{E}[e^{-r\tau} g(X_\tau)],$$

where $\mathcal{T}_{t,T}$ is the set of stopping time with values in $[t,T]$, and is solution of the variational inequality

$$\left\{ \begin{array}{rcll} \min\big[-\partial_t u - \hat{\mathcal{L}} u, u - g\big] & = & 0, & \text{on } [0,T) \times (0,\infty)^d \\ u(T,.) & = & g, & \text{on } (0,\infty)^d, \end{array} \right.$$

with

$$\hat{\mathcal{L}} u(t,x) = \frac{1}{2} \sum_{i=1}^d \sigma_i^2 x_i^2 D_{x_i}^2 u(t,x) + r \sum_{i=1}^d x_i D_{x_i} u(t,x) - ru(t,x),$$

as proved e.g. in [JLL90].

Let us define the change of function v by: $u(t,x) = e^{rt} v(t, \log(x))$, (where log is applied component-wise), which is solution of the following variational inequality

$$\left\{ \begin{array}{rcll} \min\left(-\partial_t v - \mathcal{L} v, v - \hat{g}\right) & = & 0, & \text{on } [0,T) \times \mathbb{R}^d \\ v(T,.) & = & \hat{g}, & \text{on } \mathbb{R}^d, \end{array} \right. \tag{5.3}$$

where

$$\hat{g}(t,x) = e^{-rt} g(e^x),$$

$$\mathcal{L} v = \frac{1}{2} \sum_{i=1}^d \sigma_i^2 D_{x_i}^2 v_{ii} + \sum_{i=1}^d (r - \frac{1}{2}\sigma_i^2) D_{x_i} v_i.$$

In this section, we test the scheme described in section 3.3 on (5.3) in the special case of a geometrical put with strike $K = 1$, $T = 1$, $r = 0.05$, $X_0^i = 1$, $\sigma_i = 0.2$ for $i = 1$ to d, and payoff $(K - \prod_{i=1}^d X_t^i)_+$, as considered previously in [BW12]. In dimension d, the case boils down to the resolution of an American option in dimension $d = 1$: indeed, the option payoff involving only the product of the asset values, it can be written as the payoff of a single asset with a trend equal to dr and a volatility $\sigma_1 \sqrt{d}$, so that it can be very accurately estimated e.g. with a tree-based method. Results given in Table 11 show that scheme (3.10) is very accurate for the pricing of American options.

Dimension	nb step	value	std	reference
1	10	0.06047	0.00023	0.060903
1	20	0.060789	0.00021	0.060903
1	40	0.061122	0.00015	0.060903
1	80	0.0613818	0.00019	0.060903
5	10	0.10537	0.00014	0.10738
5	20	0.10657	0.00011	0.10738
5	40	0.10725	0.00012	0.10738
5	80	0.107650	0.00016	0.10738
10	10	0.12637	0.00014	0.12996
10	20	0.128292	0.00011	0.12996
10	40	0.12937	0.00014	0.12996
10	80	0.129923	0.00016	0.12996
20	10	0.1443	0.00014	0.1510
20	20	0.147781	0.00012	0.1510
20	40	0.149560	0.00012	0.1510
20	80	0.15050	0.00010	0.1510
40	10	0.15512	0.00018	0.1680
40	20	0.16167	0.00015	0.1680
40	40	0.16487	0.00011	0.1680
40	80	0.16665	0.00013	0.1680
40	160	0.16758	0.00016	0.1680

Table 11: Estimates of the American option using RDBDP. Average and standard deviation over 40 independent runs for different numbers of time steps are reported.

References

[Avi09] R. Avikainen. "On irregular functionals of SDEs and the Euler scheme". In: *Finance and Stochastics* 13 (2009), pp. 381–401.

[BC08] B. Bouchard and J.F. Chassagneux. "Discrete-time approximation for continuously and discretely reflected BSDEs". In: *Stochastic Processes and their Applications* 118 (2008), pp. 2269–2293.

[BCJ19] S. Becker, P. Cheredito, and A. Jentzen. "Deep optimal stopping". In: *Journal of Machine learning research* 20 (2019), pp. 1–25.

[BEJ19] C. Beck, W. E, and A. Jentzen. "Machine learning approximation algorithms for high-dimensional fully nonlinear partial differential equations and second-order backward stochastic differential equations". In: *Journal of Nonlinear Science* 29 (2019), pp. 1563–1619.

[BP03] V. Bally and G. Pagès. "Error analysis of the quantization algorithm for obstacle problems". In: *Stochastic Processes and their Applications* 106 (2003), pp. 1–40.

[BT04] B. Bouchard and N. Touzi. "Discrete-time approximation and Monte-Carlo simulation of backward stochastic differential equations". In: *Stochastic Processes and their applications* 111.2 (2004), pp. 175–206.

[BW12] B. Bouchard and X. Warin. "Monte-Carlo valuation of American options: facts and new algorithms to improve existing methods". In: *Numerical methods in finance*. Springer, 2012, pp. 215–255.

[CWNMW19] Q. Chan-Wai-Nam, J. Mikael, and X. Warin. "Machine Learning for semi linear PDEs". In: *Journal of Scientific Computing* 79.3 (2019), pp. 1667–1712.

[EHJ17] W. E, J. Han, and A. Jentzen. "Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations". In: *Communications in Mathematics and Statistics* 5.4 (2017), pp. 349–380.

[EK+97] N. El Karoui et al. "Reflected Solutions of Backward SDEs, and related obstacle problems for PDEs". In: *Annals of Probability* 25.2 (1997), pp. 702–737.

[GLW05] E. Gobet, J.P. Lemor, and X. Warin. "A regression-based Monte Carlo method to solve backward stochastic differential equations". In: *The Annals of Applied Probability* 15.3 (2005), pp. 2172–2202.

[HJE18] J. Han, A. Jentzen, and W. E. "Solving high-dimensional partial differential equations using deep learning". In: *Proceedings of the National Academy of Sciences* 115.34 (2018), pp. 8505–8510.

[HL+16] P. Henry-Labordere et al. "Branching diffusion representation of semilinear PDEs and Monte Carlo approximation". In: *Annales de l'Institut Henri Poincaré (B) Probabilités et Statistiques* (2016). to appear.

[HL18] J. Han and J. Long. "Convergence of the Deep BSDE Method for Coupled FBSDEs". arXiv:1811.01165. 2018.

[HSW89] K. Hornik, M. Stinchcombe, and H. White. "Multilayer feedforward networks are universal approximators". In: *Neural Networks* 2.5 (1989), pp. 359–366.

[HSW90] K. Hornik, M. Stinchcombe, and H. White. "Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks". In: *Neural Networks* 3(5) (1990), pp. 551–560.

[Hut+18] M. Hutzenthaler et al. "Overcoming the curse of dimensionality in the numerical approximation of semilinear parabolic partial differential equations". In: *arXiv preprint arXiv:1807.01212* (2018).

[Hut+19] M. Hutzenthaler et al. "A proof that rectified deep neural networks overcome the curse of dimensionality in the numerical approximation of semilinear heat equations". ArXiv:1901.10854. 2019.

[JLL90] P. Jaillet, D. Lamberton, and B. Lapeyre. "Variational Inequalities and the Pricing of American Options". In: *Acta Applicandae Mathematicae* 21(3) (1990), pp. 263–289.

[LGW06] J.P. Lemor, E. Gobet, and X. Warin. "Rate of convergence of an empirical regression method for solving generalized backward stochastic differential equations". In: *Bernoulli* 12.5 (2006), pp. 889–916.

[PP90] E. Pardoux and S. Peng. "Adapted solution of a backward stochastic differential equation". In: *Systems & Control Letters* 14.1 (1990), pp. 55–61.

[SS18] J. Sirignano and K. Spiliopoulos. "DGM: A deep learning algorithm for solving partial differential equations". In: *Journal of Computational Physics* 375 (2018), pp. 1339–1364.

[War18a] X. Warin. "Monte Carlo for high-dimensional degenerated Semi Linear and Full Non Linear PDEs". In: *arXiv preprint arXiv:1805.05078* (2018).

[War18b] X. Warin. "Nesting Monte Carlo for high-dimensional non-linear PDEs". In: *Monte Carlo Methods and Applications* 24.4 (2018), pp. 225–247.

[Zha04] J. Zhang. "A numerical scheme for BSDE's". In: *The Annals of Applied Probability* 14.1 (2004), pp. 459–488.

[E+18] W. E et al. "On multilevel Picard numerical approximations for high-dimensional nonlinear parabolic partial differential equations and high-dimensional nonlinear backward stochastic differential equations". In: *to appear in Journal of Scientific Computing* (2018).