

# A deep learning approach for computations of exposure profiles for high-dimensional Bermudan options



Kristoffer Andersson<sup>a,\*</sup>, Cornelis W. Oosterlee<sup>a,b</sup>

<sup>a</sup> National Research Institute for Mathematics and Computer Science, P.O. Box 94079, Amsterdam, 1090 GB, the Netherlands

<sup>b</sup> Mathematical Institute, Utrecht University, the Netherlands

## ARTICLE INFO

### Article history:

Received 6 September 2020

Revised 10 April 2021

Accepted 22 April 2021

Available online 26 May 2021

### Keywords:

Optimal stopping

Deep learning

Expected exposure

Potential future exposure

XVA

## ABSTRACT

In this paper, we propose a neural network-based method for approximating expected exposures and potential future exposures of Bermudan options. In a first phase, the method relies on the Deep Optimal Stopping algorithm (DOS) proposed by Becker, Cheridito, and Jentzen (2019), which learns the optimal stopping rule from Monte-Carlo samples of the underlying risk factors. Cashflow paths are then created by applying the learned stopping strategy on a new set of realizations of the risk factors. Furthermore, in a second phase the cashflow paths are projected onto the risk factors to obtain approximations of pathwise option values. The regression step is carried out by ordinary least squares as well as neural networks, and it is shown that the latter results in more accurate approximations.

The expected exposure is formulated, both in terms of the cashflow paths and in terms of the pathwise option values and it is shown that a simple Monte-Carlo average yields accurate approximations in both cases. The potential future exposure is estimated by the empirical  $\alpha$ -percentile.

Finally, it is shown that the expected exposures, as well as the potential future exposures can be computed under either, the risk neutral measure, or the real world measure, without having to re-train the neural networks.

© 2021 The Authors. Published by Elsevier Inc.  
This is an open access article under the CC BY license  
(<http://creativecommons.org/licenses/by/4.0/>)

## 1. Introduction

The exposure of a financial contract is the maximum amount that an investor stands to lose if the counterparty is unable to fulfill its obligations, for instance, due to a default. This means that, in addition to the market risk, a so-called counterparty credit risk (CCR) needs to be accounted for. Over the counter (OTC) derivatives, *i.e.*, contracts written directly between counterparties, instead of through a central clearing party, are today mainly subject to so-called valuation adjustments (XVA). These valuation adjustments aim to adjust the value of an OTC derivative for certain risk factors, *e.g.*, credit valuation adjustment (CVA), adjusting the value for CCR, funding valuation adjustment (FVA), adjusting for funding cost of an uncollateralized derivative or capital valuation adjustment (KVA), adjusting for future capital costs. The financial exposure is central in calculations of many of the XVAs (for an in-depth overview of XVAs, we refer to [22] and [20]).

\* Corresponding author.

E-mail address: [kristoffer.andersson@cw.nl](mailto:kristoffer.andersson@cw.nl) (K. Andersson).

For a European style option with non-negative value, the exposure is simply the option value at some future time, given all the financial information available today *i.e.*,

$$E_t^{\text{Eur}} = V_t(X_t),$$

where  $X_t$  is the  $d$ -dimensional vector of underlying risk factors. However, for derivatives with early-exercise features, we also need to take into account the possibility that the option has been exercised prior to  $t$ , sending the exposure to zero. In this paper, the focus is on the Bermudan option which gives the holder the right to exercise at finitely many, predefined exercise dates between  $t_0$  and maturity  $T$ . For early-exercise options, the exposure needs to be adjusted for the possibility that the stopping time  $\tau$ , representing the instance of time at which the option is exercised, is smaller than or equal to  $t$ . This leads to the following definition of the exposure

$$E_t^{\text{Ber}} = V_t(X_t) \mathbb{I}_{\{\tau > t\}},$$

where  $\mathbb{I}_{\{\cdot\}}$  is the indicator function. Two of the most common ways to measure the exposure are the expected exposure (EE), which is the expected future value of the exposure, and the potential future exposure (PFE), which is some  $\alpha$ -percentile of the future exposure (usually  $\alpha = 97.5\%$  for the upper, and  $\alpha = 2.5\%$  for the lower tails of the distribution). To accurately compute EE and PFE of a Bermudan option, it is crucial to use an algorithm which is able to accurately approximate, not only  $V_t(X_t)$ , but also  $\mathbb{I}_{\{\tau > t\}}$ .

It is common to compute the EE and the PFE with simulation-based methods, usually from realizations of approximate exposures, from which the distribution of  $E_t$  is approximated by its empirical counterpart. A generic scheme for approximation of exposure profiles is given below:

1. At each exercise date, find a representation,  $v_t : \mathbb{R}^d \rightarrow \mathbb{R}$ , of the true value function  $V_t(\cdot)$ ;
2. Generate trajectories of the underlying risk factors, and at each exercise date, evaluate<sup>1</sup>  $v_t(X_t(\omega))$ ;
3. At the earliest exercise date where  $v_t(X_t(\omega)) \leq g(X_t(\omega))$ , where  $g$  is the pay-off function, set  $\tau(\omega) = t$ ;
4. The distribution of  $E_t^{\text{Ber}}$ , is then approximated by the empirical distribution created by many trajectories of the form  $v_t(X_t(\omega)) \mathbb{I}_{\{\tau(\omega) > t\}}$ . For instance, if the target statistic is the EE, the estimation is the sample mean of the exposure paths.

Step 1 above corresponds to the option valuation problem, which can be tackled by several different methods, all with their own advantages and disadvantages. For instance, the value function can be approximated by solving an associated partial differential equation (PDE), which is done in *e.g.*, [18,24,26,37,41], or the value function can be approximated by a Fourier transform methodology, which is done in *e.g.*, [15,16,43]. Furthermore, classical tree-based methods such as [11,27,38], 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  $d = 4$ ), see [7]. In higher dimensions, Monte-Carlo-based methods are often used, see *e.g.*, [2,8,10,13,29,34]. Monte-Carlo-based methods can generate highly accurate option values at  $t_0$ , *i.e.*,  $v_{t_0}(X_{t_0} = x_{t_0})$ , given that all trajectories of the underlying risk factors are initiated at some deterministic  $x_{t_0}$ . However, at intermediate exercise dates,  $v_t(X_t(\omega))$  might not be an equally good approximation, due to the need of cross sectional regression. We show with numerical examples that, even though the approximation is good on average, the option value is underestimated in certain regions, which is compensated by overestimated values in other regions. For European options, this seems to have minor effect on EE and PFE, but for Bermudan options the effect can be significant due to step 3 above. In addition to the classical regression methods, with *e.g.*, polynomial basis functions, which are cited above, there are several papers in which a neural network plays the role of the basis functions, see *e.g.*, [6,31–33].

In this paper we suggest the use of the Deep Optimal Stopping (DOS) algorithm, proposed in [5], to approximate the optimal stopping strategy. The DOS algorithm approximates the optimal stopping strategy by expressing the stopping time in terms of binary decision functions, which are approximated by deep neural networks. The DOS algorithm is very suitable for exposure computations, since the stopping strategy is computed directly, not as a consequence of the approximate value function, as is the case with most other algorithms. This procedure leads to highly accurate approximations of the exercise boundaries. Furthermore, we propose a neural network-based regression method (NN-regression) to approximate  $V_t(\cdot)$ , which relies on the stopping strategy, approximated by the DOS algorithm. Although NN-regression is needed in order to fully approximate the distributions of future exposures, it turns out that for some target statistics, it is in fact sufficient to approximate the optimal stopping strategy. This is, for instance, the case for some estimates of the EE.

Another advantage of the method is that the EE and the PFE easily can be computed under the risk neutral measure, as well as the real world measure. When the EE is used to compute CVA, the calculations should be done under the risk neutral measure, since CVA is a tradable asset and any other measure (for fixed numéraire) would create opportunities of arbitrage. For other applications, such as for computations of KVA (valuation adjustment to account for future capital costs associated to a derivative), the EE should be calculated under the real world measure. The reason for this is that the KVA, among other things, depends on the required CCR capital which is a function of the EE, which in this case, ideally, should be calculated under the real world measure. For an explanation of KVA in general, see *e.g.*, [21] and for a discussion about the effect of different measures in KVA computations see [28].

<sup>1</sup> We define a filtered probability space in the next section from which  $\omega$  is an element. In this section,  $\omega$  should be viewed as an outcome of a random experiment.

Finally, we emphasize that, even though the focus of this paper is to compute exposure profiles, the algorithms proposed are flexible and with small adjustments other kinds of risk measures could be computed. The first reason for only studying exposure profiles is, as mentioned above, that it is an important building block in computations (or approximations) of many XVAs as well as expected shortfall. Another reason is that there are a number of similar studies in the existing literature (see e.g., [14,17,40]). However, it should be pointed out that the algorithms proposed in this paper are not limited to computations of exposures. As an example, consider the CVA of a Bermudan option, expressed as the difference between the risky (including CCR) and the risk-free (excluding CCR) option values. To accurately compute the risky option value, one needs to adjust the exercise strategy for the default risk of the counterparty. Therefore, it is not sufficient to compute the (risk-free) exposure when computing the CVA (for a discussion see e.g., [4,9]). With minor adjustments to the proposed algorithm, one could compute this kind of advanced CVA (as opposed to the approximation, which is based on the exposure).

This paper is structured as follows: In Section 2, the mathematical formulation of a Bermudan option and its exposure profiles are given. Furthermore, the Bermudan option, as well as the exposure profiles are reformulated to fit the algorithms introduced in later sections. The DOS algorithm is described in Section 3, and we propose some adjustments to make the training procedure more efficient. In Section 4, we present a classical ordinary least squares regression-based method (OLS-regression), as well as the NN-regression to approximate pathwise option values. In Section 5, the EE and PFE formulas are reformulated in a way such that they can easily be estimated by a combination of the algorithms presented in Sections 3 and 4, and a simple Monte-Carlo sampling. Finally, in Section 6, numerical results of the different algorithms, are presented for Bermudan options following the Black-Scholes dynamics and the Heston model stochastic volatility dynamics.

## 2. Problem formulation

For  $d, N \in \mathbb{N} \setminus \{0\}$ , let  $X = (X_t)_{t=0}^N$  be an  $\mathbb{R}^d$ -valued discrete-time Markov process on a complete probability space  $(\Omega, \mathcal{F}, \mathbb{A})$ . The outcome set  $\Omega$  is the set of all possible realizations of the stochastic economy,  $\mathcal{F}$  is a  $\sigma$ -algebra on  $\Omega$  and we define  $\mathcal{F}_n$  as the sub- $\sigma$ -algebra generated by  $(X_m)_{m=0}^n$ . With little loss of generality, we restrict ourselves to the case when  $X$  is constructed from time snapshots of a continuous-time Markov process at monitoring dates  $\{t_0, t_1, \dots, t_N\}$ . The probability measure  $\mathbb{A}$  is a generic notation, representing either the real world measure, or the risk neutral measure, denoted  $\mathbb{P}$  and  $\mathbb{Q}$ , respectively.

If not specifically stated otherwise, equalities and inequalities of random variables should be interpreted in either a  $\mathbb{P}$ -almost surely or a  $\mathbb{Q}$ -almost surely sense, depending on context.

### 2.1. Bermudan options, stopping decisions and exercise regions

A Bermudan option is an exotic derivative that gives the holder the opportunity to exercise the option at a finite number of exercise dates, like one per month. We define the exercise dates as  $\mathbb{T} = \{t_0, t_1, \dots, t_N\}$ , which for simplicity coincide with the monitoring dates. Furthermore, for  $t_n \in \mathbb{T}$ , we let the remaining exercise dates be defined as  $\mathbb{T}_n = \{t_n, t_{n+1}, \dots, t_N\}$ . Let  $\tau$  be an  $X$ -stopping time, i.e., a random variable defined on  $(\Omega, \mathcal{F}, \mathbb{A})$ , taking on values in  $\mathbb{T}$  such that for all  $t_n \in \mathbb{T}$ , it holds that the event  $\{\tau = t_n\} \in \mathcal{F}_n$ . Assume a risk-free rate  $r \in \mathbb{R}$  and let the risk-less savings account process,  $M(t) = e^{r(t-t_0)}$ , be our numéraire. For all  $t \in \mathbb{T}$ , let  $g_t : \mathbb{R}^d \rightarrow \mathbb{R}$  be a measurable function which returns the immediate pay-off of the option, if exercised at market state  $(t, X_t = x \in \mathbb{R}^d)$ . The initial value of a Bermudan option, i.e., the value at market state  $(t_0, X_{t_0} = x_{t_0} \in \mathbb{R}^d)$ , is given by

$$V_{t_0}(x_{t_0}) = \sup_{\tau \in \mathcal{T}} \mathbb{E}_{\mathbb{Q}} \left[ e^{-r(\tau-t_0)} g_{\tau}(X_{\tau}) \mid X_{t_0} = x_{t_0} \right], \tag{2.1}$$

where  $\mathcal{T}$  is the set of all  $X$ -stopping times. Assuming the option has not been exercised prior to some  $t_n \in \mathbb{T}$ , the option value at  $t_n$  is given by

$$V_{t_n}(X_{t_n}) = \sup_{\tau \in \mathcal{T}_n} \mathbb{E}_{\mathbb{Q}}^n \left[ e^{-r(\tau-t_n)} g_{\tau}(X_{\tau}) \right], \tag{2.2}$$

where  $\mathcal{T}_n$  is the set of all  $X$ -stopping times greater than or equal to  $t_n$  and we define  $\mathbb{E}_{\mathbb{A}}^n[\cdot] = \mathbb{E}_{\mathbb{A}}[\cdot | \mathcal{F}_n]$ . To guarantee that (2.1) and (2.2) are well-posed, we assume that for  $t \in \mathbb{T}$  it holds that  $\mathbb{E}_{\mathbb{Q}}[|g_t(X_t)|] < \infty$ .

To concretize (2.1) and (2.2), we view the problem from the holder's perspective. At each exercise date,  $t_n \in \mathbb{T}$ , the holder of the option is facing the decision whether or not to exercise the option, and receive the immediate pay-off. Due to the Markovian nature of  $X$ , the decisions are of Markov-type (Markov decision process (MDP)), meaning that all the information needed to make an optimal decision (in the sense of using a decision policy such that the supremum in (2.1) or (2.2) is attained) is contained in the current market state,  $(t_n, X_{t_n})$ . With this in mind, we define for each  $t_n \in \mathbb{T}$ , the exercise region,  $\mathcal{E}_n$ , in which it is optimal to exercise, and the continuation region,  $\mathcal{C}_n$ , in which it is optimal to hold on, by

$$\mathcal{E}_n = \{x \in \mathbb{R}^d \mid V_{t_n}(x) = g_{t_n}(x)\}, \quad \mathcal{C}_n = \{x \in \mathbb{R}^d \mid V_{t_n}(x) > g_{t_n}(x)\}.$$

Note that  $\mathcal{E}_n \cup \mathcal{C}_n = \mathbb{R}^d$  and  $\mathcal{E}_n \cap \mathcal{C}_n = \emptyset$ .

Below, we give a short motivation for how these decisions can be expressed mathematically and how we can formulate a stopping time in terms of (stopping) decisions at each exercise date. For a more detailed motivation we refer to [5]. We

introduce the following notation for measurable functions<sup>2</sup>

$$\mathcal{D}(A_1; A_2) = \{ f : A_1 \rightarrow A_2 \mid f \text{ measurable} \}.$$

Furthermore, for  $P \in \mathbb{N}$ , let  $\mathcal{D}(A; B)^P$  denote the  $P$ :th Cartesian product of the set  $\mathcal{D}(A_1; A_2)$ . Define the decision functions  $f_0, f_1, \dots, f_N \in \mathcal{D}(\mathbb{R}^d; \{0, 1\})$ , with  $f_N \equiv 1$ , and denote  $\mathbf{f}_n = (f_n, f_{n+1}, \dots, f_N)$  with  $\mathbf{f} = \mathbf{f}_0$ . An  $X$ -stopping time can then be defined as

$$\tau_n(\mathbf{f}_n) = \sum_{m=n}^N t_m f_m(X_{t_m}) \prod_{j=n}^{m-1} (1 - f_j(X_{t_j})), \tag{2.3}$$

with  $\tau(\cdot) = \tau_0(\cdot)$  and where the empty product is defined as 1. We emphasize that

$\tau_n(\mathbf{f}_n) = \tau_n(f_n(X_{t_n}), f_{n+1}(X_{t_{n+1}}), \dots, f_N(X_{t_N}))$  but to make the notation cleaner, we do not specify this dependency explicitly. When it is not clear from the context which process the decision function  $\mathbf{f}_n$  is acting on, we will denote the stopping time by  $\tau_n(\mathbf{f}_n(X))$ , where we recall that  $X = (X_{t_m})_{m=n}^N$ . As a candidate for optimal stopping decisions, we define

$$\mathbf{f}^* \in \arg \max_{\mathbf{f} \in \mathcal{D}(\mathbb{R}^d; \{0,1\})^{N+1}} \mathbb{E}_{\mathbb{Q}} \left[ e^{-r(\tau(\mathbf{f})-t_0)} g_{\tau(\mathbf{f})}(X_{\tau(\mathbf{f})}) \mid X_{t_0} = x_0 \right]. \tag{2.4}$$

We define the fair price of an option that follows the strategy constructed by combining (2.3) and (2.4) as

$$V_n^*(X_{t_n}) = \mathbb{E}_{\mathbb{Q}}^n \left[ e^{-r(\tau_n(\mathbf{f}_n^*)-t_n)} g_{\tau_n(\mathbf{f}_n^*)}(X_{\tau_n(\mathbf{f}_n^*)}) \right]. \tag{2.5}$$

The following theorem states that (2.5), in fact, coincides with the fair price of the option as defined in (2.2).

**Theorem 2.1.** For all  $t_n \in \mathbb{T}$ , and  $V$  and  $V^*$  as defined in (2.2) and (2.5), respectively, it holds that<sup>3</sup>

$$V_n(X_{t_n}) = V_n^*(X_{t_n}).$$

**Proof.** Follows directly from [5, Theorem 1].  $\square$

### 2.2. Exposure profiles

For  $t_n \in \mathbb{T}$  and  $\alpha \in (0, 1)$ , we define the expected exposure (EE) and the potential future exposure (PFE) under the generic probability measure  $\mathbb{A}$  as

$$EE_{\mathbb{A}}(t_n) = \mathbb{E}_{\mathbb{A}} \left[ V_{t_n}(X_{t_n}) \mathbb{I}_{\{\tau(\mathbf{f}^*) > t_n\}} \mid X_{t_0} = x_{t_0} \right], \tag{2.6}$$

$$PFE_{\mathbb{A}}^{\alpha}(t_n) = \inf \left\{ a \in \mathbb{R} \mid \mathbb{A} \left( V_{t_n}(X_{t_n}) \mathbb{I}_{\{\tau(\mathbf{f}^*) > t_n\}} \leq a \mid X_{t_0} = x_{t_0} \right) \geq \alpha \right\}. \tag{2.7}$$

Note that the option value, given by (2.2), is a conditional expectation of future cashflows, which is by definition measured with  $\mathbb{Q}$ . The exposure profiles under the  $\mathbb{P}$ -measure, on the other hand, are statistics of the option value at some future time,  $t_n$ , under the assumption that the distribution  $X_{t_m}$  conditional on  $X_{t_0} = x_{t_0}$  is considered under the  $\mathbb{P}$ -measure.

In the theorem below, the expected exposures are reformulated in terms of discounted cashflows and the decision functions introduced in Section 2.1. It will become clear in later sections that this is a more tractable form for the algorithms used in this paper.

**Theorem 2.2.** Let  $\mathbb{Q}$  and  $\mathbb{P}$  be probability measures on the measurable space  $(\Omega, \mathcal{F})$  and assume that the laws of  $X$  under  $\mathbb{Q}$  and  $\mathbb{P}$  are absolutely continuous with respect to the Lebesgue measure. Then, the expected exposure, (2.6) under the  $\mathbb{Q}$ - and  $\mathbb{P}$ -measures, satisfies

$$EE_{\mathbb{Q}}(t_n) = \mathbb{E}_{\mathbb{Q}} \left[ e^{-r(\tau_n(\mathbf{f}_n^*)-t_n)} g_{\tau_n(\mathbf{f}_n^*)}(X_{\tau_n(\mathbf{f}_n^*)}) \mathbb{I}_{\{\tau(\mathbf{f}^*) > t_n\}} \mid X_{t_0} = x_{t_0} \right], \tag{2.8}$$

$$EE_{\mathbb{P}}(t_n) = \mathbb{E}_{\mathbb{Q}} \left[ e^{-r(\tau_n(\mathbf{f}_n^*)-t_n)} g_{\tau_n(\mathbf{f}_n^*)}(X_{\tau_n(\mathbf{f}_n^*)}) \mathbb{I}_{\{\tau(\mathbf{f}^*) > t_n\}} l(X_{t_n}, X_{t_{n-1}}, \dots, X_{t_0}) \mid X_{t_0} = x_{t_0} \right], \tag{2.9}$$

where  $l(y_n, y_{n-1}, \dots, y_0) = \prod_{k=1}^n \frac{p_{X_{t_k} | X_{t_{k-1}}}(y_k | y_{k-1})}{q_{X_{t_k} | X_{t_{k-1}}}(y_k | y_{k-1})}$ , with  $q_{X_{t_k} | X_{t_{k-1}}}(y_k | y_{k-1})$  and

$p_{X_{t_k} | X_{t_{k-1}}}(y_k | y_{k-1})$  being transition densities for  $X$  under the measures  $\mathbb{Q}$  and  $\mathbb{P}$ , respectively. Note that  $l(y_n, y_{n-1}, \dots, y_0)$  is the Radon-Nikodym derivative of  $\mathbb{P}$  with respect to  $\mathbb{Q}$  evaluated at  $(y_n, y_{n-1}, \dots, y_0)$ .

**Proof.** Proof in Appendix A.

<sup>2</sup> We assume measurable spaces  $(A_1, \mathcal{A}_1)$  and  $(A_2, \mathcal{A}_2)$  and measurable functions with respect to  $\sigma$ -algebras  $\mathcal{A}_1$  and  $\mathcal{A}_2$ . This assumption holds in all cases in this paper.

<sup>3</sup> We recall that equalities and inequalities are in an  $\mathbb{A}$ -almost sure sense.

**Theorem 2.2** opens up for approximations of the expected exposure directly from the discounted cashflows. We now consider the special case, when  $X$  is described by a diffusion-type stochastic differential equation (SDE). Let  $\mu^{\mathbb{Q}} : [t_0, t_N] \times \mathbb{R}^d \rightarrow \mathbb{R}^d$  and  $\sigma : [t_0, t_N] \times \mathbb{R}^d \rightarrow \mathbb{R}^d \times \mathbb{R}^d$  be the drift and diffusion coefficients, respectively, and let  $(W_t^{\mathbb{Q}})_{t \in [t_0, t_N]}$  be a  $d$ -dimensional standard Brownian motion under the measure  $\mathbb{Q}$ . We assume that  $\mu^{\mathbb{Q}}$  and  $\sigma$  satisfy the usual conditions (see e.g., [36]) for existence of a unique, strong solution  $X = (X_t)_{t \in [t_0, t_N]}$  to

$$dX_t = \mu^{\mathbb{Q}}(t, X_t)dt + \sigma(t, X_t)dW_t^{\mathbb{Q}}, \quad t \in [t_0, t_N]; \quad X_{t_0} = x_{t_0}.$$

Let  $\mu^{\mathbb{P}} : [t_0, t_N] \times \mathbb{R}^d \rightarrow \mathbb{R}^d$  and assume that  $\sigma(t, X_t)$  is invertible and that for  $t \in [t_0, t_N]$ <sup>4</sup>  $\|\sigma^{-1}(t, X_t)\|_{\max}$  is bounded almost surely. Then,  $(W_t^{\mathbb{P}})_{t \in [t_0, t_N]}$  given by

$$dW_t^{\mathbb{P}} = -\sigma^{-1}(t, X_t)(\mu^{\mathbb{P}}(t, X_t) - \mu^{\mathbb{Q}}(t, X_t))dt + dW_t^{\mathbb{Q}}$$

is a Brownian motion under the measure  $\mathbb{P}$ . Furthermore, under the measure  $\mathbb{P}$  it holds almost surely that  $X$  is described by

$$dX_t = \mu^{\mathbb{P}}(t, X_t)dt + \sigma(t, X_t)dW_t^{\mathbb{P}}, \quad t \in [t_0, t_N]; \quad X_{t_0} = x_{t_0}. \tag{2.10}$$

As a way to rewrite  $EE_{\mathbb{P}}(t)$  as a conditional expectation under the measure  $\mathbb{Q}$ , we define a "dummy" process  $U = (U_t)_{t \in [t_0, t_N]}$ , which follows the SDE

$$dU_t = \mu^{\mathbb{P}}(t, U_t)dt + \sigma(t, U_t)dW_t^{\mathbb{Q}}, \quad t \in [t_0, t_N]; \quad U_{t_0} = x_{t_0}. \tag{2.11}$$

The reason for introducing this process is that  $U$  has the same distribution under the measure  $\mathbb{Q}$  as  $X$  has under the measure  $\mathbb{P}$ . We can then express  $EE_{\mathbb{P}}(t_n)$  with only  $\mathbb{Q}$ -expectations in the following way

$$\begin{aligned} EE_{\mathbb{P}}(t_n) &= \mathbb{E}_{\mathbb{P}} \left[ V_{t_n}(X_{t_n}) \mathbb{I}_{\{\tau(\mathcal{F}^*(X)) > t_n\}} \mid X_{t_0} = x_{t_0} \right] = \mathbb{E}_{\mathbb{Q}} \left[ V_{t_n}(U_{t_n}) \mathbb{I}_{\{\tau(\mathcal{F}^*(U)) > t_n\}} \mid U_{t_0} = x_{t_0} \right] \\ &= \mathbb{E}_{\mathbb{Q}} \left[ \mathbb{E}_{\mathbb{Q}} \left[ e^{-r(\tau_n(\mathcal{F}_n^*(X)) - t_n)} g_{\tau_n(\mathcal{F}_n^*(X))}(X_{\tau_n(\mathcal{F}_n^*(X))}) \mid X_{t_n} = U_{t_n} \right] \times \mathbb{I}_{\{\tau(\mathcal{F}^*(U)) > t_n\}} \mid U_{t_0} = x_{t_0} \right]. \end{aligned} \tag{2.12}$$

**Remark 2.1.** Regarding the equality between the right hand side of the first line and the second line, we want to emphasize that  $X$  under the measure  $\mathbb{P}$ , and  $U$  under the measure  $\mathbb{Q}$  do not represent the same stochastic process (as they would have done after a change of measure). If they were, then the conditional expectation would change, and the equality would not hold. To enforce the equality to hold we could have corrected with the Radon–Nikodym derivative, and obtained (2.9). However, to find a way to write  $EE_{\mathbb{P}}(t_n)$  without having to include the Radon–Nikodym derivative, we introduce another process  $U$ , which is distributed in such a way that the equality holds, i.e., the conditional expectation under  $\mathbb{P}$ , when using  $X$ , should equal the conditional expectation under  $\mathbb{Q}$ , when using  $U$ . For this to hold, it is sufficient that the distribution of  $X$  under the measure  $\mathbb{P}$  equals the distribution of  $U$  under the measure  $\mathbb{Q}$ , which is satisfied when  $X$  follows (2.10) and  $U$  follows (2.11).

**Remark 2.2.** The final equality is obtained by the fact that  $V_{t_n}(U_{t_n})$  is the option value at the (random) state  $(t_n, U_{t_n})$ , given by (2.5).

Before we get rid of the inner expectation in (2.12), we need to define a process following (2.11) on  $[t_0, t_n]$ , and the dynamics of (2.10) on  $[t_n, t_N]$  with (stochastic) initial condition  $X_{t_n} = U_{t_n}$ . We denote such a process by  $\tilde{X}^{t_n} = (\tilde{X}_t^{t_n})_{t \in [t_0, t_N]}$ , and conclude that  $\tilde{X}^{t_n}$  should satisfy the following SDE

$$d\tilde{X}_t^{t_n} = \mu^{\mathbb{P}, \mathbb{Q}, t_n}(t, \tilde{X}_t^{t_n})dt + \sigma(t, \tilde{X}_t^{t_n})dW_t^{\mathbb{Q}}, \quad t \in [t_0, t_N]; \quad \tilde{X}_{t_0}^{t_n} = x_{t_0}, \tag{2.13}$$

where  $\mu^{\mathbb{P}, \mathbb{Q}, t_n}(t, \cdot) = \mu^{\mathbb{P}}(t, \cdot) \mathbb{I}_{\{t \leq t_n\}} + \mu^{\mathbb{Q}}(t, \cdot) \mathbb{I}_{\{t > t_n\}}$ . Note that we have implicitly assumed that also  $\mu^{\mathbb{P}}$  satisfies the usual conditions for existence of a unique strong solution,  $U$ , to (2.11). As a consequence of this assumption we are also guaranteed that there exists a unique strong solution,  $\tilde{X}^{t_n}$ , to (2.13). We can then use the law of total expectation to obtain

$$\begin{aligned} EE_{\mathbb{P}}(t_n) &= \mathbb{E}_{\mathbb{Q}} \left[ V_{t_n}(\tilde{X}_{t_n}^{t_n}) \mathbb{I}_{\{\tau(\mathcal{F}^*) > t_n\}} \mid \tilde{X}_{t_0}^{t_n} = x_{t_0} \right] \\ &= \mathbb{E}_{\mathbb{Q}} \left[ e^{-r(\tau_n(\mathcal{F}_n^*) - t_n)} g_{\tau_n(\mathcal{F}_n^*)}(\tilde{X}_{\tau_n(\mathcal{F}_n^*)}^{t_n}) \mathbb{I}_{\{\tau(\mathcal{F}^*) > t_n\}} \mid \tilde{X}_{t_0}^{t_n} = x_{t_0} \right], \end{aligned} \tag{2.14}$$

where we remind ourselves that  $\tau(\mathcal{F}^*) = \tau(\mathcal{F}^*(\tilde{X}^{t_n}))$  and  $\tau_n(\mathcal{F}_n^*) = \tau_n(\mathcal{F}_n^*(\tilde{X}^{t_n}))$ .

In the next sections we describe a method to approximate  $\mathcal{F}^*(\cdot)$ . It is then straightforward to estimate (2.8), (2.9) and (2.14) by Monte-Carlo sampling. Furthermore, in Section 4, we introduce a method to approximate the price function  $V_{t_n}(\cdot)$ , which makes it straightforward to also approximate the potential future exposure (2.7).

<sup>4</sup> For a matrix  $A \in \mathbb{R}^{d \times d}$ ,  $\|A\|_{\max} = \max_{ij} |A_{ij}|$ .

### 3. Learning stopping decisions

In the first part of this section, we present the DOS algorithm, which was proposed in [5]. The idea is to use fully connected neural networks to approximate the decision functions introduced in the previous section. The neural networks are optimized backwards in time with the objective to maximize the expected discounted cashflow at each exercise date. In the second part of this section, we suggest some adjustments that can be done in order to make the optimization more efficient.

#### 3.1. The Deep Optimal Stopping algorithm

As indicated above, the core of the algorithm is to approximate decision functions. To be more precise, for  $n \in \{0, 1, \dots, N\}$ , the decision function  $f_n$  is approximated by a fully connected neural network of the form  $f_n^{\theta_n} : \mathbb{R}^d \rightarrow \{0, 1\}$ , where  $\theta_n \in \mathbb{R}^{q_n}$  is a vector containing all the  $q_n \in \mathbb{N}$  trainable parameters in network  $n$ . We assume that the initial state,  $x_0 \in \mathbb{R}^d$ , is such that it is sub-optimal to exercise the option at  $t_0$ , and therefore set  $\theta_0$  such that  $f_0^{\theta_0}(x_0) = 0$  (for a further discussion, see Remark 6 in [5]). Since binary decision functions are discontinuous, and therefore unsuitable for gradient-type optimization algorithms, we use as an intermediate step, the neural network  $F_n^{\theta_n} : \mathbb{R}^d \rightarrow (0, 1)$ . Instead of a binary decision, the output of the neural network  $F_n^{\theta_n}$  can be viewed as the probability<sup>5</sup> 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  $f_n^{\theta_n}(\cdot) = \alpha \circ F_n^{\theta_n}(\cdot)$ , where  $\alpha(x) = \mathbb{I}_{\{x \geq 1/2\}}$ . Our objective is to find  $\theta_n$  such that

$$\mathbb{E}_{\mathbb{Q}}^n \left[ f_n^{\theta_n}(X_{t_n}) g_{t_n}(X_{t_n}) + (1 - f_n^{\theta_n}(X_{t_n})) e^{-r(\tau_{n+1}(f_{n+1}^*) - t_n)} g_{\tau_{n+1}(f_{n+1}^*)}(X_{\tau_{n+1}(f_{n+1}^*)}) \right], \tag{3.1}$$

is as close as possible to  $V_{t_n}(X_{t_n})$  (in mean squared sense), where  $f_{n+1}^*$  is the vector of optimal decision functions, defined in (2.4). Although (3.1) is an accurate representation of the optimization problem, it gives us some practical problems. In general, we have no access to either  $f_{n+1}^*$  or the distribution of  $V_{t_n}(X_{t_n})$  and in most cases the expectation needs to be approximated. We however notice that at maturity, the option value is equal to its intrinsic value, i.e.,  $V_{t_N}(\cdot) \equiv g_{t_N}(\cdot)$ , which implies that  $f_N^* \equiv 1$  and  $\tau_N(f_N^*) = t_N$ . With this insight, we can write (3.1) with  $n = N - 1$  in the form

$$\mathbb{E}_{\mathbb{Q}}^{N-1} \left[ f_{N-1}^{\theta_{N-1}}(X_{t_{N-1}}) g_{t_{N-1}}(X_{t_{N-1}}) + (1 - f_{N-1}^{\theta_{N-1}}(X_{t_{N-1}})) e^{-r(t_N - t_{N-1})} g_{t_N}(X_{t_N}) \right], \tag{3.2}$$

which can be approximated by a sample mean of (3.2) obtained by  $M \in \mathbb{N}$  realizations of  $X$ . Note that, in such a sample mean, the only unknown entity is the parameter  $\theta_{N-1}$  in the decision function  $f_{N-1}^{\theta_{N-1}}$ . We want to find  $\theta_{N-1}$  such that the approximation (sample mean) of (3.2) is maximized, since it represents the average cashflow in  $[t_{N-1}, t_N]$ . Once  $\theta_{N-1}$  is optimized, we use this parameter and find  $\theta_{N-2}$  such that the average cashflow on  $[t_{N-2}, t_N]$  is maximized.

In the next section, we explain the parameters  $\theta_n$  and present the structure for the neural networks used in this paper.

##### 3.1.1. Specification of the neural networks used

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

- 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  $n \in \{1, 2, \dots, N - 1\}$  networks. The input is assumed to be the market state  $x_{t_n}^{\text{train}} \in \mathbb{R}^d$ , 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  $(\text{vec}(x_{t_n}^{\text{train}}(m)), g_{t_n}(x_{t_n}^{\text{train}}(m))) \in \mathbb{R}^{d+1}$ , which would give  $\mathfrak{D}^{\text{input}} = d + 1$ . With mathematically redundant, we refer to the fact that the decision can be made by the current state of the underlying assets captured in  $x_{t_n}^{\text{train}}(m)$ . The immediate pay-off,  $g(x_{t_n}^{\text{train}}(m))$ , provides some additional structure of the problem, which may help in the training phase of the neural network. In [5], the authors claim that by adding the immediate pay-off to the input, the efficiency of the algorithm was improved;
- For network  $n \in \{1, 2, \dots, N - 1\}$ , we denote the number of layers by  $\mathfrak{L}_n \in \mathbb{N}$ , and for layer  $\ell \in \{1, 2, \dots, \mathfrak{L}_n\}$ , the number of nodes by  $\mathfrak{N}_{n,\ell} \in \mathbb{N}$ . Note that  $\mathfrak{N}_{n,1} = \mathfrak{D}^{\text{input}}$ ;
- For network  $n \in \{1, 2, \dots, N\}$ , and layer  $\ell \in \{2, 3, \dots, \mathfrak{L}_n\}$  we denote the weight matrix, acting between layers  $\ell - 1$  and  $\ell$ , by  $w_{n,\ell} \in \mathbb{R}^{\mathfrak{N}_{n,\ell-1} \times \mathfrak{N}_{n,\ell}}$ , and the bias vector by  $b_{n,\ell} \in \mathbb{R}^{\mathfrak{N}_{n,\ell}}$ ;
- For network  $n \in \{1, 2, \dots, N\}$ , and layer  $\ell \in \{2, 3, \dots, \mathfrak{L}_n\}$  we denote the (scalar) activation function by  $a_{n,\ell} : \mathbb{R} \rightarrow \mathbb{R}$  and the vector activation function by  $\mathbf{a}_{n,\ell} : \mathbb{R}^{\mathfrak{N}_{n,\ell}} \rightarrow \mathbb{R}^{\mathfrak{N}_{n,\ell}}$ , which for  $x = (x_1, x_2, \dots, x_{\mathfrak{N}_{n,\ell}})$  is defined by

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

<sup>5</sup> Although 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  $\mathcal{F}_t$ -measurable.

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

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

$$F_n^{\theta_n}(\cdot) = L_{n,\mathfrak{L}_n} \circ L_{n,\mathfrak{L}_{n-1}} \circ \dots \circ L_{n,1}(\cdot),$$

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

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

where  $w_{n,\ell}^T$  is 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  $\theta_n = \{w_{n,2}, b_{n,2}, w_{n,3}, b_{n,3}, \dots, w_{n,\mathfrak{L}_n}, b_{n,\mathfrak{L}_n}\}$ . Furthermore, since we have  $N - 1$  neural networks, we denote by  $\theta_n = \{\theta_n, \theta_{n+1}, \dots, \theta_{N-1}\}$  the trainable parameters in the neural networks at exercise dates  $\mathbb{T}_n$  and by  $\theta = \theta_1$ .

### 3.1.2. 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, so-called, valuation data.

The training part of the algorithm is summarized below in pseudo code.

**Training phase:** Sample  $M_{\text{train}} \in \mathbb{N}$  independent realizations of  $X$ , which for  $m \in \{1, 2, \dots, M_{\text{train}}\}$  are denoted  $(x_{t_n}^{\text{train}}(m))_{n=0}^N$ . At maturity, define the cashflow as  $\text{CF}_N(m) = g_{t_N}(x_{t_N}^{\text{train}}(m))$ . For  $n = N - 1, N - 2, \dots, 1$ , do the following:

1. Define the (empirical) loss function

$$L(\theta; x^{\text{train}}) = -\frac{1}{M_{\text{train}}} \sum_{m=1}^{M_{\text{train}}} F_n^\theta(x_{t_n}^{\text{train}}(m)) g_{t_n}(x_{t_n}^{\text{train}}(m)) + (1 - F_n^\theta(x_{t_n}^{\text{train}}(m))) e^{-r(t_{n+1}-t_n)} \text{CF}_{n+1}(m), \tag{3.3}$$

and find  $\hat{\theta}_n \in \mathbb{R}^{q_n}$ , which approximates  $\hat{\theta}_n^* \in \arg \min_{\theta \in \mathbb{R}^{q_n}} L(\theta; x^{\text{train}})$ .

The minus sign in the loss-function transforms the problem from a maximization to minimization, which is the standard formulation in the machine learning community. Note the straightforward relationship between the loss function and the average cashflows in (3.2). In practice, the data is often divided into mini-batches, for which the loss-function is minimized consecutively;

2. For  $m = 1, 2, \dots, M_{\text{train}}$ , update the discounted cashflow according to:

$$\text{CF}_n(m) = f_n^{\hat{\theta}_n}(x_{t_n}^{\text{train}}(m)) g_{t_n}(x_{t_n}^{\text{train}}(m)) + (1 - f_n^{\hat{\theta}_n}(x_{t_n}^{\text{train}}(m))) e^{-r(t_{n+1}-t_n)} \text{CF}_{n+1}(m).$$

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 of different activation functions, see e.g., [35];
- The batch size,  $B_n \in \{1, 2, \dots, M_{\text{train}}\}$ , is the number of training samples used for each update of  $\theta_n$ , i.e., with  $B_n = M_{\text{train}}$ , the loss function as in (3.3). Note that if we want all batches to be of equal size, we need to choose  $B_n$  to be a multiplier of  $M_{\text{train}}$ ;
- For each update of  $\theta_n$ , we use an optimization algorithm, for which a common choice is the Adam optimizer, proposed in [30]. 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,  $\theta_n$  is adjusted after each batch.

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,  $\{\theta_1, \theta_2, \dots, \theta_{N-1}\}$ , have been optimized we can use the algorithm for valuation. **Valuation phase:** Sample  $M_{\text{val}} \in \mathbb{N}$  independent realizations of  $X$ , denoted  $(x_{t_n}^{\text{val}}(m))_{n=0}^N$ . We emphasize that the valuation data should be generated independently from the training data. Denote the vector of decision

functions by  $f_n^{\hat{\theta}} = (f_n^{\hat{\theta}_n}, f_{n+1}^{\hat{\theta}_{n+1}}, \dots, f_{N-1}^{\hat{\theta}_{N-1}})$ , and  $f^{\hat{\theta}} = f_0^{\hat{\theta}}$ . We then obtain for sample  $m$ , i.e.,  $x^{\text{val}}(m)$ , the following stopping rule at time  $t_n$

$$\tau_n(f_n^{\hat{\theta}}(x^{\text{val}}(m))) = \sum_{m=n}^N t_m f_m^{\hat{\theta}_m}(x_{t_m}^{\text{val}}(m)) \prod_{j=0}^{m-1} (1 - f_j^{\hat{\theta}_j}(x_{t_j}^{\text{val}}(m))).$$

The estimated option value at  $t_0$  is then given by

$$\hat{V}_{t_0}(x_0) = \frac{1}{M_{\text{val}}} \sum_{m=1}^{M_{\text{val}}} e^{-r(\tau(f^{\hat{\theta}}) - t_0)} g_{\tau(f^{\hat{\theta}})}(x_{\tau(f^{\hat{\theta}})}^{\text{val}}(m)), \tag{3.4}$$

where we recall that  $\tau(f^{\hat{\theta}}) = \tau_0(f_0^{\hat{\theta}}(x^{\text{val}}(m)))$ . Note that, by construction, any stopping strategy is sub-optimal, implying that the estimate (3.4) is biased low. It should be pointed out that it is possible to derive a biased high estimate of (2.1) from a dual formulation of the optimal stopping problem, which is described in [5]. In addition, numerical results in [5] show a tight interval for the biased low and biased high estimates for a wide range of different problems.

### 3.2. Proposed adjustments to the algorithm

The presentation of the DOS algorithm in [5] is in a general form. In addition to the pricing of Bermudan options, the authors considered the non-Markovian problem to optimally stop a fractional Brownian motion (this is done by including also the historical states in the current state of the system). Since the aim of this paper is more specific (to approximate exposure profiles of Bermudan options), it is natural to use the known underlying structure of this specific problem. In this section we use some properties of the specific problems, and propose some adjustments to the DOS-algorithm, which make the training procedure more efficient.

#### 3.2.1. Reuse of neural network parameters

The first proposed adjustment is to reuse parameters of neural networks that have already been optimized. We note that for a single Bermudan option (possibly with a high-dimensional underlying asset price process) the pay-off functions are identical at all exercise dates, i.e.,  $g_{t_n} = g_{t_m}$  for all  $t_n, t_m \in \mathbb{T}$ . In this case the stopping rules at adjacent exercise dates are similar, especially when  $t_{n+1} - t_n$  is small. We therefore use the stopping strategy at  $t_{n+1}$  as an "initial guess" for the stopping strategy at  $t_n$ . This is done by initializing the trainable parameters in network  $n$  by the already optimized parameters in network  $n + 1$ , i.e., at  $t_n$ , initialize  $\theta_n$  by  $\hat{\theta}_{n+1}$ . This allows us to use smaller learning rates leading to a more efficient algorithm.

#### 3.2.2. Use simple stopping decisions when possible

The term simple stopping decisions is loosely defined as stopping decisions that follow directly without any sophisticated optimization algorithm. The most obvious example is when the contract is out-of-the-money, in which case it is never optimal to exercise. For  $t_n \in \mathbb{T}$ , we define the set of in-the-money points and out-of-the-money points as

$$\text{ITM}_n = \{x \in \mathbb{R}^d \mid g_{t_n}(x) > 0\}, \quad \text{OTM}_n = \{x \in \mathbb{R}^d \mid g_{t_n}(x) = 0\},$$

respectively. Another, less obvious insight is that, given a single Bermudan option with identical pay-off functions at all exercise dates, if it is optimal to exercise at  $(t_n, x)$ , then it is also optimal to exercise at  $(t_{n+1}, x)$ . Or in other words, the exercise region is non-decreasing with time. This statement is formulated as a theorem below.

**Theorem 3.1.** Define the set of exercise dates by  $\{t_0, \dots, t_n, t_{n+1}, \dots, t_N, t_N + \Delta\}$ , and let  $\Delta = t_{n+1} - t_n = t_{N+1} - t_N \geq 0$ . Note that an equidistant time grid is sufficient, but not necessary for the above to be satisfied. Moreover, assume that  $V_{t_n}(\cdot; t_N) = V_{t_{n+1}}(\cdot; t_N + \Delta)$ , where  $t_N$  and  $t_N + \Delta$  indicate the maturity of otherwise identical contracts of the form (2.2), with  $g = g_{t_n}$  for all exercise dates  $t_n$ . Then, for any  $\bar{x} \in \mathcal{E}_n$ , it holds that  $\bar{x} \in \mathcal{E}_{n+1}$ .

**Proof.** Proof in Appendix A.

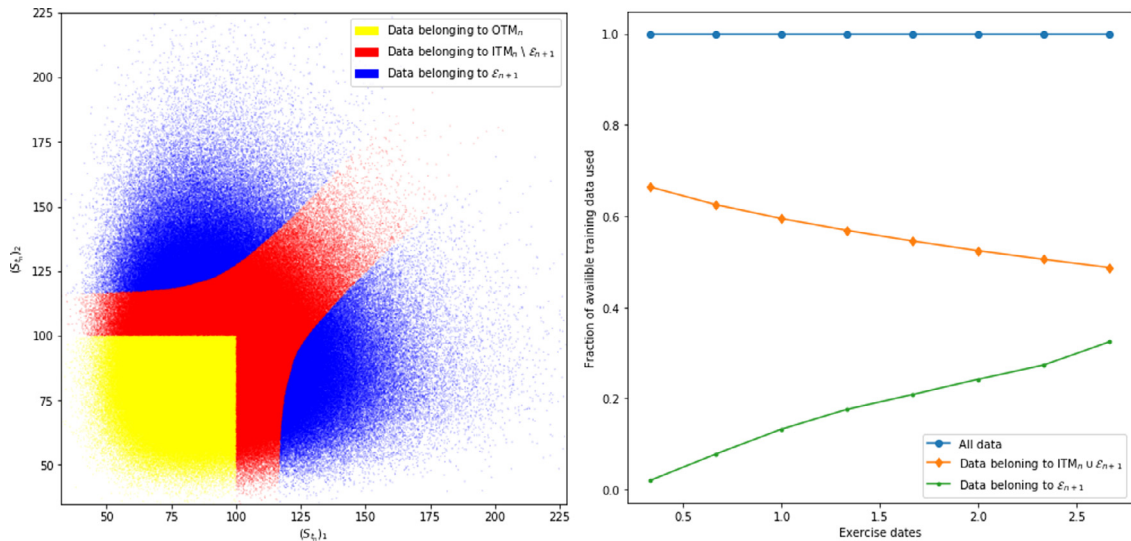
Theorem 3.1 shows that the exercise region is non-decreasing with time, but since the optimization of the neural network parameters is carried out backwards in time we instead use the fact that the continuation region is non-increasing backwards in time. In practice, this leads to the following three alternatives:

- A1. Use all training data in the optimization algorithm (as the algorithm is described in Section 3.1);
- A2. At  $t_n \in \mathbb{T}$ , use the subset of the training data satisfying  $x_{t_n}^{\text{val}}(m) \in \text{ITM}_n$  in the optimization algorithm. Define the decision functions as

$$f_n^{\hat{\theta}_n}(\cdot) = \mathbb{I}_{\{g_{t_n}(\cdot) > 0\}}(\alpha \circ F_n^{\hat{\theta}_n}(\cdot)).$$

To only use "in the money paths" is also employed in the Least Squares Method (LSM), proposed in [34];





**Fig. 1.** **Left:** Blue points in  $\mathcal{E}_{n+1}$  (used for optimization in A3), blue and red points in  $\text{ITM}_n$  (used for optimization in A2) and the blue, red and yellow points are all the available data (used for optimization in A1). **Right:** The fraction of the total data used in each case at each exercise date. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

A3. At  $t_n$  use the subset of the training data  $x_{t_n}^{\text{train}}(m) \in \mathcal{E}_{n+1}$  in the optimization algorithm. Define the decision functions as

$$f_n^{\hat{\theta}_n}(\cdot) = f_{n+1}^{\hat{\theta}_{n+1}}(\cdot) \circ (\alpha \circ F_n^{\hat{\theta}_n}(\cdot)).$$

In Fig. 1 the three cases above are visualized for a two-dimensional max call option at one of the exercise dates. To the left we have the blue points belonging to  $\mathcal{E}_{n+1}$  (used for optimization in A3), the blue and red points belong to  $\text{ITM}_n$  (used for optimization in A2) and the blue, red and yellow points are all the available data (used for optimization in A1). To the right, we see the fraction of the total data used in each case at each exercise date.

#### 4. Learning pathwise option values

In Section 3 an algorithm to learn stopping decisions was described and (3.4) gives an approximation of the option value at time  $t_0$ , given some deterministic initial state  $X_{t_0} = x_{t_0} \in \mathbb{R}^d$ . As described in Section 2.2, to compute exposure profiles we sometimes need additional information about the future distribution of the option values. In this section, we present two methods to approximate the pathwise option values at all exercise dates. The first method is the well-established Ordinary Least Squares (OLS) regression and the second method is a neural network-based least squares regression. Both methods rely on projections of conditional expectations on a finite-dimensional function space.

##### 4.1. Formulation of regression problem

Central for the regression algorithms presented in this section is the cashflow process,  $Y = (Y_{t_n})_{n=0}^N$ , defined as<sup>6</sup>

$$Y_{t_n} = e^{-r(\tau_n(f_n^*) - t_n)} g_{\tau_n(f_n^*)}(X_{\tau_n(f_n^*)}), \tag{4.1}$$

where  $\tau_n(\cdot)$  and  $f^*$  are defined in (2.3) and (2.4), respectively. We assume that for  $t_n \in \mathbb{T}$  it holds that  $\mathbb{E}_{\mathbb{Q}}[g_{t_n}(X_{t_n})^2] < \infty$ , which also implies that  $\mathbb{E}_{\mathbb{Q}}[Y_{t_n}^2] < \infty$ . The following theorem states that the option value, at some  $t_n \in \mathbb{T}$ , is equivalent (in  $L_2$  sense) to the so-called regression function. Furthermore, we see that the regression function can be obtained by solving a minimization problem.

**Theorem 4.1.** Let  $Y_{t_n}$  be as defined in (4.1) and for  $h_n \in \mathcal{D}(\mathbb{R}^d, \mathbb{R})$ , define the so-called  $L_2$ -risk as

$$\mathbb{E}_{\mathbb{Q}}[|h_n(X_{t_n}) - Y_{t_n}|^2].$$

It then holds that

$$\mathbb{E}_{\mathbb{Q}}[|V_{t_n}(X_{t_n}) - Y_{t_n}|^2] = \min_{h_n \in \mathcal{D}(\mathbb{R}^d, \mathbb{R})} \mathbb{E}_{\mathbb{Q}}[|h_n(X_{t_n}) - Y_{t_n}|^2],$$

<sup>6</sup> Note that  $Y$  is the discounted cashflow-process, which in the training phase was denoted by  $\text{CF}_n$ . Here, we instead use  $Y$  to emphasize that the pathwise valuation problem is a standard regression problem in which  $X$  and  $Y$  usually are the observation vector, and the response variable, respectively.

or equivalently

$$V_{t_n}(\cdot) \in \arg \min_{h_n \in \mathcal{D}(\mathbb{R}^d; \mathbb{R})} \mathbb{E}_{\mathbb{Q}}[|h_n(X_{t_n}) - Y_{t_n}|^2].$$

**Proof.** This is a straightforward consequence of the fact that the conditional expectation is the (least-squares) projection onto the Markov states,  $h_n(X_{t_n})$ .  $\square$

In practice, the distribution of  $(X, Y)$  is usually unknown. On the other hand, we are often able to generate samples distributed as<sup>7</sup>  $(X, Y)$ . We consider some  $t_n \in \mathbb{T}$ , and generate  $M_{\text{reg}} \in \mathbb{N}$  independent realizations of the regression pair  $(X_{t_n}, Y_{t_n})$ , which we denote by  $(x_{t_n}^{\text{reg}}(m), y_{t_n}^{\text{reg}}(m))_{m=1}^{M_{\text{reg}}}$ . Similarly, we define the empirical  $L_2$ -risk by

$$\frac{1}{M_{\text{reg}}} \sum_{m=1}^{M_{\text{reg}}} |h_n(x_{t_n}^{\text{reg}}(m)) - y_{t_n}^{\text{reg}}(m)|^2.$$

With a fixed sample of regression pairs it is of course possible to find a function  $h \in \mathcal{D}(\mathbb{R}^d; \mathbb{R})$  such that the empirical  $L_2$ -risk equals zero. However, such a function is not a consistent estimator in general. Therefore, we want to use a smaller class of more regular functions. When choosing the function class, which we denote by  $\mathcal{A}_M$ , we need to keep two aspects in mind;

- P1. It should be "rich enough" to be able to approximate  $V_{t_n}(\cdot)$  sufficiently accurately;
- P2. It should not be "too rich" since that may cause the empirical  $L_2$ -risk being an inaccurate approximation of the  $L_2$ -risk. Since this problem is more severe for smaller  $M_{\text{reg}}$ , it is reasonable to have the sample size in mind when choosing the function class, and hence the subscript  $M$  on  $\mathcal{A}_M$ , where "reg" is dropped for notational convenience. A too rich function class may lead to what is known as overfitting in the machine learning community.

Given a sample and a function class  $\mathcal{A}_M$ , we define the empirical regression function as

$$m_M(\cdot) \in \arg \min_{h \in \mathcal{A}_M} \frac{1}{M_{\text{reg}}} \sum_{m=1}^{M_{\text{reg}}} |h(x_{t_n}^{\text{reg}}(m)) - y_{t_n}^{\text{reg}}(m)|^2.$$

In the next section, we introduce the two different types of function classes that are used in this paper.

#### 4.2. Ordinary least squares regression

At  $t_n \in \mathbb{T}$ , we assume that  $V_{t_n}(X_{t_n})$  can be represented by a linear combination of a countable set of  $\mathcal{F}_n$ -measurable basis functions. We denote by  $\{\phi_b\}_{b=0}^{\infty}$  the basis functions and given optimal parameters  $\alpha_{t_n}^{(1)}, \alpha_{t_n}^{(2)}, \dots$  (in the sense that the  $L_2$ -risk against  $V_{t_n}(X_{t_n})$  is minimized) and define

$$v(t_n, X_{t_n}) = \sum_{b=1}^{\infty} \alpha_{t_n}^{(b)} \phi_b(X_{t_n}).$$

For practical purposes we use the first  $B \in \mathbb{N}$  basis functions, so that

$$v_B(t_n, X_{t_n}) = \sum_{b=1}^B \alpha_{t_n}^{(b)} \phi_b(X_{t_n}). \tag{4.2}$$

We now want to estimate (4.2) by projecting a sample of realizations of  $(X_{t_n}, Y_{t_n})$  onto the  $B$  first basis functions. This procedure is similar to LSM, [34]. In the LSM, only ITM samples are used, which is motivated by the fact that it is never optimal to exercise an option that is OTM and the objective is to find the optimal exercise strategy. Furthermore, the authors claim that the number of basis functions needed to obtain an accurate approximation is significantly reduced since the approximation region is reduced by only considering ITM paths. However, this is not possible in our case since we need to approximate the option everywhere<sup>8</sup> On the other hand, the exercise region,  $\mathcal{E}_n$ , has already been approximated (as described in Section 3.1) and the option value in the exercise region is always known. This means that, similar to the LSM, the approximation region can be reduced (in many cases significantly) by only considering samples belonging to the continuation region,  $\mathcal{C}_n$ .

Given a sample of regression pairs  $(x_{t_n}^{\text{reg}}(m), y_{t_n}^{\text{reg}}(m))_{m=1}^{M_{\text{reg}}}$ , we let  $M_{\text{reg}}^{\mathcal{C}_n}$  denote the number of samples belonging to  $\mathcal{C}_n$  and let  $(x_{t_n}^{\text{reg}}(m), y_{t_n}^{\text{reg}}(m))_{m=1}^{M_{\text{reg}}^{\mathcal{C}_n}}$  be our new samples of regression pairs (where the indexation has been appropriately changed).

<sup>7</sup> In practise, we can only generate samples distributed as  $(X, \hat{Y})$ , where  $\hat{Y}$  is the approximate discounted cashflow process obtained by using the neural network-based decision functions instead of the optimal decision functions in (4.1).

<sup>8</sup> By "everywhere" we mean the region in which the distribution of  $X_n$  has positive density. Of course, this is in many cases  $\mathbb{R}^d$ , so in practice by "everywhere" we mean the region in which the density is significantly positive.

Assuming  $M_{\text{reg}}^{c_n} \geq 1$ , we want to find a set of regression coefficients  $\hat{\alpha}_{t_n} = (\hat{\alpha}_{t_n}^{(1)}, \dots, \hat{\alpha}_{t_n}^{(B)})$  such that the following empirical  $L_2$ -risk is minimized

$$\frac{1}{M_{\text{reg}}^{c_n}} \sum_{m=1}^{M_{\text{reg}}^{c_n}} \left| \sum_{b=1}^B \alpha_{t_n}^{(b)} \phi_b(x_{t_n}^{\text{reg}}(m)) - y_{t_n}^{\text{reg}}(m) \right|^2. \tag{4.3}$$

For notational convenience, we use the compact notation  $\mathbf{x}_{t_n} = (x_{t_n}^{\text{reg}}(1), \dots, x_{t_n}^{\text{reg}}(M_{\text{reg}}^{c_n}))$ ,  $\mathbf{y}_{t_n} = (y_{t_n}^{\text{reg}}(1), \dots, y_{t_n}^{\text{reg}}(M_{\text{reg}}^{c_n}))$  and  $\phi(\mathbf{x}_{t_n})$  being the  $M_{\text{reg}}^{c_n} \times B$ -matrix with  $[\phi(\mathbf{x}_{t_n})]_{i,j} = \phi_j(x_{t_n}^{\text{reg}}(j))$ . It is a well-known fact (see e.g., [23]) that  $\hat{\alpha}_{t_n}$  is given by

$$\hat{\alpha}_{t_n} = (\phi(\mathbf{x}_{t_n})^T \phi(\mathbf{x}_{t_n}))^{-1} \phi(\mathbf{x}_{t_n})^T \mathbf{y}_{t_n}, \tag{4.4}$$

where we note that, if we choose linearly independent basis functions, matrix inversion in (4.4) exists almost surely since  $X_{t_n}$  has a density<sup>9</sup> We define the estimator

$$\hat{v}_{B,K}(t_n, \cdot) = \sum_{b=0}^B \hat{\alpha}_{t_n}^{(b)} \phi_b(\cdot). \tag{4.5}$$

If  $M_{\text{reg}}^{c_n} = 0$  we know that all samples are in the exercise region and we simply set  $\hat{v}_{B,K}(\cdot) \equiv g_{t_n}(\cdot)$ . Since the LSM is one of the most established algorithms for valuation of Bermudan options, the theoretical properties are extensively studied and many of the results can also be applied to the algorithm above. We assume that  $X_{t_n}$  has non-zero density in  $C_n$ , implying that  $M_{\text{reg}}^{c_n}$  approaches infinity when  $M_{\text{reg}}$  approaches infinity, almost surely. Furthermore, since the regression pairs are independently and identically distributed, it holds that  $\hat{v}_{B,K}(t_n, X_{t_n})$  converges both in mean square and in probability to  $v_B(t_n, X_{t_n})$  as  $M_{\text{reg}}$  approaches infinity (see e.g., [42]). To make it more clear when comparing the OLS-approximator of the option value to the neural network approximator (to be defined in the next section), we use the following notation

$$\hat{v}_{t_n}^{\text{OLS}}(\cdot) = \hat{v}_{B,M}(t_n, \cdot), \tag{4.6}$$

where we assume that  $B$  and  $M$  are chosen such that both accuracy and time complexity are taken into account.

A nice property of OLS regression is that, given  $B$  and a sample of regression pairs, we have a closed-form expression for the optimal parameters and thus also the regression function (4.5). On the other hand, we may face memory or runtime issues for large  $B$  and  $M_{\text{reg}}^{c_n}$  due to (4.4).

### 4.3. Neural network regression

In this section we present, a simple neural network approximation of  $V_{t_n}(\cdot)$ . The neural network is a mapping,  $v_{\varphi_n} : \mathbb{R}^d \rightarrow \mathbb{R}$ , parametrized by the  $p_{t_n} \in \mathbb{N}$  trainable parameters  $\varphi_n \in \mathbb{R}^{p_{t_n}}$ . The objective is to find  $\varphi_n$  such that the empirical  $L_2$ -risk

$$\frac{1}{M_{\text{reg}}^{c_n}} \sum_{m=1}^{M_{\text{reg}}^{c_n}} |v_{\varphi_n}(x_{t_n}^{\text{reg}}(m)) - y_{t_n}^{\text{reg}}(m)|^2 \tag{4.7}$$

is minimized. We denote by  $\hat{\varphi}_n$  an optimized version of  $\varphi_n$  and define our neural network approximator of the option price at  $t_n$  by

$$\hat{v}_{t_n}^{\text{NN}}(\cdot) = v_{\hat{\varphi}_n}(\cdot).$$

To avoid repetition, the description of the neural networks in Section 3.1.1 is also valid for the neural network used here. However, one important difference is the output, which in this section is an approximation of the option value, and should hence take on values in  $(0, \infty)$ . A natural choice as activation function in the output layer is therefore  $\text{ReLU}(\cdot) = \max\{0, \cdot\}$ . Furthermore, by shifting the output with  $-g_{t_n}(\cdot)$ , i.e., designing the neural network to output  $v_{\hat{\varphi}_n}(\cdot) - g_{t_n}(\cdot)$  and defining  $\hat{v}_{t_n}^{\text{NN}}(\cdot) = v_{\hat{\varphi}_n}(\cdot) + g_{t_n}(\cdot)$  we can, for all  $x \in \mathbb{R}^d$ , enforce  $\hat{v}^{\text{NN}}(x) \geq g_{t_n}(x)$  by using ReLU as activation function in the output layer. In many cases it seems to be beneficial to use the identity as the activation function in the output layer. This could possibly be explained by the fact that when using the ReLU as activation function, the gradient of the loss function, (4.7) (with respect to the input) may vanish during training. This, in turn leads to an inefficient use of a gradient descent type algorithm for the optimization problem.

Another difference, which has to do with the training phase, is that the optimization of the parameters  $\varphi_n$  does not have to be carried out recursively. This opens up the possibility for parallelization of the code.

By comparing (4.7) to (4.3) we see that the optimization problems are similar. There are, however, some major differences. In Section 4.2, we have a closed-form expression for the optimal parameters resulting in the final regression function (4.5). This is not the case for the neural network regression and we therefore need to use an optimization algorithm to approximate the optimal parameters. On the other hand, as mentioned in Section 4.2 it is sometimes hard to find basis functions that are flexible enough. This problem can be overcome with neural networks, which are known to be good global approximators.

<sup>9</sup> In practice we run into troubles if we choose  $B$  too high since the approximation of the matrix inverse may be unstable.

### 5. Approximation algorithms for exposure profiles

In this section, we introduce different ways to estimate (2.6) and (2.7) relying on Monte-Carlo sampling and the approximation algorithms described in Sections 3 and 4. Furthermore, a simple example is presented and visualized, which aims to provide an intuitive understanding of the different methods. Finally, the advantages and disadvantages of each method are presented in a table.

In this section, the neural network-based approximation of the value function of the option, introduced in Section 4.3 is used. However, it would have been possible to use the OLS-based approximation from Section 4.2, instead.

We use  $M \in \mathbb{N}$  independent realizations of  $X$ , which for  $m \in \{1, 2, \dots, M\}$  are denoted by  $x(m) = (x_t(m))_{n=0}^N$  for  $t \in \mathbb{T}$ . When  $X$  is given by (2.13), realization  $m$  is denoted by  $\tilde{x}^{t_n}(m) = (\tilde{x}_t^{t_n}(m))_{t \in [t_0, t_N]}$ , where we recall that superscript  $t_n$  refers to the point in time where the discontinuity of the drift coefficient is located. We introduce the following two approximations of the expected exposure under the risk neutral measure

$$\hat{E}^1_{\mathbb{Q}}(t_n) = \frac{1}{M} \sum_{m=1}^M \hat{v}_{t_n}^{NN}(x_{t_n}(m)) \mathbb{I}_{\{\tau(f^\theta) > t_n\}}, \tag{5.1}$$

$$\hat{E}^2_{\mathbb{Q}}(t_n) = \frac{1}{M} \sum_{m=1}^M e^{-r(\tau_n(f^\theta) - t_n)} g_{\tau_n(f^\theta)} \left( x_{\tau_n(f^\theta)}(m) \right) \mathbb{I}_{\{\tau(f^\theta) > t_n\}}. \tag{5.2}$$

For the expected exposure under the real world measure, we have the following three approximations

$$\hat{E}^1_{\mathbb{P}}(t_n) = \frac{1}{M} \sum_{m=1}^M \hat{v}_{t_n}^{NN}(\tilde{x}_{t_n}^{t_n}(m)) \mathbb{I}_{\{\tau(f^\theta) > t_n\}}, \tag{5.3}$$

$$\hat{E}^2_{\mathbb{P}}(t_n) = \frac{1}{M} \sum_{m=1}^M e^{-r(\tau_n(f^\theta) - t_n)} g_{\tau_n(f^\theta)} \left( \tilde{x}_{\tau_n(f^\theta)}^{t_n}(m) \right) \mathbb{I}_{\{\tau(f^\theta) > t_n\}}, \tag{5.4}$$

$$\hat{E}^3_{\mathbb{P}}(t_n) = \frac{1}{M} \sum_{m=1}^M e^{-r(\tau_n(f^\theta) - t_n)} g_{\tau_n(f^\theta)} \left( x_{\tau_n(f^\theta)}(m) \right) \mathbb{I}_{\{\tau(f^\theta) > t_n\}} l(x_{t_n}(m), \dots, x_{t_0}(m)), \tag{5.5}$$

where  $l$  is the likelihood ratio function defined in Theorem 2.2. We note that (5.1) and (5.3) are the only approximations that require a calculation of the option value. On the other hand, we need  $X$  to be described by a diffusion-type SDE in (5.4) and we need to know the density functions to calculate (5.5). To define the approximations of the

potential future exposure, we start by defining the order statistic of  $\left( \hat{v}_{t_n}^{NN}(x_{t_n}(m)) \mathbb{I}_{\{\tau(f^\theta) > t_n\}} \right)_{m=1}^M$ , i.e., the vector given

by  $\left( \hat{v}_{t_n}^{NN}(x_{t_n}(\tilde{m}_1)) \mathbb{I}_{\{\tau(f^\theta) > t_n\}}, \dots, \hat{v}_{t_n}^{NN}(x_{t_n}(\tilde{m}_M)) \mathbb{I}_{\{\tau(f^\theta) > t_n\}} \right)$  satisfying  $\hat{v}_{t_n}^{NN}(x_{t_n}(\tilde{m}_i)) \leq \hat{v}_{t_n}^{NN}(x_{t_n}(\tilde{m}_j))$  whenever  $i \leq j$ . Furthermore, we define

$$i_\alpha = \begin{cases} \lceil \alpha M \rceil, & \text{for } \alpha \geq 0.5, \\ \lfloor \alpha M \rfloor, & \text{for } \alpha < 0.5. \end{cases}$$

The approximations of the potential future exposure are then defined as

$$\hat{P}\hat{E}^{\alpha}_{\mathbb{Q}}(t_n) = \hat{v}_{t_n}^{NN}(x_{t_n}(\tilde{m}_{i_\alpha})) \mathbb{I}_{\{\tau(f^\theta) > t_n\}},$$

$$\hat{P}\hat{E}^{\alpha}_{\mathbb{P}}(t_n) = \hat{v}_{t_n}^{NN}(\tilde{x}_{t_n}^{t_n}(\tilde{m}_{i_\alpha})) \mathbb{I}_{\{\tau(f^\theta) > t_n\}}.$$

In Table 1, some characteristics of the calculations needed for each approximation are given.

To explain the different approximations in a concrete setting we turn to a simple example.

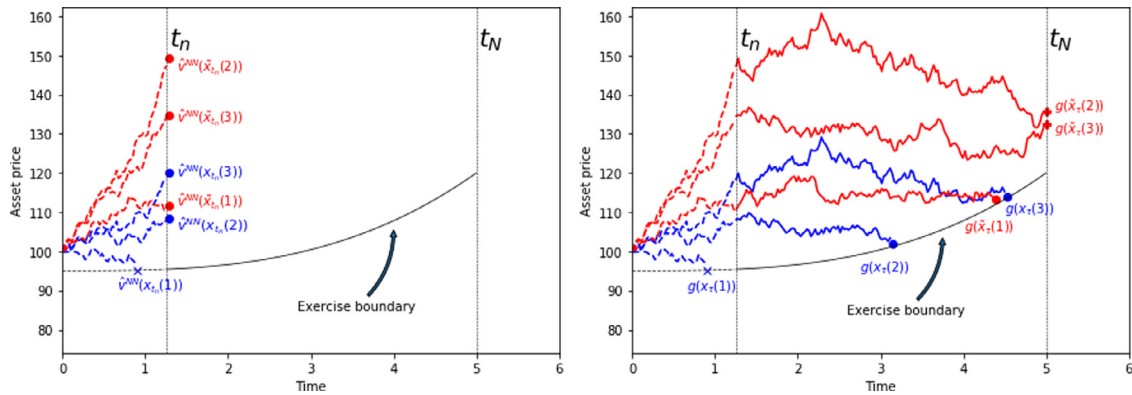
**Example 5.1.** Consider a one-dimensional American put option, where we for simplicity assume that  $r = 0$ . We are interested in the expected exposure and the potential future exposure at time  $t_n \in (t_0, t_N)$  given that we have full knowledge of the market at time  $t_0$ . We give a short explanation of the intuition behind the different methods by referring to Fig. 2, where the problem is visualized.

We start by  $\hat{E}^1_{\mathbb{Q}}(t_n)$  and  $\hat{E}^1_{\mathbb{P}}(t_n)$  for which we only use the figure to the left. For  $\hat{E}^1_{\mathbb{Q}}(t_n)$  we follow the blue samples and note that samples 2 and 3 are not exercised prior to, or at  $t_n$ , which means that the indicator function in (5.1) equals 1.

**Table 1**

A specification of which approximations/entities that are required in order to carry out the different calculations. If required ✓, otherwise ✖.

	Specification of requirements for each approximation			
	Approx- imation of the functional form $V_n(\cdot)$	Sampling from $\tilde{X}^{t_n}$	Known density functions	$X$ must be given by diffusion-type SDE
$\hat{E}_Q^1$	✓	✖	✖	✖
$\hat{E}_Q^2$	✖	✖	✖	✖
$\hat{E}_P^1$	✓	✓	✖	✖
$\hat{E}_P^2$	✖	✓	✖	✓
$\hat{E}_P^3$	✖	✖	✓	✖
$P\hat{F}E_Q$	✓	✖	✖	✖
$P\hat{F}E_P$	✓	✓	✖	✖



**Fig. 2.** Blue trajectories are distributed as  $X$  and red trajectories are distributed as  $\tilde{X}^{t_n}$ . The boundary for immediate exercise is pointed out and should be interpreted as, as soon a trajectory touches the boundary, the option is exercised and the holder receives the immediate pay-off. Recall that the exercise boundaries are calculated in order to be optimal under the  $\mathbb{Q}$ -measure. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Sample 1, on the other hand, touches the exercise region prior to  $t_n$  and has therefore already been exercised, which means that

$$\begin{aligned} \hat{E}_Q^1(t_n) &= \frac{1}{3} \left( \hat{v}^{NN}(x_{t_n}(1)) \mathbb{I}_{\{\tau(f^\theta) > t_n\}} + \hat{v}^{NN}(x_{t_n}(2)) \mathbb{I}_{\{\tau(f^\theta) > t_n\}} + \hat{v}^{NN}(x_{t_n}(3)) \mathbb{I}_{\{\tau(f^\theta) > t_n\}} \right) \\ &= \frac{1}{3} \left( \hat{v}^{NN}(x_{t_n}(2)) + \hat{v}^{NN}(x_{t_n}(3)) \right). \end{aligned}$$

When focusing on the red samples instead, we see that no sample touches the exercise region prior to  $t_n$  and we obtain

$$\hat{E}_P^1(t_n) = \frac{1}{3} \left( \hat{v}^{NN}(\tilde{x}_{t_n}^1) + \hat{v}^{NN}(\tilde{x}_{t_n}^2) + \hat{v}^{NN}(\tilde{x}_{t_n}^3) \right).$$

Similarly, we can e.g., state that  $P\hat{F}E_Q^{2.5} = 0$  and  $P\hat{F}E_P^{97.5} = \hat{v}^{NN}(\tilde{x}_{t_n}^2)$ .

Moving on to  $\hat{E}_P^2(t_n)$  and  $\hat{E}_P^3(t_n)$ , we shift focus to the figure to the right. For  $\hat{E}_P^2(t_n)$  we want to use the red samples and notice that samples 2 and 3 end up out of the money. We therefore obtain

$$\hat{E}_P^2(t_n) = \frac{1}{3} g_{\tau_n}(f_n^\theta) \left( \tilde{x}_{\tau_n}^{t_n}(f_n^\theta)(1) \right).$$

For  $\hat{E}_P^3(t_n)$ , we instead consider the blue samples and see that sample 1 is exercised prior to  $t_n$  and samples 2 and 3 end up in the money. However, we also need to adjust the estimate for using the wrong state process<sup>10</sup> This is done by multiplying each term with the likelihood ratios  $l(x(2))$  and  $l(x(3))$  to finally obtain

$$\hat{E}_P^3(t_n) = \frac{1}{3} \left( g_{\tau_n}(f_n^\theta) \left( x_{\tau_n}(f_n^\theta)(2) \right) l(x(2)) + g_{\tau_n}(f_n^\theta) \left( x_{\tau_n}(f_n^\theta)(3) \right) l(x(3)) \right).$$

<sup>10</sup> The samples are generated from the state process under the  $\mathbb{Q}$ -measure between  $t_0$  and  $t_n$  which, if not corrected, would be in conflict with the definition of  $\hat{E}_P(t_n)$ .

The last estimate,  $\hat{E}_{\mathbb{Q}}^2(t_n)$ , is obtained by removing the likelihood ratios from the estimate for  $\hat{E}_{\mathbb{P}}^3(t_n)$ .

To conclude, we note that Fig. 2 displays, to the left, the cases where functional form approximations of the option values are used and to the right the cases where cashflow-paths are used (this can be read in Table 1). Furthermore, blue and red trajectories are distributed according to  $X$  and  $\tilde{X}^{tn}$ , respectively (this can also be read in Table 1).

### 6. Numerical results

In this section, the proposed algorithms are compared with two different Monte-Carlo-based algorithms for a two-dimensional case. We focus on both the accuracy of the computed exercise boundaries and the exposure profiles. Furthermore, the exercise boundary is compared with the exercise boundary for the corresponding American option, computed by a finite element method, from the PDE-formulation of the problem. Exposure profiles are then computed both under the risk neutral and the real world measures for problems in 2 and 30 dimensions. Finally, we compare the OLS-regression with the NN-regression.

The underlying model for the risk factors is in many cases the Black–Scholes model but we will show that the algorithms also perform well with more complicated dynamics, like the Heston model stochastic volatility dynamics.

In all neural networks used in this paper we use three hidden layers with 30 nodes in each. We use ReLU as the activation function in the hidden layers and Adam as the optimization algorithm with a constant learning rate of 0.0005.

#### 6.1. Black–Scholes dynamics

In the Black–Scholes setting the only risk factors are the  $d \in \mathbb{N}$ , underlying assets, denoted by  $S$ . We assume a constant risk-free interest rate  $r \in \mathbb{R}$ , and for each asset  $i \in \{1, 2, \dots, d\}$ , volatility  $\sigma_i \in (0, \infty)$ , and constant dividend rate  $q_i \in (0, \infty)$ . The state process  $X$  is then given by the asset process  $S = (S_t)_{t \in [t_0, t_N]}$ , i.e.,  $X = S$ , following the SDE

$$\frac{d(S_t)_i}{(S_t)_i} = (A_i - q_i)dt + \sigma_i d(W_t^{\mathbb{A}})_i, \quad (S_{t_0})_i = (s_{t_0})_i; \quad t \in [t_0, t_N], \tag{6.1}$$

with initial state  $(s_{t_0})_i \in (0, \infty)$ , and where  $\mathbb{A}$  is either the real world measure  $\mathbb{P}$  or the risk neutral measure  $\mathbb{Q}$ . In the real world setting  $A_i = \mu_i \in \mathbb{R}$  and in the risk neutral setting  $A_i = r$ . The process  $W^{\mathbb{A}} = (W_t^{\mathbb{A}})_{t \in [t_0, t_N]}$  is a  $d$ -dimensional Brownian motion under the measure  $\mathbb{A}$ . Furthermore,  $W^{\mathbb{A}}$  is correlated with correlation parameters  $\rho_{ij} \in [-1, 1]$ , i.e., for  $i, j \in \{1, 2, \dots, d\}$ , we have that  $\mathbb{E}^{\mathbb{A}}[d(W_t^{\mathbb{A}})_i d(W_t^{\mathbb{A}})_j] = \rho_{ij} dt$ , with  $\rho_{ii} = 1$ . Moreover, for  $t_0 \leq u \leq t \leq t_N$  a closed-form solution to (6.1) is given by

$$(S_t)_i = (S_u)_i \exp\left( (A_i - q_i - \frac{1}{2}\sigma_i^2)(t - u) + \sigma_i((W_t^{\mathbb{A}})_i - (W_u^{\mathbb{A}})_i) \right). \tag{6.2}$$

We note that  $\log S$  has Gaussian increments under both the  $\mathbb{P}$ - and the  $\mathbb{Q}$ -measures which implies that we have access to a closed-form expressions for the transition densities of  $S$ , and in turn also to the likelihood ratio in (5.5).

#### 6.1.1. Bermudan max-call option

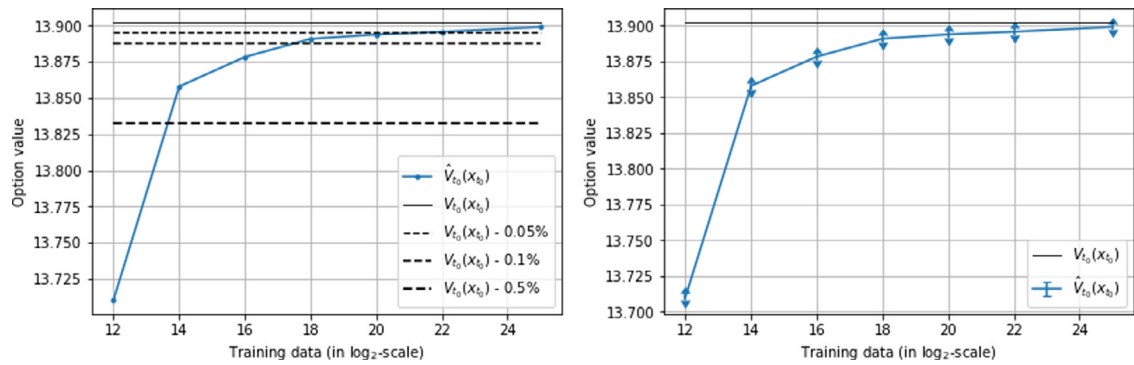
At exercise, a Bermudan max-call option pays the positive part of the maximum of the underlying assets after subtraction of a fixed amount  $K \in \mathbb{R}$ . This implies an identical pay-off function at all exercise dates, given by

$$g(s) = (\max\{s_1, s_2, \dots, s_d\} - K)^+,$$

where  $s = (s_1, s_2, \dots, s_d) \in (0, \infty)^d$  and for  $c \in \mathbb{R}$ ,  $(c)^+ = \max\{c, 0\}$ .

We choose to focus on Bermudan max-call options for two reasons: first, there exist plenty of comparable results in the literature (see e.g., [5,10,29]); second, and more importantly, the exercise region is nontrivial with several interesting features. For example,  $\max\{s_1, s_2, \dots, s_d\}$  is not a sufficient statistic for making optimal exercise decisions, meaning that we cannot easily reduce the dimensionality (all dimensions are needed in order to price the option). This is not the case with, e.g., the geometric-average call option with pay-off function  $g(s) = \left(\frac{1}{d} \prod_{i=1}^d s_i - K\right)^+$ , since  $\frac{1}{d} \prod_{i=1}^d s_i$  is a sufficient statistic for optimal exercise decisions (only the geometric average is needed to price the option, meaning that the problem can be reduced to 1 dimension, see e.g., [19]). Another example is the arithmetic-average call option with pay-off function  $g(s) = \left(\frac{1}{d} \sum_{i=1}^d s_i - K\right)^+$ . Similar to the max-call option,  $\frac{1}{d} \sum_{i=1}^d s_i$  is not a sufficient statistic for optimal exercise decisions, but on the other hand the exercise region is convex<sup>11</sup> in the underlying assets for fixed  $t$ . (see [12, Proposition 6.1]). Convexity of the exercise region does not hold for the max-call option, making it hard to capture the exercise region when global polynomials are used as basis functions in e.g., the LSM. Methods which instead rely on local regression can, to some extent, overcome this problem but it is difficult to decide how the localization (usually localization of the state space) should be done, especially in high dimensions.

<sup>11</sup> In this section we have assumed that results for exercise regions for American options also hold for their Bermudan counterparts.



**Fig. 3.** Convergence of approximate option values in the amount of training data. Reference value 13.902, computed by a binomial lattice model in [2]. **Left:** With different levels of deviation from the reference value. **Right:** With empirical 95%-confidence intervals of sample mean.

In the numerical experiments we use the following parameters  $r = 0.05$  and for  $i, j \in \{1, 2, \dots, d\}$ ,  $q_i = 0.1$ ,  $\sigma_i = 0.2$ , for  $i \neq j$ ,  $\rho_{ij} = 0$ ,  $N = 9$ ,  $t_0 = 0$ ,  $t_N = 3$ ,  $(s_{t_0})_i = 100$  and  $K = 100$ . We want to emphasize that the choice of having no correlation between assets is due to the fact that this case has been studied thoroughly in the literature (see e.g., [5,10,12,29]). To verify that the algorithm is also able to tackle problems with correlated assets, we replicated an experiment in [33], of pricing a put option on the arithmetic average of 5 correlated assets and obtained the same price<sup>12</sup>

6.1.2. Approximation of the option value at initial time

The performance of the DOS algorithm is thoroughly explored for a wide range of different examples in [5]. However, the convergence with respect to the amount of training data used,  $M_{\text{train}}$  was not given. We therefore present, in Fig. 3, an example of how the option price at  $t_0$  is converging to a reference value in terms of the amount of training data. In the considered example, we use the parameter values given at the end of Section 6.1.1 with  $d = 2$ , i.e., a two-dimensional Bermudan max-call option. The reference value (13.902) for  $V_{t_0}(x_{t_0})$  is computed by a binomial lattice model in [2]. The DOS approximation, denoted by  $\hat{V}_{t_0}(x_{t_0})$ , is computed according to (3.4). To be more specific, one set of neural networks is trained (one at each time-point) for each  $M_{\text{train}} \in \{2^{12}, 2^{14}, 2^{16}, 2^{18}, 2^{20}, 2^{22}, 2^{25}\}$ , as described in Section 3.1.2. For each  $M_{\text{train}}$ , the option value,  $V_{t_0}(x_{t_0})$  is computed 20 times (with 20 independent realizations of  $X$ ) with  $M_{\text{val}} = 2^{20}$  and the average of these 20 values is presented in Fig. 3. Furthermore, the figure to the right displays empirical 95%-confidence intervals of the sample mean, which is computed by adding and subtracting  $\frac{1.96}{\sqrt{19}}$  times the sample standard deviation. Note that the empirical confidence intervals are related to the variability within the 20 Monte-Carlo valuations and not to the variability between different executions of the training algorithm. Therefore, they stay roughly constant for different sizes of the training data.

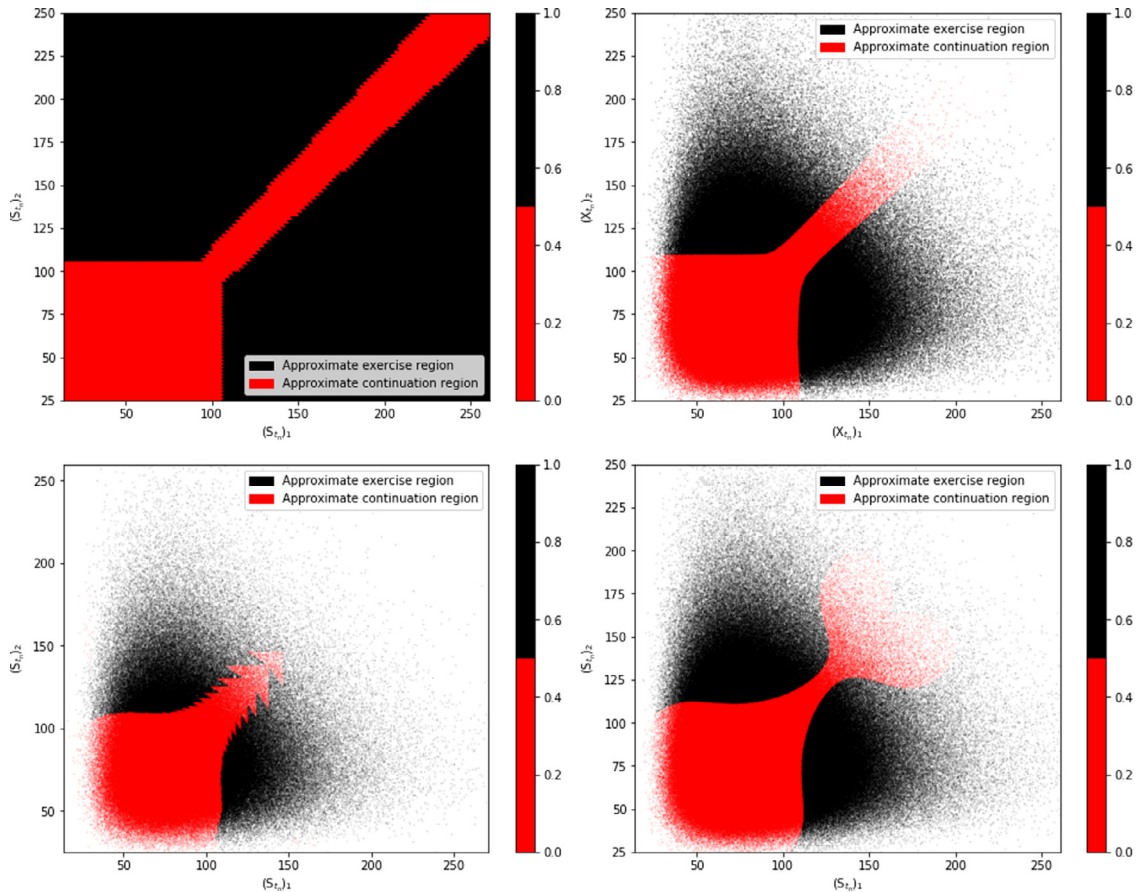
6.1.3. Comparison with Monte-Carlo-based algorithms

We start with a short introduction of the two Monte-Carlo-based algorithms with which we compare results for the two-dimensional max-call option.

The least squares method (LSM) proposed in [34], is one of the most used methods for pricing American/Bermudan options. The method approximates exercise boundaries and computes the option value from discounted cashflows. The regression is carried out globally, i.e., with the same regression function over the entire state space. However, if one is only interested in the option value at  $t_0$ , it is beneficial to only consider ITM-samples in the regression state. This is done when the exercise boundary, and  $\text{PFE}_{97.5}$  are approximated. For approximating EE and  $\text{PFE}_{2.5}$  we need the entire distribution of future option values forcing us to also include OTM-samples in the regression. We use as basis functions the first 6 Laguerre polynomials  $L_n(x) = e^{-x/2} \frac{e^x}{n!} \frac{d^n}{dx^n} (x^n e^{-x})$  of  $(S_{t_n})_1$  and  $(S_{t_n})_2$  and the first 4 powers of  $\max\{\log((S_{t_n})_1), \log((S_{t_n})_2)\}$  for all  $t_n \in \mathbb{T}$ . Note that since we have no correlation between  $(S)_1$  and  $(S)_2$ , we do not include cross-terms in the basis.

The second algorithm is the stochastic grid bundling method (SGBM) proposed in [29], in which regression is carried out locally in so-called bundles. In the SGBM the target function in the regression phase is the option value which makes it suitable for approximations of exposure profiles. There are several papers on computations of exposure profiles with the SGBM, see e.g., [14]. At each exercise date  $t_n \in \mathbb{T}$ , we use as basis functions: a constant, the first 4 powers of  $(S_{t_n})_1$  and  $(S_{t_n})_2$ , and the first 2 powers of  $\max\{\log((S_{t_n})_1), \log((S_{t_n})_2)\}$ . Furthermore, the state space is divided into 32 equally-sized bundles based on  $\max\{(S_{t_n})_1, (S_{t_n})_2\}$ . The reason for using different basis functions for LSM and SGBM is that regression is carried out locally in each bundle for SGBM, while it is carried out globally in the LSM. This means that we need a more flexible basis for the LSM in order to capture the global behaviour of the sought function.

<sup>12</sup> We obtained the price 0.1804, which is the same price up to the given accuracy of 4 digits, presented in [33, Parameters as in Set II, results in Table 3 on p. 22].



**Fig. 4.** Approximate exercise boundaries for a two-dimensional max-call option at  $t_8 \approx 2.67$ . **From top left to bottom right:** FEM (American option), DOS, SGBM and LSM respectively.

Before presenting any results, we recall from Eqs. (2.6) and (2.7) that the expected exposure and the potential future exposure are two statistics of

$$V_{t_n}(S_{t_n})\mathbb{I}_{\{\tau > t_n\}}, \tag{6.3}$$

where  $\tau$  is an  $S$ -stopping time. This means that approximations of the exposure profiles are sensitive to, not only the option value itself, but also the exercise strategy. The SGBM and LSM only compute the exercise strategy implicitly, i.e., by comparing the approximate option value and the immediate pay-off. Therefore small errors of the approximate option value close to the exercise boundary can lead to significant errors in the exercise strategy. We therefore start by presenting a comparison of the exercise boundaries. As a reference, we use the exercise boundary for the corresponding American option, which is computed from the PDE-formulation with the finite element method used in [3]. We note that since the PDE formulation refers to the American option, the exercise boundary differs slightly<sup>13</sup> from the exercise boundary of the Bermudan counterpart, which we are interested in.

In Fig. 4, a comparison of the exercise boundaries at  $t_8 \approx 2.67$  for the different algorithms is presented. As we can see, the DOS algorithm captures the shape of the exercise regions while both the SGBM and the LSM seem to struggle, especially with the part of the continuation region along (and around) the line  $(S_{t_n})_1 = (S_{t_n})_2$ , in particular for high values of  $(S_{t_n})_1$  and  $(S_{t_n})_2$ . The irregularly shaped features in the continuation region for the SGBM are a consequence of the local regression. In particular, the triangular shapes come from the specific choice of bundling rule (bundling based on  $\max\{(S_{t_n})_1, (S_{t_n})_2\}$ ).

Moving on to the exposure profiles, we see in Fig. 5, that even though the DOS algorithm and the SGBM seem to agree on the exposure profile in general, we notice a difference in the  $PFE_{97.5}$ . This is a consequence of the slight bias towards classifying samples as belonging to the continuation region, which is shown in Fig. 5, to the right.

The LSM is performing worse, both in terms of accuracy of exposure profiles and bias towards miss-classification. This is however not a surprise, since the LSM is tailored to calculate the option value at  $t_0$ .

<sup>13</sup> In fact, the continuation region for an American option is a subset of the continuation region for the Bermudan counterpart.



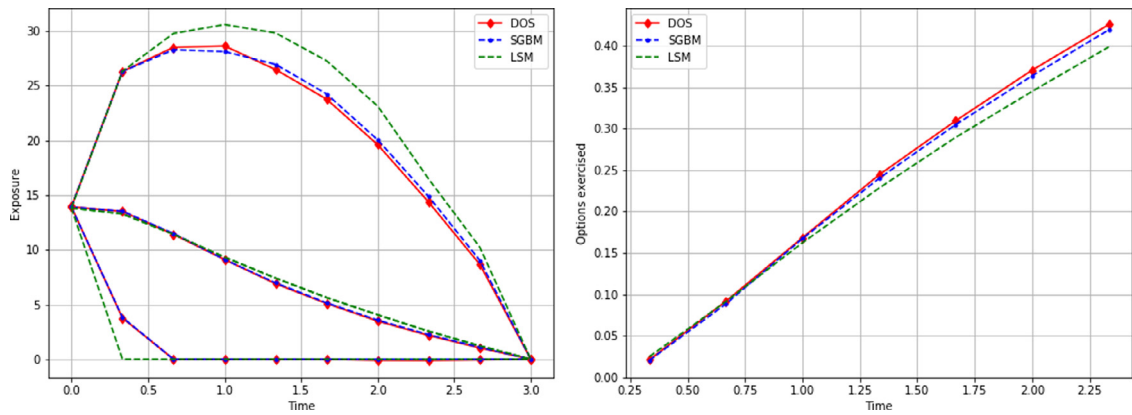


Fig. 5. Comparison of DOS, SGBM and LSM for a two-dimensional max-call option. **Left:** Expected exposure and potential future exposures at 97.5%- and 2.5%-levels. **Right:** Proportion of options exercised at different exercise dates.

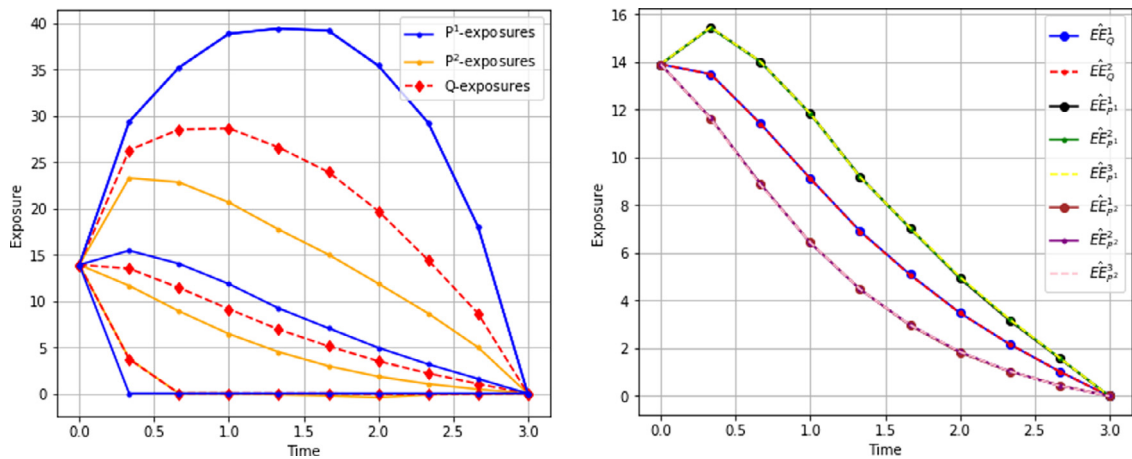


Fig. 6. Approximate exposures, at the exercise dates over the life time of the contract. **Top:** Case I. **Bottom:** Case II. **Left:** For  $i = 1$  and  $i = 2$ ,  $\hat{P}E_{P^1}^{97.5}$ ,  $\hat{P}E_Q^{97.5}$ ,  $\hat{E}E_{P^1}^1$ ,  $\hat{E}E_Q^1$ ,  $\hat{P}E_{P^1}^{2.5}$  and  $\hat{P}E_Q^{2.5}$ . **Right:** For  $i = 1$  and  $i = 2$ ,  $\hat{E}E_Q^1$ ,  $\hat{E}E_Q^2$ ,  $\hat{E}E_{P^1}^1$ ,  $\hat{E}E_{P^1}^2$  and  $\hat{E}E_{P^1}^3$ .

Finally, it should be pointed out that for both the SGBM and LSM, it could very well be the case that another set of basis functions would better capture the shape of the exercise boundaries. In this two-dimensional example one could probably use geometric intuition to come up with a better set of basis functions, but in higher dimensions, and for more complicated pay-off functions, this becomes difficult.

#### 6.1.4. Exposure profiles under different measures

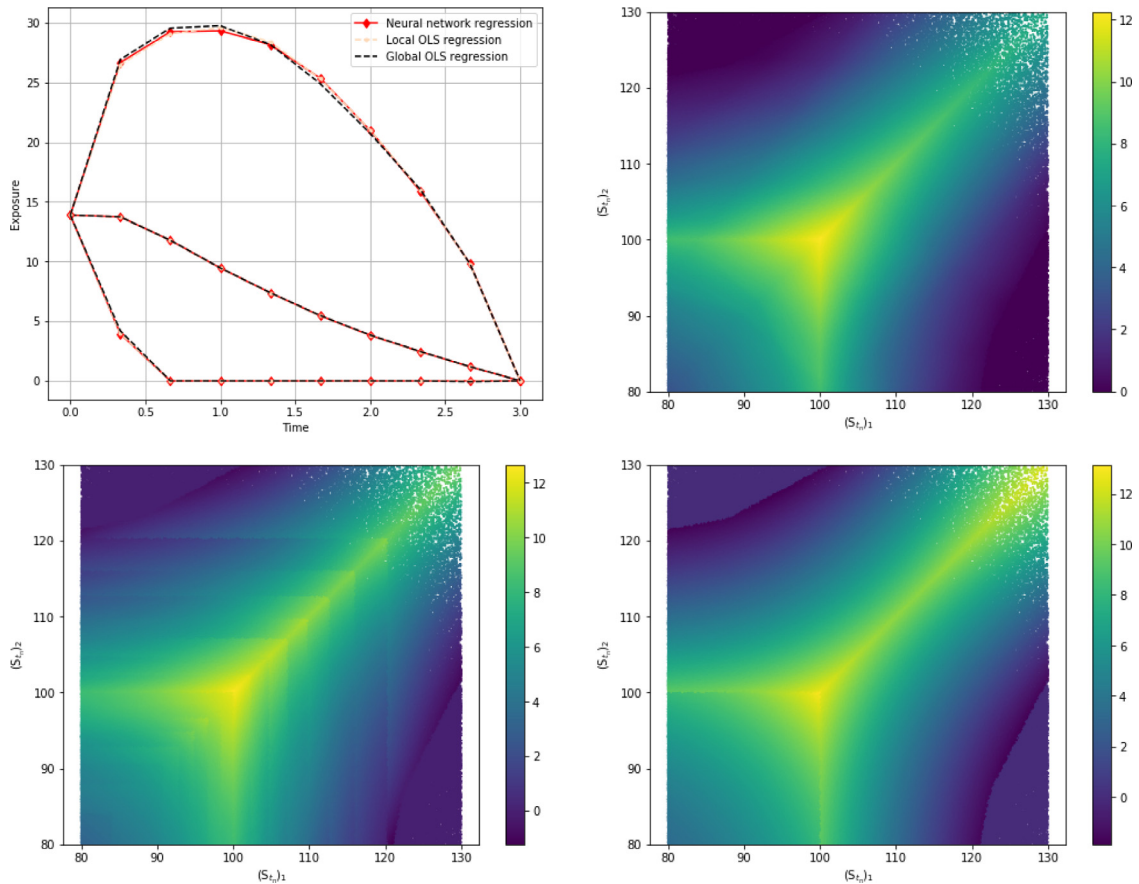
In this section we compare exposure profiles under different measures for the max-call option, in 2 and 30 dimensions. In Case I we set  $d = 2$  and  $\mathbb{P}^1$  and  $\mathbb{P}^2$  such that for  $i \in \{1, 2\}$ , we have drifts  $(\mu_1)_i = 15\%$  and  $(\mu_2)_i = -5\%$ . In Case II we set  $d = 30$  and  $\mathbb{P}^1$  and  $\mathbb{P}^2$  such that for  $i \in \{1, \dots, 30\}$ , we have drifts  $(\mu_1)_i = 7.5\%$  and  $(\mu_2)_i = 2.5\%$ .

Fig. 6 shows exposure profiles in 2 and 30 dimensions on the left side. On the right, we see a comparison of the different ways to compute expected exposures which all agree to high accuracy.

#### 6.1.5. Comparison of the OLS-regression and the NN-regression for approximation of pathwise option values

Finally, we compare the performance of the OLS-regression, with the NN-regression, introduced in Subsections 4.2 and 4.3. We emphasise that both OLS-regression and NN-regression are regressing the current state on discounted cashflow paths approximated by the DOS algorithm (described in Section 3). They can therefore be seen as phase 2 of an algorithm generating pathwise option values.

After conducting numerical experiments on a variety of different examples we conclude that the expected exposures are very similar for the two regression methods. However, the potential future exposure is not always captured by the OLS-regression. The difficulty lies in finding a set of computationally feasible basis functions, flexible enough to accurately capture a complicated function surface on a large domain (similar problem as for the LSM in Section 6.1.3). To overcome this problem, we also implement a slightly different version of the algorithm, where we instead carry out local regression in



**Fig. 7.** Comparison of NN-regression, local OLS-regression and global OLS-regression for a two-dimensional max-call option. For all three regression techniques, the DOS is used to approximate the optimal stopping strategy. **Top left:** Exposure profiles. **From top right to bottom left:** Approximate risk premium for holding option instead of immediate exercise at  $t_2 \approx 0.667$ , i.e.,  $v_{t_2}^Z(\cdot) - g_{t_2}(\cdot)$ , with  $Z$  representing NN, OLS<sub>loc</sub> and OLS<sub>glob</sub>, respectively.

the continuation regions. To be able to differentiate between the local and global OLS-regressions, we denote the regression functions by  $v^{OLS_{loc}}$  and  $v^{OLS_{glob}}$ , respectively. The localization procedure is done similarly as in the SGBM, i.e., at each  $t \in \mathbb{T}$ , the state space is divided into bundles of equal size (in terms of the number of samples in each bundle), based on  $\max\{(S_t)_1, (S_t)_2\}$ . With local OLS-regression we obtain almost identical exposure profiles as with NN-regression. In Fig. 7, on top to the left, the exposure profiles computed with the three different algorithms are displayed. Furthermore, from top right to bottom right, we compare the approximate risk premia for holding instead of immediately exercising the option at  $t_2 \approx 0.667$  and some  $x \in \mathbb{R}^2$ , i.e.,  $v_{t_2}^Z(x) - g_{t_2}(x)$ , with  $Z$  representing, in order NN, OLS<sub>glob</sub> and OLS<sub>loc</sub>. We know that for all  $x \in \mathbb{R}^2$ , it holds that  $V_{t_2}^Z(x) - g_{t_2}(x) \geq 0$ . We see in Fig. 7, top right, that this is captured by the NN-regression, since the values range from 0 to just above 12. When we carefully evaluate the values of the risk premia computed with local and global OLS-regression (Fig. 7 bottom left and bottom right) we see that negative values exist in both plots. We see similar phenomena for high values, i.e., the range is stretched upwards in comparison to the values obtained with NN-regression. The reason for the negative values is that  $v^{OLS_{loc}}$  and  $v^{OLS_{glob}}$  underestimate the option values close to the boundary (which in this case coincides with the exercise boundary since the regression is carried out only in the continuation region). To compensate for this, we see a tendency of higher values in the center (not close to the exercise boundaries) of the continuation regions. As a final remark, we note that this behaviour is reduced by using local regression instead of global regression (the range of values in Fig. 7 is tighter for local than global regression). On the other hand, we note discontinuities in Fig. 7 bottom, left, stemming from the localized regression, of the risk premium computed with the local OLS-regression. The discontinuity in Fig. 7, bottom right, is a boundary issue of the OLS-regression (regression is only carried out in the continuation region since we set the value equal to the immediate pay-off in the exercise region).

Even though, the local OLS-regression is less accurate when it comes to computations of exposure profiles than the other two algorithms in this section, it is clear by comparing Figs. 5 and 7, that it outperforms the LSM. This is not a surprise since:

1. The accumulated error in the LSM (because of recursive dependency of the regression functions) is significantly reduced since the regression functions are not sequentially dependent. The discounted cashflows are projected onto the risk factors directly. This implies that the only error accumulation (over time), in the OLS-regression originates from the DOS algorithm, which computes the exercise boundaries with high accuracy;
2. By recalling equation (6.3), we note that a less accurate stopping strategy may cause a less accurate exposure.

### 6.2. Heston model dynamics

In this section we assume a one-dimensional underlying asset following the Heston stochastic volatility model [25], which is considered under the risk neutral measure. (We omit the explicit notation of the probability measure used in this section.) In this setting, the market is described by, the underlying asset price process  $S = (S_t)_{t \in [t_0, t_N]}$ , and the instantaneous variance process  $\nu = (\nu_t)_{t \in [t_0, t_N]}$ . The state process is the two-dimensional process  $X = (\nu, S)$ , which satisfies the system of SDEs

$$dS_t = (r - q)S_t dt + \sqrt{\nu_t} S_t dW_t^S, \quad S_{t_0} = s_{t_0}; \quad t \in [t_0, t_N], \tag{6.4}$$

$$d\nu_t = \kappa(\theta - \nu_t)dt + \xi \sqrt{\nu_t} dW_t^\nu, \quad \nu_{t_0} = \nu_0; \quad t \in [t_0, t_N], \tag{6.5}$$

with risk-free interest rate  $r \in \mathbb{R}$ , dividend rate  $q \in (0, \infty)$ , initial conditions  $s_{t_0}, \nu_0 \in (0, \infty)$ , speed of mean reversion  $\kappa \in (0, \infty)$ , long term mean of the variance process  $\theta \in (0, \infty)$ , and volatility coefficient of the variance process  $\xi \in (0, \infty)$ . Furthermore,  $(W_t^S)_{t \in [t_0, t_N]}$  and  $(W_t^\nu)_{t \in [t_0, t_N]}$  are two one-dimensional, standard Brownian motions, satisfying  $\mathbb{E}[dW_t^S dW_t^\nu] = \rho_{\nu,S} dt$ , for some correlation parameter  $\rho_{\nu,S} \in (-1, 1)$ . In this paper, we use the QE-scheme, proposed in [1] to approximate<sup>14</sup>  $(S, \nu)$ . If necessary, we use a finer time grid for the approximation of  $(S, \nu)$  than the exercise dates,  $\mathbb{T}$ .

We consider a standard Bermudan put option, i.e., identical pay-off functions at all exercise dates, only depending on the underlying asset, with the pay-off function  $g(s) = (K - s)^+$ , for  $s \in (0, \infty)$ , and strike  $K$ .

#### 6.2.1. Comparison with Monte-Carlo-based algorithms

In this section, we again compare the DOS algorithm with the two Monte-Carlo-based algorithms, SGBM and LSM. We use the following set of model parameters:  $r = 0.04$ ,  $q = 0$ ,  $s_{t_0} = 100$ ,  $\kappa = 1.15$ ,  $\theta = 0.0348$ ,  $\xi = 0.459$ ,  $\nu_0 = 0.0348$  and  $\rho_{\nu,S} = -0.64$ , and the contract parameters:  $T = 0.25$ ,  $N = 10$ ,  $K = 100$ . Note that for this set of parameters, the Feller condition is not satisfied. The parameters coincide with Set B in [39], in which the valuation is carried out with the so-called 2D-COS method. The 2D-COS method is a Fourier-based method and is assumed to yield highly accurate valuation of the option.

For the LSM, we use as basis functions, for  $t_n \in \mathbb{T}$ , Laguerre polynomials of degree 3 of  $S_{t_n}$ , Laguerre polynomials of degree 3 of  $\nu_{t_n}$  and  $\nu_{t_n} S_{t_n}$  (only the constant basis function is used). For the SGBM, we use 32 equally-sized bundles based on  $S_{t_n}$  and for  $t_n \in \mathbb{T}$ , we use as basis functions a constant,  $S_{t_n}$ ,  $S_{t_n} \nu_{t_n}$  and the first 3 powers of  $\nu_{t_n}$ . These parameters are chosen such that the approximate option value at  $t_0$  is as close as possible to the reference value of 3.198944, for a Bermudan option with 10 exercise dates, retrieved from [39]. The obtained option values (for each algorithm, average value of 10 runs) at  $t_0$  were 3.1792, 3.2033, and 3.1222 for the DOS algorithm, the SGBM and the LSM, respectively.

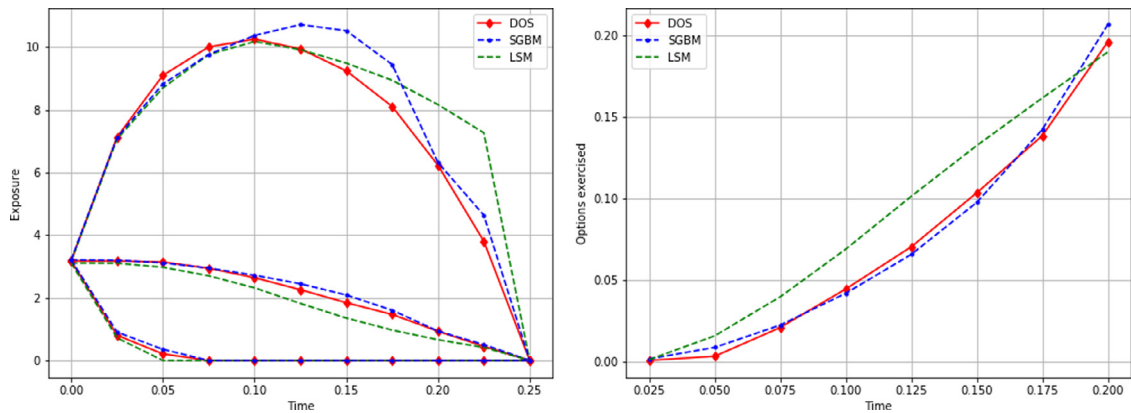
It should be stressed that the LSM and the SGBM in this section could be enhanced by specific choices of basis functions and/or bundling strategies. The reason for the present comparison to still be relevant is to demonstrate the flexibility of DOS and the NN-regression, in which nothing has been changed (from the examples with Black–Scholes dynamics) except the dynamics of the stochastic process from which we generate training data.

Fig. 8 shows the exposure profiles and the exercise frequency computed with the three algorithms. We note that the DOS and the SGBM agree well on the EE and the lower percentile of the PFE, but differ significantly on the upper PFE.

**Remark 6.1.** Regarding computational time, the training of Phase I takes less than one minute with GPU-based parallelized computations (as in [5]) and about 30 times slower on a CPU. Phase II has similar computation time, which is not surprising since the size and shape of the neural networks used are similar.

**Remark 6.2.** In Section 3.2.2, three versions of using which training data were presented. We noticed that when applying A3, the accuracy of the stopping decisions was slightly lower than with A1 and A2 (although less than 0.5% lower), as the neural network does not "see" enough points close to the exercise boundary. This could be mitigated by also including some of the points closest to the boundary (blue points close to the red domain in Fig. 1). However, A2 was used in all experiments.

<sup>14</sup> For notational convenience, the state process is denoted by  $X$ , which falsely indicates that we have an exact form for  $X$ . This is because our focus is on approximating option values, not the underlying state process. It should however be mentioned that  $X$  needs to be approximated in this section.



**Fig. 8.** Comparison of the DOS algorithm, the SGBM and the LSM for a Bermudan put option under the Heston model. **Left:** Expected exposure and potential future exposures at 97.5%- and 2.5%-levels. **Right:** Proportion of options exercised.

**7. Conclusions**

The expected exposure (EE) is important in risk-management and a central component in XVA computations. In this paper we show how EE for options with early exercise features can be formulated in terms of stopping decisions without first having to compute the pathwise option values (which is the standard approach today). In addition, if the full exposure profile is required, a second regression phase is proposed. Overall, the methodology excels in situations with a complicated exercise boundary, potentially with a high-dimensional underlying asset dynamics. The demonstrated capability of going back and forth between risk neutral and real world measures is important for some XVA computations as well as in the insurance industry.

To make the method even more flexible, in a follow up paper, the algorithm is extended to handle a portfolio with different financial derivatives to be able to compute XVA on a portfolio level. Another interesting line of research would be to consider model parameters and contract details such as pay-off function, contract length as inputs to the neural network to be able to solve a similar problem for a family of derivatives simultaneously.

**Declarations of interest**

None

**Acknowledgments**

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 Skłodowska–Curie grant agreement No. 813261. Furthermore, we are grateful for valuable comments of two anonymous referees and discussions with Adam Andersson, Andrea Fontanari, Lech A. Grzelak, and Shashi Jain regarding the content of this paper.

**Appendix A. Proofs**

**Proof of Theorem 2.2..** Equation (2.8) is directly obtained by the definition of the option value and the law of total expectation. To prove (2.9), we use the definition of the option value, the law of total expectation and by the Radon–Nikodym derivative, we have

$$\begin{aligned}
 \mathbb{E}_{\mathbb{P}}(t_n) &= \int_{(y_n, \dots, y_1) \in \mathbb{R}^d \times \dots \times \mathbb{R}^d} \mathbb{E}_{\mathbb{Q}} \left[ e^{-r(\tau_n(f_n^*) - t_n)} g_{\tau_n(f_n^*)}(X_{\tau_n(f_n^*)}) | X_{t_n} = y_n \right] \\
 &\quad \times \frac{p_{X_{t_n}, \dots, X_{t_1}}(y_n, \dots, y_1)}{q_{X_{t_n}, \dots, X_{t_1}}(y_n, \dots, y_1)} q_{X_{t_n}, \dots, X_{t_1}}(y_n, \dots, y_1) dy_n \cdots dy_1,
 \end{aligned}
 \tag{A.1}$$

where  $q_{X_{t_n}, \dots, X_{t_1}}(y_n, \dots, y_1)$  and  $p_{X_{t_n}, \dots, X_{t_1}}(y_n, \dots, y_1)$  are joint densities of

$X_{t_n}, \dots, X_{t_1}$  (conditioned on  $X_{t_0} = x$ ) under the measures  $\mathbb{P}$  and  $\mathbb{Q}$ , respectively. The fact that  $\mathbb{P}$  and  $\mathbb{Q}$  are equivalent, guarantees that the quotient in (A.1) is well defined. Furthermore, due to the Markov property of  $X$ , we have

$$p_{X_{t_n}, \dots, X_{t_1}}(y_n, \dots, y_1) = p_{X_{t_n} | X_{t_{n-1}}}(y_n | y_{n-1}) \times \dots \times p_{X_{t_1} | X_{t_0}}(y_1 | y_0).
 \tag{A.2}$$

The same argumentation holds for  $q_{X_{t_n}, \dots, X_{t_1}}(y_n, \dots, y_1)$ . The proof is finalized by inserting the product of density functions (A.2) in (A.1) and writing the expression as a  $\mathbb{Q}$  expectation, and again, use the law of total expectation.  $\square$

**Proof of Theorem 3.1.** Since  $\bar{x} \in \mathcal{E}_n$ ,  $V_{t_n}(\bar{x}; t_N) = g(\bar{x})$  and  $V_{t_n}(\bar{x}; t_N) = V_{t_{n+1}}(\bar{x}; t_N + \Delta)$  we have that  $V_{t_{n+1}}(\bar{x}; t_N + \Delta) = g(\bar{x})$ . From (2.2) we also see that  $V_{t_{n+1}}(\bar{x}; t_N) \leq V_{t_{n+1}}(\bar{x}; t_N + \Delta)$  and  $V_{t_{n+1}}(\bar{x}; t_N) \geq g(\bar{x})$ . Therefore  $V_{t_{n+1}}(\bar{x}; t_N) = g(\bar{x})$  and  $\bar{x} \in \mathcal{E}_{n+1}$ .  $\square$

## References

- [1] L. Andersen, Efficient simulation of the heston stochastic volatility model, Available at SSRN 946405 (2007).
- [2] L. Andersen, M. Broadie, Primal-dual simulation algorithm for pricing multidimensional american options, *Manage. Sci.* 50 (9) (2004) 1222–1234.
- [3] I. Arregui, B. Salvador, D. Ševčovič, C. Vázquez, PDE models for american options with counterparty risk and two stochastic factors: mathematical analysis and numerical solution, *Comput. Math. Appl.* (2019).
- [4] R. Baviera, G. La Bua, P. Pellicoli, CVA with wrong-way risk in the presence of early exercise, in: *Innovations in Derivatives Markets*, Springer, Cham, 2016, pp. 103–116.
- [5] S. Becker, P. Cheridito, A. Jentzen, Deep optimal stopping, *J. Mach. Learn. Res.* 20 (2019).
- [6] S. Becker, P. Cheridito, A. Jentzen, Pricing and hedging american-style options with deep learning, arXiv preprint arXiv:1912.11060 (2019).
- [7] R. Bellman, Dynamic programming, *Science* 153 (3731) (1966) 34–37.
- [8] D. Belomestny, C. Bender, J. Schoenmakers, True upper bounds for bermudan products via non-nested monte carlo, *Math. Finance: Int. J. Math. Stat. Financ. Econ.* 19 (1) (2009) 53–71.
- [9] M. Breton, O. Marzouk, An efficient method to price counterparty risk, *Groupe d'études et de recherche en analyse des décisions*, 2014.
- [10] M. Broadie, M. Cao, Improved lower and upper bound algorithms for pricing american options by simulation, *Quant. Finance* 8 (8) (2008) 845–861.
- [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. Broadie, J. Detemple, The valuation of american options on multiple assets, *Math. Finance* 7 (3) (1997) 241–286.
- [13] M. Broadie, P. Glasserman, A stochastic mesh method for pricing high-dimensional american options, *J. Comput. Finance* 7 (2004) 35–72.
- [14] C. De Graaf, Q. Feng, D. Kandhai, C. Oosterlee, Efficient computation of exposure profiles for counterparty credit risk, *Int. J. Theor. Appl. Finance* 17 (04) (2014) 1450024.
- [15] F. Fang, C. Oosterlee, Pricing early-exercise and discrete barrier options by fourier-cosine series expansions, *Numerische Mathematik* 114 (1) (2009) 27.
- [16] F. Fang, C. Oosterlee, A fourier-based valuation method for bermudan and barrier options under heston's model, *SIAM J. Financ. Math.* 2 (1) (2011) 439–463.
- [17] Q. Feng, S. Jain, P. Karlsson, D. Kandhai, C.W. Oosterlee, Efficient computation of exposure profiles on real-world and risk-neutral scenarios for bermudan swaptions, Available at SSRN 2790874 (2016).
- [18] P. Forsyth, K. Vetzal, Quadratic convergence for valuing american options using a penalty method, *SIAM J. Sci. Comput.* 23 (6) (2002) 2095–2122.
- [19] P. Glasserman, *Monte Carlo Methods in Financial Engineering*, 53, Springer Science & Business Media, 2013.
- [20] A. Green, XVA: Credit, Funding and Capital Valuation Adjustments, John Wiley & Sons, 2015.
- [21] A.D. Green, C. Kenyon, C. Dennis, KVA: capital valuation adjustment, *Risk*, December (2014).
- [22] J. Gregory, *The xVA challenge: Counterparty credit risk, funding, collateral and capital*, John Wiley & Sons, 2015.
- [23] L. Györfi, M. Kohler, A. Krzyżak, H. Walk, *A Distribution-free Theory of Nonparametric Regression*, Springer Science & Business Media, 2006.
- [24] T. Haentjens, K. in't Hout, ADI schemes for pricing american options under the heston model, *Appl. Math. Finance* 22 (3) (2015) 207–237.
- [25] S. Heston, A closed-form solution for options with stochastic volatility with applications to bond and currency options, *Rev. Financ. Stud.* 6 (2) (1993) 327–343.
- [26] K. in't Hout, *Numerical partial differential equations in finance explained; an introduction to computational finance*, Palgrave Macmillan UK, 2017.
- [27] J. Jackwerth, Option-implied risk-neutral distributions and implied binomial trees: a literature review, *J. Derivat.* 7 (2) (1999) 66–82.
- [28] S. Jain, P. Karlsson, D. Kandhai, KVA, mind your p's and q's!, *Wilmott* 2019 (102) (2019) 60–73.
- [29] S. Jain, C. Oosterlee, The stochastic grid bundling method: efficient pricing of bermudan options and their greeks, *Appl. Math. Comput.* 269 (2015) 412–431.
- [30] D. 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).
- [31] M. Kohler, A. Krzyżak, N. Todorovic, Pricing of high-dimensional american options by neural networks, *Math. Finance: Int. J. Math. Stat. Financ. Econ.* 20 (3) (2010) 383–410.
- [32] B. Lapeyre, J. Lelong, Neural network regression for bermudan option pricing, arXiv preprint arXiv:1907.06474 (2019).
- [33] V. Lokeshwar, V. Bhardawaj, S. Jain, Neural network for pricing and universal static hedging of contingent claims, Available at SSRN 3491209 (2019).
- [34] F. Longstaff, E. Schwartz, Valuing american options by simulation: a simple least-squares approach, *Rev. Financ. Stud.* 14 (1) (2001) 113–147.
- [35] 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).
- [36] B. Oksendal, *Stochastic Differential Equations: An Introduction with Applications*, Springer Science & Business Media, 2013.
- [37] C. Reisinger, J.H. Witte, On the use of policy iteration as an easy way of pricing american options, *SIAM J. Financ. Math.* 3 (1) (2012) 459–478.
- [38] M. Rubinstein, Edgeworth binomial trees, *J. Derivat.* 5 (1998) 20–27.
- [39] M. Ruijter, C. Oosterlee, Two-dimensional fourier cosine series expansion method for pricing financial options, *SIAM J. Sci. Comput.* 34 (5) (2012) B642–B671.
- [40] Y. Shen, J.A. Van Der Weide, J.H. Anderluh, A benchmark approach of counterparty credit exposure of bermudan option under lévy process: the monte carlo-COS method, *Procedia Comput. Sci.* 18 (2013) 1163–1171.
- [41] C. Vázquez, An upwind numerical approach for an american and european option pricing model, *Appl. Math. Comput.* 97 (2–3) (1998) 273–286.
- [42] H. White, *Asymptotic Theory for Econometricians*, Academic press, 2014.
- [43] O. Zhyldevskyy, A fast fourier transform technique for pricing american options under stochastic volatility, *Rev. Derivat. Res.* 13 (1) (2010) 1–24.