Pathwise XVAs: The Direct Scheme*

Lokman Abbas-Turki[1] Stéphane Crépey[1] Bouazza Saadeddine[1,2,3]
Wissal Sabbagh[4]

October 31, 2022

Abstract

Motivated by the equations of cross valuation adjustments (XVAs) in the realistic case where capital is deemed fungible as a source of funding for variation margin, we introduce a simulation scheme for a class of anticipated BSDEs, where the coefficient entails a conditional expected shortfall of the martingale part of the solution. The scheme is explicit in time and uses neural network least-squares and quantile regressions for the embedded conditional expectations and expected shortfall computations. An a posteriori Monte Carlo validation procedure allows assessing the regression error of the scheme at each time step. The superiority of this scheme with respect to Picard iterations is illustrated in a high-dimensional and hybrid market/default risks XVA use-case.

Keywords: anticipated BSDE, neural network regression and quantile regression, cross-valuation adjustments (XVA).

Mathematics Subject Classification: 62M45, 65C30, 91G20, 91G40, 91G60, 91G70.

1 Introduction

(Crépey, Sabbagh, and Song, 2020) establishes the well-posedness of an anticipated BSDE (ABSDE) in the line of (Peng and Yang, 2009), but for a coefficient entailing a conditional *expected shortfall* of a future increment of the *martingale part* of the solution. Such a coefficient occurs in the equations of cross valuation adjustments (XVAs),

[1] *Laboratoire de Probabilités, Statistique et Modélisation (LPSM), Sorbonne Université et Université Paris Cité, CNRS UMR 8001*

[2] *LaMME, Univ Evry, CNRS, Université Paris-Saclay, 91037, Evry, France.*

[3] *Quantitative Research GMD/GMT Crédit Agricole CIB, 92160 Montrouge.*

[4] *CREST-ENSAE.*

Corresponding author: `stephane.crepey@lpsm.paris`

Acknowledgement: This research has been conducted with the support of the Chair *Capital Markets Tomorrow : Modeling and Computational Issues* under the aegis of the Institut Europlace de Finance, a joint initiative of Laboratoire de Probabilités, Statistique et Modélisation (LPSM) / Université Paris Cité and Crédit Agricole CIB.

*GPU and python implementation available on https://github.com/BouazzaSE/NeuralXVA.

accounting for the possibility to use capital at risk as a source of funding for variation margin. In the present paper we address the numerical solution of such ABSDEs and their XVA application.

Numerical schemes for BSDEs include backward dynamic programming based on Euler (Bouchard and Touzi, 2004; Zhang, 2004) or higher order (Chassagneux and Crisan, 2014) schemes, combined with regression (Gobet, Lemor, and Warin, 2005; Huré, Pham, and Warin, 2020), Malliavin (Crisan, Manolarakis, and Touzi, 2010) or cubature (Lyons and Victoir, 2004) methods to estimate the embedded conditional expectations. These admit extensions to jump-diffusions (Bouchard and Élie, 2008), reflected BSDEs (Chassagneux and Richou, 2019), forward-backward SDEs (Delarue and Menozzi, 2006), quadratic BSDEs (Chassagneux and Richou, 2016) or McKean-Vlasov BSDEs (Chassagneux, Crisan, and Delarue, 2019). Alternative machine learning schemes (E, Han, and Jentzen, 2017; Teng, 2022) open the door to the numerical solution of BSDEs in dimensions 100 to 1000 (instead of, say, 10 otherwise), however the mathematical convergence analysis of these schemes is still very incomplete. Numerical schemes for BSDEs also include Monte Carlo branching schemes (Henry-Labordere, Tan, and Touzi, 2017) or multilevel Picard schemes (Weinan, Hutzenthaler, Jentzen, and Kruse, 2019), but these only provide time 0 estimates (unless they are applied in a nested fashion at each outer node of a nested simulation), making them unsuitable for our path-wise XVA purposes in this work.

There is not much literature on the numerical treatment of ABSDEs. As in the present paper, but in a purely Brownian setup, Agarwal, Marco, Gobet, López-Salas, Noubiagain, and Zhou (2019) consider an ABSDE involving a conditional expected shortfall as anticipated term (by contrast with a conditional expectation in the previous ABSDE literature). Exploiting the short horizon of the anticipation in the equation (one week in their case versus one year in ours), they devise approximations by standard BSDEs, which allows them to avoid the difficulty posed by the regression of the anticipated terms[1]. The XVA ABSDEs received a first numerical treatment in Albanese, Caenazzo, and Crépey (2017) by nested Monte Carlo[2], using Picard iterations to decouple the solution from the embedded conditional risk measures and ignoring the conditionings in the latter[3] to avoid multiply nested Monte Carlo. The other natural approach to address such problems numerically is regression-based Monte Carlo, i.e. iterated regressions (or more general supervised learning algorithms) that are used for cutting the recursively nested levels of Monte Carlo to which a naive implementation of the equations conducts. A first take in this direction, still using Picard iterations for decoupling purposes, was implemented in Albanese, Crépey, Hoskinson, and Saadeddine (2021), leveraging on the elicitability of the embedded risk measures for learning not only the XVAs, but also these risk measures: conditional value-at-risk for dynamic initial margin calculations and conditional expected shortfall[4] for dynamic economic capital calculations. Proceeding

[1] cf. the beginning of Section 3.2 in Agarwal, Marco, Gobet, López-Salas, Noubiagain, and Zhou (2019).

[2] cf. also Abbas-Turki, Crépey, and Saadeddine (2022).

[3] i.e. computing unconditional risk measures instead of conditional ones.

[4] that is elicitable jointly with value-at-risk.

in this way allowed Albanese, Crépey, Hoskinson, and Saadeddine (2021) to learn the embedded conditional risk measures, instead of treating them numerically as constants in Albanese, Caenazzo, and Crépey (2017).

In the present paper we introduce an explicit time-discretization scheme which, in conjunction with a refined neural net regression approach for the embedded conditional expectations and risk measures, leads to a direct algorithm for computing the XVA metrics, without Picard iterations. A numerical benchmark of both schemes in a realistic XVA setup emphasizes the superiority of the explicit scheme with respect to the Picard one.

The paper is outlined as follows. Section 2 recasts the generic ABSDE of Crépey, Sabbagh, and Song (2020) in a Markovian setup amenable to numerical simulations. Section 3 introduces the related regression-based explicit and implicit/Picard simulation schemes. Section 4 provides a numerical benchmark of both schemes applied to XVA computations. Section 5 discusses the outputs of the paper in relation with the literature and introduces future research perspectives. The specification of the algorithms to the XVA equations is detailed in Section A.

1.1 Standing Notation

We denote by:

- $|\cdot|$, an Euclidean norm in the dimension of its arguments;
- $T \in (0,\infty)$, a constant time horizon;
- $(\Omega, \mathcal{A}, \mathfrak{F}, \mathbb{P})$, a filtered probability space, for a probability measure $\mathbb{P}$ on the measurable space $(\Omega, \mathcal{A})$ and a complete and right-continuous filtration $\mathfrak{F} = (\mathfrak{F}_t)_{0\le t\le T}$ of sub-σ fields of $\mathcal{A}$;
- $\mathbb{E}$, the $\mathbb{P}$ expectation, and $\mathbb{P}_t$, $\mathbb{E}_t$, and $\mathbb{ES}_t$, the $(\mathfrak{F}_t, \mathbb{P})$ conditional probability, expectation, and expected shortfall at some given quantile level $\alpha \in (\frac{1}{2}, 1)$.

By the latter we mean, for each $\mathfrak{F}_T$ measurable, $\mathbb{P}$ integrable, random variable ℓ,

$$\mathbb{ES}_t(\ell) = \mathbb{E}_t\big(\ell | \ell \ge q_t^\alpha(\ell)\big), \tag{1}$$

in which $q_t^\alpha(\ell)$ denotes the $(\mathfrak{F}_t, \mathbb{P})$ conditional left-quantile[5] of level α of ℓ. We recall that[6], for any $\mathfrak{F}_T$ measurable, $\mathbb{P}$ integrable random variables ℓ and ℓ',

$$|\mathbb{ES}_t(\ell) - \mathbb{ES}_t(\ell')| \;\le\; (1-\alpha)^{-1}\mathbb{E}_t[|\ell - \ell'|]. \tag{2}$$

2 Limiting Equations

In this section we specify the stochastic differential equations addressed from a numerical viewpoint in later sections.

[5] value-at-risk.

[6] additionally assuming ℓ and ℓ' atomless given $\mathfrak{F}_t$, without harm for the XVA applications targeted in this work; cf. e.g. Barrera, Crépey, Diallo, Fort, Gobet, and Stazhynski (2019, Lemma A.6, Eq. (A.16)).

2.1 Spaces and Martingale Representation

Given nonnegative integers d and q, we denote by W, an $(\mathfrak{F},\mathbb{P})$ standard d variate Brownian motion, and $\nu=(\nu^k)$, an integer valued random measure[7] on $\{0,1\}^q$, with $\mathbb{P}$ compensatrix

$$d\mu_t^k = d\nu_t^k - \gamma_t^k dt,\ k\in\{0,1\}^q,$$

for some nonnegative real valued predictable processes γ^k. Given any positive integer l, we introduce:

- $\mathcal{S}_2^l$, the space of $\mathbb{R}^l$ valued $\mathfrak{F}$ adapted càdlàg processes Y such that

$$\|Y\|^2_{\mathcal{S}_2^l} = \mathbb{E}\big[\sup_{0\le t\le T}|Y_t|^2\big] < +\infty;$$

- $\mathcal{H}_2^l$, the space of $\mathbb{R}^{l\otimes d}$ valued $\mathfrak{F}$ progressive processes Z such that

$$\|Z\|^2_{\mathcal{H}_2^l} = \mathbb{E}\big[\int_0^T |Z_t|^2 dt\big] < +\infty;$$

- $\widetilde{\mathcal{H}}_2^l$, the space of $\mathbb{R}^{l\otimes 2^q}$ valued $\mathfrak{F}$ predictable processes U such that

$$\|U\|^2_{\widetilde{\mathcal{H}}_2^l} = \mathbb{E}\Big[\int_0^T |U|_t^2 dt\Big] < +\infty \text{ , where } |U|_t^2 = \sum_k \gamma_t^k |U_t^k|^2.$$

We use $\int_0^t U_s d\mu_s$ as shorthand for $\sum_{k\in\{0,1\}^q}\int_0^t U_s^k d\mu_s^k$. In the case where $l=1$ we drop the index l, e.g. we write $\mathcal{S}_2$ instead of $\mathcal{S}_2^1$.

Assumption 2.1 *Every $(\mathfrak{F},\mathbb{P})$ martingale in $\mathcal{S}_2$ starting from 0 has a representation of the form*

$$\int_0^{\cdot} Z_t dW_t + \int_0^{\cdot} U_t d\mu_t, \tag{3}$$

for some $Z\in\mathcal{H}_2$ and $U\in\widetilde{\mathcal{H}}_2$.

2.2 The Markovian Anticipated BSDE

Given a positive integer p, let X in $\mathcal{S}_2^p$ satisfy

$$dX_t = b(t,X_t)dt + \sigma(t,X_t)dW_t, \tag{4}$$

for coefficients $b(t,x)$ and $\sigma(t,x)$ Lipschitz in x uniformly in $t\in[0,T]$ and with linear growth in x. Hence the SDE (4) is classically well-posed in S_2^p, for any constant initial

[7] see Jacod (1979).

condition $x \in \mathbb{R}^p$. We write $\mathcal{X} = (X, J)$, where a $\{0,1\}^q$ valued "Markov chain like" model component J[8] satisfies

$$dJ_t = \sum_{k \in \{0,1\}^q} (k - J_{t-}) d\nu_t^k \tag{5}$$

and the compensator $\gamma_t^k dt$ of each $d\nu_t^k$ satisfies $\gamma_t^k = \mathbb{1}_{\{J_{t-} \neq k\}} \gamma_k(t, X_t)$, for some continuous functions $\gamma_k(t,x)$ of (t,x). Hence ν_t^k counts the number of transitions of J to the state k on $(0,t]$.

We write $f(t, \mathcal{X}_t, \cdots)$ as shorthand for $f_{J_t}(t, X_t, \cdots)$, for any function $f = f_k(t, x, \cdots)$. Given a positive integer l, let $\phi = \phi_k(x)$ define for each k an $\mathbb{R}^l$ valued continuous function on $\mathbb{R}^p$, $f = f_k(t, x, y, \varrho)$ define for each k an $\mathbb{R}^l$ valued continuous function on $[0,T] \times \mathbb{R}^p \times \mathbb{R}^l \times \mathbb{R}$, and $M \mapsto \mathbb{ES}_\cdot(\Phi_\cdot(M))$ define a map from $\mathcal{S}_2^l$ into the space of $\mathfrak{F}$ predictable[9] processes, where[10]

$$\Phi_{\bar{t}}(M) := \Phi\big(t; \mathcal{X}_{[t,\bar{t}]}, M_{[t,\bar{t}]} - M_t\big), \tag{6}$$

for some deterministic maps $t \mapsto \bar{t} \in [t,T]$[11] and Φ of time t and càdlàg paths $\mathbf{x}$ and $\mathbf{m}$ on $[t,\bar{t}]$ such that $\mathbf{m}_t = 0$. Assuming $\Phi_{\bar{t}}(M)$ integrable for each time t and $M \in \mathcal{S}_2^l$, we consider the following backward stochastic differential equation (BSDE) for Y in $\mathcal{S}_2^l$:

$$Y_t = \mathbb{E}_t\Big[\phi(\mathcal{X}_T) + \int_t^T f\big(s, \mathcal{X}_s, Y_s, \mathbb{ES}_s(\Phi_{\bar{s}}(M))\big) ds\Big],\ t \leq T, \tag{7}$$

where M, also required to belong to $\mathcal{S}_2^l$, is the canonical Doob-Meyer martingale component of the special semimartingale Y. The BSDE (7) is anticipated in the sense of Peng and Yang (2009) that its coefficient at time t depends on the future (already known at t) of the solution. However, as detailed in Section 1, the nature of this dependence is different from the one in Peng and Yang (2009). More precisely, the present paper provides the following Markovian specification of the semimartingale setup of Crépey, Sabbagh, and Song (2020).

Assumption 2.2 **(i)** The function $f = f_k(t, x, y, \varrho)$ is Λ_f Lipschitz in (y, ϱ) ;

(ii) The processes $\mathbb{ES}_\cdot|(\Phi_\cdot(0)|)$ and $|f(\cdot, \mathcal{X}_\cdot, 0, 0)|$ are in $\mathcal{H}_2$;

(iii) Φ is Lipschitz with respect to its last argument in the sense that for every $t \in [0,T]$,

$$|\Phi(t; \mathbf{x}, \mathbf{m}) - \Phi(t; \mathbf{x}, \mathbf{m}')| \leq \Lambda_\Phi |\mathbf{m}_{\bar{t}} - \mathbf{m}'_{\bar{t}}| \tag{8}$$

holds for all càdlàg paths $\mathbf{x}, \mathbf{m}, \mathbf{m}'$ on $[t,\bar{t}]$ such that $\mathbf{m}_t = \mathbf{m}'_t = 0$.

[8] but with transition probabilities modulated by X.

[9] assuming the raw process $\Phi_{\bar{t}}(M)$ càdlàg in t, see Crépey, Sabbagh, and Song (2020, Lemma 2.1).

[10] see e.g. the last line in (29).

[11] e.g. $\bar{t} = (t+1) \wedge T$ in our XVA use case of Section 4.

Remark 2.1 *Assumption 2.2(iii) strongly points out to the case where $\Phi_{\bar{t}}(M)$ only depends on M through $M_{\bar{t}} - M_t$, which indeed corresponds to our XVA use case later below. However, the algorithms of Section 3 are not restricted to this case and Assumption 2.2(iii) only yields a sufficient condition for the conclusion of Theorem 2.1 below to hold.*

Lemma 2.1 *There exists a positive constant Λ_ρ such that*

$$|\mathbb{ES}_t(\Phi_{\bar{t}}(M)) - \mathbb{ES}_t(\Phi_{\bar{t}}(M'))|^2 \leq \Lambda_\rho^2 \mathbb{E}_t\Big[\int_t^{\bar{t}} \big(|Z_s - Z'_s|^2 + |U - U'|_s^2\big) ds\Big] \tag{9}$$

holds for any $M, M' \in \mathcal{S}_2^l$, where (Z, U) and (Z', U') in $\mathcal{H}_2^l \times \widetilde{\mathcal{H}}_2^l$ are the integrands in the martingale representations (3) *of $M - M_0$ and $M' - M'_0$.*

Proof. By (2), we have

$$\begin{aligned}
&(1-\alpha)^2 |\mathbb{ES}_t(\Phi_{\bar{t}}(M)) - \mathbb{ES}_t(\Phi_{\bar{t}}(M'))|^2 \leq \\
&\Big(\mathbb{E}_t\Big[\Phi\big(t; \mathcal{X}_{[t,\bar{t}]}, M_{[t,\bar{t}]} - M_t\big) - \Phi\big(t; \mathcal{X}_{[t,\bar{t}]}, M'_{[t,\bar{t}]} - M'_t\big)\Big]\Big)^2 \leq \\
&\mathbb{E}_t\Big[\Big(\Phi\big(t; \mathcal{X}_{[t,\bar{t}]}, M_{[t,\bar{t}]} - M_t\big) - \Phi\big(t; \mathcal{X}_{[t,\bar{t}]}, M'_{[t,\bar{t}]} - M'_t\big)\Big)^2\Big],
\end{aligned}$$

by the (conditional) Jensen inequality. Moreover, denoting $\delta M = M - M'$, the Lipschitz condition (8) yields

$$\Big(\Phi\big(t; \mathcal{X}_{[t,\bar{t}]}, M_{[t,\bar{t}]} - M_t\big) - \Phi\big(t; \mathcal{X}_{[t,\bar{t}]}, M'_{[t,\bar{t}]} - M'_t\big)\Big)^2 \leq \Lambda_\Phi^2 |\delta M_{\bar{t}} - \delta M_t|^2,$$

where, with $\delta Z = Z - Z'$ and $\delta U = U - U'$,

$$\delta M_{\bar{t}} - \delta M_t = \int_t^{\bar{t}} \delta Z_s dW_s + \int_t^{\bar{t}} \delta U_s d\mu_s,$$

hence

$$|\delta M_{\bar{t}} - \delta M_t|^2 = \sum_{\iota=1}^{l} \big(\int_t^{\bar{t}} \delta Z_s^\iota dW_s + \int_t^{\bar{t}} \delta U_s^\iota d\mu_s\big)^2.$$

Therefore

$$(1-\alpha)^2 |\mathbb{ES}_t(\Phi_{\bar{t}}(M)) - \mathbb{ES}_t(\Phi_{\bar{t}}(M'))|^2 \leq \Lambda_\Phi^2 \mathbb{E}_t \sum_{\iota=1}^{l} \big(\int_t^{\bar{t}} \delta Z_s^\iota dW_s + \int_t^{\bar{t}} \delta U_s^\iota d\mu_s\big)^2. \tag{10}$$

As a local martingale in $\mathcal{S}_2$, each process $\int_t^\cdot \delta Z_s^\iota dW_s + \int_t^\cdot \delta U_s^\iota d\mu_s$ is a square integrable martingale over $[t, \bar{t}]$. The (conditional) Burkholder inequality applied to this process then yields

$$\mathbb{E}_t\big(\int_t^{\bar{t}} \delta Z_s^\iota dW_s + \int_t^{\bar{t}} \delta U_s^\iota d\mu_s\big)^2 \leq C \mathbb{E}_t\Big[\int_t^{\bar{t}} \big(|\delta Z_s^\iota|^2 + |\delta U^\iota|_s^2\big) ds\Big],$$

so that (10) entails (9). ∎

Theorem 2.1 *The ABSDE* (7) *has a unique special semimartingale solution* Y *in* $\mathcal{S}_2^l$ *with martingale component* M *in* $\mathcal{S}_2^l$. *The process* Y *is the limit in* $\mathcal{S}_2^l$ *of the Picard iteration defined by* $Y^0 = 0$ *and, for* $j \geq 1$,

$$Y_t^j = \mathbb{E}_t\Big[\phi(\mathcal{X}_T) + \int_t^T f\big(s, \mathcal{X}_s, Y_s^{j-1}, \mathbb{ES}_s(\Phi_{\bar{s}}(M^{j-1}))\big)ds\Big], \tag{11}$$

where $M^{j-1} \in \mathcal{S}_2^l$ *is the martingale part of the special semimartingale* $Y^{(j-1)} \in \mathcal{S}_2^l$.

Proof. Assumptions 2.2(i) and (ii) imply that the processes

$$\sup_{|y|\leq c} |f(\cdot, \mathcal{X}_\cdot, y, \mathbb{ES}_\cdot\Phi_\cdot(0)) - f(\cdot, \mathcal{X}_\cdot, 0, \mathbb{ES}_\cdot\Phi_\cdot(0))|^{\frac{1}{2}}$$

(for every $c > 0$), as well as $|f(\cdot, \mathcal{X}_\cdot, 0, \mathbb{ES}_\cdot\Phi_\cdot(0))|$, are in $\mathcal{H}_2$, which is (Crépey et al., 2020, Assumption 3.2(iii)), whereas (Crépey et al., 2020, Assumption 3.2 (i), (ii), and (iv)) are implied by our Assumption 2.2(i) and the Lipschitz property of the functions ϕ_k combined with the standard a priori bound estimate $\|X\|^2_{\mathcal{S}_2^p} \leq C\left(1 + |x|^2\right)$ on X (with constant initial condition x). Moreover, (9) corresponds to (Crépey et al., 2020, Assumption 3.1). Hence (Crépey et al., 2020, Assumptions 3.1 and 3.2) hold and the result follows by an application of (Crépey, Sabbagh, and Song, 2020, Theorem 3.1). ∎

3 Approximation Schemes

3.1 Time Discretizations

Let there be given a deterministic time-grid $0 = t_0 < t_1 < \cdots < t_n = T$ with mesh size[12] h. We write $\Delta t_{i+1} = t_{i+1} - t_i$. Let $\bar{t}_i$ denote an approximation on the grid of $\overline{t}_i$[13]. Let there also be given, on this time grid, simulatable approximations $\mathcal{X}^h$ (assumed Markovian with respect to its own filtration) to $\mathcal{X}$[14] and $\Phi^h_{\bar{t}_i}(M^h)$ to $\Phi_{\bar{t}}(M)$, with M^h approximating M on the time-grid and $\Phi^h_{\bar{t}_i}(M^h)$ of the form[15]

$$\Phi^h\big(t_i; \mathcal{X}^h_{\{t_i,\cdots,\bar{t}_i\}}, M^h_{\{t_i,\cdots,\bar{t}_i\}} - M^h_{t_i}\big), \tag{12}$$

for some deterministic map Φ^h of grid times t_i and discrete paths $\mathbf{x}^h$ and $\mathbf{m}^h$ on $\{t_i, \cdots, \bar{t}_i\}$ such that $\mathbf{m}^h_{t_i} = 0$.

The explicit time discretization for $(Y, \mathbb{ES}_\cdot(\Phi_\cdot(M)))$ (with M the martingale part of the solution Y to (7)) is the process (Y^h, ρ^h) defined at grid times by $Y^h_{t_n} = \phi(\mathcal{X}^h_T), \rho^h_{t_n} =$

[12] maximum time step.
[13] cf. after (6).
[14] e.g. the Euler scheme for X and a related approximation for J.
[15] cf. (6).

$\Phi_T^h(0)$ and, for i decreasing from $n-1$ to 0,

$$\begin{aligned} Y_{t_i}^h &= \mathbb{E}_{t_i}^h\Big[Y_{t_{i+1}}^h + f\big(t_i, \mathcal{X}_{t_i}^h, Y_{t_{i+1}}^h, \rho_{t_{i+1}}^h\big)\Delta t_{i+1}\Big] \\ \rho_{t_i}^h &= \mathbb{ES}_{t_i}^h\Big(\Phi_{\bar{t}_i}^h\big(\underbrace{Y_{t_\cdot}^h + \sum_{\imath<\cdot} f(t_\imath, \mathcal{X}_{t_\imath}^h, Y_{t_{\imath+1}}^h, \rho_{t_{\imath+1}}^h)\Delta t_{\imath+1}}_{M^h}\big)\Big), \end{aligned} \tag{13}$$

where the $\cdot^h$ in $\mathbb{E}_{t_i}^h$ and $\mathbb{ES}_{t_i}^h$ mean that the corresponding conditional expectations and expected shortfalls are in reference to the filtration of $\mathcal{X}^h$.

The Picard iteration associated with the implicit time discretization for $(Y, \mathbb{ES}_\cdot\Phi_\cdot(M))$ is defined by the sequence of discrete time processes $\rho^{0,h} = 0$ and, for j increasing from 1 to ∞: $Y_{t_n}^{j,h} = \phi(\mathcal{X}_T^h)$ and, for i decreasing from $n-1$ to 0,

$$\begin{aligned} Y_{t_i}^{j,h} &= \mathbb{E}_{t_i}^h\Big[Y_{t_{i+1}}^{j,h} + f\big(t_i, \mathcal{X}_{t_i}^h, \check{Y}_{t_i}^{j-1,h}, \rho_{t_i}^{j-1,h}\big)\Delta t_{i+1}\Big], \\ \rho_{t_i}^{j,h} &= \mathbb{ES}_{t_i}^h\Big(\Phi_{\bar{t}_i}^h\big(\underbrace{Y_{t_\cdot}^{j,h} + \sum_{\imath<\cdot} f(t_\imath, \mathcal{X}_{t_\imath}^h, \check{Y}_{t_\imath}^{j-1,h}, \rho_{t_\imath}^{j-1,h})\Delta t_{\imath+1}}_{M^{j,h}}\big)\Big), \end{aligned} \tag{14}$$

where

$$\check{Y}_{t_i}^{j-1,h} = \mathbb{1}_{j=1} Y_{t_{i+1}}^{j,h} + \mathbb{1}_{j\geq 2} Y_{t_i}^{j-1,h}. \tag{15}$$

Remark 3.1 *For this variant of the Picard scheme, $Y^{j,h}$ corresponds to an explicit scheme for the standard BSDE resulting from the replacement of $\mathbb{ES}_s(\Phi_{\bar{s}}(M))$ by 0 in (7). In the XVA setup of Section 4, this first Picard iteration yields the XVA numbers ignoring the possibility to use capital at risk for variation margin funding purposes (Albanese et al., 2017, Section 5.2), the comparison of which with the final XVA numbers, obtained in the limit of the Picard iteration, is interesting from a financial viewpoint.*

The time-consistency of these schemes, i.e. the convergence of the Y^h (resp. $Y^{h,j}$) to Y as h goes to 0 (resp. h goes to 0 and j goes to infinity), can be studied by the techniques initiated in (Bouchard and Touzi, 2004; Zhang, 2004)[16]. Our focus hereafter, instead, is the discretization in space of (13), (14).

3.2 Fully Discrete Algorithms

Whenever a process M^h on the time grid is such that $M^h_{\{t=t_i,\cdots,\bar{t}_i\}} - M^h_{t_i}$ is a measurable functional of $(t_i, \mathcal{X}^h_{t_i}), \ldots, (\bar{t}_i, \mathcal{X}^h_{\bar{t}_i})$ with $\Phi^h_{\bar{t}}(M^h)$ square integrable, Barrera, Crépey,

[16] see Section 1.

Gobet, Nguyen, and Saadeddine (2022, Theorem 2.3)[17] yields, with $\varphi = \varphi_k(t,x)$ and $\phi = \phi_k(t,x)$: $\mathbb{ES}_t^h(\Phi_{\bar{t}}^h(M^h)) = (1-\alpha)^{-1}\phi^*(t,\mathcal{X}_t^h)$, where

$$\phi_\cdot^*(t,\cdot) = \underset{\phi_\cdot(t,\cdot)\in\mathcal{B}}{\arg\min}\,\mathbb{E}[(\Phi_{\bar{t}}^h(M^h)\mathbb{1}_{\{\Phi_{\bar{t}}^h(M^h)\geq\varphi^*(t,\mathcal{X}_t^h)\}} - \phi(t,\mathcal{X}_t^h))^2], \tag{16}$$

in which

$$\varphi_\cdot^*(t,\cdot) = \underset{\varphi_\cdot(t,\cdot)\in\mathcal{B}}{\arg\min}\,\mathbb{E}[(\varphi(t,\mathcal{X}_t^h) + (1-\alpha)^{-1}(\Phi_{\bar{t}}^h(M^h) - \varphi(t,\mathcal{X}_t^h))^+], \tag{17}$$

both minimizations bearing over the set $\mathcal{B}$ of the Borel functions of (x,k).

By nonparametric quantile regression estimates of $\varphi^*(t,\mathcal{X}_t^h)$ and $\phi^*(t,\mathcal{X}_t^h)$, we mean any functions $\widehat{\varphi}^*(t,\cdot)$ and $\widehat{\phi}^*(t,\cdot)$ obtained by solving the respective problems (17) and (16) with $\mathcal{B}$ approximated by a to-be-specified hypothesis space of functions, $\mathbb{E}$ by the sample mean over a sufficiently large number of independent realizations of $\mathcal{X}^h$, minimization by numerical minimization through Adam stochastic gradient descent (Kingma and Ba, 2015), and φ^* in (16) by $\widehat{\varphi}^*$.

The fully (time and space) discrete counterparts of (13) and (14) follow by estimating, at each grid time $t = t_i$ going backward, the embedded conditional expectations (resp. expected shortfalls) through nonparametric least-squares regression against $\mathcal{X}_t^h$ (resp. quantile regression against $\mathcal{X}_t^h$ as explained above), which we write: $\widehat{Y}_{t_n}^h = \phi(\mathcal{X}_T^h), \widehat{\rho}_{t_n}^h = \Phi_T^h(0)$ and, for i decreasing from $n-1$ to 0,

$$\begin{aligned}
\widehat{Y}_{t_i}^h &= \widehat{\mathbb{E}}_{t_i}^h\Big[\widehat{Y}_{t_{i+1}}^h + f\big(t_i,\mathcal{X}_{t_i}^h,\widehat{Y}_{t_{i+1}}^h,\widehat{\rho}_{t_{i+1}}^h\big)\Delta t_{i+1}\Big]\\
\widehat{\rho}_{t_i}^h &= \widehat{\mathbb{ES}}_{t_i}^h\Big(\Phi_{\bar{t}_i}^h\big(\underbrace{Y_{t_\cdot}^h + \sum_{\imath<\cdot} f(t_\imath,\mathcal{X}_{t_\imath}^h,\widehat{Y}_{t_{\imath+1}}^h,\widehat{\rho}_{t_{\imath+1}}^h)\Delta t_{\imath+1}}_{\widehat{M}^h}\big)\Big),
\end{aligned} \tag{18}$$

respectively $\widehat{\rho}^{0,h} = 0$ and, for j increasing from 1 to ∞: $\widehat{Y}_{t_n}^{j,h} = \phi(\mathcal{X}_T^h)$ and, for i decreasing from $n-1$ to 0,

$$\begin{aligned}
\widehat{Y}_{t_i}^{j,h} &= \widehat{\mathbb{E}}_{t_i}^h\Big[\widehat{Y}_{t_{i+1}}^{j,h} + f\big(t_i,\mathcal{X}_{t_i}^h,\widetilde{Y}_{t_i}^{j-1,h},\widehat{\rho}_{t_i}^{j-1,h}\big)\Delta t_{i+1}\Big],\\
\widehat{\rho}_{t_i}^{j,h} &= \widehat{\mathbb{ES}}_{t_i}^h\Big(\Phi_{\bar{t}_i}^h\big(\underbrace{\widehat{Y}_{t_\cdot}^{j,h} + \sum_{\imath<\cdot} f(t_\imath,\mathcal{X}_{t_\imath}^h,\widetilde{Y}_{t_\imath}^{j-1,h},\widehat{\rho}_{t_\imath}^{j-1,h})\Delta t_{\imath+1}}_{\widehat{M}^{j,h}}\big)\Big),
\end{aligned} \tag{19}$$

where[18]

$$\widetilde{Y}_{t_i}^{j-1,h} = \mathbb{1}_{j=1}\widehat{Y}_{t_{i+1}}^{j,h} + \mathbb{1}_{j\geq 2}\widehat{Y}_{t_i}^{j-1,h}. \tag{20}$$

Note that

$$\widehat{M}_{t_i}^h = \widehat{M}_{t_{i+1}}^h + \widehat{Y}_{t_i}^h - \mathcal{Y}_{t_{i+1}}^h \text{ , where } \mathcal{Y}_{t_{i+1}}^h = \widehat{Y}_{t_{i+1}}^h + f\big(t_i,\mathcal{X}_{t_i}^h,\widehat{Y}_{t_{i+1}}^h,\widehat{\rho}_{t_{i+1}}^h\big)\Delta t_{i+1}$$

[17] additionally assuming $\Phi_{\bar{t}}^h(M^h)$ atomless given $\mathfrak{F}_t$.

[18] cf. (15) and Remark 3.1.

and

$$\widehat{M}_{t_i}^{j,h} = \widehat{M}_{t_{i+1}}^{j,h} + \widehat{Y}_{t_i}^{j,h} - \mathcal{Y}_{t_{i+1}}^{j,h} \text{ , where } \mathcal{Y}_{t_{i+1}}^{j,h} = \widehat{Y}_{t_{i+1}}^{j,h} + f\big(t_i, \mathcal{X}_{t_i}^h, \widetilde{Y}_{t_i}^{j-1,h}, \widehat{\rho}_{t_i}^{j-1,h}\big)\Delta t_{i+1},$$

which is exploited in the end of the respective Algorithms 1 and 2 below to compute $\widehat{M}_{t_i}^h$ and $\widehat{M}_{t_i}^{j,h}$ iteratively (for decreasing i).

As in Abbas-Turki, Crépey, and Saadeddine (2022), given weight matrices $A^{[L+1]} \in \mathbb{R}^{1\times u}, \ldots, A^{[\ell]} \in \mathbb{R}^{u\times u}, \ldots, A^{[1]} \in \mathbb{R}^{u\times(p+q)}$, biases $b^{[L+1]} \in \mathbb{R}, \ldots, b^{[\ell]} \in \mathbb{R}^u, \ldots, b^{[1]} \in \mathbb{R}^u$, and a scalar non-linearity ς applied element-wise, let, for every $z = (x,k) \in \mathbb{R}^p \times \mathbb{R}^q \equiv \mathbb{R}^{p+q}$,

$$\begin{aligned}
&\zeta^{[0]}(z; A, b) = z\\
&\zeta^{[\ell]}(z; A, b) = \varsigma(A^{[\ell]}\zeta^{[\ell-1]}(z; A, b) + b^{[\ell]}), \quad \ell = 1, \ldots, L\\
&\zeta^{[L+1]}(z; A, b) = A^{[L+1]}\zeta^{(L)}(z; A, b) + b^{[L+1]},
\end{aligned}$$

where A and b denote the respective concatenations of the $A^{[\ell]}$ and of the $b^{[\ell]}$. The function $\mathbb{R}^{p+q} \ni z \mapsto \zeta^{[L+1]}(z; A, b) \in \mathbb{R}$ then implements a neural network with L hidden layers, u neurons per hidden layer, ς as an activation function applied on each hidden unit, and no non-linearity at the output layer. Using the corresponding parameterized set of functions $\mathbb{R}^{p+q} \ni z \mapsto \zeta^{[L+1]}(z; A, b) \in \mathbb{R}$ as hypothesis space in the description following (16)-(17), we obtain Algorithms 1 (for the explicit scheme) and 2 (for the implicit/Picard scheme) below, both using Algorithm 0 as an elementary learning block. At the beginning of the algorithms, i.e. before any learning is performed, the weights are initialized randomly the way explained in Goodfellow, Bengio, and Courville (2016). In our numerical experiments, we use softplus activation functions in the hidden layers, in combination with the related weight initialization scheme in He, Zhang, Ren, and Sun (2015).

Remark 3.2 *Assuming the same number of epochs during each run of the elementary learning block of Algorithm 0 for all schemes, the Picard scheme with j Picard iterations implies j times more regressions than the explicit scheme. Hence, by design, it is at a computational disadvantage compared to the explicit scheme of Algorithm 1. In order to try and make the Picard scheme more competitive, we propose to modify Algorithm 2 by leveraging its iterative nature through Picard iterations. Namely, given a computational budget equivalent to n_{train} training epochs in the explicit scheme, assuming that we have a target of j Picard iterations in the implicit scheme, we only train for only n_{train}/j epochs at each Picard iteration, reusing the obtained neural network weights as an initialization for the neural networks at the next Picard iteration (for $j \geq 2$), versus at the next time step in Algorithms 1 and 2 (and also for $j = 1$ in the presently considered variant of Algorithm 2). This ensures that the total computational cost of j Picard iterations is roughly the same as the cost of learning in the explicit scheme. This modification is achieved by changing the $i+1$ to $i + \mathbb{1}_{\{j=1\}}$ in lines 10, 12 and 32 of Algorithm 2.*

Algorithm 0: Elementary learning block for least squares (ls) and quantile (qle) neural net regressions of cash flows $\xi^\jmath$ against features $\chi^\jmath$ indexed by $\jmath \in \mathcal{J}$

name : BaseAlg
input : $\{(\chi^\jmath, \xi^\jmath), \jmath \in \mathcal{J}\}$, a partition B of $\mathcal{J}$, a number of epochs $E \in \mathbb{N}^\star$, a learning rate $\eta > 0$, initial values for the network parameters A and b, type of regression *regr*
output: Trained parameters A and b

define $\mathcal{L}(A, b, \text{batch}) =$

$$\begin{cases} \dfrac{1}{|\text{batch}|} \displaystyle\sum_{\jmath \in \text{batch}} (\zeta^{[L+1]}(\chi^\jmath; A, b) - \xi^\jmath)^2 & \text{if } \textit{regr} = \text{ls} \\ \dfrac{1}{|\text{batch}|} \displaystyle\sum_{\jmath \in \text{batch}} (\xi^\jmath - \zeta^{[L+1]}(\chi^\jmath; A, b))^+ + (1-\alpha)\zeta^{[L+1]}(\chi^\jmath; A, b) & \text{if } \textit{regr} = \text{qle} \end{cases}$$

for $\textit{epoch} = 1, \ldots, E$ **do** `// loop over epochs`
  for $\textit{batch} \in B$ **do** `// loop over batches`
    for $\ell = 1, \ldots, L+1$ **do**
      $A^{[\ell]} \leftarrow A^{[\ell]} - \eta \nabla_{A^{[\ell]}} \mathcal{L}(A, b, \text{batch})$
      $b^{[\ell]} \leftarrow b^{[\ell]} - \eta \nabla_{b^{[\ell]}} \mathcal{L}(A, b, \text{batch})$
    end
  end
end

3.3 A Posteriori Analysis of the Regression Error

A well-established BSDE spatial error analysis strategy consists in analysing the accumulation, over (discrete) time i decreasing from $n-1$ to 0, of three error components (Gobet, 2016, Eqn. (VIII.3.8)): (i) a *bias* between the function u^i representing $Y^h_{t_i}$ as $u^i(\mathcal{X}^h_{t_i})$ (for a suitable measurable function $u^i = u^i_k(x)$) and the hypothesis space of functions in which $\widehat{Y}^h_{t_i}$ is sought after, (ii) a *"variance"* in the sense of the regression estimation error, and (iii) a term of *propagation* at time i of the error at time $i+1$. This is at least the strategy in the standard case where the embedded conditional expectations are estimated by parametric least-squares regressions that can be performed exactly, for instance by singular value decomposition (Gobet, 2016, Section VIII.2.2). Neural net parameterizations for the targeted functions (conditional expectations or expected shortfalls in the case of our ABSDEs) instead lead to "nonlinear regressions" that can only be performed by numerical, nonconvex minimization. When state-of-the-art, fine-tuned, Adam variants of stochastic gradient descents are used in this regard, the ensuing minimization can be very efficient numerically. However, there is no known learning algorithm solving such nonconvex minimization problems with an a priori error bound. Hence, when the learning iteration terminates, we do not have any guarantee on the quality of the approximation. In other words, there is a fourth *numerical minimization* error component on top of the three other ones in the above and this fourth error component cannot be controlled ex ante.

Algorithm 1: Explicit backward learning scheme for the ABSDE (7) based on time-discretized simulated paths $\mathcal{X}^{h,\jmath}$ of $\mathcal{X}$ indexed by $\jmath \in \mathcal{J}$

name : ExplicitBackwardAlg
input : current state $\mathcal{X}_0 = x, \{\{\mathcal{X}^{h,\jmath}_{t_i}, 1 \leq i \leq n\}, \phi(\mathcal{X}^{h,\jmath}_T), \jmath \in \mathcal{J}\}$, a partition B of $\mathcal{J}$, a number of epochs $E \in \mathbb{N}^\star$, a learning rate $\eta > 0$
output: $\widehat{Y}^h_0$ and learned parameters $\{(A^{\mathbb{V}\mathrm{a}\mathbb{R}}_i, b^{\mathbb{V}\mathrm{a}\mathbb{R}}_i), (A^{\mathbb{E}\mathbb{S}}_i, b^{\mathbb{E}\mathbb{S}}_i), (A^\iota_i, b^\iota_i), \iota \in \{1, \ldots, l\}, i \in \{1, \ldots, n\}\}$

For all $\jmath \in \mathcal{J}$, let $y^\jmath \in \mathbb{R}^l$ and, for each $i = 0 \cdots n$, $\widehat{M}^{h,\jmath}_{t_i} = 0 \in \mathbb{R}^l$
Initialize parameters $\{(A^{\mathbb{V}\mathrm{a}\mathbb{R}}_n, b^{\mathbb{V}\mathrm{a}\mathbb{R}}_n), (A^{\mathbb{E}\mathbb{S}}_n, b^{\mathbb{E}\mathbb{S}}_n)\}$ of the networks approximating the $\mathbb{V}\mathrm{a}\mathbb{R}$ and $\mathbb{E}\mathbb{S}$ at terminal time-step n
Initialize the parameters $\{(A^\iota_n, b^\iota_n), \iota \in \{1, \ldots, l\}\}$ of the least-squares networks, indexed by $\iota \in \{1, \ldots, l\}$, at terminal time-step n
foreach $\jmath \in \mathcal{J}$ **do** $y^\jmath \leftarrow \phi(\mathcal{X}^{h,\jmath}_T)$, $\widehat{M}^{h,\jmath}_{t_n} \leftarrow 0$
for $i = n - 1 \ldots 1$ **do**
  foreach $\jmath \in \mathcal{J}$ **do** $\xi^\jmath \leftarrow \Phi^h_{t_i}(\widehat{M}^{h,\jmath})$ `// We first learn the` $\mathbb{V}\mathrm{a}\mathbb{R}$ `using a quantile regression`
  $A^{\mathbb{V}\mathrm{a}\mathbb{R}}_i, b^{\mathbb{V}\mathrm{a}\mathbb{R}}_i \leftarrow \mathrm{BaseAlg}(\{(\mathcal{X}^{h,\jmath}_{t_i}, \xi^\jmath), \jmath \in \mathcal{J}\}, B, E, \eta, A^{\mathbb{V}\mathrm{a}\mathbb{R}}_{i+1}, b^{\mathbb{V}\mathrm{a}\mathbb{R}}_{i+1}, \mathrm{qle})$ `// Then we deduce the` $\mathbb{E}\mathbb{S}$ `using an ls regression`
  $\xi^\jmath \leftarrow \frac{1}{1-\alpha}\xi^\jmath \mathbf{1}_{\{\xi^\jmath \geq \zeta^{[L+1]}(\mathcal{X}^{h,\jmath}_{t_i}; A^{\mathbb{V}\mathrm{a}\mathbb{R}}_i, b^{\mathbb{V}\mathrm{a}\mathbb{R}}_i)\}}$
  $A^{\mathbb{E}\mathbb{S}}_i, b^{\mathbb{E}\mathbb{S}}_i \leftarrow \mathrm{BaseAlg}(\{(\mathcal{X}^{h,\jmath}_{t_i}, \xi^\jmath), \jmath \in \mathcal{J}\}, B, E, \eta, A^{\mathbb{E}\mathbb{S}}_{i+1}, b^{\mathbb{E}\mathbb{S}}_{i+1}, \mathrm{ls})$
  `// We compute the integrand` ξ `that needs to be projected to get the solution of the BSDE at the current time step`
  $\varrho^\jmath \leftarrow \zeta^{[L+1]}(\mathcal{X}^{h,\jmath}_{t_{i+1}}; A^{\mathbb{E}\mathbb{S}}_{i+1}, b^{\mathbb{E}\mathbb{S}}_{i+1})$
  $\xi^\jmath \leftarrow y^\jmath + f(t_i, \mathcal{X}^{h,\jmath}_{t_i}, y^\jmath, \varrho^\jmath)\Delta t_{i+1}$
  for $\iota = 1 \ldots l$ **do**
    $A^\iota_i, b^\iota_i \leftarrow \mathrm{BaseAlg}(\{(\mathcal{X}^{h,\jmath}_{t_i}, \xi^\jmath_\iota), \jmath \in \mathcal{J}\}, B, E, \eta, A^\iota_{i+1}, b^\iota_{i+1}, \mathrm{ls})$
  end
  `// Compute` $\widehat{M}^h_{t_i}$
  for $\jmath \in \mathcal{J}$ **do**
    for $\iota = 1 \ldots l$ **do**
      $y^\jmath_\iota \leftarrow \zeta^{[L+1]}(\mathcal{X}^{h,\jmath}_{t_i}; A^\iota_i, b^\iota_i)$
    end
    $\widehat{M}^{h,\jmath}_{t_i} \leftarrow \widehat{M}^{h,\jmath}_{t_{i+1}} + y^\jmath - \xi^\jmath$
  end
end
for $\jmath \in \mathcal{J}$ **do**
  $\varrho^\jmath \leftarrow \zeta^{[L+1]}(\mathcal{X}^{h,\jmath}_{t_1}; A^{\mathbb{E}\mathbb{S}}_1, b^{\mathbb{E}\mathbb{S}}_1)$
  $\xi^\jmath \leftarrow y^\jmath + f(0, x, y^\jmath, \varrho^\jmath)\Delta t_1$
end
$\widehat{Y}^h_0 \leftarrow sampleMean(\xi)$

Algorithm 2: Picard backward learning scheme for the ABSDE (7) based on time-discretized simulated paths $\mathcal{X}^{h,\jmath}$ of $\mathcal{X}$ indexed by $\jmath \in \mathcal{J}$

name : PicardBackwardAlg
input : current state $\mathcal{X}_0 = x$, $\{\{\mathcal{X}^{h,\jmath}_{t_i}, 1 \le i \le n\}, \phi(\mathcal{X}^{h,\jmath}_T), \jmath \in \mathcal{J}\}$, a partition B of $\mathcal{J}$, a number of epochs $E \in \mathbb{N}^\star$, a learning rate $\eta > 0$ and number P of picard iterations
output: $\widehat{Y}^{P,h}_0$ and learned parameters $\{(A^{\mathbb{V}\mathrm{a}\mathbb{R}}_i, b^{\mathbb{V}\mathrm{a}\mathbb{R}}_i), (A^{\mathbb{E}\mathbb{S}}_i, b^{\mathbb{E}\mathbb{S}}_i), (A^\iota_i, b^\iota_i), \iota \in \{1,\dots,l\}, i \in \{1,\dots,n\}\}$

For all $\jmath \in \mathcal{J}$, let $y^\jmath \in \mathbb{R}^l$ and, for each $i = 0 \cdots n$, $\widehat{M}^{\mathrm{curr},h,\jmath}_{t_i} = \widehat{M}^{\mathrm{prev},h,\jmath}_{t_i} = 0 \in \mathbb{R}^l$
Initialize parameters $\{(A^{\mathbb{V}\mathrm{a}\mathbb{R}}_n, b^{\mathbb{V}\mathrm{a}\mathbb{R}}_n), (A^{\mathbb{E}\mathbb{S}}_n, b^{\mathbb{E}\mathbb{S}}_n)\}$ of the networks approximating the $\mathbb{V}\mathrm{a}\mathbb{R}$ and $\mathbb{E}\mathbb{S}$ at terminal time-step n
Initialize the parameters of the networks $\{(A^\iota_n, b^\iota_n), \iota \in \{1,\dots,l\}\}$ at terminal time-step n
foreach $\jmath \in \mathcal{J}$ **do** $y^\jmath \leftarrow \phi(\mathcal{X}^{h,\jmath}_{t_n})$, $\widehat{M}^{\mathrm{prev},h,\jmath}_{t_n} \leftarrow 0$
`// curr and prev below = current and previous Picard iterations`
for $j = 1 \dots P$ **do**
  foreach $\jmath \in \mathcal{J}$ **do** $\widehat{M}^{\mathrm{curr},h,\jmath}_{t_n} \leftarrow 0$
  for $i = n-1 \dots 1$ **do**
    foreach $\jmath \in \mathcal{J}$ **do** $\xi^\jmath \leftarrow \Phi^h_{t_i}(\widehat{M}^{prev,h,\jmath})$
    if $j > 1$ **then**
      `// We first learn the` $\mathbb{V}\mathrm{a}\mathbb{R}$ `using a quantile regression`
      $A^{\mathbb{V}\mathrm{a}\mathbb{R}}_i, b^{\mathbb{V}\mathrm{a}\mathbb{R}}_i \leftarrow \mathrm{BaseAlg}(\{(\mathcal{X}^{h,\jmath}_{t_i}, \xi^\jmath), \jmath \in \mathcal{J}\}, B, E, \eta, A^{\mathbb{V}\mathrm{a}\mathbb{R}}_{i+1}, b^{\mathbb{V}\mathrm{a}\mathbb{R}}_{i+1}, \mathrm{qle})$
      `// Then we deduce the` $\mathbb{E}\mathbb{S}$ `using an ls regression`
      $\xi^\jmath \leftarrow \frac{1}{1-\alpha}\xi^\jmath \mathbf{1}_{\{\xi^\jmath \ge \zeta^{[L+1]}(\mathcal{X}^{h,\jmath}_{t_i}; A^{\mathbb{V}\mathrm{a}\mathbb{R}}_i, b^{\mathbb{V}\mathrm{a}\mathbb{R}}_i)\}}$
      $A^{\mathbb{E}\mathbb{S}}_i, b^{\mathbb{E}\mathbb{S}}_i \leftarrow \mathrm{BaseAlg}(\{(\mathcal{X}^{h,\jmath}_{t_i}, \xi^\jmath), \jmath \in \mathcal{J}\}, B, E, \eta, A^{\mathbb{E}\mathbb{S}}_{i+1}, b^{\mathbb{E}\mathbb{S}}_{i+1}, \mathrm{ls})$
    end
    `// We compute the integrand` ξ `that needs to be projected to get the solution of the BSDE at the current time step`
    for $\jmath \in \mathcal{J}$ **do**
      foreach $\iota = 1 \dots l$ **do** $y^\jmath_\iota \leftarrow \zeta^{[L+1]}(\mathcal{X}^{h,\jmath}_{t_{i+1}}; A^\iota_{i+1}, b^\iota_{i+1})$
      $\xi^\jmath \leftarrow y^\jmath$
      if $j > 1$ **then**
        foreach $\iota = 1 \dots l$ **do** $y^\jmath_\iota \leftarrow \zeta^{[L+1]}(\mathcal{X}^{h,\jmath}_{t_i}; A^\iota_i, b^\iota_i)$
      end
      $\varrho^\jmath \leftarrow \zeta^{[L+1]}(\mathcal{X}^{h,\jmath}_{t_i}; A^{\mathbb{E}\mathbb{S}}_i, b^{\mathbb{E}\mathbb{S}}_i)$ if $j > 1$, and 0 otherwise
      $\xi^\jmath \leftarrow \xi^\jmath + f(t_i, \mathcal{X}^{h,\jmath}_{t_i}, y^\jmath, \varrho^\jmath)\Delta t_{i+1}$
    end
    if $i = 1$ **then**
      if $j > 1$ **then**
        $mean^{prev} \leftarrow mean^{curr}$
        $es^{prev} \leftarrow sampleES(\Phi^h_{t_0}(\widehat{M}^{prev,h,\jmath}), \jmath \in \mathcal{J})$
        $mean^{curr} \leftarrow sampleMean(\zeta^{[L+1]}(\mathcal{X}^{h,\jmath}_{t_1}; A^\iota_1, b^\iota_1), \jmath \in \mathcal{J}) + f(t_0, \mathcal{X}^{h,j}_{t_0}, mean^{prev}, es^{prev})\Delta t_1$
      else
        $mean^{curr} \leftarrow sampleMean(\zeta^{[L+1]}(\mathcal{X}^{h,\jmath}_{t_1}; A^\iota_1, b^\iota_1) + f(t_1, \mathcal{X}^{h,j}_{t_1}, y^\jmath, \rho^\jmath)\Delta t_1, \jmath \in \mathcal{J})$
      end
    end
    foreach $\iota = 1 \dots l$ **do** $A^\iota_i, b^\iota_i \leftarrow \mathrm{BaseAlg}(\{(\mathcal{X}^{h,\jmath}_{t_i}, \xi^\jmath_\iota), \jmath \in \mathcal{J}\}, B, E, \eta, A^\iota_{i+1}, b^\iota_{i+1}, \mathrm{ls})$
    `// Compute` $\widehat{M}^{\mathrm{curr},h}_{t_i}$
    if $j > 1$ **then**
      foreach $\iota = 1 \dots l$ **do** $y^\jmath_\iota \leftarrow \zeta^{[L+1]}(\mathcal{X}^{h,\jmath}_{t_i}; A^\iota_i, b^\iota_i)$
    end
    foreach $\jmath \in \mathcal{J}$ **do** $\widehat{M}^{h,\mathrm{curr},\jmath}_i \leftarrow \widehat{M}^{\mathrm{curr},h,\jmath}_{t_{i+1}} + y^\jmath - \xi^\jmath$
  end
  foreach $i = n-1 \dots 1$ **and** $\jmath \in \mathcal{J}$ **do** $\widehat{M}^{\mathrm{prev},h,\jmath}_{t_i} \leftarrow \widehat{M}^{\mathrm{curr},h,\jmath}_{t_i}$
end
$\widehat{Y}^{P,h}_0 \leftarrow sampleMean(\zeta^{[L+1]}(\mathcal{X}^{h,\jmath}_{t_1}; A^\iota_1, b^\iota_1), \jmath \in \mathcal{J}) + f(t_0, \mathcal{X}^{h,j}_{t_0}, mean^{curr}, es)\Delta t_1$

However, we can still assess the local regression error of the schemes by the following *a posteriori Monte Carlo procedure*. We assume $l = 1$ and a uniform time step $\Delta t_{i+1} = \Delta t$, in the XVA motivated case[19]

$$\bar{t} = (t+1) \wedge T \text{ and } \Phi_{\bar{t}}(M) = M_{\bar{t}} - M_t. \tag{21}$$

The subsequent developments are done in the case of the explicit scheme, but similar computations would yield similar outputs in the case of the implicit/Picard scheme. Letting $m = [\frac{1}{\Delta t}]$, the time-explicit fully discrete scheme (18) here reduces to $\widehat{Y}^h_{t_n} = \phi(\mathcal{X}^h_T), \widehat{\rho}^h_{t_n} = 0$ and, for i decreasing from $n-1$ to 0,

$$\begin{aligned}
\widehat{Y}^h_{t_i} &= \widehat{\mathbb{E}}^h_{t_i}[\widehat{Y}^h_{t_{i+1}} + f(t_i, \mathcal{X}^h_{t_i}, \widehat{Y}^h_{t_{i+1}}, \widehat{\rho}^h_{t_{i+1}})\Delta t] \\
\widehat{\rho}^h_{t_i} &= \widehat{\mathbb{ES}}^h_{t_i}[\widehat{Y}^h_{t_{(i+m)\wedge n}} - \widehat{Y}^h_{t_i} + \sum_{\imath=i}^{(i+m-1)\wedge(n-1)} f(t_\imath, \mathcal{X}^h_{t_\imath}, \widehat{Y}^h_{t_{\imath+1}}, \widehat{\rho}^h_{t_{\imath+1}})\Delta t].
\end{aligned}$$

We also define the auxiliary scheme $\widetilde{Y}^h_{t_n} = \phi(\mathcal{X}^h_T), \widetilde{\rho}^h_{t_n} = 0$ and, for i decreasing from $n-1$ to 0,

$$\begin{aligned}
\widetilde{Y}^h_{t_i} &= \mathbb{E}^h_{t_i}[\widehat{Y}^h_{t_{i+1}} + f(t_i, \mathcal{X}^h_{t_i}, \widehat{Y}^h_{t_{i+1}}, \widehat{\rho}^h_{t_{i+1}})\Delta t] \\
\widetilde{\rho}^h_{t_i} &= \mathbb{ES}^h_{t_i}[\widehat{Y}^h_{t_{(i+m)\wedge n}} - \widehat{Y}^h_{t_i} + \sum_{\imath=i}^{(i+m-1)\wedge(n-1)} f(t_\imath, \mathcal{X}^h_{t_\imath}, \widehat{Y}^h_{t_{\imath+1}}, \widehat{\rho}^h_{t_{\imath+1}})\Delta t].
\end{aligned}$$

Let

$$\epsilon_{t_i} = |\widetilde{Y}^h_{t_i} - \widehat{Y}^h_{t_i}|. \tag{22}$$

Proceeding as in Abbas-Turki, Crépey, and Saadeddine (2022, Section 5.1), one can estimate $\mathbb{E}[\epsilon^2_{t_i}]$ *without computing* $\widetilde{Y}^h_{t_i}$, by Monte Carlo using a so-called twin simulation scheme. The latter consists in a Monte Carlo estimation of $\mathbb{E}[\epsilon^2_{t_i}]$ based on the following representation involving two copies $\widehat{Y}^{h,1}_{t_{i+1}}$ and $\widehat{Y}^{h,2}_{t_{i+1}}$ of $\widehat{Y}^h_{t_{i+1}} = u^{i+1}(\mathcal{X}^h_{t_{i+1}})$ and $\widehat{\rho}^{h,1}_{t_{i+1}}$ and $\widehat{\rho}^{h,2}_{t_{i+1}}$ of $\widehat{\rho}^h_{t_{i+1}} = v^{i+1}(\mathcal{X}^h_{t_{i+1}})$, where u^{i+1} and v^{i+1} are the regressed functional forms of $\widehat{Y}^h_{t_{i+1}}$ and $\widehat{\rho}^h_{t_{i+1}}$, with the two copies independent conditionally on $\mathcal{X}^h_{t_i}$[20]:

$$\begin{aligned}
\mathbb{E}[\epsilon^2_{t_i}] = \mathbb{E}[&|\widehat{Y}^h_{t_i}|^2 - 2\widehat{Y}^h_{t_i}(\widehat{Y}^h_{t_{i+1}} + f(t_i, \mathcal{X}^h_{t_i}, \widehat{Y}^h_{t_{i+1}}, \widehat{\rho}^h_{t_i+1})\Delta t) + \\
&(\widehat{Y}^{h,1}_{t_{i+1}} + f(t_i, \mathcal{X}^h_{t_i}, \widehat{Y}^{h,1}_{t_{i+1}}, \widehat{\rho}^{h,2}_{t_i+1})\Delta t)(\widehat{Y}^{h,2}_{t_{i+1}} + f(t_i, \mathcal{X}^h_{t_i}, \widehat{Y}^{h,2}_{t_{i+1}}, \widehat{\rho}^{h,2}_{t_i+1})\Delta t)].
\end{aligned} \tag{23}$$

As detailed in Barrera, Crépey, Gobet, Nguyen, and Saadeddine (2022, Section 4.4), the $e_{t_i} = |\widetilde{\rho}^h_{t_i} - \widehat{\rho}^h_{t_i}|$ can also be estimated by twin simulation, without having to compute the $\widetilde{Y}^h_{t_i}$ and $\widetilde{\rho}^h_{t_i}$.

[19] a bit simplified for the sake of readibility here, compare with (29)-(30).

[20] i.e. one simulates two independent realizations of $\mathcal{X}^h_{t_{i+1}}$, given the same starting point $\mathcal{X}^h_{t_i}$, and then takes their images by the learned functionals u^{i+1} and v^{i+1}.

Proceeding in this way, we obtain an a posteriori Monte Carlo procedure to assess, at least locally in time, the spatial error of our ABSDE numerical schemes[21]. In the case where the error is not good enough, one can improve the stochastic gradient descent, in first attempt, and then act on the hypothesis spaces, e.g. , in the case of neural networks, try to train with more layers/units or better architectures.

Remark 3.3 *In the standard BSDE case where $f_k(t,x,y,\varrho) = f_k(t,x,y)$, i.e. when there is no dependence of the coefficient of the BSDE on ρ, we have by Λ_f Lipschitz continuity of $f_k(t,x,y)$ with respect to y:*

$$\mathbb{E}[|Y^h_{t_i} - \widetilde{Y}^h_{t_i}|] \leq (1+\Lambda_f \Delta t)\mathbb{E}[|Y^h_{t_{i+1}} - \widehat{Y}^h_{t_{i+1}}|]$$

and the triangular inequality yields

$$\mathbb{E}[|Y^h_{t_i} - \widehat{Y}^h_{t_i}|] \leq (1+\Lambda_f \Delta t)\mathbb{E}[|Y^h_{t_{i+1}} - \widehat{Y}^h_{t_{i+1}}|] + \mathbb{E}[|\widehat{Y}^h_{t_i} - \widetilde{Y}^h_{t_i}|]. \tag{24}$$

Hence, using the inequality $\mathbb{E}[\epsilon_{t_i}] \leq \sqrt{\mathbb{E}[\epsilon^2_{t_i}]}$:

$$\mathbb{E}[|Y^h_{t_i} - \widehat{Y}^h_{t_i}|] \leq \sum_{\imath=i}^{n-1} (1+\Lambda_f \Delta t)^{\imath - i} \sqrt{\mathbb{E}[\epsilon^2_{t_\imath}]}, \tag{25}$$

where each $\mathbb{E}[\epsilon^2_{t_\imath}]$ in (25) can be computed by twin Monte Carlo based on (23)[22], for two copies $\widehat{Y}^{h,1}_{t_{i+1}}$ and $\widehat{Y}^{h,2}_{t_{i+1}}$ of $\widehat{Y}^h_{t_{i+1}}$ independent conditionally on $\mathcal{X}^h_{t_i}$.

In the anticipated case, the analogous propagation of the local regression error terms $\epsilon_{t_\imath}$ and $e_{t_\imath}$ into global regression error controls for $\mathbb{E}[|Y^h_{t_i} - \widehat{Y}^h_{t_i}|]$ and $\mathbb{E}[|\rho^h_{t_i} - \widehat{\rho}^h_{t_i}|]$ is more involved.

4 XVA Application

We consider a bank dealing financial derivatives with multiple counterparties indexed by c, with default times $\tau^{(c)}$, where all portfolios are uncollateralized with zero recovery in the case of defaults (all assumed instantaneously settled). For notational simplicity we assume no contractual cash flows between the bank and client c at $\tau^{(c)}$. We denote by $T > 0$ the final maturity of the derivative portfolio of the bank and by $\mathrm{MtM}^{(c)}$ the aggregated mark-to-market process (counterparty-risk-free valuation) of the portfolio of the bank with counterparty c. The bank, with risky funding spread process $\gamma^{(b)}$, is required to maintain a minimum amount of capital at risk, at the level of an economic capital (EC) defined below as an expected shortfall of the bank trading loss over one year, at a confidence level $\alpha \in (\frac{1}{2}, 1)$. The bank is assumed perfectly hedged in terms of market risk, hence its trading loss reduces to the one of its CVA and FVA desks. For the sake of brevity in notation, we omit the discountings at the risk-free rate in the equations

[21] see e.g. Figures 11-12 in Section 4.1.

[22] ignoring the $\widehat{\rho}^h_{t_i+1}$ there.

(in other terms, we use the risk-free asset as a numéraire), while preserving them in the numerical codes. We assume a KVA risk premium, i.e. bank shareholders earn a hurdle rate $r > 0$ on their capital at risk. Finally we assume as in (Crépey, Sabbagh, and Song, 2020) that the bank can use its capital as a risk-free funding source.

As detailed in Crépey (2022); Albanese, Crépey, Hoskinson, and Saadeddine (2021), this yields the following CVA (credit valuation adjustment), FVA (funding valuation adjustment), EC (economic capital) and KVA (capital valuation adjustment) equations, where $J_t^{(c)} = \mathbb{1}_{\{t<\tau^{(c)}\}}$ (so $-dJ_t^{(c)} = \delta_{\tau^{(c)}}(dt)$, the Dirac measure at time $\tau^{(c)}$): For $t \le T$,

$$\begin{aligned}
\mathrm{CVA}_t &= \mathbb{E}_t\left[\sum_c \int_t^T (\mathrm{MtM}_s^{(c)})^+ \delta_{\tau^{(c)}}(ds)\right] \\
\mathrm{FVA}_t &= \mathbb{E}_t\left[\int_t^T \gamma_s^{(b)} \left(\sum_c J_s^{(c)} \mathrm{MtM}_s^{(c)} - \mathrm{CVA}_s - \mathrm{FVA}_s - \max(\mathrm{EC}_s, \mathrm{KVA}_s)\right)^+ ds\right] \\
\mathrm{KVA}_t &= \mathbb{E}_t\left[\int_t^T r e^{-r(s-t)} \max(\mathrm{EC}_s, \mathrm{KVA}_s) ds\right],
\end{aligned} \tag{26}$$

where, with $\bar{t} = (t+1) \wedge T$,

$$\mathrm{EC}_t = \mathbb{ES}_t[L_{\bar{t}} - L_t], \tag{27}$$

in which the loss process L satisfies, starting from $L_0 = 0$:

$$\begin{aligned}
dL_t = &\sum_c (\mathrm{MtM}_t^{(c)})^+ \delta_{\tau^{(c)}}(dt) + d\,\mathrm{CVA}_t + \\
&\gamma_s^{(b)} \left(\sum_c J_t^{(c)} \mathrm{MtM}_t^{(c)} - \mathrm{CVA}_t - \mathrm{FVA}_t - \max(\mathrm{EC}_t, \mathrm{KVA}_t)\right)^+ dt + d\,\mathrm{FVA}_t\,.
\end{aligned} \tag{28}$$

All the random variables $J_t^{(c)}$ and $\mathrm{MtM}_t^{(c)}$, as well the pre-default intensity $\gamma_t^{(b)}$ of the bank, are assumed to be $\sigma(\mathcal{X}_t)$-measurable. Hence so is CVA_t as per the first line in (26). The FVA and the KVA equations in (26) can then be written in the form (7), for $l = 2$ and

$$\begin{aligned}
&Y_t = \begin{pmatrix} \mathrm{FVA}_t, \\ e^{-rt}\,\mathrm{KVA}_t \end{pmatrix},\ \phi = (0,0)^\top, \\
&f_k(t, x, (y_1, y_2)^\top, \varrho) = \\
&\begin{pmatrix} \gamma_k^{(b)}(t,x)\big(\sum_c J_k^{(c)}(t,x)\mathrm{MtM}_k^{(c)}(t,x) - \mathrm{CVA}_k(t,x) - y_1 - \max(\varrho, e^{rt} y_2)\big)^+ \\ r \max(e^{-rt}\varrho, y_2) \end{pmatrix},
\end{aligned} \tag{29}$$

where we denote $Z_k(t,x) = \mathbb{E}[Z_t | X_t = x, J_t = k]$, for any process Z,

$$\Phi_{\bar{\cdot}}(M) = \int_{\cdot}^{\bar{\cdot}} \left(\sum_c (\mathrm{MtM}_t^{(c)})^+ \delta_{\tau^{(c)}}(dt) + d\,\mathrm{CVA}_t + dM_t^1\right), \text{ for any } M = (M^1, M^2)^\top \in \mathcal{S}_2^2.$$

Note that, for (Y, M) solving the ABSDE (7) corresponding to the specification (29), we have

$$dM_t^1 = d\,\mathrm{FVA}_t + \gamma_t^{(b)} \left(\sum_c J_t^{(c)}\, \mathrm{MtM}_t^{(c)} - \mathrm{CVA}_t - \mathrm{FVA}_t - \max(\mathrm{EC}_t, \mathrm{KVA}_t) \right)^+ dt,$$

hence in view also of (28):

$$\Phi_{\bar{t}}(M) = L_{\bar{t}} - L_t, \tag{30}$$

which completes the connection with (26)-(27).

Remark 4.1 *In this XVA case, due to the special form* (30) *of* $\Phi_{\bar{t}}(M)$*, one does not need to maintain full* processes $\widehat{M}_{t_i}^h$ *and* $\widehat{M}_{t_i}^{j,h}, i = 0 \cdots n$*, to deal with the generic equations* (7) *and* (11) *à la* (18) */ Algorithm 1 and* (19) */ Algorithm 2. As detailed in Section A, it suffices to update iteratively in (decreasing) time* t_i *the random variables* $L_{\bar{t}_i} - L_{t_i}$ *or (in the Picard case)* $L_{\bar{t}_i}^{(j)} - L_{t_i}^{(j)}$ *the way detailed in* (31) *and* (32)*.*

Note that the corresponding functional Φ satisfies in view of its formulation in (29)[23]:

$$|\Phi(t; \mathbf{x}, \mathbf{m}) - \Phi(t; \mathbf{x}, \mathbf{m}')| \leq |\mathbf{m}_{\bar{t}} - \mathbf{m}'_{\bar{t}}|,$$

so that Assumption 2.2(iii) holds with $\Lambda_\Phi = 1$. Assumption 2.2(i) is readily checked. Assumption 2.2(ii) holds provided the process $\left(\gamma^{(b)}(t, \mathcal{X}_t)(\sum_c J(t, \mathcal{X}_t)\mathrm{MtM}^{(c)}(t, \mathcal{X}_t)\right)$ is in $\mathcal{H}^2$, e.g. for $\gamma^{(b)}$ bounded and $\left(\mathrm{MtM}^{(c)}(t, \mathcal{X}_t)\right)$ in $\mathcal{H}^2$ for each client c.

In view of the first line in (26), the CVA process can be estimated by nonparametric least squares regression in space, at each grid pricing time t_i, based on Monte Carlo paths of the forward process $\mathcal{X}$[24]. The exercise is made delicate, however, by the hybrid nature of $\mathcal{X}$, which includes both diffusive (market risk) and discrete (default risk) components. But this difficulty can be solved by adopting the hierarchical simulation scheme of Abbas-Turki, Crépey, and Saadeddine (2022), whereby many default trajectories are simulated conditional on each simulated trajectory of the market. In view of this, the CVA process is treated hereafter as a given (already estimated) process.

The FVA, EC, and KVA equations are challenging due to their coupling via the loss process L. However, this coupling can be overcome by a combination of time discretization schemes and (or not) Picard iterations as presented in the more general context of Section 3. The specification of both schemes to the XVA case, as well as a direct variant of the implicit scheme[25] that is also available in this case, are detailed in Section A.1. The embedded regressions and quantile regressions are implemented using a neural network of one hidden layer with 38 neurons[26], the way detailed in Section 3.2.

[23] cf. (6).

[24] or its time-discretized version $\mathcal{X}^h$, cf. Section A.1.

[25] without need for Picard iterations, but at the cost of a shift of time step in EC computations.

[26] for experimentation with the network architecture see Albanese, Crépey, Hoskinson, and Saadeddine (2021, Section 5.1, Figure 2).

Remark 4.2 *In practice, for variance reduction purposes, we use a default intensities based reformulation of the CVA, instead of the definition based on default indicators in* (26)*. The default indicators based CVA was used in Abbas-Turki, Crépey, and Saadeddine (2022) mainly for benchmarking reasons. Still we do use the hierarchical simulation scheme of Abbas-Turki, Crépey, and Saadeddine (2022), simulating many default paths given each realization of the diffusion processes, in order to help with learnings where we do not have the convenience of using default intensities (given the presence of default terms in the loss* (28)*, which occur nonlinearly in EC computations, and of default terms under the* $(\cdot)^+$ *in the FVA computations).*

4.1 Numerical Results

For our numerical experiments, we assume 10 economies, each represented by a short-rate process with Vasicek dynamics, 9 cross-currency rate processes with log-normal dynamics, 8 counterparties each with a stochastic default intensity process following CIR dynamics, adding up to 28 diffusive risk factors (when accounting for the spread of the bank which is also assumed to be driven by a CIR process) and 8 default indicator processes. In order to have analytic mark-to-markets, we assume that the portfolio of the bank is comprised of 100 interest rate swaps, assumed to be at-par at time 0. The characteristics of the swaps (notional, maturity, counterparty and currency) are drawn randomly. In particular, the maturities of the swaps range between 0.9375 and $T = 10$ years. For all modeling and implementation details, as well as numerical values of all the parameters, the reader is referred to https://github.com/BouazzaSE/NeuralXVA.

For the purpose of a time discretization analysis, we consider multiple time discretizations $(h^{(\kappa)})_\kappa$ such that

$$h^{(\kappa)} = \{t_i^{(\kappa)} := i\frac{T}{2^\kappa}, i = 0, \dots, 2^\kappa\}.$$

In Figures 1, 2 and 5, we tested for $\kappa \in \{5, 6, 7, 8\}$ (τ in the figures then corresponds to $\frac{T}{2^8}$). Figures 1-2 and Table 1 show the convergence of the Picard iterations, using Algorithm 2, of the FVA process toward a solution visually very close, already for $j = 4$, to the one of Figure 5 provided by the explicit scheme of Algorithm 1. However the Picard scheme comes at the cost of $j =$ 3-4 times longer computations (for reference, the explicit scheme takes 7mins48secs using the finest time grid, i.e. $\kappa = 8$, on an NVidia A100 GPU).

We also attempted to improve the Picard scheme using less epochs and reusing weights across Picard iterations as discussed in Remark 3.2, but the resulting iterates exhibit a slight instability when using finer time step, i.e. for $\kappa = 8$: see the bottom graphs in Figures 3-4 and Table 2. A possible explanation is that Algorithms 1 and 2 accumulate a large number of stochastic gradient descents via the weights reused across pricing times, with integrands involving the same stochastic processes just observed at different but close time steps. With weights reused across Picard iterations in the modified Picard scheme of Remark 3.2, instead, the integrands are not guaranteed to be close. The results obtained in Figures 1-2 and Table 1 using the standard Picard scheme

	$j=1$	$j=2$	$j=3$	$j=4$	Explicit
$h=\frac{T}{2^5}$	463.279938	433.832031	434.391296	433.753998	434.65167
$h=\frac{T}{2^6}$	461.329926	433.141876	434.036011	433.835052	433.60974
$h=\frac{T}{2^7}$	461.031097	432.506531	433.631531	431.789215	433.18683
$h=\frac{T}{2^8}$	460.326050	433.123596	431.992859	432.098328	434.29538

Table 1: FVA_0 under the Picard iteration scheme with reuse of weigthts across time steps (i.e. Algorithm 2) vs. the explicit scheme.

	$j=1$	$j=2$	$j=3$	$j=4$	Explicit
$h=\frac{T}{2^8}$	498.9785	416.31674	464.44170	386.07004	434.295380
$h=\frac{T}{2^7}$	481.9461	443.68940	440.41504	449.64874	433.186829
$h=\frac{T}{2^6}$	449.9881	435.35910	429.49142	424.67087	433.609741
$h=\frac{T}{2^5}$	462.9291	438.05840	435.67453	437.58957	434.651672

Table 2: FVA_0 under the modified Picard vs the explicit scheme.

with reuse of the weights across time steps, i.e. Algorithm 2, support this conjecture. These results mutually validate the explicit and the standard Picard scheme against each other. However, the standard Picard scheme is not competitive with respect to the explicit scheme in terms of computation times (cf. Remark 3.2).

To sum up, the explicit scheme overperforms the Picard iterations, whether that a common computation time is allocated to both schemes and the explicit scheme is more accurate, or that both schemes converge but this is then at the cost of several times longer computations in the case of the Picard iterations. Figures 6, 7, 8 and 9 show plots of profiles for respectively the CVA, FVA, KVA and EC when using the explicit scheme. Figure 10 illustrates the convergence in time of the scheme. The solid purple curves in Figures 11 and 12 exhibit the local regression errors $\sqrt{\mathbb{E}[\epsilon_{t_i}^2]}$ of the scheme[27] for the FVA and the KVA. The dashed grey curves represent the corresponding L^2 training losses. The comparison between the grey and purple curves shows the benefit of the a posteriori Monte Carlo local regression error estimates (23) with respect to the L^2 training losses that would be used as a naive (but overconservative) error estimate.

5 Conclusion

The recent and fastly growing machine learning literature on the solution of high-dimensional nonlinear BSDEs/PDEs contains, at least, two branches. The first one, in the line of E, Han, and Jentzen (2017), consists in learning together the time-0 value of the solution and the gradient-process of the latter through a single learning task.

[27] cf. (22).

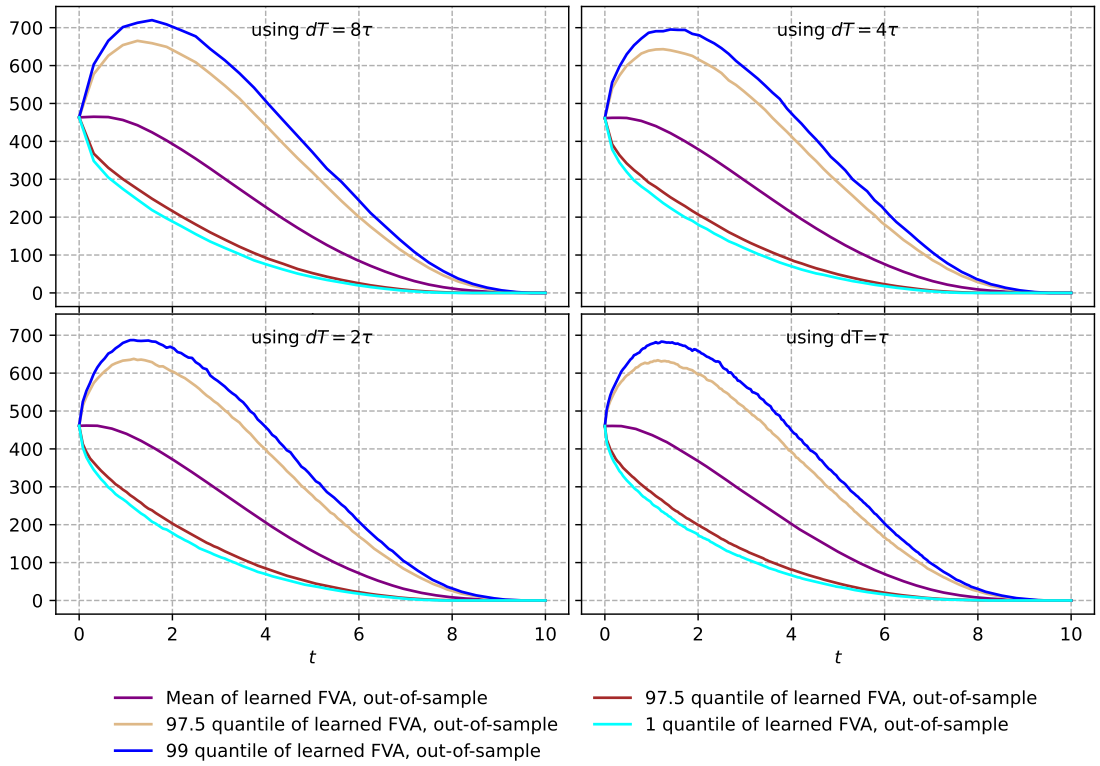

Figure 1: FVA profiles obtained after $j = 1$ Picard iteration for the implicit scheme.

Examples in the XVA space include Henry-Labordère (2019) or Gnoatto, Reisinger, and Picarelli (2021). The former reference provides insightful PDE views on the CVA and MVA, while it is tempting to adopt an approach, as in the second reference, where the XVAs and their sensitivities are obtained simultaneously. However, the stylized equations of Henry-Labordère (2019) are quite far from actual XVA equations. The XVA equations of Gnoatto, Reisinger, and Picarelli (2021) are more realistic, but they are still restricted to computations at the level of one client of the bank and they are handled by reduction of filtration in the line of Crépey and Song (2015), so that the default times ultimately disappear from the equations. Such an approach does not leverage to several clients, whose default times enter the XVA equations of the bank in a nonlinear fashion (and therefore have to be simulated as we do). The second stream of literature, that includes Huré, Pham, and Warin (2020) or the present paper, learns the solution time step after time step (in backward time), much like classical dynamic programming algorithms, except that machine learning techniques are used for solving the local equations which then arise at each successive decreasing pricing time step.

In the case of our ABSDE XVA equations, on the one hand, the global approach would not be viable on realistic problems stated at the portfolio level, because of the huge RAM memory demand of the corresponding global training task. On the other hand, the local approach benefits from a particular synergy between the successive local training

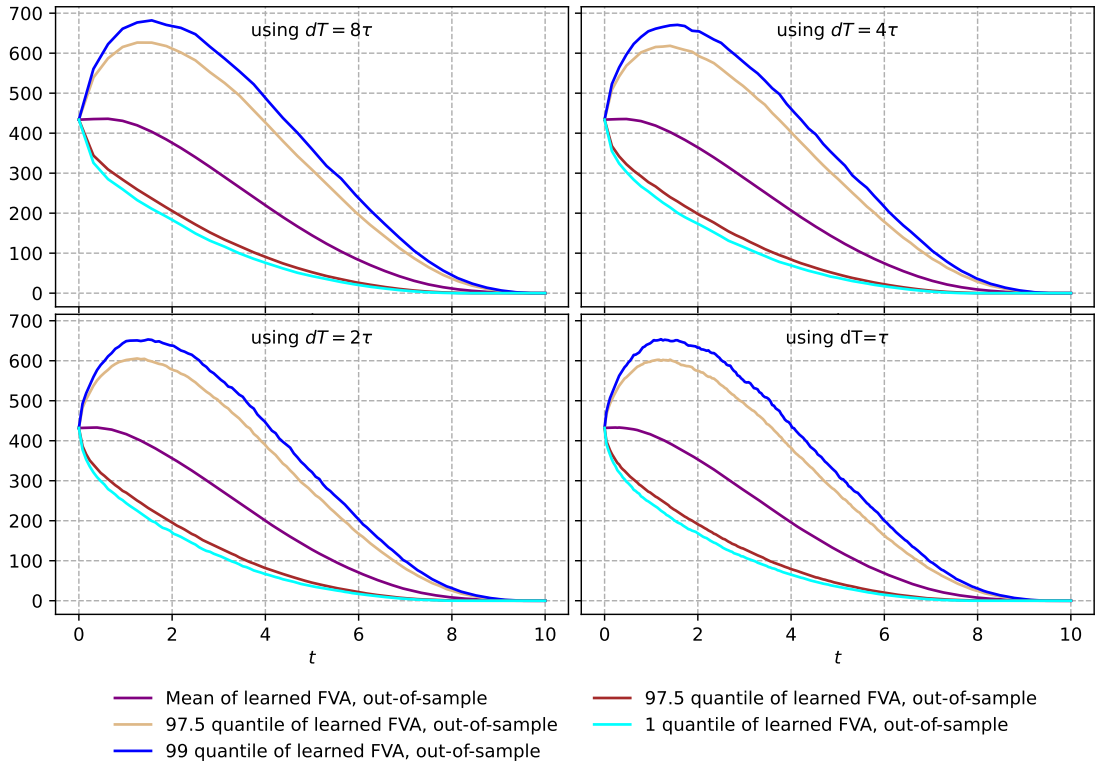

Figure 2: FVA profiles obtained after $j = 4$ Picard iteration for the implicit scheme.

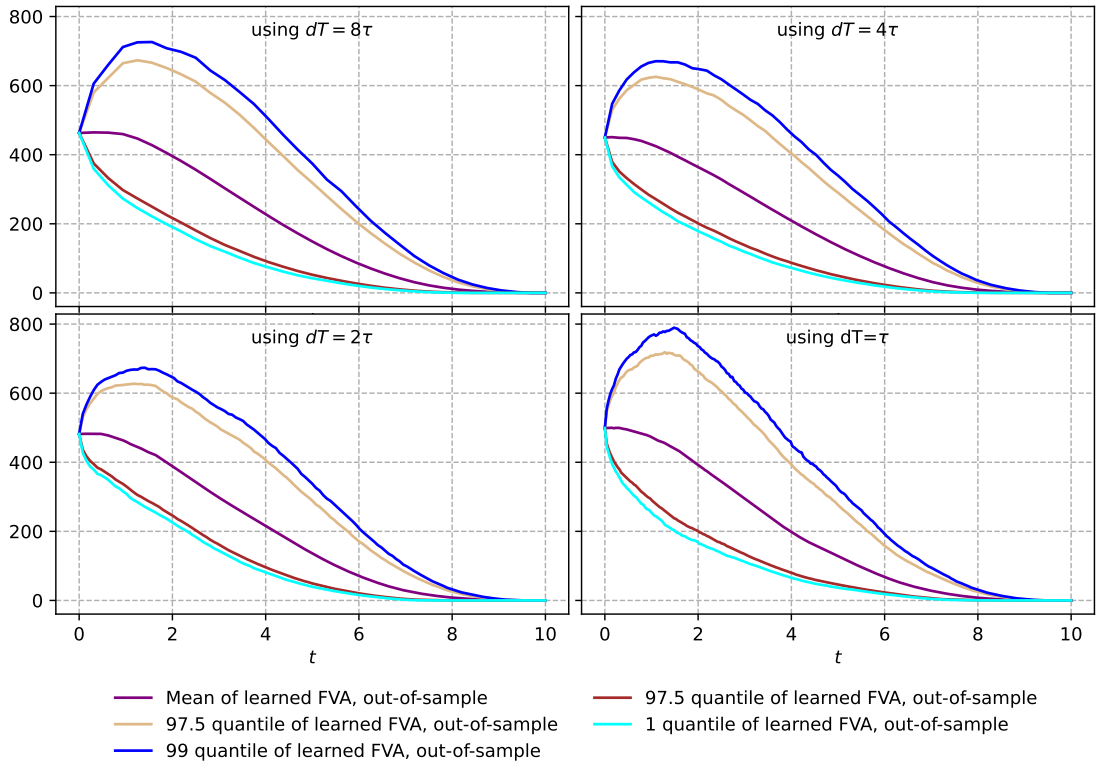

Figure 3: FVA profiles obtained after $j = 1$ Picard iteration for the implicit scheme, reusing the weights of the previous Picard iteration at each learning.

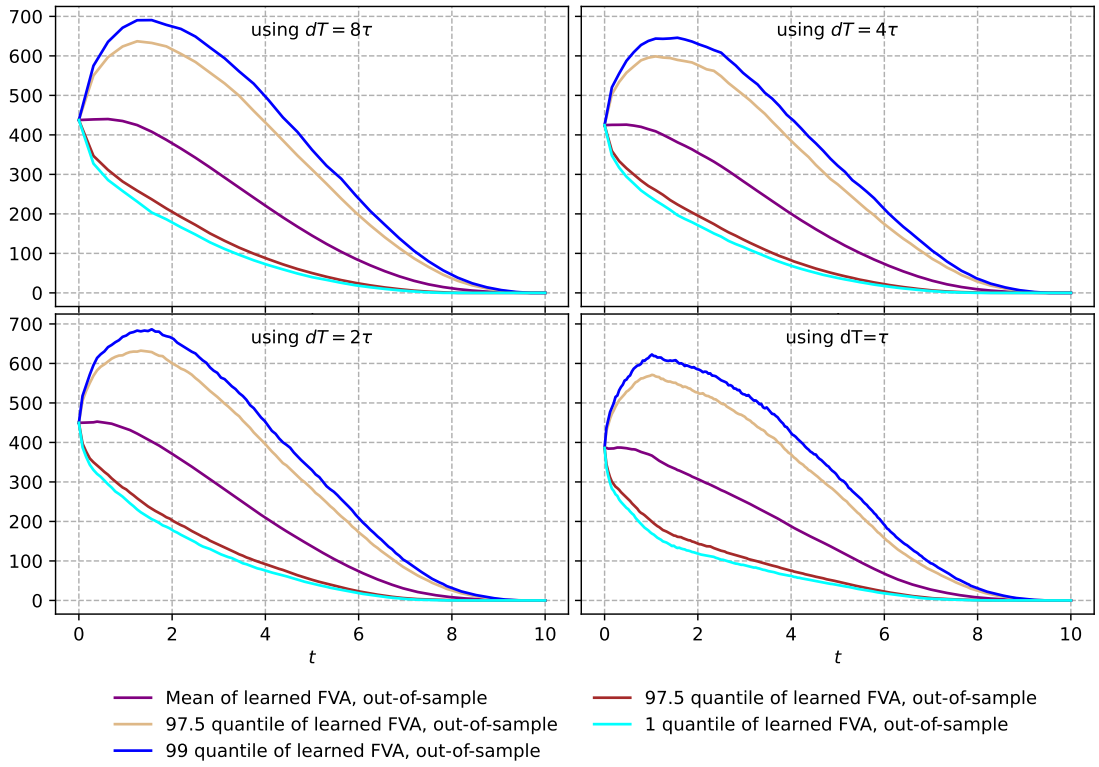

Figure 4: FVA profiles obtained after $j = 4$ Picard iterations for the implicit scheme, reusing the weights of the previous Picard iteration at each learning.

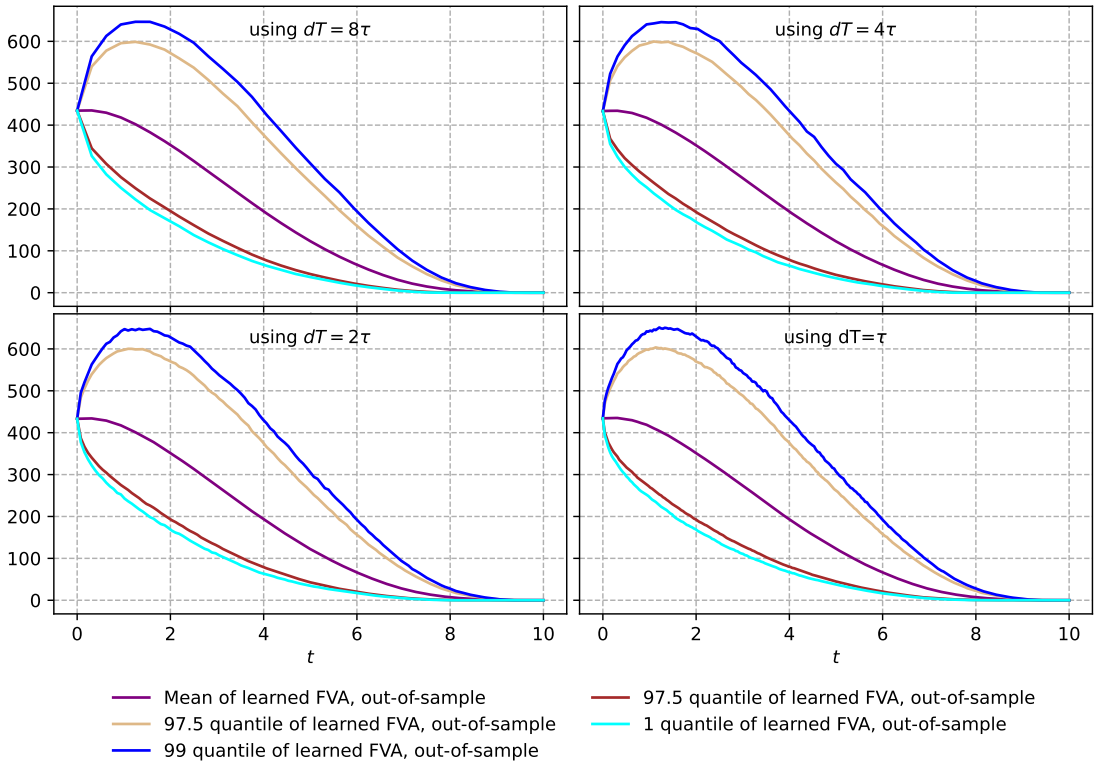

Figure 5: FVA profiles using an explicit scheme.

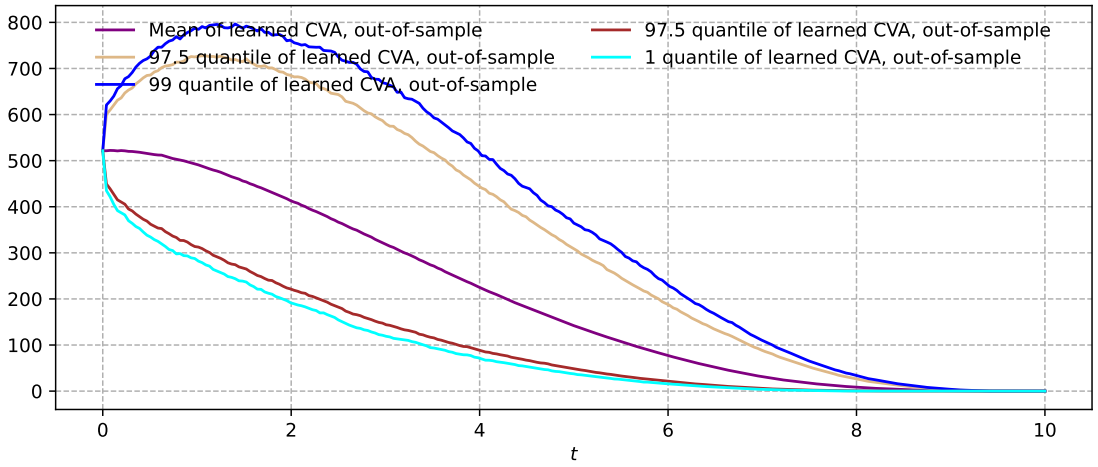

Figure 6: CVA profiles using an explicit scheme and a fine time discretization ($\kappa = 10$)

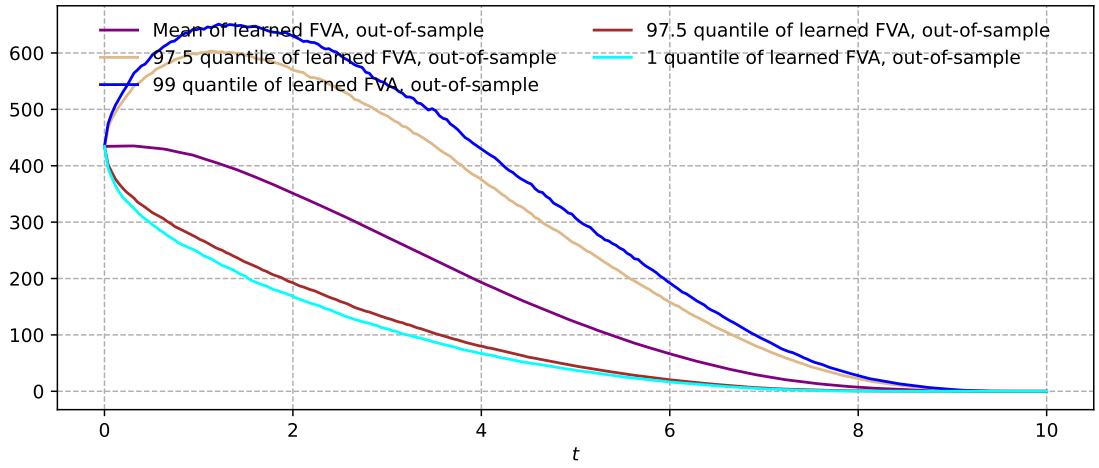

Figure 7: FVA profiles using an explicit scheme and a fine time discretization ($\kappa = 10$)

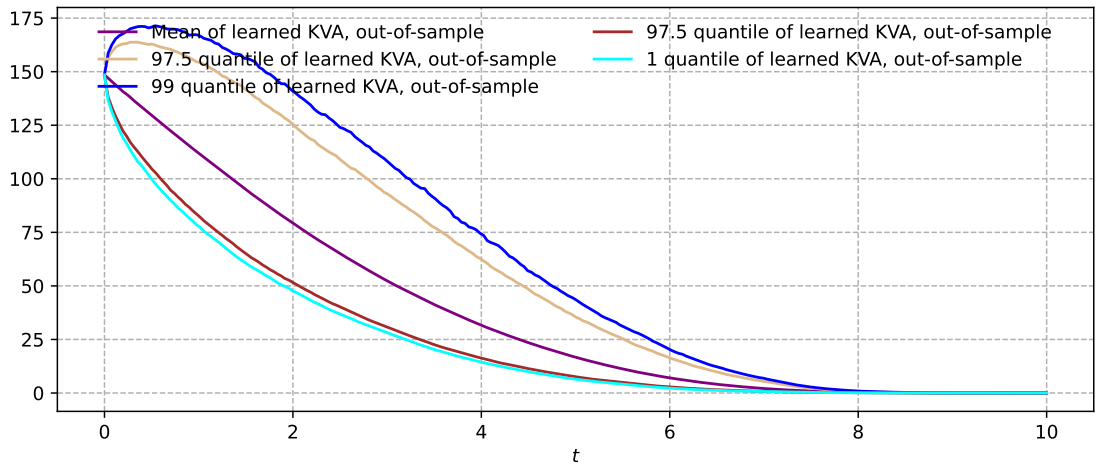

Figure 8: KVA profiles using an explicit scheme and a fine time discretization ($\kappa = 10$)

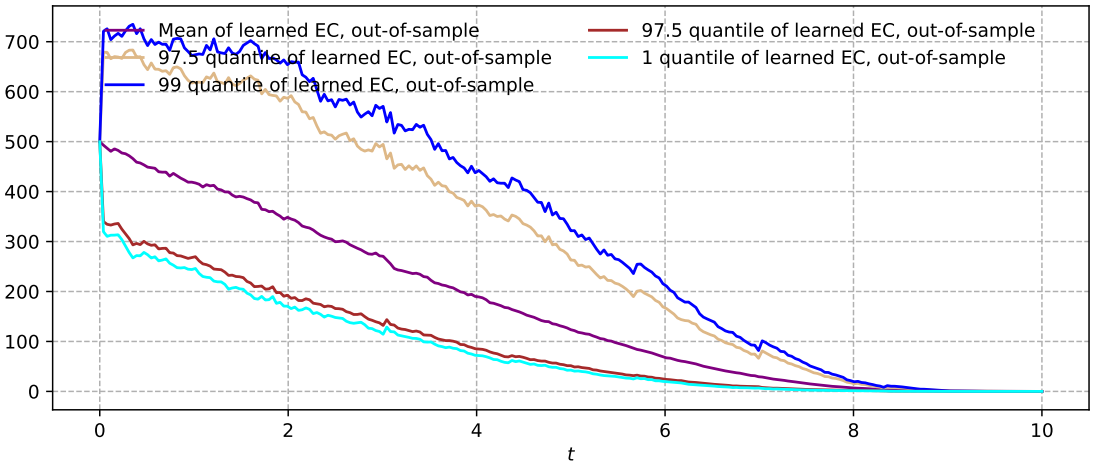

Figure 9: EC profiles using an explicit scheme and a fine time discretization ($\kappa = 10$)

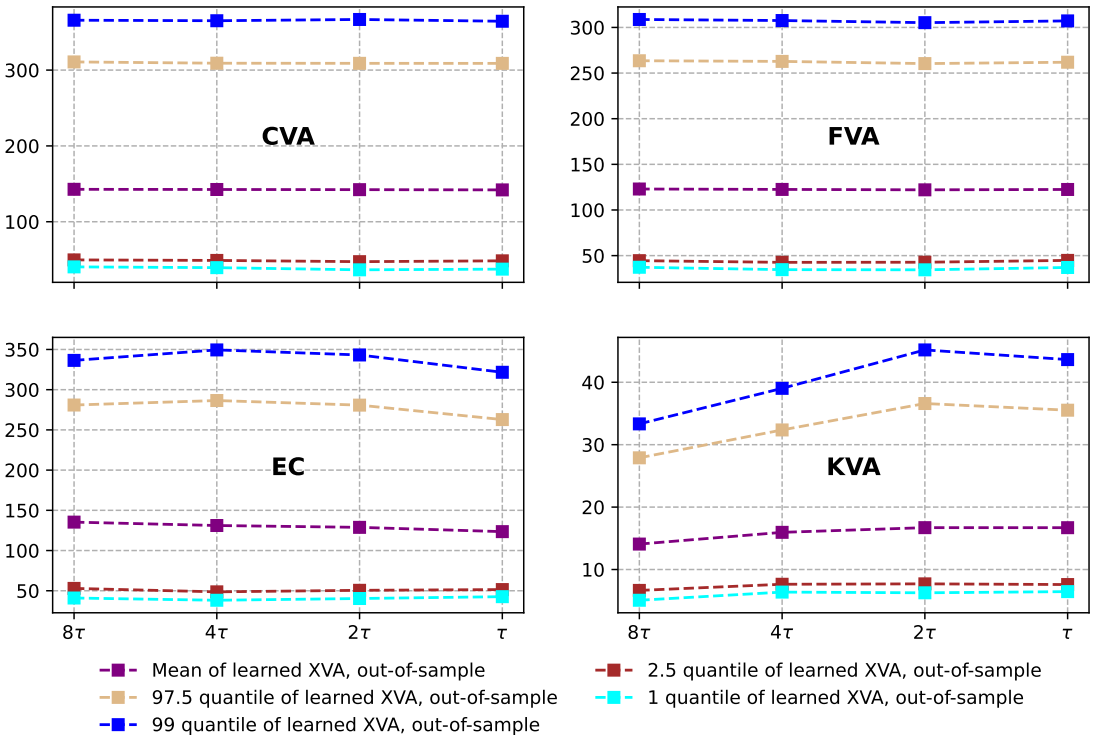

Figure 10: Mean and quantiles of CVA, FVA, KVA and EC learned by the explicit scheme at $t = \frac{T}{2}$ for different sizes of the time step.

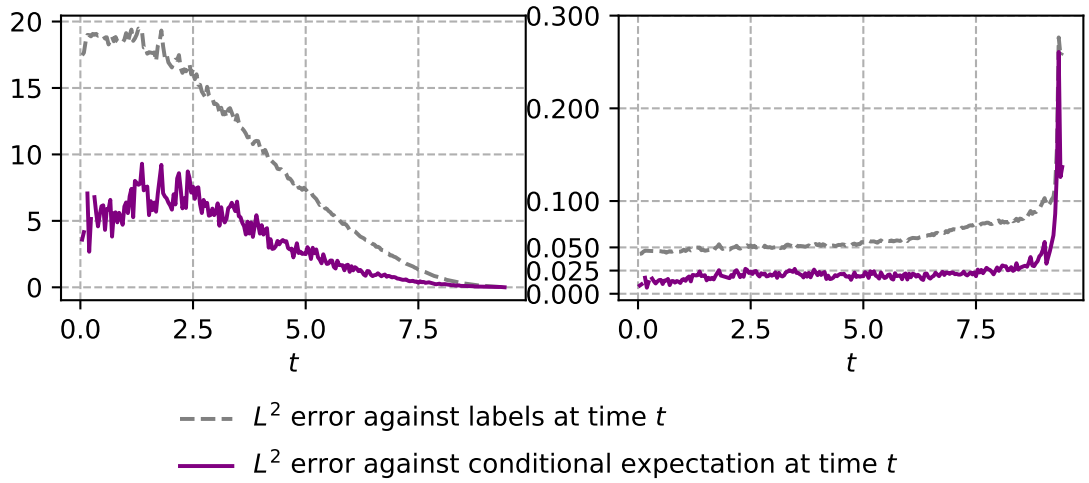

Figure 11: Local regression errors $\sqrt{\mathbb{E}[(\epsilon_{t_i}^{fva})^2]}$ (solid purple) vs. L^2 training losses (dashed grey). Left panel: raw errors. Right panel: errors normalized at each time step t_i by the L^2 norm of $\widehat{\mathrm{FVA}}_{t_i}^h$.

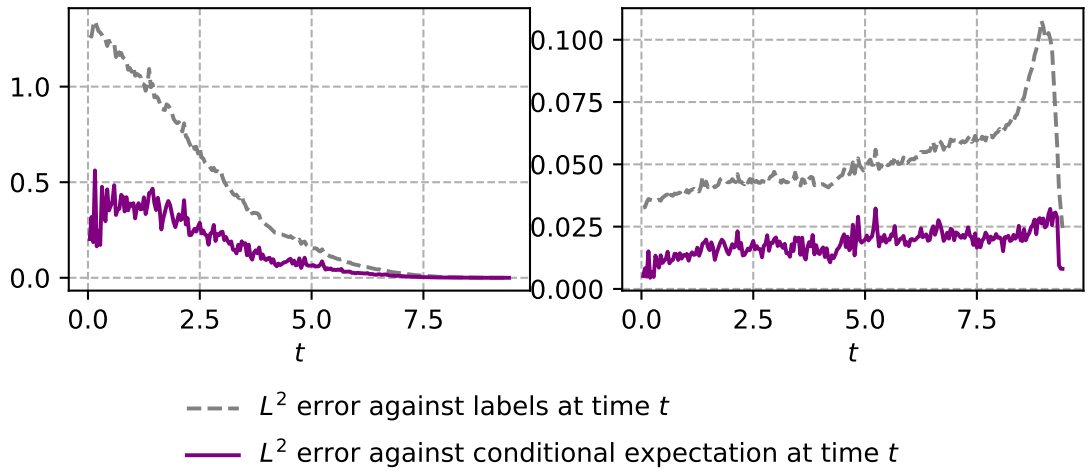

Figure 12: Local regression errors $\sqrt{\mathbb{E}[(\epsilon_{t_i}^{kva})^2]}$ (solid purple) vs. L^2 training losses (dashed grey). Left panel: raw errors. Right panel: errors normalized at each time step t_i by the L^2 norm of $\widehat{\mathrm{KVA}}_{t_i}^h$.

tasks involved. In fact, as all XVA equations are endowed with zero terminal conditions, the variance of the cash flows ("regressands"), which enter the successive learning tasks as input data at the decreasing pricing time steps, increases progressively (pricing time step after pricing time step), whereas the variance of the features ("regressors"), i.e. the risk factors, decreases. As a result, the difficulty of the training tasks gradually increases throughout the course of the algorithm[28]. But the next training task also greatly (and increasingly) benefits from all previous ones, via the use of the weights trained at a time step as initialization for the weights at the next time step. This is probably one of the reasons behind the robustness of our approach[29], when a global approach would fail on unsolvable memory occupation issues.

Regarding the comparison between the direct explicit scheme and the implicit scheme solved by Picard iteration, the explicit scheme emerges from the present study as the preferred alternative. The main contribution of the paper is therefore the explicit scheme of Algorithm 1. However, for the initialization reflected in (19) / Algorithm 2 and (32), the output of the first Picard iteration (3 or 4 are typically enough to ensure convergence in practice) is interesting from a financial interpretation viewpoint in an XVA setup, as this first iteration corresponds to the XVA numbers ignoring the possibility to use capital at risk for variation margin funding purposes (Albanese et al., 2017, Section 5.2).

From an algorithmic viewpoint, work in progress aims at demonstrating how the regression-based XVA simulation framework of this paper can be leveraged to also encompass XVA sensitivities or hedging ratios more generally. Note that AAD sensitivities computational techniques à la (Baydin, Pearlmutter, Radul, and Siskind, 2018; Savine, 2018) are not a viable alternative in a path-wise XVA setup, where the path-wise XVA metrics are already the output of optimization training procedures: AAD sensitivity techniques can only be available in more rudimentory XVA setups. From a mathematical viewpoint, the establishment of a Feynman-Kac representation for the solution of the limiting ABSDE (7), as well as the study of the time-consistency of both schemes, or of the propagation of the local into global spatial regression errors[30], are challenging open issues.

A XVA Numerical Schemes

In this section we detail a version of the generic algorithms of Section 3.2 specialized[31] to the XVA equations (26)–(28) (cf. (29)). We also provide a direct variant of the implicit

[28] In order to see this in a simplified setup, consider linear regression instead of neural networks. When close to $t = 0$, the variances of the regressors tend to 0, which leads to ill-conditioned covariance matrices, while the variance of the regressands increases, which makes the regression even more unstable.

[29] provided the hierarchical simulation technique of Abbas-Turki, Crépey, and Saadeddine (2022) and the best practice risk measure estimates of Barrera, Crépey, Gobet, Nguyen, and Saadeddine (2022) are used, as done in our numerics.

[30] cf. the end of Remark 3.3.

[31] cf. in particular Remark 4.1.

scheme[32] that is also available in this case. We assume a uniform time step $\Delta t = h$ to alleviate the notation. By least squares (resp. quantile) regressions below, we always mean *neural network* least squares (resp. quantile) regressions conducted against all risk factors (market risk factors and client default indicators) at each decreasing time t_i, the way detailed in Section 3.2.

A.1 Explicit scheme

Here we use the following time-discretization, skipping all the $\widehat{\cdot}$ and $\cdot^h$ to alleviate the notation and writing $\bar{t}_i = t_{i+1/h} \wedge t_n$: $\mathrm{CVA}_{t_n} = \mathrm{FVA}_{t_n} = \mathrm{KVA}_{t_n} = 0$ and, for $i = n-1 \cdots 0$,

$$\mathrm{CVA}_{t_i} = \mathbb{E}_{t_i}\left[\sum_c \sum_{i \le \imath \le n-1} (\mathrm{MtM}^{(c)}_{t_{\imath+1}})^+ \mathbb{1}_{\{t_\imath < \tau^{(c)} \le t_{\imath+1}\}}\right]$$

$$\mathrm{FVA}_{t_i} = \mathbb{E}_{t_i}\left[\mathrm{FVA}_{t_{i+1}} + \right.$$

$$\left. h\gamma^{(b)}_{t_i}\left(\sum_c \mathrm{MtM}^{(c)}_{t_i} \mathbb{1}_{\{\tau^{(c)} > t_i\}} - \mathrm{CVA}_{t_i} - \mathrm{FVA}_{t_{i+1}} - \max(\mathrm{EC}_{t_{i+1}}, \mathrm{KVA}_{t_{i+1}})\right)^+\right]$$

$$\mathrm{KVA}_{t_i} = \exp(-rh)\mathbb{E}_{t_i}[\mathrm{KVA}_{t_{i+1}} + rh \max(\mathrm{EC}_{t_{i+1}}, \mathrm{KVA}_{t_{i+1}})]$$

$$\mathrm{EC}_{t_i} = \mathbb{ES}_{t_i}[L_{\bar{t}_i} - L_{t_i}] \text{ , where}$$

$$\begin{aligned} L_{t_{i+1}} - L_{t_i} = {} & \mathrm{CVA}_{t_{i+1}} - \mathrm{CVA}_{t_i} + (\mathrm{MtM}^{(c)}_{t_i})^+ \mathbb{1}_{\{t_i < \tau^{(c)} \le t_{i+1}\}} + \mathrm{FVA}_{t_{i+1}} - \mathrm{FVA}_{t_i} + \\ & h\gamma^{(b)}_{t_i}\left(\sum_c \mathrm{MtM}^{(c)}_{t_i} \mathbb{1}_{\{\tau^{(c)} > t_i\}} - \mathrm{CVA}_{t_i} - \mathrm{FVA}_{t_{i+1}} - \max(\mathrm{EC}_{t_{i+1}}, \mathrm{KVA}_{t_{i+1}})\right)^+ . \end{aligned} \tag{31}$$

The explicit scheme naturally lifts the coupling visible in (26)–(28) between EC, KVA, and FVA. Specifically, assuming that all the $\mathrm{XVA}_{t_{i+1}}$ and $\mathrm{EC}_{t_{i+1}}$ have already been learned, we compute:

1 CVA_{t_i}, by a least-squares regression of $\sum_c \sum_{\imath \le i \le n-1} (\mathrm{MtM}^{(c)}_{t_i})^+ \mathbb{1}_{\{t_\imath < \tau^{(c)} \le t_{\imath+1}\}}$[33] against all risk factors at time t_i;

2 FVA_{t_i}, through a least-squares regression of

$$\mathrm{FVA}_{t_{i+1}} + h\gamma^{(b)}_{t_i}\left(\sum_c \mathrm{MtM}^{(c)}_{t_i} \mathbb{1}_{\{\tau^{(c)} > t_i\}} - \mathrm{CVA}_{t_i} - \mathrm{FVA}_{t_{i+1}} - \max(\mathrm{EC}_{t_{i+1}}, \mathrm{KVA}_{t_{i+1}})\right)^+$$

against all risk factors at time t_i;

[32] without need for Picard iterations, but at the cost of a shift of one time step in EC.

[33] or equivalent variance-reduced cash flows formulated in terms of the default *intensities* as explained after (29).

3 KVA_{t_i}, through a least-squares regression of

$$\exp(-rh)(\mathrm{KVA}_{t_{i+1}} + rh \max(\mathrm{EC}_{t_{i+1}}, \mathrm{KVA}_{t_{i+1}}))$$

against all risk factors at time t_i;

4 EC_{t_i}, through quantile regression of $L_{\bar{t}_i} - L_{t_i}$ followed by a least-squares regression to deduce the expected shortfall as detailed in Section 3.2, both regressions involving all the risk factors at time t_i.

A.2 Implicit schemes

We define and compute the CVA as in the explicit scheme. For the rest of the XVAs, we introduce Picard iterations, starting from $\mathrm{EC}^0 \equiv 0$ followed by, for increasing $j \geq 1$: $\mathrm{FVA}^j_{t_n} = \mathrm{KVA}^j_{t_n} = 0$ and, for $i = n-1 \cdots 0$,

$$\begin{aligned}
\mathrm{FVA}^j_{t_i} &= \mathbb{E}_{t_i}\Big[\mathrm{FVA}^j_{t_{i+1}} + \\
&\qquad h\gamma^{(b)}_{t_i}\left(\sum_c \mathrm{MtM}^{(c)}_{t_i} \mathbb{1}_{\{\tau^{(c)}>t_i\}} - \mathrm{CVA}_{t_i} - \widetilde{\mathrm{FVA}}^{(j-1)}_{t_i} - \max(\mathrm{EC}^{j-1}_{t_i}, \widetilde{\mathrm{KVA}}^{j-1}_{t_i})\right)^+\Big] \\
\mathrm{KVA}^j_{t_i} &= \exp(-rh)\mathbb{E}_{t_i}[\mathrm{KVA}^j_{t_{i+1}} + rh \max(\mathrm{EC}^{j-1}_{t_i}, \widetilde{\mathrm{KVA}}^{(j-1)}_{t_i})] \\
\mathrm{EC}^j_{t_i} &= \mathbb{ES}_{t_i}[L^j_{\bar{t}_i} - L^j_{t_i}] \text{ , where} \qquad (32) \\
L^j_{t_{i+1}} - L^j_{t_i} &= \mathrm{CVA}_{t_{i+1}} - \mathrm{CVA}_{t_i} + (\mathrm{MtM}^{(c)}_{t_i})^+ \mathbb{1}_{\{t_i<\tau^{(c)}\leq t_{i+1}\}} + \mathrm{FVA}^j_{t_{i+1}} - \mathrm{FVA}^j_{t_i} + \\
&\qquad h\gamma^{(b)}_{t_i}\left(\sum_c \mathrm{MtM}^{(c)}_{t_i} \mathbb{1}_{\{\tau^{(c)}>t_i\}} - \mathrm{CVA}_{t_i} - \widetilde{\mathrm{FVA}}^{j-1}_{t_i} - \max(\mathrm{EC}^{j-1}_{t_i}, \widetilde{\mathrm{KVA}}^{j-1}_{t_i})\right)^+,
\end{aligned}$$

where we set[34]

$$\widetilde{\mathrm{FVA}}^{j-1}_{t_i} = \mathbb{1}_{j=1}\,\mathrm{FVA}^j_{t_{i+1}} + \mathbb{1}_{j\geq 2}\,\mathrm{FVA}^{j-1}_{t_i},\ \widetilde{\mathrm{KVA}}^{j-1}_{t_i} = \mathbb{1}_{j=1}\,\mathrm{KVA}^j_{t_{i+1}} + \mathbb{1}_{j\geq 2}\,\mathrm{KVA}^{j-1}_{t_i}. \qquad (33)$$

In this scheme the coupling between EC, KVA, and FVA is removed by the Picard iterations in j. Specifically, assuming that all the $\mathrm{XVA}^{j-1}_{t_i}$ (for $j \geq 2$), $\mathrm{XVA}^j_{t_{i+1}}$, and $\mathrm{EC}^{j-1}_{t_i}$ have already been learned, we compute:

1 $\mathrm{FVA}^j_{t_i}$, by least-squares regression of

$$\begin{aligned}
&\mathrm{FVA}^j_{t_{i+1}} + \\
&h\gamma^{(b)}_{t_i}\left(\sum_c \mathrm{MtM}^{(c)}_{t_i} \mathbb{1}_{\{\tau^{(c)}>t_i\}} - \mathrm{CVA}_{t_i} - \widetilde{\mathrm{FVA}}^{j-1}_{t_i} - \max(\mathrm{EC}^{j-1}_{t_i}, \mathrm{KVA}^{j-1}_{t_i})\right)^+
\end{aligned}$$

against the risk factors at time t_i;

[34] cf. (15) and (20).

2 $\mathrm{KVA}^j_{t_i}$, through a least-squares regression of

$$\exp(-rh)(\mathrm{KVA}^j_{t_{i+1}} + rh \max(\mathrm{EC}^{j-1}_{t_i}, \widetilde{\mathrm{KVA}}^{j-1}_{t_i}))$$

against all risk factors at time t_i;

3 $\mathrm{EC}^j_{t_i}$, through quantile regression of $L^j_{\bar{t}_i} - L^j_{t_i}$ followed by a least-squares regression to deduce the expected shortfall as detailed in Section 3.2, both regressions involving all the risk factors at time t_i.

Remark A.1 *In this semilinear case with nonlinearities of the coefficients only of the form* $\cdot^+$*, one can also introduce the following hybrid scheme, which is implicit in* FVA *and* KVA*, while not requiring Picard iterations:* $\mathrm{EC}_{t_n} = \mathrm{FVA}_{t_n} = \mathrm{KVA}_{t_n} = 0$ *and, for* $i = n-1 \cdots 0$,

$$\begin{aligned}
\mathrm{KVA}_{t_i} &= \exp(-rh)(\mathbb{E}_{t_i}[\mathrm{KVA}_{t_{i+1}}] + rh \max(\mathrm{EC}_{t_{i+1}}, \mathrm{KVA}_{t_i})) \\
\mathrm{FVA}_{t_i} &= \mathbb{E}_{t_i}\Big[\mathrm{FVA}_{t_{i+1}} + \\
&\qquad h\gamma^{(b)}_{t_i}\left(\sum_c \mathrm{MtM}^{(c)}_{t_i} \mathbb{1}_{\{\tau^{(c)} > t_i\}} - \mathrm{CVA}_{t_i} - \mathrm{FVA}_{t_i} - \max(\mathrm{EC}_{t_{i+1}}, \mathrm{KVA}_{t_i})\right)^+\Big] \\
L_{t_{i+1}} - L_{t_i} &= \mathrm{CVA}_{t_{i+1}} - \mathrm{CVA}_{t_i} + (\mathrm{MtM}^{(c)}_{t_i})^+ \mathbb{1}_{\{t_i < \tau^{(c)} \le t_{i+1}\}} + \mathrm{FVA}_{t_{i+1}} - \mathrm{FVA}_{t_i} \\
&\quad + h\gamma^{(b)}_{t_i}\left(\sum_c \mathrm{MtM}^{(c)}_{t_i} \mathbb{1}_{\{\tau^{(c)} > t_i\}} - \mathrm{CVA}_{t_i} - \mathrm{FVA}_{t_i} - \max(\mathrm{EC}_{t_{i+1}}, \mathrm{KVA}_{t_i})\right)^+ \\
\mathrm{EC}_{t_i} &= \mathbb{ES}_{t_i}[L_{\bar{t}_i} - L_{t_i}].
\end{aligned} \tag{34}$$

Indeed, this hybrid scheme admits the following explicit reformulation: $\mathrm{EC}_{t_n} = \mathrm{FVA}_{t_n} = \mathrm{KVA}_{t_n} = 0$ *and, for* $i = n-1 \cdots 0$,

$$\begin{aligned}
\mathrm{KVA}_{t_i} &= \exp(-rh)\big(\mathbb{E}_{t_i}[\mathrm{KVA}_{t_{i+1}}] + rh \max\{\mathrm{EC}_{t_{i+1}}, \tfrac{\exp(-rh)}{1-rh\exp(-rh)}\,\mathrm{KVA}_{t_i}\}\big) \\
\mathrm{FVA}_{t_i} &= \mathbb{E}_{t_i}[\mathrm{FVA}_{t_{i+1}}] + \tfrac{h\gamma^{(b)}_{t_i}}{1+h\gamma^{(b)}_{t_i}} \\
&\qquad \Big(\textstyle\sum_c \mathrm{MtM}^{(c)}_{t_i} \mathbb{1}_{\{\tau^{(c)} > t_i\}} - \mathrm{CVA}_{t_i} - \mathbb{E}_{t_i}[\mathrm{FVA}_{t_{i+1}}] - \max(\mathrm{EC}_{t_{i+1}}, \mathrm{KVA}_{t_i})\Big)^+,
\end{aligned}$$

and the same last two lines as (34).

References

Abbas-Turki, L., S. Crépey, and B. Saadeddine (2022). Pathwise CVA regressions with oversimulated defaults. *Mathematical Finance*. Forthcoming.

Agarwal, A., S. D. Marco, E. Gobet, J. López-Salas, F. Noubiagain, and A. Zhou (2019). Numerical approximations of McKean anticipative backward stochastic differential equations arising in initial margin requirements. *ESAIM: Proceedings and Surveys 65*, 1–26.

Albanese, C., S. Caenazzo, and S. Crépey (2017). Credit, funding, margin, and capital valuation adjustments for bilateral portfolios. *Probability, Uncertainty and Quantitative Risk 2*(7), 26 pages.

Albanese, C., S. Crépey, R. Hoskinson, and B. Saadeddine (2021). XVA analysis from the balance sheet. *Quantitative Finance 21*(1), 99–123.

Barrera, D., S. Crépey, B. Diallo, G. Fort, E. Gobet, and U. Stazhynski (2019). Stochastic approximation schemes for economic capital and risk margin computations. *ESAIM: Proceedings and Surveys 65*, 182–218.

Barrera, D., S. Crépey, E. Gobet, H.-D. Nguyen, and B. Saadeddine (2022). Learning value-at-risk and expected shortfall. arXiv:2209.06476.

Baydin, A. G., B. A. Pearlmutter, A. A. Radul, and J. M. Siskind (2018). Automatic differentiation in machine learning: a survey. *Journal of Marchine Learning Research 18*, 1–43.

Bouchard, B. and R. Élie (2008). Discrete time approximation of decoupled forward-backward SDE with jumps. *Stochastic Processes and Applications 118*(1), 53–75.

Bouchard, B. and N. Touzi (2004). Discrete-time approximation and Monte-carlo simulation of backward stochastic differential equations. *Stochastic Processes and their applications 111*(2), 175–206.

Chassagneux, J.-F. and D. Crisan (2014). Runge–Kutta schemes for backward stochastic differential equations. *The Annals of Applied Probability 24*(2), 679–720.

Chassagneux, J.-F., D. Crisan, and F. Delarue (2019). Numerical method for FBSDEs of McKean-Vlasov type. *The Annals of Applied Probability 29*(3), 1640–1684.

Chassagneux, J.-F. and A. Richou (2016). Numerical simulation of quadratic BSDEs. *The Annals of Applied Probability 26*(1), 262–304.

Chassagneux, J.-F. and A. Richou (2019). Rate of convergence for the discrete-time approximation of reflected BSDEs arising in switching problems. *Stochastic Processes and their Applications 129*(11), 4597–4637.

Crépey, S. (2022). Positive XVAs. *Frontiers of Mathematical Finance 1*(3), 425–465. doi: 10.3934/fmf.2022003.

Crépey, S., W. Sabbagh, and S. Song (2020). When capital is a funding source: The anticipated backward stochastic differential equations of X-Value Adjustments. *SIAM Journal on Financial Mathematics 11*(1), 99–130.

Crépey, S. and S. Song (2015). BSDEs of counterparty risk. *Stochastic Processes and their Applications 125*(8), 3023–3052.

Crisan, D., K. Manolarakis, and N. Touzi (2010). On the Monte Carlo simulation of BSDEs: An improvement on the Malliavin weights. *Stochastic Processes and their Applications 120*(7), 1133–1158.

Delarue, F. and S. Menozzi (2006). A forward–backward stochastic algorithm for quasi-linear PDEs. *The Annals of Applied Probability 16*(1), 140–184.

E, W., J. Han, and A. Jentzen (2017). Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. *Communications in Mathematics and Statistics 5*(4), 370–398.

Gnoatto, A., C. Reisinger, and A. Picarelli (2021). Deep xVA solver–a neural network based counterparty credit risk management framework. *arXiv:2005.02633*.

Gobet, E. (2016). *Monte-Carlo methods and stochastic processes: from linear to non-linear*. Chapman and Hall/CRC.

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

Goodfellow, I., Y. Bengio, and A. Courville (2016). *Deep Learning*. MIT Press.

He, K., X. Zhang, S. Ren, and J. Sun (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034.

Henry-Labordère, P. (2019). CVA and IM: welcome to the machine. *Risk Magazine*, March. In preprint form: ssrn.3071506.

Henry-Labordere, P., X. Tan, and N. Touzi (2017). Unbiased simulation of stochastic differential equations. *The Annals of Applied Probability 27*(6), 3305–3341.

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

Jacod, J. (1979). *Calcul Stochastique et Problèmes de Martingales*. Lecture Notes Math. 714. Springer.

Kingma, D. P. and J. Ba (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Lyons, T. and N. Victoir (2004). Cubature on Wiener space. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences 460*(2041), 169–198.

Peng, S. and Z. Yang (2009). Anticipated backward stochastic differential equations. *The Annals of Probability 37*(3), 877–902.

Savine, A. (2018). *Modern computational finance: AAD and parallel simulations*. John Wiley & Sons.

Teng, L. (2022). Gradient boosting-based numerical methods for high-dimensional backward stochastic differential equations. *Applied Mathematics and Computation 426*, 127119.

Weinan, E., M. Hutzenthaler, A. Jentzen, and T. Kruse (2019). On multilevel Picard numerical approximations for high-dimensional nonlinear parabolic partial differential equations and high-dimensional nonlinear backward stochastic differential equations. *Journal of Scientific Computing 79*(3), 1534–1571.

Zhang, J. (2004). A numerical scheme for BSDEs. *The Annals of Applied Probability 14*(1), 459–488.