ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Seminar for Applied Mathematics
SAM

Solving high-dimensional optimal stopping problems using deep learning

S. Becker and P. Cheridito and A. Jentzen and T. Welti

Research Report No. 2019-60
November 2019

Seminar für Angewandte Mathematik
Eidgenössische Technische Hochschule
CH-8092 Zürich
Switzerland

Solving high-dimensional optimal stopping problems using deep learning

Sebastian Becker[1], Patrick Cheridito[2], Arnulf Jentzen[3], and Timo Welti[4]

[1] ZENAI AG, Zürich, Switzerland, e-mail: sebastian.becker@zenai.ch

[2] RiskLab, Department of Mathematics, ETH Zürich, Switzerland, e-mail: patrick.cheridito@math.ethz.ch

[3] Seminar for Applied Mathematics, Department of Mathematics, ETH Zürich, Switzerland, e-mail: arnulf.jentzen@sam.math.ethz.ch

[4] Seminar for Applied Mathematics, Department of Mathematics, ETH Zürich, Switzerland, e-mail: timo.welti@sam.math.ethz.ch

November 13, 2019

Abstract

Nowadays many financial derivatives which are traded on stock and futures exchanges, such as American or Bermudan options, are of early exercise type. Often the pricing of early exercise options gives rise to high-dimensional optimal stopping problems, since the dimension corresponds to the number of underlyings in the associated hedging portfolio. High-dimensional optimal stopping problems are, however, notoriously difficult to solve due to the well-known curse of dimensionality. In this work we propose an algorithm for solving such problems, which is based on deep learning and computes, in the context of early exercise option pricing, both approximations for an optimal exercise strategy and the price of the considered option. The proposed algorithm can also be applied to optimal stopping problems that arise in other areas where the underlying stochastic process can be efficiently simulated. We present numerical results for a large number of example problems, which include the pricing of many high-dimensional American and Bermudan options such as, for example, Bermudan max-call options in up to 5000 dimensions. Most of the obtained results are compared to reference values computed by exploiting the specific problem design or, where available, to reference values from the literature. These numerical results suggest that the proposed algorithm is highly effective in the case of many underlyings, in terms of both accuracy and speed.

Keywords: American option, Bermudan option, financial derivative, derivative pricing, option pricing, optimal stopping, curse of dimensionality, deep learning

Contents

1 Introduction

Nowadays many financial derivatives which are traded on stock and futures exchanges, such as American or Bermudan options, are of early exercise type. Contrary to European options, the holder of such an option has the right to exercise before the time of maturity. In models from mathematical finance for the appropriate pricing of early exercise options this aspect gives rise to optimal stopping problems. The dimension of such optimal stopping problems can often be quite high since it corresponds to the number of underlyings, that is, the number of considered financial assets in the hedging portfolio associated to the optimal stopping problem. High-dimensional optimal stopping problems are, however, notoriously difficult to solve due to the well-known *curse of dimensionality* (cf. Bellman [10]). Such optimal stopping problems can in nearly all cases not be solved explicitly and it is an active topic of research to design and analyse approximation methods which are capable of approximately solving possibly high-dimensional optimal stopping problems. Many different approaches for numerically solving optimal stopping problems and, in particular, American and Bermudan option pricing problems have been studied in the literature, cf., e.g., [87, 33, 5, 27, 1, 24, 88, 2, 26, 73, 89, 77, 80, 4, 41, 3, 22, 25, 42, 49, 36, 39, 44, 18, 66, 76, 28, 37, 53, 19, 21, 23, 61, 65, 69, 74, 14, 62, 64, 78, 11, 12, 68, 34, 52, 63, 13, 17, 79, 29, 47, 15, 16, 55, 71, 67, 20, 30, 83, 8, 9, 45, 84, 90, 7, 40, 46, 70, 72]. For example, such approaches include approximating the Snell envelope or continuation values (cf., e.g., [87, 5, 27, 73]), computing optimal exercise boundaries (cf., e.g., [2]), and dual methods (cf., e.g., [77, 49]). Whereas in [49, 64] artificial neural networks with one hidden layer have been employed to approximate continuation values, more recently numerical approximation methods for American/Bermudan option pricing that are based on deep learning have been introduced, see [84, 83, 9, 40, 70]. More precisely, in [84, 83] deep artificial neural networks are used to approximately solve the corresponding obstacle partial differential equation problem, in [9] the corresponding optimal stopping problem is tackled directly with a deep learning based algorithm, [40] applies an extension of the deep BSDE solver from [48, 35] to the corresponding reflected backward stochastic differential equation problem, and in [70] a deep artificial neural network based variant of the classical algorithm introduced by Longstaff & Schwartz [73] is examined.

In this work we propose an algorithm for solving general possibly high-dimensional optimal stopping problems, cf. Framework 3.2 in Subsection 3.2 below. In spirit it is similar to the algorithm introduced in [9]. The proposed algorithm is based on deep learning and computes both approximations for an optimal stopping strategy and the optimal expected pay-off associated to the considered optimal stopping problem. In the context of pricing early exercise options these correspond to approximations for an optimal exercise strategy and the price of the considered option, respectively. The derivation and implementation of the proposed algorithm consist of essentially the following three steps.

(I) A neural network architecture for in an appropriate sense 'randomised' stopping times (cf. (31) in Subsection 2.4 below) is established in such a way that varying the neural network parameters leads to different randomised stopping times being expressed. This neural network architecture is used to replace the supremum of the expected pay-off over suitable stopping times (which constitutes the generic optimal stopping problem) by the supremum of a suitable objective function over neural network parameters (cf. (38)–(39) in Subsection 2.5 below).

(II) A stochastic gradient ascent-type optimisation algorithm is employed to compute neural network parameters that approximately maximise the objective function (cf. Subsection 2.6).

(III) From these neural network parameters and the corresponding randomised stopping time, a true stopping time is constructed which serves as the approximation for an optimal stopping strategy (cf. (44) and (46) in Subsection 2.7 below). In addition, an approximation for the optimal expected pay-off is obtained by computing a Monte Carlo approximation of the expected pay-off under this approximately optimal stopping strategy (cf. (45) in Subsection 2.7).

It follows from (III) that the proposed algorithm computes a low-biased approximation of the optimal expected pay-off (cf. (48) below). Yet a large number of numerical experiments where a reference value is available (cf. Section 4) shows that the bias appears to become small quickly during training and that a very satisfying accuracy can be achieved in short computation time, even in high dimensions (see the end of this introduction below for a brief overview of the numerical computations that have been performed). Moreover, in (I) we resort to randomised stopping times in order to circumvent the discrete nature of stopping times that attain only finitely many different values. As a result it is possible in (II) to tackle the arising optimisation problem with a stochastic gradient ascent-type algorithm. Furthermore, while the focus in this article lies on American and Bermudan option pricing, the proposed algorithm can also be applied to optimal stopping problems that arise in other areas where the underlying stochastic process can be efficiently simulated. Apart from this, we only rely on the assumption that the stochastic process to be optimally stopped is a Markov process (cf. Subsection 2.4). But this assumption is no substantial restriction since on the one hand it is automatically fulfilled in many relevant problems and on the other hand a discrete stochastic process that is not a Markov process can be replaced by a Markov process of higher dimension that aggregates all necessary information (cf., e.g., [9, Subsection 4.3] and, e.g., Subsection 4.4.4 below).

Next we make a short comparison to the algorithm introduced in [9]. The latter is based on introducing for every point in time where stopping is permitted an auxiliary optimal stopping problem for which stopping is only allowed at that point in time or later (cf. [9, (4) in Subsection 2.1]). Starting at maturity, these auxiliary problems are solved recursively backwards until the initial time is reached. Thereby, in every new step neural network parameters are learned for an objective function that depends, in particular, on the parameters found in the previous steps (cf. [9, Subsection 2.3]). In contrast, in (I) a single objective function is designed which allows to search in (II) for neural network parameters that maximise the expected pay-off simultaneously over (randomised) stopping times which may decide to stop at any of the admissible points in time. Therefore, the algorithm proposed here does not rely on a recursion over the different time points. In addition, the construction of the final approximation for an optimal stopping strategy in (III) differs from a corresponding construction in [9].

The remainder of this article is organised as follows. In Section 2 we present the main ideas from which the proposed algorithm is derived. More specifically, in Subsection 2.1 we illustrate how an optimal stopping problem in the context of American option pricing is typically formulated. Thereafter, a replacement of this continuous time problem by a corresponding discrete time optimal stopping problem is discussed by means of an example in Subsection 2.2. Subsection 2.3 is devoted to the statement and proof of an elementary, but crucial result (see Lemma 2.2) about factorising general discrete stopping times in terms of compositions of measurable functions, which lies at the heart of the neural network architecture we propose in Subsection 2.4 to approximate general discrete stopping times. This construction, in turn, is exploited in Subsection 2.5 to transform the discrete time optimal stopping problem from Subsection 2.2 into the search of a maximum of a suitable objective function (cf. (I) above). In Subsection 2.6 we suggest to employ stochastic gradient ascent-type optimisation algorithms to find approximate maximum points of the objective function (cf. (II) above). As a last step, we explain in Subsection 2.7 how we calculate final approximations for both the American option price as well as an optimal exercise strategy (cf. (III) above). In Section 3 we introduce the proposed algorithm in a concise way, first for a special case for the sake of clarity (see Subsection 3.1) and second in more generality so that, in particular, a rigorous description of our implementations is fully covered (see Subsections 3.2–3.3). Following this, in Section 4 first a few theoretical results are presented (see Subsection 4.1), which are used to design numerical example problems and to provide reference values. Thereafter, we describe in detail a large number of example problems, on which our proposed algorithm has been tested, and present numerical results for each of these problems. In particular, the examples include the optimal stopping of Brownian motions (see Subsection 4.3.1), the pricing of certain exotic American geometric average put and call-type options (see Subsection 4.3.2), the pricing of Bermudan max-call options in up to 5000 dimensions (see Subsection 4.4.1), the pricing of an American strangle spread basket option in five dimensions (see Subsection 4.4.2), the pricing of an American put basket option in Dupire's local volatility model in five dimensions (see Subsection 4.4.3), and the pricing of an exotic path-dependent financial derivative of a single underlying, which is modelled as a 100-dimensional optimal stopping problem (see Subsection 4.4.4). The numerical results for the examples in Subsections 4.3.1.2, 4.3.2.1, 4.3.2.2, 4.3.2.3, and 4.4.1.3 are compared to calculated reference values that can be easily obtained due to the specific design of the considered optimal stopping problem. Moreover, the examples in Subsections 4.3.2.2, 4.4.1.1, 4.4.1.3, 4.4.2, 4.4.3, and 4.4.4 are taken from the literature and our corresponding numerical results are compared to reference values from the literature (where available).

2 Main ideas of the proposed algorithm

In this section we outline the main ideas that lead to the formulation of the proposed algorithm in Subsections 3.1–3.2 below by considering the example of pricing an American option. The proposed algorithm in Framework 3.2 below is however general enough to also be applied to optimal stopping problems where there are no specific assumptions on the dynamics of the underlying stochastic process, as long as it can be cheaply simulated (cf. Subsection 3.3). Furthermore, often in practice and, in particular, in the case of Bermudan option pricing (cf. many of the examples in Section 4) the optimal stopping problem of interest is not a continuous time problem but is already formulated in discrete time. In such a situation there is no need for a time discretisation as described in Subsection 2.2 below and the proposed algorithm in Framework 3.2 can be applied directly.

2.1 The American option pricing problem

Let $T \in (0,\infty)$, $d \in \mathbb{N}$, let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space with a filtration $\mathscr{F} = (\mathscr{F}_t)_{t\in[0,T]}$ that satisfies the usual conditions (cf., e.g., [57, Definition 2.25 in Section 1.2]), let $\xi\colon \Omega \to \mathbb{R}^d$ be an $\mathscr{F}_0/\mathcal{B}(\mathbb{R}^d)$-measurable function which satisfies for all $p \in (0,\infty)$ that $\mathbb{E}\big[\|\xi\|_{\mathbb{R}^d}^p\big] < \infty$, let $W\colon [0,T]\times\Omega \to \mathbb{R}^d$ be a standard $(\Omega, \mathcal{F}, \mathbb{P}, \mathscr{F})$-Brownian motion with continuous sample paths, let $\mu\colon \mathbb{R}^d \to \mathbb{R}^d$ and $\sigma\colon \mathbb{R}^d \to \mathbb{R}^{d\times d}$ be Lipschitz continuous functions, let $X\colon [0,T]\times\Omega \to \mathbb{R}^d$ be an $\mathscr{F}$-adapted continuous solution process of the SDE

$$dX_t = \mu(X_t)\,dt + \sigma(X_t)\,dW_t, \qquad X_0 = \xi, \qquad t \in [0,T], \tag{1}$$

let $\mathbb{F} = (\mathbb{F}_t)_{t\in[0,T]}$ be the filtration generated by X, and let $g\colon [0,T]\times\mathbb{R}^d \to \mathbb{R}$ be a continuous and at most polynomially growing function. We think of X as a model for the price processes of d underlyings (say, d stock prices) under the risk-neutral pricing measure $\mathbb{P}$ (cf., e.g., Kallsen [56]) and we are then interested in approximatively pricing the American option on the process $(X_t)_{t\in[0,T]}$ with the discounted pay-off function $g\colon [0,T]\times\mathbb{R}^d \to \mathbb{R}$, that is, we intend to compute the real number

$$\sup\Big\{\mathbb{E}\big[g(\tau, X_\tau)\big] : \substack{\tau\colon \Omega\to[0,T] \text{ is an} \\ \mathbb{F}\text{-stopping time}}\Big\}. \tag{2}$$

In addition to the *price* of the American option in the model (1) there is also a high demand from the financial engineering industry to compute an approximately *optimal exercise strategy*, that is, to compute a stopping time which approximately reaches the supremum in (2).

In a very simple example of (1)–(2), we can think of an *American put option* in the one-dimensional Black–Scholes model, in which there are an interest rate $r \in \mathbb{R}$, a dividend rate $\delta \in [0, \infty)$, a volatility $\beta \in (0, \infty)$, and a strike price $K \in (0, \infty)$ such that it holds for all $x \in \mathbb{R}$, $t \in [0, T]$ that $d = 1$, $\mu(x) = (r - \delta)\, x$, $\sigma(x) = \beta\, x$, and $g(t, x) = e^{-rt} \max\{K - x, 0\}$.

2.2 Temporal discretisation

To derive the proposed approximation algorithm we first apply the Euler–Maruyama scheme to the stochastic differential equation (1) (cf. (5)–(6) below) and we employ a suitable time discretisation for the optimal stopping problem (2). For this let $N \in \mathbb{N}$ be a natural number and let $t_0, t_1, \ldots, t_N \in [0, T]$ be real numbers with

$$0 = t_0 < t_1 < \cdots < t_N = T \tag{3}$$

(such that the maximal mesh size $\max_{n \in \{0,1,\ldots,N-1\}}(t_{n+1} - t_n)$ is sufficiently small). Observe that (1) ensures that for all $n \in \{0, 1, \ldots, N-1\}$ it holds $\mathbb{P}$-a.s. that

$$X_{t_{n+1}} = X_{t_n} + \int_{t_n}^{t_{n+1}} \mu(X_s)\, ds + \int_{t_n}^{t_{n+1}} \sigma(X_s)\, dW_s. \tag{4}$$

Note that (4) suggests for every $n \in \{0, 1, \ldots, N-1\}$ that

$$X_{t_{n+1}} \approx X_{t_n} + \mu(X_{t_n})\,(t_{n+1} - t_n) + \sigma(X_{t_n})\left(W_{t_{n+1}} - W_{t_n}\right). \tag{5}$$

The approximation scheme associated to (5) is referred to as the Euler–Maruyama scheme in the literature (cf., e.g., Maruyama [75] and Kloeden & Platen [60]). More formally, let $\mathcal{X} \colon \{0, 1, \ldots, N\} \times \Omega \to \mathbb{R}^d$ be the stochastic process which satisfies for all $n \in \{0, 1, \ldots, N-1\}$ that $\mathcal{X}_0 = \xi$ and

$$\mathcal{X}_{n+1} = \mathcal{X}_n + \mu(\mathcal{X}_n)\,(t_{n+1} - t_n) + \sigma(\mathcal{X}_n)\left(W_{t_{n+1}} - W_{t_n}\right) \tag{6}$$

and let $\mathfrak{F} = (\mathfrak{F}_n)_{n \in \{0,1,\ldots,N\}}$ be the filtration generated by $\mathcal{X}$. Combining this with (5) suggests the approximation

$$\sup\left\{\mathbb{E}\big[g(t_\tau, \mathcal{X}_\tau)\big] : \substack{\tau \colon \Omega \to \{0,1,\ldots,N\} \text{ is an} \\ \mathfrak{F}\text{-stopping time}}\right\} \approx \sup\left\{\mathbb{E}\big[g(\tau, X_\tau)\big] : \substack{\tau \colon \Omega \to [0,T] \text{ is an} \\ \mathbb{F}\text{-stopping time}}\right\} \tag{7}$$

for the price (2) of the American option in Subsection 2.1. Below we employ, in particular, (7) to derive the proposed approximation algorithm.

2.3 Factorisation lemma for stopping times

The derivation of the proposed approximation algorithm is in parts based on an elementary reformulation of time-discrete stopping times (cf. the left hand side of (7)) in terms of measurable functions that appropriately characterise the behaviour of the stopping time; cf. (10) and (9) in Lemma 2.2 below. The proof of Lemma 2.2 employs the following well-known factorisation result, Lemma 2.1. Lemma 2.1 follows, e.g., from Klenke [59, Corollary 1.97].

Lemma 2.1 (Factorisation lemma). *Let $(S, \mathcal{S})$ be a measurable space, let Ω be a set, let $B \in \mathcal{B}(\mathbb{R} \cup \{-\infty, \infty\})$, and let $X\colon \Omega \to S$ and $Y\colon \Omega \to B$ be functions. Then it holds that Y is $\{X^{-1}(A)\colon A \in \mathcal{S}\}/\mathcal{B}(B)$-measurable if and only if there exists an $\mathcal{S}/\mathcal{B}(B)$-measurable function $f\colon S \to B$ such that*

$$Y = f \circ X. \tag{8}$$

We are now ready to present the above mentioned Lemma 2.2. This elementary lemma is a consequence of Lemma 2.1 above.

Lemma 2.2 (Factorisation lemma for stopping times). *Let $d, N \in \mathbb{N}$, let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, let $\mathcal{X}\colon \{0, 1, \ldots, N\} \times \Omega \to \mathbb{R}^d$ be a stochastic process, and let $\mathbb{F} = (\mathbb{F}_n)_{n \in \{0,1,\ldots,N\}}$ be the filtration generated by $\mathcal{X}$. Then*

(i) for all Borel measurable functions $\mathbb{U}_n\colon (\mathbb{R}^d)^{n+1} \to \{0, 1\}$, $n \in \{0, 1, \ldots, N\}$, with $\forall\, x_0, x_1, \ldots, x_N \in \mathbb{R}^d\colon \sum_{n=0}^{N} \mathbb{U}_n(x_0, x_1, \ldots, x_n) = 1$ it holds that the function

$$\Omega \ni \omega \mapsto \sum_{n=0}^{N} n\, \mathbb{U}_n\big(\mathcal{X}_0(\omega), \mathcal{X}_1(\omega), \ldots, \mathcal{X}_n(\omega)\big) \in \{0, 1, \ldots, N\} \tag{9}$$

is an $\mathbb{F}$-stopping time and

(ii) for every $\mathbb{F}$-stopping time $\tau\colon \Omega \to \{0, 1, \ldots, N\}$ there exist Borel measurable functions $\mathbb{U}_n\colon (\mathbb{R}^d)^{n+1} \to \{0, 1\}$, $n \in \{0, 1, \ldots, N\}$, which satisfy $\forall\, x_0, x_1, \ldots, x_N \in \mathbb{R}^d\colon \sum_{n=0}^{N} \mathbb{U}_n(x_0, x_1, \ldots, x_n) = 1$ and

$$\tau = \sum_{n=0}^{N} n\, \mathbb{U}_n(\mathcal{X}_0, \mathcal{X}_1, \ldots, \mathcal{X}_n). \tag{10}$$

Proof of Lemma 2.2. Note that for all Borel measurable functions $\mathbb{U}_n\colon (\mathbb{R}^d)^{n+1} \to \{0, 1\}$, $n \in \{0, 1, \ldots, N\}$, with $\forall\, x_0, x_1, \ldots, x_N \in \mathbb{R}^d\colon \sum_{n=0}^{N} \mathbb{U}_n(x_0, x_1, \ldots, x_n) = 1$ and all $k \in \{0, 1, \ldots, N\}$ it holds that

$$\begin{aligned} &\left\{\omega \in \Omega\colon \sum_{n=0}^{N} n\, \mathbb{U}_n\big(\mathcal{X}_0(\omega), \mathcal{X}_1(\omega), \ldots, \mathcal{X}_n(\omega)\big) = k\right\} \\ &= \big\{\omega \in \Omega\colon \mathbb{U}_k\big(\mathcal{X}_0(\omega), \mathcal{X}_1(\omega), \ldots, \mathcal{X}_k(\omega)\big) = 1\big\} \\ &= \left\{\omega \in \Omega\colon (\mathcal{X}_0(\omega), \mathcal{X}_1(\omega), \ldots, \mathcal{X}_k(\omega)) \in \underbrace{(\mathbb{U}_k)^{-1}(\{1\})}_{\in \mathcal{B}((\mathbb{R}^d)^{k+1})}\right\} \in \mathbb{F}_k. \end{aligned} \tag{11}$$

This establishes (i). It thus remains to prove (ii). For this let $\tau\colon \Omega \to \{0, 1, \ldots, N\}$ be an $\mathbb{F}$-stopping time. Observe that for every function $\varrho\colon \Omega \to \{0, 1, \ldots, N\}$ and every $\omega \in \Omega$ it holds that

$$\varrho(\omega) = \sum_{n=0}^{N} n\, \mathbb{1}_{\{\varrho = n\}}(\omega). \tag{12}$$

Next note that for every $n \in \{0, 1, \ldots, N\}$ it holds that the function

$$\Omega \ni \omega \mapsto \mathbb{1}_{\{\tau = n\}}(\omega) \in \{0, 1\} \tag{13}$$

is $\mathbb{F}_n/\mathcal{B}(\{0, 1\})$-measurable. This and the fact that

$$\forall\, n \in \{0, 1, \ldots, N\}\colon \sigma_\Omega\big((\mathcal{X}_0, \mathcal{X}_1, \ldots, \mathcal{X}_n)\big) = \mathbb{F}_n \tag{14}$$

ensures that for every $n \in \{0, 1, \dots, N\}$ it holds that the function

$$\Omega \ni \omega \mapsto \mathbb{1}_{\{\tau = n\}}(\omega) \in \{0, 1\} \tag{15}$$

is $\sigma_\Omega((\mathcal{X}_0, \mathcal{X}_1, \dots, \mathcal{X}_n))/\mathcal{B}(\{0, 1\})$-measurable. Lemma 2.1 hence demonstrates that there exist Borel measurable functions $\mathbb{V}_n \colon (\mathbb{R}^d)^{n+1} \to \{0, 1\}$, $n \in \{0, 1, \dots, N\}$, which satisfy for all $n \in \{0, 1, \dots, N\}$, $\omega \in \Omega$ that

$$\mathbb{1}_{\{\tau = n\}}(\omega) = \mathbb{V}_n\big(\mathcal{X}_0(\omega), \mathcal{X}_1(\omega), \dots, \mathcal{X}_n(\omega)\big). \tag{16}$$

Next let $\mathbb{U}_n \colon (\mathbb{R}^d)^{n+1} \to \mathbb{R}$, $n \in \{0, 1, \dots, N\}$, be the functions which satisfy for all $n \in \{0, 1, \dots, N\}$, $x_0, x_1, \dots, x_n \in \mathbb{R}^d$ that

$$\begin{aligned} &\mathbb{U}_n(x_0, x_1, \dots, x_n) \\ &= \max\{\mathbb{V}_n(x_0, x_1, \dots, x_n), n + 1 - N\} \left[1 - \sum_{k=0}^{n-1} \mathbb{U}_k(x_0, x_1, \dots, x_k)\right]. \end{aligned} \tag{17}$$

Observe that (17), in particular, ensures that for all $x_0, x_1, \dots, x_N \in \mathbb{R}^d$ it holds that

$$\mathbb{U}_N(x_0, x_1, \dots, x_N) = \left[1 - \sum_{k=0}^{N-1} \mathbb{U}_k(x_0, x_1, \dots, x_k)\right]. \tag{18}$$

Hence, we obtain that for all $x_0, x_1, \dots, x_N \in \mathbb{R}^d$ it holds that

$$\sum_{k=0}^{N} \mathbb{U}_k(x_0, x_1, \dots, x_k) = 1. \tag{19}$$

In addition, note that (17) assures that for all $x_0 \in \mathbb{R}^d$ it holds that

$$\mathbb{U}_0(x_0) = \mathbb{V}_0(x_0). \tag{20}$$

Induction, the fact that

$$\forall\, n \in \{0, 1, \dots, N\},\ x_0, x_1, \dots, x_n \in \mathbb{R}^d \colon \mathbb{V}_n(x_0, x_1, \dots, x_n) \in \{0, 1\}, \tag{21}$$

and (17) hence demonstrate that for all $n \in \{0, 1, \dots, N\}$, $x_0, x_1, \dots, x_n \in \mathbb{R}^d$ it holds that

$$\left\{\mathbb{U}_0(x_0), \mathbb{U}_1(x_0, x_1), \dots, \mathbb{U}_n(x_0, x_1, \dots, x_n), \sum_{k=0}^{n} \mathbb{U}_k(x_0, x_1, \dots, x_k)\right\} \subseteq \{0, 1\}. \tag{22}$$

Moreover, note that (17), induction, and the fact that the functions $\mathbb{V}_n \colon (\mathbb{R}^d)^{n+1} \to \{0, 1\}$, $n \in \{0, 1, \dots, N\}$, are Borel measurable ensure that for every $n \in \{0, 1, \dots, N\}$ it holds that the function

$$(\mathbb{R}^d)^{n+1} \ni (x_0, x_1, \dots, x_n) \mapsto \mathbb{U}_n(x_0, x_1, \dots, x_n) \in \{0, 1\} \tag{23}$$

is also Borel measurable. In the next step we observe that (20), (17), (21), and induction assure that for all $n \in \{0, 1, \dots, N\}$, $x_0, x_1, \dots, x_n \in \mathbb{R}^d$ with $n + 1 - N \leq \sum_{k=0}^{n} \mathbb{V}_k(x_0, x_1, \dots, x_k) \leq 1$ it holds that

$$\forall\, k \in \{0, 1, \dots, n\} \colon \mathbb{U}_k(x_0, x_1, \dots, x_k) = \mathbb{V}_k(x_0, x_1, \dots, x_k). \tag{24}$$

In addition, note that (16) shows that for all $\omega \in \Omega$ it holds that

$$\sum_{k=0}^{N} \mathbb{V}_k\big(\mathcal{X}_0(\omega), \mathcal{X}_1(\omega), \ldots, \mathcal{X}_k(\omega)\big) = \sum_{k=0}^{N} \mathbb{1}_{\{\tau=k\}}(\omega) = 1. \tag{25}$$

This, (24), and again (16) imply that for all $k \in \{0, 1, \ldots, N\}$, $\omega \in \Omega$ it holds that

$$\mathbb{U}_k\big(\mathcal{X}_0(\omega), \mathcal{X}_1(\omega), \ldots, \mathcal{X}_k(\omega)\big) = \mathbb{V}_k\big(\mathcal{X}_0(\omega), \mathcal{X}_1(\omega), \ldots, \mathcal{X}_k(\omega)\big) = \mathbb{1}_{\{\tau=k\}}(\omega). \tag{26}$$

Equation (12) hence proves that for all $\omega \in \Omega$ it holds that

$$\tau(\omega) = \sum_{n=0}^{N} n\, \mathbb{U}_n\big(\mathcal{X}_0(\omega), \mathcal{X}_1(\omega), \ldots, \mathcal{X}_n(\omega)\big). \tag{27}$$

Combining this with (19) and (23) establishes (ii). The proof of Lemma 2.2 is thus complete. □

2.4 Neural network architectures for stopping times

In the next step we employ multilayer neural network approximations for the functions $\mathbb{U}_n\colon (\mathbb{R}^d)^{n+1} \to \{0,1\}$, $n \in \{0, 1, \ldots, N\}$, in the factorisation lemma, Lemma 2.2 above. In the following we refer to these functions as 'stopping time factors'. Consider again the setting in Subsections 2.1–2.2, for every $\mathfrak{F}$-stopping time $\tau\colon \Omega \to \{0, 1, \ldots, N\}$ let $\mathbb{U}_{n,\tau}\colon (\mathbb{R}^d)^{n+1} \to \{0,1\}$, $n \in \{0, 1, \ldots, N\}$, be Borel measurable functions which satisfy $\forall\, x_0, x_1, \ldots, x_N \in \mathbb{R}^d\colon \sum_{n=0}^{N} \mathbb{U}_{n,\tau}(x_0, x_1, \ldots, x_n) = 1$ and

$$\tau = \sum_{n=0}^{N} n\, \mathbb{U}_{n,\tau}(\mathcal{X}_0, \mathcal{X}_1, \ldots, \mathcal{X}_n) \tag{28}$$

(see (ii) in Lemma 2.2), let $\nu \in \mathbb{N}$ be a sufficiently large natural number, and for every $n \in \{0, 1, \ldots, N\}$, $\theta \in \mathbb{R}^\nu$ let $u_{n,\theta}\colon \mathbb{R}^d \to (0,1)$ and $U_{n,\theta}\colon (\mathbb{R}^d)^{n+1} \to (0,1)$ be Borel measurable functions which satisfy for all $x_0, x_1, \ldots, x_n \in \mathbb{R}^d$ that

$$U_{n,\theta}(x_0, x_1, \ldots, x_n) = \max\{u_{n,\theta}(x_n), n+1-N\} \left[1 - \sum_{k=0}^{n-1} U_{k,\theta}(x_0, x_1, \ldots, x_k)\right] \tag{29}$$

(cf. (17) above). Observe that for all $\theta \in \mathbb{R}^\nu$, $x_0, x_1, \ldots, x_N \in \mathbb{R}^d$ it holds that

$$\sum_{n=0}^{N} U_{n,\theta}(x_0, x_1, \ldots, x_n) = 1. \tag{30}$$

We think of $\nu \in \mathbb{N}$ as the number of parameters in the employed artificial neural networks and for every appropriate $\mathfrak{F}$-stopping time $\tau\colon \Omega \to \{0, 1, \ldots, N\}$ we think of the functions $U_{n,\theta}\colon (\mathbb{R}^d)^{n+1} \to (0,1)$ for $n \in \{0, 1, \ldots, N\}$ and suitable $\theta \in \mathbb{R}^\nu$ as appropriate approximations for the stopping time factors $\mathbb{U}_{n,\tau}\colon (\mathbb{R}^d)^{n+1} \to \{0,1\}$, $n \in \{0, 1, \ldots, N\}$. Due to (30) for every $\theta \in \mathbb{R}^\nu$ the stochastic process

$$\{0, 1, \ldots, N\} \times \Omega \ni (n, \omega) \mapsto U_{n,\theta}\big(\mathcal{X}_0(\omega), \mathcal{X}_1(\omega), \ldots, \mathcal{X}_n(\omega)\big) \in (0,1) \tag{31}$$

can also be viewed as an in an appropriate sense 'randomised' stopping time (cf., e.g., [86, Definition 1 in Subsection 3.1] and, e.g., [38, Section 1.1]). Furthermore, since

$\mathcal{X}\colon \{0,1,\dots,N\}\times\Omega\to\mathbb{R}^d$ is a Markov process, for every $n\in\{0,1,\dots,N\}$ we only consider functions $u_{n,\theta}\colon\mathbb{R}^d\to(0,1)$, $\theta\in\mathbb{R}^\nu$, which are defined on $\mathbb{R}^d$ instead of $(\mathbb{R}^d)^{n+1}$ and which in (29) only depend on $x_n\in\mathbb{R}^d$ instead of $(x_0,x_1,\dots,x_n)\in(\mathbb{R}^d)^{n+1}$ (cf. (29) and (17) above and [9, Theorem 1 and Remark 2 in Subsection 2.1]). We suggest to choose the functions $u_{n,\theta}\colon\mathbb{R}^d\to(0,1)$, $\theta\in\mathbb{R}^\nu$, $n\in\{0,1,\dots,N-1\}$, as multilayer feedforward neural networks (cf. [9, Corollary 5 in Subsection 2.2] and, e.g., [31, 50, 6]). For example, for every $k\in\mathbb{N}$ let $\mathcal{L}_k\colon\mathbb{R}^k\to\mathbb{R}^k$ be the function which satisfies for all $x=(x_1,\dots,x_k)\in\mathbb{R}^k$ that

$$\mathcal{L}_k(x)=\left(\frac{\exp(x_1)}{\exp(x_1)+1},\frac{\exp(x_2)}{\exp(x_2)+1},\dots,\frac{\exp(x_k)}{\exp(x_k)+1}\right), \tag{32}$$

for every $\theta=(\theta_1,\dots,\theta_\nu)\in\mathbb{R}^\nu$, $v\in\mathbb{N}_0$, $k,l\in\mathbb{N}$ with $v+k(l+1)\le\nu$ let $A^{\theta,v}_{k,l}\colon\mathbb{R}^l\to\mathbb{R}^k$ be the affine linear function which satisfies for all $x=(x_1,\dots,x_l)\in\mathbb{R}^l$ that

$$A^{\theta,v}_{k,l}(x)=\begin{pmatrix}\theta_{v+1} & \theta_{v+2} & \dots & \theta_{v+l}\\ \theta_{v+l+1} & \theta_{v+l+2} & \dots & \theta_{v+2l}\\ \theta_{v+2l+1} & \theta_{v+2l+2} & \dots & \theta_{v+3l}\\ \vdots & \vdots & \vdots & \vdots\\ \theta_{v+(k-1)l+1} & \theta_{v+(k-1)l+2} & \dots & \theta_{v+kl}\end{pmatrix}\begin{pmatrix}x_1\\x_2\\x_3\\ \vdots\\ x_l\end{pmatrix}+\begin{pmatrix}\theta_{v+kl+1}\\ \theta_{v+kl+2}\\ \theta_{v+kl+3}\\ \vdots\\ \theta_{v+kl+k}\end{pmatrix}, \tag{33}$$

and assume for all $n\in\{0,1,\dots,N-1\}$, $\theta\in\mathbb{R}^\nu$ that $\nu\ge N(2d+1)(d+1)$ and

$$u_{n,\theta}=\mathcal{L}_1\circ A^{\theta,(2nd+n)(d+1)}_{1,d}\circ\mathcal{L}_d\circ A^{\theta,(2nd+n+1)(d+1)}_{d,d}\circ\mathcal{L}_d\circ A^{\theta,((2n+1)d+n+1)(d+1)}_{d,d}. \tag{34}$$

The functions in (34) provide artificial neural networks with 4 layers (1 input layer with d neurons, 2 hidden layers each with d neurons, and 1 output layer with 1 neuron) and the multidimensional version of the standard logistic function $\mathbb{R}\ni x\mapsto {}^{\exp(x)}/_{(\exp(x)+1)}\in(0,1)$ (see (32) above) as activation functions. In our numerical simulations in Section 4 we use this type of activation function only just in front of the output layer and we employ instead the multidimensional version of the rectifier function $\mathbb{R}\ni x\mapsto\max\{x,0\}\in[0,\infty)$ as activation functions just in front of the hidden layers. But in order to keep the illustration here as short as possible we only employ the multidimensional version of the standard logistic function as activation functions in (32)–(34) above. Furthermore, note that in contrast to the choice of the functions $u_{n,\theta}\colon\mathbb{R}^d\to(0,1)$, $\theta\in\mathbb{R}^\nu$, $n\in\{0,1,\dots,N-1\}$, the choice of the functions $u_{N,\theta}\colon\mathbb{R}^d\to(0,1)$, $\theta\in\mathbb{R}^\nu$, has no influence on the approximate stopping time factors $U_{n,\theta}\colon(\mathbb{R}^d)^{n+1}\to(0,1)$, $\theta\in\mathbb{R}^\nu$, $n\in\{0,1,\dots,N\}$ (cf. (29) above).

2.5 Formulation of the objective function

Recall that we intend to compute the real number

$$\sup\left\{\mathbb{E}\big[g(t_\tau,\mathcal{X}_\tau)\big]\colon {\scriptstyle \tau\colon\Omega\to\{0,1,\dots,N\}\text{ is an}\atop \mathfrak{F}\text{-stopping time}}\right\} \tag{35}$$

as an approximation of the American option price (2) (cf. (7)). By employing neural network architectures for stopping times (cf. Subsection 2.4 above), we next propose to replace the search over all $\mathfrak{F}$-stopping times for finding the supremum in (35) by a search over the artificial neural network parameters $\theta\in\mathbb{R}^\nu$ (see (38) below). For this, observe that (28) implies for all $\mathfrak{F}$-stopping times $\tau\colon\Omega\to\{0,1,\dots,N\}$ and all $n\in\{0,1,\dots,N\}$ that

$$\mathbb{1}_{\{\tau=n\}}=\mathbb{U}_{n,\tau}(\mathcal{X}_0,\mathcal{X}_1,\dots,\mathcal{X}_n). \tag{36}$$

Therefore, for all $\mathfrak{F}$-stopping times $\tau\colon \Omega \to \{0, 1, \dots, N\}$ it holds that

$$\begin{aligned} g(t_\tau, \mathcal{X}_\tau) &= \left[\sum_{n=0}^{N} \mathbb{1}_{\{\tau=n\}}\right] g(t_\tau, \mathcal{X}_\tau) = \sum_{n=0}^{N} \mathbb{1}_{\{\tau=n\}}\, g(t_n, \mathcal{X}_n) \\ &= \sum_{n=0}^{N} \mathbb{U}_{n,\tau}(\mathcal{X}_0, \mathcal{X}_1, \dots, \mathcal{X}_n)\, g(t_n, \mathcal{X}_n). \end{aligned} \tag{37}$$

Combining this with (i) in Lemma 2.2 and (30) inspires the approximation

$$\begin{aligned} &\sup\left\{\mathbb{E}\big[g(t_\tau, \mathcal{X}_\tau)\big] : \substack{\tau\colon \Omega\to\{0,1,\dots,N\} \text{ is an} \\ \mathfrak{F}\text{-stopping time}}\right\} \\ &= \sup\left\{\mathbb{E}\left[\sum_{n=0}^{N} \mathbb{U}_{n,\tau}(\mathcal{X}_0, \mathcal{X}_1, \dots, \mathcal{X}_n)\, g(t_n, \mathcal{X}_n)\right] : \substack{\tau\colon \Omega\to\{0,1,\dots,N\} \text{ is an} \\ \mathfrak{F}\text{-stopping time}}\right\} \\ &= \sup\left\{\mathbb{E}\left[\sum_{n=0}^{N} \mathbb{V}_n(\mathcal{X}_0, \mathcal{X}_1, \dots, \mathcal{X}_n)\, g(t_n, \mathcal{X}_n)\right] : \substack{\mathbb{V}_n\colon (\mathbb{R}^d)^{n+1}\to\{0,1\},\, n\in\{0,1,\dots,N\}, \\ \text{are Borel measurable functions with} \\ \forall\, x_0,x_1,\dots,x_N\in\mathbb{R}^d\colon\ \sum_{n=0}^{N} \mathbb{V}_n(x_0,x_1,\dots,x_n)=1}\right\} \\ &= \sup\left\{\mathbb{E}\left[\sum_{n=0}^{N} \mathfrak{V}_n(\mathcal{X}_0, \mathcal{X}_1, \dots, \mathcal{X}_n)\, g(t_n, \mathcal{X}_n)\right] : \substack{\mathfrak{V}_n\colon (\mathbb{R}^d)^{n+1}\to[0,1],\, n\in\{0,1,\dots,N\}, \\ \text{are Borel measurable functions with} \\ \forall\, x_0,x_1,\dots,x_N\in\mathbb{R}^d\colon\ \sum_{n=0}^{N} \mathfrak{V}_n(x_0,x_1,\dots,x_n)=1}\right\} \\ &= \sup\left\{\mathbb{E}\left[\sum_{n=0}^{N} \mathfrak{U}_n(\mathcal{X}_0, \mathcal{X}_1, \dots, \mathcal{X}_n)\, g(t_n, \mathcal{X}_n)\right] : \substack{\mathfrak{U}_n\colon (\mathbb{R}^d)^{n+1}\to(0,1),\, n\in\{0,1,\dots,N\}, \\ \text{are Borel measurable functions with} \\ \forall\, x_0,x_1,\dots,x_N\in\mathbb{R}^d\colon\ \sum_{n=0}^{N} \mathfrak{U}_n(x_0,x_1,\dots,x_n)=1}\right\} \\ &\qquad\approx \sup\left\{\mathbb{E}\left[\sum_{n=0}^{N} U_{n,\theta}(\mathcal{X}_0, \mathcal{X}_1, \dots, \mathcal{X}_n)\, g(t_n, \mathcal{X}_n)\right] : \theta \in \mathbb{R}^\nu\right\}. \end{aligned} \tag{38}$$

In view of this, our numerical solution for approximatively computing (35) consists of trying to find an approximate maximiser of the objective function

$$\mathbb{R}^\nu \ni \theta \mapsto \mathbb{E}\left[\sum_{n=0}^{N} U_{n,\theta}(\mathcal{X}_0, \mathcal{X}_1, \dots, \mathcal{X}_n)\, g(t_n, \mathcal{X}_n)\right] \in \mathbb{R}. \tag{39}$$

2.6 Stochastic gradient ascent optimisation algorithms

Local/global maxima of the objective function (39) can be approximately reached by maximising the expectation of the random objective function

$$\mathbb{R}^\nu \times \Omega \ni (\theta, \omega) \mapsto \sum_{n=0}^{N} U_{n,\theta}(\mathcal{X}_0(\omega), \mathcal{X}_1(\omega), \dots, \mathcal{X}_n(\omega))\, g(t_n, \mathcal{X}_n(\omega)) \in \mathbb{R} \tag{40}$$

by means of a stochastic gradient ascent-type optimisation algorithm. This yields a sequence of random parameter vectors along which we expect the objective function (39) to increase. More formally, applying under suitable hyptheses stochastic gradient ascent-type optimisation algorithms to (39) results in random approximations

$$\Theta_m = (\Theta_m^{(1)}, \dots, \Theta_m^{(\nu)})\colon \Omega \to \mathbb{R}^\nu \tag{41}$$

for $m \in \{0, 1, 2, \dots\}$ of the local/global maximum points of the objective function (39), where $m \in \{0, 1, 2, \dots\}$ is the number of steps of the employed stochastic gradient ascent-type optimisation algorithm.

2.7 Price and optimal exercise time for American-style options

The approximation algorithm sketched in Subsection 2.6 above allows us to approximatively compute both the *price* (see (2) above) and an *optimal exercise strategy* for the American option (cf. Subsection 2.1). Let $M \in \mathbb{N}$ and consider a realisation $\widehat{\Theta}_M \in \mathbb{R}^\nu$ of the random variable $\Theta_M \colon \Omega \to \mathbb{R}^\nu$. Then for sufficiently large $N, \nu, M \in \mathbb{N}$ a candidate for a suitable approximation of the American option price is the real number

$$\mathbb{E}\left[\sum_{n=0}^{N} U_{n,\widehat{\Theta}_M}(\mathcal{X}_0, \mathcal{X}_1, \ldots, \mathcal{X}_n)\, g(t_n, \mathcal{X}_n)\right] \tag{42}$$

and a candidate for a suitable approximation of an optimal exercise strategy for the American option is the function

$$\Omega \ni \omega \mapsto \sum_{n=0}^{N} n\, U_{n,\widehat{\Theta}_M}(\mathcal{X}_0(\omega), \mathcal{X}_1(\omega), \ldots, \mathcal{X}_n(\omega)) \in [0, N]. \tag{43}$$

Note, however, that in general the function (43) does not take values in $\{0, 1, \ldots, N\}$ and hence is not a proper stopping time. Similarly, note that in general it is not clear whether there exists an exercise strategy such that the number (42) is equal to the expected discounted pay-off under this exercise strategy. For these reasons we suggest other candidates for suitable approximations of the price and an optimal exercise strategy for the American option. More specifically, for every $\theta \in \mathbb{R}^\nu$ let $\tau_\theta \colon \Omega \to \{0, 1, \ldots, N\}$ be the $\mathfrak{F}$-stopping time given by

$$\tau_\theta = \min\left\{ n \in \{0, 1, \ldots, N\} \colon \sum_{k=0}^{n} U_{k,\theta}(\mathcal{X}_0, \ldots, \mathcal{X}_k) \geq 1 - U_{n,\theta}(\mathcal{X}_0, \ldots, \mathcal{X}_n) \right\}. \tag{44}$$

Then for sufficiently large $N, \nu, M \in \mathbb{N}$ we use a suitable Monte Carlo approximation of the real number

$$\mathbb{E}\Big[g\big(t_{\tau_{\widehat{\Theta}_M}}, \mathcal{X}_{\tau_{\widehat{\Theta}_M}}\big) \Big] \tag{45}$$

as a suitable implementable approximation of the price of the American option (cf. (2) above and (58) below) and we use the random variable

$$\tau_{\widehat{\Theta}_M} \colon \Omega \to \{0, 1, \ldots, N\} \tag{46}$$

as a suitable implementable approximation of an optimal exercise strategy for the American option. Note that one has

$$\begin{aligned} \tau_{\widehat{\Theta}_M} &= \min\left\{ n \in \{0, 1, \ldots, N\} \colon U_{n,\widehat{\Theta}_M}(\mathcal{X}_0, \ldots, \mathcal{X}_n) \geq 1 - \sum_{k=0}^{n} U_{k,\widehat{\Theta}_M}(\mathcal{X}_0, \ldots, \mathcal{X}_k) \right\} \\ &= \min\left\{ n \in \{0, 1, \ldots, N\} \colon U_{n,\widehat{\Theta}_M}(\mathcal{X}_0, \ldots, \mathcal{X}_n) \geq \sum_{k=n+1}^{N} U_{k,\widehat{\Theta}_M}(\mathcal{X}_0, \ldots, \mathcal{X}_k) \right\}. \end{aligned} \tag{47}$$

This shows that the exercise strategy $\tau_{\widehat{\Theta}_M} \colon \Omega \to \{0, 1, \ldots, N\}$ exercises at the first time index $n \in \{0, 1, \ldots, N\}$ for which the approximate stopping time factor associated to the mesh point t_n is at least as large as the combined approximate stopping time factors associated to all later mesh points. Furthermore, observe that it holds that

$$\mathbb{E}\Big[g\big(t_{\tau_{\widehat{\Theta}_M}}, \mathcal{X}_{\tau_{\widehat{\Theta}_M}}\big) \Big] \leq \sup\Big\{ \mathbb{E}\big[g(t_\tau, \mathcal{X}_\tau) \big] \colon {\scriptstyle \tau \colon \Omega \to \{0,1,\ldots,N\} \text{ is an} \atop \mathfrak{F}\text{-stopping time}} \Big\}. \tag{48}$$

Roughly speaking, this illustrates that Monte Carlo approximations of the number (45) are typically low-biased approximations for the American option price (2). Finally, we point out that, in comparison with the deep learning based approximation method for solving optimal stopping problems in [9], the parameters $\widehat{\Theta}_M \in \mathbb{R}^\nu$ determining an approximate optimal exercise strategy (cf. (46) above) are obtained using a single training procedure to approximately maximise a single objective function (cf. (39) above) and not found recursively through a sequence of training procedures along with different random objective functions (cf. [9, Subsections 2.2–2.3]).

3 Details of the proposed algorithm

3.1 Formulation of the proposed algorithm in a special case

In this subsection we describe the proposed algorithm in the specific situation where the objective is to solve the American option pricing problem described in Subsection 2.1, where *batch normalisation* (see Ioffe & Szegedy [51]) is not employed in the proposed algorithm, and where the plain vanilla stochastic gradient ascent approximation method with a constant learning rate $\gamma \in (0, \infty)$ and without mini-batches is the employed stochastic approximation algorithm. The general framework, which includes the setting in this subsection as a special case, can be found in Subsection 3.2 below.

Framework 3.1 (Specific case). *Let $T, \gamma \in (0, \infty)$, $d, N \in \mathbb{N}$, $\nu = N(2d+1)(d+1)$, let $\mu\colon \mathbb{R}^d \to \mathbb{R}^d$, $\sigma\colon \mathbb{R}^d \to \mathbb{R}^{d\times d}$, and $g\colon [0,T]\times\mathbb{R}^d \to \mathbb{R}$ be Borel measurable functions, let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, let $\xi^m\colon \Omega \to \mathbb{R}^d$, $m \in \mathbb{N}$, be independent random variables, let $W^m\colon [0,T]\times\Omega \to \mathbb{R}^d$, $m \in \mathbb{N}$, be independent $\mathbb{P}$-standard Brownian motions with continuous sample paths, assume that $(\xi^m)_{m\in\mathbb{N}}$ and $(W^m)_{m\in\mathbb{N}}$ are independent, let $t_0, t_1, \ldots, t_N \in [0,T]$ be real numbers with $0 = t_0 < t_1 < \ldots < t_N = T$, let $\mathcal{X}^m\colon \{t_0, t_1, \ldots, t_N\}\times\Omega \to \mathbb{R}^d$, $m \in \mathbb{N}$, be the stochastic processes which satisfy for all $m \in \mathbb{N}$, $n \in \{0, 1, \ldots, N-1\}$ that $\mathcal{X}^m_{t_0} = \xi^m$ and*

$$\mathcal{X}^m_{t_{n+1}} = \mathcal{X}^m_{t_n} + \mu\big(\mathcal{X}^m_{t_n}\big)(t_{n+1} - t_n) + \sigma\big(\mathcal{X}^m_{t_n}\big)\big(W^m_{t_{n+1}} - W^m_{t_n}\big), \tag{49}$$

for every $k \in \mathbb{N}$ let $\mathcal{L}_k\colon \mathbb{R}^k \to \mathbb{R}^k$ be the function which satisfies for all $x = (x_1, \ldots, x_k) \in \mathbb{R}^k$ that

$$\mathcal{L}_k(x) = \left(\frac{\exp(x_1)}{\exp(x_1)+1}, \frac{\exp(x_2)}{\exp(x_2)+1}, \ldots, \frac{\exp(x_k)}{\exp(x_k)+1}\right), \tag{50}$$

for every $\theta = (\theta_1, \ldots, \theta_\nu) \in \mathbb{R}^\nu$, $v \in \mathbb{N}_0$, $k, l \in \mathbb{N}$ with $v + k(l+1) \le \nu$ let $A^{\theta,v}_{k,l}\colon \mathbb{R}^l \to \mathbb{R}^k$ be the function which satisfies for all $x = (x_1, \ldots, x_l) \in \mathbb{R}^l$ that

$$A^{\theta,v}_{k,l}(x) = \left(\theta_{v+kl+1} + \left[\sum_{i=1}^{l} x_i\, \theta_{v+i}\right], \ldots, \theta_{v+kl+k} + \left[\sum_{i=1}^{l} x_i\, \theta_{v+(k-1)l+i}\right]\right), \tag{51}$$

for every $\theta \in \mathbb{R}^\nu$ let $u_{n,\theta}\colon \mathbb{R}^d \to (0,1)$, $n \in \{0, 1, \ldots, N\}$, be functions which satisfy for all $n \in \{0, 1, \ldots, N-1\}$ that

$$u_{n,\theta} = \mathcal{L}_1 \circ A^{\theta,(2nd+n)(d+1)}_{1,d} \circ \mathcal{L}_d \circ A^{\theta,(2nd+n+1)(d+1)}_{d,d} \circ \mathcal{L}_d \circ A^{\theta,((2n+1)d+n+1)(d+1)}_{d,d}, \tag{52}$$

for every $n \in \{0, 1, \ldots, N\}$, $\theta \in \mathbb{R}^\nu$ let $U_{n,\theta}\colon (\mathbb{R}^d)^{n+1} \to (0,1)$ be the function which satisfies for all $x_0, x_1, \ldots, x_n \in \mathbb{R}^d$ that

$$U_{n,\theta}(x_0, x_1, \ldots, x_n) = \max\{u_{n,\theta}(x_n), n+1-N\}\left[1 - \sum_{k=0}^{n-1} U_{k,\theta}(x_0, x_1, \ldots, x_k)\right], \tag{53}$$

for every $m \in \mathbb{N}$ let $\phi^m \colon \mathbb{R}^\nu \times \Omega \to \mathbb{R}$ be the function which satisfies for all $\theta \in \mathbb{R}^\nu$, $\omega \in \Omega$ that

$$\phi^m(\theta,\omega) = \sum_{n=0}^{N} \Big[U_{n,\theta}\big(\mathcal{X}^m_{t_0}(\omega), \mathcal{X}^m_{t_1}(\omega), \ldots, \mathcal{X}^m_{t_n}(\omega)\big)\, g\big(t_n, \mathcal{X}^m_{t_n}(\omega)\big)\Big], \tag{54}$$

for every $m \in \mathbb{N}$ let $\Phi^m \colon \mathbb{R}^\nu \times \Omega \to \mathbb{R}^\nu$ be the function which satisfies for all $\theta \in \mathbb{R}^\nu$, $\omega \in \Omega$ that

$$\Phi^m(\theta,\omega) = (\nabla_\theta \phi^m)(\theta,\omega), \tag{55}$$

let $\Theta \colon \mathbb{N}_0 \times \Omega \to \mathbb{R}^\nu$ be a stochastic process which satisfies for all $m \in \mathbb{N}$ that

$$\Theta_m = \Theta_{m-1} + \gamma \cdot \Phi^m(\Theta_{m-1}), \tag{56}$$

and for every $j \in \mathbb{N}$, $\theta \in \mathbb{R}^\nu$ let $\tau_{j,\theta} \colon \Omega \to \{t_0, t_1, \ldots, t_N\}$ be the random variable given by

$$\tau_{j,\theta} = \min\bigg\{ s \in [0,T] \colon \Big(\exists\, n \in \{0,1,\ldots,N\} \colon \big[s = t_n \text{ and } \textstyle\sum_{k=0}^{n} U_{k,\theta}\big(\mathcal{X}^j_{t_0}, \ldots, \mathcal{X}^j_{t_k}\big) \geq 1 - U_{n,\theta}\big(\mathcal{X}^j_{t_0}, \ldots, \mathcal{X}^j_{t_n}\big)\big]\Big)\bigg\}. \tag{57}$$

Consider the setting in Framework 3.1, assume that μ and σ are globally Lipschitz continuous, and assume that g is continuous and at most polynomially growing. In the case of sufficiently large $N, M, J \in \mathbb{N}$ and sufficiently small $\gamma \in (0,\infty)$ we then think of the random real number

$$\frac{1}{J}\Bigg[\sum_{j=1}^{J} g\big(\tau_{M+j,\Theta_M}, \mathcal{X}^{M+j}_{\tau_{M+j,\Theta_M}}\big)\Bigg] \tag{58}$$

as an approximation of the price of the American option with the discounted pay-off function g and for every $j \in \mathbb{N}$ we think of the random variable

$$\tau_{M+j,\Theta_M} \colon \Omega \to \{t_0, t_1, \ldots, t_N\} \tag{59}$$

as an approximation of an *optimal exercise strategy* associated to the underlying time-discrete path $(\mathcal{X}^{M+j}_t)_{t \in \{t_0,t_1,\ldots,t_N\}}$ (cf. Subsection 2.1 above and Section 4 below).

3.2 Formulation of the proposed algorithm in the general case

In this subsection we extend the framework in Subsection 3.1 above and describe the proposed algorithm in the general case.

Framework 3.2. *Let $T \in (0,\infty)$, $d, N, M, \nu, \varsigma, \varrho \in \mathbb{N}$, let $g \colon [0,T] \times \mathbb{R}^d \to \mathbb{R}$ be a Borel measurable function, let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, let $t_0, t_1, \ldots, t_N \in [0,T]$ be real numbers with $0 = t_0 < t_1 < \ldots < t_N = T$, let $\mathcal{X}^{m,j} = (\mathcal{X}^{m,j,(1)}, \ldots, \mathcal{X}^{m,j,(d)}) \colon \{t_0, t_1, \ldots, t_N\} \times \Omega \to \mathbb{R}^d$, $j \in \mathbb{N}$, $m \in \mathbb{N}_0$, be i.i.d. stochastic processes, for every $n \in \{0,1,\ldots,N\}$, $\theta \in \mathbb{R}^\nu$, $\mathbf{s} \in \mathbb{R}^\varsigma$ let $u^{\theta,\mathbf{s}}_n \colon \mathbb{R}^d \to (0,1)$ be a function, for every $n \in \{0,1,\ldots,N\}$, $\theta \in \mathbb{R}^\nu$, $\mathbf{s} \in \mathbb{R}^\varsigma$ let $U^{\theta,\mathbf{s}}_n \colon (\mathbb{R}^d)^{n+1} \to (0,1)$ be the function which satisfies for all $x_0, x_1, \ldots, x_n \in \mathbb{R}^d$ that*

$$U^{\theta,\mathbf{s}}_n(x_0, x_1, \ldots, x_n) = \max\big\{u^{\theta,\mathbf{s}}_n(x_n), n+1-N\big\}\left[1 - \sum_{k=0}^{n-1} U^{\theta,\mathbf{s}}_k(x_0, x_1, \ldots, x_k)\right], \tag{60}$$

let $(J_m)_{m\in\mathbb{N}_0} \subseteq \mathbb{N}$ be a sequence, for every $m \in \mathbb{N}$, $\mathbf{s} \in \mathbb{R}^\varsigma$ let $\phi^{m,\mathbf{s}} \colon \mathbb{R}^\nu \times \Omega \to \mathbb{R}$ be the function which satisfies for all $\theta \in \mathbb{R}^\nu$, $\omega \in \Omega$ that

$$\phi^{m,\mathbf{s}}(\theta,\omega) = \frac{1}{J_m} \sum_{j=1}^{J_m} \sum_{n=0}^{N} \Big[U_n^{\theta,\mathbf{s}}\big(\mathcal{X}_{t_0}^{m,j}(\omega), \mathcal{X}_{t_1}^{m,j}(\omega), \ldots, \mathcal{X}_{t_n}^{m,j}(\omega)\big)\, g\big(t_n, \mathcal{X}_{t_n}^{m,j}(\omega)\big)\Big], \tag{61}$$

for every $m \in \mathbb{N}$, $\mathbf{s} \in \mathbb{R}^\varsigma$ let $\Phi^{m,\mathbf{s}} \colon \mathbb{R}^\nu \times \Omega \to \mathbb{R}^\nu$ be a function which satisfies for all $\omega \in \Omega$, $\theta \in \{\eta \in \mathbb{R}^\nu \colon \phi^{m,\mathbf{s}}(\cdot,\omega) \colon \mathbb{R}^\nu \to \mathbb{R}$ is differentiable at $\eta\}$ that

$$\Phi^{m,\mathbf{s}}(\theta,\omega) = (\nabla_\theta \phi^{m,\mathbf{s}})(\theta,\omega), \tag{62}$$

let $\mathcal{S} \colon \mathbb{R}^\varsigma \times \mathbb{R}^\nu \times (\mathbb{R}^d)^{\{0,1,\ldots,N-1\}\times\mathbb{N}} \to \mathbb{R}^\varsigma$ be a function, for every $m \in \mathbb{N}$ let $\Psi_m \colon \mathbb{R}^\varrho \times \mathbb{R}^\nu \to \mathbb{R}^\varrho$ and $\psi_m \colon \mathbb{R}^\varrho \to \mathbb{R}^\nu$ be functions, let $\mathbb{S} \colon \mathbb{N}_0 \times \Omega \to \mathbb{R}^\varsigma$, $\Xi \colon \mathbb{N}_0 \times \Omega \to \mathbb{R}^\varrho$, and $\Theta \colon \mathbb{N}_0 \times \Omega \to \mathbb{R}^\nu$ be stochastic processes which satisfy for all $m \in \mathbb{N}$ that

$$\mathbb{S}_m = \mathcal{S}\big(\mathbb{S}_{m-1}, \Theta_{m-1}, (\mathcal{X}_{t_n}^{m,j})_{(n,j)\in\{0,1,\ldots,N-1\}\times\mathbb{N}}\big), \tag{63}$$

$$\Xi_m = \Psi_m(\Xi_{m-1}, \Phi^{m,\mathbb{S}_m}(\Theta_{m-1})), \qquad \textit{and} \qquad \Theta_m = \Theta_{m-1} + \psi_m(\Xi_m), \tag{64}$$

for every $j \in \mathbb{N}$, $\theta \in \mathbb{R}^\nu$, $\mathbf{s} \in \mathbb{R}^\varsigma$ let $\tau^{j,\theta,\mathbf{s}} \colon \Omega \to \{t_0, t_1, \ldots, t_N\}$ be the random variable given by

$$\tau^{j,\theta,\mathbf{s}} = \min\Bigg\{ s \in [0,T] \colon \Big(\exists\, n \in \{0,1,\ldots,N\} \colon \big[s = t_n \textit{ and } \textstyle\sum_{k=0}^{n} U_k^{\theta,\mathbf{s}}\big(\mathcal{X}_{t_0}^{0,j},\ldots,\mathcal{X}_{t_k}^{0,j}\big) \geq 1 - U_n^{\theta,\mathbf{s}}\big(\mathcal{X}_{t_0}^{0,j},\ldots,\mathcal{X}_{t_n}^{0,j}\big)\big]\Big)\Bigg\}, \tag{65}$$

and let $\mathcal{P} \colon \Omega \to \mathbb{R}$ be the random variable which satisfies for all $\omega \in \Omega$ that

$$\mathcal{P}(\omega) = \frac{1}{J_0}\left[\sum_{j=1}^{J_0} g\big(\tau^{j,\Theta_M(\omega),\mathbb{S}_M(\omega)}(\omega), \mathcal{X}^{0,j}_{\tau^{j,\Theta_M(\omega),\mathbb{S}_M(\omega)}(\omega)}(\omega)\big)\right]. \tag{66}$$

Consider the setting in Framework 3.2. Under suitable further assumptions, in the case of sufficiently large $N, M, \nu, J_0 \in \mathbb{N}$ we think of the random real number

$$\mathcal{P} = \frac{1}{J_0}\left[\sum_{j=1}^{J_0} g\big(\tau^{j,\Theta_M,\mathbb{S}_M}, \mathcal{X}^{0,j}_{\tau^{j,\Theta_M,\mathbb{S}_M}}\big)\right] \tag{67}$$

as an approximation of the price of the American option with the discounted pay-off function g and for every $j \in \mathbb{N}$ we think of the random variable

$$\tau^{j,\Theta_M,\mathbb{S}_M} \colon \Omega \to \{t_0, t_1, \ldots, t_N\} \tag{68}$$

as an approximation of an *optimal exercise strategy* associated to the underlying time-discrete path $(\mathcal{X}_t^{0,j})_{t\in\{t_0,t_1,\ldots,t_N\}}$ (cf. Subsection 2.1 above and Section 4 below).

3.3 Comments on the proposed algorithm

Note that the lack in Framework 3.2 of any assumptions on the dynamics of the stochastic process $(\mathcal{X}_t^{0,1})_{t\in\{t_0,t_1,\ldots,t_N\}}$ allows us to approximatively compute the optimal pay-off as well as an optimal exercise strategy for very general optimal stopping problems where, in particular, the stochastic process under consideration is not necessarily a solution of any SDE. We only require that the stochastic process $(\mathcal{X}_t^{0,1})_{t\in\{t_0,t_1,\dots,t_N\}}$ can be simulated efficiently and formally we still rely on the Markov assumption (cf. Subsection 2.4 above). In addition, observe that any particular choice of the functions $u_N^{\theta,\mathbf{s}}\colon \mathbb{R}^d \to (0,1)$, $\mathbf{s} \in \mathbb{R}^\varsigma$, $\theta \in \mathbb{R}^\nu$, has no influence on the proposed algorithm (cf. (60)). Furthermore, the dynamics in (64) associated with the stochastic processes $(\Xi_m)_{m\in\mathbb{N}_0}$ and $(\Theta_m)_{m\in\mathbb{N}_0}$ allow us to incorporate different stochastic approximation algorithms such as

- plain vanilla stochastic gradient ascent with or without mini-batches (cf. (56) above) as well as
- adaptive moment estimation (Adam) with mini-batches (cf. Kingma & Ba [58] and (92)–(93) below)

into the algorithm in Subsection 3.2 (cf. E, Han, & Jentzen [35, Subsection 3.3]). The dynamics in (63) associated with the stochastic process $(\mathbb{S}_m)_{m\in\mathbb{N}_0}$ in turn, allow us to incorporate batch normalisation (cf. Ioffe & Szegedy [51] and Section 4 below) into the algorithm in Subsection 3.2. In that case we think of $(\mathbb{S}_m)_{m\in\mathbb{N}_0}$ as a bookkeeping process keeping track of approximatively calculated means and standard deviations as well as of the number of steps $m \in \mathbb{N}_0$ of the employed stochastic approximation algorithm.

4 Numerical examples of pricing American-style derivatives

In this section we test the algorithm in Framework 3.2 on several different examples of pricing American-style financial derivatives.

In each of the examples below we employ the general approximation algorithm in Framework 3.2 above in conjunction with the Adam optimiser (see Kingma & Ba [58]) with varying learning rates and with mini-batches (see Subsection 4.2 below for a precise description).

Furthermore, in the context of Framework 3.2 we employ $N-1$ fully-connected feedforward neural networks in each of our implementations for the examples below where the initial value $\mathcal{X}_{t_0}^{0,1}$ is deterministic. In that case the data entering the functions $u_0^{\theta,\mathbf{s}}\colon \mathbb{R}^d \to (0,1)$, $\mathbf{s} \in \mathbb{R}^\varsigma$, $\theta \in \mathbb{R}^\nu$, is deterministic (cf. (60)–(61)). Therefore, a training procedure is not necessary for the approximative calculations of these functions but is only carried out for the functions $u_1^{\theta,\mathbf{s}}, \dots, u_{N-1}^{\theta,\mathbf{s}}\colon \mathbb{R}^d \to (0,1)$, $\mathbf{s} \in \mathbb{R}^\varsigma$, $\theta \in \mathbb{R}^\nu$. If, however, the initial value $\mathcal{X}_{t_0}^{0,1}$ is not deterministic (see the example in Subsection 4.4.4 below), a training procedure is carried out for all the functions $u_0^{\theta,\mathbf{s}}, u_1^{\theta,\mathbf{s}}, \dots, u_{N-1}^{\theta,\mathbf{s}}\colon \mathbb{R}^d \to (0,1)$, $\mathbf{s} \in \mathbb{R}^\varsigma$, $\theta \in \mathbb{R}^\nu$, and in that case we hence employ N fully-connected feedforward neural networks (cf. [9, Remark 6 in Subsection 2.3]).

All neural networks employed have one input layer, two hidden layers, and one output layer. As non-linear activation functions just in front of the two hidden layers we employ the multidimensional version of the rectifier function $\mathbb{R} \ni x \mapsto \max\{x,0\} \in [0,\infty)$, whereas just in front of the output layer we employ the standard logistic function $\mathbb{R} \ni x \mapsto {}^{\exp(x)}/_{(\exp(x)+1)} \in (0,1)$ as non-linear activation function. In addition, batch normalisation (see Ioffe & Szegedy [51]) is applied just before the first linear transformation, just before each of the two non-linear activation functions in front of the hidden layers as well as just before the non-linear activation function in front of the output layer. We use Xavier initialisation (see Glorot & Bengio [43]) to initialise all weights in the neural networks.

All the examples presented below were implemented in Python. The corresponding Python codes were run, unless stated otherwise (see Subsection 4.4.1.2 as well as the last sentence in Subsection 4.4.1.3 below), in single precision (float32) on a NVIDIA GeForce GTX 1080 GPU with 1974 MHz core clock and 8 GB GDDR5X memory with 1809.5 MHz clock rate, where the underlying system consisted of an Intel Core i7-6800K 3.4 GHz CPU with 64 GB DDR4-2133 memory running Tensorflow 1.5 on Ubuntu 16.04. We would like to point out that no special emphasis has been put on optimising computation speed. In many cases some of the algorithm parameters could be adjusted in order to obtain similarly accurate results in shorter runtime.

4.1 Theoretical considerations

Before we present the optimal stopping problem examples on which we have tested the algorithm in Framework 3.2 (see Subsections 4.3–4.4 below), we recall a few theoretical results, which are used to design some of these examples and to provide reference values. The elementary and well-known result in Lemma 4.1 below specifies the distributions of linear combinations of independent and identically distributed centred Gaussian random variables which take values in a separable normed $\mathbb{R}$-vector space.

Lemma 4.1. *Let $n \in \mathbb{N}$, $\gamma = (\gamma_1, \dots, \gamma_n) \in \mathbb{R}^n$, let $(V, \|\cdot\|_V)$ be a separable normed $\mathbb{R}$-vector space, let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, and let $X_i \colon \Omega \to V$, $i \in \{1, \dots, n\}$, be i.i.d. centred Gaussian random variables. Then it holds that*

$$\left(\sum_{i=1}^{n} \gamma_i \, X_i \right) (\mathbb{P})_{\mathcal{B}(V)} = (\|\gamma\|_{\mathbb{R}^n} \, X_1)(\mathbb{P})_{\mathcal{B}(V)}. \tag{69}$$

Proof of Lemma 4.1. Throughout this proof let $Y_1, Y_2 \colon \Omega \to V$ be the random variables given by $Y_1 = \sum_{i=1}^{n} \gamma_i \, X_i$ and $Y_2 = \|\gamma\|_{\mathbb{R}^n} \, X_1$. Note that for every $\varphi \in V'$ it holds that $\varphi \circ X_i \colon \Omega \to \mathbb{R}$, $i \in \{1, \dots, n\}$, are independent and identically distributed centred Gaussian random variables. This implies that for all $\varphi \in V'$ it holds that

$$\begin{aligned}
\mathbb{E}\big[e^{\mathbf{i}\,\varphi(Y_1)}\big] &= \mathbb{E}\big[e^{\mathbf{i}\sum_{i=1}^{n} \gamma_i\,\varphi(X_i)}\big] = \mathbb{E}\left[\prod_{i=1}^{n} e^{\mathbf{i}\,\gamma_i\,\varphi(X_i)}\right] = \prod_{i=1}^{n} \mathbb{E}\big[e^{\mathbf{i}\,(\gamma_i\,\varphi)(X_i)}\big] \\
&= \prod_{i=1}^{n} \exp\!\big(-\tfrac{1}{2}\,\mathbb{E}\big[|(\gamma_i\,\varphi)(X_i)|^2\big]\big) = \prod_{i=1}^{n} \exp\!\big(-\tfrac{1}{2}\,\mathbb{E}\big[|(\gamma_i\,\varphi)(X_1)|^2\big]\big) \\
&= \exp\!\left(-\tfrac{1}{2}\,\mathbb{E}\left[\sum_{i=1}^{n} |\gamma_i\,\varphi(X_1)|^2\right]\right) = \exp\!\big(-\tfrac{1}{2}\,\mathbb{E}\big[|(\|\gamma\|_{\mathbb{R}^n}\,\varphi)(X_1)|^2\big]\big) \\
&= \mathbb{E}\big[e^{\mathbf{i}\,\|\gamma\|_{\mathbb{R}^n}\,\varphi(X_1)}\big] = \mathbb{E}\big[e^{\mathbf{i}\,\varphi(Y_2)}\big].
\end{aligned} \tag{70}$$

This and, e.g., Jentzen, Salimova, & Welti [54, Lemma 4.10] establish that

$$Y_1(\mathbb{P})_{\mathcal{B}(V)} = Y_2(\mathbb{P})_{\mathcal{B}(V)}. \tag{71}$$

The proof of Lemma 4.1 is thus complete. □

The next elementary and well-known corollary follows directly from Lemma 4.1.

Corollary 4.2. *Let $d \in \mathbb{N}$, $\gamma = (\gamma_1, \dots, \gamma_d) \in \mathbb{R}^d$, let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, and let $W = (W^{(1)}, \dots, W^{(d)}) \colon [0, T] \times \Omega \to \mathbb{R}^d$ be a $\mathbb{P}$-standard Brownian motion with continuous sample paths. Then it holds that*

$$\left(\sum_{i=1}^{d} \gamma_i \, W^{(i)} \right) (\mathbb{P})_{\mathcal{B}(C([0,T],\mathbb{R}))} = \big(\|\gamma\|_{\mathbb{R}^d} \, W^{(1)}\big)(\mathbb{P})_{\mathcal{B}(C([0,T],\mathbb{R}))}. \tag{72}$$

The next elementary and well-known result, Proposition 4.3 below, states that the distribution of a product of multiple correlated geometric Brownian motions is equal to the distribution of a single particular geometric Brownian motion.

Proposition 4.3. *Let $T, \epsilon \in (0,\infty)$, $d \in \mathbb{N}$, $\mathfrak{S} = (\varsigma_1, \ldots, \varsigma_d) \in \mathbb{R}^{d\times d}$, $\xi = (\xi_1, \ldots, \xi_d), \alpha = (\alpha_1, \ldots, \alpha_d), \beta = (\beta_1, \ldots, \beta_d) \in \mathbb{R}^d$, let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, let $\mathscr{F}^{(i)} = (\mathscr{F}^{(i)}_t)_{t\in[0,T]}$, $i \in \{1,2\}$, be filtrations on $(\Omega, \mathcal{F}, \mathbb{P})$ that satisfy the usual conditions, let $W = (W^{(1)}, \ldots, W^{(d)})\colon [0,T]\times\Omega \to \mathbb{R}^d$ be a standard $(\Omega, \mathcal{F}, \mathbb{P}, \mathscr{F}^{(1)})$-Brownian motion with continuous sample paths, let $\mathrm{w}\colon [0,T]\times\Omega\to\mathbb{R}$ be a standard $(\Omega, \mathcal{F}, \mathbb{P}, \mathscr{F}^{(2)})$-Brownian motion with continuous sample paths, let $\mu\colon \mathbb{R}^d \to \mathbb{R}^d$, $\sigma\colon \mathbb{R}^d\to\mathbb{R}^{d\times d}$, $\mathbf{P}\colon C([0,T],\mathbb{R}^d) \to C([0,T],\mathbb{R})$, and $\mathbf{G}\colon C([0,T],\mathbb{R})\to C([0,T],\mathbb{R})$ be the functions which satisfy for all $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$, $u^{(1)} = (u^{(1)}_s)_{s\in[0,T]}, \ldots, u^{(d)} = (u^{(d)}_s)_{s\in[0,T]} \in C([0,T],\mathbb{R})$, $t\in[0,T]$ that $\mu(x) = (\alpha_1 x_1, \ldots, \alpha_d x_d)$, $\sigma(x) = \operatorname{diag}(\beta_1 x_1, \ldots, \beta_d x_d)\,\mathfrak{S}^*$, $(\mathbf{G}[u^{(1)}])_t = \exp\!\big(\epsilon\,[\sum_{i=1}^d \alpha_i - \|\beta_i\varsigma_i\|^2_{\mathbb{R}^d}/2]\,t + \epsilon\,\|\mathfrak{S}\,\beta\|_{\mathbb{R}^d}\,u^{(1)}_t\big)\prod_{i=1}^d|\xi_i|^\epsilon$, and $(\mathbf{P}[(u^{(1)}, \ldots, u^{(d)})])_t = \prod_{i=1}^d\big|u^{(i)}_t\big|^\epsilon$, let $X = (X^{(1)}, \ldots, X^{(d)})\colon [0,T]\times\Omega\to\mathbb{R}^d$ be an $\mathscr{F}^{(1)}$-adapted stochastic process with continuous sample paths, let $Y\colon [0,T]\times\Omega\to\mathbb{R}$ be an $\mathscr{F}^{(2)}$-adapted stochastic process with continuous sample paths, and assume that for all $t\in[0,T]$ it holds $\mathbb{P}$-a.s. that*

$$X_t = \xi + \int_0^t \mu(X_s)\,ds + \int_0^t \sigma(X_s)\,dW_s, \tag{73}$$

$$Y_t = \prod_{i=1}^d |\xi_i|^\epsilon + \left(\epsilon\left[\sum_{i=1}^d \alpha_i - \tfrac{\|\beta_i\varsigma_i\|^2_{\mathbb{R}^d}}{2}\right] + \tfrac{\|\epsilon\,\mathfrak{S}\,\beta\|^2_{\mathbb{R}^d}}{2}\right)\int_0^t Y_s\,ds + \epsilon\,\|\mathfrak{S}\,\beta\|_{\mathbb{R}^d}\int_0^t Y_s\,d\mathrm{w}_s. \tag{74}$$

Then

(i) for all $i\in\{1,\ldots,d\}$, $t\in[0,T]$ it holds $\mathbb{P}$-a.s. that

$$X^{(i)}_t = \exp\!\left(\left[\alpha_i - \frac{\|\beta_i\varsigma_i\|^2_{\mathbb{R}^d}}{2}\right]t + \beta_i\langle\varsigma_i, W_t\rangle_{\mathbb{R}^d}\right)\xi_i, \tag{75}$$

(ii) it holds that $\mathbf{P}$ and $\mathbf{G}$ are continuous functions, and

(iii) it holds that

$$(\mathbf{P}\circ X)(\mathbb{P})_{\mathcal{B}(C([0,T],\mathbb{R}))} = (\mathbf{G}\circ \mathrm{w})(\mathbb{P})_{\mathcal{B}(C([0,T],\mathbb{R}))} = Y(\mathbb{P})_{\mathcal{B}(C([0,T],\mathbb{R}))}. \tag{76}$$

Proof of Proposition 4.3. Throughout this proof let $\gamma = (\gamma_1, \ldots, \gamma_d)\in\mathbb{R}^d$ be the vector given by $\gamma = \mathfrak{S}\,\beta$, let $Z^{(i)}\colon [0,T]\times\Omega\to\mathbb{R}$, $i\in\{1,\ldots,d\}$, be the stochastic processes which satisfy for all $i\in\{1,\ldots,d\}$, $t\in[0,T]$ that

$$Z^{(i)}_t = \left[\alpha_i - \frac{\|\beta_i\varsigma_i\|^2_{\mathbb{R}^d}}{2}\right]t + \beta_i\langle\varsigma_i, W_t\rangle_{\mathbb{R}^d}, \tag{77}$$

and let $\tilde{\mathbf{G}}\colon C([0,T],\mathbb{R})\to C([0,T],\mathbb{R})$ be the function which satisfies for all $u = (u_s)_{s\in[0,T]} \in C([0,T],\mathbb{R})$, $t\in[0,T]$ that

$$(\tilde{\mathbf{G}}[u])_t = \exp\!\left(\epsilon\left[\sum_{i=1}^d \alpha_i - \frac{\|\beta_i\varsigma_i\|^2_{\mathbb{R}^d}}{2}\right]t + \epsilon\,u_t\right)\prod_{i=1}^d|\xi_i|^\epsilon. \tag{78}$$

Observe that for all $i\in\{1,\ldots,d\}$, $t\in[0,T]$ it holds $\mathbb{P}$-a.s. that

$$X^{(i)}_t = \xi_i + \alpha_i\int_0^t X^{(i)}_s\,ds + \beta_i\int_0^t X^{(i)}_s\langle\varsigma_i, dW_s\rangle_{\mathbb{R}^d}. \tag{79}$$

In addition, note that (77) implies that for all $i \in \{1, \ldots, d\}$, $t \in [0, T]$ it holds $\mathbb{P}$-a.s. that

$$Z_t^{(i)} = \int_0^t \alpha_i - \frac{\|\beta_i \varsigma_i\|_{\mathbb{R}^d}^2}{2}\, ds + \int_0^t \beta_i \langle \varsigma_i, dW_s \rangle_{\mathbb{R}^d}. \tag{80}$$

Itô's formula hence shows that for all $i \in \{1, \ldots, d\}$, $t \in [0, T]$ it holds $\mathbb{P}$-a.s. that

$$\begin{aligned} e^{Z_t^{(i)}} \xi_i &= \xi_i + \left[\alpha_i - \frac{\|\beta_i \varsigma_i\|_{\mathbb{R}^d}^2}{2}\right] \int_0^t e^{Z_s^{(i)}} \xi_i \, ds + \beta_i \int_0^t e^{Z_s^{(i)}} \xi_i \langle \varsigma_i, dW_s \rangle_{\mathbb{R}^d} \\ &\quad + \frac{\|\beta_i \varsigma_i\|_{\mathbb{R}^d}^2}{2} \int_0^t e^{Z_s^{(i)}} \xi_i \, ds \\ &= \xi_i + \alpha_i \int_0^t e^{Z_s^{(i)}} \xi_i \, ds + \beta_i \int_0^t e^{Z_s^{(i)}} \xi_i \langle \varsigma_i, dW_s \rangle_{\mathbb{R}^d}. \end{aligned} \tag{81}$$

Combining this and (79) with, e.g., Da Prato & Zabczyk [32, (i) in Theorem 7.4] proves that for all $i \in \{1, \ldots, d\}$, $t \in [0, T]$ it holds $\mathbb{P}$-a.s. that

$$X_t^{(i)} = e^{Z_t^{(i)}} \xi_i = \exp\!\left(\left[\alpha_i - \frac{\|\beta_i \varsigma_i\|_{\mathbb{R}^d}^2}{2}\right] t + \beta_i \langle \varsigma_i, W_t \rangle_{\mathbb{R}^d} \right) \xi_i. \tag{82}$$

This establishes (i). In the next step note that (ii) is clear. It thus remains to prove (iii). For this observe that (i) establishes that for all $t \in [0, T]$ it holds $\mathbb{P}$-a.s. that

$$\begin{aligned} (\mathbf{P}[X])_t &= \prod_{i=1}^d |X_t^{(i)}|^\epsilon = \prod_{i=1}^d \left[\exp\!\left(\epsilon \left[\alpha_i - \frac{\|\beta_i \varsigma_i\|_{\mathbb{R}^d}^2}{2}\right] t + \epsilon\, \beta_i \langle \varsigma_i, W_t \rangle_{\mathbb{R}^d} \right) |\xi_i|^\epsilon \right] \\ &= \exp\!\left(\epsilon \left[\sum_{i=1}^d \alpha_i - \frac{\|\beta_i \varsigma_i\|_{\mathbb{R}^d}^2}{2} \right] t + \epsilon \left\langle \sum_{i=1}^d \varsigma_i \beta_i, W_t \right\rangle_{\mathbb{R}^d} \right) \prod_{i=1}^d |\xi_i|^\epsilon \\ &= \exp\!\left(\epsilon \left[\sum_{i=1}^d \alpha_i - \frac{\|\beta_i \varsigma_i\|_{\mathbb{R}^d}^2}{2} \right] t + \epsilon \, \langle \gamma, W_t \rangle_{\mathbb{R}^d} \right) \prod_{i=1}^d |\xi_i|^\epsilon \\ &= \left(\tilde{\mathbf{G}} \left[\sum_{i=1}^d \gamma_i \, W^{(i)} \right] \right)_t. \end{aligned} \tag{83}$$

Continuity hence implies that it holds $\mathbb{P}$-a.s. that

$$\mathbf{P}[X] = \tilde{\mathbf{G}} \left[\sum_{i=1}^d \gamma_i \, W^{(i)} \right]. \tag{84}$$

Moreover, note that (i) shows that for all $t \in [0, T]$ it holds $\mathbb{P}$-a.s. that

$$\begin{aligned} Y_t &= \exp\!\left(\left\{ \epsilon \left[\sum_{i=1}^d \alpha_i - \frac{\|\beta_i \varsigma_i\|_{\mathbb{R}^d}^2}{2} \right] + \frac{\|\epsilon \, \mathfrak{S} \, \beta\|_{\mathbb{R}^d}^2}{2} - \frac{\|\epsilon \, \mathfrak{S} \, \beta\|_{\mathbb{R}^d}^2}{2} \right\} t + \epsilon \, \|\mathfrak{S} \, \beta\|_{\mathbb{R}^d} \, \mathrm{w}_t \right) \prod_{i=1}^d |\xi_i|^\epsilon \\ &= \exp\!\left(\epsilon \left[\sum_{i=1}^d \alpha_i - \frac{\|\beta_i \varsigma_i\|_{\mathbb{R}^d}^2}{2} \right] t + \epsilon \, \|\mathfrak{S} \, \beta\|_{\mathbb{R}^d} \, \mathrm{w}_t \right) \prod_{i=1}^d |\xi_i|^\epsilon \\ &= (\mathbf{G}[\mathrm{w}])_t. \end{aligned} \tag{85}$$

This and continuity establish that it holds $\mathbb{P}$-a.s. that

$$Y = \mathbf{G}[\mathrm{w}]. \tag{86}$$

Furthermore, observe that Corollary 4.2 ensures that

$$\left(\sum_{i=1}^{d} \gamma_i \, W^{(i)}\right)(\mathbb{P})_{\mathcal{B}(C([0,T],\mathbb{R}))} = \big(\|\gamma\|_{\mathbb{R}^d} \, W^{(1)}\big)(\mathbb{P})_{\mathcal{B}(C([0,T],\mathbb{R}))}. \tag{87}$$

The fact that $\tilde{\mathbf{G}} \colon C([0,T],\mathbb{R}) \to C([0,T],\mathbb{R})$ is a Borel measurable function, (ii), the fact that $\forall\, u \in C([0,T],\mathbb{R}) \colon \tilde{\mathbf{G}}[\|\gamma\|_{\mathbb{R}^d} \, u] = \mathbf{G}[u]$, (84), and (86) hence demonstrate that

$$\begin{aligned}
(\mathbf{P} \circ X)(\mathbb{P})_{\mathcal{B}(C([0,T],\mathbb{R}))} &= \left(\tilde{\mathbf{G}} \circ \left(\sum_{i=1}^{d} \gamma_i \, W^{(i)}\right)\right)(\mathbb{P})_{\mathcal{B}(C([0,T],\mathbb{R}))} \\
&= \big(\tilde{\mathbf{G}} \circ \big(\|\gamma\|_{\mathbb{R}^d} \, W^{(1)}\big)\big)(\mathbb{P})_{\mathcal{B}(C([0,T],\mathbb{R}))} = \big(\mathbf{G} \circ W^{(1)}\big)(\mathbb{P})_{\mathcal{B}(C([0,T],\mathbb{R}))} \\
&= (\mathbf{G} \circ \mathrm{w})(\mathbb{P})_{\mathcal{B}(C([0,T],\mathbb{R}))} = Y(\mathbb{P})_{\mathcal{B}(C([0,T],\mathbb{R}))}.
\end{aligned} \tag{88}$$

The proof of Proposition 4.3 is thus complete. □

In the next result, Lemma 4.4 below, we recall the well-known formula for the price of a European call option on a single stock in the Black–Scholes model.

Lemma 4.4. *Let $T, \xi, \sigma \in (0,\infty)$, $r, c \in \mathbb{R}$, let $\Phi \colon \mathbb{R} \to \mathbb{R}$ be the function which satisfies for all $x \in \mathbb{R}$ that $\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} \, e^{-\frac{1}{2}y^2} \, dy$, let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space with a filtration $\mathscr{F} = (\mathscr{F}_t)_{t \in [0,T]}$ that satisfies the usual conditions, let $W \colon [0,T] \times \Omega \to \mathbb{R}$ be a standard $(\Omega, \mathcal{F}, \mathbb{P}, \mathscr{F})$-Brownian motion with continuous sample paths, and let $X \colon [0,T] \times \Omega \to \mathbb{R}$ be an $\mathscr{F}$-adapted stochastic process with continuous sample paths which satisfies that for all $t \in [0,T]$ it holds $\mathbb{P}$-a.s. that*

$$X_t = \xi + (r - c) \int_0^t X_s \, ds + \sigma \int_0^t X_s \, dW_s. \tag{89}$$

Then it holds for all $K \in \mathbb{R}$ that

$$\begin{aligned}
&\mathbb{E}\big[e^{-rT} \max\{X_T - K, 0\}\big] \\
&= \begin{cases} e^{-cT} \xi \, \Phi\Big(\frac{(r - c + \frac{\sigma^2}{2})T + \ln(\xi/K)}{\sigma\sqrt{T}}\Big) - K e^{-rT} \, \Phi\Big(\frac{(r - c - \frac{\sigma^2}{2})T + \ln(\xi/K)}{\sigma\sqrt{T}}\Big) & : K > 0 \\ e^{-cT}\xi - K e^{-rT} & : K \leq 0 \end{cases}.
\end{aligned} \tag{90}$$

4.2 Setting

Framework 4.5. *Assume Framework 3.2, let $\zeta_1 = 0.9$, $\zeta_2 = 0.999$, $\varepsilon \in (0,\infty)$, $(\gamma_m)_{m \in \mathbb{N}} \subseteq (0,\infty)$, $\xi = (\xi_1, \dots, \xi_d) \in \mathbb{R}^d$, let $\mathscr{F} = (\mathscr{F}_t)_{t \in [0,T]}$ be a filtration on $(\Omega, \mathcal{F}, \mathbb{P})$ that satisfies the usual conditions, let $W^{m,j} = (W^{m,j,(1)}, \dots, W^{m,j,(d)}) \colon [0,T] \times \Omega \to \mathbb{R}^d$, $j \in \mathbb{N}$, $m \in \mathbb{N}_0$, be independent standard $(\Omega, \mathcal{F}, \mathbb{P}, \mathscr{F})$-Brownian motions with continuous sample paths, let $\mu \colon \mathbb{R}^d \to \mathbb{R}^d$ and $\sigma \colon \mathbb{R}^d \to \mathbb{R}^{d \times d}$ be Lipschitz continuous functions, let $X = (X^{(1)}, \dots, X^{(d)}) \colon [0,T] \times \Omega \to \mathbb{R}^d$ be an $\mathscr{F}$-adapted stochastic process with continuous sample paths which satisfies that for all $t \in [0,T]$ it holds $\mathbb{P}$-a.s. that*

$$X_t = \xi + \int_0^t \mu(X_s) \, ds + \int_0^t \sigma(X_s) \, dW_s^{0,1}, \tag{91}$$

assume for all $n \in \{0, 1, \dots, N\}$ that $\varrho = 2\nu$, $\Xi_0 = 0$, and $t_n = \frac{nT}{N}$, and assume for all $m \in \mathbb{N}$, $x = (x_1, \dots, x_\nu)$, $y = (y_1, \dots, y_\nu)$, $\eta = (\eta_1, \dots, \eta_\nu) \in \mathbb{R}^\nu$ that

$$\Psi_m(x, y, \eta) = \big(\zeta_1 x + (1 - \zeta_1)\eta, \zeta_2 y + (1 - \zeta_2)((\eta_1)^2, \dots, (\eta_\nu)^2)\big) \tag{92}$$

and

$$\psi_m(x,y) = \left(\left[\sqrt{\tfrac{|y_1|}{1-(\zeta_2)^m}} + \varepsilon \right]^{-1} \frac{\gamma_m x_1}{1-(\zeta_1)^m}, \ldots, \left[\sqrt{\tfrac{|y_\nu|}{1-(\zeta_2)^m}} + \varepsilon \right]^{-1} \frac{\gamma_m x_\nu}{1-(\zeta_1)^m} \right). \tag{93}$$

Equations (92)–(93) in Framework 4.5 describe the Adam optimiser with possibly varying learning rates (cf. Kingma & Ba [58] and, e.g., E, Han, & Jentzen [35, (4.3)–(4.4) in Subsection 4.1 and (5.4)–(5.5) in Subsection 5.2]). Furthermore, in the context of pricing American-style financial derivatives, we think

- of T as the maturity,
- of d as the dimension of the associated optimal stopping problem,
- of N as the time discretisation parameter employed,
- of M as the total number of training steps employed in the Adam optimiser,
- of g as the discounted pay-off function,
- of $\{t_0, t_1, \ldots, t_N\}$ as the discrete time grid employed,
- of J_0 as the number of Monte Carlo samples employed in the final integration for the price approximation,
- of $(J_m)_{m\in\mathbb{N}}$ as the sequence of batch sizes employed in the Adam optimiser,
- of ζ_1 as the momentum decay factor, of ζ_2 as the second momentum decay factor, and of ε as the regularising factor employed in the Adam optimiser,
- of $(\gamma_m)_{m\in\mathbb{N}}$ as the sequence of learning rates employed in the Adam optimiser,
- and, where applicable, of X as a continuous-time model for d underlying stock prices with initial prices ξ, drift coefficient function μ, and diffusion coefficient function σ.

Moreover, note that for every $m \in \mathbb{N}_0$, $j \in \mathbb{N}$ the stochastic processes $W^{m,j,(1)} = (W_t^{m,j,(1)})_{t\in[0,T]}$, $\ldots$, $W^{m,j,(d)} = (W_t^{m,j,(d)})_{t\in[0,T]}$ are the components of the d-dimensional standard Brownian motion $W^{m,j} = (W_t^{m,j})_{t\in[0,T]}$ and hence each a one-dimensional standard Brownian motion.

4.3 Examples with known one-dimensional representation

In this subsection we test the algorithm in Framework 3.2 in the case of several very simple optimal stopping problem examples in which the d-dimensional optimal stopping problem under consideration has been designed in such a way that it can be represented as a one-dimensional optimal stopping problem. This representation allows us to employ a numerical method for the one-dimensional optimal stopping problem to compute reference values for the original d-dimensional optimal stopping problem. We refer to Subsection 4.4 below for more challenging examples where a one-dimensional representation is not known.

Dimension d	Mean of $\mathcal{P}$	Standard deviation of $\mathcal{P}$	Runtime in sec. for one realisation of $\mathcal{P}$
1	7.890	0.004	4.2
5	7.892	0.007	4.2
10	7.892	0.005	4.4
50	7.890	0.005	5.5
100	7.891	0.004	7.3
500	7.891	0.005	24.3
1000	7.892	0.007	54.4

Table 1: Numerical simulations of the algorithm in Framework 3.2 for optimally stopping a correlated Brownian motion in the case of the Bermudan two-exercise put-type example in Subsection 4.3.1.1.

4.3.1 Optimal stopping of a Brownian motion

4.3.1.1 A Bermudan two-exercise put-type example

In this subsection we test the algorithm in Framework 3.2 on the example of optimally stopping a correlated Brownian motion under a put option inspired pay-off function with two possible exercise dates.

Assume Framework 4.5, let $r = 0.02 = 2\%$, $\beta = 0.3 = 30\%$, $\chi = 95$, $K = 90$, $Q = (Q_{i,j})_{(i,j)\in\{1,\ldots,d\}^2}, \mathfrak{S} \in \mathbb{R}^{d\times d}$ satisfy for all $i \in \{1,\ldots,d\}$ that $Q_{i,i} = 1$, $\forall\, j \in \{1,\ldots,d\} \setminus \{i\} \colon Q_{i,j} = 0.1$, and $\mathfrak{S}\,\mathfrak{S}^* = Q$, let $\mathbb{F} = (\mathbb{F}_t)_{t\in[0,T]}$ be the filtration generated by $W^{0,1}$, and assume for all $m, j \in \mathbb{N}$, $n \in \{0, 1, 2\}$, $s \in [0, T]$, $x = (x_1,\ldots,x_d) \in \mathbb{R}^d$ that $T = 1$, $N = 2$, $M = 500$, $\mathcal{X}^{m-1,j}_{t_n} = \mathfrak{S}\, W^{m-1,j}_{t_n}$, $J_0 = 4\,096\,000$, $J_m = 8192$, $\varepsilon = 10^{-8}$, $\gamma_m = 5\,[10^{-2}\,\mathbb{1}_{[1,100]}(m) + 10^{-3}\,\mathbb{1}_{(100,300]}(m) + 10^{-4}\,\mathbb{1}_{(300,\infty)}(m)]$, and

$$g(s, x) = e^{-rs} \max\!\left\{K - \exp\!\left(\left[r - \tfrac{1}{2}\beta^2\right]s + \tfrac{\beta\sqrt{10}}{\sqrt{d(d+9)}}[x_1 + \ldots + x_d]\right)\chi, 0\right\}. \tag{94}$$

Note that the distribution of the random variable $\Omega \ni \omega \mapsto ([0, T] \ni t \mapsto g(t, \mathfrak{S}\, W^{0,1}_t(\omega)) \in \mathbb{R}) \in C([0, T], \mathbb{R})$ does not depend on the dimension d (cf. Corollary 4.2). The random variable $\mathcal{P}$ provides approximations for the real number

$$\sup\!\left\{\mathbb{E}\big[g(\tau, \mathfrak{S}\, W^{0,1}_\tau)\big] \colon {\scriptstyle \tau\colon \Omega\to\{t_0,t_1,t_2\} \text{ is an} \atop (\mathbb{F}_t)_{t\in\{t_0,t_1,t_2\}}\text{-stopping time}}\right\}. \tag{95}$$

In Table 1 we show approximations for the mean and for the standard deviation of $\mathcal{P}$ and the average runtime in seconds needed for calculating one realisation of $\mathcal{P}$ for $d \in \{1, 5, 10, 50, 100, 500, 1000\}$. For each case the calculations of the results in Table 1 are based on 10 independent realisations of $\mathcal{P}$, which have been obtained from an implementation in PYTHON.

4.3.1.2 An American put-type example

In this subsection we test the algorithm in Framework 3.2 on the example of optimally stopping a standard Brownian motion under a put option inspired pay-off function.

Assume Framework 4.5, let $r = 0.06 = 6\%$, $\beta = 0.4 = 40\%$, $\chi = K = 40$, let $\mathbb{F} = (\mathbb{F}_t)_{t\in[0,T]}$ be the filtration generated by $W^{0,1}$, let $\mathfrak{F} = (\mathfrak{F}_t)_{t\in[0,T]}$ be the filtration generated by $W^{0,1,(1)}$, and assume for all $m, j \in \mathbb{N}$, $n \in \{0, 1,\ldots,N\}$, $s \in [0, T]$, $x = (x_1,\ldots,x_d) \in$

Dimension d	Mean of $\mathcal{P}$	Standard deviation of $\mathcal{P}$	Rel. L^1-approx. err.	Standard deviation of the rel. approx. err.	Runtime in sec. for one realisation of $\mathcal{P}$
1	5.311	0.002	0.0013	0.0004	78.6
5	5.310	0.003	0.0015	0.0005	91.3
10	5.309	0.003	0.0017	0.0005	104.6
50	5.306	0.003	0.0022	0.0006	215.7
100	5.305	0.004	0.0025	0.0006	245.1
500	5.298	0.003	0.0037	0.0005	1006.3
1000	5.294	0.003	0.0046	0.0006	2266.0

Table 2: Numerical simulations of the algorithm in Framework 3.2 for optimally stopping a standard Brownian motion in the case of the American put-type example in Subsection 4.3.1.2. In the approximative calculations of the relative approximation errors the exact number (97) has been replaced by the value 5.318, which has been obtained using the binomial tree method on M. Smirnov's website [85].

$\mathbb{R}^d$ that $T = 1$, $N = 50$, $M = 1500\,\mathbb{1}_{[1,50]}(d) + 1800\,\mathbb{1}_{(50,100]}(d) + 3000\,\mathbb{1}_{(100,\infty)}(d)$, $\mathcal{X}^{m-1,j}_{t_n} = W^{m-1,j}_{t_n}$, $J_0 = 4\,096\,000$, $J_m = 8192\,\mathbb{1}_{[1,50]}(d) + 4096\,\mathbb{1}_{(50,100]}(d) + 2048\,\mathbb{1}_{(100,\infty)}(d)$, $\varepsilon = 0.001$, $\gamma_m = 5\,[10^{-2}\,\mathbb{1}_{[1,M/3]}(m) + 10^{-3}\,\mathbb{1}_{(M/3,2M/3]}(m) + 10^{-4}\,\mathbb{1}_{(2M/3,\infty)}(m)]$, and

$$g(s,x) = e^{-rs} \max\Big\{K - \exp\Big(\big[r - \tfrac{1}{2}\beta^2\big]s + \tfrac{\beta}{\sqrt{d}}[x_1 + \ldots + x_d]\Big)\chi, 0\Big\}. \tag{96}$$

The random variable $\mathcal{P}$ provides approximations for the real number

$$\sup\Big\{\mathbb{E}\big[g(\tau, W^{0,1}_\tau)\big] : \substack{\tau\colon \Omega\to[0,T] \text{ is an} \\ \mathbb{F}\text{-stopping time}}\Big\}. \tag{97}$$

We show approximations for the mean of $\mathcal{P}$, for the standard deviation of $\mathcal{P}$, and for the relative L^1-approximation error associated to $\mathcal{P}$, the uncorrected sample standard deviation of the relative approximation error associated to $\mathcal{P}$, and the average runtime in seconds needed for calculating one realisation of $\mathcal{P}$ for $d \in \{1, 5, 10, 50, 100, 500, 1000\}$ in Table 2. For each case the calculations of the results in Table 2 are based on 10 independent realisations of $\mathcal{P}$, which have been obtained from an implementation in Python. Furthermore, in the approximative calculations of the relative approximation error associated to $\mathcal{P}$ the exact number (97) has been replaced, independently of the dimension d, by the real number

$$\sup\Big\{\mathbb{E}\Big[e^{-r\tau}\max\Big\{K - \exp\Big(\big[r - \tfrac{1}{2}\beta^2\big]\tau + \beta\, W^{0,1,(1)}_\tau\Big)\chi, 0\Big\}\Big] : \substack{\tau\colon \Omega\to[0,T] \text{ is an} \\ \mathfrak{F}\text{-stopping time}}\Big\} \tag{98}$$

(cf. Corollary 4.2), which, in turn, has been replaced by the value 5.318 (cf. Longstaff & Schwartz [73, Table 1 in Section 3]). This value has been calculated using the binomial tree method on M. Smirnov's website [85] with 20 000 nodes. Note that (98) corresponds to the price of an American put option on a single stock in the Black–Scholes model with initial stock price χ, interest rate r, volatility β, strike price K, and maturity T.

4.3.2 Geometric average-type options

4.3.2.1 An American geometric average put-type example

In this subsection we test the algorithm in Framework 3.2 on the example of pricing an American geometric average put-type option on up to 200 distinguishable stocks in the

Black–Scholes model.

Assume Framework 4.5, assume that $d \in \{40, 80, 120, \ldots\}$, let $\beta = (\beta_1, \ldots, \beta_d) \in \mathbb{R}^d$, $\rho, \tilde{\delta}, \tilde{\beta}, \delta_1, \delta_2, \ldots, \delta_d \in \mathbb{R}$, $r = 0.6$, $K = 95$, $\tilde{\xi} = 100$ satisfy for all $i \in \{1, \ldots, d\}$ that $\beta_i = \min\{0.04\,[(i-1) \bmod 40], 1.6 - 0.04\,[(i-1) \bmod 40]\}$, $\rho = \frac{1}{d}\|\beta\|^2_{\mathbb{R}^d} = \frac{1}{40}\sum_{i=1}^{40}(\beta_i)^2 = 0.2136$, $\delta_i = r - \frac{\rho}{d}\big(i - \frac{1}{2}\big) - \frac{1}{5\sqrt{d}}$, $\tilde{\delta} = r - \frac{1}{\sqrt{d}}\sum_{i=1}^{d}(r - \delta_i) + \frac{\sqrt{d}-1}{2d}\|\beta\|^2_{\mathbb{R}^d} = r - \frac{\rho}{2} - \frac{1}{5} = 0.2932$, and $\tilde{\beta} = \frac{1}{\sqrt{d}}\|\beta\|_{\mathbb{R}^d} = \sqrt{\rho}$, let $Y\colon [0,T] \times \Omega \to \mathbb{R}$ be an $\mathscr{F}$-adapted stochastic process with continuous sample paths which satisfies that for all $t \in [0,T]$ it holds $\mathbb{P}$-a.s. that

$$Y_t = \tilde{\xi} + (r - \tilde{\delta})\int_0^t Y_s\,ds + \tilde{\beta}\int_0^t Y_s\,dW_s^{0,1,(1)}, \tag{99}$$

let $\mathbb{F} = (\mathbb{F}_t)_{t\in[0,T]}$ be the filtration generated by X, let $\mathfrak{F} = (\mathfrak{F}_t)_{t\in[0,T]}$ be the filtration generated by Y, and assume for all $m, j \in \mathbb{N}$, $n \in \{0, 1, \ldots, N\}$, $i \in \{1, \ldots, d\}$, $s \in [0,T]$, $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$ that $T = 1$, $N = 100$, $M = 1800\,\mathbb{1}_{[1,120]}(d) + 3000\,\mathbb{1}_{(120,\infty)}(d)$, $J_0 = 4\,096\,000$, $J_m = 8192\,\mathbb{1}_{[1,120]}(d) + 4096\,\mathbb{1}_{(120,\infty)}(d)$, $\varepsilon = 10^{-8}$, $\gamma_m = 5\,[10^{-2}\,\mathbb{1}_{[1,M/3]}(m) + 10^{-3}\,\mathbb{1}_{(M/3,2M/3]}(m) + 10^{-4}\,\mathbb{1}_{(2M/3,\infty)}(m)]$, $\xi_i = (100)^{1/\sqrt{d}}$, $\mu(x) = ((r-\delta_1)\,x_1, \ldots, (r-\delta_d)\,x_d)$, $\sigma(x) = \operatorname{diag}(\beta_1 x_1, \ldots, \beta_d x_d)$, that

$$\mathcal{X}_{t_n}^{m-1,j,(i)} = \exp\!\big(\big[r - \delta_i - \tfrac{1}{2}(\beta_i)^2\big]t_n + \beta_i\,W_{t_n}^{m-1,j,(i)}\big)\,\xi_i, \tag{100}$$

and that

$$g(s,x) = e^{-rs}\max\!\left\{K - \left[\prod_{k=1}^{d}|x_k|^{1/\sqrt{d}}\right], 0\right\}. \tag{101}$$

The random variable $\mathcal{P}$ provides approximations for the price

$$\sup\!\left\{\mathbb{E}\big[g(\tau, X_\tau)\big] : \substack{\tau\colon \Omega\to[0,T] \text{ is an} \\ \mathbb{F}\text{-stopping time}}\right\}. \tag{102}$$

In Table 3 we show approximations for the mean of $\mathcal{P}$, for the standard deviation of $\mathcal{P}$, and for the relative L^1-approximation error associated to $\mathcal{P}$, the uncorrected sample standard deviation of the relative approximation error associated to $\mathcal{P}$, and the average runtime in seconds needed for calculating one realisation of $\mathcal{P}$ for $d \in \{40, 80, 120, 160, 200\}$. For each case the calculations of the results in Table 3 are based on 10 independent realisations of $\mathcal{P}$, which have been obtained from an implementation in Python. Furthermore, in the approximative calculations of the relative approximation error associated to $\mathcal{P}$ the exact value of the price (102) has been replaced, independently of the dimension d, by the real number

$$\sup\!\left\{\mathbb{E}\big[e^{-r\tau}\max\{K - Y_\tau, 0\}\big] : \substack{\tau\colon \Omega\to[0,T] \text{ is an} \\ \mathfrak{F}\text{-stopping time}}\right\}, \tag{103}$$

(cf. Proposition 4.3), which, in turn, has been replaced by the value 6.545. The latter has been calculated using the binomial tree method on M. Smirnov's website [85] with 20 000 nodes. Note that (103) corresponds to the price of an American put option on a single stock in the Black–Scholes model with initial stock price $\tilde{\xi}$, interest rate r, dividend rate $\tilde{\delta}$, volatility $\tilde{\beta}$, strike price K, and maturity T.

4.3.2.2 An American geometric average call-type example

In this subsection we test the algorithm in Framework 3.2 on the example of pricing an American geometric average call-type option on up to 100 correlated stocks in the Black–Scholes model. This example is taken from Sirignano & Spiliopoulos [82, Section 4].

Dimension d	Mean of $\mathcal{P}$	Standard deviation of $\mathcal{P}$	Rel. L^1-approx. err.	Standard deviation of the rel. approx. err.	Runtime in sec. for one realisation of $\mathcal{P}$
40	6.510	0.004	0.0053	0.0006	477.7
80	6.508	0.003	0.0056	0.0005	793.5
120	6.505	0.003	0.0061	0.0005	934.7
160	6.504	0.003	0.0062	0.0005	1203.9
200	6.504	0.004	0.0063	0.0005	1475.0

Table 3: Numerical simulations of the algorithm in Framework 3.2 for pricing the American geometric average put-type option from the example in Subsection 4.3.2.1. In the approximative calculations of the relative approximation errors the exact value of the price (102) has been replaced by the value 6.545, which has been obtained using the binomial tree method on M. Smirnov's website [85].

Assume Framework 4.5, let $r = 0\%$, $\delta = 0.02 = 2\%$, $\beta = 0.25 = 25\%$, $K = \tilde{\xi} = 1$, $Q = (Q_{i,j})_{(i,j)\in\{1,\dots,d\}^2}, \mathfrak{S} = (\varsigma_1, \dots, \varsigma_d) \in \mathbb{R}^{d\times d}$, $\tilde{\delta}, \tilde{\beta} \in \mathbb{R}$ satisfy for all $i \in \{1, \dots, d\}$ that $Q_{i,i} = 1$, $\forall\, j \in \{1, \dots, d\} \setminus \{i\}\colon Q_{i,j} = 0.75$, $\mathfrak{S}^* \mathfrak{S} = Q$, $\tilde{\delta} = \delta + \frac{1}{2}(\beta^2 - (\tilde{\beta})^2)$, and $\tilde{\beta} = \frac{\beta}{2d}\sqrt{d\,(3d+1)}$, let $Y\colon [0,T] \times \Omega \to \mathbb{R}$ be an $\mathscr{F}$-adapted stochastic process with continuous sample paths which satisfies that for all $t \in [0,T]$ it holds $\mathbb{P}$-a.s. that

$$Y_t = \tilde{\xi} + (r - \tilde{\delta}) \int_0^t Y_s\, ds + \tilde{\beta} \int_0^t Y_s\, dW_s^{0,1,(1)}, \tag{104}$$

let $\mathbb{F} = (\mathbb{F}_t)_{t\in[0,T]}$ be the filtration generated by X, let $\mathfrak{F} = (\mathfrak{F}_t)_{t\in[0,T]}$ be the filtration generated by Y, and assume for all $m, j \in \mathbb{N}$, $n \in \{0, 1, \dots, N\}$, $i \in \{1, \dots, d\}$, $s \in [0,T]$, $x = (x_1, \dots, x_d) \in \mathbb{R}^d$ that $T = 2$, $N = 50$, $M = 1600$, $J_0 = 4\,096\,000$, $J_m = 8192$, $\varepsilon = 10^{-8}$, $\gamma_m = 5\,[10^{-2}\,\mathbb{1}_{[1,400]}(m) + 10^{-3}\,\mathbb{1}_{(400,800]}(m) + 10^{-4}\,\mathbb{1}_{(800,\infty)}(m)]$, $\xi_i = 1$, $\mu(x) = (r - \delta)\,x$, $\sigma(x) = \beta\,\mathrm{diag}(x_1, \dots, x_d)\,\mathfrak{S}^*$, that

$$\mathcal{X}_{t_n}^{m-1,j,(i)} = \exp\Big(\big[r - \delta - \tfrac{1}{2}\beta^2\big]t_n + \beta\,\big\langle \varsigma_i, W_{t_n}^{m-1,j}\big\rangle_{\mathbb{R}^d}\Big)\,\xi_i, \tag{105}$$

and that

$$g(s,x) = e^{-rs} \max\Bigg\{\Bigg[\prod_{k=1}^d |x_k|^{1/d}\Bigg] - K, 0\Bigg\}. \tag{106}$$

The random variable $\mathcal{P}$ provides approximations for the price

$$\sup\Big\{\mathbb{E}\big[g(\tau, X_\tau)\big] : \substack{\tau\colon \Omega\to[0,T] \text{ is an} \\ \mathbb{F}\text{-stopping time}}\Big\}. \tag{107}$$

Table 4 shows approximations for the mean of $\mathcal{P}$, for the standard deviation of $\mathcal{P}$, for the real number

$$\sup\Big\{\mathbb{E}\big[e^{-r\tau} \max\{Y_\tau - K, 0\}\big] : \substack{\tau\colon \Omega\to[0,T] \text{ is an} \\ \mathfrak{F}\text{-stopping time}}\Big\}, \tag{108}$$

and for the relative L^1-approximation error associated to $\mathcal{P}$, the uncorrected sample standard deviation of the relative approximation error associated to $\mathcal{P}$, and the average runtime in seconds needed for calculating one realisation of $\mathcal{P}$ for $d \in \{3, 20, 100\}$. The approximative calculations of the mean of $\mathcal{P}$, of the standard deviation of $\mathcal{P}$, and of the relative L^1-approximation error associated to $\mathcal{P}$, the computations of the uncorrected

Dimension d	Mean of $\mathcal{P}$	Standard deviation of $\mathcal{P}$	Price (108)	Rel. L^1-approx. err.	Standard deviation of the rel. approx. err.	Runtime in sec. for one realisation of $\mathcal{P}$
3	0.10699	0.00007	0.10719	0.0019	0.0006	92.1
20	0.10007	0.00006	0.10033	0.0026	0.0006	146.9
100	0.09903	0.00006	0.09935	0.0032	0.0006	409.0

Table 4: Numerical simulations of the algorithm in Framework 3.2 for pricing the American geometric average call-type option from the example in Subsection 4.3.2.2. In the approximative calculations of the relative approximation errors the exact value of the price (107) has been replaced by the number (108), which has been approximatively calculated using the binomial tree method on M. Smirnov's website [85].

sample standard deviation of the relative approximation error associated to $\mathcal{P}$ as well as the computations of the average runtime for calculating one realisation of $\mathcal{P}$ in Table 4 each are based on 10 independent realisations of $\mathcal{P}$, which have been obtained from an implementation in PYTHON. Furthermore, in the approximative calculations of the relative approximation error associated to $\mathcal{P}$ the exact value of the price (107) has been replaced by the number (108) (cf. Proposition 4.3), which has been approximatively calculated using the binomial tree method on M. Smirnov's website [85] with 20 000 nodes. Note that (108) corresponds to the price of an American call option on a single stock in the Black–Scholes model with initial stock price $\tilde{\xi}$, interest rate r, dividend rate $\tilde{\delta}$, volatility $\tilde{\beta}$, strike price K, and maturity T.

4.3.2.3 Another American geometric average call-type example

In this subsection we test the algorithm in Framework 3.2 on the example of pricing an American geometric average call-type option on up to 400 distinguishable stocks in the Black–Scholes model.

Assume Framework 4.5, assume that $d \in \{40, 80, 120, \ldots\}$, let $\beta = (\beta_1, \ldots, \beta_d) \in \mathbb{R}^d$, $\alpha_1, \ldots, \alpha_d \in \mathbb{R}$, $r, \tilde{\beta} \in (0, \infty)$, $K = 95$, $\tilde{\xi} = 100$ satisfy for all $i \in \{1, \ldots, d\}$ that $\beta_i = \frac{0.4\, i}{d}$, $\alpha_i = \min\{0.01\,[(i-1) \bmod 40], 0.4 - 0.01\,[(i-1) \bmod 40]\}$, $r = \frac{1}{d}\sum_{i=1}^d \alpha_i - \frac{d-1}{2d^2}\|\beta\|_{\mathbb{R}^d}^2 = 0.1 - \frac{0.08}{d^2}(d-1)\big(\frac{d}{3} + \frac{1}{2} + \frac{1}{6d}\big)$, and $\tilde{\beta} = \frac{1}{d}\|\beta\|_{\mathbb{R}^d} = \frac{0.4}{d}\big(\frac{d}{3} + \frac{1}{2} + \frac{1}{6d}\big)^{1/2}$, let $Y\colon [0,T] \times \Omega \to \mathbb{R}$ be an $\mathscr{F}$-adapted stochastic process with continuous sample paths which satisfies that for all $t \in [0,T]$ it holds $\mathbb{P}$-a.s. that

$$Y_t = \tilde{\xi} + r \int_0^t Y_s \, ds + \tilde{\beta} \int_0^t Y_s \, dW_s^{0,1,(1)}, \tag{109}$$

let $\mathbb{F} = (\mathbb{F}_t)_{t \in [0,T]}$ be the filtration generated by X, let $\mathfrak{F} = (\mathfrak{F}_t)_{t \in [0,T]}$ be the filtration generated by Y, and assume for all $m, j \in \mathbb{N}$, $n \in \{0, 1, \ldots, N\}$, $i \in \{1, \ldots, d\}$, $s \in [0,T]$, $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$ that $T = 3$, $N = 50$, $M = 1500$, $J_0 = 4\,096\,000$, $J_m = 8192$, $\varepsilon = 10^{-8}$, $\gamma_m = 5\,[10^{-2}\,\mathbb{1}_{[1,M/3]}(m) + 10^{-3}\,\mathbb{1}_{(M/3,2M/3]}(m) + 10^{-4}\,\mathbb{1}_{(2M/3,\infty)}(m)]$, $\xi_i = 100$, $\mu(x) = (\alpha_1 x_1, \ldots, \alpha_d x_d)$, $\sigma(x) = \operatorname{diag}(\beta_1 x_1, \ldots, \beta_d x_d)$, that

$$\mathcal{X}_{t_n}^{m-1,j,(i)} = \exp\!\big(\big[\alpha_i - \tfrac{1}{2}(\beta_i)^2\big] t_n + \beta_i \, W_{t_n}^{m-1,j,(i)}\big)\, \xi_i, \tag{110}$$

and that

$$g(s,x) = e^{-rs} \max\!\left\{ \left[\prod_{k=1}^d |x_k|^{1/d} \right] - K, 0 \right\}. \tag{111}$$

Dimension d	Mean of $\mathcal{P}$	Standard deviation of $\mathcal{P}$	Price (113)	Rel. L^1-approx. err.	Standard deviation of the rel. approx. err.	Runtime in sec. for one realisation of $\mathcal{P}$
40	23.6877	0.0030	23.6883	0.00012	0.00004	196.3
80	23.7235	0.0020	23.7235	0.00006	0.00005	323.3
120	23.7360	0.0019	23.7357	0.00006	0.00005	442.7
160	23.7415	0.0007	23.7419	0.00003	0.00002	569.9
200	23.7451	0.0014	23.7456	0.00005	0.00004	692.9
400	23.7528	0.0009	23.7531	0.00004	0.00002	1434.7

Table 5: Numerical simulations of the algorithm in Framework 3.2 for pricing the American geometric average call-type option from the example in Subsection 4.3.2.3. In the approximative calculations of the relative approximation errors the exact value of the price (112) has been replaced by the number (113), which has been approximatively computed in MATLAB.

The random variable $\mathcal{P}$ provides approximations for the price

$$\sup\Big\{\mathbb{E}\big[g(\tau, X_\tau)\big] : {\scriptstyle \tau\colon \Omega\to[0,T] \text{ is an} \atop \scriptstyle \mathbb{F}\text{-stopping time}}\Big\}. \tag{112}$$

In Table 5 we show approximations for the mean of $\mathcal{P}$, for the standard deviation of $\mathcal{P}$, for the real number

$$\mathbb{E}\big[e^{-rT}\max\{Y_T - K, 0\}\big], \tag{113}$$

and for the relative L^1-approximation error associated to $\mathcal{P}$, the uncorrected sample standard deviation of the relative approximation error associated to $\mathcal{P}$, and the average runtime in seconds needed for calculating one realisation of $\mathcal{P}$ for $d \in \{40, 80, 120, 160, 200, 400\}$. The approximative calculations of the mean of $\mathcal{P}$, of the standard deviation of $\mathcal{P}$, and of the relative L^1-approximation error associated to $\mathcal{P}$, the computations of the uncorrected sample standard deviation of the relative approximation error associated to $\mathcal{P}$ as well as the computations of the average runtime for calculating one realisation of $\mathcal{P}$ in Table 5 each are based on 10 independent realisations of $\mathcal{P}$, which have been obtained from an implementation in PYTHON. Moreover, in the approximative calculations of the relative approximation error associated to $\mathcal{P}$ the exact value of the price (112) has been replaced by the real number

$$\sup\Big\{\mathbb{E}\big[e^{-r\tau}\max\{Y_\tau - K, 0\}\big] : {\scriptstyle \tau\colon \Omega\to[0,T] \text{ is an} \atop \scriptstyle \mathfrak{F}\text{-stopping time}}\Big\} \tag{114}$$

(cf. Proposition 4.3). It is well-known (cf., e.g., Shreve [81, Corollary 8.5.3]) that the number (114) is equal to the number (113), which has been approximatively computed in MATLAB R2017b using Lemma 4.4 above. Note that (114) corresponds to the price of an American call option on a single stock in the Black–Scholes model with initial stock price $\tilde{\xi}$, interest rate r, volatility $\tilde{\beta}$, strike price K, and maturity T, while (113) corresponds to the price of a European call option on a single stock in the Black–Scholes model with initial stock price $\tilde{\xi}$, interest rate r, volatility $\tilde{\beta}$, strike price K, and maturity T.

4.4 Examples without known one-dimensional representation

In Subsection 4.3 above numerical results for examples with a one-dimensional representation can be found. We test in this subsection several examples where such a representation is not known.

4.4.1 Max-call options

4.4.1.1 A Bermudan max-call standard benchmark example

In this subsection we test the algorithm in Framework 3.2 on the example of pricing a Bermudan max-call option on up to 500 stocks in the Black–Scholes model (see [9, Subsection 4.1]). In the case of up to five underlying stocks this example is a standard benchmark example in the literature (see, e.g., [25, Subsection 5.4], [73, Subsection 8.1], [3, Section 4], [49, Subsection 5.1], [77, Subsection 4.3], [39, Subsection 3.9], [18, Subsection 4.2], [23, Subsection 5.3], [14, Subsection 6.1], [52, Subsection 4.1], [79, Subsection 7.2], [16, Subsection 6.1], [71, Subsection 5.2.1]).

Assume Framework 4.5, let $r = 0.05 = 5\%$, $\delta = 0.1 = 10\%$, $\beta = 0.2 = 20\%$, $K = 100$, let $\mathbb{F} = (\mathbb{F}_t)_{t\in[0,T]}$ be the filtration generated by X, and assume for all $m, j \in \mathbb{N}$, $n \in \{0, 1, \ldots, N\}$, $i \in \{1, \ldots, d\}$, $s \in [0, T]$, $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$ that $T = 3$, $N = 9$, $M = 3000 + d$, $J_0 = 4\,096\,000$, $J_m = 8192$, $\varepsilon = 0.1$, $\gamma_m = 5\,[10^{-2}\,\mathbb{1}_{[1,500+d/5]}(m) + 10^{-3}\,\mathbb{1}_{(500+d/5,1500+3d/5]}(m) + 10^{-4}\,\mathbb{1}_{(1500+3d/5,\infty)}(m)]$, $\xi_i = \xi_1$, $\mu(x) = (r - \delta)\,x$, $\sigma(x) = \beta\,\mathrm{diag}(x_1, \ldots, x_d)$, that

$$\mathcal{X}_{t_n}^{m-1,j,(i)} = \exp\!\big(\big[r - \delta - \tfrac{1}{2}\beta^2\big]t_n + \beta\,W_{t_n}^{m-1,j,(i)}\big)\,\xi_1, \tag{115}$$

and that

$$g(s, x) = e^{-rs}\max\!\big\{\max\{x_1, \ldots, x_d\} - K, 0\big\}. \tag{116}$$

The random variable $\mathcal{P}$ provides approximations for the price

$$\sup\!\Big\{\mathbb{E}\big[g(\tau, X_\tau)\big] : {\scriptstyle \tau\colon \Omega\to\{t_0,t_1,\ldots,t_N\} \text{ is an} \atop (\mathbb{F}_t)_{t\in\{t_0,t_1,\ldots,t_N\}}\text{-stopping time}}\Big\}. \tag{117}$$

In Table 6 we show approximations for the mean and for the standard deviation of $\mathcal{P}$, binomial approximations as well as 95% confidence intervals for the price (117) according to Andersen & Broadie [3, Table 2 in Section 4] (where available), 95% confidence intervals for the price (117) according to Broadie & Cao [23, Table 3 in Subsection 5.3] (where available), and the average runtime in seconds needed for calculating one realisation of $\mathcal{P}$ for $(d, \xi_1) \in \{2, 3, 5\} \times \{90, 100, 110\}$. In addition, we list approximations for the mean and for the standard deviation of $\mathcal{P}$ and the average runtime in seconds needed for calculating one realisation of $\mathcal{P}$ for $(d, \xi_1) \in \{10, 20, 30, 50, 100, 200, 500\} \times \{90, 100, 110\}$ in Table 7. The approximative calculations of the mean and of the standard deviation of $\mathcal{P}$ as well as the computations of the average runtime for calculating one realisation of $\mathcal{P}$ in Tables 6–7 each are based on 10 independent realisations of $\mathcal{P}$, which have been obtained from an implementation in Python.

4.4.1.2 A high-dimensional Bermudan max-call benchmark example

In this subsection we test the algorithm in Framework 3.2 on the example of pricing the Bermudan max-call option from the example in Subsection 4.4.1.1 in a case with 5000 underlying stocks. All Python codes corresponding to this example were run in single precision (float32) on a NVIDIA Tesla P100 GPU with 1328 MHz core clock and 16 GB HBM2 memory with 1408 MHz clock rate.

Assume Framework 4.5, let $r = 0.05 = 5\%$, $\delta = 0.1 = 10\%$, $\beta = 0.2 = 20\%$, $K = 100$, let $\mathbb{F} = (\mathbb{F}_t)_{t\in[0,T]}$ be the filtration generated by X, and assume for all $m, j \in \mathbb{N}$, $n \in \{0, 1, \ldots, N\}$, $i \in \{1, \ldots, d\}$, $s \in [0, T]$, $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$ that $T = 3$, $d = 5000$,

Dimension d	Initial value ξ_1	Mean of $\mathcal{P}$	Standard deviation of $\mathcal{P}$	Binomial value in [3]	95% confidence interval in [3]	95% confidence interval in [23]	Runtime in sec. for one realisation of $\mathcal{P}$
2	90	8.072	0.005	8.075	[8.053, 8.082]	–	31.3
2	100	13.899	0.008	13.902	[13.892, 13.934]	–	31.7
2	110	21.344	0.006	21.345	[21.316, 21.359]	–	31.7
3	90	11.275	0.005	11.29	[11.265, 11.308]	–	32.5
3	100	18.687	0.006	18.69	[18.661, 18.728]	–	32.6
3	110	27.560	0.009	27.58	[27.512, 27.663]	–	32.5
5	90	16.628	0.010	–	[16.602, 16.655]	[16.620, 16.653]	33.3
5	100	26.144	0.008	–	[26.109, 26.292]	[26.115, 26.164]	32.9
5	110	36.763	0.011	–	[36.704, 36.832]	[36.710, 36.798]	33.3

Table 6: Numerical simulations of the algorithm in Framework 3.2 for pricing the Bermudan max-call option from the example in Subsection 4.4.1.1 for $d \in \{2, 3, 5\}$.

Dimension d	Initial value ξ_1	Mean of $\mathcal{P}$	Standard deviation of $\mathcal{P}$	Runtime in sec. for one realisation of $\mathcal{P}$
10	90	26.200	0.010	35.5
10	100	38.278	0.010	35.5
10	110	50.817	0.011	35.5
20	90	37.697	0.011	42.6
20	100	51.569	0.008	42.6
20	110	65.514	0.010	42.6
30	90	44.822	0.008	49.2
30	100	59.521	0.010	49.2
30	110	74.231	0.010	49.2
50	90	53.897	0.008	63.4
50	100	69.574	0.012	63.3
50	110	85.256	0.013	63.3
100	90	66.361	0.016	101.1
100	100	83.386	0.009	101.2
100	110	100.429	0.014	101.2
200	90	78.996	0.009	179.1
200	100	97.411	0.011	179.1
200	110	115.827	0.009	179.0
500	90	95.976	0.007	507.7
500	100	116.249	0.013	507.4
500	110	136.541	0.005	507.8

Table 7: Numerical simulations of the algorithm in Framework 3.2 for pricing the Bermudan max-call option from the example in Subsection 4.4.1.1 for $d \in \{10, 20, 30, 50, 100, 200, 500\}$.

$N = 9$, $J_0 = 2^{20}$, $J_m = 1024$, $\varepsilon = 10^{-8}$, $\gamma_m = 10^{-2}\,\mathbb{1}_{[1,2000]}(m) + 10^{-3}\,\mathbb{1}_{(2000,4000]}(m) + 10^{-4}\,\mathbb{1}_{(4000,\infty)}(m)$, $\xi_i = 100$, $\mu(x) = (r-\delta)\,x$, $\sigma(x) = \beta\,\mathrm{diag}(x_1,\ldots,x_d)$, that

$$\mathcal{X}^{m-1,j,(i)}_{t_n} = \exp\!\big(\big[r - \delta - \tfrac{1}{2}\beta^2\big]t_n + \beta\,W^{m-1,j,(i)}_{t_n}\big)\,\xi_i, \tag{118}$$

and that

$$g(s,x) = e^{-rs}\max\!\big\{\max\{x_1,\ldots,x_d\} - K, 0\big\}. \tag{119}$$

For sufficiently large $M \in \mathbb{N}$ the random variable $\mathcal{P}$ provides approximations for the price

$$\sup\!\left\{\mathbb{E}\big[g(\tau, X_\tau)\big] : \substack{\tau\colon \Omega\to\{t_0,t_1,\ldots,t_N\}\text{ is an}\\ (\mathbb{F}_t)_{t\in\{t_0,t_1,\ldots,t_N\}}\text{-stopping time}}\right\}. \tag{120}$$

In Table 8 we show a realisation of $\mathcal{P}$, a 95% confidence interval for the corresponding realisation of the random variable

$$\Omega \ni \mathbf{w} \mapsto \mathbb{E}\Big[g\Big(\tau^{1,\Theta_M(\mathbf{w}),\mathbb{S}_M(\mathbf{w})}, \mathcal{X}^{0,1}_{\tau^{1,\Theta_M(\mathbf{w}),\mathbb{S}_M(\mathbf{w})}}\Big)\Big] \in \mathbb{R}, \tag{121}$$

the corresponding realisation of the relative approximation error associated to $\mathcal{P}$, and the runtime in seconds need or calculating the realisation of $\mathcal{P}$ for $M \in \{0, 250, 500, \ldots, 2000\}$ $\cup\{6000\}$. In addition, Figure 1 depicts a realisation of the relative approximation error associated to $\mathcal{P}$ against $M \in \{0, 10, 20, \ldots, 2000\}$. For each case the 95% confidence interval for the realisation of the random variable (121) in Table 8 has been computed based on the corresponding realisation of $\mathcal{P}$, the corresponding sample standard deviation, and the 0.975 quantile of the standard normal distribution (cf., e.g., [9, Subsection 3.3]). Moreover, in the approximative calculations of the realisation of the relative approximation error associated to $\mathcal{P}$ in Table 8 and Figure 1 the exact value of the price (120) has been replaced by the value 165.430, which corresponds to a realisation of $\mathcal{P}$ with $M = 6000$ (cf. Table 8).

Number of steps M	Realisation of $\mathcal{P}$	95% confidence interval	Rel. approx. error	Runtime in sec.
0	106.711	[106.681, 106.741]	0.35495	157.3
250	132.261	[132.170, 132.353]	0.20050	271.7
500	156.038	[155.975, 156.101]	0.05677	386.0
750	103.764	[103.648, 103.879]	0.37276	500.4
1000	161.128	[161.065, 161.191]	0.02601	614.3
1250	162.756	[162.696, 162.816]	0.01616	728.8
1500	164.498	[164.444, 164.552]	0.00563	842.8
1750	163.858	[163.803, 163.913]	0.00950	957.3
2000	165.452	[165.400, 165.505]	0.00014	1071.9
6000	165.430	[165.378, 165.483]	0.00000	2899.5

Table 8: Numerical simulations of the algorithm in Framework 3.2 for pricing the Bermudan max-call option on 5000 stocks from the example in Subsection 4.4.1.2. In the approximative calculations of the relative approximation error the exact value of the price (120) has been replaced by the value 165.430, which corresponds to a realisation of $\mathcal{P}$ with $M = 6000$.

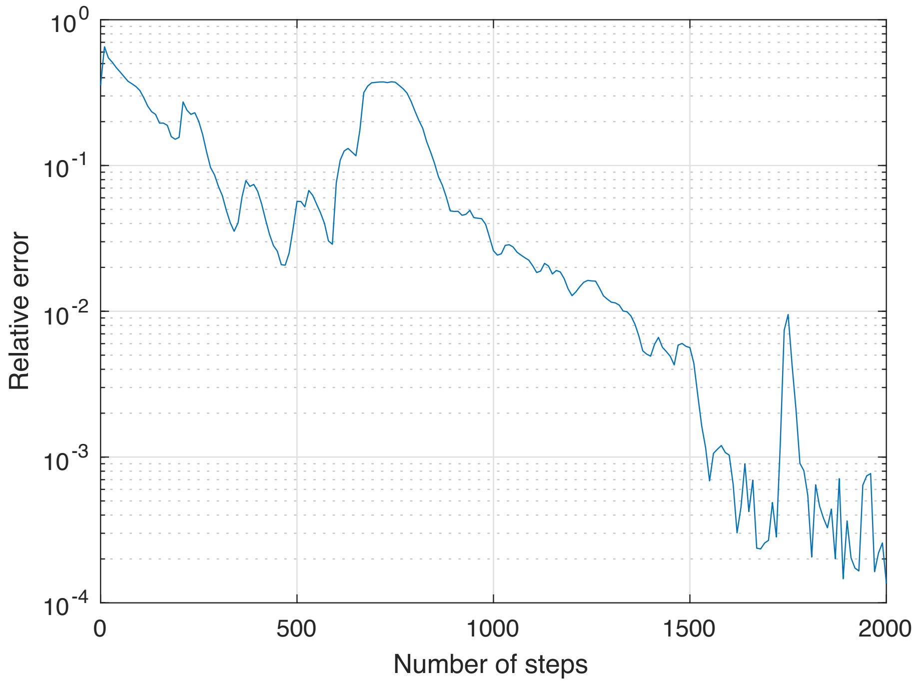

Figure 1: Plot of a realisation of the relative approximation error $\frac{|\mathcal{P}-165.430|}{165.430}$ against $M \in \{0, 10, 20, \ldots, 2000\}$ in the case of the Bermudan max-call option on 5000 stocks from the example in Subsection 4.4.1.2.

4.4.1.3 Another Bermudan max-call example

In this subsection we test the algorithm in Framework 3.2 on the example of pricing a Bermudan max-call option for different maturities and strike prices on up to 400 *correlated* stocks, that do not pay dividends, in the Black–Scholes model. This example is taken from Barraquand & Martineau [5, Section VII].

Assume Framework 4.5, let $\tau = {}^{30}/_{365}$, $r = 0.05\,\tau = 5\%\cdot\tau$, $\beta = 0.4\sqrt{\tau} = 40\%\cdot\sqrt{\tau}$, $K \in \{35, 40, 45\}$, $Q = (Q_{i,j})_{(i,j)\in\{1,\ldots,d\}^2}, \mathfrak{S} = (\varsigma_1, \ldots, \varsigma_d) \in \mathbb{R}^{d\times d}$ satisfy for all $i \in \{1, \ldots, d\}$ that $Q_{i,i} = 1$, $\forall\, j \in \{1, \ldots, d\} \setminus \{i\}\colon Q_{i,j} = 0.5$, and $\mathfrak{S}^*\,\mathfrak{S} = Q$, let $\mathbb{F} = (\mathbb{F}_t)_{t\in[0,T]}$ be the filtration generated by X, and assume for all $m, j \in \mathbb{N}$, $n \in \{0, 1, \ldots, N\}$, $i \in \{1, \ldots, d\}$, $s \in [0, T]$, $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$ that $N = 10$, $M = 1600$, $J_0 = 4\,096\,000$, $J_m = 8192$, $\varepsilon = 0.001$, $\gamma_m = 5\,[10^{-2}\,\mathbb{1}_{[1,400]}(m) + 10^{-3}\,\mathbb{1}_{(400,800]}(m) + 10^{-4}\,\mathbb{1}_{(800,\infty)}(m)]$, $\xi_i = 40$, $\mu(x) = r\,x$, $\sigma(x) = \beta\,\mathrm{diag}(x_1, \ldots, x_d)\,\mathfrak{S}^*$, that

$$\mathcal{X}^{m-1,j,(i)}_{t_n} = \exp\!\Big(\big[r - \tfrac{1}{2}\beta^2\big]t_n + \beta\,\big\langle \varsigma_i, W^{m-1,j}_{t_n}\big\rangle_{\mathbb{R}^d}\Big)\,\xi_i, \tag{122}$$

and that

$$g(s, x) = e^{-rs}\max\big\{\max\{x_1, \ldots, x_d\} - K, 0\big\}. \tag{123}$$

The random variable $\mathcal{P}$ provides approximations for the price

$$\sup\Big\{\mathbb{E}\big[g(\tau, X_\tau)\big] : {\scriptstyle \tau\colon \Omega\to\{t_0,t_1,\ldots,t_N\} \text{ is an} \atop (\mathbb{F}_t)_{t\in\{t_0,t_1,\ldots,t_N\}}\text{-stopping time}}\Big\}. \tag{124}$$

In Table 9 we show approximations for the mean and for the standard deviation of $\mathcal{P}$, Monte Carlo approximations for the to (124) corresponding European max-call option price

$$\mathbb{E}\big[g(T, X_T)\big], \tag{125}$$

approximations for the price (124) according to [5, Table 4 in Section VII] (where available), and the average runtime in seconds needed for calculating one realisation of $\mathcal{P}$ for

$$(d, T, K) \in \left\{ \begin{array}{lll} (10,1,35), & (10,1,40), & (10,1,45), \\ (10,4,35), & (10,4,40), & (10,4,45), \\ (10,7,35), & (10,7,40), & (10,7,45), \\ (400,12,35), & (400,12,40), & (400,12,45) \end{array} \right\}. \tag{126}$$

The approximative calculations of the mean and of the standard deviation of $\mathcal{P}$ as well as the computations of the average runtime for calculating one realisation of $\mathcal{P}$ in Table 9 each are based on 10 independent realisations of $\mathcal{P}$, which have been obtained from an implementation in PYTHON. Furthermore, the Monte Carlo approximations for the European price (125) in Table 9 each are calculated in double precision (float64) and are based on $2 \cdot 10^{10}$ independent realisations of the random variable $\Omega \ni \omega \mapsto g(T, X_T(\omega)) \in \mathbb{R}$.

Dimension d	Maturity T	Strike price K	Mean of $\mathcal{P}$	Standard deviation of $\mathcal{P}$	European price (125)	Price in [5]	Runtime in sec. for one realisation of $\mathcal{P}$
10	1	35	10.365	0.002	10.365	10.36	24.4
10	1	40	5.540	0.002	5.540	5.54	24.3
10	1	45	1.897	0.001	1.896	1.90	24.3
10	4	35	16.519	0.003	16.520	16.53	24.3
10	4	40	11.869	0.007	11.870	11.87	24.3
10	4	45	7.801	0.003	7.804	7.81	24.3
10	7	35	20.913	0.007	20.916	20.92	24.3
10	7	40	16.374	0.008	16.374	16.38	24.3
10	7	45	12.271	0.006	12.277	12.28	24.3
400	12	35	55.712	0.009	55.714	–	247.7
400	12	40	50.969	0.020	50.964	–	247.8
400	12	45	46.233	0.010	46.234	–	247.6

Table 9: Numerical simulations of the algorithm in Framework 3.2 for pricing the Bermudan max-call option from the example in Subsection 4.4.1.3.

4.4.2 An American strangle spread basket option

In this subsection we test the algorithm in Framework 3.2 on the example of pricing an American strangle spread basket option on five correlated stocks in the Black–Scholes model. This example is taken from Kohler, Krzyżak, & Todorovic [64, Section 4] (see also Kohler [61, Section 3] and Kohler, Krzyżak, & Walk [65, Section 4]).

Assume Framework 4.5, let $r = 0.05 = 5\%$, $K_1 = 75$, $K_2 = 90$, $K_3 = 110$, $K_4 = 125$, let $\mathfrak{S} = (\varsigma_1, \ldots, \varsigma_5) \in \mathbb{R}^{5\times 5}$ be given by

$$\mathfrak{S} = \begin{pmatrix} 0.3024 & 0.1354 & 0.0722 & 0.1367 & 0.1641 \\ 0.1354 & 0.2270 & 0.0613 & 0.1264 & 0.1610 \\ 0.0722 & 0.0613 & 0.0717 & 0.0884 & 0.0699 \\ 0.1367 & 0.1264 & 0.0884 & 0.2937 & 0.1394 \\ 0.1641 & 0.1610 & 0.0699 & 0.1394 & 0.2535 \end{pmatrix}, \tag{127}$$

let $\mathbb{F} = (\mathbb{F}_t)_{t\in[0,T]}$ be the filtration generated by X, and assume for all $m, j \in \mathbb{N}$, $n \in \{0, 1, \ldots, N\}$, $i \in \{1, \ldots, d\}$, $s \in [0, T]$, $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$ that $T = 1$, $d = 5$, $N = 48$, $M = 750$, $J_0 = 4\,096\,000$, $J_m = 8192$, $\varepsilon = 10^{-8}$, $\gamma_m = 5\,[10^{-2}\,\mathbb{1}_{[1,250]}(m) + 10^{-3}\,\mathbb{1}_{(250,500]}(m) + 10^{-4}\,\mathbb{1}_{(500,\infty)}(m)]$, $\xi_i = 100$, $\mu(x) = r\,x$, $\sigma(x) = \operatorname{diag}(x_1, \ldots, x_d)\,\mathfrak{S}^*$, that

$$\mathcal{X}_{t_n}^{m-1,j,(i)} = \exp\Big(\big[r - \tfrac{1}{2}\|\varsigma_i\|_{\mathbb{R}^d}^2\big]t_n + \big\langle \varsigma_i, W_{t_n}^{m-1,j}\big\rangle_{\mathbb{R}^d}\Big)\,\xi_i, \tag{128}$$

and that

$$\begin{aligned} g(s,x) = {}& -e^{-rs}\max\left\{K_1 - \frac{1}{d}\left[\sum_{k=1}^{d} x_k\right], 0\right\} + e^{-rs}\max\left\{K_2 - \frac{1}{d}\left[\sum_{k=1}^{d} x_k\right], 0\right\} \\ & + e^{-rs}\max\left\{\frac{1}{d}\left[\sum_{k=1}^{d} x_k\right] - K_3, 0\right\} - e^{-rs}\max\left\{\frac{1}{d}\left[\sum_{k=1}^{d} x_k\right] - K_4, 0\right\}. \end{aligned} \tag{129}$$

The random variable $\mathcal{P}$ provides approximations for the price

$$\sup\left\{\mathbb{E}\big[g(\tau, X_\tau)\big] : \substack{\tau\colon \Omega\to[0,T] \text{ is an} \\ \mathbb{F}\text{-stopping time}}\right\}. \tag{130}$$

Table 10 shows approximations for the mean and for the standard deviation of $\mathcal{P}$, a lower bound for the price (130) according to Kohler, Krzyżak, & Todorovic [64, Figure 4.5 in Section 4] (cf. also Kohler [61, Figure 2 in Section 3] and, for an upper bound for the price (130), Kohler, Krzyżak, & Walk [65, Figure 4.2 in Section 4]), and the average runtime in seconds needed for calculating one realisation of $\mathcal{P}$. Since the mean of $\mathcal{P}$ is also a lower bound for the price (130), a higher value indicates a better approximation for the price (130) (cf. Table 10). The approximative calculations of the mean and of the standard deviation of $\mathcal{P}$ as well as the computation of the average runtime for calculating one realisation of $\mathcal{P}$ in Table 10 each are based on 10 independent realisations of $\mathcal{P}$, which have been obtained from an implementation in Python.

Mean of $\mathcal{P}$	Standard deviation of $\mathcal{P}$	Lower bound in [64]	Runtime in sec. for one realisation of $\mathcal{P}$
11.794	0.004	11.75	46.4

Table 10: Numerical simulations of the algorithm in Framework 3.2 for pricing the American strangle spread basket option from the example in Subsection 4.4.2.

4.4.3 An American put basket option in Dupire's local volatility model

In this subsection we test the algorithm in Framework 3.2 on the example of pricing an American put basket option on five stocks in Dupire's local volatility model. This example is taken from Labart & Lelong [68, Subsection 6.3] with the modification that we also consider the case where the underlying stocks do not pay any dividends.

Assume Framework 4.5, let $L = 10$, $r = 0.05 = 5\%$, $\delta \in \{0\%, 10\%\}$, $K = 100$, assume for all $i \in \{1, \ldots, d\}$, $x \in \mathbb{R}^d$ that $\xi_i = 100$ and $\mu(x) = (r - \delta)\,x$, let $\beta\colon [0, T] \times \mathbb{R} \to \mathbb{R}$ and $\boldsymbol{\sigma}\colon [0, T] \times \mathbb{R}^d \to \mathbb{R}^{d\times d}$ be the functions which satisfy for all $t \in [0, T]$, $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$ that

$$\beta(t, x_1) = 0.6\,e^{-0.05\sqrt{t}}\big(1.2 - e^{-0.1\,t - 0.001(e^{rt}x_1 - \xi_1)^2}\big)\,x_1 \tag{131}$$

and $\boldsymbol{\sigma}(t,x) = \operatorname{diag}(\beta(t,x_1),\beta(t,x_2),\ldots,\beta(t,x_d))$, let $S = (S^{(1)},\ldots,S^{(d)})\colon [0,T]\times\Omega \to \mathbb{R}^d$ be an $\mathscr{F}$-adapted stochastic process with continuous sample paths which satisfies that for all $t \in [0,T]$ it holds $\mathbb{P}$-a.s. that

$$S_t = \xi + \int_0^t \mu(S_s)\, ds + \int_0^t \boldsymbol{\sigma}(s,S_s)\, dW_s^{0,1}, \tag{132}$$

let $\mathcal{Y}^{m,j} = (\mathcal{Y}^{m,j,(1)},\ldots,\mathcal{Y}^{m,j,(d)})\colon [0,T]\times\Omega \to \mathbb{R}^d$, $j \in \mathbb{N}$, $m \in \mathbb{N}_0$, be the stochastic processes which satisfy for all $m \in \mathbb{N}_0$, $j \in \mathbb{N}$, $\ell \in \{0,1,\ldots,L-1\}$, $t \in \big[\frac{\ell T}{L}, \frac{(\ell+1)T}{L}\big]$, $i \in \{1,\ldots,d\}$ that $\mathcal{Y}_0^{m,j,(i)} = \log(\xi_i)$ and

$$\begin{aligned} \mathcal{Y}_t^{m,j,(i)} &= \mathcal{Y}_{\ell T/L}^{m,j(i)} + \big(t - \tfrac{\ell T}{L}\big)\Big(r - \delta - \tfrac{1}{2}\big[\beta\big(\tfrac{\ell T}{L}, \exp\big(\mathcal{Y}_{\ell T/L}^{m,j,(i)}\big)\big)\big]^2\Big) \\ &\quad + \big(\tfrac{tL}{T} - \ell\big)\beta\big(\tfrac{\ell T}{L}, \exp\big(\mathcal{Y}_{\ell T/L}^{m,j,(i)}\big)\big)\big(W_{(\ell+1)T/L}^{m,j,(i)} - W_{\ell T/L}^{m,j,(i)}\big), \end{aligned} \tag{133}$$

let $\mathbb{F} = (\mathbb{F}_t)_{t\in[0,T]}$ be the filtration generated by S, let $\mathfrak{F} = (\mathfrak{F}_t)_{t\in[0,T]}$ be the filtration generated by $\mathcal{Y}^{0,1}$, and assume for all $m,j \in \mathbb{N}$, $n \in \{0,1,\ldots,N\}$, $s \in [0,T]$, $x = (x_1,\ldots,x_d) \in \mathbb{R}^d$ that $T = 1$, $d = 5$, $M = 1200$, $\mathcal{X}_{t_n}^{m-1,j} = \mathcal{Y}_{t_n}^{m-1,j}$, $J_0 = 4\,096\,000$, $J_m = 8192$, $\varepsilon = 10^{-8}$, $\gamma_m = 5\,[10^{-2}\,\mathbb{1}_{[1,400]}(m) + 10^{-3}\,\mathbb{1}_{(400,800]}(m) + 10^{-4}\,\mathbb{1}_{(800,\infty)}(m)]$, and

$$g(s,x) = e^{-rs}\max\!\left\{K - \frac{1}{d}\left[\sum_{i=1}^d \exp(x_i)\right], 0\right\}. \tag{134}$$

The random variable $\mathcal{P}$ provides approximations for the price

$$\sup\!\left\{\mathbb{E}\big[g(\tau,\mathcal{Y}_\tau^{0,1})\big] : \substack{\tau\colon\Omega\to[0,T] \text{ is an} \\ \mathfrak{F}\text{-stopping time}}\right\}, \tag{135}$$

which, in turn, is an approximation for the price

$$\sup\!\left\{\mathbb{E}\!\left[e^{-r\tau}\max\!\left\{K - \frac{1}{d}\left[\sum_{i=1}^d S_\tau^{(i)}\right], 0\right\}\right] : \substack{\tau\colon\Omega\to[0,T] \text{ is an} \\ \mathbb{F}\text{-stopping time}}\right\}. \tag{136}$$

In Table 11 we show approximations for the mean and for the standard deviation of $\mathcal{P}$ and the average runtime in seconds needed for calculating one realisation of $\mathcal{P}$ for $(\delta,N) \in \{0\%,10\%\} \times \{5,10,50,100\}$. For each case the calculations of the results in Table 11 are based on 10 independent realisations of $\mathcal{P}$, which have been obtained from an implementation in Python. According to [68, Subsection 6.3], the value 6.30 is an approximation for a to (135) corresponding price in the case $\delta = 10\%$. Furthermore, the to (135) corresponding European put basket option price $\mathbb{E}\big[g(T,\mathcal{Y}_T^{0,1})\big]$ has been approximatively calculated using a Monte Carlo approximation based on 10^{10} realisations of the random variable $\Omega \ni \omega \mapsto g(T,\mathcal{Y}_T^{0,1}(\omega)) \in \mathbb{R}$, which resulted in the value 1.741 in the case $\delta = 0\%$ and in the value 6.304 in the case $\delta = 10\%$.

4.4.4 A path-dependent financial derivative

In this subsection we test the algorithm in Framework 3.2 on the example of pricing a specific path-dependent financial derivative contingent on prices of a single underlying stock in the Black–Scholes model, which is formulated as a 100-dimensional optimal stopping problem. This example is taken from Tsitsiklis & Van Roy [88, Section IV] with the modification that we consider a finite instead of an infinite time horizon.

Dividend rate δ	Time discretisation parameter N	Mean of $\mathcal{P}$	Standard deviation of $\mathcal{P}$	Runtime in sec. for one realisation of $\mathcal{P}$
0%	5	1.935	0.001	12.7
0%	10	1.978	0.001	21.3
0%	50	1.975	0.002	69.2
0%	100	1.971	0.002	137.5
10%	5	6.301	0.004	12.8
10%	10	6.303	0.003	21.4
10%	50	6.304	0.003	69.2
10%	100	6.303	0.004	137.5

Table 11: Numerical simulations of the algorithm in Framework 3.2 for pricing the American put basket option in Dupire's local volatility model from the example in Subsection 4.4.3. The corresponding European put basket option price is approximately equal to the value 1.741 in the case $\delta = 0\%$ and to the value 6.304 in the case $\delta = 10\%$.

Assume Framework 4.5, let $r = 0.0004 = 0.04\%$, $\beta = 0.02 = 2\%$, let $\mathcal{W}^{m,j} \colon [0,\infty) \times \Omega \to \mathbb{R}$, $j \in \mathbb{N}$, $m \in \mathbb{N}_0$, be independent $\mathbb{P}$-standard Brownian motions with continuous sample paths, let $S^{m,j} \colon [-100,\infty) \times \Omega \to \mathbb{R}$, $j \in \mathbb{N}$, $m \in \mathbb{N}_0$, and $\mathcal{Y}^{m,j} \colon \mathbb{N}_0 \times \Omega \to \mathbb{R}^{100}$, $j \in \mathbb{N}$, $m \in \mathbb{N}_0$, be the stochastic processes which satisfy for all $m,n \in \mathbb{N}_0$, $j \in \mathbb{N}$, $t \in [-100,\infty)$ that $S^{m,j}_t = \exp\big(\big[r - \frac{1}{2}\beta^2\big](t+100) + \beta\, \mathcal{W}^{m,j}_{t+100}\big)\, \xi_1$ and

$$\begin{aligned} \mathcal{Y}^{m,j}_n &= \Big(\tfrac{S^{m,j}_{n-99}}{S^{m,j}_{n-100}}, \tfrac{S^{m,j}_{n-98}}{S^{m,j}_{n-100}}, \ldots, \tfrac{S^{m,j}_{n}}{S^{m,j}_{n-100}}\Big) \\ &= \big(\exp\big(\big[r - \tfrac{1}{2}\beta^2\big] + \beta\,[\mathcal{W}^{m,j}_{n+1} - \mathcal{W}^{m,j}_n]\big), \exp\big(2\,\big[r - \tfrac{1}{2}\beta^2\big] + \beta\,[\mathcal{W}^{m,j}_{n+2} - \mathcal{W}^{m,j}_n]\big), \\ &\qquad \ldots, \exp\big(100\,\big[r - \tfrac{1}{2}\beta^2\big] + \beta\,[\mathcal{W}^{m,j}_{n+100} - \mathcal{W}^{m,j}_n]\big)\big), \end{aligned} \tag{137}$$

let $\mathbb{F} = (\mathbb{F}_n)_{n\in\mathbb{N}_0}$ be the filtration generated by $\mathcal{Y}^{0,1}$, and assume for all $m,j \in \mathbb{N}$, $n \in \{0,1,\ldots,N\}$, $s \in [0,T]$, $x = (x_1,\ldots,x_d) \in \mathbb{R}^d$ that $T \in \mathbb{N}$, $d = 100$, $N = T$, $M = 1200\,\mathbb{1}_{[1,150]}(T) + 1500\,\mathbb{1}_{(150,250]}(T) + 3000\,\mathbb{1}_{(250,\infty)}(T)$, $\mathcal{X}^{m-1,j}_n = \mathcal{Y}^{m-1,j}_n$, $J_0 = 4\,096\,000$, $J_m = 8192\,\mathbb{1}_{[1,150]}(T) + 4096\,\mathbb{1}_{(150,250]}(T) + 512\,\mathbb{1}_{(250,\infty)}(T)$, $\varepsilon = 10^{-8}$, $\gamma_m = 5\,[10^{-2}\,\mathbb{1}_{[1,M/3]}(m) + 10^{-3}\,\mathbb{1}_{(M/3,2M/3]}(m) + 10^{-4}\,\mathbb{1}_{(2M/3,\infty)}(m)]$, and $g(s,x) = e^{-rs}\,x_{100}$. The random variable $\mathcal{P}$ provides approximations for the real number

$$\sup\Big\{\mathbb{E}\Big[e^{-r\tau}\tfrac{S^{0,1}_\tau}{S^{0,1}_{\tau-100}}\Big] \colon \substack{\tau\colon \Omega\to\{0,1,\ldots,T\} \text{ is an} \\ (\mathbb{F}_n)_{n\in\{0,1,\ldots,T\}}\text{-stopping time}}\Big\}. \tag{138}$$

In Table 12 we show approximations for the mean and for the standard deviation of $\mathcal{P}$ and the average runtime in seconds needed for calculating one realisation of $\mathcal{P}$ for $T \in \{100, 150, 200, 250, 1000\}$. For each case the calculations of the results in Table 12 are based on 10 independent realisations of $\mathcal{P}$, which have been obtained from an implementation in Python. Note that in this example time is measured in days and that, roughly speaking, (138) corresponds to the price of a financial derivative which, if the holder decides to exercise, pays off the amount given by the ratio between the current underlying stock price and the underlying stock price 100 days before (cf. [88, Section IV] for more details). According to [88, Subsection IV.D], the value 1.282 is a lower bound for the price

$$\sup\Big\{\mathbb{E}\Big[e^{-r\tau}\tfrac{S^{0,1}_\tau}{S^{0,1}_{\tau-100}}\Big] \colon \substack{\tau\colon \Omega\to\mathbb{N}_0 \text{ is an} \\ \mathbb{F}\text{-stopping time}}\Big\}, \tag{139}$$

which corresponds to the price (138) in the case of an infinite time horizon. Since the mean of $\mathcal{P}$ is a lower bound for the price (138), which, in turn, is a lower bound for

Time horizon T	Mean of $\mathcal{P}$	Standard deviation of $\mathcal{P}$	Runtime in sec. for one realisation of $\mathcal{P}$
100	1.2721	0.0001	475.5
150	1.2821	0.0001	724.8
200	1.2894	0.0002	653.1
250	1.2959	0.0001	838.7
1000	1.3002	0.0006	1680.1

Table 12: Numerical simulations of the algorithm in Framework 3.2 for pricing the path-dependent financial derivative from the example in Subsection 4.4.4. According to [88, Subsection IV.D], the value 1.282 is a lower bound for the price (139).

the price (139), a higher value indicates a better approximation for the price (139). In addition, observe that the price (138) is non-decreasing in T. While in our numerical simulations the approximate value of the mean of $\mathcal{P}$ is less or equal than 1.282 for comparatively small time horizons, i.e., for $T \leq 150$, it is already higher for slightly larger time horizons, i.e., for $T \geq 200$ (see Table 12).

Acknowledgements

This project has been partially supported through the ETH Research Grant ETH-47 15-2 "Mild stochastic calculus and numerical approximations for nonlinear stochastic evolution equations with Lévy noise", by the project "Construction of New Smoothness Spaces on Domains" (project number I 3403) funded by the Austrian Science Fund (FWF), and by the project "Deep artificial neural network approximations for stochastic partial differential equations: Algorithms and convergence proofs" (project number 184220) funded by the Swiss National Science Foundation (SNSF).

References

[1] AitSahlia, F., and Carr, P. American options: a comparison of numerical methods. In *Numerical methods in finance*, vol. 13 of *Publ. Newton Inst.* Cambridge Univ. Press, Cambridge, 1997, pp. 67–87.

[2] Andersen, L. A simple approach to the pricing of bermudan swaptions in the multi-factor libor market model. *Journal of Computational Finance 3*, 2 (2000), 5–32.

[3] Andersen, L., and Broadie, M. Primal-dual simulation algorithm for pricing multidimensional american options. *Management Science 50*, 9 (2004), 1222–1234.

[4] Bally, V., and Pagès, G. Error analysis of the optimal quantization algorithm for obstacle problems. *Stochastic Process. Appl. 106*, 1 (2003), 1–40.

[5] Barraquand, J., and Martineau, D. Numerical valuation of high dimensional multivariate american securities. *The Journal of Financial and Quantitative Analysis 30*, 3 (1995), 383–405.

[6] Barron, A. R. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Trans. Inform. Theory 39*, 3 (1993), 930–945.

[7] Bayer, C., Häppölä, J., and Tempone, R. Implied stopping rules for American basket options from Markovian projection. *Quant. Finance 19*, 3 (2019), 371–390.

[8] Bayer, C., Tempone, R., and Wolfers, S. Pricing American Options by Exercise Rate Optimization. *arXiv:1809.07300* (2018), 15 pages.

[9] Becker, S., Cheridito, P., and Jentzen, A. Deep optimal stopping. *arXiv:1804.05394* (2018), 24 pages.

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

[11] Belomestny, D. On the rates of convergence of simulation-based optimization algorithms for optimal stopping problems. *Ann. Appl. Probab. 21*, 1 (2011), 215–239.

[12] Belomestny, D. Pricing Bermudan options by nonparametric regression: optimal rates of convergence for lower estimates. *Finance Stoch. 15*, 4 (2011), 655–683.

[13] Belomestny, D. Solving optimal stopping problems via empirical dual optimization. *Ann. Appl. Probab. 23*, 5 (2013), 1988–2019.

[14] Belomestny, D., Bender, C., and Schoenmakers, J. True upper bounds for Bermudan products via non-nested Monte Carlo. *Math. Finance 19*, 1 (2009), 53–71.

[15] Belomestny, D., Dickmann, F., and Nagapetyan, T. Pricing Bermudan options via multilevel approximation methods. *SIAM J. Financial Math. 6*, 1 (2015), 448–466.

[16] Belomestny, D., Ladkau, M., and Schoenmakers, J. Multilevel simulation based policy iteration for optimal stopping—convergence and complexity. *SIAM/ASA J. Uncertain. Quantif. 3*, 1 (2015), 460–483.

[17] Belomestny, D., Schoenmakers, J., and Dickmann, F. Multilevel dual approach for pricing American style derivatives. *Finance Stoch. 17*, 4 (2013), 717–742.

[18] Bender, C., Kolodko, A., and Schoenmakers, J. Policy iteration for American options: overview. *Monte Carlo Methods Appl. 12*, 5-6 (2006), 347–362.

[19] Bender, C., Kolodko, A., and Schoenmakers, J. Enhanced policy iteration for American options via scenario selection. *Quant. Finance 8*, 2 (2008), 135–146.

[20] Bender, C., Schweizer, N., and Zhuo, J. A primal-dual algorithm for BSDES. *Math. Finance 27*, 3 (2017), 866–901.

[21] Berridge, S. J., and Schumacher, J. M. An irregular grid approach for pricing high-dimensional American options. *J. Comput. Appl. Math. 222*, 1 (2008), 94–111.

[22] Bouchard, B., and Touzi, N. Discrete-time approximation and Monte-Carlo simulation of backward stochastic differential equations. *Stochastic Process. Appl. 111*, 2 (2004), 175–206.

[23] Broadie, M., and Cao, M. Improved lower and upper bound algorithms for pricing American options by simulation. *Quant. Finance 8*, 8 (2008), 845–861.

[24] Broadie, M., and Glasserman, P. Pricing American-style securities using simulation. *J. Econom. Dynam. Control 21*, 8-9 (1997), 1323–1352. Computational financial modelling.

[25] Broadie, M., and Glasserman, P. A stochastic mesh method for pricing high-dimensional American options. *J. Comp. Fin. 7*, 4 (2004), 35–72.

[26] Broadie, M., Glasserman, P., and Ha, Z. Pricing American options by simulation using a stochastic mesh with optimized weights. In *Probabilistic constrained optimization*, vol. 49 of *Nonconvex Optim. Appl.* Kluwer Acad. Publ., Dordrecht, 2000, pp. 26–44.

[27] Carriere, J. F. Valuation of the early-exercise price for options using simulations and nonparametric regression. *Insurance Math. Econom. 19*, 1 (1996), 19–30.

[28] Chen, N., and Glasserman, P. Additive and multiplicative duals for American option pricing. *Finance Stoch. 11*, 2 (2007), 153–179.

[29] Christensen, S. A method for pricing American options using semi-infinite linear programming. *Math. Finance 24*, 1 (2014), 156–172.

[30] Company, R., Egorova, V., Jódar, L., and Soleymani, F. Computing stable numerical solutions for multidimensional American option pricing problems: a semi-discretization approach. *arXiv:1701.08545* (2017), 16 pages.

[31] Cybenko, G. Approximation by superpositions of a sigmoidal function. *Math. Control Signals Systems 2*, 4 (1989), 303–314.

[32] Da Prato, G., and Zabczyk, J. *Stochastic equations in infinite dimensions*, vol. 44 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, 1992.

[33] Davis, M. H. A., and Karatzas, I. A deterministic approach to optimal stopping. In *Probability, statistics and optimisation*, Wiley Ser. Probab. Math. Statist. Probab. Math. Statist. Wiley, Chichester, 1994, pp. 455–466.

[34] Desai, V. V., Farias, V. F., and Moallemi, C. C. Pathwise optimization for optimal stopping problems. *Management Science 58*, 12 (2012), 2292–2308.

[35] E, W., Han, J., and Jentzen, A. Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. *Commun. Math. Stat. 5*, 4 (2017), 349–380.

[36] Egloff, D. Monte Carlo algorithms for optimal stopping and statistical learning. *Ann. Appl. Probab. 15*, 2 (2005), 1396–1432.

[37] Egloff, D., Kohler, M., and Todorovic, N. A dynamic look-ahead Monte Carlo algorithm for pricing Bermudan options. *Ann. Appl. Probab. 17*, 4 (2007), 1138–1171.

[38] Ferguson, T. S. Optimal Stopping and Applications: Chapter 1. Stopping Rule Problems. Mathematics Department, UCLA, available online at https://www.math.ucla.edu/ tom/Stopping/sr1.pdf, last accessed on August 7, 2019.

[39] Firth, N. P. High dimensional American options. PhD thesis, University of Oxford, 2005, available online at https://ora.ox.ac.uk/objects/uuid:dd46853a-35a1-44a3-a6a4-e39e13bfe78c, last accessed on August 7, 2019.

[40] Fujii, M., Takahashi, A., and Takahashi, M. Asymptotic Expansion as Prior Knowledge in Deep Learning Method for High dimensional BSDEs. *Asia-Pacific Financial Markets* (2019).

[41] García, D. Convergence and biases of Monte Carlo estimates of American option prices using a parametric exercise rule. *J. Econom. Dynam. Control 27*, 10 (2003), 1855–1879.

[42] Glasserman, P. *Monte Carlo methods in financial engineering*, vol. 53 of *Applications of Mathematics (New York).* Springer-Verlag, New York, 2004. Stochastic Modelling and Applied Probability.

[43] Glorot, X., and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010), Y. W. Teh and M. Titterington, Eds., vol. 9 of *Proceedings of Machine Learning Research*, PMLR, pp. 249–256.

[44] Gobet, E., Lemor, J.-P., and Warin, X. A regression-based Monte Carlo method to solve backward stochastic differential equations. *Ann. Appl. Probab. 15*, 3 (2005), 2172–2202.

[45] Goldberg, D. A., and Chen, Y. Beating the curse of dimensionality in options pricing and optimal stopping. *arXiv:1807.02227* (2018), 62 pages.

[46] Goudenège, L., Molent, A., and Zanette, A. Machine Learning for Pricing American Options in High Dimension. *arXiv:1903.11275* (2019), 11 pages.

[47] Guyon, J., and Henry-Labordère, P. *Nonlinear option pricing.* Chapman & Hall/CRC Financial Mathematics Series. CRC Press, Boca Raton, FL, 2014.

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

[49] Haugh, M. B., and Kogan, L. Pricing American options: a duality approach. *Oper. Res. 52*, 2 (2004), 258–270.

[50] Hornik, K., Stinchcombe, M., and White, H. Multilayer feedforward networks are universal approximators. *Neural Networks 2*, 5 (1989), 359 – 366.

[51] Ioffe, S., and Szegedy, C. *Batch normalization: accelerating deep network training by reducing internal covariate shift.* Proceedings of The 32nd International Conference on Machine Learning (ICML), June 2015.

[52] Jain, S., and Oosterlee, C. W. Pricing high-dimensional Bermudan options using the stochastic grid method. *Int. J. Comput. Math. 89*, 9 (2012), 1186–1211.

[53] Jamshidian, F. The duality of optimal exercise and domineering claims: a Doob-Meyer decomposition approach to the Snell envelope. *Stochastics 79*, 1-2 (2007), 27–60.

[54] Jentzen, A., Salimova, D., and Welti, T. Strong convergence for explicit space-time discrete numerical approximation methods for stochastic Burgers equations. *arXiv:1710.07123* (2017), 60 pages. Published in *J. Math. Anal. Appl.*

[55] Jiang, D. R., and Powell, W. B. An approximate dynamic programming algorithm for monotone value functions. *Oper. Res. 63*, 6 (2015), 1489–1511.

[56] Kallsen, J. *Option Pricing.* Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 599–613.

[57] Karatzas, I., and Shreve, S. E. *Brownian motion and stochastic calculus*, second ed., vol. 113 of *Graduate Texts in Mathematics.* Springer-Verlag, New York, 1991.

[58] Kingma, D., and Ba, J. *Adam: a method for stochastic optimization.* Proceedings of the International Conference on Learning Representations (ICLR), 2015.

[59] Klenke, A. *Probability theory.* Universitext. Springer-Verlag London Ltd., London, 2008. A comprehensive course, Translated from the 2006 German original.

[60] Kloeden, P. E., and Platen, E. *Numerical solution of stochastic differential equations*, vol. 23 of *Applications of Mathematics (New York).* Springer-Verlag, Berlin, 1992.

[61] Kohler, M. A regression-based smoothing spline Monte Carlo algorithm for pricing American options in discrete time. *AStA Adv. Stat. Anal. 92*, 2 (2008), 153–178.

[62] Kohler, M. A review on regression-based Monte Carlo methods for pricing American options. In *Recent developments in applied probability and statistics.* Physica, Heidelberg, 2010, pp. 37–58.

[63] Kohler, M., and Krzyżak, A. Pricing of American options in discrete time using least squares estimates with complexity penalties. *J. Statist. Plann. Inference 142*, 8 (2012), 2289–2307.

[64] Kohler, M., Krzyżak, A., and Todorovic, N. Pricing of high-dimensional American options by neural networks. *Math. Finance 20*, 3 (2010), 383–410.

[65] Kohler, M., Krzyżak, A., and Walk, H. Upper bounds for Bermudan options on Markovian data using nonparametric regression and a reduced number of nested Monte Carlo steps. *Statist. Decisions 26*, 4 (2008), 275–288.

[66] Kolodko, A., and Schoenmakers, J. Iterative construction of the optimal Bermudan stopping time. *Finance Stoch. 10*, 1 (2006), 27–49.

[67] Kulikov, A. V., and Gusyatnikov, P. P. Stopping times for fractional brownian motion. In *Computational Management Science: State of the Art 2014*, R. J. Fonseca, G.-W. Weber, and J. Telhada, Eds., vol. 682 of *Lecture Notes in Economics and Mathematical Systems.* Springer International Publishing, Cham, 2016, pp. 195–200.

[68] Labart, C., and Lelong, J. A Parallel Algorithm for solving BSDEs - Application to the pricing and hedging of American options. *arXiv:1102.4666* (2011), 25 pages. Published in *Monte Carlo Methods Appl.*

[69] Lamberton, D., and Lapeyre, B. *Introduction to stochastic calculus applied to finance*, second ed. Chapman & Hall/CRC Financial Mathematics Series. Chapman & Hall/CRC, Boca Raton, FL, 2008.

[70] Lapeyre, B., and Lelong, J. Neural network regression for Bermudan option pricing. *arXiv:1907.06474* (2019), 15 pages.

[71] Lelong, J. Pricing American options using martingale bases. *arXiv:1604.03317* (2016), 25 pages.

[72] Lelong, J. Pricing path-dependent Bermudan options using Wiener chaos expansion: an embarrassingly parallel approach. *arXiv:1901.05672* (2019), 23 pages.

[73] Longstaff, F. A., and Schwartz, E. S. Valuing American options by simulation: A simple least-squares approach. *Rev. Fin. Stud. 14*, 1 (2001), 113–147.

[74] Lord, R., Fang, F., Bervoets, F., and Oosterlee, C. W. A fast and accurate FFT-based method for pricing early-exercise options under Lévy processes. *SIAM J. Sci. Comput. 30*, 4 (2008), 1678–1705.

[75] Maruyama, G. Continuous Markov processes and stochastic equations. *Rend. Circ. Mat. Palermo (2) 4* (1955), 48–90.

[76] Peskir, G., and Shiryaev, A. *Optimal stopping and free-boundary problems.* Lectures in Mathematics ETH Zürich. Birkhäuser Verlag, Basel, 2006.

[77] Rogers, L. C. G. Monte Carlo valuation of American options. *Math. Finance 12*, 3 (2002), 271–286.

[78] Rogers, L. C. G. Dual valuation and hedging of Bermudan options. *SIAM J. Financial Math. 1*, 1 (2010), 604–608.

[79] Schoenmakers, J., Zhang, J., and Huang, J. Optimal dual martingales, their analysis, and application to new algorithms for Bermudan products. *SIAM J. Financial Math. 4*, 1 (2013), 86–116.

[80] Schweizer, M. On Bermudan options. In *Advances in finance and stochastics.* Springer, Berlin, 2002, pp. 257–270.

[81] Shreve, S. E. *Stochastic calculus for finance. II.* Springer Finance. Springer-Verlag, New York, 2004. Continuous-time models.

[82] Sirignano, J., and Spiliopoulos, K. DGM: A deep learning algorithm for solving partial differential equations. *arXiv:1708.07469* (2017), 7 pages.

[83] Sirignano, J., and Spiliopoulos, K. Stochastic gradient descent in continuous time. *SIAM J. Financial Math. 8*, 1 (2017), 933–961.

[84] Sirignano, J., and Spiliopoulos, K. DGM: a deep learning algorithm for solving partial differential equations. *J. Comput. Phys. 375* (2018), 1339–1364.

[85] Smirnov, M. Javascript options and implied volatility calculator. http://www.math.columbia.edu/ smirnov/options13.html. Columbia University. Last accessed on August 7, 2019.

[86] Solan, E., Tsirelson, B., and Vieille, N. Random Stopping Times in Stopping Problems and Stopping Games. *arXiv:1211.5802* (2012), 21 pages.

[87] Tilley, J. A. Valuing american options in a path simulation model. *Transactions of the Society of Actuaries 45* (1993), 83–104.

[88] Tsitsiklis, J. N., and Van Roy, B. Optimal stopping of Markov processes: Hilbert space theory, approximation algorithms, and an application to pricing high-dimensional financial derivatives. *IEEE Trans. Automat. Control 44*, 10 (1999), 1840–1851.

[89] Tsitsiklis, J. N., and Van Roy, B. Regression methods for pricing complex American-style options. *IEEE Transactions on Neural Networks 12*, 4 (2001), 694–703.

[90] Wang, H., Chen, H., Sudjianto, A., Liu, R., and Shen, Q. Deep Learning-Based BSDE Solver for Libor Market Model with Application to Bermudan Swaption Pricing and Hedging. *arXiv:1807.06622* (2018), 36 pages.