Journal of Scientific Computing (2021) 88:73
https://doi.org/10.1007/s10915-021-01590-0

Solving the Kolmogorov PDE by Means of Deep Learning

Christian Beck[1,2] · Sebastian Becker[1,3] · Philipp Grohs[4] · Nor Jaafari[3] · Arnulf Jentzen[2,5]

Received: 6 January 2020 / Revised: 16 April 2021 / Accepted: 18 April 2021

Abstract
Stochastic differential equations (SDEs) and the Kolmogorov partial differential equations (PDEs) associated to them have been widely used in models from engineering, finance, and the natural sciences. In particular, SDEs and Kolmogorov PDEs, respectively, are highly employed in models for the approximative pricing of financial derivatives. Kolmogorov PDEs and SDEs, respectively, can typically not be solved explicitly and it has been and still is an active topic of research to design and analyze numerical methods which are able to approximately solve Kolmogorov PDEs and SDEs, respectively. Nearly all approximation methods for Kolmogorov PDEs in the literature suffer under the curse of dimensionality or only provide approximations of the solution of the PDE at a single fixed space-time point. In this paper we derive and propose a numerical approximation method which aims to overcome both of the above mentioned drawbacks and intends to deliver a numerical approximation of the Kolmogorov PDE on an entire region $[a, b]^d$ without suffering from the curse of dimensionality. Numerical results on examples including the heat equation, the Black–Scholes model, the stochastic Lorenz equation, and the Heston model suggest that

✉ Nor Jaafari
nor.jaafari@zenai.ch

Christian Beck
christian.beck@yahoo.de

Sebastian Becker
sebastian.becker@math.ethz.ch

Philipp Grohs
philipp.grohs@univie.ac.at

Arnulf Jentzen
ajentzen@uni-muenster.de

1 Department of Mathematics, ETH Zurich, Zurich, Switzerland

2 Applied Mathematics Münster: Institute for Analysis and Numerics, WWU Münster, Münster, Germany

3 ZENAI AG, Zurich, Switzerland

4 Faculty of Mathematics and Research Platform Data Science, University of Vienna, Vienna, Austria

5 Seminar for Applied Mathematics, ETH Zurich, Zurich, Switzerland

Published online: 31 July 2021

the proposed approximation algorithm is quite effective in high dimensions in terms of both accuracy and speed.

Keywords Numerical approximation method · Stochastic differential equations · Kolmogorov equations · Deep learning

1 Introduction

Stochastic differential equations (SDEs) and the Kolmogorov partial differential equations (PDEs) associated to them have been widely used in models from engineering, finance, and the natural sciences. In particular, SDEs and Kolmogorov PDEs, respectively, are highly employed in models for the approximative pricing of financial derivatives. Kolmogorov PDEs and SDEs, respectively, can typically not be solved explicitly and it has been and still is an active topic of research to design and analyze numerical methods which are able to approximately solve Kolmogorov PDEs and SDEs, respectively (see, e.g., [22,49] and the references mentioned therein). In particular, there are nowadays several different types of numerical approximation methods for Kolmogorov PDEs in the literature including deterministic numerical approximation methods such as finite differences based approximation methods (cf., for example, [11,43,60]) and finite elements based approximation methods (cf., for example, [12,61]) as well as random numerical approximation methods based on Monte Carlo methods (cf., for example, [19,22]) and discretizations of the underlying SDEs (cf., for example, [42,49] and the references mentioned therein). The above mentioned deterministic approximation methods for PDEs work quite efficiently in one or two space dimensions but cannot be used in the case of high-dimensional PDEs as they suffer from the so-called curse of dimensionality (cf. Bellman [6]) in the sense that the computational effort of the considered approximation algorithm grows exponentially in the PDE dimension. The above mentioned random numerical approximation methods involving Monte Carlo approximations typically overcome this curse of dimensionality but only provide approximations of the Kolmogorov PDE at a single fixed space-time point.

The key contribution of this paper is to derive and propose a numerical approximation method which aims to overcome both of the above mentioned drawbacks and intends to deliver a numerical approximation of the Kolmogorov PDE on an entire region $[a, b]^d$ without suffering from the curse of dimensionality. The numerical scheme, which we propose in this work, is inspired by recently developed deep learning based approximation algorithms for PDEs in the literature (cf., for example, [5,18,28,30,58,59]). To derive the proposed approximation scheme we first reformulate the considered Kolmogorov PDE as a suitable infinite dimensional stochastic optimization problem (see items (ii)–(iii) in Proposition 1.1 below for details). This infinite dimensional stochastic optimization problem is then temporally discretized by means of suitable discretizations of the underlying SDE and it is spatially discretized by means of fully connected deep artificial neural network approximations (see (38) in Sect. 3.3 as well as Sects. 3.1–3.2 below). The resulting finite dimensional stochastic optimization problem is then solved by means of stochastic gradient descent type optimization algorithms (see (40) in Sect. 3.3, Framework 3.1 in Sect. 3.4, Framework 3.2 in Sect. 3.5, as well as (54)–(55) in Sect. 4.1). We test the proposed approximation method numerically in the case of several examples of SDEs and PDEs, respectively (see Sects. 4.2–4.6 below for details). The obtained numerical results indicate that the proposed approximation algorithm is quite effective in high dimensions in terms of both accuracy and speed.

As mentioned above, a key component of the proposed approximation algorithm is to reformulate a Kolmogorov PDE as a learning problem, that is to say as an infinite dimensional optimization problem. To familiarize the reader with this reformulation we present in the following result a way how solutions of Kolmogorov PDEs can be identified as solutions of infinite dimensional optimization problems.

Proposition 1.1 *Let $d, m \in \mathbb{N}$, $T \in (0, \infty)$, $a \in \mathbb{R}$, $b \in (a, \infty)$, let $\mu \colon \mathbb{R}^d \to \mathbb{R}^d$ and $\sigma \colon \mathbb{R}^d \to \mathbb{R}^{d \times m}$ be globally Lipschitz continuous, let $\varphi \colon \mathbb{R}^d \to \mathbb{R}$ be a function, let $u = (u(t,x))_{(t,x) \in [0,T] \times \mathbb{R}^d} \in C^{1,2}([0,T] \times \mathbb{R}^d, \mathbb{R})$ be a function with at most polynomially growing partial derivatives which satisfies for every $t \in [0,T]$, $x \in \mathbb{R}^d$ that $u(0,x) = \varphi(x)$ and*

$$\tfrac{\partial u}{\partial t}(t,x) = \tfrac{1}{2} \operatorname{Trace}_{\mathbb{R}^d}\big(\sigma(x)[\sigma(x)]^*(\operatorname{Hess}_x u)(t,x)\big) + \langle \mu(x), (\nabla_x u)(t,x) \rangle_{\mathbb{R}^d}, \tag{1}$$

let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space with a normal filtration $(\mathbb{F}_t)_{t \in [0,T]}$, let $W \colon [0,T] \times \Omega \to \mathbb{R}^m$ be a standard $(\mathbb{F}_t)_{t \in [0,T]}$-Brownian motion, let $\xi \colon \Omega \to [a,b]^d$ be a continuous uniformly distributed $\mathbb{F}_0/\mathcal{B}([a,b]^d)$-measurable random variable, and let $\mathbb{X} = (\mathbb{X}_t)_{t \in [0,T]} \colon [0,T] \times \Omega \to \mathbb{R}^d$ be an $(\mathbb{F}_t)_{t \in [0,T]}$-adapted stochastic process with continuous sample paths which satisfies that for every $t \in [0,T]$ it holds $\mathbb{P}$-a.s. that

$$\mathbb{X}_t = \xi + \int_0^t \mu(\mathbb{X}_s)\, ds + \int_0^t \sigma(\mathbb{X}_s)\, dW_s. \tag{2}$$

Then

(i) *it holds that there exists a unique continuous $U \colon [a,b]^d \to \mathbb{R}$ such that*

$$\mathbb{E}\big[|\varphi(\mathbb{X}_T) - U(\xi)|^2\big] = \inf_{v \in C([a,b]^d, \mathbb{R})} \mathbb{E}\big[|\varphi(\mathbb{X}_T) - v(\xi)|^2\big], \tag{3}$$

and

(ii) *it holds for every $x \in [a,b]^d$ that $U(x) = u(T,x)$.*

Proposition 1.1 is an extract of Corollary 2.4 (see Sect. 2.3 below) which constitutes the main theoretical motivation for the proposed approximation algorithm.

The remainder of this article is organized as follows. In Sect. 2 we derive the proposed approximation algorithm (see Sects. 2.1–3.3 below) and we present a detailed description of the proposed approximation algorithm in a special case (see Sect. 3.4 below) as well as in the general case (see Sect. 3.5 below). In Sect. 4 we test the proposed algorithm numerically in the case of several examples of SDEs and PDEs, respectively. The employed source codes for the numerical simulations in Sect. 4 can be found on GitHub (see https://github.com/sebbecker/kolmogorov).

2 Reformulation of Kolmogorov Partial Differential Equations (PDEs) as Stochastic Learning Problems

In this section we describe the approximation problem which we intend to solve (see Sect. 2.1 below) and we derive (see Sects. 2.2–3.3 below) and specify (see Sects. 3.4–3.5 below) the numerical scheme which we suggest to use to solve this approximation problem (cf., for example, E et al. [58], Han et al. [28], Fujii et al. [18], and Henry-Labordere [30] for related derivations and related approximation schemes).

2.1 Kolmogorov Partial Differential Equations (PDEs)

Let $T \in (0, \infty)$, $d \in \mathbb{N}$, 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, let $\varphi\colon \mathbb{R}^d \to \mathbb{R}$ be a function, and let $u = (u(t,x))_{(t,x)\in[0,T]\times\mathbb{R}^d} \in C^{1,2}([0,T]\times\mathbb{R}^d, \mathbb{R})$ be a function with at most polynomially growing partial derivatives which satisfies for every $t \in [0,T]$, $x \in \mathbb{R}^d$ that $u(0,x) = \varphi(x)$ and

$$\tfrac{\partial u}{\partial t}(t,x) = \tfrac{1}{2}\,\mathrm{Trace}_{\mathbb{R}^d}\big(\sigma(x)[\sigma(x)]^*(\mathrm{Hess}_x\, u)(t,x)\big) + \langle \mu(x), (\nabla_x u)(t,x)\rangle_{\mathbb{R}^d}. \tag{4}$$

Our goal is to approximately calculate the function $\mathbb{R}^d \ni x \mapsto u(T,x) \in \mathbb{R}$ on some subset of $\mathbb{R}^d$. To fix ideas we consider real numbers $a, b \in \mathbb{R}$ with $a < b$ and we suppose that our goal is to approximately calculate the function $[a,b]^d \ni x \mapsto u(T,x) \in \mathbb{R}$.

2.2 On Stochastic Differential Equations and Kolmogorov PDEs

In this subsection we provide a probabilistic representation for the solutions of the PDE (4), that is, we recall the classical Feynman–Kac formula for the PDE (4) (cf., for example, Øksendal [52, Chapter 8]).

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space with a normal filtration $(\mathbb{F}_t)_{t\in[0,T]}$, let $W\colon [0,T]\times\Omega \to \mathbb{R}^d$ be a standard $(\Omega, \mathcal{F}, \mathbb{P}, (\mathbb{F}_t)_{t\in[0,T]})$-Brownian motion, and for every $x \in \mathbb{R}^d$ let $X^x = (X^x_t)_{t\in[0,T]}\colon [0,T]\times\Omega \to \mathbb{R}^d$ be an $(\mathbb{F}_t)_{t\in[0,T]}$-adapted stochastic process with continuous sample paths which satisfies that for every $t \in [0,T]$ it holds $\mathbb{P}$-a.s. that

$$X^x_t = x + \int_0^t \mu(X^x_s)\,ds + \int_0^t \sigma(X^x_s)\,dW_s. \tag{5}$$

The Feynman–Kac formula (cf., for example, Hairer et al. [27, Corollary 4.17 and Remark 4.1]) and (4) hence yield that for every $x \in \mathbb{R}^d$ it holds that

$$u(T,x) = \mathbb{E}\big[u(0, X^x_T)\big] = \mathbb{E}\big[\varphi(X^x_T)\big]. \tag{6}$$

2.3 Formulation as Minimization Problem

In the next step we exploit (6) to formulate a minimization problem which is uniquely solved by the function $[a,b]^d \ni x \mapsto u(T,x) \in \mathbb{R}$ (cf. (4) above). For this we first recall the L^2-minimization property of the expectation of a real-valued random variable (see Lemma 2.1 below). Then we extend this minimization result to certain random fields (see Proposition 2.2 below). Thereafter, we apply Proposition 2.2 to random fields in the context of the Feynman–Kac representation (6) to obtain Corollary 2.4 below. Corollary 2.4 provides a minimization problem (see, for instance, (28) below) which has the function $[a,b]^d \ni x \mapsto u(T,x) \in \mathbb{R}$ as the unique global minimizer. Our proof of Corollary 2.4 is based on the elementary auxiliary result in Lemma 2.3.

Lemma 2.1 *Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space and let $X\colon \Omega \to \mathbb{R}$ be an $\mathcal{F}/\mathcal{B}(\mathbb{R})$-measurable random variable which satisfies $\mathbb{E}[|X|^2] < \infty$. Then*

(i) *it holds for every $y \in \mathbb{R}$ that*

$$\mathbb{E}\big[|X-y|^2\big] = \mathbb{E}\big[|X-\mathbb{E}[X]|^2\big] + |\mathbb{E}[X]-y|^2, \tag{7}$$

(ii) *it holds that there exists a unique real number $z \in \mathbb{R}$ such that*

$$\mathbb{E}\big[|X-z|^2\big] = \inf_{y\in\mathbb{R}} \mathbb{E}\big[|X-y|^2\big], \tag{8}$$

and

(iii) *it holds that*

$$\mathbb{E}\big[|X - \mathbb{E}[X]|^2\big] = \inf_{y\in\mathbb{R}} \mathbb{E}\big[|X - y|^2\big]. \tag{9}$$

Proof of Lemma 2.1 Observe that the fact that $\mathbb{E}[|X|] < \infty$ ensures that for every $y \in \mathbb{R}$ it holds that

$$\begin{aligned}\mathbb{E}\big[|X - y|^2\big] &= \mathbb{E}\big[|X - \mathbb{E}[X] + \mathbb{E}[X] - y|^2\big]\\ &= \mathbb{E}\big[|X - \mathbb{E}[X]|^2 + 2(X - \mathbb{E}[X])(\mathbb{E}[X] - y) + |\mathbb{E}[X] - y|^2\big]\\ &= \mathbb{E}\big[|X - \mathbb{E}[X]|^2\big] + 2(\mathbb{E}[X] - y)\mathbb{E}\big[X - \mathbb{E}[X]\big] + |\mathbb{E}[X] - y|^2\\ &= \mathbb{E}\big[|X - \mathbb{E}[X]|^2\big] + |\mathbb{E}[X] - y|^2.\end{aligned} \tag{10}$$

This establishes item (i). Item (ii) and item (iii) are immediate consequences of item (i). The proof of Lemma 2.1 is thus completed. □

Proposition 2.2 *Let $a \in \mathbb{R}$, $b \in (a, \infty)$, let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, let $X = (X_x)_{x\in[a,b]^d} \colon [a,b]^d \times \Omega \to \mathbb{R}$ be $(\mathcal{B}([a,b]^d) \otimes \mathcal{F})/\mathcal{B}(\mathbb{R})$-measurable, assume for every $x \in [a,b]^d$ that $\mathbb{E}[|X_x|^2] < \infty$, and assume that $[a,b]^d \ni x \mapsto \mathbb{E}[X_x] \in \mathbb{R}$ is continuous. Then*

(i) *it holds that there exists a unique continuous $u \colon [a,b]^d \to \mathbb{R}$ such that*

$$\int_{[a,b]^d} \mathbb{E}\big[|X_x - u(x)|^2\big]\,dx = \inf_{v\in C([a,b]^d,\mathbb{R})} \left(\int_{[a,b]^d} \mathbb{E}\big[|X_x - v(x)|^2\big]\,dx \right) \tag{11}$$

and

(ii) *it holds for every $x \in [a,b]^d$ that $u(x) = \mathbb{E}[X_x]$.*

Proof of Proposition 2.2 Observe that item (i) in Lemma 2.1 and the hypothesis that $\forall\, x \in [a,b]^d \colon \mathbb{E}[|X_x|^2] < \infty$ ensure that for every $u \colon [a,b]^d \to \mathbb{R}$ and every $x \in [a,b]^d$ it holds that $\mathbb{E}\big[|X_x - u(x)|^2\big] = \mathbb{E}\big[|X_x - \mathbb{E}[X_x]|^2\big] + |\mathbb{E}[X_x] - u(x)|^2$. Fubini's theorem (see, e.g., Klenke [40, Theorem 14.16]) hence proves that for every continuous $u \colon [a,b]^d \to \mathbb{R}$ it holds that

$$\int_{[a,b]^d} \mathbb{E}\big[|X_x - u(x)|^2\big]\,dx = \int_{[a,b]^d} \mathbb{E}\big[|X_x - \mathbb{E}[X_x]|^2\big]\,dx + \int_{[a,b]^d} |\mathbb{E}[X_x] - u(x)|^2\,dx. \tag{12}$$

The hypothesis that $[a,b]^d \ni x \mapsto \mathbb{E}[X_x] \in \mathbb{R}$ is continuous therefore demonstrates that

$$\begin{aligned}&\int_{[a,b]^d} \mathbb{E}\big[|X_x - \mathbb{E}[X_x]|^2\big]\,dx \geq \inf_{v\in C([a,b]^d,\mathbb{R})} \left(\int_{[a,b]^d} \mathbb{E}\big[|X_x - v(x)|^2\big]\,dx \right)\\ &= \inf_{v\in C([a,b]^d,\mathbb{R})} \left(\int_{[a,b]^d} \mathbb{E}\big[|X_x - \mathbb{E}[X_x]|^2\big]\,dx + \int_{[a,b]^d} |\mathbb{E}[X_x] - v(x)|^2\,dx \right)\\ &\geq \inf_{v\in C([a,b]^d,\mathbb{R})} \left(\int_{[a,b]^d} \mathbb{E}\big[|X_x - \mathbb{E}[X_x]|^2\big]\,dx \right) = \int_{[a,b]^d} \mathbb{E}\big[|X_x - \mathbb{E}[X_x]|^2\big]\,dx.\end{aligned} \tag{13}$$

Hence, we obtain that

$$\int_{[a,b]^d} \mathbb{E}\big[|X_x - \mathbb{E}[X_x]|^2\big]\,dx = \inf_{v\in C([a,b]^d,\mathbb{R})} \left(\int_{[a,b]^d} \mathbb{E}\big[|X_x - v(x)|^2\big]\,dx \right). \tag{14}$$

Again the fact that $[a,b]^d \ni x \mapsto \mathbb{E}[X_x] \in \mathbb{R}$ is continuous therefore proves that there exists a continuous $u\colon [a,b]^d \to \mathbb{R}$ such that

$$\int_{[a,b]^d} \mathbb{E}\big[|X_x - u(x)|^2\big]\,dx = \inf_{v\in C([a,b]^d,\mathbb{R})} \left(\int_{[a,b]^d} \mathbb{E}\big[|X_x - v(x)|^2\big]\,dx \right). \tag{15}$$

Next observe that (12) and (14) yield that for every continuous $u\colon [a,b]^d \to \mathbb{R}$ with

$$\int_{[a,b]^d} \mathbb{E}\big[|X_x - u(x)|^2\big]\,dx = \inf_{v\in C([a,b]^d,\mathbb{R})} \left(\int_{[a,b]^d} \mathbb{E}\big[|X_x - v(x)|^2\big]\,dx \right) \tag{16}$$

it holds that

$$\begin{aligned} &\int_{[a,b]^d} \mathbb{E}\big[|X_x - \mathbb{E}[X_x]|^2\big]\,dx \\ &= \inf_{v\in C([a,b]^d,\mathbb{R})} \left(\int_{[a,b]^d} \mathbb{E}\big[|X_x - v(x)|^2\big]\,dx \right) = \int_{[a,b]^d} \mathbb{E}\big[|X_x - u(x)|^2\big]\,dx \\ &= \int_{[a,b]^d} \mathbb{E}\big[|X_x - \mathbb{E}[X_x]|^2\big]\,dx + \int_{[a,b]^d} |\mathbb{E}[X_x] - u(x)|^2\,dx. \end{aligned} \tag{17}$$

Hence, we obtain that for every continuous $u\colon [a,b]^d \to \mathbb{R}$ with

$$\int_{[a,b]^d} \mathbb{E}\big[|X_x - u(x)|^2\big]\,dx = \inf_{v\in C([a,b]^d,\mathbb{R})} \left(\int_{[a,b]^d} \mathbb{E}\big[|X_x - v(x)|^2\big]\,dx \right) \tag{18}$$

it holds that $\int_{[a,b]^d} |\mathbb{E}[X_x] - u(x)|^2\,dx = 0$. This and again the hypothesis that $[a,b]^d \ni x \mapsto \mathbb{E}[X_x] \in \mathbb{R}$ is continuous yield that for every continuous $u\colon [a,b]^d \to \mathbb{R}$ with

$$\int_{[a,b]^d} \mathbb{E}\big[|X_x - u(x)|^2\big]\,dx = \inf_{v\in C([a,b]^d,\mathbb{R})} \left(\int_{[a,b]^d} \mathbb{E}\big[|X_x - v(x)|^2\big]\,dx \right) \tag{19}$$

and every $x \in [a,b]^d$ it holds that $u(x) = \mathbb{E}[X_x]$. Combining this with (15) completes the proof of Proposition 2.2. □

Lemma 2.3 *Let $d, m \in \mathbb{N}$, $T \in (0,\infty)$, $L, a \in \mathbb{R}$, $b \in (a,\infty)$, let $\mu\colon \mathbb{R}^d \to \mathbb{R}^d$ and $\sigma\colon \mathbb{R}^d \to \mathbb{R}^{d\times m}$ satisfy for every $x, y \in \mathbb{R}^d$ that $\max\{\|\mu(x) - \mu(y)\|_{\mathbb{R}^d}, \|\sigma(x) - \sigma(y)\|_{\mathrm{HS}(\mathbb{R}^m,\mathbb{R}^d)}\} \leq L\|x - y\|_{\mathbb{R}^d}$, let $\Phi\colon C([0,T],\mathbb{R}^d) \to \mathbb{R}$ be an at most polynomially growing continuous function, let $(\Omega,\mathcal{F},\mathbb{P})$ be a probability space with a normal filtration $(\mathbb{F}_t)_{t\in[0,T]}$, let $\xi\colon \Omega \to [a,b]^d$ be a continuous uniformly distributed $\mathbb{F}_0/\mathcal{B}([a,b]^d)$-measurable random variable, let $W\colon [0,T]\times\Omega \to \mathbb{R}^m$ be a standard $(\mathbb{F}_t)_{t\in[0,T]}$-Brownian motion, for every $x \in [a,b]^d$ let $X^x = (X^x_t)_{t\in[0,T]}\colon [0,T]\times\Omega \to \mathbb{R}^d$ be an $(\mathbb{F}_t)_{t\in[0,T]}$-adapted stochastic process with continuous sample paths which satisfies that for every $t \in [0,T]$ it holds $\mathbb{P}$-a.s. that*

$$X^x_t = x + \int_0^t \mu(X^x_s)\,ds + \int_0^t \sigma(X^x_s)\,dW_s, \tag{20}$$

and let $\mathbb{X}\colon [0,T]\times\Omega \to \mathbb{R}^d$ be an $(\mathbb{F}_t)_{t\in[0,T]}$-adapted stochastic process with continuous sample paths which satisfies that for every $t \in [0,T]$ it holds $\mathbb{P}$-a.s. that

$$\mathbb{X}_t = \xi + \int_0^t \mu(\mathbb{X}_s)\,ds + \int_0^t \sigma(\mathbb{X}_s)\,dW_s. \tag{21}$$

Then

(i) *it holds for every* $x \in [a,b]^d$ *that* $\Omega \ni \omega \mapsto \Phi((X_t^x(\omega))_{t\in[0,T]}) \in \mathbb{R}$ *and* $\Omega \ni \omega \mapsto \Phi((\mathbb{X}_t(\omega))_{t\in[0,T]}) \in \mathbb{R}$ *are* $\mathcal{F}/\mathcal{B}(\mathbb{R})$*-measurable,*
(ii) *it holds for every* $p \in [2,\infty)$, $x, y \in [a,b]^d$ *that*

$$\left(\mathbb{E}\left[\sup_{t\in[0,T]} \|X_t^x - X_t^y\|_{\mathbb{R}^d}^p\right]\right)^{1/p} \leq \sqrt{2}\exp\left(L^2 T\left[p+\sqrt{T}\right]^2\right)\|x-y\|_{\mathbb{R}^d}, \tag{22}$$

(iii) *it holds for every* $x \in [a,b]^d$ *that* $\mathbb{E}\big[|\Phi((X_t^x)_{t\in[0,T]})| + |\Phi((\mathbb{X}_t)_{t\in[0,T]})|\big] < \infty$,
(iv) *it holds that* $[a,b]^d \ni x \mapsto \mathbb{E}[\Phi((X_t^x)_{t\in[0,T]})] \in \mathbb{R}$ *is continuous, and*
(v) *it holds that*

$$\mathbb{E}\big[\Phi((\mathbb{X}_t)_{t\in[0,T]})\big] = \tfrac{1}{(b-a)^d}\left(\int_{[a,b]^d} \mathbb{E}\big[\Phi((X_t^x)_{t\in[0,T]})\big]\,dx\right). \tag{23}$$

Proof of Lemma 2.3 The proof of Lemma 2.3 is essentially well-known in the scientific literature; cf., for instance, Rogers and Williams [53, Corollary V.11.7 and Theorem V.13.1] or the arXiv version of this article. □

Corollary 2.4 *Let* $d, m \in \mathbb{N}$, $T \in (0,\infty)$, $a \in \mathbb{R}$, $b \in (a,\infty)$, *let* $\mu\colon \mathbb{R}^d \to \mathbb{R}^d$ *and* $\sigma\colon \mathbb{R}^d \to \mathbb{R}^{d\times m}$ *be globally Lipschitz continuous, let* $\varphi\colon \mathbb{R}^d \to \mathbb{R}$ *be a function, let* $u = (u(t,x))_{(t,x)\in[0,T]\times\mathbb{R}^d} \in C^{1,2}([0,T]\times\mathbb{R}^d, \mathbb{R})$ *be a function with at most polynomially growing partial derivatives which satisfies for every* $t \in [0,T]$, $x \in \mathbb{R}^d$ *that* $u(0,x) = \varphi(x)$ *and*

$$\tfrac{\partial u}{\partial t}(t,x) = \tfrac{1}{2}\operatorname{Trace}_{\mathbb{R}^d}\big(\sigma(x)[\sigma(x)]^*(\operatorname{Hess}_x u)(t,x)\big) + \langle \mu(x), (\nabla_x u)(t,x)\rangle_{\mathbb{R}^d}, \tag{24}$$

let $(\Omega, \mathcal{F}, \mathbb{P})$ *be a probability space with a normal filtration* $(\mathbb{F}_t)_{t\in[0,T]}$*, let* $W\colon [0,T]\times\Omega \to \mathbb{R}^m$ *be a standard* $(\mathbb{F}_t)_{t\in[0,T]}$*-Brownian motion, let* $\xi\colon \Omega \to [a,b]^d$ *be a continuous uniformly distributed* $\mathbb{F}_0/\mathcal{B}([a,b]^d)$*-measurable random variable, and let* $\mathbb{X} = (\mathbb{X}_t)_{t\in[0,T]}\colon [0,T]\times\Omega \to \mathbb{R}^d$ *be an* $(\mathbb{F}_t)_{t\in[0,T]}$*-adapted stochastic process with continuous sample paths which satisfies that for every* $t \in [0,T]$ *it holds* $\mathbb{P}$*-a.s. that*

$$\mathbb{X}_t = \xi + \int_0^t \mu(\mathbb{X}_s)\,ds + \int_0^t \sigma(\mathbb{X}_s)\,dW_s. \tag{25}$$

Then

(i) *it holds that* $\varphi\colon \mathbb{R}^d \to \mathbb{R}$ *is twice continuously differentiable with at most polynomially growing derivatives,*
(ii) *it holds that there exists a unique continuous* $U\colon [a,b]^d \to \mathbb{R}$ *such that*

$$\mathbb{E}\big[|\varphi(\mathbb{X}_T) - U(\xi)|^2\big] = \inf_{v\in C([a,b]^d,\mathbb{R})} \mathbb{E}\big[|\varphi(\mathbb{X}_T) - v(\xi)|^2\big], \tag{26}$$

and
(iii) *it holds for every* $x \in [a,b]^d$ *that* $U(x) = u(T,x)$.

Proof of Corollary 2.4 Corollary 2.4 is a direct consequence of combining Proposition 2.2 and Lemma 2.3 with, e.g., Cox et al. [14, Theorem 3.5], Hutzenthaler et al. [34, Proposition 4.5], Aliprantis and Border [2, Lemma 4.51], and Hairer et al. [27, Corollary 4.17]. □

In the next step we use Corollary 2.4 to obtain a minimization problem which is uniquely solved by $[a,b]^d \ni x \mapsto u(T,x) \in \mathbb{R}$. More specifically, let $\xi\colon \Omega \to [a,b]^d$ be a continuously uniformly distributed $\mathbb{F}_0/\mathcal{B}([a,b]^d)$-measurable random variable, and let

$\mathbb{X}\colon [0,T]\times\Omega\to\mathbb{R}^d$ be an $(\mathbb{F}_t)_{t\in[0,T]}$-adapted stochastic process with continuous sample paths which satisfies that for every $t\in[0,T]$ it holds $\mathbb{P}$-a.s. that

$$\mathbb{X}_t = \xi + \int_0^t \mu(\mathbb{X}_s)\,ds + \int_0^t \sigma(\mathbb{X}_s)\,dW_s. \tag{27}$$

Corollary 2.4 then guarantees that $[a,b]^d \ni x \mapsto u(T,x)\in\mathbb{R}$ is the unique global minimizer of

$$C([a,b]^d,\mathbb{R}) \ni v \mapsto \mathbb{E}\big[|\varphi(\mathbb{X}_T) - v(\xi)|^2\big] \in \mathbb{R}. \tag{28}$$

In the following two subsections we derive an approximated minimization problem by discretizing the stochastic process $\mathbb{X}\colon [0,T]\times\Omega\to\mathbb{R}^d$ (see Sect. 3.1 below) and by employing a deep neural network approximation for $\mathbb{R}^d\ni x\mapsto u(T,x)\in\mathbb{R}$ (see Sect. 3.2 below).

3 Derivation and Description of the Proposed Approximation Algorithm

3.1 Discretization of the Stochastic Differential Equation

In this subsection we use the Euler–Maruyama scheme (cf., for example, Kloeden and Platen [41] and Maruyama [46]) to temporally discretize the solution process $\mathbb{X}$ of the SDE (27).

More specifically, let $N\in\mathbb{N}$, let $t_0,t_1,\ldots,t_N\in[0,\infty)$ satisfy that

$$0 = t_0 < t_1 < \ldots < t_N = T. \tag{29}$$

Note that (27) implies that for every $n\in\{0,1,\ldots,N-1\}$ it holds $\mathbb{P}$-a.s. that

$$\mathbb{X}_{t_{n+1}} = \mathbb{X}_{t_n} + \int_{t_n}^{t_{n+1}} \mu(\mathbb{X}_s)\,ds + \int_{t_n}^{t_{n+1}} \sigma(\mathbb{X}_s)\,dW_s. \tag{30}$$

This suggests that for sufficiently small mesh size $\sup_{n\in\{0,1,\ldots,N-1\}}(t_{n+1}-t_n)$ it holds that

$$\mathbb{X}_{t_{n+1}} \approx \mathbb{X}_{t_n} + \mu(\mathbb{X}_{t_n})\,(t_{n+1}-t_n) + \sigma(\mathbb{X}_{t_n})\,(W_{t_{n+1}} - W_{t_n}). \tag{31}$$

Let $\mathcal{X}\colon\{0,1,\ldots,N\}\times\Omega\to\mathbb{R}^d$ be the stochastic process which satisfies for every $n\in\{0,1,\ldots,N-1\}$ that $\mathcal{X}_0=\xi$ and

$$\mathcal{X}_{n+1} = \mathcal{X}_n + \mu(\mathcal{X}_n)\,(t_{n+1}-t_n) + \sigma(\mathcal{X}_n)\,(W_{t_{n+1}} - W_{t_n}). \tag{32}$$

Observe that (31) and (32) suggest, in turn, that for every $n\in\{0,1,2,\ldots,N\}$ it holds that

$$\mathcal{X}_n \approx \mathbb{X}_{t_n}. \tag{33}$$

Convergence results for the Euler–Maruyama scheme are well-known in the literature (cf., for example, Kloeden and Platen [41], Milstein [48], Müller-Gronbach and Ritter [51], and the references mentioned therein).

3.2 Deep Artificial Neural Network Approximations

In this subsection we employ suitable approximations for the solution $\mathbb{R}^d\ni x\mapsto u(T,x)\in\mathbb{R}$ of the PDE (4) at time T.

More specifically, let $v \in \mathbb{N}$ and let $\mathbb{U} = (\mathbb{U}(\theta, x))_{(\theta,x)\in\mathbb{R}^v\times\mathbb{R}^d} \colon \mathbb{R}^v \times \mathbb{R}^d \to \mathbb{R}$ be continuous. For every *suitable* $\theta \in \mathbb{R}^v$ and every $x \in [a, b]^d$ we think of $\mathbb{U}(\theta, x) \in \mathbb{R}$ as an appropriate approximation

$$\mathbb{U}(\theta, x) \approx u(T, x) \tag{34}$$

of $u(T, x)$. We suggest to choose $\mathbb{U} \colon \mathbb{R}^v \times \mathbb{R}^d \to \mathbb{R}$ as a deep neural network (cf., for example, Bishop [9]). For instance, let $\mathcal{L}_d \colon \mathbb{R}^d \to \mathbb{R}^d$ satisfy for every $x = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d$ that

$$\mathcal{L}_d(x) = \left(\frac{\exp(x_1)}{\exp(x_1) + 1}, \frac{\exp(x_2)}{\exp(x_2) + 1}, \ldots, \frac{\exp(x_d)}{\exp(x_d) + 1} \right) \tag{35}$$

(multidimensional version of the standard logistic function), for every $k, l \in \mathbb{N}$, $v \in \mathbb{N}_0 = \{0\} \cup \mathbb{N}$, $\theta = (\theta_1, \theta_2, \ldots, \theta_v) \in \mathbb{R}^v$ with $v + l(k + 1) \leq v$ let $A^{\theta,v}_{k,l} \colon \mathbb{R}^k \to \mathbb{R}^l$ satisfy for every $x = (x_1, x_2, \ldots, x_k) \in \mathbb{R}^k$ that

$$A^{\theta,v}_{k,l}(x) = \begin{pmatrix} \theta_{v+1} & \theta_{v+2} & \ldots & \theta_{v+k} \\ \theta_{v+k+1} & \theta_{v+k+2} & \ldots & \theta_{v+2k} \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{v+(l-1)k+1} & \theta_{v+(l-1)k+2} & \ldots & \theta_{v+lk} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{pmatrix} + \begin{pmatrix} \theta_{v+kl+1} \\ \theta_{v+kl+2} \\ \vdots \\ \theta_{v+kl+l} \end{pmatrix}, \tag{36}$$

let $s \in \{3, 4, 5, 6, \ldots\}$, assume that $(s - 1)d(d + 1) + d + 1 \leq v$, and let $\mathbb{U} \colon \mathbb{R}^v \times \mathbb{R}^d \to \mathbb{R}$ satisfy for every $\theta \in \mathbb{R}^v$, $x \in \mathbb{R}^d$ that

$$\mathbb{U}(\theta, x) = \big(A^{\theta,(s-1)d(d+1)}_{d,1} \circ \mathcal{L}_d \circ A^{\theta,(s-2)d(d+1)}_{d,d} \circ \ldots \circ \mathcal{L}_d \circ A^{\theta,d(d+1)}_{d,d} \circ \mathcal{L}_d \circ A^{\theta,0}_{d,d}\big)(x). \tag{37}$$

The function $\mathbb{U} \colon \mathbb{R}^v \times \mathbb{R}^d \to \mathbb{R}$ in (37) describes an artificial neural network with $s + 1$ layers (1 input layer with d neurons, $s - 1$ hidden layers with d neurons each, and 1 output layer with 1 neuron) and standard logistic functions as activation functions (cf., for instance, Bishop [9]).

3.3 Stochastic Gradient Descent-Type Minimization

As described in Sect. 3.2 for every *suitable* $\theta \in \mathbb{R}^v$ and every $x \in [a, b]^d$ we think of $\mathbb{U}(\theta, x) \in \mathbb{R}$ as an appropriate approximation of $u(T, x) \in \mathbb{R}$. In this subsection we intend to find a *suitable* $\theta \in \mathbb{R}^v$ as an approximate minimizer of

$$\mathbb{R}^v \ni \theta \mapsto \mathbb{E}\big[|\varphi(\mathcal{X}_N) - \mathbb{U}(\theta, \xi)|^2\big] \in \mathbb{R}. \tag{38}$$

To be more specific, we intend to find an approximate minimizer of the function in (38) through a stochastic gradient descent-type minimization algorithm (cf., for instance, Ruder [54, Sect. 4], Jentzen et al. [37], and the references mentioned therein). For this we approximate the derivative of the function in (38) by means of the Monte Carlo method. In this subsection we employ for illustrative reasons as minimization algorithm a stochastic gradient descent minimization scheme with constant learning rate. However, later and, in particular, in the numerical experiments we employ more sophisticated stochastic gradient descent type minimization schemes like, e.g., the Adam optimizer.

More precisely, let $\xi^{(m)} \colon \Omega \to [a, b]^d$, $m \in \mathbb{N}_0$, be i.i.d. continuously uniformly distributed $\mathbb{F}_0/\mathcal{B}([a, b]^d)$-measurable random variables, let $W^{(m)} \colon [0, T] \times \Omega \to \mathbb{R}^d$, $m \in \mathbb{N}_0$, be i.i.d. standard $(\mathbb{F}_t)_{t\in[0,T]}$-Brownian motions, for every $m \in \mathbb{N}_0$ let $\mathcal{X}^{(m)} =$

$(\mathcal{X}_n^{(m)})_{n\in\{0,1,\ldots,N\}}\colon \{0,1,\ldots,N\}\times\Omega\to\mathbb{R}^d$ be the stochastic process which satisfies for every $n\in\{0,1,\ldots,N-1\}$ that $\mathcal{X}_0^{(m)}=\xi^{(m)}$ and

$$\mathcal{X}_{n+1}^{(m)}=\mathcal{X}_n^{(m)}+\mu(\mathcal{X}_n^{(m)})\,(t_{n+1}-t_n)+\sigma(\mathcal{X}_n^{(m)})\,(W_{t_{n+1}}^{(m)}-W_{t_n}^{(m)}),\tag{39}$$

let $\gamma\in(0,\infty)$, and let $\Theta\colon\mathbb{N}_0\times\Omega\to\mathbb{R}^\nu$ be a stochastic process which satisfies for every $m\in\mathbb{N}_0$ that

$$\Theta_{m+1}=\Theta_m-2\gamma\cdot\big(\mathbb{U}(\Theta_m,\xi^{(m)})-\varphi(\mathcal{X}_N^{(m)})\big)\cdot(\nabla_\theta\mathbb{U})(\Theta_m,\xi^{(m)}).\tag{40}$$

Roughly speaking, we think for every sufficiently large $m\in\mathbb{N}$ of the random variable $\Theta_m\colon\Omega\to\mathbb{R}^\nu$ as a suitable approximation of a local minimum point of the function in (38). Moreover, we think for every sufficiently large $m\in\mathbb{N}$ of the random function $[a,b]^d\ni x\mapsto\mathbb{U}(\Theta_n,x)\in\mathbb{R}$ as a suitable approximation of $[a,b]^d\ni x\mapsto u(T,x)\in\mathbb{R}$.

3.4 Description of the Algorithm in a Special Case

In this subsection we give a description of the proposed approximation method in a special case, that is, we describe the proposed approximation method in the specific case where a particular neural network approximation is chosen and where the plain-vanilla stochastic gradient descent method with a constant learning rate is the employed stochastic minimization algorithm (cf. (40) above). For the purpose of readability we describe a special case that abstains from elaborate Machine Learning tools such as the Adam optimizer or batch normalization. For a more general description of the proposed approximation method we refer the reader to Sect. 3.5 below. For a description of the specific implementation used to test the proposed approximation algorithm see Sect. 4 below.

Framework 3.1 *Let $T,\gamma\in(0,\infty)$, $a\in\mathbb{R}$, $b\in(a,\infty)$, $d,N\in\mathbb{N}$, $s\in\{3,4,5,\ldots\}$, let $\nu=sd(d+1)$, let $t_0,t_1,\ldots,t_N\in[0,T]$ satisfy*

$$0=t_0<t_1<\cdots<t_N=T,\tag{41}$$

let $\mu\colon\mathbb{R}^d\to\mathbb{R}^d$ and $\sigma\colon\mathbb{R}^d\to\mathbb{R}^{d\times d}$ be continuous, let $(\Omega,\mathcal{F},\mathbb{P},(\mathbb{F}_t)_{t\in[0,T]})$ be a filtered probability space, let $\xi^{(m)}\colon\Omega\to[a,b]^d$, $m\in\mathbb{N}_0$, be i.i.d. continuously uniformly distributed $\mathbb{F}_0/\mathcal{B}([a,b]^d)$-measurable random variables, let $W^{(m)}\colon[0,T]\times\Omega\to\mathbb{R}^d$, $m\in\mathbb{N}_0$, be i.i.d. standard $(\mathbb{F}_t)_{t\in[0,T]}$-Brownian motions, for every $m\in\mathbb{N}_0$ let $\mathcal{X}^{(m)}\colon\{0,1,\ldots,N\}\times\Omega\to\mathbb{R}^d$ be the stochastic process which satisfies for every $n\in\{0,1,\ldots,N-1\}$ that $\mathcal{X}_0^{(m)}=\xi^{(m)}$ and

$$\mathcal{X}_{n+1}^{(m)}=\mathcal{X}_n^{(m)}+\mu(\mathcal{X}_n^{(m)})\,(t_{n+1}-t_n)+\sigma(\mathcal{X}_n^{(m)})\,(W_{t_{n+1}}^{(m)}-W_{t_n}^{(m)}),\tag{42}$$

let $\mathcal{L}_d\colon\mathbb{R}^d\to\mathbb{R}^d$ satisfy for every $x=(x_1,x_2,\ldots,x_d)\in\mathbb{R}^d$ that

$$\mathcal{L}_d(x)=\left(\frac{\exp(x_1)}{\exp(x_1)+1},\frac{\exp(x_2)}{\exp(x_2)+1},\ldots,\frac{\exp(x_d)}{\exp(x_d)+1}\right),\tag{43}$$

for every $k,l\in\mathbb{N}$, $v\in\mathbb{N}_0=\{0\}\cup\mathbb{N}$, $\theta=(\theta_1,\theta_2,\ldots,\theta_\nu)\in\mathbb{R}^\nu$ with $v+l(k+1)\leq\nu$ let $A_{k,l}^{\theta,v}\colon\mathbb{R}^k\to\mathbb{R}^l$ be as in (36), let $\mathbb{U}\colon\mathbb{R}^\nu\times\mathbb{R}^d\to\mathbb{R}$ satisfy for every $\theta\in\mathbb{R}^\nu$, $x\in\mathbb{R}^d$ that

$$\mathbb{U}(\theta,x)=\big(A_{d,1}^{\theta,(s-1)d(d+1)}\circ\mathcal{L}_d\circ A_{d,d}^{\theta,(s-2)d(d+1)}\circ\ldots\circ\mathcal{L}_d\circ A_{d,d}^{\theta,d(d+1)}\circ\mathcal{L}_d\circ A_{d,d}^{\theta,0}\big)(x),\tag{44}$$

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

$$\Theta_{m+1} = \Theta_m - 2\gamma \cdot \big(\mathbb{U}(\Theta_m, \xi^{(m)}) - \varphi(\mathcal{X}_N^{(m)})\big) \cdot (\nabla_\theta \mathbb{U})(\Theta_m, \xi^{(m)}) \tag{45}$$

Loosely speaking, we think for every sufficiently large $m \in \mathbb{N}$ and every $x \in [a, b]^d$ of the random variable $\mathbb{U}(\Theta_m, x)\colon \Omega \to \mathbb{R}$ in Framework 3.1 as a suitable approximation $\mathbb{U}(\Theta_m, x) \approx u(T, x)$ of the quantity $u(T, x) \in \mathbb{R}$ where $u = u(t, x)_{(t,x)\in[0,T]\times\mathbb{R}^d} \in C^{1,2}([0, T] \times \mathbb{R}^d, \mathbb{R})$ is a function with at most polynomially growing partial derivatives which satisfies for every $t \in [0, T]$, $x \in \mathbb{R}^d$ that $u(0, x) = \varphi(x)$ and

$$\tfrac{\partial u}{\partial t}(t, x) = \tfrac{1}{2} \operatorname{Trace}_{\mathbb{R}^d}\big(\sigma(x)[\sigma(x)]^*(\operatorname{Hess}_x u)(t, x)\big) + \langle \mu(x), (\nabla_x u)(t, x)\rangle_{\mathbb{R}^d} \tag{46}$$

(cf. (4) above).

3.5 Description of the Algorithm in the General Case

In this subsection we provide a general framework which covers the approximation method derived in Sects. 2.1–3.3 above and which allows, in addition, to incorporate other minimization algorithms (cf., for example, Kingma and Ba [39], Ruder [54], E et al. [58], and Han et al. [28]) than just the plain vanilla stochastic gradient descent method. The proposed approximation algorithm is an extension of the approximation algorithm in E et al. [58] and Han et al. [28] in the special case of linear Kolmogorov partial differential equations.

The overall error of the proposed approximation algorithm typically emerges from three different sources: the approximation error, the statistical or generalization error, and the optimization error (cf., for example, [4]). In our situation, the approximation error would measure how well the exact solution of a Kolmogorov partial differential equation may be approximated by neural nets of certain architectures. Several results for the approximation of exact solutions of partial differential equations by neural nets are by now available in the scientific literature (see, e.g., [10,23–25,44] and the references mentioned therein). The generalization error would measure how well the exact distributions of the solutions of stochastic differential equations associated with Kolmogorov partial differential equations are reflected by (approximatively) sampling from the corresponding stochastic differential equations. For results on the generalization error see, for example, [8,15,26,47,57] and the references mentioned therein. The optimization error would measure how close the output of the employed optimization algorithm gets to the exact solution of the approximative optimization problem. The optimization error has been successfully analyzed in the scientific literature in the case of convex objective functions (see, for example, [3,7,37] and the references mentioned therein). The optimization problems which appear in connection with the approximation scheme proposed in this article, however, are nonconvex optimization problems (see, e.g., [13,16,17,45] and the references mentioned therein for first results on nonconvex optimization problems). Nevertheless, analyzing the optimization error in the context of the approximation scheme proposed in this article as well as, more generally, in the context of training neural networks, remains a topic of future research.

Framework 3.2 *Let* $T \in (0, \infty)$, $N, d, \varrho, \nu, \varsigma \in \mathbb{N}$, *let* $H\colon [0, T]^2 \times \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$, $\varphi\colon \mathbb{R}^d \to \mathbb{R}$ *be functions, let* $(\Omega, \mathcal{F}, \mathbb{P}, (\mathbb{F}_t)_{t\in[0,T]})$ *be a filtered probability space, let* $W^{m,j}\colon [0, T] \times \Omega \to \mathbb{R}^d$, $m \in \mathbb{N}_0$, $j \in \mathbb{N}$, *be i.i.d. standard* $(\mathbb{F}_t)_{t\in[0,T]}$*-Brownian motions on* $(\Omega, \mathcal{F}, \mathbb{P})$, *let* $\xi^{m,j}\colon \Omega \to \mathbb{R}^d$, $m \in \mathbb{N}_0$, $j \in \mathbb{N}$, *be i.i.d.* $\mathbb{F}_0/\mathcal{B}(\mathbb{R}^d)$*-measurable random variables, let* $t_0, t_1, \ldots, t_N \in [0, T]$ *satisfy* $0 = t_0 < t_1 < \ldots < t_N = T$, *for every* $\theta \in \mathbb{R}^{\nu}$, $j \in \mathbb{N}$, $\mathbf{s} \in \mathbb{R}^{\varsigma}$ *let* $\mathbb{U}^{\theta,j,\mathbf{s}}\colon \mathbb{R}^d \to \mathbb{R}$ *be a function, for every* $m \in \mathbb{N}_0$, $j \in \mathbb{N}$ *let*

$\mathcal{X}^{m,j} = (\mathcal{X}^{m,j}_n)_{n\in\{0,1,\ldots,N\}} \colon \{0,1,\ldots,N\}\times\Omega \to \mathbb{R}^d$ be a stochastic process which satisfies for every $n \in \{0,1,\ldots,N-1\}$ that $\mathcal{X}^{m,j}_0 = \xi^{m,j}$ and

$$\mathcal{X}^{m,j}_{n+1} = H(t_n, t_{n+1}, \mathcal{X}^{m,j}_n, W^{m,j}_{t_{n+1}} - W^{m,j}_{t_n}), \tag{47}$$

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

$$\phi^{m,\mathbf{s}}(\theta,\omega) = \frac{1}{J_m}\sum_{j=1}^{J_m}\Big[\mathbb{U}^{\theta,j,\mathbf{s}}\big(\xi^{m,j}(\omega)\big) - \varphi\big(\mathcal{X}^{m,j}_N(\omega)\big)\Big]^2, \tag{48}$$

for every $m \in \mathbb{N}_0$, $\mathbf{s} \in \mathbb{R}^{\varsigma}$ let $G^{m,\mathbf{s}} \colon \mathbb{R}^{\nu}\times\Omega \to \mathbb{R}^{\nu}$ satisfy for every $\omega\in\Omega$, $\theta \in \{\eta \in \mathbb{R}^{\nu} \colon \phi^{m,\mathbf{s}}(\cdot,\omega) \colon \mathbb{R}^{\nu} \to \mathbb{R}$ is differentiable at $\eta\}$ that

$$G^{m,\mathbf{s}}(\theta,\omega) = (\nabla_\theta \phi^{m,\mathbf{s}})(\theta,\omega), \tag{49}$$

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

$$\mathbb{S}_{m+1} = \mathcal{S}\big(\mathbb{S}_m, \Theta_m, (\mathcal{X}^{m,i}_N)_{i\in\mathbb{N}}\big), \qquad \Xi_{m+1} = \Psi_m(\Xi_m, G^{m,\mathbb{S}_{m+1}}(\Theta_m)), \tag{50}$$

$$\text{and} \qquad \Theta_{m+1} = \Theta_m - \Phi_m(\Xi_{m+1}). \tag{51}$$

Roughly speaking, we think for every sufficiently large $m\in\mathbb{N}$ and every $x\in[a,b]^d$ of the random variable $\mathbb{U}^{\Theta_m,1,\mathbb{S}_m}(x)\colon\Omega\to\mathbb{R}$ in Framework 3.2 as a suitable approximation $\mathbb{U}^{\Theta_m,1,\mathbb{S}_m}(x) \approx u(T,x)$ of the quantity $u(T,x)\in\mathbb{R}$ where $u = u(t,x)_{(t,x)\in[0,T]\times\mathbb{R}^d} \in C^{1,2}([0,T]\times\mathbb{R}^d,\mathbb{R})$ is a function with at most polynomially growing partial derivatives which satisfies for every $t\in[0,T]$, $x\in\mathbb{R}^d$ that $u(0,x)=\varphi(x)$ and

$$\tfrac{\partial u}{\partial t}(t,x) = \tfrac{1}{2}\,\mathrm{Trace}_{\mathbb{R}^d}\big(\sigma(x)[\sigma(x)]^*(\mathrm{Hess}_x\, u)(t,x)\big) + \langle \mu(x), (\nabla_x u)(t,x)\rangle_{\mathbb{R}^d}, \tag{52}$$

where $\mu\colon\mathbb{R}^d\to\mathbb{R}^d$ and $\sigma\colon\mathbb{R}^d\to\mathbb{R}^{d\times d}$ are sufficiently regular functions (cf. (4) above).

4 Examples

In this section we test the proposed approximation algorithm (see Sect. 2 above) in the case of several examples of SDEs and Kolmogorov PDEs, respectively. In particular, in this section we apply the proposed approximation algorithm to the heat equation (cf. Sect. 4.2 below), to independent geometric Brownian motions (cf. Sect. 4.3 below), to the Black–Scholes model (cf. Sect. 4.4 below), to stochastic Lorenz equations (cf. Sect. 4.5 below), and to the Heston model (cf. Sect. 4.6 below). In the case of each of the examples below we employ the general approximation algorithm in Framework 3.2 above in conjunction with the Adam optimizer (cf. Kingma and Ba [39]) with mini-batches of size 8192 in each iteration step (see Sect. 4.1 below for a precise description). Moreover, we employ a fully-connected feedforward neural network with one input layer, two hidden layers, and one one-dimensional output layer in our implementations in the case of each of these examples. We also use batch normalization (cf. Ioffe and Szegedy [36]) just before the first linear transformation, just before each of the two nonlinear activation functions in front of the hidden layers as well as just after the last linear transformation. For the two nonlinear activation functions we employ the multidimensional version of the function $\mathbb{R}\ni x\mapsto \tanh(x)\in(-1,1)$. All weights in the neural network are

initialized by means of the Xavier initialization (cf. Glorot and Bengio [20]). The runtimes in seconds in Tables 1, 2, 3, 4, 5 and 6 represent average computation times in seconds for our implementations of the proposed approximation method over 5 independent runs (see https://github.com/seb-becker/kolmogorov for more details on the implementation). In particular, the runtimes take the model construction times as well as the training times into account. All computations were performed 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. The underlying system consisted of an Intel Core i7-6800K CPU with 64 GB DDR4-2133 memory running Tensorflow 1.5 on Ubuntu 16.04.

4.1 Setting

Framework 4.1 *Assume Framework 3.2, let* $\varepsilon = 10^{-8}$, $\mathfrak{b}_1 = \frac{9}{10}$, $\mathfrak{b}_2 = \frac{999}{1000}$, $(\gamma_m)_{m\in\mathbb{N}_0} \subseteq (0,\infty)$, *let* $\mathrm{Pow}_r \colon \mathbb{R}^\nu \to \mathbb{R}^\nu$, $r \in (0,\infty)$, *satisfy for every* $r \in (0,\infty)$, $x = (x_1,\ldots,x_\nu) \in \mathbb{R}^\nu$ *that*

$$\mathrm{Pow}_r(x) = (|x_1|^r, \ldots, |x_\nu|^r), \tag{53}$$

assume for every $m \in \mathbb{N}_0$, $i \in \{0,1,\ldots,N\}$ *that* $J_m = 8192$, $t_i = \frac{iT}{N}$, $\varrho = 2\nu$, $T = 1$, $\gamma_m = 10^{-3}\mathbb{1}_{[0,250,000]}(m) + 10^{-4}\mathbb{1}_{(250,000,500,000]}(m) + 10^{-5}\mathbb{1}_{(500,000,\infty)}(m)$, *assume for every* $m \in \mathbb{N}_0$, $x = (x_1,\ldots,x_\nu)$, $y = (y_1,\ldots,y_\nu)$, $\eta = (\eta_1,\ldots,\eta_\nu) \in \mathbb{R}^\nu$ *that*

$$\Psi_m(x,y,\eta) = (\mathfrak{b}_1 x + (1-\mathfrak{b}_1)\eta, \mathfrak{b}_2 y + (1-\mathfrak{b}_2)\,\mathrm{Pow}_2(\eta)) \tag{54}$$

and

$$\Phi_m(x,y) = \left(\left[\sqrt{\tfrac{|y_1|}{1-(\mathfrak{b}_2)^{m+1}}} + \varepsilon\right]^{-1} \frac{\gamma_m x_1}{1-(\mathfrak{b}_1)^{m+1}}, \ldots, \left[\sqrt{\tfrac{|y_\nu|}{1-(\mathfrak{b}_2)^{m+1}}} + \varepsilon\right]^{-1} \frac{\gamma_m x_\nu}{1-(\mathfrak{b}_1)^{m+1}}\right). \tag{55}$$

Equations (54) and (55) in Framework 4.1 describe the Adam optimizer (cf. Kingma and Ba [39], e.g., Han et al. [28, (32)–(33) in Sect. 4.2 and (90)–(91) in Sect. 5.2]). In the setting of Framework 4.1 we present for $D = [0,1]^d$ in Table 1 in Sect. 4.2, for $D = [90,110]^d$ in Table 3 in Sect. 4.3, for $D = [90,110]^d$ in Table 4 in Sect. 4.4, for $D = [\frac{1}{2},\frac{3}{2}] \times [8,10] \times [10,12]$ in Table 5 in Sect. 4.5, and for $D = \times_{i=1}^{25}([90,110] \times [0.02,0.2])$ in Table 6 in Sect. 4.6 statistical estimations of the relative $L^1(|\lambda(D)|^{-1}\lambda_D;\mathbb{R})$-approximation error

$$\frac{1}{\lambda(D)} \int_D \left| \frac{u(T,x) - \mathbb{U}^{\Theta_m,1,\mathbb{S}_m}(x)}{u(T,x)} \right| dx \tag{56}$$

associated to $(\mathbb{U}^{\Theta_m,1,\mathbb{S}_m}(x))_{x\in D}$, of the relative $L^2(|\lambda(D)|^{-1}\lambda_D;\mathbb{R})$-approximation error

$$\sqrt{\frac{1}{\lambda(D)} \int_D \left| \frac{u(T,x) - \mathbb{U}^{\Theta_m,1,\mathbb{S}_m}(x)}{u(T,x)} \right|^2 dx} \tag{57}$$

associated to $(\mathbb{U}^{\Theta_m,1,\mathbb{S}_m}(x))_{x\in D}$, and of the relative $L^\infty(\lambda_D;\mathbb{R})$-approximation error

$$\sup_{x\in D} \left| \frac{u(T,x) - \mathbb{U}^{\Theta_m,1,\mathbb{S}_m}(x)}{u(T,x)} \right| \tag{58}$$

associated to $(\mathbb{U}^{\Theta_m,1,\mathbb{S}_m}(x))_{x\in D}$ against certain values of $m \in \mathbb{N}_0$.

4.2 Heat Equation

In this subsection we apply the proposed approximation algorithm to the heat equation (see (59) below).

Assume Framework 4.1, assume for every $s, t \in [0, T]$, $x, w \in \mathbb{R}^d$, $m \in \mathbb{N}_0$ that $N = 1$, $d = 100$, $\nu = d(2d) + (2d)^2 + 2d = 2d(3d + 1)$, $\varphi(x) = \|x\|^2_{\mathbb{R}^d}$, $H(s, t, x, w) = x + \sqrt{2}\,\mathrm{Id}_{\mathbb{R}^d} w$, assume that $\xi^{0,1} \colon \Omega \to \mathbb{R}^d$ is continuous uniformly distributed on $[0, 1]^d$, and let $u = (u(t, x))_{(t,x)\in[0,T]\times\mathbb{R}^d} \in C^{1,2}([0, T] \times \mathbb{R}^d, \mathbb{R})$ be an at most polynomially growing function which satisfies for every $t \in [0, T]$, $x \in \mathbb{R}^d$ that $u(0, x) = \varphi(x)$ and

$$(\tfrac{\partial u}{\partial t})(t, x) = (\Delta_x u)(t, x). \tag{59}$$

Combining, e.g., Lemma 4.2 below with, e.g., Hairer et al. [27, Corollary 4.17 and Remark 4.1] shows that for every $t \in [0, T]$, $x \in \mathbb{R}^d$ it holds that

$$u(t, x) = \|x\|^2_{\mathbb{R}^d} + 2dt. \tag{60}$$

Table 1 approximately presents the relative $L^1(\lambda_{[0,1]^d}; \mathbb{R})$-approximation error associated to $(\mathbb{U}^{\Theta_m,1,\mathbb{S}_m}(x))_{x\in[0,1]^d}$ (cf. (56) above), the relative $L^2(\lambda_{[0,1]^d}; \mathbb{R})$-approximation error associated to $(\mathbb{U}^{\Theta_m,1,\mathbb{S}_m}(x))_{x\in[0,1]^d}$ (cf. (57) above), and the relative $L^\infty(\lambda_{[0,1]^d}; \mathbb{R})$-approximation error associated to $(\mathbb{U}^{\Theta_m,1,\mathbb{S}_m}(x))_{x\in[0,1]^d}$ (cf. (58) above) against $m \in \{0, 10{,}000, 50{,}000, 100{,}000, 150{,}000, 200{,}000, 500{,}000, 750{,}000\}$. Figure 1 approximately depicts the relative $L^1(\lambda_{[0,1]^d}; \mathbb{R})$-approximation error associated to $(\mathbb{U}^{\Theta_m,1,\mathbb{S}_m}(x))_{x\in[0,1]^d}$ (cf. (56) above), the relative $L^2(\lambda_{[0,1]^d}; \mathbb{R})$-approximation error associated to $(\mathbb{U}^{\Theta_m,1,\mathbb{S}_m}(x))_{x\in[0,1]^d}$ (cf. (57) above), and the relative $L^\infty(\lambda_{[0,1]^d}; \mathbb{R})$-approximation error associated to $(\mathbb{U}^{\Theta_m,1,\mathbb{S}_m}(x))_{x\in[0,1]^d}$ (cf. (58) above) against $m \in \{0, 100, 200, 300, \ldots, 299{,}800, 299{,}900, 300{,}000\}$. In our numerical simulations for Table 1 and Fig. 1 we calculated the exact solution of the PDE (59) by means of Lemma 4.2 below (cf. (60) above) and we approximately calculated the relative $L^1(\lambda_{[0,1]^d}; \mathbb{R})$-approximation error (cf. (56) above), the relative $L^2(\lambda_{[0,1]^d}; \mathbb{R})$-approximation error (cf. (57) above), and the relative $L^\infty(\lambda_{[0,1]^d}; \mathbb{R})$-approximation error (cf. (58) above) for $m \in \{0, 10{,}000, 50{,}000, 100{,}000, 150{,}000, 200{,}000, 500{,}000, 750{,}000\}$ by means of Monte Carlo approximations with 10240000 samples in the case of each one of the above mentioned error criteria (see Lemma 4.3 below). In addition, we present in Table 2 statistical estimations of the relative $L^1(\mathbb{P}; L^1(\lambda_{[0,1]^d}; \mathbb{R}))$-approximation error associated to $(\mathbb{U}^{\Theta_m,1,\mathbb{S}_m(x)})_{x\in[0,1]^d}$ (cf. (61) below), the relative $L^2(\mathbb{P}; L^2(\lambda_{[0,1]^d}; \mathbb{R}))$-approximation error associated to $(\mathbb{U}^{\Theta_m,1,\mathbb{S}_m(x)})_{x\in[0,1]^d}$ (cf. (62) below), and the relative $L^2(\mathbb{P}; L^\infty(\lambda_{[0,1]^d}; \mathbb{R}))$-approximation error associated to $(\mathbb{U}^{\Theta_m,1,\mathbb{S}_m(x)})_{x\in[0,1]^d}$ (cf. (63) below) against $m \in \{0, 10{,}000, 50{,}000, 100{,}000, 150{,}000, 200{,}000, 500{,}000, 750{,}000\}$. In our numerical simulations for Table 2 we calculated the exact solution of the PDE (59) by means of Lemma 4.2 below (cf. (60) above), we approximately calculated the relative $L^1(\mathbb{P}; L^1(\lambda_{[0,1]^d}; \mathbb{R}))$-approximation error

$$\mathbb{E}\bigg[\int_{[0,1]^d} \bigg|\frac{u(T, x) - \mathbb{U}^{\Theta_m,1,\mathbb{S}_m}(x)}{u(T, x)}\bigg| \, dx\bigg] \tag{61}$$

for $m \in \{0, 10{,}000, 50{,}000, 100{,}000, 150{,}000, 200{,}000, 500{,}000, 750{,}000\}$ by means of Monte Carlo approximations with 10240000 samples for the Lebesgue integral and 5 samples for the expectation, we approximately calculated the relative $L^2(\mathbb{P}; L^2(\lambda_{[0,1]^d}; \mathbb{R}))$-appro-

Table 1 Approximative presentations of the relative approximation errors in (56)–(58) for the heat equation in (59)

Number of steps	Relative $L^1(\lambda_{[0,1]^d};\mathbb{R})$-error	Relative $L^2(\lambda_{[0,1]^d};\mathbb{R})$-error	Relative $L^\infty(\lambda_{[0,1]^d};\mathbb{R})$-error	Runtime in seconds
0	0.998253	0.998254	1.003524	0.5
10,000	0.957464	0.957536	0.993083	44.6
50,000	0.786743	0.786806	0.828184	220.8
100,000	0.574013	0.574060	0.605283	440.8
150,000	0.361564	0.361594	0.384105	661.0
200,000	0.150346	0.150362	0.164140	880.8
500,000	0.000882	0.001112	0.007360	2200.7
750,000	0.000822	0.001036	0.007423	3300.6

ximation error

$$\left(\mathbb{E}\left[\int_{[0,1]^d}\left|\frac{u(T,x)-\mathbb{U}^{\Theta_m,1,\mathbb{S}_m}(x)}{u(T,x)}\right|^2 dx\right]\right)^{1/2} \tag{62}$$

for $m \in \{0, 10{,}000, 50{,}000, 100{,}000, 150{,}000, 200{,}000, 500{,}000, 750{,}000\}$ by means of Monte Carlo approximations with 10240000 samples for the Lebesgue integral and 5 samples for the expectation, and we approximately calculated the relative $L^2(\mathbb{P}; L^\infty(\lambda_{[0,1]^d};\mathbb{R}))$-approximation error

$$\left(\mathbb{E}\left[\sup_{x\in[0,1]^d}\left|\frac{u(T,x)-\mathbb{U}^{\Theta_m,1,\mathbb{S}_m}(x)}{u(T,x)}\right|^2\right]\right)^{1/2} \tag{63}$$

for $m \in \{0, 10{,}000, 50{,}000, 100{,}000, 150{,}000, 200{,}000, 500{,}000, 750{,}000\}$ by means of Monte Carlo approximations with 10240000 samples for the supremum (see Lemma 4.3 below) and 5 samples for the expectation. The following elementary result, Lemma 4.2 below, specifies the explicit solution of the PDE (59) above (cf. (60) above). For completeness we also provide here a proof for Lemma 4.2.

Lemma 4.2 *Let $T \in (0,\infty)$, $d \in \mathbb{N}$, let $C \in \mathbb{R}^{d\times d}$ be a strictly positive and symmetric matrix, and let $u\colon [0,T]\times\mathbb{R}^d \to \mathbb{R}$ satisfy for every $(t,x)\in[0,T]\times\mathbb{R}^d$ that*

$$u(t,x) = \|x\|^2_{\mathbb{R}^d} + t\,\mathrm{Trace}_{\mathbb{R}^d}(C). \tag{64}$$

Then

(i) *it holds that $u \in C^\infty([0,T]\times\mathbb{R}^d,\mathbb{R})$ is at most polynomially growing and*
(ii) *it holds for every $t\in[0,T]$, $x\in\mathbb{R}^d$ that*

$$(\tfrac{\partial u}{\partial t})(t,x) = \tfrac{1}{2}\,\mathrm{Trace}_{\mathbb{R}^d}\big(C(\mathrm{Hess}_x\,u)(t,x)\big). \tag{65}$$

Proof of Lemma 4.2 First, note that u is a polynomial. This establishes item (i). Moreover, note that for every $(t,x)\in[0,T]\times\mathbb{R}^d$ it holds that

$$(\tfrac{\partial u}{\partial t})(t,x) = \mathrm{Trace}_{\mathbb{R}^d}(C), \qquad (\nabla_x u)(t,x) = 2x, \tag{66}$$

$$\text{and} \qquad (\mathrm{Hess}_x\,u)(t,x) = \big(\tfrac{\partial}{\partial x}(\nabla_x u)\big)(t,x) = 2\,\mathrm{Id}_{\mathbb{R}^d}. \tag{67}$$

Table 2 Approximative presentations of the relative approximation errors in (61)–(63) for the heat equation in (59)

Number of steps	Relative $L^1(\mathbb{P}; L^1(\lambda_{[0,1]^d}; \mathbb{R}))$-error	Relative $L^2(\mathbb{P}; L^2(\lambda_{[0,1]^d}; \mathbb{R}))$-error	Relative $L^2(\mathbb{P}; L^\infty(\lambda_{[0,1]^d}; \mathbb{R}))$-error	Mean runtime in seconds
0	1.000310	1.000311	1.005674	0.6
10,000	0.957481	0.957554	0.993097	44.7
50,000	0.786628	0.786690	0.828816	220.4
100,000	0.573867	0.573914	0.605587	440.5
150,000	0.361338	0.361369	0.382967	660.8
200,000	0.001387	0.001741	0.010896	880.9
500,000	0.000883	0.001112	0.008017	2201.0
750,000	0.000822	0.001038	0.007547	3300.4

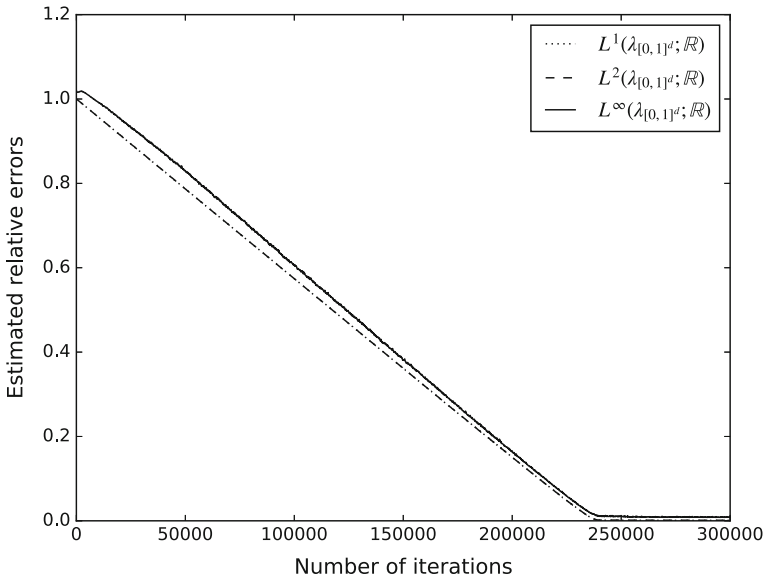

Fig. 1 Approximative plots of the relative approximation errors in (56)–(58) for the heat equation in (59)

Hence, we obtain for every $(t, x) \in [0, T] \times \mathbb{R}^d$ that

$$(\tfrac{\partial u}{\partial t})(t,x) - \tfrac{1}{2}\,\mathrm{Trace}_{\mathbb{R}^d}\big(C(\mathrm{Hess}_x\, u)(t,x)\big) = \mathrm{Trace}_{\mathbb{R}^d}(C) - \tfrac{1}{2}\,\mathrm{Trace}_{\mathbb{R}^d}(2C) = 0. \quad (68)$$

This proves item (ii). The proof of Lemma 4.2 is thus completed. □

Lemma 4.3 below discloses the strategy how we approximatively calculate the $L^\infty(\lambda_{[0,1]^d}; \mathbb{R})$-errors in (58) and (63) above. For completeness, we provide a proof of Lemma 4.3.

Lemma 4.3 *Let $d \in \mathbb{N}$, $a \in \mathbb{R}$, $b \in (a, \infty)$, let $f\colon [a,b]^d \to \mathbb{R}$ be continuous, let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, let $X_n\colon \Omega \to [a,b]^d$, $n \in \mathbb{N}$, be i.i.d. random variables, and assume that X_1 is continuous uniformly distributed on $[a,b]^d$. Then*

(i) *it holds that*

$$\mathbb{P}\left(\limsup_{N\to\infty}\left|\left[\max_{1\le n\le N} f(X_n)\right] - \left[\sup_{x\in[a,b]^d} f(x)\right]\right| = 0\right) = 1 \quad (69)$$

and

(ii) *it holds for every $p \in (0,\infty)$ that*

$$\limsup_{N\to\infty}\mathbb{E}\left[\left|\left[\max_{1\le n\le N} f(X_n)\right] - \left[\sup_{x\in[a,b]^d} f(x)\right]\right|^p\right] = 0. \quad (70)$$

Proof of Lemma 4.3 First, observe that the fact that $f\colon [a,b]^d \to \mathbb{R}$ is continuous and the fact that $[a,b]^d \subseteq \mathbb{R}^d$ is compact demonstrate that

(I) there exists $\xi \in [a,b]^d$ which satisfies that $f(\xi) = \sup_{x\in[a,b]^d} f(x)$,

(II) there exists $\varepsilon\colon (0,\infty) \to (0,\infty)$ which satisfies that for all $\delta \in (0,\infty)$, $x \in [a,b]^d$ with $\|x - \xi\|_{\mathbb{R}^d} < \delta$ it holds that $|f(x) - f(\xi)| \le \varepsilon(\delta)$, and

(III) there exists a random variable $Y\colon \Omega \to \mathbb{R}$ which satisfies that

$$\mathbb{P}\left(\limsup_{N\to\infty}\left|\left[\max_{1\leq n\leq N} f(X_n)\right] - Y\right| = 0\right) = 1. \tag{71}$$

Hence, we obtain for all $\delta \in (0,\infty)$ that

$$\begin{aligned}\mathbb{P}\left(\sup_{x\in[a,b]^d} f(x) - Y \geq \delta\right) &\leq \mathbb{P}\left(\bigcap_{n\in\mathbb{N}}\left\{\|X_n - \xi\|_{\mathbb{R}^d} \geq \varepsilon(\delta)\right\}\right)\\ &\leq \limsup_{n\to\infty}\left[1 - \left(\frac{2\varepsilon(\delta)}{\sqrt{d}(b-a)}\right)^d\right]^n = 0.\end{aligned} \tag{72}$$

This establishes item (i). The fact that $f\colon [a,b]^d \to \mathbb{R}$ is globally bounded, item (i), and Lebesgue's dominated convergence theorem ensure that for every $p \in (0,\infty)$ it holds that

$$\limsup_{N\to\infty} \mathbb{E}\left[\left|\max\nolimits_{1\leq i\leq N} f(X_i) - \sup\nolimits_{x\in[a,b]^d} f(x)\right|^p\right] = 0. \tag{73}$$

This establishes item (ii). The proof of Lemma 4.3 is thus completed. □

4.3 Geometric Brownian Motions

In this subsection we apply the proposed approximation algorithm to a Black–Scholes PDE with independent underlying geometric Brownian motions.

Assume Framework 4.1, let[1] $r = \frac{1}{20}$, $\delta = \frac{1}{10}$, $\mu = r - \delta = -\frac{1}{20}$, $\sigma_1 = \frac{1}{10} + \frac{1}{200}$, $\sigma_2 = \frac{1}{10} + \frac{2}{200}, \ldots, \sigma_{100} = \frac{1}{10} + \frac{100}{200}$, assume for every $s, t \in [0,T], x = (x_1, x_2, \ldots, x_d), w = (w_1, w_2, \ldots, w_d) \in \mathbb{R}^d, m \in \mathbb{N}_0$ that $d = 100, \varphi(x) = \exp(-rT)\max\{[\max_{i\in\{1,2,\ldots,d\}} x_i] - 100, 0\}$, $N = 1$, and

$$\begin{aligned}H(s,t,x,w) = \Big(&x_1 \exp\big(\big(\mu - \tfrac{|\sigma_1|^2}{2}\big)(t-s) + \sigma_1 w_1\big), \ldots,\\ &x_d \exp\big(\big(\mu - \tfrac{|\sigma_d|^2}{2}\big)(t-s) + \sigma_d w_d\big)\Big),\end{aligned} \tag{74}$$

assume that $\xi^{0,1}\colon \Omega \to \mathbb{R}^d$ is continuous uniformly distributed on $[90,110]^d$, and let $u = (u(t,x))_{t\in[0,T], x\in\mathbb{R}^d} \in C([0,T]\times\mathbb{R}^d, \mathbb{R})$ be an at most polynomially growing function which satisfies for every $t \in (0,T], x \in \mathbb{R}^d$ that $u(0,x) = \varphi(x), u|_{(0,T]\times\mathbb{R}^d} \in C^{1,2}((0,T]\times\mathbb{R}^d, \mathbb{R})$, and

$$\big(\tfrac{\partial u}{\partial t}\big)(t,x) = \tfrac{1}{2}\sum_{i=1}^d |\sigma_i x_i|^2 \big(\tfrac{\partial^2 u}{\partial x_i^2}\big)(t,x) + \mu \sum_{i=1}^d x_i \big(\tfrac{\partial u}{\partial x_i}\big)(t,x). \tag{75}$$

The Feynman–Kac formula (cf., for example, Hairer et al. [27, Corollary 4.17]) shows that for every standard Brownian motion $\mathcal{W} = (\mathcal{W}^{(1)}, \ldots, \mathcal{W}^{(d)})\colon [0,T]\times\Omega \to \mathbb{R}^d$ and every $t \in [0,T], x = (x_1, \ldots, x_d) \in \mathbb{R}^d$ it holds that

$$u(t,x) = \mathbb{E}\left[\varphi\left(x_1 \exp\left(\sigma_1 \mathcal{W}_t^{(1)} + \left(\mu - \tfrac{|\sigma_1|^2}{2}\right)t\right), \ldots, x_d \exp\left(\sigma_d \mathcal{W}_t^{(d)} + \left(\mu - \tfrac{|\sigma_d|^2}{2}\right)t\right)\right)\right]. \tag{76}$$

[1] The parameter r models a riskless interest rate and the parameter δ models a continuous dividend payment. For simplicity we assumed that every stock has the same dividend rate.

Table 3 approximately presents the relative $L^1(20^{-d}\lambda_{[90,110]^d};\mathbb{R})$-approximation error associated to $(\mathbb{U}^{\Theta_m,1,\mathbb{S}_m}(x))_{x\in[90,110]^d}$ (cf. (56) above), the relative $L^2(20^{-d}\lambda_{[90,110]^d};\mathbb{R})$-approximation error associated to $(\mathbb{U}^{\Theta_m,1,\mathbb{S}_m}(x))_{x\in[90,110]^d}$ (cf. (57) above), and the relative $L^\infty(\lambda_{[90,110]^d};\mathbb{R})$-approximation error associated to $(\mathbb{U}^{\Theta_m,1,\mathbb{S}_m}(x))_{x\in[90,110]^d}$ (cf. (58) above) against $m \in \{0, 25{,}000, 50{,}000, 100{,}000, 150{,}000, 250{,}000, 500{,}000, 750{,}000\}$. In our numerical simulations for Table 3 we approximately calculated the exact solution of the PDE (75) by means of (76) and Monte Carlo approximations with 1048576 samples and we approximately calculated the relative $L^1(20^{-d}\lambda_{[90,110]^d};\mathbb{R})$-approximation error (cf. (56) above), the relative $L^2(20^{-d}\lambda_{[90,110]^d};\mathbb{R})$-approximation error (cf. (57) above), and the relative $L^\infty(\lambda_{[90,110]^d};\mathbb{R})$-approximation error (cf. (58) above) for $m \in \{0, 25{,}000, 50{,}000, 100{,}000, 150{,}000, 250{,}000, 500{,}000, 750{,}000\}$ by means of Monte Carlo approximations with 81920 samples in the case of each one of the above mentioned error criteria (see Lemma 4.3 above).

4.4 Black–Scholes Model with Correlated Noise

In this subsection we apply the proposed approximation algorithm to a Black–Scholes PDE with correlated noise.

Assume Framework 4.1, let $r = \frac{1}{20}$, $\delta = \frac{1}{10}$, $\mu = r - \delta = -\frac{1}{20}$, $\beta_1 = \frac{1}{10} + \frac{1}{200}$, $\beta_2 = \frac{1}{10} + \frac{2}{200}$, …, $\beta_{100} = \frac{1}{10} + \frac{100}{200}$, $Q = (Q_{i,j})_{(i,j)\in\{1,2,\ldots,100\}}$, $\Sigma = (\Sigma_{i,j})_{(i,j)\in\{1,2,\ldots,100\}} \in \mathbb{R}^{100\times 100}$, $\varsigma_1, \varsigma_2, \ldots, \varsigma_{100} \in \mathbb{R}^{100}$, assume for every $s, t \in [0,T]$, $x = (x_1, x_2, \ldots, x_d)$, $w = (w_1, w_2, \ldots, w_d) \in \mathbb{R}^d$, $m \in \mathbb{N}_0$, $i, j, k \in \{1, 2, \ldots, 100\}$ with $i < j$ that $N = 1$, $d = 100$, $\nu = d(2d) + (2d)^2 + 2d = 2d(3d+1)$, $Q_{k,k} = 1$, $Q_{i,j} = Q_{j,i} = \frac{1}{2}$, $\Sigma_{i,j} = 0$, $\Sigma_{k,k} > 0$, $\Sigma\Sigma^* = Q$ (cf., for example, Golub and Van Loan [21, Theorem 4.2.5]), $\varsigma_k = (\beta_k\Sigma_{k,1}, \beta_k\Sigma_{k,2}, \ldots, \beta_k\Sigma_{k,100})$, $\varphi(x) = \exp(-\mu T)\max\{110 - [\min_{i\in\{1,2,\ldots,d\}} x_i], 0\}$, and

$$\begin{aligned} H(s,t,x,w) = \Big(&x_1 \exp\Big((\mu - \tfrac{1}{2}\|\varsigma_1\|^2_{\mathbb{R}^d})(t-s) + \langle \varsigma_1, w\rangle_{\mathbb{R}^d}\Big), \ldots, \\ &x_d \exp\Big((\mu - \tfrac{1}{2}\|\varsigma_d\|^2_{\mathbb{R}^d})(t-s) + \langle \varsigma_d, w\rangle_{\mathbb{R}^d}\Big)\Big), \end{aligned} \tag{77}$$

assume that $\xi^{0,1}\colon \Omega \to \mathbb{R}^d$ is continuous uniformly distributed on $[90, 110]^d$, and let $u = (u(t,x))_{t\in[0,T],x\in\mathbb{R}^d} \in C([0,T]\times\mathbb{R}^d, \mathbb{R})$ be an at most polynomially growing continuous function which satisfies for every $t \in (0,T]$, $x \in \mathbb{R}^d$ that $u(0,x) = \varphi(x)$, $u|_{(0,T]\times\mathbb{R}^d} \in C^{1,2}((0,T]\times\mathbb{R}^d, \mathbb{R})$, and

$$(\tfrac{\partial u}{\partial t})(t,x) = \tfrac{1}{2}\sum_{i,j=1}^{d} x_i x_j \langle \varsigma_i, \varsigma_j\rangle_{\mathbb{R}^d} (\tfrac{\partial^2 u}{\partial x_i x_j})(t,x) + \mu \sum_{i=1}^{d} x_i (\tfrac{\partial u}{\partial x_i})(t,x). \tag{78}$$

The Feynman–Kac formula (cf., for example, Hairer et al. [27, Corollary 4.17]) shows that for every standard Brownian motion $\mathcal{W} = (\mathcal{W}^{(1)}, \ldots, \mathcal{W}^{(d)})\colon [0,T]\times\Omega \to \mathbb{R}^d$ and every $t \in [0,T]$, $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$ it holds that

$$\begin{aligned} u(t,x) = \mathbb{E}\Big[\varphi\Big(&x_1 \exp\Big(\langle \varsigma_1, \mathcal{W}_t\rangle_{\mathbb{R}^d} + \Big(\mu - \tfrac{\|\varsigma_1\|^2_{\mathbb{R}^d}}{2}\Big)t\Big), \ldots, \\ &x_d \exp\Big(\langle \varsigma_d, \mathcal{W}_t\rangle_{\mathbb{R}^d} + \Big(\mu - \tfrac{\|\varsigma_d\|^2_{\mathbb{R}^d}}{2}\Big)t\Big)\Big)\Big]. \end{aligned} \tag{79}$$

Table 3 Approximative presentations of the relative approximation errors in (56)–(58) for the Black–Scholes PDE with independent underlying geometric Brownian motions in (75)

Number of steps	Relative $L^1(20^{-d}\lambda_{[90,110]^d};\mathbb{R})$-error	Relative $L^2(20^{-d}\lambda_{[90,119]^d};\mathbb{R})$-error	Relative $L^\infty(\lambda_{[90,110]^d};\mathbb{R})$-error	Runtime in seconds
0	1.004285	1.004286	1.009524	1
25,000	0.842938	0.843021	0.87884	110.2
50,000	0.684955	0.685021	0.719826	219.5
100,000	0.371515	0.371551	0.387978	437.9
150,000	0.064605	0.064628	0.072259	656.2
250,000	0.001220	0.001538	0.010039	1092.6
500,000	0.000949	0.001187	0.005105	2183.8
750,000	0.000902	0.001129	0.006028	3275.1

Table 4 approximately presents the relative $L^1(20^{-d}\lambda_{[90,110]^d};\mathbb{R})$-approximation error associated to $(\mathbb{U}^{\Theta_m,1,\mathbb{S}_m}(x))_{x\in[90,110]^d}$ (cf. (56) above), the relative $L^2(20^{-d}\lambda_{[90,110]^d};\mathbb{R})$-approximation error associated to $(\mathbb{U}^{\Theta_m,1,\mathbb{S}_m}(x))_{x\in[90,110]^d}$ (cf. (57) above), and the relative $L^\infty(\lambda_{[90,110]^d};\mathbb{R})$-approximation error associated to $(\mathbb{U}^{\Theta_m,1,\mathbb{S}_m}(x))_{x\in[90,110]^d}$ (cf. (58) above) against $m \in \{0, 25{,}000, 50{,}000, 100{,}000, 150{,}000, 250{,}000, 500{,}000, 750{,}000\}$. In our numerical simulations for Table 4 we approximately calculated the exact solution of the PDE (78) by means of (79) and Monte Carlo approximations with 1048576 samples and we approximately calculated the relative $L^1(20^{-d}\lambda_{[90,110]^d};\mathbb{R})$-approximation error (cf. (56) above), the relative $L^2(20^{-d}\lambda_{[90,110]^d};\mathbb{R})$-approximation error (cf. (57) above), and the relative $L^\infty(\lambda_{[90,110]^d};\mathbb{R})$-approximation error (cf. (58) above) for $m \in \{0, 25{,}000, 50{,}000, 100{,}000, 150{,}000, 250{,}000, 500{,}000, 750{,}000\}$ by means of Monte Carlo approximations with 81920 samples in the case of each one of the above mentioned error criteria (see Lemma 4.3 above).

4.5 Stochastic Lorenz Equations

In this subsection we apply the proposed approximation algorithm to the stochastic Lorenz equation.

Assume Framework 4.1, let $\alpha_1 = 10, \alpha_2 = 14, \alpha_3 = \frac{8}{3}, \beta = \frac{3}{20}, D = [\frac{1}{2}, \frac{3}{2}] \times [8, 10] \times [10, 12]$, let $\mu\colon \mathbb{R}^d \to \mathbb{R}^d$ be a function, assume for every $s, t \in [0, T]$, $x = (x_1, x_2, \ldots, x_d)$, $w = (w_1, w_2, \ldots, w_d) \in \mathbb{R}^d$, $m \in \mathbb{N}_0$ that $N = 100$, $d = 3$, $\nu = (d + 20)d + (d + 20)^2 + (d + 20) = (d + 20)(2d + 21)$, $\mu(x) = (\alpha_1(x_2 - x_1), \alpha_2 x_1 - x_2 - x_1 x_3, x_1 x_2 - \alpha_3 x_3)$, $\varphi(x) = \|x\|^2_{\mathbb{R}^d}$, and

$$H(s, t, x, w) = x + \mu(x)(t - s)\mathbb{1}_{[0,N/T]}(\|\mu(x)\|_{\mathbb{R}^d}) + \beta w \tag{80}$$

(cf., for example, Hutzenthaler et al. [32], Hutzenthaler et al. [33], Hutzenthaler et al. [35], Milstein and Tretyakov [50], Sabanis [55,56], and the references mentioned therein for related temporal numerical approximation schemes for SDEs), assume that $\xi^{0,1}\colon \Omega \to \mathbb{R}^d$ is continuous uniformly distributed on D, and let $u = (u(t, x))_{(t,x)\in[0,T]\times\mathbb{R}^d} \in C^{1,2}([0, T] \times \mathbb{R}^d, \mathbb{R})$ be an at most polynomially growing function (cf., for example, Hairer et al. [27, Corollary 4.17] and Hörmander [31, Theorem 1.1]) which satisfies for every $t \in [0, T]$, $x \in \mathbb{R}^d$ that $u(0, x) = \varphi(x)$ and

$$\begin{aligned}(\tfrac{\partial u}{\partial t})(t, x) &= \tfrac{\beta^2}{2}(\Delta_x u)(t, x) + \alpha_1(x_2 - x_1)(\tfrac{\partial u}{\partial x_1})(t, x) \\ &\quad + (\alpha_2 x_1 - x_2 - x_1 x_3)(\tfrac{\partial u}{\partial x_2})(t, x) + (x_1 x_2 - \alpha_3 x_3)(\tfrac{\partial u}{\partial x_3})(t, x).\end{aligned} \tag{81}$$

Table 5 approximately presents the relative $L^1(4^{-1}\lambda_D;\mathbb{R})$-approximation error associated to $(\mathbb{U}^{\Theta_m,1,\mathbb{S}_m}(x))_{x\in D}$ (cf. (56) above), the relative $L^2(4^{-1}\lambda_D;\mathbb{R})$-approximation error associated to $(\mathbb{U}^{\Theta_m,1,\mathbb{S}_m}(x))_{x\in D}$ (cf. (57) above), and the relative $L^\infty(\lambda_D;\mathbb{R})$-approximation error associated to $(\mathbb{U}^{\Theta_m,1,\mathbb{S}_m}(x))_{x\in D}$ (cf. (58) above) against $m \in \{0, 25{,}000, 50{,}000, 100{,}000, 150{,}000, 250{,}000, 500{,}000, 750{,}000\}$. In our numerical simulations for Table 5 we approximately calculated the exact solution of the PDE (81) by means of Monte Carlo approximations with 1048576 samples and temporal SDE-discretizations based on (80) with 100 equidistant time steps and we approximately calculated the relative $L^1(4^{-1}\lambda_D;\mathbb{R})$-approximation error (cf. (56) above), the relative $L^2(4^{-1}\lambda_D;\mathbb{R})$-approximation error (cf. (57) above), and the relative $L^\infty(\lambda_D;\mathbb{R})$-approximation error (cf. (58) above) for $m \in \{0, 25{,}000, 50{,}000, 100{,}000, 150{,}000, 250{,}000, 500{,}000, 750{,}000\}$ by means of Monte Carlo approximations

Table 4 Approximative presentations of the relative approximation errors in (56)–(58) for the Black–Scholes PDE with correlated noise in (78)

Number of steps	Relative $L^1(20^{-d}\lambda_{[90,110]^d};\mathbb{R})$ -error	Relative $L^2(20^{-d}\lambda_{[90,110]^d};\mathbb{R})$ -error	Relative $L^\infty(\lambda_{[90,110]^d};\mathbb{R})$ -error	Runtime in seconds
0	1.003383	1.003385	1.011662	0.8
25,000	0.631420	0.631429	0.640633	112.1
50,000	0.269053	0.269058	0.275114	223.3
100,000	0.000752	0.000948	0.00553	445.8
150,000	0.000694	0.00087	0.004662	668.2
250,000	0.000604	0.000758	0.006483	1119.3
500,000	0.000493	0.000615	0.002774	2292.8
750,000	0.000471	0.00059	0.002862	3466.8

with 20480 samples in the case of each one of the above mentioned error criteria (see Lemma 4.3 above).

4.6 Heston Model

In this subsection we apply the proposed approximation algorithm to the Heston model in (83) below.

Assume Framework 4.1, let $\delta = 25$, $\alpha = \frac{1}{20}$, $\kappa = \frac{6}{10}$, $\theta = \frac{1}{25}$, $\beta = \frac{1}{5}$, $\varrho = -\frac{4}{5}$, $D = \times_{i=1}^{\delta}\big([90, 110] \times [0.02, 0.2]\big)$, let $e_i \in \mathbb{R}^{50}$, $i \in \{1, 2, \ldots, 50\}$, satisfy that $e_1 = (1, 0, 0, \ldots, 0, 0) \in \mathbb{R}^{50}$, $e_2 = (0, 1, 0, \ldots, 0, 0) \in \mathbb{R}^{50}$, ..., $e_{50} = (0, 0, 0, \ldots, 0, 1) \in \mathbb{R}^{50}$, assume for every $s, t \in [0, T]$, $x = (x_1, x_2, \ldots, x_d)$, $w = (w_1, w_2, \ldots, w_d) \in \mathbb{R}^d$ that $N = 100$, $d = 2\delta = 50$, $\nu = (d + 50)d + (d + 50)^2 + (d + 50) = (d + 50)(2d + 51)$, $\varphi(x) = \exp(-\alpha T)\max\{110 - [\sum_{i=1}^{\delta} \frac{x_{2i-1}}{\delta}], 0\}$, and

$$
\begin{aligned}
H(s,t,x,w) = \sum_{i=1}^{\delta} \Bigg(& \Big[x_{2i-1} \exp\Big((\alpha - \tfrac{x_{2i}}{2})(t-s) + w_{2i-1}\sqrt{x_{2i}} \Big) \Big] e_{2i-1} \\
& + \Big[\max\Big\{ \Big[\max\big\{ \tfrac{\beta}{2}\sqrt{t-s}, \max\{ \tfrac{\beta}{2}\sqrt{t-s}, \sqrt{x_{2i}} \} + \tfrac{\beta}{2}\big(\rho w_{2i-1} + [1-\rho^2]^{1/2} w_{2i}\big) \big\} \Big]^2 \\
& + \big(\kappa\theta - \tfrac{\beta^2}{4} - \kappa x_{2i}\big)(t-s), 0 \Big\} \Big] e_{2i} \Bigg)
\end{aligned} \tag{82}
$$

(cf. Hefter and Herzwurm [29, Sect. 1]), assume that $\xi^{0,1} \colon \Omega \to \mathbb{R}^d$ is continuous uniformly distributed on D, and let $u = (u(t,x))_{t\in[0,T], x\in\mathbb{R}^d} \in C([0,T] \times \mathbb{R}^d, \mathbb{R})$ be an at most polynomially growing function (cf., for example, Alfonsi [1, Proposition 4.1]) which satisfies for every $t \in (0, T]$, $x \in \mathbb{R}^d$ that $u(0, x) = \varphi(x)$, $u|_{(0,T]\times\mathbb{R}^d} \in C^{1,2}((0,T] \times \mathbb{R}^d, \mathbb{R})$, and

$$
\begin{aligned}
(\tfrac{\partial u}{\partial t})(t,x) = & \Bigg[\sum_{i=1}^{\delta} \Big(\alpha x_{2i-1} (\tfrac{\partial u}{\partial x_{2i-1}})(t,x) + \kappa(\theta - x_{2i})(\tfrac{\partial u}{\partial x_{2i}})(t,x) \Big) \Bigg] \\
& + \Bigg[\sum_{i=1}^{\delta} \frac{|x_{2i}|}{2} \Big(|x_{2i-1}|^2 (\tfrac{\partial^2 u}{\partial x_{2i-1}^2})(t,x) + 2x_{2i-1}\beta\varrho (\tfrac{\partial^2 u}{\partial x_{2i-1}\partial x_{2i}})(t,x) + \beta^2 (\tfrac{\partial^2 u}{\partial x_{2i}^2})(t,x) \Big) \Bigg].
\end{aligned} \tag{83}
$$

Table 6 approximately presents the relative $L^1(|\lambda(D)|^{-1}\lambda_D; \mathbb{R})$-approximation error associated to $(\mathbb{U}^{\Theta_m,1,\mathbb{S}_m}(x))_{x\in D}$ (cf. (56) above), the relative $L^2(|\lambda(D)|^{-1}\lambda_D; \mathbb{R})$-approximation error associated to $(\mathbb{U}^{\Theta_m,1,\mathbb{S}_m}(x))_{x\in D}$ (cf. (57) above), and the relative $L^\infty(\lambda_D; \mathbb{R})$-approximation error associated to $(\mathbb{U}^{\Theta_m,1,\mathbb{S}_m}(x))_{x\in D}$ (cf. (58) above) against $m \in \{0, 25{,}000, 50{,}000, 100{,}000, 150{,}000, 250{,}000, 500{,}000, 750{,}000\}$. In our numerical simulations for Table 6 we approximately calculated the exact solution of the PDE (83) by means of Monte Carlo approximations with 1048576 samples and temporal SDE-discretizations based on (82) with 100 equidistant time steps and we approximately calculated the relative $L^1(|\lambda(D)|^{-1}\lambda_D; \mathbb{R})$-approximation error (cf. (56) above), the relative $L^2(|\lambda(D)|^{-1}\lambda_D; \mathbb{R})$-approximation error (cf. (57) above), and the relative $L^\infty(\lambda_D; \mathbb{R})$-approximation error (cf. (58) above) for $m \in \{0, 25{,}000, 50{,}000, 100{,}000, 150{,}000, 250{,}000, 500{,}000, 750{,}000\}$ by means of Monte Carlo approximations with 10240 samples in the case of each one of the above mentioned error criteria (see Lemma 4.3 above).

Table 5 Approximative presentations of the relative approximation errors in (56)–(58) for the stochastic Lorenz equation in (81)

Number of steps	Relative $L^1(4^{-1}\lambda_D;\mathbb{R})$-error	Relative $L^2(4^{-1}\lambda_D;\mathbb{R})$-error	Relative $L^\infty(\lambda_D;\mathbb{R})$-error	Runtime in seconds
0	0.995732	0.995732	0.996454	1.0
25,000	0.905267	0.909422	1.247772	750.1
50,000	0.801935	0.805497	1.115690	1461.7
100,000	0.599847	0.602630	0.823042	2932.1
150,000	0.392394	0.394204	0.542209	4423.3
250,000	0.000732	0.000811	0.002865	7327.9
500,000	0.000312	0.000365	0.003158	14,753.0
750,000	0.000187	0.000229	0.001264	21,987.4

Table 6 Approximative presentations of the relative approximation errors in (56)–(58) for the Heston model in (83)

Number of steps	Relative $L^1(\lvert\lambda(D)\rvert^{-1}\lambda_D;\mathbb{R})$-error	Relative $L^2(\lvert\lambda(D)\rvert^{-1}\lambda_D;\mathbb{R})$-error	Relative $L^\infty(\lambda_D;\mathbb{R})$-error	Runtime in seconds
0	1.038045	1.038686	1.210235	1.0
25,000	0.005691	0.007215	0.053298	688.4
50,000	0.005115	0.006553	0.036513	1375.2
100,000	0.004749	0.005954	0.032411	2746.8
150,000	0.006465	0.008581	0.051907	4120.2
250,000	0.005075	0.006378	0.024458	6867.5
500,000	0.002082	0.002704	0.019604	13,763.7
750,000	0.00174	0.002233	0.012466	20,758.8

4.7 Conclusion

In Tables 1, 2, 3, 4, 5 and 6 above we present for a number of examples statistical estimations of different relative approximation errors for the proposed deep learning based approximation method, where the approximation errors are measured in the L^1, the L^2, and the L^∞ sense. The numerical simulations suggest that the deep learning based approximation method proposed in this work (see Framework 3.2 in Sect. 3.5 above) is indeed able to approximately calculate the solution of high-dimensional Kolmogorov PDEs not just at fixed space-time points but even on entire regions, that is, for example, on $[0, 1]^{100}$ in the case of the 100-dimensional heat equation in (59) in Sect. 4.2, on $[90, 110]^{100}$ in the case of the 100-dimensional Black–Scholes PDE with underlying independent geometric Brownian motions in (75) in Sect. 4.3, on $[90, 100]^{100}$ in the case of the 100-dimensional Black–Scholes PDE with correlated noise in (78) in Sect. 4.4, and on $\times_{i=1}^{25}\big([90, 110] \times [0.02, 0.2]\big)$ in the case of the 50-dimensional Kolmogorov PDE in (83) in Sect. 4.6. In constrast to the results for the relative L^1-approximation error and the relative L^2-approximation errors, however, the results for the relative L^∞-approximation errors in Tables 1 and 3, 4, 5 and 6 should be taken with great caution as the L^∞-approximation errors were calculated by means of Monte Carlo approximations according to Lemma 4.3 and the convergence rate there might very well be extremely slow (see, e.g., [38, Sect. 5.2]).

Funding The fifth author acknowledges funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy EXC 2044-390685587, Mathematics Muenster: Dynamics-Geometry-Structure.

Declarations

Conflict of interest The authors declare that there is no conflict of interests.

Availability of Data and Materials Not applicable.

Code Availability Relevant source codes can be downloaded from the GitHub repository at https://github.com/seb-becker/kolmogorov.

References

1. Alfonsi, A.: On the discretization schemes for the CIR (and Bessel squared) processes. Monte Carlo Methods Appl. **11**(4), 355–384 (2005)
2. Aliprantis, C.D., Border, K.C.: Infinite Dimensional Analysis: A Hitchhiker's Guide. Springer, Berlin (2006)
3. Bach, F., Moulines, E.: Non-strongly-convex smooth stochastic approximation with convergence rate $O(1/n)$. In: Advances in Neural Information Processing Systems, pp. 773–781 (2013)
4. Beck, C., Jentzen, A., Kuckuck, B.: Full error analysis for the training of deep neural networks. arXiv:1910.00121 (2019)
5. Becker, S., Cheridito, P., Jentzen, A.: Deep optimal stopping. J. Mach. Learn. Res. **20**, 74, 25 (2019)
6. Bellman, R.E.: Dynamic Programming. Princeton University Press, Princeton (1957)
7. Bercu, B., Fort, J.-C.: Generic stochastic gradient methods. In: Wiley Encyclopedia of Operations Research and Management Science, pp. 1–8 (2011)
8. Berner, J., Grohs, P., Jentzen, A.: Analysis of the generalization error: empirical risk minimization over deep artificial neural networks overcomes the curse of dimensionality in the numerical approximation of Black–Scholes partial differential equations. SIAM J. Math. Data Sci. **2**(3), 631–657 (2020)
9. Bishop, C.M.: Pattern Recognition and Machine Learning. Information Science and Statistics. Springer, New York (2006)

10. Bölcskei, H., Grohs, P., Kutyniok, G., Petersen, P.: Optimal approximation with sparsely connected deep neural networks. SIAM J. Math. Data Sci. **1**(1), 8–45 (2019)
11. Brennan, M.J., Schwartz, E.S.: Finite difference methods and jump processes arising in the pricing of contingent claims: a synthesis. J. Financ. Quant. Anal. **13**(3), 461–474 (1978)
12. Brenner, S., Scott, R.: The Mathematical Theory of Finite Element Methods, vol. 15. Springer, Berlin (2007)
13. Chau, N.H., Moulines, É., Rásonyi, M., Sabanis, S., Zhang, Y.: On stochastic gradient Langevin dynamics with dependent data streams: the fully non-convex case. arXiv:1905.13142 (2019)
14. Cox, S., Hutzenthaler, M., Jentzen, A. Local.: Lipschitz continuity in the initial value and strong completeness for nonlinear stochastic differential equations. arXiv:1309.5595 (2013). Accepted in Mem. Am. Math. Soc
15. Cucker, F., Smale, S.: On the mathematical foundations of learning. Bull. Am. Math. Soc. (N. S.) **39**(1), 1–49 (2002)
16. Dereich, S., Müller-Gronbach, T.: General multilevel adaptations for stochastic approximation algorithms of Robbins–Monro and Polyak–Ruppert type. Numer. Math. **142**(2), 279–328 (2019)
17. Fehrman, B., Gess, B., Jentzen, A.: Convergence rates for the stochastic gradient descent method for non-convex objective functions. J. Mach. Learn. Res. **21**, 136, 48 (2020)
18. Fujii, M., Takahashi, A., Takahashi, M.: Asymptotic expansion as prior knowledge in deep learning method for high dimensional BSDEs. Asia-Pac. Financ. Mark. **26**(3), 391–408 (2019)
19. Giles, M.B.: Multilevel Monte Carlo path simulation. Oper. Res. **56**(3), 607–617 (2008)
20. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pp. 249–256 (2010)
21. Golub, G.H., Van Loan, C.F.: Matrix Computations, Johns Hopkins Studies in the Mathematical Sciences, 4th edn. Johns Hopkins University Press, Baltimore (2013)
22. Graham, C., Talay, D.: Stochastic Simulation and Monte Carlo Methods, Volume 68 of Stochastic Modelling and Applied Probability. Springer, Heidelberg (2013). Mathematical foundations of stochastic simulation
23. Grohs, P., Hornung, F., Jentzen, A., von Wurstemberger, P.A.: Proof that artificial neural networks overcome the curse of dimensionality in the numerical approximation of Black–Scholes partial differential equations. arXiv:1809.02362 (2018). Accepted in Mem. Am. Math. Soc
24. Grohs, P., Hornung, F., Jentzen, A., Zimmermann, P.: Space-time error estimates for deep neural network approximations for differential equations. arXiv:1908.03833 (2019)
25. Grohs, P., Perekrestenko, D., Elbrächter, D., Bölcskei, H. Deep neural network approximation theory. arXiv:1901.02220 (2019)
26. Györfi, L., Kohler, M., Krzyżak, A., Walk, H.: A Distribution-Free Theory of Nonparametric Regression. Springer Series in Statistics. Springer, New York (2002)
27. Hairer, M., Hutzenthaler, M., Jentzen, A.: Loss of regularity for Kolmogorov equations. Ann. Probab. **43**(2), 468–527 (2015)
28. Han, J., Jentzen, A., E, W.: Solving high-dimensional partial differential equations using deep learning. Proc. Natl. Acad. Sci. U. S. A. **115**(34), 8505–8510 (2018)
29. Hefter, M., Herzwurm, A.: Strong convergence rates for Cox–Ingersoll–Ross processes-full parameter range. J. Math. Anal. Appl. **459**(2), 1079–1101 (2018)
30. Henry-Labordere, P.: Deep primal-dual algorithm for BSDEs: applications of machine learning to CVA and IM. SSRN Electron. J. (2017). Available at SSRN: https://ssrn.com/abstract=3071506
31. Hörmander, L.: Hypoelliptic second order differential equations. Acta Math. **119**, 147–171 (1967)
32. Hutzenthaler, M., Jentzen, A.: Numerical approximations of stochastic differential equations with non-globally Lipschitz continuous coefficients. Mem. Am. Math. Soc. **236**, 1112, v+99 (2015)
33. Hutzenthaler, M., Jentzen, A., Kloeden, P.E.: Strong convergence of an explicit numerical method for SDEs with nonglobally Lipschitz continuous coefficients. Ann. Appl. Probab. **22**(4), 1611–1641 (2012)
34. Hutzenthaler, M., Jentzen, A., Salimova, D.: Strong convergence of full-discrete nonlinearity-truncated accelerated exponential Euler-type approximations for stochastic Kuramoto–Sivashinsky equations. Commun. Math. Sci. **16**(6), 1489–1529 (2018)
35. Hutzenthaler, M., Jentzen, A., Wang, X.: Exponential integrability properties of numerical approximation processes for nonlinear stochastic differential equations. Math. Comput. **87**(311), 1353–1413 (2018)
36. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv:1502.03167 (2015)
37. Jentzen, A., von Wurstemberger, P.: Lower error bounds for the stochastic gradient descent optimization algorithm: sharp convergence rates for slowly and fast decaying learning rates. J. Complexity **57**, 101438, 16 (2020)

38. Jentzen, A., Welti, T.: Overall error analysis for the training of deep neural networks via stochastic gradient descent with random initialisation. arXiv:2003.01291 (2020)
39. Kingma, D., Ba, J.: Adam: a method for stochastic optimization. Proceedings of the International Conference on Learning Representations (ICLR) (2015)
40. Klenke, A.: Probability Theory. Universitext, 2nd edn. Springer, London (2014). A comprehensive course
41. Kloeden, P.E., Platen, E.: Numerical Solution of Stochastic Differential Equations, Volume 23 of Applications of Mathematics (New York). Springer, Berlin (1992)
42. Kloeden, P.E., Platen, E., Schurz, H.: Numerical Solution of SDE Through Computer Experiments. Springer, Berlin (2012)
43. Kushner, H.J.: Finite difference methods for the weak solutions of the Kolmogorov equations for the density of both diffusion and conditional diffusion processes. J. Math. Anal. Appl. **53**(2), 251–265 (1976)
44. Kutyniok, G., Petersen, P., Raslan, M., Schneider, R.: A theoretical analysis of deep neural networks and parametric PDEs. arXiv:1904.00377 (2019)
45. Lei, Y., Hu, T., Li, G., Tang, K.: Stochastic gradient descent for nonconvex learning without bounded gradient assumptions. IEEE Trans. Neural Netw. Learn. Syst. **31**(10), 4394–4400 (2020)
46. Maruyama, G.: Continuous Markov processes and stochastic equations. Rend. Circ. Mat. Palermo **2**(4), 48–90 (1955)
47. Massart, P.: Concentration Inequalities and Model Selection, Volume 1896 of Lecture Notes in Mathematics. Springer, Berlin (2007). Lectures from the 33rd Summer School on Probability Theory held in Saint-Flour, July 6–23, 2003. With a foreword by Jean Picard
48. Milstein, G.N.: Numerical Integration of Stochastic Differential Equations, Volume 313 of Mathematics and its Applications. Kluwer Academic Publishers Group, Dordrecht (1995). Translated and revised from the 1988 Russian original
49. Milstein, G.N., Tretyakov, M.V.: Stochastic Numerics for Mathematical Physics, Scientific Computation. Springer, Berlin (2004)
50. Milstein, G.N., Tretyakov, M.V.: Numerical integration of stochastic differential equations with nonglobally Lipschitz coefficients. SIAM J. Numer. Anal. **43**(3), 1139–1154 (2005)
51. Müller-Gronbach, T., Ritter, K.: Minimal errors for strong and weak approximation of stochastic differential equations. In: Monte Carlo and Quasi-Monte Carlo Methods 2006, pp. 53–82. Springer, Berlin 2008
52. Øksendal, B.: Stochastic Differential Equations, Universitext, 6th edn. Springer, Berlin (2003). An introduction with applications
53. Rogers, L.C.G., Williams, D.: Diffusions, Markov Processes, and Martingales. Volume 2. Cambridge Mathematical Library. Cambridge University Press, Cambridge (2000). Itô calculus, Reprint of the second (1994) edition
54. Ruder, S.: An overview of gradient descent optimization algorithms. arXiv:1609.04747 (2016)
55. Sabanis, S.: A note on tamed Euler approximations. Electron. Commun. Probab. **18**(47), 10 (2013)
56. Sabanis, S.: Euler approximations with varying coefficients: the case of superlinearly growing diffusion coefficients. Ann. Appl. Probab. **26**(4), 2083–2105 (2016)
57. Shalev-Shwartz, S., Ben-David, S.: Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press, Cambridge (2014)
58. E, W., Han, J., Jentzen, A.: Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. Commun. Math. Stat. **5**(4), 349–380 (2017)
59. E, W., Yu, B.: The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems. Commun. Math. Stat. **6**(1), 1–12 (2018)
60. Zhao, J., Davison, M., Corless, R.M.: Compact finite difference method for American option pricing. J. Comput. Appl. Math. **206**(1), 306–321 (2007)
61. Zienkiewicz, O.C., Taylor, R.L., Zienkiewicz, O.C., Taylor, R.L.: The Finite Element Method, vol. 3. McGraw-Hill, London (1977)