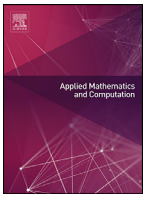
Applied Mathematics and Computation 409 (2021) 126399

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc



Deep learning for CVA computations of large portfolios of financial derivatives

Kristoffer Andersson[a,*], Cornelis W. Oosterlee[a,b]

[a] CWI - National Research Institute for Mathematics and Computer Science, Amsterdam, the Netherlands
[b] Delft Institute of Applied Mathematics, Delft University of Technology, Delft, the Netherlands

ARTICLE INFO

Article history:
Received 25 February 2021
Revised 18 May 2021
Accepted 19 May 2021

Keywords:
Portfolio CVA
Expected shortfall
WWR
Bermudan options
Deep learning

ABSTRACT

In this paper, we propose a neural network-based method for CVA computations of a portfolio of derivatives. In particular, we focus on portfolios consisting of a combination of derivatives, with and without true optionality, *e.g.*, a portfolio of a mix of European- and Bermudan-type derivatives. CVA is computed, with and without netting, for different levels of WWR and for different levels of credit quality of the counterparty. We show that the CVA is overestimated with up to 25% by using the standard procedure of not adjusting the exercise strategy for the default-risk of the counterparty. For the Expected Shortfall of the CVA dynamics, the overestimation was found to be more than 100% in some non-extreme cases.

1. Introduction

In this paper, we consider a set of financial contracts, which we refer to as the *portfolio of derivatives*, or just the *portfolio*, written between two parties. The first party is referred to as the *bank* and is considered to be default-free. The second party, which may default, is referred to as the *counterparty*. We take the perspective of the default-free bank in order to investigate some of the risks associated with a defaultable counterparty. It is straightforward to extend the methodologies used in this paper to a defaultable bank as well as to multiple counterparties.

1.1. Risk-free valuation

We consider the problem of finding the value of a portfolio of derivatives with early-exercise features. In particular, we are focusing on portfolios with multiple derivatives with true optionality, *e.g.*, American or Bermudan derivatives. We construct a portfolio of J derivatives, where the individual derivatives depend on $d_1, d_2, \dots, d_J$ risk factors. This means that we could face high-dimensionality in two ways:

1. Derivative j could depend on a large number of risk factors, *i.e.*, d_j could be large;
2. We could have many derivatives in the portfolio, *i.e.*, J could be large.

[*] Corresponding author. National Research Institute for Mathematics and Computer Science, P.O. Box 94079, 1090 GB, Amsterdam, the Netherlands.
E-mail address: kristoffer.andersson@cwi.nl (K. Andersson).

https://doi.org/10.1016/j.amc.2021.126399

Table 1
Exposure given different exercise decisions at t, with and without a netting agreement.

	Without netting	With netting
Exposure - no exercise	20	10
Exposure - exercise one of the American options	10	0
Exposure - exercise both American options	0	0

In [1], a neural network-based method for valuation of a single Bermudan derivative was proposed and proved to be highly accurate for derivatives with up to 100 risk factors. Later, the algorithm was extended in [2] to also include pathwise valuations of the derivative (in contrast to only finding the value at the initial time). In this paper, we extend [1] and [2] to the portfolio case, *i.e.,* finding the value of a large portfolio of, possibly high-dimensional, derivatives with true optionality, without having to compute the value of each individual derivative.

In a traditional setting, the so-called continuation value is computed, and subsequently, the value of the derivative is given by the maximum of the continuation value and the immediate pay-off. For a single derivative, this is straightforward. For instance, the continuation value can be computed by solving an associated PDE, which is done in *e.g.,* [3–6] and [7], or the continuation value can be approximated by a Fourier transform methodology, which is done in e.g., [8,9] and [10]. Furthermore, classical tree-based methods such as [11,12] and [13], can be used. These types of methods are, in general, highly accurate but they suffer severely from the curse of dimensionality, meaning that they are computationally feasible only in low dimensions (say up to 4 risk factors), see [14]. In higher dimensions, Monte-Carlo-based methods are often used, see *e.g.,* [15–18] and [19]. Monte-Carlo-based methods can generate highly accurate derivative values at the initial time, but often less accurate values between the initial time and maturity of the contract.

In contrast to the single derivative case, it is not enough to know the continuation value of a portfolio (with more than one derivative) in order to decide optimally which derivatives should be exercised. Therefore, it is common to do the valuation at the level of each derivative, and then add the individual values of each derivative to obtain the portfolio value. This becomes cumbersome for large portfolios. As mentioned above, the methodology used in this paper, generalizes [1] and [2], in which the optimal exercise policy is approximated by maximizing expected discounted cash-flows, *i.e.,* the continuation value is not computed. By not relying on computations of the continuation value, the algorithm is able to compute the portfolio value without having to compute the individual values for each derivative.

1.2. Risky valuation and CVA

The Credit Valuation Adjustment (CVA) is the difference between the risk-free portfolio value and the risky portfolio value, where the risky portfolio value is defined as the portfolio value when taking default risk of the counterparty into account. While there is no ambiguity of the risk-free portfolio value, it is not completely clear how the risky portfolio value should be computed. The question is whether the exercise policy should be adjusted for the fact that the counterparty may default. For instance, if the counterparty ends up in financial distress, it is reasonable to assume that the bank (which in this paper is assumed to be the risk-free party) would be more willing to exercise the callable derivatives, in order to lower its exposure to the counterparty. Even though it seems common to ignore the effect of a defaultable counterparty when computing risky derivative values, it has been discussed in the literature, see *e.g.,* [20–23]. In the case of a single derivative [20] states that the exercise region for a risk-free derivative is always a subset of the exercise region for a risky counterpart. However, in the case of a portfolio, the situation is more complex, and depends on contractual details such as the close-out and netting agreements. One consequence is that, in the presence of a netting agreement, the exercise decisions can no longer be made individually. To explain this, we give a simple example.

Example 1.1. Assume that we have a portfolio, consisting of three derivatives, one European future and two American options. All contracts are initialized at time 0, mature at time T and depend on the same risk-factor $(X_t)_{t\in[0,T]}$. Assume that, at time $t\in(0,T)$, and given $X_t=x$, the intrinsic values are

$$V^{\text{future}}(t,x)=-10,\quad V_1^{\text{Am}}(t,x)=10,\quad V_2^{\text{Am}}(t,x)=10,$$

and the immediate pay-off for the American options satisfy

$$g_1^{\text{Am}}(t,x)<10,\quad g_2^{\text{Am}}(t,x)<10.$$

In a risk free environment (no-defaultable counterparty), it is sub-optimal to exercise the American options. However, in case of a defaultable counterparty, the situation is less trivial. In Table 1, the exposure to the counterparty, given different exercise decisions at t, is given with and without a netting agreement.

If the counterparty is in severe financial distress, then it is likely optimal for the bank to exercise both American options in the case of no netting agreement, and one of them in the case of a netting agreement. From this simple example, two things become clear; 1) The exercise decisions for the American options are affected not only by a risky counterparty, but also by whether or not a netting agreement exists. 2) in the presence of netting, exercise decisions cannot be made for one derivative in isolation, but only for all the American options simultaneously.

In general, for a risky portfolio, it is not possible to describe the value of a single derivative, but only the value of the entire portfolio. This is an interesting problem since almost all existing algorithms rely on exercise decisions made in isolation and risky derivative values that can be added up to obtain the risky portfolio value.

If this is not taken into account we would obtain a biased low valuation for the risky portfolio by using a sub-optimal exercise strategy. Since the CVA is the difference between the risk-free and risky portfolio values, we would obtain an overestimation of the CVA. Furthermore, this effect is likely to increase with decreasing credit quality of the counterparty. In practice, this means that the counterparty is paying a CVA which is based on a sub-optimal exercise strategy used by the bank, which is out of control for the counterparty. Even more problematic is that the overestimation of the CVA is higher for counterparties that already are under financial distress.

One could argue that it is reasonable for the bank to charge the counterparty the higher CVA, since the bank will probably not follow the theoretically optimal risky exercise strategy. However, there is another level of complexity not yet discussed. When the mark-to-market (MtM) CVA moves in time against the bank, the bank could face losses, not because the counterparty actually defaults, but because disadvantageous changes in the MtM CVA. For instance, in Basel III [24] the following is stated:

"*Under Basel II, the risk of counterparty default and credit mitigation risk were addressed but mark-to-market losses due to credit valuation adjustments (CVA) were not. During the global financial crisis, however, roughly two-thirds of losses attributed to counterparty credit risk were due to CVA losses and only about one-third were due to actual defaults.*"

This is further discussed in [25], in which the authors also recommend computations of different risk measures for the future distribution of CVA. Two examples of such measures are the Value at Risk of the CVA (VaR-CVA) and the Expected Shortfall of the CVA (ES-CVA). The advantage of the ES-CVA is that it is a coherent risk-measure, and we therefore focus on ES-CVA in this paper.

1.3. Structure of the paper

In Section 2 the mathematical problem formulation is given. We define the risk-free and risky portfolios, close-out agreements both with and without netting agreements and the associate CVA. Furthermore, the problems are formulated in terms of so-called decision functions, which control the exercise strategies. In Section 3, the algorithms are presented. In the first part, the algorithm for learning optimal exercise strategies is given and in the second part, an algorithm for learning pathwise entities such as the pathwise portfolio exposure is presented. Finally, in Section 4 numerical experiments are presented. The experiments include a first part, in which risk-free values are computed and compared to a well-established regression based method. In the second part we compare CVA computed with the risk-free and the risky exercise strategy to verify that, indeed, the CVA is often overestimated with algorithms in use today. We present comparisons with and without netting, for different levels of Wrong Way Risk (WWR), and for different credit quality of the counterparty. As a final example, we analyse the effect of the different exercise strategies on ES-CVA. In the Appendix, we provide some additional details on the algorithms and the specific choice of neural networks.

2. Problem formulation

Let $(\Omega, \mathcal{F}, \mathbb{Q})$ be a probability space completed with the $\mathbb{Q}-$null-sets of $\mathcal{F}$. For $T \in (0, \infty)$, and[1] $d \in \mathbb{N}$, let $X : [0, T] \times \Omega \to \mathbb{R}^d$ and $r : [0, T] \times \Omega \to \mathbb{R}$ represent the (market) risk-factors of the portfolio and the short rate, respectively. Furthermore, we denote by τ^{D} the default event of the counterparty, which is a stopping time defined on $(\Omega, \mathcal{F}, \mathbb{Q})$ and we let $\mathbb{1}^{\mathrm{D}} : [0, T] \times \Omega \to \{0, 1\}$, be the *jump-to-default* process given by

$$\mathbb{1}^{\mathrm{D}}_t \coloneqq \mathbb{I}_{\{t<\tau^{\mathrm{D}}\}}. \tag{2.1}$$

The information structure is given by the sub$-\sigma-$algebras generated by X, r and $\mathbb{1}^{\mathrm{D}}$, *i.e.*, $\mathcal{H}^X_t = \sigma(X_s : s \in [0, t])$, $\mathcal{H}^r_t = \sigma(r_s : s \in [0, t])$ and $\mathcal{G}_t = \sigma(\mathbb{1}^{\mathrm{D}}_s : s \in [0, t])$ and we define the enlarged filtrations $\mathcal{H}_t = \mathcal{H}^X_t \wedge \mathcal{H}^r_t$ and $\mathcal{F}_t = \mathcal{H}_t \wedge \mathcal{G}_t$. In this paper, we use either a constant short rate (risk-free rate), or we view the short rate as one of the risk factors. In the latter case, we model the short rate as one of the d component processes of X, which implies that, $\mathcal{H}_t = \mathcal{H}^X_t$. The motivation for introducing a separate notation for the short rate is to simplify the notation when the short rate is used to discount cash-flows. For commonly used conditional expectations, we introduce the short-hand notations $\mathbb{E}_{t,x}[\,\cdot\,] \coloneqq \mathbb{E}^{\mathbb{Q}}[\,\cdot\,|\,X_t = x]$, $\mathbb{E}_{t,x,\nu}[\,\cdot\,] \coloneqq \mathbb{E}^{\mathbb{Q}}[\,\cdot\,|\,X_t = x,\, \mathbb{1}^{\mathrm{D}}_t = \nu]$ and $\mathbb{E}_t[\,\cdot\,] \coloneqq \mathbb{E}^{\mathbb{Q}}[\,\cdot\,|\,\mathcal{H}_t]$.

We use a numéraire, which, for $t \in [0, T]$, is defined by $B_t \coloneqq \exp(\int_0^t r_s \mathrm{d}s)$, which should be interpreted as the value at time t of a savings-account, which was worth 1 at time 0. For $t, u \in [0, T]$ with $t \le u$, we use $D_{t,u} \coloneqq \frac{B_t}{B_u}$ to discount a cash-flow obtained at time u back to time t. The measure $\mathbb{Q}$ is the risk-free measure, under which all tradeable assets are martingales relative to the numéraire, *e.g.*, if component $i \in \{1, 2, \ldots, d\}$ of X is tradeable, then $\frac{(X_t)_i}{B_t}$ is a $\mathbb{Q}-$martingale.

If not specifically stated otherwise, equalities and inequalities of random variables should be interpreted in a $\mathbb{Q}-$almost sure sense.

[1] We use $\mathbb{N} = \{1, 2, 3 \ldots\}$ and $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$, and $\mathbb{R}_+ = (0, \infty)$.

2.1. A portfolio of derivatives

We assume a portfolio of $J \in \mathbb{N}$ derivatives. For $t \in [0, T]$, and for derivative $j \in \{1, 2, \ldots, J\}$, we denote the set of exercise dates greater than or equal to t by $\mathbb{T}_j(t) \subseteq [0, T]$, and set $\mathbb{T}(t) = \{\mathbb{T}_1(t), \mathbb{T}_2(t), \ldots, \mathbb{T}_J(t)\}$. In other words, $\mathbb{T}_j(t)$ represents the remaining exercise dates of derivative j at time t and $\mathbb{T}(t)$ is the a list of the remaining exercise dates for all the J derivatives. Note that for a European-type contract, the only exercise date is at the maturity, for a Bermudan-type contract there are multiple exercise dates, and for an American-type contract, there are infinitely many exercise dates. We emphasize that the exercise dates are simply subsets of the time interval $[0, T]$, and provide no information on which exercise policy to follow, except in some trivial cases *e.g.,* when there is only one exercise date.

Since we want to be able to treat derivatives with early-exercise features, we need to introduce a framework for stopping times. For $j \in \{1, 2, \ldots, J\}$, an $X-$stopping time with respect to $\mathbb{T}_j(0)$, is a random variable, τ_j, defined on $(\Omega, \mathcal{F}, \mathbb{Q})$, taking on values in $\mathbb{T}_j(0)$, such that for all $s \in \mathbb{T}_j(t)$, it holds that the event $\{\tau_j = s\} \in \mathcal{H}_s$. Furthermore, we define an $X^{t,x}-$stopping time as an $X-$stopping time, conditional on $X_t = x$, and $\tau \geq t$.

For each derivative, $j \in \{1, 2, \ldots, J\}$ we use individual pay-off functions, $g_j : [0, T] \times \mathbb{R}^d \to \mathbb{R}$, which, for $t, s \in [0, T]$, with $s \geq t$, are assumed to satisfy

$$\mathbb{E}_0[|D_{t,s} g_j(s, X_s)|^2] < \infty. \tag{2.2}$$

Since we are treating portfolios where the individual derivatives may have different maturities, we set each pay-off function to zero for all times larger than its maturity, *i.e.*, for $j \in \{1, 2, \ldots, J\}$, $x \in \mathbb{R}^d$ and for $t > \max\{\mathbb{T}_j(0)\}$, we set $g_j(t, x) \equiv 0$, where $\max\{\mathbb{T}_j(0)\}$ represents the largest element belonging to the set $\mathbb{T}_j(0)$.

2.2. Risk-free and risky portfolio valuation without netting

The value of a derivative (not) taking default risk of the counterparty into account is referred to as the *risky* (*risk-free*) value. We define the risk-free and the risky values of derivative $j \in \{1, 2, \ldots, J\}$, at market state $(t \in [0, T], X_t = x \in \mathbb{R}^d)$, and default state $\mathbb{1}_t^{\mathrm{D}} = \nu \in \{0, 1\}$, by

$$V_j(t, x) \coloneqq \sup_{\tau \in \mathcal{T}_j(t)} \mathbb{E}_{t,x}\Big[D_{t,\tau} g_j(\tau, X_\tau)\Big], \quad \text{(risk-free value)}, \tag{2.3}$$

$$\begin{aligned} U_j(t, x, \nu) := \nu \sup_{\tau \in \mathcal{T}_j(t)} \mathbb{E}_{t,x,1}\Big[&\mathbb{1}_\tau^{\mathrm{D}} D_{t,\tau} g_j(\tau, X_\tau) \\ &+ (1 - \mathbb{1}_\tau^{\mathrm{D}}) D_{t,\tau^{\mathrm{D}}}\Big(R V_j(\tau^{\mathrm{D}}, X_{\tau^{\mathrm{D}}})^+ + V_j(\tau^{\mathrm{D}}, X_{\tau^{\mathrm{D}}})^-\Big)\Big], \quad \text{(risky value)}, \end{aligned} \tag{2.4}$$

where $\mathcal{T}_j(t)$ is the set of all $X-$stopping times taking on values in $\mathbb{T}_j(t)$ and for $x \in \mathbb{R}$, $(x)^+ = \max\{0, x\}$ and $(x)^- = \min\{0, x\}$. In the above, we assume a close-out agreement which uses the risk-free derivative values as reference valuation. At default of the counterparty, the bank receives only a fraction, $R \in [0, 1)$, referred to as the *recovery-rate*, of the positive part of each derivative. On the other hand, each derivative with a negative risk-free value at default needs to be added entirely to the portfolio.

Note that for the risky value we need additional information of prior defaults of the counterparty, which is captured in the realization, $\nu \in \{0, 1\}$, of the jump-to-default process, *i.e.,* $\nu = 1$ if no default has occurred prior to, or at t, and $\nu = 0$ otherwise. The notation above trivially holds for European-type derivatives since the only exercise date is at maturity of the contract. Furthermore, a barrier-type feature could be added by also including a spatial dimension to $\mathbb{T}_j(0)$. The value of a portfolio, consisting of J derivatives, at market state $(t, X_t = x)$ and default state $\mathbb{1}_t^{\mathrm{D}} = \nu$, without netting, is given by

$$\Pi^V(t, x) \coloneqq \sum_{j=1}^{J} V_j(t, x), \quad \text{(risk-free portfolio value)},$$

$$\Pi^U(t, x, \nu) \coloneqq \sum_{j=1}^{J} U_j(t, x, \nu) = \nu \sum_{j=1}^{J} U_j(t, x, 1), \quad \text{(risky portfolio value)}.$$

Using (2.3) and (2.4), the above can be written as

$$\Pi^V(t, x) = \sum_{j=1}^{J} \sup_{\tau_j \in \mathcal{T}_j(t)} \mathbb{E}_{t,x}\Big[D_{t,\tau_j} g_j(\tau_j, X_{\tau_j})\Big] \tag{2.5}$$

$$\begin{aligned} \Pi^U(t, x, \nu) = \nu \sum_{j=1}^{J} \sup_{\tau_j \in \mathcal{T}_j(t)} \mathbb{E}_{t,x,1}\Bigg[&\mathbb{1}_{\tau_j}^{\mathrm{D}} D_{t,\tau_j} g_j(\tau_j, X_{\tau_j}) \\ &+ (1 - \mathbb{1}_{\tau_j}^{\mathrm{D}}) D_{t,\tau^{\mathrm{D}}}\Big(R V_j(\tau^{\mathrm{D}}, X_{\tau^{\mathrm{D}}})^+ + V_j(\tau^{\mathrm{D}}, X_{\tau^{\mathrm{D}}})^-\Big)\Bigg]. \end{aligned} \tag{2.6}$$

Since the aim is to approximate the optimal exercise policy with neural networks, we wish to re-formulate the problem into an optimization problem, in which the target function can be represented by a neural network. Following [1] and [2] we use so-called decision functions, to determine for each derivative and given a market state, whether or not to exercise the derivative. For $j \in \{1, 2, \dots, J\}$, decision function j denoted by f_j, is of the form $f_j : [0, T] \times \mathbb{R}^d \to \{0, 1\}$. In order to guarantee that an exercise decision can only occur at an exercise date, we require for $s \notin \mathbb{T}_j(0)$, that $f_j(s, \cdot) \equiv 0$.

We now restrict our attention to the case when there, for each derivative, is a finite number of exercise dates, *i.e.*, for $j \in \{1, 2, \dots, J\}$, it holds that $|\mathbb{T}_j(0)| \in \mathbb{N}$. From a theoretical perspective, this excludes American-type derivatives, but from a practical perspective, an infinite number of exercise dates is often approximated by a large, but finite, number of exercise dates. This implies that we can still consider American-type derivatives by increasing the number of exercise dates until the derivative value converges (until the value does not increase with additional exercise dates). We denote by $\mathbb{T}^{\Pi}(t)$ the set of dates which represent an exercise date for at least one of the J derivatives. Mathematically, we define the exercise dates of the portfolio as

$$\mathbb{T}^{\Pi}(t) \coloneqq \bigcup_{j=1}^{J} \mathbb{T}_j(t), \tag{2.7}$$

and the number of unique exercise dates in the portfolio is given by $N = |\mathbb{T}^{\Pi}(0)|$. We assume that the initial time $T_0 = 0$ is not an exercise date for any of the derivatives.

To simplify, we use the following notation for the N exercise dates, the risk-factors evaluated at the N exercise dates, and the discounting between exercise dates

$$\mathbb{T}^{\Pi}(0) = \{T_1, T_2, \dots, T_N = T\}, \tag{2.8}$$

$$X_k \coloneqq X_{T_k}, \text{ and } D_{k,\ell} \coloneqq D_{T_k,T_\ell} \quad \text{for } k, \ell = 1, 2, \dots, N. \tag{2.9}$$

Furthermore, for $t \in [0, T]$, the risk-factor process on $[t, T]$, conditional on $X_t = x$, is denoted by $X^{t,x} = (X_s)_{s \in [t,T]}$, where we also note that $X^{0,x_0} = X$. The above notation allows us to express an $X^{t,x}-$stopping time in terms of decision functions[2]

$$\tau_j^n[f_j](X^{t,x}) \coloneqq \sum_{k=n}^{N} T_k f_j(T_k, X_k) \prod_{m=n}^{k-1} (1 - f_j(T_m, X_m)). \tag{2.10}$$

The notation above is used to emphasize that, the decision function f_j, controls the exercise strategy, given the stochastic process $X^{t,x}$. Moreover, $X^{t,x}$ is not just a random value at a specific time, but the entire process, starting at $X_t = x$ and until stopping occurs. In later sections the valuation of a derivative or a portfolio is formulated as an optimization problem, which is optimized by varying f_j. Although the notation is practical when optimization is discussed, it is cumbersome to use when we define the value of a derivative. We therefore use the following short-hand notation

$$\bar{\tau}_{n,j} \coloneqq \tau_j^n[f_j](X^{t,x}), \tag{2.11}$$

and keep in mind, that the strategy is controlled by a decision function, f_j, and for $u \geq t$, the event $\mathbb{I}_{\{\bar{\tau}_{n,j} \leq u\}}$ is $\sigma(X^{t,x})-$measurable. We can now define the value of the risk-free and risky derivatives, given an exercise strategy expressed in terms of decision functions. For derivative $j \in \{1, 2, \dots, J\}$, market state $(t, x) \in (T_{n-1}, T_n] \times \mathbb{R}^d$, we define the parametrized valuation functions

$$\mathcal{V}_j(t, x \,|\, f_j) \coloneqq \mathbb{E}_{t,x}\Big[D_{t,\bar{\tau}_{n,j}}\, g_j\big(\bar{\tau}_{n,j}, X_{\bar{\tau}_{n,j}}\big)\Big], \tag{2.12}$$

$$\begin{aligned}\mathcal{U}_j(t, x \,|\, f_j) :=\ & \mathbb{E}_{t,x,1}\Big[\mathbb{1}^{\mathrm{D}}_{\bar{\tau}_{n,j}} D_{t,\bar{\tau}_{n,j}} g_j\big(\bar{\tau}_{n,j}, X_{\bar{\tau}_{n,j}}\big) \\ & + \big(1 - \mathbb{1}^{\mathrm{D}}_{\bar{\tau}_{n,j}}\big) D_{t,\tau^{\mathrm{D}}} \big(RV_j(\tau^{\mathrm{D}}, X_{\tau^{\mathrm{D}}})^{+} + V_j(\tau^{\mathrm{D}}, X_{\tau^{\mathrm{D}}})^{-}\big)\Big].\end{aligned} \tag{2.13}$$

Similarly, we define the portfolio values with respect to the exercise strategy given by $\boldsymbol{f}$, as the parametrized functions

$$\Upsilon^{V}(t, x \,|\, \boldsymbol{f}) \coloneqq \sum_{j=1}^{J} \mathcal{V}_j(t, x \,|\, f_j), \quad \Upsilon^{U}(t, x, \nu \,|\, \boldsymbol{f}) \coloneqq \nu \sum_{j=1}^{J} \mathcal{U}_j(t, x \,|\, f_j), \tag{2.14}$$

where the value of the risky portfolio also depends on the default state of the counterparty, $\mathbb{1}^{\mathrm{D}}_t = \nu \in \{0, 1\}$. We now want to find decision functions such that, when inserted in (2.12) and (2.13), we obtain (2.3) and (2.4). With this in mind, we define for $j \in \{1, 2, \dots, J\}$, at $t \in [0, T]$, the (optimal) exercise regions, $\mathcal{E}_j^Z(t)$, in which it is optimal to exercise, and the (optimal) continuation regions, $\mathcal{C}_j^Z(t)$, in which it is optimal to hold on, by

$$\mathcal{E}_j^Z(t) \coloneqq \Big\{x \in \mathbb{R}^d \,|\, Z_j(t, x) = g_j(t, x) \text{ and } t \in \mathbb{T}_j(0)\Big\},$$

[2] The empty product is defined as 1.

$$\mathcal{C}_j^Z(t) \coloneqq \Big\{x \in \mathbb{R}^d \,|\, Z_j(t,x) > g_j(t,x) \text{ or } t \notin \mathbb{T}_j(0)\Big\}, \text{ for } Z \in \{V, U\}.$$

The above states that the derivative should be exercised if its value equals the immediate exercise value, and we are at an exercise date and the derivative should not be exercised if its value is greater than the immediate exercise value or if we are not at an exercise date. Note that $\mathcal{E}_j^Z(t) \cup \mathcal{C}_j^Z(t) = \mathbb{R}^d$ and $\mathcal{E}_j^Z(t) \cap \mathcal{C}_j^Z(t) = \emptyset$. For $t \in [0, T]$, a decision function, $j \in \{1, 2, \ldots, J\}$, can then be defined as

$$f_j^Z(t,x) \coloneqq \mathbb{I}_{\{x \in \mathcal{E}_j^Z(t)\}}, \text{ for } Z \in \{V, U\}. \tag{2.15}$$

Furthermore, we denote by $\boldsymbol{f}^Z$, the vector consisting of the individual decision functions

$$\boldsymbol{f}^Z(t,x) \coloneqq (f_1^Z(t,x), f_2^Z(t,x), \ldots, f_J^Z(t,x))^T, \text{ for } Z \in \{V, U\}. \tag{2.16}$$

For market states $(t,x) \in [0,T] \times \mathbb{R}^d$ and for a derivative $j \in \{1, 2, \ldots, J\}$, it holds that

$$\mathcal{V}_j\big(t, x \,|\, f_j^{\mathrm{V}}\big) = V_j(t,x), \quad \mathcal{U}_j\big(t, x \,|\, f_j^U\big) = U_j(t,x,1).$$

The validity of the above is a direct consequence of Proposition 4 in [1]. In turn, this implies that by inserting the optimal decision functions in the functionals in equations (2.14), we obtain the risky and risk-free portfolio values, i.e.,

$$\Upsilon^V\big(t, x \,|\, \boldsymbol{f}^V\big) = \Pi^V(t,x), \quad \Upsilon^U\big(t, x, \nu \,|\, \boldsymbol{f}^U\big) = \Pi^U(t,x,\nu).$$

In subsequent sections the optimal decision function $\boldsymbol{f}^Z$, for $Z \in \{V, U\}$, is approximated with a series of neural networks. The reason for using the rather complicated notation, (2.10), is that this structure allows us to view the valuation of the derivatives as an optimization problem over the set of decision functions, which we approximate on some finite-dimensional function space. One example of such function space is the functions generated by a series of neural networks with a fixed number of parameters. When we use a specific strategy, *e.g.*, $\boldsymbol{f}^V$ or $\boldsymbol{f}^U$, this is specified by adding a superscript referring to the particular strategy. For $Z \in \{V, U\}$, we define the short-hand notation

$$\bar{\tau}_{n,j}^Z \coloneqq \tau_j^n[f_j^Z](X^{t,x}), \tag{2.17}$$

where it is assumed that $t \in (T_{n-1}, T_n]$.

2.3. Risky portfolio valuation with netting

When considering the risky portfolio value with netting, the problem becomes nonlinear in the sense that the risky portfolio value is no longer the sum of the individual risky derivative values. In fact, there no longer exists ”a risky value for a single derivative”, since the valuation needs to be carried out on a portfolio level. Before we define the risky value of a netted portfolio, we need to define the process[3] $A : [0,T] \times \Omega \to \{0,1\}^J$, which for $t \in [0,T]$ and $j \in \{1, 2, \ldots, J\}$ satisfies

$$(A_t)_j = \begin{cases} 0, & \text{if derivative } j \text{ has been exercised prior to } t, \\ 1, & \text{else.} \end{cases}$$

Similar to (2.9), we use the short-hand notation for A at initial date, T_0, the exercise dates, $T_1, \ldots, T_N$,

$$A_k \coloneqq A_{T_k}, \quad \text{for } k = 0, 1, \ldots, N. \tag{2.18}$$

The process A is $\mathcal{H}_t$−measurable but it is not enough to know $X_t = x$ in order to determine A_t. The reason for defining A is that the exercise decisions for the netted risky portfolio are defined by a J−dimensional $(X^{t,x}, A^{t,\alpha})$−stopping times vector, where $A^{t,\alpha} = (A_s)_{s \in [t,T]}$ conditional on $A_t = \alpha$. This means that, at each exercise date, in addition to the current market state, we need to know which derivatives in the portfolio have been exercised prior to the current time, in order to make optimal exercise decisions. We denote, by $\mathcal{T}'(t)$, the space of $(X^{t,x}, A^{t,\alpha})$−stopping times vectors, taking on values in $\{\mathbb{T}_1(t), \mathbb{T}_2(t), \ldots, \mathbb{T}_J(t)\}$. Furthermore, for $t \in (T_{n-1}, T_n]$, we denote by τ_j^n element j of a stopping times vector $\boldsymbol{\tau}^n \in \mathcal{T}'(t)$. The netted portfolio value with a risky counterparty, given market state $X_t = x$, default state of the counterparty $\mathbb{1}_t^{\mathrm{D}} = \nu$ and portfolio state (exercise state of the derivatives in the portfolio) $A_t = \alpha$, is given by

$$\begin{aligned} \Pi^A(t,x,\nu,\alpha) :=\ & \nu \sup_{\boldsymbol{\tau} \in \mathcal{T}'(t)} \mathbb{E}_{t,x,1,a}\Bigg[\sum_{j=1}^J \alpha_j \mathbb{1}_{\tau_j}^{\mathrm{D}} D_{t,\tau_j} g_j(\tau_j, X_{\tau_j}) \\ & + D_{t,\tau^{\mathrm{D}}}\Bigg(R\Bigg(\sum_{k=1}^J \alpha_k (1 - \mathbb{1}_{\tau_k}^{\mathrm{D}}) V_k(\tau^{\mathrm{D}}, X_{\tau^{\mathrm{D}}})\Bigg)^+ + \Bigg(\sum_{\ell=1}^J \alpha_\ell (1 - \mathbb{1}_{\tau_\ell}^{\mathrm{D}}) V_\ell(\tau^{\mathrm{D}}, X_{\tau^{\mathrm{D}}})\Bigg)^-\Bigg)\Bigg], \end{aligned} \tag{2.19}$$

where $\mathbb{E}_{t,x,\nu,\alpha}[\,\cdot\,] = \mathbb{E}^{\mathbb{Q}}[\,\cdot\,|\, X_t = x, \mathbb{1}_t^{\mathrm{D}} = \nu, A_t = \alpha]$. To emphasize the importance, we put the following observations about (2.19) into two remarks.

[3] $\{0, 1\}^J$ is the Cartesian product of $\{0, 1\} \times \{0, 1\} \times \cdots \times \{0, 1\}$, J times.

Remark 2.1. The optimal stopping strategies of the individual derivatives in the portfolio are no longer independent of each other, as in (2.5) and (2.6). Furthermore, the optimal strategy depends on earlier exercise decisions, meaning that in order to make the exercise decisions Markovian, we need to include information about earlier decisions. The reason for this is the non-linearity in the two sums inside the expectation in (2.19). Therefore, the optimal stopping strategies need to be computed for the entire portfolio simultaneously. To the best of our knowledge, this has not been done in an ordinary least squares setting before. However, it is discussed in a PDE framework in the case of a portfolio of American swaptions in [23].

Remark 2.2. The value of the risky portfolio with netting, depends on the J risk-free derivative values and we are therefore required to approximate the risk-free derivative values. The reason for this is that, at default, the risk-free value of the portfolio is used as reference value in the close-out agreement (see Equation (2.19)). If we restrict our portfolio to derivatives with positive pay-off functions, then V_j can be replaced by g_j (by the definition of V_j and the law of iterated expectations). Furthermore, in the restricted portfolio, the values with and without netting coincide.

An $(X^{t,x}, A^{t,\alpha})-$stopping times vector can be defined by

$$\boldsymbol{\tau}^n[\boldsymbol{f}](X^{t,x}, A^{t,x}) \coloneqq \sum_{k=n}^{N} T_k \boldsymbol{f}(T_k, X_k, A_k) \odot \prod_{M=n}^{k-1} (\mathbf{1}_J - \boldsymbol{f}(T_m, X_m, A_m)), \tag{2.20}$$

where $\odot$ is element-wise multiplication and $\mathbf{1}_J$ is the $J-$dimensional vector with only ones, $(1, 1, \dots, 1)^T$. We denote element j of the stopping times vector by $\tau_j^n[\boldsymbol{f}](X^{t,x}, A^{t,\alpha}) = (\boldsymbol{\tau}^n[\boldsymbol{f}](X^{t,x}, A^{t,x}))_j$. We emphasize that each element of the stopping time vector depends on $\boldsymbol{f}$ and not only an element j which is the case without netting. Similar to (2.11), we introduce a short-hand notation, which simplifies the valuation function

$$\hat{\boldsymbol{\tau}}_n \coloneqq \boldsymbol{\tau}^n[\boldsymbol{f}](X^{t,x}, A^{t,x}), \quad \hat{\tau}_{n,j} \coloneqq (\hat{\boldsymbol{\tau}}_n)_j. \tag{2.21}$$

We here use "$\hat{\tau}$ ", instead of "$\bar{\tau}$" as in (2.11), to emphasize that the stopping time also takes $A^{t,\alpha}$ as an argument. The netted risky portfolio value, given the exercise strategy obtained by decision function $\boldsymbol{f}$, is then given by

$$\begin{aligned}\Upsilon^A\big(t, x, \nu, \alpha \,|\, \boldsymbol{f}\big) := \nu \mathbb{E}_{t,x,1,\alpha}\Bigg[& \sum_{j=1}^{J} \alpha_j \mathbb{1}^{\mathrm{D}}_{\hat{\tau}_{n,j}} D_{t,\hat{\tau}_{n,j}} g_j\big(\hat{\tau}_{n,j}, X_{\hat{\tau}_{n,j}}\big) \\ & + D_{t,\tau^{\mathrm{D}}}\Bigg(R\Bigg(\sum_{j=1}^{J} \alpha_j (1 - \mathbb{1}^{\mathrm{D}}_{\hat{\tau}_{n,j}}) V_j(\tau^{\mathrm{D}}, X_{\tau^{\mathrm{D}}})\Bigg)^+ + \Bigg(\sum_{j=1}^{J} \alpha_j (1 - \mathbb{1}^{\mathrm{D}}_{\hat{\tau}_{n,j}}) V_j(\tau^{\mathrm{D}}, X_{\tau^{\mathrm{D}}})\Bigg)^-\Bigg)\Bigg],\end{aligned} \tag{2.22}$$

where we, again, remind ourselves that the exercise strategy is controlled by $\boldsymbol{f}$, and for $u \geq t$, the event $\mathbb{I}_{\{\hat{\tau}_{t,j} \leq u\}}$ is $\sigma(X^{t,x}, A^{t,\alpha})-$measurable.

For a netted portfolio, the optimal exercise regions, described in Section 2.2, are less trivial. Firstly, they become dependent on the state of earlier exercise decisions, $A_t = \alpha_t \in \{0, 1\}^J$. Secondly, the exercise region for derivative $j \in \{1, 2, \dots, J\}$ is expressed under the condition that an optimal exercise strategy for the other $J - 1$ derivatives is applied. Therefore, we only describe the optimal decision function as belonging to the supremum over the space, $\mathcal{D}$, of all measurable functions, $f : [0, T] \times \mathbb{R}^d \times \{0, 1\}^J \to \{0, 1\}^J$,

$$\boldsymbol{f}^A \in \arg\max_{\boldsymbol{f} \in \mathcal{D}} \Upsilon^A\big(0, x_0, \nu_0, \alpha_0 \,|\, \boldsymbol{f}\big), \tag{2.23}$$

where $\nu_0 = 1$ (no default prior to or at $t = 0$) and $a_0 = (1, 1, \dots, 1)^T$ (no derivatives have been exercised prior to $t = 0$). We then assume that, given the state $(t, X_t = x, \mathbb{1}^{\mathrm{D}}_t = \nu, A_t = \alpha)$, the following holds

$$\Upsilon^A(t, x, \nu, \alpha \,|\, \boldsymbol{f}^A) = \Pi^A(t, x, \nu, \alpha). \tag{2.24}$$

Similar to (2.17), when we want to emphasize the particular choice of decision function, $\boldsymbol{f}^A$, we use the short-hand notation

$$\hat{\boldsymbol{\tau}}^A_n = \boldsymbol{\tau}^n[\boldsymbol{f}^A](X^{t,x}, A^{t,\alpha}) \quad \text{and} \quad \hat{\tau}^A_{n,j} = \big(\boldsymbol{\tau}^n[\boldsymbol{f}^A](X^{t,x}, A^{t,\alpha})\big)_j, \tag{2.25}$$

where it is assumed that $t \in (T_{n-1}, T_n]$.

2.4. Credit valuation adjustment of a derivative portfolio

The formal definition of CVA is the difference between the risk-free and the risky portfolio value. Given models of the underlying market and default events of our counterparty, the above definition of CVA is straightforward for a portfolio consisting of derivatives without optionality *e.g.,* European options, barrier options etc. When it comes to portfolios consisting of derivatives with true optionality, *e.g.,* the Bermudan options, American options etc. the standard procedure is not clear. In this section, we define the CVA for portfolios of derivatives with true optionality as well as some approximations, which simplify the computations. In the definitions of CVA, we use the portfolio valuations in terms of optimally chosen decision functions given in equations (2.14) and (2.24). The CVA at $(t = 0, X_0 = x_0)$, with and without netting, respectively, are given by

$$\text{CVA} \coloneqq \Upsilon^V\left(0, x_0 \,|\, \boldsymbol{f}^V\right) - \Upsilon^U\left(0, x_0, 1 \,|\, \boldsymbol{f}^U\right), \quad \text{(without netting)}, \tag{2.26}$$

$$\text{CVA}^{\text{Net}} \coloneqq \Upsilon^V\left(0, x_0 \,|\, \boldsymbol{f}^V\right) - \Upsilon^A\left(0, x_0, 1, \mathbf{1}_J \,|\, \boldsymbol{f}^A\right), \quad \text{(with netting)}. \tag{2.27}$$

A commonly used approximation is to apply the same exercise strategy to the risk-free and risky portfolios. One such approximation is defined as

$$\overline{\text{CVA}} \coloneqq \Upsilon^V\left(0, x_0 \,|\, \boldsymbol{f}^V\right) - \Upsilon^U\left(0, x_0, 1 \,|\, \boldsymbol{f}^V\right), \quad \text{(Risk-free strategy, without netting)}, \tag{2.28}$$

$$\overline{\text{CVA}}^{\text{Net}} \coloneqq \Upsilon^V\left(0, x_0 \,|\, \boldsymbol{f}^V\right) - \Upsilon^A\left(0, x_0, 1, \mathbf{1}_J \,|\, \boldsymbol{f}^V\right), \quad \text{(Risk-free strategy, with netting)}. \tag{2.29}$$

The only difference between (2.26)-(2.27) and (2.28)-(2.29) is that in the latter the risk-free strategy is used also for the risky portfolios. One could also think of other definitions, *e.g.*, using the risky strategies for both portfolios. This particular choice is motivated by the fact that $\boldsymbol{f}^U$ and $\boldsymbol{f}^A$ are, in general, dependent on $\boldsymbol{f}^V$ through the close-out agreements in (2.13) and (2.22). Moreover, $\boldsymbol{f}^V$ is a sub-optimal strategy for both risky portfolios leading to $\text{CVA} \le \overline{\text{CVA}}$, and $\text{CVA}^{\text{Net}} \le \overline{\text{CVA}}^{\text{Net}}$, which is beneficial for the bank (but certainly not for the counterparty). If we instead use only the risky descision functions, *i.e.*, replacing $\boldsymbol{f}^V$ with $\boldsymbol{f}^U$ in (2.28) and $\boldsymbol{f}^A$ in (2.29), we would obtain an underestimation of the CVAs, which would be unacceptable for the bank.

As mentioned in the Introduction, the bank is exposed to the risk of CVA losses, as a consequence of the MtM value of the CVA moving against the bank. We therefore want to follow the evolution of the CVA over time, to gain insights in its distribution. Of particular interest is the tail distribution of the CVA, for times between initial time and the maturity of the portfolio. To explore this, we define the dynamic versions of (2.26)-(2.29), which are stochastic processes depending on the market and portfolio state processes X and A. For $t \in [0, T]$, the dynamic versions of the CVAs (and their approximations) are given by the following random variables

$$\begin{aligned}
\text{CVA}(t, X_t, A_t) &\coloneqq \sum_{j=1}^{J} \left(\mathcal{V}_j\left(t, X_t \,|\, f_j^V\right) - \mathcal{U}_j\left(t, X_t \,|\, f_j^U\right)\right)(A_t)_j, \\
\text{CVA}^{\text{net}}\left(t, X_t, A_t\right) &\coloneqq \sum_{j=1}^{J} \mathcal{V}_j(t, X_t \,|\, f_j^V\big)(A_t)_j - \Upsilon^A\left(t, X_t, 1, A_t \,|\, \boldsymbol{f}^A\right), \\
\overline{\text{CVA}}(t, X_t, A_t) &\coloneqq \sum_{j=1}^{J} \left(\mathcal{V}_j\left(t, X_t \,|\, f_j^V\right) - \mathcal{U}_j\left(t, X_t \,|\, f_j^V\right)\right)(A_t)_j, \\
\overline{\text{CVA}}^{\text{net}}(t, X_t, A_t) &\coloneqq \sum_{j=1}^{J} \mathcal{V}_j(t, X_t \,|\, f_j^V\big)(A_t)_j - \Upsilon^A\left(t, X_t, 1, A_t \,|\, \boldsymbol{f}^V\right).
\end{aligned}$$

In the above, the CVA is conditional on that the counterparty has not defaulted prior to, or at, t (it does not make sense to calculate the CVA if the counterparty has already defaulted). From the above we can define the Expected value of the CVA (E-CVA), and for $\alpha \in (0, 1)$, the $\alpha-$level of Value at Risk of the CVA (VaR-CVA) and Expected Shortfall of the CVA (ES-CVA),

$$\text{E-CVA}(t) \coloneqq \mathbb{E}[\text{CVA}(t, X_t, A_t) \,|\, \mathbb{1}_t = 1], \tag{2.30}$$

$$\text{VaR-CVA}_\alpha(t) \coloneqq \inf\left\{P \in \mathbb{R} \,\middle|\, \mathbb{Q}\left(\text{CVA}(t, X_t, A_t) \le P\right) \ge \alpha\right\}, \tag{2.31}$$

$$\text{ES-CVA}_\alpha(t) \coloneqq \mathbb{E}\left[\text{CVA}(t, X_t, A_t) \,\middle|\, \mathbb{1}_t = 1\,,\ \text{CVA}(t, X_t, A_t) \ge \text{VaR-CVA}_\alpha(t)\right]. \tag{2.32}$$

In a similar way E-$\overline{\text{CVA}}(t)$, ES-$\overline{\text{CVA}}_\alpha(t)$, E-$\text{CVA}^{\text{net}}(t)$, ES-$\text{CVA}_\alpha^{\text{net}}(t)$, E-$\overline{\text{CVA}}^{\text{net}}(t)$ and

ES-$\overline{\text{CVA}}_\alpha^{\text{net}}(t)$ are defined. The expression for the ES-CVA looks complicated but is basically just the expected value of the $\alpha-$tail of the CVA distribution. We focus on ES-CVA instead of VaR-CVA because it is a coherent risk measure and VaR-CVA is not.

Remark 2.3. Since ES $-$ CVA is a non-traded risk measure, it should ideally be computed under the real world measure $\mathbb{P}$, see *e.g.*, [25] for a detailed discussion. To be precise, (X_t, A_t) should be generated under the $\mathbb{P}-$measure and, the CVA, which is a tradeable asset, should be computed under the $\mathbb{Q}-$measure. It is straightforward to adjust the algorithms in this paper be able to compute ES $-$ CVA under the $\mathbb{P}-$measure, see [2] for details in the special case $J = 1$.

2.5. Exposure profiles

In this subsection we discuss the concept of exposure profiles for a portfolio of derivatives. The financial exposure (of the bank) is defined as the maximum amount the bank stands to loose if the counterparty defaults. The exposure profile is loosely defined as the distribution of the exposure over time. The exposures, with and without netting, are defined as

$$\mathrm{E}_t^{\mathrm{Net}} \coloneqq \max\left\{\sum_{j=1}^{J} V_j(t, X_t)(A_t)_j,\ 0\right\}, \quad \mathrm{E}_t \coloneqq \sum_{j=1}^{J} \max\left\{V_j(t, X_t)(A_t)_j,\ 0\right\},$$

where we recall that $(A_t)_j = \mathbb{I}_{\{\tau_j > t\}}$ with τ_j being the exercise date for derivative j. Furthermore, for a portfolio without netting, the expected exposure (EE), and for $\alpha \in (0, 1)$, the potential future exposure (PFE) are defined as

$$\mathrm{EE}(t) \coloneqq \mathbb{E}_0\Big[D_{0,t}\,\mathrm{E}_t\Big], \tag{2.33}$$

$$\mathrm{PFE}_\alpha(t) \coloneqq \inf\Big\{P \in \mathbb{R} \,\Big|\, \mathbb{Q}\Big(D_{0,t}\,\mathrm{E}_t \le P\Big) \ge \alpha\Big\}. \tag{2.34}$$

Both the expectation and the probability in (2.33) and (2.34) should be interpreted as conditional on $X_0 = x_0 \in \mathbb{R}^d$. The EE and PFE in the presence of netting, denoted by $\mathrm{EE}^{\mathrm{Net}}(\cdot)$ and $\mathrm{PFE}_\alpha^{\mathrm{Net}}(\cdot)$, and are obtained by instead using the netted exposure in (2.33) and (2.34).

If we assume a constant recovery rate $R \in [0, 1)$, and that X and $\mathbb{1}^{\mathrm{D}}$ are independent, *i.e.*, the default event of the counterparty is independent of the risk factors, then (2.28) and (2.29) can be written as

$$\overline{\mathrm{CVA}} = (1 - R)\int_0^T \mathrm{EE}(t)\mathbb{Q}\Big(\tau^{\mathrm{D}} \in [t + \mathrm{d}t)\Big), \quad \overline{\mathrm{CVA}}^{\mathrm{Net}} = (1 - R)\int_0^T \mathrm{EE}^{\mathrm{Net}}(t)\mathbb{Q}\Big(\tau^{\mathrm{D}} \in [t + \mathrm{d}t)\Big),$$

which can be approximated by

$$\begin{aligned}\overline{\mathrm{CVA}} &\approx (1 - R)\sum_{m=1}^{M} \mathrm{EE}(t_m)\mathbb{Q}\Big(\tau^{\mathrm{D}} \in (t_{m-1}, t_m]\Big),\\ \overline{\mathrm{CVA}}^{\mathrm{Net}} &\approx (1 - R)\sum_{m=1}^{M} \mathrm{EE}^{\mathrm{Net}}(t_m)\mathbb{Q}\Big(\tau^{\mathrm{D}} \in (t_{m-1}, t_m]\Big),\end{aligned}$$

for some partition of $[0, T]$, with $t_0 = 0$ and $t_M = T$. The above formulations require access to the density of default events, but may be more accurate, especially for large M and a small probability of default (with a simulation based approach, problems with a low probability of default can often be tackled with variance reduction techniques).

3. Algorithms

In the first part of this section, we present a neural network-based method to approximate the decision functions introduced in the previous section. The method generalizes the Deep Optimal Stopping proposed in [1] and extended in [2], which approximates stopping decisions for a single derivative, to be applicable also for portfolios of derivatives with early-exercise features. Furthermore, for the risky portfolios, the algorithm is extended to be able to deal with default risk of the counterparty. The algorithm is based on a series of neural networks, which are optimized backwards in time with the objective to maximize the expected discounted cash-flows.

In the second part of this section, the exercise policy obtained from the approximate decision functions is applied pathwise on realizations of the risk factors of each derivative in the portfolio to generate pathwise cash-flows. These cash-flows are used in a neural network based regression algorithm to approximate pathwise derivative values. These pathwise derivative values can then be used to compute important risk management measures.

Due to the problem formulation which gives high dimensionality both in the underlying asset and in the number of derivatives in the portfolio, the notation may be somewhat difficult to follow. We therefore refer to [2] for a, perhaps, more straightforward introduction to the algorithms for a single derivative.

3.1. Phase i: Learning exercise strategy

As indicated above, the core of the algorithm is to approximate decision functions, in order to obtain good approximations of the value of a portfolio of derivatives. We approximate the decision functions $\boldsymbol{f}^V$, $\boldsymbol{f}^U$ and $\boldsymbol{f}^A$, with fully connected neural networks. To be more precise, let $N = |\mathbb{T}^{\Pi}(0)|$, for $n \in \{1, 2, \ldots, N\}$ and for $Z \in \{V, U\}$, the decision function $\boldsymbol{f}^Z(T_n, \cdot)$, is approximated by a fully connected neural network of the form $\boldsymbol{f}^{\theta_n} : \mathbb{R}^d \to \{0,\ 1\}^J$, where $\theta_n \in \mathbb{R}^{q_n}$ is a vector containing all the $q_n \in \mathbb{N}$ trainable parameters in network n. The decision function $\boldsymbol{f}^A(T_n, \cdot, \cdot)$ is approximated by similar neural networks, with the only difference that the input also includes information of which derivatives in the portfolio have been exercised prior to T_n, *i.e.*, $\boldsymbol{f}^{\theta_n} : \mathbb{R}^d \times \{0, 1\}^J \to \{0,\ 1\}^J$.

Since binary decision functions are discontinuous, and therefore unsuitable for gradient-type optimization algorithms, we use as an intermediate step, the neural network $\boldsymbol{F}^{\theta_n} : \mathbb{R}^d \to (0,\,1)^J$. Instead of a binary decision, the output of the neural network $\boldsymbol{F}^{\theta_n}$ can be viewed as the probability[4] for exercise to be optimal. This output is then mapped to 1 for values above (or equal to) 0.5, and to 0 otherwise, by defining $\boldsymbol{f}^{\theta_n}(\cdot) = \mathfrak{a} \circ \boldsymbol{F}^{\theta_n}(\cdot)$, where $\mathfrak{a}$ is a component-wise round-off function, *i.e.*, for $j \in \{1, 2, \dots, J\}$, and $x \in \mathbb{R}^d$, the j:th component of $\mathfrak{a}(x)$ is given by $(\mathfrak{a}(x))_j = \mathbb{I}_{\{x_j \geq 1/2\}}$. For each $Z \in \{V, U\}$, our aim is to adjust the parameters $\theta_1, \theta_2, \dots, \theta_N$ such that

$$(\boldsymbol{f}^Z(T_1, \cdot), \boldsymbol{f}^Z(T_2, \cdot), \dots, \boldsymbol{f}^Z(T_N, \cdot))^T \approx (\boldsymbol{f}^{\theta_1}, \boldsymbol{f}^{\theta_2}, \dots, \boldsymbol{f}^{\theta_N})^T =: \mathbb{f}^{\Theta}, \tag{3.1}$$

$$(\boldsymbol{f}^A(T_1, \cdot, \cdot), \boldsymbol{f}^A(T_2, \cdot, \cdot), \dots, \boldsymbol{f}^A(T_N, \cdot, \cdot)),^T \approx (\boldsymbol{f}^{\theta_1}, \boldsymbol{f}^{\theta_2}, \dots, \boldsymbol{f}^{\theta_N})^T =: \mathbb{f}^{\Theta}, \tag{3.2}$$

where we recall that $\Theta = \{\theta_1, \theta_2, \dots, \theta_N\}$. For $n \in \{1, 2, \dots, N\}$, we define the sequence of neural networks, approximating the decision functions at exercise dates $T_n, T_{n+1}, \dots T_N$, by $\mathbb{f}^{\Theta}_n := (\boldsymbol{f}^{\theta_n}, \boldsymbol{f}^{\theta_{n+1}}, \dots, \boldsymbol{f}^{\theta_N})^T$. Note that the input dimension for the neural networks is different when we want to approximate $\boldsymbol{f}^V$ and $\boldsymbol{f}^U$ compared to when we want to approximate $\boldsymbol{f}^A$. To avoid having to introduce an extra layer of notation, we use for all networks Θ to denote the set of parameters, and keep in mind that the dimension depends on the specific problem considered. Although the above provides a good intuition for what we want to accomplish, it is not clear in which sense we want the functions to be similar, or how to adjust the parameters to achieve this. To approach a more tractable form, from a computational perspective, for $t \in (T_{n-1}, T_n]$, we insert (3.1) in (2.10) and (3.2) in (2.20) to obtain

$$\boldsymbol{\tau}[\mathbb{f}^{\Theta}_n](X^{t,x}) = \sum_{k=n}^{N} T_k \boldsymbol{f}^{\theta_k}(X_k) \odot \prod_{m=k}^{N} \Big(\mathbf{1}_J - \boldsymbol{f}^{\theta_m}(X_m)\Big), \tag{3.3}$$

$$\boldsymbol{\tau}[\mathbb{f}^{\Theta}_n](X^{t,x}, A^{t,\alpha}) = \sum_{k=n}^{N} T_k \boldsymbol{f}^{\theta_k}(X_k, A_k) \odot \prod_{m=k}^{N} \Big(\mathbf{1}_J - \boldsymbol{f}^{\theta_m}(X_m, A_m)\Big). \tag{3.4}$$

Note that (3.3) is a $J-$dimensional vector of $X-$stopping times and (3.4) is a $J-$dimensional $(X, A)-$stopping times vector, which depends on $\mathbb{f}^{\Theta}_n$ on a structural level but also on the randomness of the stochastic process $X^{t,x}$ (and $A^{t,\alpha}$ for (3.4)). For notational convenience, we use the short hand notation $\bar{\boldsymbol{\tau}}^{\Theta}_n = \boldsymbol{\tau}[\mathbb{f}^{\Theta}_n](X^{t,x})$ (or $\hat{\boldsymbol{\tau}}^{\Theta}_n = \boldsymbol{\tau}[\mathbb{f}^{\Theta}_n](X^{t,x}, A^{t,\alpha})$, when approximating $\boldsymbol{f}^A$), and for element $j \in \{1, 2, \dots, J\}$, $\bar{\tau}^{\Theta}_{n,j} = (\bar{\boldsymbol{\tau}}^{\Theta}_n)_j$ (or $\hat{\tau}^{\Theta}_{n,j} = (\hat{\boldsymbol{\tau}}^{\Theta}_n)_j$). We are now ready to define our objective, which, for $T_n \in \mathbb{T}^{\Pi}(0)$, is to find θ_n such that the expected future cash-flows are maximized. The cash-flows can be divided into three categories:

1. The cash-flows obtained by the derivatives exercised at the present time T_n;
2. The cash-flows obtained at later exercise dates prior to default of the counterparty;
3. The cash-flows obtained at default of the counterparty, according to the close-out agreement.

For $n \in \{1, 2, \dots, N\}$ and $j \in \{1, 2, \dots, J\}$, we denote dimension j of decision function $\boldsymbol{f}^{\theta_n}$ by

$$\big(\boldsymbol{f}^{\theta_n}\big)_j = f^{\theta_n}_j, \quad \text{and } \big(\boldsymbol{F}^{\theta_n}\big)_j = F^{\theta_n}_j$$

Given that no default has occurred prior to T_n, (and the exercise state $A_n = \alpha$ for the risky portfolio with netting) the expected cash-flows, that we want to maximize, are given below.

Risk -free portfolio:

$$\mathbb{E}_{T_n}\Bigg[\sum_{j=1}^{J} f^{\theta_n}_j(X_n)\, g_j(T_n, X_n) + \Big(1 - f^{\theta_n}_j(X_n)\Big) D_{T_n, \bar{\tau}^V_{n+1,j}} g_j\Big(\bar{\tau}^V_{n+1,j}, X_{\bar{\tau}^V_{n+1,j}}\Big)\Bigg], \tag{3.5}$$

Risk y portfolio without netting:

$$\begin{aligned}\mathbb{E}_{T_n}\Bigg[& \textstyle\sum_{j=1}^{J} f^{\theta_n}_j(X_n)\, g_j(T_n, X_n) + \Big(1 - f^{\theta_n}(X_n)\Big)\Big(\mathbb{1}^{\mathrm{D}}_{\bar{\tau}^U_{n+1,j}} D_{T_n, \bar{\tau}^U_{n+1,j}} g_j\Big(\bar{\tau}^U_{n+1,j}, X_{\bar{\tau}^U_{n+1,j}}\Big) \\ & + \Big(1 - \mathbb{1}^{\mathrm{D}}_{\bar{\tau}^U_{n+1,j}}\Big) D_{t,\tau^{\mathrm{D}}}\Big(RV_j(\tau^{\mathrm{D}}, X_{\tau^{\mathrm{D}}})^+ + V_j(\tau^{\mathrm{D}}, X_{\tau^{\mathrm{D}}})^-\Big)\Big)\Bigg],\end{aligned} \tag{3.6}$$

[4] However the interpretation as a probability may be helpful, one should be careful since it is not a rigorous mathematical statement. It should be clear that there is nothing random about the stopping decisions, since the stopping time is $\mathcal{H}_t-$measurable. It can also be interpreted as a measure on how certain we can be that exercise is optimal.

Risk y portfolio with netting:

$$\begin{aligned}\mathbb{E}_{T_n}\Bigg[& \textstyle\sum_{j=1}^{J}\alpha_j f_j^{\theta_n}(X_n,\alpha)\,g_j(T_n,X_n)+\alpha_j\big(1-f_j^{\theta_n}(X_n,\alpha)\big)\mathbb{1}^{\mathrm{D}}_{\hat{\tau}^U_{n+1,j}}D_{T_n,\hat{\tau}^U_{n+1,j}}g_j\Big(\hat{\tau}^U_{n+1,j},X_{\hat{\tau}^U_{n+1,j}}\Big)\\ & +R\Big(\textstyle\sum_{k=1}^{J}\alpha_k\big(1-f_k^{\theta_n}(X_n,\alpha)\big)\Big(1-\mathbb{1}^{\mathrm{D}}_{\hat{\tau}^U_{n+1,k}}\Big)D_{t,\tau^{\mathrm{D}}}V_k(\tau^{\mathrm{D}},X_{\tau^{\mathrm{D}}})\Big)^{+}\\ & +\Big(\textstyle\sum_{\ell=1}^{J}\alpha_\ell\big(1-f_\ell^{\theta_n}(X_n,\alpha)\big)\Big(1-\mathbb{1}^{\mathrm{D}}_{\hat{\tau}^U_{n+1,\ell}}\Big)D_{t,\tau^{\mathrm{D}}}V_\ell(\tau^{\mathrm{D}},X_{\tau^{\mathrm{D}}})\Big)^{-}\Bigg].\end{aligned} \tag{3.7}$$

We want to optimize θ_n, such that the above are as close as possible (in mean squared sense) to $\Pi^V(T_n,X_n)$, $\Pi^U(T_n,X_n,1)$ and $\Pi^A(T_n,X_n,1,\alpha)$, respectively.

Remark 3.1. The objectives for the risky portfolios, in (3.6) and (3.7), both depend on the risk-free valuation of the derivatives. Therefore, in order to approximate the risky decision functions, we first need to approximate the risk-free exercise strategy, and the risk-free derivative values. In the next subsection, we explain how V_j can be approximated.

Although (3.5)-(3.7) are accurate representations of the optimization problems, they give us some practical problems. In general, we have no access to $\boldsymbol{f}^V$, $\boldsymbol{f}^U$ and $\boldsymbol{f}^A$ which control $\bar{\boldsymbol{\tau}}^V_{n+1}$, $\bar{\boldsymbol{\tau}}^U_{n+1}$ and $\hat{\boldsymbol{\tau}}^A_{n+1}$. Another problem is that, in general, we have no access to the true distributions of the portfolio values for comparison. However, if $T_{\tilde{N}}$ is the maturity of derivative $j\in\{1,2,\dots,J\}$, it is optimal to exercise as long as the pay-off value is positive, by the definition of the decision functions. We can therefore set

$$f_j^{\theta_{\tilde{N}}}(\,\cdot\,)\coloneqq\mathbb{I}_{\{g_j(T_{\tilde{N}},\,\cdot\,)>0\}},\quad\text{ and }\quad f_j^{\theta_k}(\,\cdot\,)\coloneqq 0,\ \text{ for } k>\tilde{N}.$$

Furthermore, at T_N, the maturity of the portfolio, the positive part of the pay-off value equals the derivative value (if no default in $(T_{N-1},T_N]$, in the risky cases). At T_{N-1}, Equation (3.5) then becomes

$$\mathbb{E}_{T_{N-1}}\left[\sum_{j=1}^{J}f_j^{\theta_{N-1}}(X_N)\,g_j(T_{N-1},X_{N-1})+D_{N-1,N}\Big(1-f_j^{\theta_{N-1}}(X_{N-1})\Big)\,g_j(T_N,X_N)\right]. \tag{3.8}$$

Recall that if T_N is greater than the maturity of contract j, we have $g_j(T_N,\cdot)\equiv 0$. Since all components in (3.8) are known except for the decision function $\boldsymbol{f}^{\theta_{N-1}}$, we want to find θ_{N-1}, such that a Monte-Carlo approximations of (3.8) is maximized. Given $M\in\mathbb{N}$ samples, distributed as X, which for $m\in\{1,2,\dots,M\}$ is denoted by $x=(x_t(m))_{t\in[0,T]}$, we approximate (3.8) by

$$\frac{1}{M}\sum_{m=1}^{M}\sum_{j=1}^{J}f_j^{\theta_{N-1}}(x_{N-1}(m))\,g_j(T_{N-1},x_{N-1}(m))+D_{N-1,N}\Big(1-f_j^{\theta_{N-1}}(x_{N-1}(m))\Big)\,g_j(T_N,x_N(m)). \tag{3.9}$$

The only unknown entity in (3.9) is the parameter θ_{N-1} in the decision function $\boldsymbol{f}^{\theta_{N-1}}=(f_1^{\theta_{N-1}},\dots,f_J^{\theta_{N-1}})^T$. Furthermore, we wish to find θ_{N-1} such that (3.9) is maximized, since it represents the average cash-flow in $[t_{N-1},t_N]$. Once θ_{N-1} is optimized, we use this parameter to set up a similar expression for the expected cash-flow on $[t_{N-2},t_N]$, which is maximized by finding an optimal θ_{N-2}. This procedure is then iteratively continued until also $\theta_{N-3},\theta_{N-4},\dots,\theta_1$ are optimized. The procedure is similar for the risky portfolios, but based on (3.6) or (3.7) instead. This implies that we also need to sample default events of the counterparty. We denote by θ^*_n the optimized version of parameter θ_n and the sequence of optimized parameters for the networks at exercise dates $T_n,T_{n+1},\dots,T_N$ are defined as

$$\Theta^*_n\coloneqq\{\theta^*_n,\theta^*_{n+1},\dots,\theta^*_N\},$$

and for notational convenience, we define the complete sequence of parameters as $\Theta^*\coloneqq\Theta^*_1$.

Remark 3.2. Since we are considering a portfolio in which all the derivatives may have a different set of exercise dates, we have that for $T_n\in\mathbb{T}^{\Pi}(0)$, there are $J^{Ex}_n\in\{1,2,\dots,J\}$ derivatives that may be exercised. Therefore, we only need to compute J^{Ex}_n of the J dimensions of $\boldsymbol{f}^{\theta_n}$ and can by default set the remaining $J-J^{Ex}_n$ dimensions of $\boldsymbol{f}^{\theta_n}$ to 0. This can be done by appropriately adjusting some weights and biases.

To keep the flow of the paper, the details of the algorithms and the parameters θ_n are given in the Appendix.

3.2. Phase II: Learning pathwise derivative values and portfolio exposures

As mentioned in Remark 3.1, the risk-free derivative values need to be approximated pathwise in order to approximate the risky decision functions. Moreover, the pathwise derivative values are required to approximate the exposure profiles for the risk-free, as well as the risky portfolios. In this subsection, we focus on the risk-free portfolios, but the extension to risky portfolios is straightforward.

We use the risk-free, stopping strategy from subsection 3.1 to generate pathwise cash-flows, which are in turn used to approximate the pathwise derivative values. Let $T_{n-1}, T_n \in \mathbb{T}^{\Pi}(0)$, for $t \in (T_{n-1}, T_n]$, we denote the vector-valued discounting process and pay-off function, respectively, by

$$\boldsymbol{D}(t, \boldsymbol{\tau}_n^V) \coloneqq \begin{pmatrix} D(t, \tau_{n,1}^V) \\ \vdots \\ D(t, \tau_{n,J}^V) \end{pmatrix} \quad \text{and,} \quad \boldsymbol{g}(\boldsymbol{\tau}_n^V, X^{t,x}) \coloneqq \begin{pmatrix} g_1(\tau_{n,1}^V, X_{\tau_{n,1}^V}) \\ \vdots \\ g_J(\tau_{n,J}^V, X_{\tau_{n,J}^V}) \end{pmatrix}. \tag{3.10}$$

Using the notation above, we define the vector-valued cash-flow process as

$$\boldsymbol{Y}_t \coloneqq \boldsymbol{D}_{t,\boldsymbol{\tau}_n^V} \odot \boldsymbol{g}(\boldsymbol{\tau}_n^V, X_{\boldsymbol{\tau}_n^V}), \tag{3.11}$$

and for $j \in \{1, 2, \dots, J\}$, we denote the j:th element of $\boldsymbol{Y}_t$ by $Y_{t,j}$, and we emphasize that $\boldsymbol{Y}_t$ is not $\mathcal{H}_t-$measurable.

3.2.1. Regression problems

In this subsection we use standard regression theory, see *e.g.*, [26], to show that derivative values, exposures, and other entities of interest, can be formulated as the solution to certain minimization problems. We introduce the following notation for measurable functions[5]

$$\mathcal{D}(C_1; C_2) \coloneqq \{ f : C_1 \to C_2 \mid f \text{ measurable}\}. \tag{3.12}$$

First, we recall a basic property of the regression function. Let $\mathcal{X} : [0, T] \times \Omega \to C_1$ and $\mathcal{Y} : [0, T] \times \Omega \to C_2$ be a stochastic process, which for $t, u \in [0, T]$, with $t \le u$, satisfies $\mathbb{E}_0[|\mathcal{Y}_t|^2] < \infty$. We define the regression function, which satisfies

$$m(t, \cdot) \in \operatorname*{arg\,min}_{\boldsymbol{h} \in \mathcal{D}(C_1;C_2)} \mathbb{E}_t[\|\boldsymbol{h}(\mathcal{X}_t) - \mathcal{Y}_u]\|_2^2], \tag{3.13}$$

where $\|\cdot\|_2$ is the Euclidean norm. It then holds, for $x \in \mathbb{R}^a$, that the regression function is given by the conditional expectation

$$m(t, \chi) = \mathbb{E}^{\mathbb{Q}}\big[\mathcal{Y}_u \,\big|\, \mathcal{X}_t = \chi\big]. \tag{3.14}$$

Using the notation from above, and by choosing C_1, C_2, $\mathcal{X}$ and $\mathcal{Y}$ in (3.13) and (3.14) wisely, we can approximate different entities related to *e.g.*, the exposure profiles or the pathwise CVA. For instance the exposures, both with and without netting, can be approximated from the solution of a minimization problem of the form in (3.13). For $T_{n-1}, T_n \in \mathbb{T}^{\Pi}(0)$, let $t \in (T_{n-1}, T_n]$ and denote $\boldsymbol{V}(t, \cdot) = (V_1(t, \cdot), \dots V_J(t, \cdot))^T$, it then holds that

$$\boldsymbol{V}(t, \cdot) \in \operatorname*{arg\,min}_{\boldsymbol{h} \in \mathcal{D}(\mathbb{R}^d;\mathbb{R}^J)} \mathbb{E}_t\Big[\|\boldsymbol{h}(X_t) - \boldsymbol{Y}_t\|_2^2\Big], \tag{3.15}$$

$$\sum_{j=1}^{J} V_j(t, \cdot)(A_t)_j \in \operatorname*{arg\,min}_{h \in \mathcal{D}(\mathbb{R}^d \times \{0,1\}^J;\mathbb{R})} \mathbb{E}_t\left[\left| h(X_t, A_t) - \sum_{j=1}^{J} Y_{t,j}(A_t)_j \right|^2\right]. \tag{3.16}$$

In (3.15), we have $C_1 = \mathbb{R}^d$, $C_2 = \mathbb{R}^J$, $\mathcal{X} = X$ and $\mathcal{Y} = \boldsymbol{Y}$ and in (3.16), we have $C_1 = \mathbb{R}^d \times \{0, 1\}^J$, $C_2 = \mathbb{R}$, $\mathcal{X} = (X, A)$ and $\mathcal{Y} = \sum_{j=1}^{J} Y_{\cdot,j}(A)_j$. For $s \in [0, T]$ and for $j \in \{1, 2, \dots, J\}$, by (2.2), it holds that $\mathbb{E}_0[|Y_{s,j}|^2] < \infty$, and therefore also $\mathbb{E}_0[\|\boldsymbol{Y}_s\|_2^2] < \infty$. Now, (3.15) holds trivially since by definition $\boldsymbol{V}(t, x) = \mathbb{E}_{t,x}[\boldsymbol{D}_{t,\boldsymbol{\tau}_n^V} \odot \boldsymbol{g}(\boldsymbol{\tau}_n^V, X_{\boldsymbol{\tau}_n^V})] = \mathbb{E}_{t,x}[\boldsymbol{Y}_t]$. For (3.16), it follows that

$$\sum_{j=1}^{J} V_j(t, x)(A_t)_j = \sum_{j=1}^{J} \mathbb{E}_{t,x}\Big[D_{t,\tau_{n,j}^V} g_j(\tau_{n,j}^V, X_{\tau_{n,j}^V})\Big](A_t)_j \tag{3.17}$$

$$= \mathbb{E}_{t,x}\left[\sum_{j=1}^{J} D_{t,\tau_{n,j}^V} g_j(\tau_{n,j}^V, X_{\tau_{n,j}^V})(A_t)_j\right] = \mathbb{E}_{t,x}\left[\sum_{j=1}^{J} Y_{t,j}(A_t)_j\right], \tag{3.18}$$

where we have used linearity of expectations and the fact that A_t is $\mathcal{H}_t-$measurable.

3.3. Neural network based regression algorithm

Since the specific details of the neural networks are transferable from A.1 and A.2, this subsection is less detailed. The main idea is to represent $\mathcal{D}(\mathbb{R}^A; \mathbb{R}^b)$ (measurable functions from $\mathbb{R}^A$ to $\mathbb{R}^b$, defined in (3.12)) by a parametrized neural network. A minimization problem of the form (3.13) can then be used as a loss function, which should be minimized by adjusting some set of trainable parameters. However, in general we have no access to $\boldsymbol{\tau}_n^V$ for $n < N$, where $N = |\mathbb{T}^{\Pi}(0)|$.

[5] We assume measurable spaces $(C_1, \mathcal{C}_1)$ and $(C_2, \mathcal{C}_2)$ and measurable functions with respect to σ-algebras $\mathcal{C}_1$ and $\mathcal{C}_2$. This assumption holds for all cases in this paper.

On the other hand, we can use the exercise strategy from Subsection 3.1, *i.e.*, approximate $\boldsymbol{\tau}_n^V$ by $\bar{\boldsymbol{\tau}}_n^{\Theta^*} = \left(\bar{\tau}_{n,1}^{\Theta^*}, \ldots, \bar{\tau}_{n,J}^{\Theta^*}\right)^T$. Furthermore, $\mathbb{E}_t[\cdot]$ in (3.14) needs to be approximated by Monte-Carlo samples. We use $M_{\text{reg}} \in \mathbb{N}$ samples, distributed as X and $\boldsymbol{Y}$, which for $m \in \{1, 2, \ldots, M_{\text{reg}}\}$ are denoted by[6] $x^{\text{reg}}(m) = (x_t^{\text{reg}}(m))_{t\in[0,T]}$ and $\boldsymbol{y}(m) = (\boldsymbol{y}_t(m))_{t\in[0,T]}$. Furthermore, we use

$$A_t(m) \approx A_t^{\Theta_n^*}(m) = \begin{pmatrix} \mathbb{I}_{\left\{\bar{\tau}_{n,1}^{\Theta^*}(m)>t\right\}}, \\ \vdots \\ \mathbb{I}_{\left\{\bar{\tau}_{n,J}^{\Theta^*}(m)>t\right\}} \end{pmatrix}$$

where for $j \in \{1, 2, \ldots, J\}$, $\bar{\tau}_{n,j}^{\Theta^*}(m) = (\boldsymbol{\tau}[\mathbb{f}_n^{\Theta^*}](x^{t,x_t^{\text{reg}}(m)}(m)))_j$.

We define for $n \in \{1, 2, \ldots, N\}$, the neural networks $\boldsymbol{h}^{\Phi_n} : \mathbb{R}^d \to \mathbb{R}^J$ and $h^{\Phi_n} : \mathbb{R}^d \times \{0, 1\}^J \to \mathbb{R}$, which are parametrized by $\Phi_n^{\text{IR}} \in \mathbb{R}^{u_n^{\text{IR}}}$ and $\Phi_n^{\text{PR}} \in \mathbb{R}^{u_n^{\text{PR}}}$ where $u_n^{\text{IR}}, u_n^{\text{PR}} \in \mathbb{N}$ are the number of trainable parameters in each network. We use as loss functions, the empirical counterparts of (3.15) and (3.16), which are given by

$$\text{DOS-IR:} \qquad \frac{1}{M_{\text{reg}}} \sum_{m=1}^{M_{\text{reg}}} \|\boldsymbol{h}^{\Phi_n^1}(x_t^{\text{reg}}(m)) - \boldsymbol{y}_t(m)\|_2^2, \tag{3.19}$$

$$\text{DOS-PR:} \qquad \frac{1}{M_{\text{reg}}} \sum_{m=1}^{M_{\text{reg}}} \left| h^{\Phi_n^2}\left(x_t^{\text{reg}}(m), A_t^{\Theta_n^*}(m)\right) - \sum_{j=1}^{J} y_{t,j}(m)\left(A_t^{\Theta_n^*}(m)\right)_j \right|^2. \tag{3.20}$$

”DOS” in DOS-IR and DOS-PR refers to the fact that the deep stopping strategy used to obtain $\boldsymbol{y}(m)$ is generated by the DOS-algorithm. ’IR’ and ’PR’ are abbreviations for ”individual regression” and ”portfolio regression”, and refer to the regression at the level of each individual derivative, and the regression at portfolio level given in (3.19) and in (3.20), respectively.

The exact algorithm for computations of pathwise CVA in order to obtain ES-CVA is not presented in details here. However, it is straightforward to adjust (3.19) and (3.20), to approximate pathwise CVA instead.

The only important adjustment to the structure of the neural networks (details in A.1) is that we want the output to be unbounded and therefore use the identity as scalar activation function in the output layers.

3.4. Combining phase i and phase II

Recall that the purpose for using regression at the level of each derivative was to be able to approximate the exposure of a portfolio of derivatives without netting agreement. If we consider derivatives with non-negative value, the definitions of exposures with and without netting agreements coincide. Therefore, only derivatives with non-negative values are considered in this paper, to be able to compare regression on derivative level with the regression on portfolio level. For $T_{n-1}, T_n \in \mathbb{T}^{\Pi}(0)$, let $t \in (T_{n-1}, T_n]$, we define the following approximators

$$\boldsymbol{V}^{\text{DOS-IR}}\left(t, \cdot \,\middle|\, \Phi_n^{\text{IR}}, \Theta^*\right) \coloneqq \boldsymbol{h}^{\Phi_n^{\text{IR}}}(\cdot), \quad \text{(individual derivative values)}, \tag{3.21}$$

$$\mathbf{E}^{\text{DOS-IR}}\left(t, \cdot \,\middle|\, \Phi_n^{\text{IR}}, \Theta^*\right) \coloneqq \boldsymbol{h}^{\Phi_n^{\text{IR}}}(\cdot) \odot A_t^{\Theta^*}, \quad \text{(derivative exposures)}, \tag{3.22}$$

$$\text{E}^{\text{DOS-IR}}\left(t, \cdot \,\middle|\, \Phi_n^{\text{IR}}, \Theta^*\right) \coloneqq \sum_{j=1}^{J} \left(\boldsymbol{h}^{\Phi_n^{\text{IR}}}(\cdot)\right)_j (A_t^{\Theta^*})_j, \quad \text{(portfolio exposure)}, \tag{3.23}$$

$$\text{E}^{\text{DOS-PR}}\left(t, \cdot \,\middle|\, \Phi_n^{\text{PR}}, \Theta^*\right) \coloneqq h^{\Phi_n^{\text{PR}}}(\cdot, A_t^{\Theta^*}), \quad \text{(portfolio exposure)}, \tag{3.24}$$

where Φ_n^{IR}, and Φ_n^{PR} are parameters optimized by minimizing (3.19) and (3.20), respectively, and Θ^* are parameters optimized according to the procedure described in the training procedure, described in Phase I, and $\mathbb{I}_t \in \{0, 1\}^J$ represents the exercise history of each derivative in the portfolio. Note that, even though Θ^* does not appear explicitly in the right hand side of (3.21), it is crucial since the cash-flow vector $\boldsymbol{y}$, used in (3.19) and (3.20), is created by applying an exercise strategy controlled by Θ^*. The approximations of EE and PFE are constructed from $M_{\text{train}} \in \mathbb{N}$ independent realizations of X, which for $m \in \{1,\ 2,\ \ldots,\ M\}$ are denoted by $(x_t^{\text{train}}(m))_{t\in[0,T]}$. For $z \in \{\text{IR}, \text{PR}\}$, the approximators are given by

$$\widehat{\text{EE}}^{\text{DOS-}z}(t) \coloneqq \sum_{m=1}^{M} \Pi^{\text{DOS-}z}\left(t, x(m) \,\middle|\, \Phi_n^z, \Theta^*\right), \tag{3.25}$$

[6] In practice we set $x_{\text{reg}} = x_{\text{val}}$, where x_{val} is defined in Phase I.

$$\widehat{\mathrm{PFE}}_{\alpha}^{\text{DOS-}z}(t) \coloneqq \Pi^{\text{DOS-}z}\Big(t, x(i_\alpha) \,\Big|\, \Phi_n^z, \Theta^*\Big), \tag{3.26}$$

where i_α is the index of the empirical $\alpha-$percentile of the vector $(\Pi^{\text{DOS-}z}(t, x(1) \,\big|\, \Phi_n^z, \Theta^*), \dots, \Pi^{\text{DOS-}z}(t, x(M) \,\big|\, \Phi_n^z, \Theta^*))$.

4. Numerical experiments

In the numerical experiments we use a Geometric Brownian Motion (GBM) to model the asset processes and an intensity model for default events of the counterparty. To be able to incorporate WWR, the default intensity is linked to the market state of the asset processes. The default event is triggered by an exogenous component, independent of observable market information. On the other hand, the intensity depends on the credit spread of the counterparty (observable from zero-coupon bonds) as well as a WWR-parameter. Our model choices are not necessarily used in practice but they serve the purpose of being easy to analyse. Especially the default model makes it straightforward to analyze the effects of the credit spread of the counterparty and the WWR-parmeter. It should be pointed out that the algorithms described in this paper are model independent in the sense that they are fully data driven. This means that as long as we can sample (or in any other way obtain) market data and default events, we can train the neural networks and the computations below can be performed.

In addition, the algorithms have also been implemented for a portfolio of Bermudan swaptions with dynamics following the one-factor Hull–White model. The results are similar, and are therefore not included in this section.

4.1. Risk-factor model

In the Black–Scholes framework, the assets are described by a $\mathbb{N}_+ \ni d-$dimensional Geometric Brownian. For $t \in [0, T]$, with constant risk-free rate $r \in \mathbb{R}$, initial state $s_{0}, \in (0, \infty)^d$, constant dividend $q \in (0, \infty)^d$ and volatility $\sigma \in (0, \infty)^d$, component $i \in \{1, 2, \dots, d\}$ of the asset process $S = (S_t)_{t\in[0,T]}$ is given by

$$(S_t)_i = (s_0)_i \exp\bigg(\Big(r - q_i - \frac{\sigma_i^2}{2}\Big)t + \sigma_i (W_t)_i\bigg), \tag{4.1}$$

where $W = (W_t)_{t\in[0,T]}$ is a correlated standard Brownian motion, satisfying for $i, j \in \{1, 2, \dots, d\}$, $\mathbb{E}_0[\mathrm{d}(W_t)_i \mathrm{d}(W_t)_j] = \rho_{ij}\mathrm{d}t$, with $\rho_{ij} \in [-1, 1]$.

4.2. Default model

Following [27], we model a default event of the counterparty as

$$\tau^{\mathrm{D}} = \inf_{t\in[0,T]} \bigg\{ t : \int_0^t \tilde{h}(u, \tilde{S}_u)\mathrm{d}u \geq E_1 \bigg\},$$

where E_1 is a random variable, uniformely distributed on [0,1]. Furthermore, the process $(\tilde{S}_t)_{t\in[0,T]}$ is the geometric average of the d component of the dividend-free version of S, given by

$$\tilde{S}_t = \prod_{i=1}^d \bigg((s_0)_i \exp\bigg(\Big(r - \frac{\sigma_i^2}{2}\Big)t + \sigma_i (W_t)_i\bigg)\bigg)^{1/d}.$$

For simplicity, without loss of generality, from now on, we assume no correlation between the components of the Brownian motions, i.e., $\rho_{ij} = 0$ for $i \neq j$. The above is a one-dimensional GBM, which can be written as

$$\tilde{S}_t = \tilde{s}_0 \exp\bigg(\Big(\tilde{\mu} - \frac{\tilde{\sigma}^2}{2}\Big)t + \tilde{\sigma}\tilde{W}_t\bigg),$$

where $\tilde{s}_0 = \Big(\prod_{i=1}^d (s_0)_i\Big)^{1/d}$, $\tilde{\sigma} = \frac{1}{d}\Big(\sum_{i=1}^d \sigma_i^2\Big)^{1/2}$, $\tilde{\mu} = r - \frac{1}{2d}\big(1 - \frac{1}{d}\big)\sum_{i=1}^d \sigma_i^2$ and $\tilde{W}_t = \frac{1}{d\tilde{\sigma}}\sum_{i=1}^d \sigma_i (W_t)_i$. For $(t, x) \in [0, T] \times \mathbb{R}_+$, $\tilde{h}$ is of the form $\tilde{h}(t, x) = c(t) + b\log x$. It can be checked that $\tilde{W} = (\tilde{W}_t)_{t\in[0,T]}$, is a 1-dimensional standard Brownian motion. By setting

$$c(t) = \bar{h} + b\log\tilde{S}_0 - \big(r - \frac{\tilde{\sigma}^2}{2}\big)bt + \frac{1}{2}b^2\tilde{\sigma}^2 t^2,$$

we obtain

$$\tilde{h}_t = \tilde{h}(t, \tilde{S}_t) = \bar{h} + \frac{1}{2}\tilde{\sigma}^2 t^2 b^2 + b\tilde{\sigma}\tilde{W}_t.$$

The economic interpretation of the above is that $\bar{h}$ is the credit spread for the counterparty and b controls the wrong way risk (WWR).

Recall that the jump-to-default process, $\mathbb{1}^{\mathrm{D}}$, is given by $\mathbb{1}_t^{\mathrm{D}} = \mathbb{I}_{\{t<\tau^{\mathrm{D}}\}}$ which gives a survival probability $G_t = \mathbb{E}^{\mathbb{Q}}[\mathbb{1}_t^{\mathrm{D}} | \mathcal{H}_t] = \exp\Big(-\int_0^t \tilde{h}_s \mathrm{d}s\Big)$ (for details, see [27]).

Table 2
The two components of the asset process, S, are represented by $x_1, x_2 \in \mathbb{R}$ and $(\cdot)^+ = \max\{\cdot, 0\}$.

Contract number	Contract name	Pay-off function
j=1	Max-call option	$(\max\{x_1, x_2\} - 100)^+$
j=2	Max-put option	$(100 - \max\{x_1, x_2\})^+$
j=3	Geometric-average-call option	$(\sqrt{x_1 x_2} - 100)^+$
j=4	Geometric-average-put option	$(100 - \sqrt{x_1 x_2})^+$
j=5	Arithmetic-average-call option	$(\frac{1}{2}(x_1 + x_2) - 100)^+$
j=6	Arithmetic-average-put option	$(100 - \frac{1}{2}(x_1 + x_2))^+$
j=7	1d-call option	$(x_1 - 100)^+$
j=8	1d-put option	$(100 - x_1)^+$

Table 3
Valuation at the level of each derivative with the portfolio version of DOS, and the SGBM. The reference solution is computed by a binomial lattice model in [15].

	j=1	**j=2**	**j=3**	**j=4**	**j=5**	**j=6**	**j=7**	**j=8**	$\Pi(0, s_0)$
Ref.	13.902	NA	NA	NA	NA	NA	NA	NA	NA
DOS	13.902	9.520	4.363	16.770	4.919	15.313	7.965	18.021	90.773
SGBM	13.899	9.780	4.366	16.770	4.971	15.327	7.963	18.033	91.108

4.3. Experiments

Contract details:

We consider a portfolio of $J = 8$ derivatives, depending on an asset process in $d = 2$ dimensions. We set $T = 3$ and use for each derivative, $j \in \{1, 2, \dots, 8\}$, the set of exercise dates $\mathbb{T}_j(t) = \mathbb{T}_j = \{0, \frac{1}{3}, \frac{2}{3}, 1, \frac{4}{3}, \frac{5}{3}, 2, \frac{7}{3}, \frac{8}{3}, 3\}$, and the pay–off functions given in Table 2.

Dynamics details:

For $i \in \{1, 2\}$, we set $(s_0)_i = 100$, $q_i = 0.1$, $r = 0.05$, $\sigma_i = 0.2$, $\rho_{ii} = 1$ and $\rho_{12} = \rho_{21} = 0$.

Neural network details:

We use $M_{\text{train}} = M_{\text{reg}} = M_{\text{reg}} = M = 2^{20}$, and for simplicity, the same structure and hyperparameter-settings are used in all networks, *i.e.,* all the networks in Phase I, and Phase II. We use training batches of size 5000, 3 hidden layers and 30 nodes in each hidden layer. Furthermore, the learning rate decreases step-wise, with equally sized steps after 100 training batches from 10^{-2} to 10^{-6}. For a discussion on the convergence of the training of the neural networks in terms of the number of training paths, M_{train}, in the case $J = 1$, we refer to [2].

4.3.1. Risk-free valuation

The purpose of the risk-free valuation is two-fold. Firstly, since the risky derivative values are used as an input in the risky-valuations, they need to be accurate. To ensure accuracy, the risk-free values are compared to a well-established existing valuation method, namely the Stochastic Grid Bundling Method (SGBM), see [19] for details. Secondly, it is in its own, an interesting and challenging problem to compute the value of a portfolio of complex derivatives with early-exercise features, without having to do one computation per derivative.

As mentioned above, we compare the algorithms introduced in earlier sections with the SGBM. We emphasize that the values for each derivative needs to be approximated individually when using SGBM, *i.e.,* we perform 8 regressions, one for each derivative. In Table 3, we compare the value at $t = 0$ for each derivative approximated with the portfolio version of the DOS-algorithm and the SGBM. In Table 3, we compare our values for each derivative at the initial time with the values obtained by SGBM. For the DOS-values, the neural network is trained five times, and evaluated on new, independent, samples and the average values are reported. For the SGBM, the regression is performed five times for each derivative, and the average values of the direct estimators are reported. It should be mentioned that, for $j \in \{3, 4, 5, 6, 7, 8\}$, the SGBM-values are biased high[7] and, for all j, the DOS-values are biased low.

In Fig. 1 to the left we compare EE, $\text{PFE}_{97.5}$ and $\text{PFE}_{2.5}$ approximated with SGBM and according to (3.25) and (3.26). To the right, we compare the derivative-wise EE approximated with SGBM and the DOS-IR based algorithm. In the DOS-IR based algorithm, the EE is approximated by evaluating (3.22) at $x(1), \dots, x(M)$ and computing the component-wise sample mean.

We conclude that, although very different in nature, the SGBM and the two methods presented in this paper, agree on the values at time 0, the tail-distribution of the portfolio exposure over time (at least at the 97.5 and 2.5 percentiles), and the average values of each derivative over time. We emphasize, that we do not claim that one of the methods perform

[7] The reason that the SGBM-values are not biased high for $j \in \{1, 2\}$ can be explained by the non-linearity of the max-operator in the pay-off functions. For more details we refer to [19].

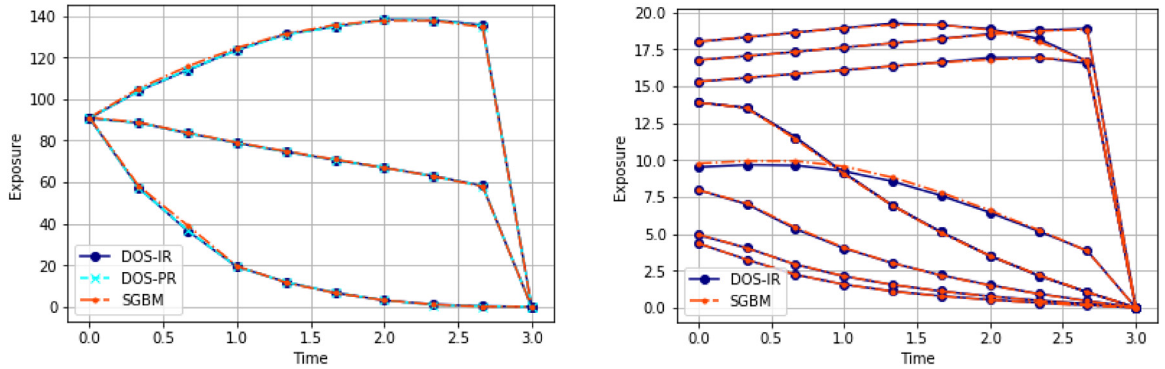


Fig. 1. **Left:** EE (lines in the middle), $\mathrm{PFE}_{97.5}$ (lines in the upper part) and $\mathrm{PFE}_{2.5}$ (lines in the lower part) at portfolio level computed with DOS-PR, DOS-IR and SGBM respectively. **Right:** EE at derivative level computed with DOS-IR and SGBM, respectively.

Table 4
Portfolio valuation with and without netting for different early-exercise policies. We use the short hand notations $\Upsilon^V[\boldsymbol{f}] = \Upsilon^V(0, s_0 \mid \boldsymbol{f})$, $\Upsilon^U[\boldsymbol{f}] = \Upsilon^U(0, s_0, 1 \mid \boldsymbol{f})$ and $\Upsilon^A[\boldsymbol{f}] = \Upsilon^A(0, s_0, 1, \mathbf{1}_9 \mid \boldsymbol{f})$. Furthermore, in this table $\boldsymbol{f}^V$, $\boldsymbol{f}^U$ and $\boldsymbol{f}^A$ represent our neural network approximations of the optimal decision functions.

	$\bar{h}$	$\Upsilon^V[\boldsymbol{f}^V]$	$\Upsilon^U[\boldsymbol{f}^U]$	$\Upsilon^U[\boldsymbol{f}^V]$	$\Upsilon^A[\boldsymbol{f}^A]$	$\Upsilon^A[\boldsymbol{f}^V]$
	0	78.62	77.47	77.41	77.86	77.84
$b = -0.2$	0.1	78.62	59.07	57.22	69.20	67.72
	0.2	78.62	47.58	42.99	64.17	60.82
	0	78.62	78.62	78.62	78.62	78.62
$b = 0$	0.1	78.62	59.50	58.00	69.55	68.39
	0.2	78.62	47.83	43.58	64.37	61.32
	0	78.62	78.28	78.27	78.59	78.58
$b = 0.2$	0.1	78.62	59.65	58.31	69.78	68.75
	0.2	78.62	47.89	43.78	64.51	61.55

better than the others. For an analysis of the difference in performance between the methods (in the special case $J = 1$), we refer to [2].

4.3.2. Risky valuation

In this subsection we focus on the impact of the exercise policy for CVA computations. To be more precise, we investigate to what extent the CVA is overestimated when using the risk-free exercise policy, with and without netting, for different levels of WWR and credit quality of the counterparty. The contracts described in Table 2 all have positive pay-offs meaning that the corresponding derivative values are also positive. This eliminates the netting effect, and therefore, we add a derivative to the portfolio, which may take on negative values. The 9:th derivative is a European-type future with pay-off $g_9(t, S_t) = 2 \times \big(80 - (S_t)_1\big)\mathbb{I}_{\{t=T\}}$. By the martingale property we have that

$$V_9(t, S_t) = \mathbb{E}_t\Big[\mathrm{e}^{-r(T-t)} g_9(T, S_T)\Big] = 160\mathrm{e}^{-r(T-t)} - 2(S_t)_1 \mathrm{e}^{-q_1(T-t)}.$$

Furthermore, in the netted portfolio, we include an interest rate-free collateral, $C = 35$. The collateral is such that, at a default, the banks exposure is lowered by 35, and added to the close-out amount. If no default occurs before maturity, then the collateral plays no role.

In Table 4, the portfolio values with and without netting are displayed for different exercise policies and for different credit qualities and WWR-parameters. We see that for low values of $\bar{h}$ (credit spread), *i.e.*, for high credit quality of the counterparty, the risk-free exercise policy is a relatively good approximation also for the risky portfolios. However, when the credit quality decreases, the importance of the correct exercise policy increases. This is also reflected in Tables 5 and 6 in which the corresponding CVA-values are displayed. In practice, a counterparty with a bad credit quality is penalized twice; first by paying a larger CVA (justified), and secondly for paying a CVA which is more overestimated (not justified). The effect is even more significant when we look at expected shortfalls. In Fig. 2, we see that the ES-CVA, at 97.5% level, is overestimated by between 19% and 67% for portfolio without netting and 27% to 103% for the portfolio with netting. Bare in mind that for this particular choice of parameters, the overestimation of the CVAs are only 7.4% and 11.4%. All values reported in this section on risky-valuation are results of only one experiment (no Monte-Carlo averages are used). The reason for this is that we are using many samples, and the variance of the results are therefore low. In addition, we have no reference values to compare with and the main point is that we obtain different values depending on what exercise strategy is used, not high accuracy of the estimated entities.

Table 5
CVA for a portfolio without netting. The relative error in the column to the right represents, in percentage, the overestimation of the CVA by using only the risk-free policy for early-exercise. 'Rel. ov.est.' is short for 'Relative overestimation'.

	$\bar{h}$	CVA	$\overline{\text{CVA}}$	Rel. ov.est.
$b = -0.2$	0	1.15	1.21	5.2%
	0.1	19.55	21.40	9.5%
	0.2	31.04	35.63	14.8%
$b = 0$	0	0	0	0%
	0.1	19.2	20.62	7.4%
	0.2	30.79	35.04	13.8%
$b = 0.2$	0	0.34	0.35	2.9%
	0.1	18.97	20.31	7.1%
	0.2	30.73	34.84	13.4%

Table 6
CVA for a portfolio with netting. The relative error in the column to the right represents, in percentage, the overestimation of the CVA by using only the risk-free policy for early-exercise. 'Rel. ov.est.' is short for 'Relative overestimation'.

	$\bar{h}$	CVA^{net}	$\overline{\text{CVA}}^{\text{net}}$	Rel. ov.est.
$b = -0.2$	0	0.78	0.834	6.4%
	0.1	9.42	10.90	15.7%
	0.2	14.45	17.80	23.2%
$b = 0$	0	0	0	0%
	0.1	9.07	10.23	11.4%
	0.2	14.25	17.30	21.4%
$b = 0.2$	0	0.0364	0.0374	5.6%
	0.1	8.83	9.87	11.7%
	0.2	14.11	17.07	21.0%

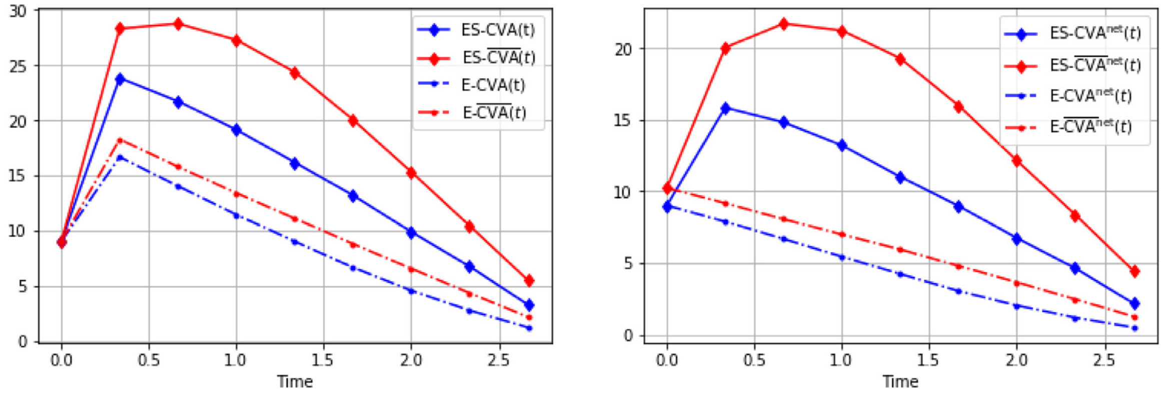


Fig. 2. Time 0 expectation and expected shortfall at level 97.5% of CVA for the risky-free and the risky exercise policies. **Left:** Without netting. **Right:** With netting.

The WWR-parameter b has a relatively small impact. This is not surprising since a positive b gives WWR for call options and Right Way Risk (RWR) for put options and a the other way around with a negative b. Since we have a portfolio consisting of a combination of puts and calls, we have a significant off-setting effect.

4.3.3. Computation time

Regarding computational time, as reported in [2], the training of Phase I takes less than one minute with GPU-based parallelized computations (as in [1]) and is about 30 times slower on a CPU for a portfolio consisting of only one derivative ($j = 1$ from Table 2 is considered). In our experiment, on a CPU, the computation time increased from less than 30 minutes for a single derivative portfolio to around 45 minutes for a portfolio of 8 derivatives. In our experience the method scales better in the number of derivatives than in the dimension of the underlying asset.

Phase II has similar computation time, which is not surprising since the size and shape of the neural networks used are similar.

5. Conclusions

This paper provides a method for computations of portfolio CVA without having to value each individual derivative. We further investigate the impact of netting and WWR in the presence of derivatives with early exercise features and true optionality. It is shown that the exercise decisions should not be made for each derivative in isolation but instead at an aggregate portfolio level with both CCR and the current state of the portfolio ((based on the knowledge which derivatives have been exercised early already) taken into account. As pointed out in Section 2.3, this leads to a situation in which the risky values of each derivative do not add up to the value of the risky portfolio. In fact, from this point of view we can no longer find a value for a single derivative in the portfolio, but only the value of the entire portfolio. This has implications on hedging the CVA since one usually decomposes the risk of a portfolio into market risk and credit risk. When the portfolio value depends on the specific counterparty (the portfolio has a higher value for a counterpparty with higher credit rating) it is not straightforward to hedge the market risk by setting up a collateralized hedge portfolio with other banks. It is beyond the scope of the paper to investigate how the risk can be decomposed into pure market risk and credit risk in this framework, but it is an interesting direction for future research.

Declaration of Competing Interest

None.

Acknowlgedgments

This project is part of the ABC-EU-XVA project and has received funding from the European Unions Horizon 2020 research and innovation programme under the Marie Skłdowska–Curie grant agreement No 813261. Furthermore, we are grateful for valuable comments of two anonymous referees.

Appendix A. Neural network details

In this Appendix, we briefly explain the structure of the neural networks used in this paper. We present pseudo code for the algorithm used to find the exercise strategy for the risk-free portfolio (the extensions to the risky-portfolios are straightforward). Furthermore, it is described how the time 0 value for a risk-free portfolio is computed.

A1. Specification of the neural networks used

For completeness, we introduce all the trainable parameters that are contained in each of the parameters $\theta_1, \theta_2, \ldots, \theta_{N-1}$, and present the structure of the networks.

In this section, the following notation is used:

- We denote the dimension of the input layers by $\mathfrak{D}^{\text{input}} \in \mathbb{N}$, and we assume the same input dimension for all $n \in \{1, 2, \ldots, N-1\}$ networks. The input is assumed to be the market state $x_n^{\text{train}} \in \mathbb{R}^d$, and hence $\mathfrak{D}^{\text{input}} = d$. However, we can add additional information to the input that is mathematically redundant but helps the training, *e.g.*, the immediate pay-off, to obtain as input[8] $\text{concat}(x_n^{\text{train}}(m),\ (g_1(T_n, x_n^{\text{train}}(m)), \ldots, g_J(T_n, x_n^{\text{train}}(m))^T))^T \in \mathbb{R}^{d+J}$, which would give $\mathfrak{D}^{\text{input}} = d + J$;
- For network $n \in \{1, 2, \ldots, N-1\}$, we denote the number of layers[9] by $\mathfrak{L}_n \in \mathbb{N}$, and for layer $\ell \in \{1, 2, \ldots, \mathfrak{L}_n\}$, the number of nodes by $\mathfrak{N}_{\ell,n} \in \mathbb{N}$. Note that $\mathfrak{N}_{n,1} = \mathfrak{D}^{\text{input}}$;
- For network $n \in \{1, 2, \ldots, N\}$, and layer $\ell \in \{2, 3, \ldots, \mathfrak{L}_n\}$ we denote the weight matrix, acting between layers $\ell - 1$ and ℓ, by $w_{n,\ell} \in \mathbb{R}^{\mathfrak{N}_{n,\ell-1} \times \mathfrak{N}_{n,\ell}}$, and the bias vector by $b_{n,\ell} \in \mathbb{R}^{\ell}$;
- For network $n \in \{1, 2, \ldots, N\}$, and layer $\ell \in \{2, 3, \ldots, \mathfrak{L}_n\}$ we denote the (scalar) activation function by $a_{n,\ell} : \mathbb{R} \to \mathbb{R}$ and the vector activation function by $\boldsymbol{a}_{n,\ell} : \mathbb{R}^{\mathfrak{N}_{n,\ell}} \to \mathbb{R}^{\mathfrak{N}_{n,\ell}}$, which, for $x = (x_1, x_2, \ldots, x_{\mathfrak{N}_{n,\ell}})$, is defined by

$$\boldsymbol{a}_{n,\ell}(x) = \begin{pmatrix} a_{n,\ell}(x_1) \\ \vdots \\ a_{n,\ell}(x_{\mathfrak{N}_{n,\ell}}) \end{pmatrix};$$

- The output of the network should belong to $(0,1)^J \subset \mathbb{R}^J$, meaning that the dimension of the output, denoted by $\mathfrak{D}^{\text{output}}$ should equal J. To enforce the output to only take on values in (0,1), we restrict ourselves to scalar-activation functions of the form $a_{n,\mathfrak{L}_n} : \mathbb{R} \to (0, 1)$.

[8] concat denotes a concatenation of vectors.

[9] Input and output layers included.

Network $n \in \{1, 2, \dots N-1\}$ is then defined by

$$\boldsymbol{F}^{\theta_n}(\cdot) = L_{n,\mathfrak{L}_n} \circ L_{n,\mathfrak{L}_n-1} \circ \cdots \circ L_{n,1}(\cdot), \tag{A.1}$$

where, for $n \in \{1, 2, \dots, N-1\}$ and for $x \in \mathbb{R}^{\mathfrak{L}_{n,\ell-1}}$, the layers are defined as

$$L_{n,\ell}(x) = \begin{cases} x, & \text{for } \ell = 1, \\ \boldsymbol{a}_{n,\ell}(w_{n,\ell}^T x + b_{n,\ell}), & \text{for } \ell \geq 2, \end{cases}$$

with $w_{n,\ell}^T$ the matrix transpose of $w_{n,\ell}$. The trainable parameters of network $n \in \{1, 2, \dots, N-1\}$ are then given by the list

$$\theta_n = \{w_{n,2}, b_{n,2}, w_{n,3}, b_{n,3}, \dots, w_{n,\mathfrak{L}_n}, b_{n,\mathfrak{L}_n}\},$$

and as already stated, $\Theta_n = \{\theta_n, \theta_{n+1}, \dots, \theta_{N-1}\}$ and $\Theta = \Theta_1$.

A2. Training and valuation

The main idea of the training and valuation procedure is to fit the parameters to some training data, and then use the fitted parameters to make informed decisions with respect to some unseen, valuation data independent of the training data. The training and valuation is described for the risk-free problem, but the procedure is similar for the risky portfolios.

Training:

Sample $M_{\text{train}} \in \mathbb{N}$ independent realizations of X, which for $m \in \{1,\ 2,\ \dots,\ M_{\text{train}}\}$ are denoted by $x^{\text{train}}(m)$. In practice, we often need to approximate X, *e.g.*, if X satisfies a stochastic differential equation (SDE) we may have to use a temporal discretization scheme. Since this is not the focus of this paper, we assume that X can be approximated pathwise arbitrarily well.

At maturity of the portfolio, we define the cash-flow corresponding to the j:th contract and the m:th realization of X as $\text{CF}_{N,j}(m) = D_{N-1,N} g_j(T_N, x_N^{\text{train}}(m))$ (recall that $g_j(t_N, \cdot) \equiv 0$ if T_N is larger than the date of maturity for contract j).

For $n = N-1,\ N-2,\ \dots,\ 1$, do the following:

1. Approximate the optimal parameter $\theta_n^{**} \in \mathbb{R}^{q_n}$, given by

$$\theta_n^{**} \in \underset{\theta \in \mathbb{R}^{q_n}}{\arg\max} \Bigg(\frac{1}{M_{\text{train}}} \sum_{m=1}^{M_{\text{train}}} \sum_{j=1}^{J} F_j^{\theta}(x_n^{\text{train}}(m))\, g_j(T_n, x_n^{\text{train}}(m)) \\ + \Big(1 - F_j^{\theta}(x_n^{\text{train}}(m))\Big) \text{CF}_{n+1,j}(m) \Bigg),$$

with $\boldsymbol{F}_n^{\theta}$ as in (A.1). The approximation of θ_n^{**} is denoted by θ_n^{*}. The above corresponds to an (empirical) loss-function of the form

$$L(\theta; x^{\text{train}}) = -\frac{1}{M_{\text{train}}} \sum_{m=1}^{M_{\text{train}}} \sum_{j=1}^{J} F_j^{\theta}(x_n^{\text{train}}(m))\, g_j\Big(T_n, x_n^{\text{train}}(m)\Big) \\ + \Big(1 - F_j^{\theta}(x_n^{\text{train}}(m)\Big) \text{CF}_{n+1,j}(m).$$

In the above we distinguish between the theoretically optimal parameter, θ_n^{**} (which we rarely find), and the parameter obtained via an optimization algorithm θ_n^{*}. The minus sign in the loss-function transforms the problem from a maximization to a minimization problem, which is the standard formulation in the machine learning community. Note the straightforward relationship between the loss function and the average cash-flows in (3.9). In practice, the data is often divided into mini-batches, for which the loss-function is minimized consecutively.

2. For $m = 1,\ 2,\ \dots,\ M_{\text{train}}$, and for $j = 1, 2 \dots, J$, update the discounted cash-flows according to:

$$\text{CF}_{n,j}(m) \\ = f_j^{\theta_n^*}(x_n^{\text{train}}(m))\, g(T_n, x_n^{\text{train}}(m)) + D_{n,n+1}\Big(1 - f_j^{\theta_n^*}(x_n^{\text{train}}(m))\Big) \text{CF}_{n+1,j}(m).$$

Note that we use the continuous versions, $F_j^{\theta_n^*}$, in the optimization phase, and the discontinuous versions $f_j^{\theta_n^*}$ when we update the cash-flows.

Below is a list of the most relevant parameters/structural choices:

- Initialization of the trainable parameters, where a typical procedure is to initialize the biases to 0, and sample the weights independently from a normal distribution;
- The activation functions $a_{\ell,n}$, which are used to add a non-linear structure to the neural networks. In our case we have the strict requirement that the activation function of the output layer maps $\mathbb{R}$ to (0,1). This could, however, be relaxed as long as the activation function is both upper and lower bounded, since we can always scale and shift such output to take on values only in (0,1). For a discussion on different activation functions, see *e.g.*, [28];

- The batch size, $\mathcal{B}_n \in \{1, 2, \dots, M_{\text{train}}\}$, is the number of training samples used for each update of θ_n, *i.e.*, with $\mathcal{B}_n = M_{\text{train}}$, the loss function is of the form defined in step 1 above. If we want all batches to be of equal size, we need to choose $\mathcal{B}_n$ to be a multiplier of M_{train};
- For each update of θ_n, we use an optimization algorithm, for which a common choice is the Adam optimizer, proposed in [29]. Depending on the choice of optimization algorithm, there are different parameters related to the specific algorithm to be chosen. One example is the so-called learning rate which decides how much the parameter, θ_n, is adjusted after each batch.

It should be pointed out that in our experience, the accuracy obtained with the algorithm is not particularly sensitive to the specific choices of the number of hidden layers, number of nodes, optimization algorithm, etc. The reason for this is likely that the function surfaces approximated are simple enough to be captured with rather small networks. Moreover, the different optimization algorithms investigated are all of gradient decent-type and therefore of similar nature. It should be mentioned that a careful hyper-parameter optimization has not been carried out.

Once the parameters have been optimized we define $\Theta^* \coloneqq \{\theta_1^*, \theta_2^*, \dots, \theta_{N-1}^*\}$, which contains all the information needed in order to use the algorithm for valuation.

Remark A.1. In the training for the risky portfolio with netting, the algorithm needs to be adjusted since it depends on the exercise history. This cannot easily be included since the algorithm is carried out backwards in time recursively and therefore, at exercise date T_n, we do not yet know which derivatives have been exercised prior to T_n. This is resolved by, at each exercise date T_n, randomly assigning a state for $\alpha_n(m) \in \{0, 1\}^J$, representing A_n, for each sample m. Since the future cash-flows depend on $\alpha_n(m)$, they need to be re-iterated for each m. This is done by iteratively, evaluating the already approximated decision functions at X for each exercise date greater than T_n, until all the derivatives are exercised, or a default of the counterparty occurs.

Valuation:

Sample $M_{\text{test}} \in \mathbb{N}$ independent realizations of X, denoted $(x_t^{\text{val}}(m))_{t\in[0,T]}$. Denote the vector of decision functions by

$$\mathbb{f}_n^{\Theta^*} = \left(\boldsymbol{f}^{\theta_n^*}, \boldsymbol{f}^{\theta_{n+1}^*}, \dots, \boldsymbol{f}^{\theta_{N-1}^*}\right),$$

and $\mathbb{f}^{\Theta^*} \coloneqq \mathbb{f}_1^{\Theta^*}$. We then obtain for sample m, *i.e.*, $x^{\text{val}}(m)$, the following stopping rule

$$\bar{\tau}_{n,j}^{\Theta^*} = \left(\boldsymbol{\tau}\left[\mathbb{f}_n^{\Theta^*}\right]\left(x^{\text{val}}(m)\right)\right)_j = \sum_{k=n}^{N} T_k\, f_j^{\theta_k^*}\left(x_k^{\text{val}}(m)\right) \prod_{\ell=1}^{k-1} \left(1 - f_j^{\theta_\ell^*}\left(x_\ell^{\text{val}}(m)\right)\right).$$

The estimated portfolio value at $t = 0$ is then given by

$$\Pi^V(0, x_0) \approx \Upsilon^V\left(0, x_0 \,|\, \mathbb{f}^{\Theta^*}\right) \approx \frac{1}{M_{\text{val}}} \sum_{m=1}^{M_{\text{val}}} \sum_{j=1}^{J} D_{0,\bar{\tau}_{0,j}^{\Theta^*}}\, g_j\left(\bar{\tau}_{0,j}^{\Theta^*}, x_{\bar{\tau}_{0,j}^{\Theta^*}}^{\text{val}}(m)\right). \tag{A.2}$$

By construction, any stopping strategy is sub-optimal, implying that the estimate (A.2) is biased low. However, it should be pointed out that it is possible to derive a biased-high estimation of $\Pi^V(0, x_0)$ from a dual formulation of the optimal stopping problem, which is described in [1]. In addition, numerical results in [1] show a tight interval for the biased low and biased high estimates for a wide range of problems.

References

[1] S. Becker, P. Cheridito, A. Jentzen, Deep optimal stopping, Journal of Machine Learning Research 20 (2019).

[2] K. Andersson, C. Oosterlee, A deep learning approach for computations of exposure profiles for high-dimensional bermudan options, arXiv preprint arXiv:2003.01977 (2020).

[3] P.A. Forsyth, K.R. Vetzal, Quadratic convergence for valuing american options using a penalty method, SIAM Journal on Scientific Computing 23 (6) (2002) 2095–2122.

[4] C. Reisinger, J.H. Witte, On the use of policy iteration as an easy way of pricing american options, SIAM Journal on Financial Mathematics 3 (1) (2012) 459–478.

[5] C. Vázquez, An upwind numerical approach for an american and european option pricing model, Appl. Math. Comput. 97 (2–3) (1998) 273–286.

[6] T. Haentjens, K.J. in't Hout, ADI schemes for pricing american options under the heston model, Applied Mathematical Finance 22 (3) (2015) 207–237.

[7] K. in't Hout, Numerical partial differential equations in finance explained; an introduction to computational finance, Palgrave Macmillan UK, 2017.

[8] F. Fang, C.W. Oosterlee, Pricing early-exercise and discrete barrier options by fourier-cosine series expansions, Numerische Mathematik 114 (1) (2009) 27.

[9] O. Zhylyevskyy, A fast fourier transform technique for pricing american options under stochastic volatility, Review of Derivatives Research 13 (1) (2010) 1–24.

[10] F. Fang, C.W. Oosterlee, A fourier-based valuation method for bermudan and barrier options under heston's model, SIAM Journal on Financial Mathematics 2 (1) (2011) 439–463.

[11] M. Broadie, J. Detemple, American option valuation: new bounds, approximations, and a comparison of existing methods, Rev. Financ. Stud. 9 (4) (1996) 1211–1250.

[12] M. Rubinstein, et al., Edgeworth binomial trees, Journal of Derivatives 5 (1998) 20–27.

[13] J.C. Jackwerth, Option-implied risk-neutral distributions and implied binomial trees: a literature review, The Journal of Derivatives 7 (2) (1999) 66–82.

[14] R. Bellman, Dynamic programming, Science 153 (3731) (1966) 34–37.

[15] L. Andersen, M. Broadie, Primal-dual simulation algorithm for pricing multidimensional american options, Manage. Sci. 50 (9) (2004) 1222–1234.

[16] F.A. Longstaff, E.S. Schwartz, Valuing american options by simulation: a simple least-squares approach, Rev. Financ. Stud. 14 (1) (2001) 113–147.

[17] M. Broadie, P. Glasserman, et al., A stochastic mesh method for pricing high-dimensional american options, Journal of Computational Finance 7 (2004) 35–72.

[18] M. Broadie, M. Cao, Improved lower and upper bound algorithms for pricing american options by simulation, Quantitative Finance 8 (8) (2008) 845–861.

[19] S. Jain, C.W. Oosterlee, The stochastic grid bundling method: efficient pricing of bermudan options and their greeks, Appl. Math. Comput. 269 (2015) 412–431.

[20] J. Hull, A. White, The impact of default risk on the prices of options and other derivative securities, Journal of Banking & Finance 19 (2) (1995) 299–322.

[21] R. Baviera, G. La Bua, P. Pellicioli, CVA with Wrong-way Risk in the Presence of Early Exercise, in: Innovations in Derivatives Markets, Springer, Cham, 2016, pp. 103–116.

[22] M. Breton, O. Marzouk, An Efficient Method to Price Counterparty Risk, Groupe d'études et de recherche en analyse des décisions, 2014 https://cdi-icd.org/en/wp-content/uploads/sites/2/2018/06/WP-14-01.pdf.

[23] R. Zhou, Back to CVA: the case of american option, Available at SSRN 3189805 (2019).

[24] Basel III: A Global Regulatory Framework for More Resilient Banks and Banking Systems., 2010. Available at: https://www.bis.org

[25] D. Brigo, M. Morini, A. Pallavicini, Counterparty credit risk, collateral and funding: With pricing cases for all asset classes, 478, John Wiley & Sons, 2013.

[26] L. Györfi, M. Kohler, A. Krzyzak, H. Walk, A distribution-free theory of nonparametric regression, Springer Science & Business Media, 2006.

[27] Q. Feng, C.W. Oosterlee, Wrong way risk modeling and computation in credit valuation adjustment for european and bermudan options, Available at SSRN 2852819 (2016).

[28] C. Nwankpa, W. Ijomah, A. Gachagan, S. Marshall, Activation functions: comparison of trends in practice and research for deep learning, arXiv preprint arXiv:1811.03378 (2018).

[29] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, arXiv preprint arXiv:1412.6980. ISO 690. Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego (2015).