Deep learning based numerical approximation algorithms for stochastic partial differential equations and high-dimensional nonlinear filtering problems

Christian Beck[1], Sebastian Becker[2], Patrick Cheridito[3], Arnulf Jentzen[4,5], and Ariel Neufeld[6]

[1] Department of Mathematics, ETH Zurich, Switzerland, e-mail: christian.beck@math.ethz.ch

[2] RiskLab, Department of Mathematics, ETH Zurich, Switzerland, e-mail: sebastian.becker@math.ethz.ch

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

[4] Faculty of Mathematics and Computer Science, University of Münster Germany, e-mail: ajentzen@uni-muenster.de

[5] Department of Mathematics, ETH Zurich Switzerland, e-mail: arnulf.jentzen@math.ethz.ch

[6] Division of Mathematical Sciences, School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore, e-mail: ariel.neufeld@ntu.edu.sg

Thursday 3rd December, 2020

Abstract

In this article we introduce and study a deep learning based approximation algorithm for solutions of stochastic partial differential equations (SPDEs). In the proposed approximation algorithm we employ a deep neural network for every realization of the driving noise process of the SPDE to approximate the solution process of the SPDE under consideration. We test the performance of the proposed approximation algorithm in the case of stochastic heat equations with additive noise, stochastic heat equations with multiplicative noise, stochastic Black–Scholes equations with multiplicative noise, and Zakai equations from nonlinear filtering. In each of these SPDEs the proposed approximation algorithm produces accurate results with short run times in up to 50 space dimensions.

Keywords: stochastic partial differential equation, numerical method, artificial neural network, nonlinear filtering, Zakai equation

Contents

1 Introduction

Stochastic partial differential equations (SPDEs) are key ingredients in numerous models in engineering, finance, and natural sciences. For example, SPDEs commonly appear in models to price interest-rate based financial derivatives (cf., for example, (1.3) in Filipović et al. [34] and Theorem 3.5 in Harms et al. [49]), to describe random surfaces appearing in surface growth models (cf., for example, (3) in Hairer [46] and (1) in Blömker & Romito [15]), to describe the temporal dynamics in Euclidean quantum field theories (cf., for example, (1.1) in Mourrat & Weber [85]), to describe velocity fields in fully developed turbulent flows (cf., for example, (1.5) in Birnir [14] and (7) in Birnir [13]), and to describe the temporal development of the concentration of an unwanted (biological or chemical) contaminant in water (e.g., in the groundwater system, in a river, or in a water basin; cf., for example, (2.2) in Kallianpur & Xiong [63] and (1.1) in Kouritzin & Long [69]). Another prominent situation where SPDEs such as the Kushner equation (cf., for example, Kushner [75]) and the Zakai equation (cf., for example, Zakai [105]) appear is in the case of nonlinear filtering problems where SPDEs describe the density of the state space of the considered system. In particular, we refer, e.g., to [16, 21, 25, 29, 36, 37] for filtering problems in financial engineering, we refer, e.g., to [18, 24, 94, 95, 98, 103] for filtering problems in chemical engineering, and we refer, e.g., to [28, 19, 20, 23, 33, 89] for filtering problems in weather forecasting. SPDEs arising in nonlinear filtering problems are usually high-dimensional as the number of dimensions corresponds to the state space of the considered filtering problem. Most of the SPDEs in the above named applications cannot be solved explicitly and for about 30 years it has been an active field of research to design and study numerical algorithms which can approximatively compute solutions of SPDEs.

In order to be able to implement an approximation scheme for evolutionary type SPDEs on a computer both the time interval $[0, T]$ as well as the infinite dimensional state space have to be discretized. Several types of temporal discretizations and spatial discretizations have been proposed and studied in the scientific literature. In particular, we refer, e.g., to [26, 41, 52, 96, 99, 104] for temporal discretizations based on the linear implicit Euler method, we refer, e.g., to [55, 60, 81, 87, 102] for temporal discretizations based on exponential Euler-type methods, we refer, e.g., to [50, 51, 96, 99] for temporal discretizations based on linear implicit Crank–Nicolson-type methods, we refer, e.g., to [1, 9, 12, 35, 44, 76] for temporal discretizations based on splitting up approximation methods, we refer, e.g., to [58, 62, 71, 92, 101] for temporal discretizations based on higher order approximation methods, we refer, e.g., to [17, 38, 64, 70, 80, 99, 104] for spatial discretizations based on finite elements methods, we refer, e.g., to [45, 83, 90, 93, 96, 100] for spatial discretizations based on finite differences methods, and we refer, e.g., to [39, 51, 68, 79, 88, 86] for spatial discretizations based on spectral Galerkin methods. Moreover, the recent article [106] employs deep neural networks to approximately solve some one-dimensional SPDEs. We also refer to the overview articles [42, 59] and to the monographs [61, 72] for further references on numerical approximation methods for SPDEs.

In this article we introduce and study a deep learning based approximation algorithm for approximating solutions of possibly high-dimensional SPDEs. In the proposed approximation algorithm we employ a deep neural network for every realization of the driving noise process of the SPDE to approximate the solution process of the SPDE under consideration. The derivation of the proposed approximation scheme is inspired by the ideas in the articles [2, 3, 48] in which deep learning based algorithms for high-dimensional PDEs have been proposed and studied. We also refer, e.g., to [4, 11, 22, 30, 31, 32, 53, 54, 57, 91, 97] and the references therein for further articles on deep learning based approximation methods for PDEs. We test the performance of the approximation algorithm proposed in this article in the case of stochastic heat equations with additive noise (see Subsection 3.1 below), stochastic heat equations with multiplicative noise (see Subsection 3.2 below), stochastic Black–Scholes equations with multiplicative noise (see Subsection 3.3 below), and Zakai equations from nonlinear filtering (see Subsection 3.4 below). In each of these SPDEs the proposed approximation algorithm produces accurate results with short run times in up to 50 space dimensions.

The remainder of this paper is organized as follows. In Section 2 we derive (see Subsections 2.1–2.6 below) and formulate (see Subsections 2.7–2.8 below) the proposed approximation algorithm for SPDEs. In Section 3 we test the performance of the proposed approximation algorithm (see Subsection 2.8 below) in the case of stochastic heat equations with additive noise (see Subsection 3.1 below), stochastic heat equations with multiplicative noise (see Subsection 3.2 below), stochastic Black–Scholes equations with multiplicative noise (see Subsection 3.3 below), and Zakai equations (see Subsection 3.4 below). In Section 4 we present the Python source codes which we have used for the numerical simulations in Section 3.

2 Derivation and description of the proposed approximation algorithm

Let $T \in (0, \infty)$, $d \in \mathbb{N}$, let $\varphi\colon \mathbb{R}^d \to \mathbb{R}$ be a continuous function, let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space with a normal filtration $(\mathcal{F}_t)_{t\in[0,T]}$ (cf., e.g., Liu & Röckner [78, Definition 2.1.11]), let $Z\colon [0,T] \times \mathbb{R}^d \times \Omega \to \mathbb{R}$ be a sufficiently regular random field which satisfies for every $x \in \mathbb{R}^d$ that $(Z_t(x))_{t\in[0,T]}\colon [0,T] \times \Omega \to \mathbb{R}$ is an $(\mathcal{F}_t)_{t\in[0,T]}$-Itô process, let $\mu\colon \mathbb{R}^d \to \mathbb{R}^d$, $f\colon \mathbb{R}^d \times \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}$, and $b\colon \mathbb{R}^d \times \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}$ be sufficiently regular functions, let $\sigma\colon \mathbb{R}^d \to \mathbb{R}^{d\times d}$ be a sufficiently regular and sufficiently non-degenerate function, and let $X\colon [0,T] \times \mathbb{R}^d \times \Omega \to \mathbb{R}$ be a random field which satisfies for every $t \in [0,T]$, $x \in \mathbb{R}^d$ that $X_t(x)\colon \Omega \to \mathbb{R}$ is $\mathcal{F}_t/\mathcal{B}(\mathbb{R})$-measurable, which satisfies for every $\omega \in \Omega$ that $(X_t(x,\omega))_{(t,x)\in[0,T]\times\mathbb{R}^d} \in C^{0,2}([0,T] \times \mathbb{R}^d, \mathbb{R})$ has at most polynomially growing

partial derivatives, and which satisfies[1] that for every $t \in [0,T]$, $x \in \mathbb{R}^d$ it holds $\mathbb{P}$-a.s. that

$$\begin{aligned} X_t(x) &= \varphi(x) + \int_0^t f\big(x, X_s(x), (\nabla X_s)(x)\big)\, ds + \int_0^t b\big(x, X_s(x), (\nabla X_s)(x)\big)\, dZ_s(x) \\ &\quad + \int_0^t \Big[\tfrac{1}{2}\operatorname{Trace}\big(\sigma(x)[\sigma(x)]^*(\operatorname{Hess} X_s)(x)\big) + \big\langle \mu(x), (\nabla X_s)(x)\big\rangle_{\mathbb{R}^d}\Big]\, ds. \end{aligned} \tag{1}$$

Our goal is to approximately calculate under suitable hypothesis the solution process $X\colon [0,T] \times \mathbb{R}^d \times \Omega \to \mathbb{R}$ of the SPDE (1).

2.1 Temporal approximations

In this subsection we discretize the SPDE (1) in time by employing the splitting-up method (cf., for example, [8, 10, 40, 43, 44, 76]) to obtain a semi-discrete approximation problem. To this end let $N \in \mathbb{N}$, $t_0, t_1, \ldots, t_N \in [0,T]$ be real numbers with

$$0 = t_0 < t_1 < \ldots < t_N = T. \tag{2}$$

Observe that (1) yields that for every $n \in \{0,1,\ldots,N-1\}$, $t \in [t_n, t_{n+1}]$, $x \in \mathbb{R}^d$ it holds $\mathbb{P}$-a.s. that

$$\begin{aligned} X_t(x) &= X_{t_n}(x) + \int_{t_n}^t f\big(x, X_s(x), (\nabla X_s)(x)\big)\, ds + \int_{t_n}^t b\big(x, X_s(x), (\nabla X_s)(x)\big)\, dZ_s(x) \\ &\quad + \int_{t_n}^t \Big[\tfrac{1}{2}\operatorname{Trace}\big(\sigma(x)[\sigma(x)]^*(\operatorname{Hess} X_s)(x)\big) + \big\langle \mu(x), (\nabla X_s)(x)\big\rangle_{\mathbb{R}^d}\Big]\, ds. \end{aligned} \tag{3}$$

This illustrates for every $n \in \{0,1,\ldots,N-1\}$, $t \in [t_n, t_{n+1}]$, $x \in \mathbb{R}^d$ that

$$\begin{aligned} X_t(x) &\approx X_{t_n}(x) \\ &\quad + \int_{t_n}^{t_{n+1}} f\big(x, X_{t_n}(x), (\nabla X_{t_n})(x)\big)\, ds + \int_{t_n}^{t_{n+1}} b\big(x, X_{t_n}(x), (\nabla X_{t_n})(x)\big)\, dZ_s(x) \\ &\quad + \int_{t_n}^t \Big[\tfrac{1}{2}\operatorname{Trace}\big(\sigma(x)[\sigma(x)]^*(\operatorname{Hess} X_s)(x)\big) + \big\langle \mu(x), (\nabla X_s)(x)\big\rangle_{\mathbb{R}^d}\Big]\, ds. \end{aligned} \tag{4}$$

This, in turn, suggests for every $n \in \{0,1,\ldots,N-1\}$, $t \in [t_n, t_{n+1}]$, $x \in \mathbb{R}^d$ that

$$\begin{aligned} X_t(x) &\approx X_{t_n}(x) + f\big(x, X_{t_n}(x), (\nabla X_{t_n})(x)\big)\,(t_{n+1} - t_n) \\ &\quad + b\big(x, X_{t_n}(x), (\nabla X_{t_n})(x)\big)\,\big(Z_{t_{n+1}}(x) - Z_{t_n}(x)\big) \\ &\quad + \int_{t_n}^t \Big[\tfrac{1}{2}\operatorname{Trace}\big(\sigma(x)[\sigma(x)]^*(\operatorname{Hess} X_s)(x)\big) + \big\langle \mu(x), (\nabla X_s)(x)\big\rangle_{\mathbb{R}^d}\Big]\, ds. \end{aligned} \tag{5}$$

[1]Note that for every $d, m \in \mathbb{N}$ and every $(d \times m)$-matrix $A \in \mathbb{R}^{d\times m}$ it holds that $A^* \in \mathbb{R}^{m\times d}$ is the transpose of A.

To derive the splitting-up approximation let $U\colon (0,T]\times\mathbb{R}^d\times\Omega\to\mathbb{R}$ be a random field which satisfies for every $\omega\in\Omega$, $n\in\{0,1,\ldots,N-1\}$ that $(U_t(x,\omega))_{(t,x)\in(t_n,t_{n+1}]\times\mathbb{R}^d}\in C^{1,2}((t_n,t_{n+1}]\times\mathbb{R}^d,\mathbb{R})$ has at most polynomially growing partial derivatives, which satisfies for every $\omega\in\Omega$, $x\in\mathbb{R}^d$ that $\int_0^T\|(\operatorname{Hess}U_s)(x,\omega)\|_{\mathbb{R}^{d\times d}}+\|(\nabla U_s)(x,\omega)\|_{\mathbb{R}^d}\,ds<\infty$, and which satisfies that for every $n\in\{0,1,\ldots,N-1\}$, $t\in(t_n,t_{n+1}]$, $x\in\mathbb{R}^d$ it holds $\mathbb{P}$-a.s. that

$$\begin{aligned}U_t(x)&=X_{t_n}(x)+f(x,X_{t_n}(x),(\nabla X_{t_n})(x))\,(t_{n+1}-t_n)\\&\quad+b(x,X_{t_n}(x),(\nabla X_{t_n})(x))\,(Z_{t_{n+1}}(x)-Z_{t_n}(x))\\&\quad+\int_{t_n}^t\Big[\tfrac{1}{2}\operatorname{Trace}\big(\sigma(x)[\sigma(x)]^*(\operatorname{Hess}U_s)(x)\big)+\big\langle\mu(x),(\nabla U_s)(x)\big\rangle_{\mathbb{R}^d}\Big]\,ds.\end{aligned}\tag{6}$$

Note that (5) and (6) suggest for every $n\in\{1,2,\ldots,N\}$, $x\in\mathbb{R}^d$ that

$$U_{t_n}(x)\approx X_{t_n}(x).\tag{7}$$

Next let $V\colon[0,T]\times\mathbb{R}^d\times\Omega\to\mathbb{R}$ be a random field which satisfies for every $\omega\in\Omega$, $n\in\{0,1,\ldots,N-1\}$ that $(V_t(x,\omega))_{(t,x)\in(t_n,t_{n+1}]\times\mathbb{R}^d}\in C^{1,2}((t_n,t_{n+1}]\times\mathbb{R}^d,\mathbb{R})$ has at most polynomially growing partial derivatives, which satisfies for every $\omega\in\Omega$, $x\in\mathbb{R}^d$ that $\int_0^T\|(\operatorname{Hess}V_s)(x,\omega)\ \|_{\mathbb{R}^{d\times d}}+\|(\nabla V_s)(x,\omega)\|_{\mathbb{R}^d}\,ds<\infty$, and which satisfies for every $n\in\{0,1,\ldots,N-1\}$, $t\in(t_n,t_{n+1}]$, $x\in\mathbb{R}^d$ that $V_0(x)=\varphi(x)$ and

$$\begin{aligned}V_t(x)&=V_{t_n}(x)+f(x,V_{t_n}(x),(\nabla V_{t_n})(x))\,(t_{n+1}-t_n)\\&\quad+b(x,V_{t_n}(x),(\nabla V_{t_n})(x))\,(Z_{t_{n+1}}(x)-Z_{t_n}(x))\\&\quad+\int_{t_n}^t\Big[\tfrac{1}{2}\operatorname{Trace}\big(\sigma(x)[\sigma(x)]^*(\operatorname{Hess}V_s)(x)\big)+\big\langle\mu(x),(\nabla V_s)(x)\big\rangle_{\mathbb{R}^d}\Big]\,ds\end{aligned}\tag{8}$$

(cf., for example, Deck & Kruse [27], Hairer et al. [47, Section 4.4], Krylov [73, Chapter 8], and Krylov [74, Theorem 4.32] for existence, uniqueness, and regularity results for (6) and (8)). Note that (6) and (8) suggest for every $n\in\{1,2,\ldots,N\}$, $x\in\mathbb{R}^d$ that

$$V_{t_n}(x)\approx U_{t_n}(x).\tag{9}$$

Combining this with (7), in turn, suggests for every $n\in\{1,2,\ldots,N\}$, $x\in\mathbb{R}^d$ that

$$V_{t_n}(x)\approx X_{t_n}(x).\tag{10}$$

Observe that the random field V is a specific splitting-up type approximation for the random field X (cf., for example, [8, 10, 40, 43, 44, 76]). In the next subsection we derive a Feynman-Kac representation for V given Z (cf., for example, Milstein & Tretjakov [84, Section 2]).

2.2 An approximate Feynman-Kac type representation

In this subsection we derive a Feynman-Kac representation for V given Z. More specifically, for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ let $\mathcal{V}^{(z)}\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ satisfy for every $n\in\{0,1,\ldots,N-1\}$ that $(\mathcal{V}_t^{(z)}(x))_{(t,x)\in(t_n,t_{n+1}]\times\mathbb{R}^d}\in C^{1,2}((t_n,t_{n+1}]\times\mathbb{R}^d,\mathbb{R})$ has at most polynomially growing partial derivatives, which satisfies for every $x\in\mathbb{R}^d$ that $\int_0^T \|(\operatorname{Hess}\mathcal{V}_s^{(z)})(x)\|_{\mathbb{R}^{d\times d}}+\|(\nabla\mathcal{V}_s^{(z)})(x)\|_{\mathbb{R}^d}\,ds<\infty$, and which satisfies for every $n\in\{0,1,\ldots,N-1\}$, $t\in(t_n,t_{n+1}]$, $x\in\mathbb{R}^d$ that $\mathcal{V}_0^{(z)}(x)=\varphi(x)$ and

$$\begin{aligned}\mathcal{V}_t^{(z)}(x)&=\mathcal{V}_{t_n}^{(z)}(x)+f(x,\mathcal{V}_{t_n}^{(z)}(x),(\nabla\mathcal{V}_{t_n}^{(z)})(x))\,(t_{n+1}-t_n)\\&\quad+b(x,\mathcal{V}_{t_n}^{(z)}(x),(\nabla\mathcal{V}_{t_n}^{(z)})(x))\,(z_{t_{n+1}}(x)-z_{t_n}(x))\\&\quad+\int_{t_n}^t\Big[\tfrac{1}{2}\operatorname{Trace}\big(\sigma(x)[\sigma(x)]^*(\operatorname{Hess}\mathcal{V}_s^{(z)})(x)\big)+\big\langle\mu(x),(\nabla\mathcal{V}_s^{(z)})(x)\big\rangle_{\mathbb{R}^d}\Big]\,ds.\end{aligned}\tag{11}$$

Note that (8) and (11) ensure that for every $\omega\in\Omega$, $t\in[0,T]$, $x\in\mathbb{R}^d$ it holds that

$$\mathcal{V}_t^{(Z_s(y,\omega))_{(s,y)\in[0,T]\times\mathbb{R}^d}}(x)=V_t(x,\omega).\tag{12}$$

Combining this with (10) suggests for every sufficiently regular function $z\colon[0,T]\times\mathbb{R}^d\to\mathbb{R}$ and every $\omega\in\Omega$, $n\in\{0,1,\ldots,N\}$, $x\in\mathbb{R}^d$ that

$$\mathcal{V}_{t_n}^{(Z_s(y,\omega))_{(s,y)\in[0,T]\times\mathbb{R}^d}}(x)\approx X_{t_n}(x,\omega).\tag{13}$$

Moreover, note that (10)–(12) suggest for every sufficiently regular function $z\colon[0,T]\times\mathbb{R}^d\to\mathbb{R}$ and every $n\in\{0,1,\ldots,N\}$, $x\in\mathbb{R}^d$ that

$$\mathcal{V}_{t_n}^{(z)}(x)\approx\mathbb{E}\big[X_{t_n}(x)\,\big|\,Z=z\big].\tag{14}$$

In the following we introduce additional artificial stochastic processes in order to incorporate a Feynman-Kac type representation into (11). Let $B\colon[0,T]\times\Omega\to\mathbb{R}^d$ be a standard $(\mathcal{F}_t)_{t\in[0,T]}$-Brownian motion, let $\xi\colon\Omega\to\mathbb{R}^d$ be an $\mathcal{F}_0/\mathcal{B}(\mathbb{R}^d)$-measurable function which satisfies for every $p\in(0,\infty)$, $x\in\mathbb{R}^d$ that $\mathbb{P}(\|\xi-x\|_{\mathbb{R}^d}\leq p)>0$ and $\mathbb{E}[\|\xi\|_{\mathbb{R}^d}^p]<\infty$, assume that Z and (ξ,B) are independent random variables, and let $Y\colon[0,T]\times\Omega\to\mathbb{R}^d$ be an $(\mathcal{F}_t)_{t\in[0,T]}$-adapted stochastic process with continuous sample paths which satisfies that for every $t\in[0,T]$ it holds $\mathbb{P}$-a.s. that

$$Y_t=\xi+\int_0^t\mu(Y_s)\,ds+\int_0^t\sigma(Y_s)\,dB_s.\tag{15}$$

Note that the assumption that for every $p\in(0,\infty)$ it holds that $\mathbb{E}[\|\xi\|_{\mathbb{R}^d}^p]<\infty$ and the assumption that $\mu\colon\mathbb{R}^d\to\mathbb{R}^d$ and $\sigma\colon\mathbb{R}^d\to\mathbb{R}^{d\times d}$ are sufficiently regular functions ensure that for every $p\in(0,\infty)$ it holds that

$$\sup_{t\in[0,T]}\mathbb{E}\big[\|Y_t\|_{\mathbb{R}^d}^p\big]<\infty.\tag{16}$$

Moreover, observe that (11) implies that for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ and every $n\in\{0,1,\ldots,N-1\}$, $t\in(t_n,t_{n+1})$, $x\in\mathbb{R}^d$ it holds that

$$\tfrac{\partial}{\partial t}\big[\mathcal{V}_t^{(z)}(x)\big] = \big\langle \mu(x), (\nabla\mathcal{V}_t^{(z)})(x)\big\rangle_{\mathbb{R}^d} + \tfrac{1}{2}\operatorname{Trace}\big(\sigma(x)[\sigma(x)]^*(\operatorname{Hess}\mathcal{V}_t^{(z)})(x)\big). \tag{17}$$

This, in turn, assures that for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ and every $n\in\{0,1,\ldots,N-1\}$, $t\in(T-t_{n+1},T-t_n)$, $x\in\mathbb{R}^d$ it holds that

$$\tfrac{\partial}{\partial t}\big[\mathcal{V}_{T-t}^{(z)}(x)\big] + \big\langle \mu(x), (\nabla\mathcal{V}_{T-t}^{(z)})(x)\big\rangle_{\mathbb{R}^d} + \tfrac{1}{2}\operatorname{Trace}\big(\sigma(x)[\sigma(x)]^*(\operatorname{Hess}\mathcal{V}_{T-t}^{(z)})(x)\big) = 0. \tag{18}$$

Next note that Itô's formula, the hypothesis that for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ and every $n\in\{0,1,\ldots,N-1\}$ it holds that $(\mathcal{V}_t^{(z)}(x))_{(t,x)\in(t_n,t_{n+1}]\times\mathbb{R}^d}\in C^{1,2}((t_n,t_{n+1}]\times\mathbb{R}^d,\mathbb{R})$ (cf. (11)), and (15) guarantee that for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ and every $n\in\{0,1,\ldots,N-1\}$, $r,t\in[T-t_{n+1},T-t_n)$ with $r<t$ it holds $\mathbb{P}$-a.s. that

$$\begin{aligned}\mathcal{V}_{T-t}^{(z)}(Y_t) &= \mathcal{V}_{T-r}^{(z)}(Y_r) + \int_r^t \big\langle (\nabla\mathcal{V}_{T-s}^{(z)})(Y_s), \sigma(Y_s)\,dB_s\big\rangle_{\mathbb{R}^d} + \int_r^t \big(\tfrac{\partial}{\partial s}\big[\mathcal{V}_{T-s}^{(z)}\big]\big)(Y_s)\,ds\\ &\quad + \int_r^t \tfrac{1}{2}\operatorname{Trace}\big(\sigma(Y_s)[\sigma(Y_s)]^*(\operatorname{Hess}\mathcal{V}_{T-s}^{(z)})(Y_s)\big)\,ds\\ &\quad + \int_r^t \big\langle \mu(Y_s), (\nabla\mathcal{V}_{T-s}^{(z)})(Y_s)\big\rangle_{\mathbb{R}^d}\,ds.\end{aligned} \tag{19}$$

Combining this with (18) implies that for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ and every $n\in\{0,1,\ldots,N-1\}$, $r,t\in[T-t_{n+1},T-t_n)$ with $r<t$ it holds $\mathbb{P}$-a.s. that

$$\mathcal{V}_{T-t}^{(z)}(Y_t) = \mathcal{V}_{T-r}^{(z)}(Y_r) + \int_r^t \big\langle (\nabla\mathcal{V}_{T-s}^{(z)})(Y_s), \sigma(Y_s)\,dB_s\big\rangle_{\mathbb{R}^d}. \tag{20}$$

Hence, we obtain that for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ and every $n\in\{0,1,\ldots,N-1\}$, $t\in(T-t_{n+1},T-t_n)$ it holds $\mathbb{P}$-a.s. that

$$\mathcal{V}_{T-t}^{(z)}(Y_t) = \mathcal{V}_{t_{n+1}}^{(z)}(Y_{T-t_{n+1}}) + \int_{T-t_{n+1}}^t \big\langle (\nabla\mathcal{V}_{T-s}^{(z)})(Y_s), \sigma(Y_s)\,dB_s\big\rangle_{\mathbb{R}^d}. \tag{21}$$

Furthermore, note that (16), the hypothesis that $\sigma\colon\mathbb{R}^d\to\mathbb{R}^{d\times d}$ is a sufficiently regular function, and the hypothesis that for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ and every $n\in\{0,1,\ldots,N-1\}$ it holds that $(t_n,t_{n+1}]\times\mathbb{R}^d\ni(t,x)\mapsto(\nabla\mathcal{V}_t^{(z)})(x)\in\mathbb{R}^d$ is an at most polynomially growing function assure that for every $n\in\{0,1,\ldots,N-1\}$, $t\in(T-t_{n+1},T-t_n)$ it holds that

$$\int_{T-t_{n+1}}^t \mathbb{E}\Big[\big\|[\sigma(Y_s)]^*(\nabla\mathcal{V}_{T-s}^{(z)})(Y_s)\big\|_{\mathbb{R}^d}^2\Big]\,ds < \infty. \tag{22}$$

Therefore, we obtain that for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ and every $n\in\{0,1,\ldots,N-1\}$, $t\in(T-t_{n+1},T-t_n)$ it holds $\mathbb{P}$-a.s. that

$$\mathbb{E}\bigg[\int_{T-t_{n+1}}^{t}\big\langle(\nabla\mathcal{V}_{T-s}^{(z)})(Y_s),\sigma(Y_s)\,dB_s\big\rangle_{\mathbb{R}^d}\,\Big|\,\mathcal{F}_{T-t_{n+1}}\bigg]=0. \tag{23}$$

This and (21) demonstrate that for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ and every $n\in\{0,1,\ldots,N-1\}$, $t\in[T-t_{n+1},T-t_n)$ it holds $\mathbb{P}$-a.s. that

$$\mathbb{E}\big[\mathcal{V}_{T-t}^{(z)}(Y_t)\,\big|\,\mathcal{F}_{T-t_{n+1}}\big]=\mathbb{E}\big[\mathcal{V}_{t_{n+1}}^{(z)}(Y_{T-t_{n+1}})\,\big|\,\mathcal{F}_{T-t_{n+1}}\big]. \tag{24}$$

The fact that for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ and every $n\in\{0,1,\ldots,N-1\}$ it holds that the function $\Omega\ni\omega\mapsto\mathcal{V}_{t_{n+1}}^{(z)}(Y_{T-t_{n+1}}(\omega))\in\mathbb{R}$ is $\mathcal{F}_{T-t_{n+1}}/\mathcal{B}(\mathbb{R})$-measurable hence implies that for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ and every $n\in\{0,1,\ldots,N-1\}$, $t\in[T-t_{n+1},T-t_n)$ it holds $\mathbb{P}$-a.s. that

$$\mathbb{E}\big[\mathcal{V}_{T-t}^{(z)}(Y_t)\,\big|\,\mathcal{F}_{T-t_{n+1}}\big]=\mathcal{V}_{t_{n+1}}^{(z)}(Y_{T-t_{n+1}}). \tag{25}$$

Next observe that the hypothesis that for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ and every $n\in\{0,1,\ldots,N-1\}$ it holds that $(\mathcal{V}_t^{(z)}(x))_{(t,x)\in(t_n,t_{n+1}]\times\mathbb{R}^d}\in C^{1,2}((t_n,t_{n+1}]\times\mathbb{R}^d,\mathbb{R})$ has at most polynomially growing partial derivatives and the fact that for every $\omega\in\Omega$ it holds that $[0,T]\ni t\mapsto Y_t(\omega)\in\mathbb{R}^d$ is a continuous function ensure that for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ and every $\omega\in\Omega$, $n\in\{0,1,\ldots,N-1\}$ it holds that

$$\limsup_{t\nearrow T-t_n}\Big|\mathcal{V}_{T-t}^{(z)}(Y_t(\omega))-\mathcal{V}_{T-t}^{(z)}(Y_{T-t_n}(\omega))\Big|=0. \tag{26}$$

Furthermore, observe that (11) and the hypothesis that for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ and every $x\in\mathbb{R}^d$ it holds that $\int_0^T\|(\operatorname{Hess}\mathcal{V}_s^{(z)})(x)\|_{\mathbb{R}^{d\times d}}+\|(\nabla\mathcal{V}_s^{(z)})(x)\|_{\mathbb{R}^d}\,ds<\infty$ show that for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ and every $\omega\in\Omega$, $n\in\{0,1,\ldots,N-1\}$ it holds that

$$\begin{aligned}&\limsup_{t\nearrow T-t_n}\Big|\mathcal{V}_{T-t}^{(z)}(Y_{T-t_n}(\omega))-\big[\mathcal{V}_{t_n}^{(z)}(Y_{T-t_n}(\omega))\\&+f\big(Y_{T-t_n}(\omega),\mathcal{V}_{t_n}^{(z)}(Y_{T-t_n}(\omega)),(\nabla\mathcal{V}_{t_n}^{(z)})(Y_{T-t_n}(\omega))\big)\,(t_{n+1}-t_n)\\&+b\big(Y_{T-t_n}(\omega),\mathcal{V}_{t_n}^{(z)}(Y_{T-t_n}(\omega)),(\nabla\mathcal{V}_{t_n}^{(z)})(Y_{T-t_n}(\omega))\big)\,\big(z_{t_{n+1}}(Y_{T-t_n}(\omega))-z_{t_n}(Y_{T-t_n}(\omega))\big)\big]\Big|=0.\end{aligned} \tag{27}$$

Combining this with (26) demonstrates that for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ and every $\omega\in\Omega$, $n\in\{0,1,\ldots,N-1\}$ it holds that

$$\begin{aligned}&\limsup_{t\nearrow T-t_n}\Big|\mathcal{V}_{T-t}^{(z)}(Y_t(\omega))-\big[\mathcal{V}_{t_n}^{(z)}(Y_{T-t_n}(\omega))\\&+f\big(Y_{T-t_n}(\omega),\mathcal{V}_{t_n}^{(z)}(Y_{T-t_n}(\omega)),(\nabla\mathcal{V}_{t_n}^{(z)})(Y_{T-t_n}(\omega))\big)\,(t_{n+1}-t_n)\\&+b\big(Y_{T-t_n}(\omega),\mathcal{V}_{t_n}^{(z)}(Y_{T-t_n}(\omega)),(\nabla\mathcal{V}_{t_n}^{(z)})(Y_{T-t_n}(\omega))\big)\,\big(z_{t_{n+1}}(Y_{T-t_n}(\omega))-z_{t_n}(Y_{T-t_n}(\omega))\big)\big]\Big|=0.\end{aligned} \tag{28}$$

In addition, note that the hypothesis that for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ and every $n\in\{0,1,\ldots,N\}$ it holds that $(\mathcal{V}_t^{(z)}(x))_{(t,x)\in(t_n,t_{n+1}]\times\mathbb{R}^d}\in C^{1,2}((t_n,t_{n+1}]\times\mathbb{R}^d,\mathbb{R})$ has at most polynomially growing partial derivatives and (16) ensure that for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ and every $p\in(0,\infty)$ it holds that

$$\left(\sup_{t\in[0,T]}\mathbb{E}\big[\|Y_t\|_{\mathbb{R}^d}^p\big]\right)+\left(\sup_{t\in[0,T)}\mathbb{E}\big[|\mathcal{V}_{T-t}^{(z)}(Y_t)|^p\big]\right)+\left(\sup_{t\in(0,T]}\mathbb{E}\big[\|(\nabla\mathcal{V}_t^{(z)})(Y_{T-t})\|_{\mathbb{R}^d}^p\big]\right)<\infty. \tag{29}$$

Next note that the fact that for every $x\in\mathbb{R}$, $\omega\in\Omega$ it holds that $X_0(x,\omega)=\varphi(x)$, the hypothesis that for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ and every $x\in\mathbb{R}^d$ it holds that $\mathcal{V}_0^{(z)}(x)=\varphi(x)$, and the hypothesis that for every $\omega\in\Omega$ it holds that $(X_t(x,\omega))_{(t,x)\in[0,T]\times\mathbb{R}^d}\in C^{0,2}([0,T]\times\mathbb{R}^d,\mathbb{R})$ has at most polynomially growing partial derivatives demonstrate that for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ it holds that $(\mathcal{V}_0^{(z)}(x))_{x\in\mathbb{R}^d}\in C^2(\mathbb{R}^d,\mathbb{R})$ has at most polynomially growing derivatives. This and (29) imply that for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ and every $p\in(0,\infty)$ it holds that

$$\left(\sup_{t\in[0,T]}\mathbb{E}\big[\|Y_t\|_{\mathbb{R}^d}^p\big]\right)+\left(\sup_{t\in[0,T]}\mathbb{E}\big[|\mathcal{V}_{T-t}^{(z)}(Y_t)|^p\big]\right)+\left(\sup_{t\in[0,T]}\mathbb{E}\big[\|(\nabla\mathcal{V}_t^{(z)})(Y_{T-t})\|_{\mathbb{R}^d}^p\big]\right)<\infty. \tag{30}$$

The hypothesis that $f\colon\mathbb{R}^d\times\mathbb{R}\times\mathbb{R}^d\to\mathbb{R}$ and $b\colon\mathbb{R}^d\times\mathbb{R}\times\mathbb{R}^d\to\mathbb{R}$ are sufficiently regular functions hence proves that for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ and every $p\in(0,\infty)$ it holds that

$$\begin{aligned}&\sup_{t\in[0,T]}\mathbb{E}\Big[\big|\mathcal{V}_t^{(z)}(Y_{T-t})\big|^p\Big]\\&+\max_{n\in\{0,1,\ldots,N-1\}}\mathbb{E}\Big[\big|\mathcal{V}_{t_n}^{(z)}(Y_{T-t_n})+f\big(Y_{T-t_n},\mathcal{V}_{t_n}^{(z)}(Y_{T-t_n}),(\nabla\mathcal{V}_{t_n}^{(z)})(Y_{T-t_n})\big)(t_{n+1}-t_n)\\&+b\big(Y_{T-t_n},\mathcal{V}_{t_n}^{(z)}(Y_{T-t_n}),(\nabla\mathcal{V}_{t_n}^{(z)})(Y_{T-t_n})\big)\big(z_{t_{n+1}}(Y_{T-t_n})-z_{t_n}(Y_{T-t_n})\big)\big|^p\Big]<\infty.\end{aligned} \tag{31}$$

Combining (28) and, e.g., Hutzenthaler et al. [55, Proposition 4.5] therefore demonstrates that for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ and every $n\in\{0,1,\ldots,N-1\}$ it holds that

$$\begin{aligned}\limsup_{t\nearrow T-t_n}\mathbb{E}\Big[\big|\mathcal{V}_{T-t}^{(z)}(Y_t)-\big[\mathcal{V}_{t_n}^{(z)}(Y_{T-t_n})+f\big(Y_{T-t_n},\mathcal{V}_{t_n}^{(z)}(Y_{T-t_n}),(\nabla\mathcal{V}_{t_n}^{(z)})(Y_{T-t_n})\big)(t_{n+1}-t_n)\\+b\big(Y_{T-t_n},\mathcal{V}_{t_n}^{(z)}(Y_{T-t_n}),(\nabla\mathcal{V}_{t_n}^{(z)})(Y_{T-t_n})\big)\big(z_{t_{n+1}}(Y_{T-t_n})-z_{t_n}(Y_{T-t_n})\big)\big]\big|\Big]=0.\end{aligned} \tag{32}$$

This and (25) yield that for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ and every $n\in\{0,1,\ldots,N-1\}$ it holds $\mathbb{P}$-a.s. that

$$\begin{aligned}
&\mathcal{V}^{(z)}_{t_{n+1}}(Y_{T-t_{n+1}})\\
&=\mathbb{E}\Big[\mathcal{V}^{(z)}_{t_n}(Y_{T-t_n})+f\big(Y_{T-t_n},\mathcal{V}^{(z)}_{t_n}(Y_{T-t_n}),(\nabla\mathcal{V}^{(z)}_{t_n})(Y_{T-t_n})\big)\,(t_{n+1}-t_n)\\
&\quad+b\big(Y_{T-t_n},\mathcal{V}^{(z)}_{t_n}(Y_{T-t_n}),(\nabla\mathcal{V}^{(z)}_{t_n})(Y_{T-t_n})\big)\,\big(z_{t_{n+1}}(Y_{T-t_n})-z_{t_n}(Y_{T-t_n})\big)\,\big|\,\mathcal{F}_{T-t_{n+1}}\Big].
\end{aligned}\tag{33}$$

The tower property for conditional expectations therefore assures that for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ and every $n\in\{0,1,\ldots,N-1\}$ it holds $\mathbb{P}$-a.s. that

$$\begin{aligned}
&\mathbb{E}\Big[\mathcal{V}^{(z)}_{t_{n+1}}(Y_{T-t_{n+1}})\,\big|\,\mathfrak{S}(Y_{T-t_{n+1}})\Big]\\
&=\mathbb{E}\Big[\mathcal{V}^{(z)}_{t_n}(Y_{T-t_n})+f\big(Y_{T-t_n},\mathcal{V}^{(z)}_{t_n}(Y_{T-t_n}),(\nabla\mathcal{V}^{(z)}_{t_n})(Y_{T-t_n})\big)\,(t_{n+1}-t_n)\\
&\quad+b\big(Y_{T-t_n},\mathcal{V}^{(z)}_{t_n}(Y_{T-t_n}),(\nabla\mathcal{V}^{(z)}_{t_n})(Y_{T-t_n})\big)\,\big(z_{t_{n+1}}(Y_{T-t_n})-z_{t_n}(Y_{T-t_n})\big)\,\big|\,\mathfrak{S}(Y_{T-t_{n+1}})\Big].
\end{aligned}\tag{34}$$

In addition, observe that the fact that for every $n\in\{0,1,\ldots,N-1\}$ it holds that the function $\Omega\ni\omega\mapsto Y_{T-t_{n+1}}(\omega)\in\mathbb{R}^d$ is $\mathfrak{S}(Y_{T-t_{n+1}})/\mathcal{B}(\mathbb{R}^d)$-measurable assures that for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ and every $n\in\{0,1,\ldots,N-1\}$ it holds $\mathbb{P}$-a.s. that

$$\mathcal{V}^{(z)}_{t_{n+1}}(Y_{T-t_{n+1}})=\mathbb{E}\Big[\mathcal{V}^{(z)}_{t_{n+1}}(Y_{T-t_{n+1}})\,\big|\,\mathfrak{S}(Y_{T-t_{n+1}})\Big].\tag{35}$$

This and (34) imply that for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ and every $n\in\{0,1,\ldots,N-1\}$ it holds $\mathbb{P}$-a.s. that

$$\begin{aligned}
&\mathcal{V}^{(z)}_{t_{n+1}}(Y_{T-t_{n+1}})\\
&=\mathbb{E}\Big[\mathcal{V}^{(z)}_{t_n}(Y_{T-t_n})+f\big(Y_{T-t_n},\mathcal{V}^{(z)}_{t_n}(Y_{T-t_n}),(\nabla\mathcal{V}^{(z)}_{t_n})(Y_{T-t_n})\big)\,(t_{n+1}-t_n)\\
&\quad+b\big(Y_{T-t_n},\mathcal{V}^{(z)}_{t_n}(Y_{T-t_n}),(\nabla\mathcal{V}^{(z)}_{t_n})(Y_{T-t_n})\big)\,\big(z_{t_{n+1}}(Y_{T-t_n})-z_{t_n}(Y_{T-t_n})\big)\,\big|\,\mathfrak{S}(Y_{T-t_{n+1}})\Big].
\end{aligned}\tag{36}$$

Equation (36) constitutes the Feynman-Kac type representation we were aiming at. In the following subsection we employ the factorization lemma (cf., for example, Becker et al. [6, Lemma 2.1] or Klenke [66, Corollary 1.97]) and the L^2-minimality property of conditional expectations (cf., for example, Klenke [66, Corollary 8.17]) to reformulate (36) as recursive minimization problems.

2.3 Formulation as iterative recursive minimization problems

In this subsection we reformulate (36) as recursive minimization problems. For this observe that (31) shows that for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ and every

$n \in \{0, 1, \ldots, N-1\}$ it holds that

$$\mathbb{E}\Big[\Big|\mathcal{V}_{t_n}^{(z)}(Y_{T-t_n}) + f\big(Y_{T-t_n}, \mathcal{V}_{t_n}^{(z)}(Y_{T-t_n}), (\nabla \mathcal{V}_{t_n}^{(z)})(Y_{T-t_n})\big)\,(t_{n+1} - t_n) + b\big(Y_{T-t_n}, \mathcal{V}_{t_n}^{(z)}(Y_{T-t_n}), (\nabla \mathcal{V}_{t_n}^{(z)})(Y_{T-t_n})\big)\big(z_{t_{n+1}}(Y_{T-t_n}) - z_{t_n}(Y_{T-t_n})\big)\Big|^2\Big] < \infty. \quad (37)$$

The factorization lemma, e.g., in [6, Lemma 2.1] (applied with $(S, \mathcal{S}) \curvearrowleft (\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$, $\Omega \curvearrowleft \Omega$, $X \curvearrowleft Y_{T-t_{n+1}}$ for $n \in \{0, 1, \ldots, N-1\}$ in the notation of [6, Lemma 2.1]), the L^2-minimality property for conditional expectations, e.g., in Klenke [66, Corollary 8.17] (applied with $X \curvearrowleft \Omega \ni \omega \mapsto \mathcal{V}^{(z)}(Y_{T-t_n}(\omega)) + f(Y_{T-t_n}(\omega), \mathcal{V}_{t_n}^{(z)}(Y_{T-t_n}(\omega)), (\nabla \mathcal{V}_{t_n}^{(z)})(Y_{T-t_n}(\omega)))$ $(t_{n+1} - t_n) + b(Y_{T-t_n}, \mathcal{V}_{t_n}^{(z)}(Y_{T-t_n}), (\nabla \mathcal{V}_{t_n}^{(z)})(Y_{T-t_n}))(z_{t_{n+1}}(Y_{T-t_n}) - z_{t_n}(Y_{T-t_n})) \in \mathbb{R}$, $\mathcal{F} \curvearrowleft \mathfrak{S}(Y_{T-t_{n+1}})$, $\mathcal{A} \curvearrowleft \mathcal{F}$ in the notation of [66, Corollary 8.17]), the fact that for every sufficiently regular function $z\colon [0,T] \times \mathbb{R}^d \to \mathbb{R}$ and every $n \in \{0, 1, \ldots, N-1\}$ it holds that $\mathbb{R}^d \ni x \mapsto \mathcal{V}_{t_{n+1}}^{(z)}(x) \in \mathbb{R}$ is a continuous function, the fact that for every Borel measurable set $A \in \mathcal{B}(\mathbb{R}^d)$ with positive Lebesgue measure it holds that $\min_{n \in \{0,1,\ldots,N-1\}} \mathbb{P}(Y_{T-t_{n+1}} \in A) > 0$, and (36) hence imply that for every sufficiently regular function $z\colon [0,T] \times \mathbb{R}^d \to \mathbb{R}$ and every $n \in \{0, 1, \ldots, N-1\}$ it holds that

$$\begin{aligned}(\mathcal{V}_{t_{n+1}}^{(z)}(x))_{x \in \mathbb{R}^d} = \operatorname*{argmin}_{u \in C(\mathbb{R}^d, \mathbb{R})} \mathbb{E}\Big[\Big|u(Y_{T-t_{n+1}}) - \big[\mathcal{V}_{t_n}^{(z)}(Y_{T-t_n}) \\ + f\big(Y_{T-t_n}, \mathcal{V}_{t_n}^{(z)}(Y_{T-t_n}), (\nabla \mathcal{V}_{t_n}^{(z)})(Y_{T-t_n})\big)\,(t_{n+1} - t_n) \\ + b\big(Y_{T-t_n}, \mathcal{V}_{t_n}^{(z)}(Y_{T-t_n}), (\nabla \mathcal{V}_{t_n}^{(z)})(Y_{T-t_n})\big)\big(z_{t_{n+1}}(Y_{T-t_n}) - z_{t_n}(Y_{T-t_n})\big)\big]\Big|^2\Big].\end{aligned} \quad (38)$$

Therefore, we obtain that for every sufficiently regular function $z\colon [0,T] \times \mathbb{R}^d \to \mathbb{R}$ and every $n \in \{1, 2, \ldots, N\}$ it holds that

$$\begin{aligned}(\mathcal{V}_{t_n}^{(z)}(x))_{x \in \mathbb{R}^d} = \operatorname*{argmin}_{u \in C(\mathbb{R}^d, \mathbb{R})} \mathbb{E}\Big[\Big|u(Y_{T-t_n}) - \big[\mathcal{V}_{t_{n-1}}^{(z)}(Y_{T-t_{n-1}}) \\ + f\big(Y_{T-t_{n-1}}, \mathcal{V}_{t_{n-1}}^{(z)}(Y_{T-t_{n-1}}), (\nabla \mathcal{V}_{t_{n-1}}^{(z)})(Y_{T-t_{n-1}})\big)\,(t_n - t_{n-1}) \\ + b\big(Y_{T-t_{n-1}}, \mathcal{V}_{t_{n-1}}^{(z)}(Y_{T-t_{n-1}}), (\nabla \mathcal{V}_{t_{n-1}}^{(z)})(Y_{T-t_{n-1}})\big)\big(z_{t_n}(Y_{T-t_{n-1}}) - z_{t_{n-1}}(Y_{T-t_{n-1}})\big)\big]\Big|^2\Big].\end{aligned} \quad (39)$$

In the following subsection we approximate for every sufficiently regular function $z\colon [0,T] \times \mathbb{R}^d \to \mathbb{R}$ and every $n \in \{1, 2, \ldots, N\}$ the function $\mathbb{R}^d \ni x \mapsto \mathcal{V}_{t_n}^{(z)}(x) \in \mathbb{R}$ by suitable deep neural networks.

2.4 Deep neural network approximations

In this subsection we employ for every sufficiently regular function $z\colon [0,T] \times \mathbb{R}^d \to \mathbb{R}$ and every $n \in \{1, 2, \ldots, N\}$ suitable approximations for the function

$$\mathbb{R}^d \ni x \mapsto \mathcal{V}_{t_n}^{(z)}(x) \in \mathbb{R}. \quad (40)$$

More specifically, let $\nu \in \mathbb{N}$ and for every sufficiently regular function $z\colon [0,T] \times \mathbb{R}^d \to \mathbb{R}$ let $\mathbb{V}_n^{(z)} = (\mathbb{V}_n^{(z)}(\theta,x))_{(\theta,x)\in\mathbb{R}^\nu\times\mathbb{R}^d}\colon \mathbb{R}^\nu \times \mathbb{R}^d \to \mathbb{R}$, $n \in \{0,1,\ldots,N\}$, be continuously differentiable functions which satisfy for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d \to \mathbb{R}$ and every $\theta \in \mathbb{R}^\nu$, $x \in \mathbb{R}^d$ that $\mathbb{V}_0^{(z)}(\theta,x) = \varphi(x)$. For every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d \to \mathbb{R}$, every $n \in \{1,2,\ldots,N\}$, $x \in \mathbb{R}^d$, and every *suitable* $\theta \in \mathbb{R}^\nu$ we think of $\mathbb{V}_n^{(z)}(\theta,x) \in \mathbb{R}$ as an appropriate approximation

$$\mathbb{V}_n^{(z)}(\theta,x) \approx \mathcal{V}_{t_n}^{(z)}(x) \tag{41}$$

of $\mathcal{V}_{t_n}^{(z)}(x)$. Combining this and (13) indicates for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d \to \mathbb{R}$, every $n \in \{1,2,\ldots,N\}$, $x \in \mathbb{R}^d$, and every *suitable* $\theta \in \mathbb{R}^\nu$ that

$$\mathbb{V}_n^{(z)}(\theta,x) \approx \mathbb{E}\big[X_{t_n}(x) \,\big|\, Z = z\big]. \tag{42}$$

For every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d \to \mathbb{R}$ we propose to choose the functions $\mathbb{V}_n^{(z)}\colon \mathbb{R}^\nu \times \mathbb{R}^d \to \mathbb{R}$, $n \in \{1,2,\ldots,N\}$, as deep neural networks (cf., for instance, [7, 77]). For example, for every $k \in \mathbb{N}$ let $\mathcal{T}_k\colon \mathbb{R}^k \to \mathbb{R}^k$ satisfy for every $x = (x_1,x_2,\ldots,x_k) \in \mathbb{R}^k$ that

$$\mathcal{T}_k(x) = \big(\tanh(x_1),\tanh(x_2),\ldots,\tanh(x_k)\big) \tag{43}$$

(multidimensional version of the tangens hyperbolicus), for every $\theta = (\theta_1,\theta_2,\ldots,\theta_\nu) \in \mathbb{R}^\nu$, $v \in \mathbb{N}_0 = \{0\} \cup \mathbb{N}$, $k,l \in \mathbb{N}$ with $v + lk + l \leq \nu$ let $A_{k,l}^{\theta,v}\colon \mathbb{R}^k \to \mathbb{R}^l$ satisfy for every $x = (x_1,x_2,\ldots,x_k) \in \mathbb{R}^k$ that

$$A_{k,l}^{\theta,v}(x) = \begin{pmatrix} \theta_{v+1} & \theta_{v+2} & \ldots & \theta_{v+k} \\ \theta_{v+k+1} & \theta_{v+k+2} & \ldots & \theta_{v+2k} \\ \theta_{v+2k+1} & \theta_{v+2k+2} & \ldots & \theta_{v+3k} \\ \vdots & \vdots & \vdots & \vdots \\ \theta_{v+(l-1)k+1} & \theta_{v+(l-1)k+2} & \ldots & \theta_{v+lk} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_k \end{pmatrix} + \begin{pmatrix} \theta_{v+lk+1} \\ \theta_{v+lk+2} \\ \theta_{v+lk+3} \\ \vdots \\ \theta_{v+lk+l} \end{pmatrix} \tag{44}$$

(affine function), let $s \in \{3,4,5,6,\ldots\}$, assume that $s(N+1)d(d+1) \leq \nu$, and for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d \to \mathbb{R}$ let $\mathbb{V}_n^{(z)}\colon \mathbb{R}^\nu \times \mathbb{R}^d \to \mathbb{R}$, $n \in \{0,1,\ldots,N\}$, satisfy for every $n \in \{1,2,\ldots,N\}$, $\theta \in \mathbb{R}^\nu$, $x \in \mathbb{R}^d$ that $\mathbb{V}_0^{(z)}(\theta,x) = \varphi(x)$ and

$$\begin{aligned} &\mathbb{V}_n^{(z)}(\theta,x) = \\ &\big(A_{d,1}^{\theta,(sn+s-1)d(d+1)} \circ \mathcal{T}_d \circ A_{d,d}^{\theta,(sn+s-2)d(d+1)} \circ \ldots \circ \mathcal{T}_d \circ A_{d,d}^{\theta,(sn+1)d(d+1)} \circ \mathcal{T}_d \circ A_{d,d}^{\theta,snd(d+1)}\big)(x). \end{aligned} \tag{45}$$

For every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d \to \mathbb{R}$ and every $n \in \{1,2,\ldots,N\}$ the function $\mathbb{V}_n^{(z)}\colon \mathbb{R}^\nu \times \mathbb{R}^d \to \mathbb{R}$ in (45) describes a fully-connected feedforward deep 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 multidimensional versions of the tangens hyperbolicus as activation functions (see (43)).

2.5 Stochastic gradient descent based minimization

We intend to find for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ *suitable* $\theta^{z,1},\theta^{z,2},\ldots,\theta^{z,N}\in\mathbb{R}^\nu$ in (41) by recursive minimization. More precisely, for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ we intend to find for $n\in\{1,2,\ldots,N\}$, $\theta^{z,0},\theta^{z,1},\ldots,\theta^{z,n-1}\in\mathbb{R}^\nu$ a suitable $\theta^{z,n}\in\mathbb{R}^\nu$ as an approximate minimizer of the function

$$\begin{aligned}\mathbb{R}^\nu\ni\theta\mapsto\mathbb{E}\Big[\Big|\mathbb{V}_n^{(z)}(\theta,Y_{T-t_n})-\big[\mathbb{V}_{n-1}^{(z)}(\theta^{z,n-1},Y_{T-t_{n-1}})\\+f\big(Y_{T-t_{n-1}},\mathbb{V}_{n-1}^{(z)}(\theta^{z,n-1},Y_{T-t_{n-1}}),(\nabla_x\mathbb{V}_{n-1}^{(z)})(\theta^{z,n-1},Y_{T-t_{n-1}})\big)(t_n-t_{n-1})\\+b\big(Y_{T-t_{n-1}},\mathbb{V}_{n-1}^{(z)}(\theta^{z,n-1},Y_{T-t_{n-1}}),(\nabla_x\mathbb{V}_{n-1}^{(z)})(\theta^{z,n-1},Y_{T-t_{n-1}})\big)\\\cdot\big(z_{t_n}(Y_{T-t_{n-1}})-z_{t_{n-1}}(Y_{T-t_{n-1}})\big)\big]\Big|^2\Big]\in\mathbb{R}\end{aligned}\tag{46}$$

(cf. (39) and (41) above). To this end for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ let $B^{z,m}\colon [0,T]\times\Omega\to\mathbb{R}^d$, $m\in\mathbb{N}_0$, be i.i.d. standard $(\mathcal{F}_t)_{t\in[0,T]}$-Brownian motions, let $\xi^{z,m}\colon\Omega\to\mathbb{R}^d$, $m\in\mathbb{N}_0$, be i.i.d. $\mathcal{F}_0/\mathcal{B}(\mathbb{R}^d)$-measurable functions, for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ and every $m\in\mathbb{N}_0$ let $Y^{z,m}\colon [0,T]\times\Omega\to\mathbb{R}^d$ be an $(\mathcal{F}_t)_{t\in[0,T]}$-adapted stochastic process with continuous sample paths which satisfies that for every $t\in[0,T]$ it holds $\mathbb{P}$-a.s. that

$$Y_t^{z,m}=\xi^{z,m}+\int_0^t\mu(Y_s^{z,m})\,ds+\int_0^t\sigma(Y_s^{z,m})\,dB_s^{z,m},\tag{47}$$

let $\gamma\in(0,\infty)$, $M\in\mathbb{N}$, and for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ let $\vartheta^{z,n}=(\vartheta_m^{z,n})_{m\in\mathbb{N}_0}\colon\mathbb{N}_0\times\Omega\to\mathbb{R}^\nu$, $n\in\{0,1,\ldots,N\}$, be stochastic processes which satisfy for every $n\in\{1,2,\ldots,N\}$, $m\in\mathbb{N}_0$ that

$$\begin{aligned}\vartheta_{m+1}^{z,n}=\vartheta_m^{z,n}-2\gamma\cdot(\nabla_\theta\mathbb{V}_n^{(z)})(\vartheta_m^{z,n},Y_{T-t_n}^{z,m})\cdot\Big[\mathbb{V}_n^{(z)}(\vartheta_m^{z,n},Y_{T-t_n}^{z,m})-\mathbb{V}_{n-1}^{(z)}(\vartheta_M^{z,n-1},Y_{T-t_{n-1}}^{z,m})\\-f\big(Y_{T-t_{n-1}}^{z,m},\mathbb{V}_{n-1}^{(z)}(\vartheta_M^{z,n-1},Y_{T-t_{n-1}}^{z,m}),(\nabla_x\mathbb{V}_{n-1}^{(z)})(\vartheta_M^{z,n-1},Y_{T-t_{n-1}}^{z,m})\big)(t_n-t_{n-1})\\-b\big(Y_{T-t_{n-1}}^{z,m},\mathbb{V}_{n-1}^{(z)}(\vartheta_M^{z,n-1},Y_{T-t_{n-1}}^{z,m}),(\nabla_x\mathbb{V}_{n-1}^{(z)})(\vartheta_M^{z,n-1},Y_{T-t_{n-1}}^{z,m})\big)\\\cdot\big(z_{t_n}(Y_{T-t_{n-1}}^{z,m})-z_{t_{n-1}}(Y_{T-t_{n-1}}^{z,m})\big)\Big]\end{aligned}\tag{48}$$

(cf. (66)–(68) below).

2.6 Temporal discretization of the auxiliary stochastic process

Equation (48) provides us an implementable numerical algorithm in the special case where for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ one can simulate exactly from the solution processes $Y^{z,m}\colon [0,T]\times\Omega\to\mathbb{R}^d$, $m\in\mathbb{N}_0$, of the SDEs in (47) (cf. also

(15) above). In the case where it is not possible to simulate for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ exactly from the solution processes $Y^{z,m}\colon [0,T]\times\Omega\to\mathbb{R}^d$, $m\in\mathbb{N}_0$, of the SDEs in (47), one can employ a numerical approximation method for SDEs, say, the Euler-Maruyama scheme, to approximatively simulate for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ from the solution processes $Y^{z,m}\colon [0,T]\times\Omega\to\mathbb{R}^d$, $m\in\mathbb{N}_0$, of the SDEs in (47). This is subject of this subsection. More formally, note that (47) implies that for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ and every $m\in\mathbb{N}_0$, $r,t\in[0,T]$ with $r\leq t$ it holds $\mathbb{P}$-a.s. that

$$Y_t^{z,m}=Y_r^{z,m}+\int_r^t\mu(Y_s^{z,m})\,ds+\int_r^t\sigma(Y_s^{z,m})\,dB_s^{z,m}. \tag{49}$$

Hence, we obtain that for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ and every $m\in\mathbb{N}_0$, $n\in\{0,1,\ldots,N-1\}$ it holds $\mathbb{P}$-a.s. that

$$Y_{T-t_n}^{z,m}=Y_{T-t_{n+1}}^{z,m}+\int_{T-t_{n+1}}^{T-t_n}\mu(Y_s^{z,m})\,ds+\int_{T-t_{n+1}}^{T-t_n}\sigma(Y_s^{z,m})\,dB_s^{z,m}. \tag{50}$$

This shows that for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ and every $m\in\mathbb{N}_0$, $n\in\{0,1,\ldots,N-1\}$ it holds $\mathbb{P}$-a.s. that

$$Y_{T-t_{N-(n+1)}}^{z,m}=Y_{T-t_{N-n}}^{z,m}+\int_{T-t_{N-n}}^{T-t_{N-(n+1)}}\mu(Y_s^{z,m})\,ds+\int_{T-t_{N-n}}^{T-t_{N-(n+1)}}\sigma(Y_s^{z,m})\,dB_s^{z,m}. \tag{51}$$

Next we introduce suitable real numbers which allow us to reformulate (51) in a more compact way. More formally, let $\tau_n\in[0,T]$, $n\in\{0,1,\ldots,N\}$, satisfy for every $n\in\{0,1,\ldots,N\}$ that

$$\tau_n=T-t_{N-n}. \tag{52}$$

Observe that (2) and (52) ensure that

$$0=\tau_0<\tau_1<\cdots<\tau_N=T. \tag{53}$$

Moreover, note that (51) and (52) demonstrate that for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ and every $m\in\mathbb{N}_0$, $n\in\{0,1,\ldots,N-1\}$ it holds $\mathbb{P}$-a.s. that

$$Y_{\tau_{n+1}}^{z,m}=Y_{\tau_n}^{z,m}+\int_{\tau_n}^{\tau_{n+1}}\mu(Y_s^{z,m})\,ds+\int_{\tau_n}^{\tau_{n+1}}\sigma(Y_s^{z,m})\,dB_s^{z,m}. \tag{54}$$

This suggests for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ and every $m\in\mathbb{N}_0$, $n\in\{0,1,\ldots,N-1\}$ that

$$Y_{\tau_{n+1}}^{z,m}\approx Y_{\tau_n}^{z,m}+\mu(Y_{\tau_n}^{z,m})\,(\tau_{n+1}-\tau_n)+\sigma(Y_{\tau_n}^{z,m})\,(B_{\tau_{n+1}}^{z,m}-B_{\tau_n}^{z,m}). \tag{55}$$

Based on (55) we now introduce for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ suitable Euler-Maruyama approximations for the solution processes $Y^{z,m}\colon [0,T]\times\Omega\to\mathbb{R}^d$, $m\in\mathbb{N}_0$, of the SDEs in (47). More formally, for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ and every $m\in\mathbb{N}_0$ let $\mathcal{Y}^{z,m}=(\mathcal{Y}^{z,m}_n)_{n\in\{0,1,\ldots,N\}}\colon\{0,1,\ldots,N\}\times\Omega\to\mathbb{R}^d$ be the stochastic process which satisfies for every $n\in\{0,1,\ldots,N-1\}$ that $\mathcal{Y}^{z,m}_0=\xi^{z,m}$ and

$$\mathcal{Y}^{z,m}_{n+1}=\mathcal{Y}^{z,m}_n+\mu(\mathcal{Y}^{z,m}_n)\,(\tau_{n+1}-\tau_n)+\sigma(\mathcal{Y}^{z,m}_n)\,(B^{z,m}_{\tau_{n+1}}-B^{z,m}_{\tau_n}). \tag{56}$$

Observe that (52), (55), and (56) suggest for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ and every $m\in\mathbb{N}_0$, $n\in\{0,1,\ldots,N\}$ that

$$\mathcal{Y}^{z,m}_n\approx Y^{z,m}_{\tau_n}=Y^{z,m}_{T-t_{N-n}}. \tag{57}$$

This, in turn, suggests for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ and every $m\in\mathbb{N}_0$, $n\in\{0,1,\ldots,N\}$ that

$$Y^{z,m}_{T-t_n}\approx\mathcal{Y}^{z,m}_{N-n}. \tag{58}$$

In the next step we employ (58) to derive for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ approximations of the stochastic processes $\vartheta^{z,n}\colon\mathbb{N}_0\times\Omega\to\mathbb{R}^\nu$, $n\in\{0,1,\ldots,N\}$, in (48) which are also implementable in the case where one cannot simulate exactly from the solution processes of the SDEs in (47). More precisely, for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ let $\Theta^{z,n}=(\Theta^{z,n}_m)_{m\in\mathbb{N}_0}\colon\mathbb{N}_0\times\Omega\to\mathbb{R}$, $n\in\{0,1,\ldots,N\}$, be stochastic processes which satisfy for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ and every $n\in\{1,2,\ldots,N\}$, $m\in\mathbb{N}_0$ that

$$\begin{aligned}\Theta^{z,n}_{m+1}&=\Theta^{z,n}_m-2\gamma\cdot(\nabla_\theta\mathbb{V}^{(z)}_n)(\Theta^{z,n}_m,\mathcal{Y}^{z,m}_{N-n})\cdot\Big[\mathbb{V}^{(z)}_n(\Theta^{z,n}_m,\mathcal{Y}^{z,m}_{N-n})-\mathbb{V}^{(z)}_{n-1}(\Theta^{z,n-1}_M,\mathcal{Y}^{z,m}_{N-n+1})\\&\quad-f\big(\mathcal{Y}^{z,m}_{N-n+1},\mathbb{V}^{(z)}_{n-1}(\Theta^{z,n-1}_M,\mathcal{Y}^{z,m}_{N-n+1}),(\nabla_x\mathbb{V}^{(z)}_{n-1})(\Theta^{z,n-1}_M,\mathcal{Y}^{z,m}_{N-n+1})\big)\,(t_n-t_{n-1})\\&\quad-b\big(\mathcal{Y}^{z,m}_{N-n+1},\mathbb{V}^{(z)}_{n-1}(\Theta^{z,n-1}_M,\mathcal{Y}^{z,m}_{N-n+1}),(\nabla_x\mathbb{V}^{(z)}_{n-1})(\Theta^{z,n-1}_M,\mathcal{Y}^{z,m}_{N-n+1})\big)\\&\quad\cdot\big(z_{t_n}(\mathcal{Y}^{z,m}_{N-n+1})-z_{t_{n-1}}(\mathcal{Y}^{z,m}_{N-n+1})\big)\Big]\end{aligned} \tag{59}$$

(cf. (66)–(68) below). Note that (48) , (58), and (59) indicate for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$, every $n\in\{1,2,\ldots,N\}$, and every sufficiently large $m\in\mathbb{N}_0$ that

$$\Theta^{z,n}_m\approx{}_{\vartheta}\vartheta^{z,n}_m. \tag{60}$$

In the following two subsections (Subsection 2.7 and Subsection 2.8) we merge the above derivations to precisely formulate the proposed approximation algorithm, first, in a special case (Subsection 2.7) and, thereafter, in the general case (Subsection 2.8).

2.7 Description of the proposed approximation algorithm in a special case

In this subsection we describe the proposed approximation algorithm in the special case where the standard Euler-Maruyama scheme (cf., e.g., Kloeden & Platen [67] and Maruyama [82]) is the employed approximation scheme for discretizing (47) (cf. (56)) and where the plain vanilla stochastic gradient descent method with constant learning rate $\gamma \in (0, \infty)$ and batch size 1 is the employed minimization algorithm. A more general description of the proposed approximation algorithm, which allows to incorporate more sophisticated machine learning approximation techniques such as batch normalization (cf., for instance, Ioffe & Szegedy [56]) and the Adam optimizer (cf., for example, Kingma & Ba [65]), can be found in Subsection 2.8 below.

Framework 2.1 (Special case)**.** *Let $T, \gamma \in (0, \infty)$, $d, N, M \in \mathbb{N}$, $\varphi \in C^2(\mathbb{R}^d, \mathbb{R})$, $s \in \{3, 4, 5, \ldots\}$, $\nu = s(N+1)d(d+1)$, $t_0, t_1, \ldots, t_N \in [0, T]$ satisfy*

$$0 = t_0 < t_1 < \ldots < t_N = T, \tag{61}$$

let $\tau_0, \tau_1, \ldots, \tau_n \in [0, T]$ satisfy for every $n \in \{0, 1, \ldots, N\}$ that $\tau_n = T - t_{N-n}$, let $f \colon \mathbb{R}^d \times \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}$, $b \colon \mathbb{R}^d \times \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}$, $\mu \colon \mathbb{R}^d \to \mathbb{R}^d$, and $\sigma \colon \mathbb{R}^d \to \mathbb{R}^{d \times d}$ be continuous functions, let $(\Omega, \mathcal{F}, \mathbb{P}, (\mathcal{F}_t)_{t \in [0,T]})$ be a filtered probability space, for every function $z \colon [0, T] \times \mathbb{R}^d \to \mathbb{R}$ let $\xi^{z,m} \colon \Omega \to \mathbb{R}^d$, $m \in \mathbb{N}_0$, be i.i.d. $\mathcal{F}_0/\mathcal{B}(\mathbb{R}^d)$-measurable random variables, let $B^{z,m} \colon [0, T] \times \Omega \to \mathbb{R}^d$, $m \in \mathbb{N}_0$, be i.i.d. standard $(\mathcal{F}_t)_{t \in [0,T]}$-Brownian motions, let $\mathcal{Y}^{z,m} \colon \{0, 1, \ldots, N\} \times \Omega \to \mathbb{R}^d$, $m \in \mathbb{N}_0$, satisfy for every $m \in \mathbb{N}_0$, $n \in \{0, 1, \ldots, N-1\}$ that $\mathcal{Y}^{z,m}_0 = \xi^{z,m}$ and

$$\mathcal{Y}^{z,m}_{n+1} = \mathcal{Y}^{z,m}_n + \mu(\mathcal{Y}^{z,m}_n)\,(\tau_{n+1} - \tau_n) + \sigma(\mathcal{Y}^{z,m}_n)\,(B^{z,m}_{\tau_{n+1}} - B^{z,m}_{\tau_n}), \tag{62}$$

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

$$\mathcal{T}_d(x) = \big(\tanh(x_1), \tanh(x_2), \ldots, \tanh(x_d)\big), \tag{63}$$

for every $\theta = (\theta_1, \theta_2, \ldots, \theta_\nu) \in \mathbb{R}^\nu$, $k, l \in \mathbb{N}$, $v \in \mathbb{N}_0 = \{0\} \cup \mathbb{N}$ with $v + lk + l \leq \nu$ 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

$$A^{\theta,v}_{k,l}(x) = \left(\theta_{v+kl+1} + \left[\sum_{i=1}^{k} x_i\,\theta_{v+i}\right], \ldots, \theta_{v+kl+l} + \left[\sum_{i=1}^{k} x_i\,\theta_{v+(l-1)k+i}\right]\right), \tag{64}$$

for every function $z \colon [0, T] \times \mathbb{R}^d \to \mathbb{R}$ let $\mathbb{V}^{(z)}_n \colon \mathbb{R}^\nu \times \mathbb{R}^d \to \mathbb{R}$, $n \in \{0, 1, \ldots, N\}$, satisfy for every $n \in \{1, 2, \ldots, N\}$, $\theta \in \mathbb{R}^\nu$, $x \in \mathbb{R}^d$ that $\mathbb{V}^{(z)}_0(\theta, x) = \varphi(x)$ and

$$\begin{aligned} &\mathbb{V}^{(z)}_n(\theta, x) = \\ &\big(A^{\theta,(sn+s-1)d(d+1)}_{d,1} \circ \mathcal{T}_d \circ A^{\theta,(sn+s-2)d(d+1)}_{d,d} \circ \ldots \circ \mathcal{T}_d \circ A^{\theta,(sn+1)d(d+1)}_{d,d} \circ \mathcal{T}_d \circ A^{\theta,snd(d+1)}_{d,d}\big)(x), \end{aligned} \tag{65}$$

for every function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ *let* $\Theta^{z,n}\colon \mathbb{N}_0\times\Omega\to\mathbb{R}^\nu$, $n\in\{0,1,\ldots,N\}$, *be stochastic processes, for every function* $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ *and every* $n\in\{1,2,\ldots,N\}$, $m\in\mathbb{N}_0$ *let* $\phi^{z,n,m}\colon \mathbb{R}^\nu\times\Omega\to\mathbb{R}$ *satisfy for every* $\theta\in\mathbb{R}^\nu$, $\omega\in\Omega$ *that*

$$\begin{aligned}\phi^{z,n,m}(\theta,\omega) &= \Big[\mathbb{V}_n^{(z)}\big(\theta,\mathcal{Y}_{N-n}^{z,m}(\omega)\big) - \mathbb{V}_{n-1}^{(z)}(\Theta_M^{z,n-1}(\omega),\mathcal{Y}_{N-n+1}^{z,m}(\omega)) - (t_n-t_{n-1})\\ &\cdot f\big(\mathcal{Y}_{N-n+1}^{z,m}(\omega),\mathbb{V}_{n-1}^{(z)}(\Theta_M^{z,n-1}(\omega),\mathcal{Y}_{N-n+1}^{z,m}(\omega)),(\nabla_x\mathbb{V}_{n-1}^{(z)})(\Theta_M^{z,n-1}(\omega),\mathcal{Y}_{N-n+1}^{z,m}(\omega))\big)\\ &- b\big(\mathcal{Y}_{N-n+1}^{z,m}(\omega),\mathbb{V}_{n-1}^{(z)}(\Theta_M^{z,n-1}(\omega),\mathcal{Y}_{N-n+1}^{z,m}(\omega)),(\nabla_x\mathbb{V}_{n-1}^{(z)})(\Theta_M^{z,n-1}(\omega),\mathcal{Y}_{N-n+1}^{z,m}(\omega))\big)\\ &\cdot\big(z_{t_n}(\mathcal{Y}_{N-n+1}^{z,m}(\omega)) - z_{t_{n-1}}(\mathcal{Y}_{N-n+1}^{z,m}(\omega))\big)\Big]^2,\end{aligned} \tag{66}$$

for every function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ *and every* $n\in\{1,2,\ldots,N\}$, $m\in\mathbb{N}_0$ *let* $\Phi^{z,n,m}\colon \mathbb{R}^\nu\times\Omega\to\mathbb{R}^\nu$ *satisfy for every* $\theta\in\mathbb{R}^\nu$, $\omega\in\Omega$ *that* $\Phi^{z,n,m}(\theta,\omega) = (\nabla_\theta\phi^{z,n,m})(\theta,\omega)$, *and assume for every function* $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ *and every* $m\in\mathbb{N}_0$, $n\in\{1,2,\ldots,N\}$ *that*

$$\Theta_{m+1}^{z,n} = \Theta_m^{z,n} - \gamma\cdot\Phi^{z,n,m}(\Theta_m^{z,n}). \tag{67}$$

In the setting of Framework 2.1 we note that for every function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ and every $n\in\{1,2,\ldots,N\}$, $m\in\mathbb{N}_0$ it holds that

$$\begin{aligned}\Theta_{m+1}^{z,n} &= \Theta_m^{z,n} - 2\gamma\cdot(\nabla_\theta\mathbb{V}_n^{(z)})(\Theta_m^{z,n},\mathcal{Y}_{N-n}^{z,m})\cdot\Big[\mathbb{V}_n^{(z)}(\Theta_m^{z,n},\mathcal{Y}_{N-n}^{z,m}) - \mathbb{V}_{n-1}^{(z)}(\Theta_M^{z,n-1},\mathcal{Y}_{N-n+1}^{z,m})\\ &- f\big(\mathcal{Y}_{N-n+1}^{z,m},\mathbb{V}_{n-1}^{(z)}(\Theta_M^{z,n-1},\mathcal{Y}_{N-n+1}^{z,m}),(\nabla_x\mathbb{V}_{n-1}^{(z)})(\Theta_M^{z,n-1},\mathcal{Y}_{N-n+1}^{z,m})\big)(t_n-t_{n-1})\\ &- b\big(\mathcal{Y}_{N-n+1}^{z,m},\mathbb{V}_{n-1}^{(z)}(\Theta_M^{z,n-1},\mathcal{Y}_{N-n+1}^{z,m}),(\nabla_x\mathbb{V}_{n-1}^{(z)})(\Theta_M^{z,n-1},\mathcal{Y}_{N-n+1}^{z,m})\big)\\ &\cdot\big(z_{t_n}(\mathcal{Y}_{N-n+1}^{z,m}) - z_{t_{n-1}}(\mathcal{Y}_{N-n+1}^{z,m})\big)\Big]\end{aligned} \tag{68}$$

(cf. (48), (59), (66), and (67) above). Moreover, in the setting of Framework 2.1 we think under suitable hypothesis for sufficiently large $N,M\in\mathbb{N}$, for sufficiently small $\gamma\in(0,\infty)$, for every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$, and for every $n\in\{0,1,\ldots,N\}$, $x\in\mathbb{R}^d$ of $\mathbb{V}_n^{(z)}(\Theta_M^{z,n},x)\colon\Omega\to\mathbb{R}$ as a suitable approximation

$$\mathbb{V}_n^{(z)}(\Theta_M^{z,n},x)\approx\mathbb{E}\big[X_{t_n}(x)\,|\,Z=z\big] \tag{69}$$

of $\mathbb{E}[X_{t_n}(x)\,|\,Z=z]$ where $X\colon[0,T]\times\mathbb{R}^d\times\Omega\to\mathbb{R}$ is a random field which satisfies for every $t\in[0,T]$, $x\in\mathbb{R}^d$ that $X_t(x)\colon\Omega\to\mathbb{R}$ is $\mathcal{F}_t/\mathcal{B}(\mathbb{R})$-measurable, which satisfies for every $\omega\in\Omega$ that $(X_t(x,\omega))_{(t,x)\in[0,T]\times\mathbb{R}^d}\in C^{0,2}([0,T]\times\mathbb{R}^d,\mathbb{R})$ has at most polynomially growing partial derivatives, and which satisfies that for every $t\in[0,T]$, $x\in\mathbb{R}^d$ it holds $\mathbb{P}$-a.s. that

$$\begin{aligned}X_t(x) &= \varphi(x) + \int_0^t f\big(x,X_s(x),(\nabla X_s)(x)\big)\,ds + \int_0^t b\big(x,X_s(x),(\nabla X_s)(x)\big)\,dZ_s(x)\\ &+ \int_0^t\Big[\tfrac{1}{2}\operatorname{Trace}\big(\sigma(x)[\sigma(x)]^*(\operatorname{Hess}X_s)(x)\big) + \big\langle\mu(x),(\nabla X_s)(x)\big\rangle_{\mathbb{R}^d}\Big]\,ds\end{aligned} \tag{70}$$

where $Z\colon[0,T]\times\mathbb{R}^d\times\Omega\to\mathbb{R}$ is a sufficiently regular random field (cf. (1), (13), (14), (41), and (42)).

2.8 Description of the proposed approximation algorithm in the general case

In this subsection we present in Framework 2.2 a general approximation algorithm which includes the proposed approximation algorithm derived in Subsections 2.1–2.7 above as a special case. Compared to Framework 2.1, Framework 2.2 allows to incorporate other minimization algorithms than just the plain vanilla stochastic gradient descent method (see, e.g., (67) in Framework 2.1 in Subsection 2.7 above) such as, for example, the Adam optimizer (cf. Kingma & Ba [65]). Furthermore, Framework 2.2 also allows to incorporate more advanced machine learning techniques like batch normalization (cf. Ioffe & Szegedy [56] and (74) below).

Framework 2.2 (General case)**.** *Let $T \in (0,\infty)$, $M, N, d, \delta, \varrho, \nu, \varsigma \in \mathbb{N}$, $(J_m)_{m\in\mathbb{N}_0} \subseteq \mathbb{N}$, $t_0, t_1, \ldots, t_N \in [0,T]$ satisfy $0 = t_0 < t_1 < \ldots < t_N = T$, let $\tau_0, \tau_1, \ldots, \tau_n \in [0,T]$ satisfy for every $n \in \{0,1,\ldots,N\}$ that $\tau_n = T - t_{N-n}$, let $f\colon \mathbb{R}^d \times \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}$, $b\colon \mathbb{R}^d \times \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^\delta$, $H\colon [0,T]^2 \times \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$, and $\mathcal{H}_n\colon \mathbb{R}^d \times \mathbb{R} \times \mathbb{R}^d \times \mathbb{R}^\delta \to \mathbb{R}$, $n \in \{1,2,\ldots,N\}$, be functions, let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space with a normal filtration $(\mathcal{F}_t)_{t\in[0,T]}$, for every function $z\colon [0,T] \times \mathbb{R}^d \to \mathbb{R}^\delta$ and every $n \in \{1,2,\ldots,N\}$ let $B^{z,n,m,j}\colon [0,T] \times \Omega \to \mathbb{R}^d$, $m \in \mathbb{N}_0$, $j \in \mathbb{N}$, be i.i.d. standard $(\mathcal{F}_t)_{t\in[0,T]}$-Brownian motions, for every function $z\colon [0,T] \times \mathbb{R}^d \to \mathbb{R}^\delta$ and every $n \in \{1,2,\ldots,N\}$ let $\xi^{z,n,m,j}\colon \Omega \to \mathbb{R}^d$, $m \in \mathbb{N}_0$, $j \in \mathbb{N}$, be i.i.d. $\mathcal{F}_0/\mathcal{B}(\mathbb{R}^d)$-measurable random variables, for every function $z\colon [0,T]\times\mathbb{R}^d \to \mathbb{R}^\delta$ let $\mathbb{V}_n^{z,j,\mathbf{s}}\colon \mathbb{R}^\nu \times \mathbb{R}^d \to \mathbb{R}$, $j \in \mathbb{N}$, $s \in \mathbb{R}^\varsigma$, $n \in \{0,1,\ldots,N\}$, be functions, for every function $z\colon [0,T] \times \mathbb{R}^d \to \mathbb{R}^\delta$ and every $n \in \{1,2,\ldots,N\}$, $m \in \mathbb{N}_0$, $j \in \mathbb{N}$ let $\mathcal{Y}^{z,n,m,j}\colon \{0,1,\ldots,N\} \times \Omega \to \mathbb{R}^d$ be a stochastic process which satisfies for every $k \in \{0,1,\ldots,N-1\}$ that $\mathcal{Y}_0^{z,n,m,j} = \xi^{z,n,m,j}$ and*

$$\mathcal{Y}_{k+1}^{z,n,m,j} = H(\tau_{k+1}, \tau_k, \mathcal{Y}_k^{z,n,m,j}, B_{\tau_{k+1}}^{z,n,m,j} - B_{\tau_k}^{z,n,m,j}), \tag{71}$$

for every function $z\colon [0,T]\times\mathbb{R}^d \to \mathbb{R}^\delta$ let $\Theta^{z,n}\colon \mathbb{N}_0 \times \Omega \to \mathbb{R}^\nu$, $n \in \{0,1,\ldots N\}$, be stochastic processes, for every function $z\colon [0,T] \times \mathbb{R}^d \to \mathbb{R}^\delta$ and every $n \in \{1,2,\ldots,N\}$, $m \in \mathbb{N}_0$, $\mathbf{s} \in \mathbb{R}^\varsigma$ let $\phi^{z,n,m,\mathbf{s}}\colon \mathbb{R}^\nu \times \Omega \to \mathbb{R}$ satisfy for every $\theta \in \mathbb{R}^\nu$, $\omega \in \Omega$ that

$$\begin{aligned}
\phi^{z,n,m,\mathbf{s}}(\theta,\omega) &= \frac{1}{J_m} \sum_{j=1}^{J_m} \Bigg[\mathbb{V}_n^{z,j,\mathbf{s}}\big(\theta, \mathcal{Y}_{N-n}^{z,n,m,j}(\omega)\big) \\
&- \mathcal{H}_n\Big(\mathcal{Y}_{N-n+1}^{z,n,m,j}(\omega), \mathbb{V}_{n-1}^{z,j,\mathbf{s}}\big(\Theta_M^{z,n-1}(\omega), \mathcal{Y}_{N-n+1}^{z,n,m,j}(\omega)\big), \\
&\qquad (\nabla_x \mathbb{V}_{n-1}^{z,j,\mathbf{s}})\big(\Theta_M^{z,n-1}(\omega), \mathcal{Y}_{N-n+1}^{z,n,m,j}(\omega)\big), z_{t_n}\big(\mathcal{Y}_{N-n+1}^{z,n,m,j}(\omega)\big) - z_{t_{n-1}}\big(\mathcal{Y}_{N-n+1}^{z,n,m,j}(\omega)\big)\Big)\Bigg]^2,
\end{aligned} \tag{72}$$

for every function $z\colon [0,T] \times \mathbb{R}^d \to \mathbb{R}^\delta$ and every $n \in \{1,2,\ldots,N\}$, $m \in \mathbb{N}_0$, $\mathbf{s} \in \mathbb{R}^\varsigma$ let $\Phi^{z,n,m,\mathbf{s}}\colon \mathbb{R}^\nu \times \Omega \to \mathbb{R}^\nu$ satisfy for every $\omega \in \Omega$, $\theta \in \{\eta \in \mathbb{R}^\nu\colon \phi^{z,n,m,\mathbf{s}}(\cdot,\omega)\colon \mathbb{R}^\nu \to \mathbb{R}$ is differentiable at $\eta\}$ that

$$\Phi^{z,n,m,\mathbf{s}}(\theta,\omega) = (\nabla_\theta \phi^{z,n,m,\mathbf{s}})(\theta,\omega), \tag{73}$$

for every function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}^\delta$ *let* $\mathcal{S}^{z,n}\colon \mathbb{R}^\varsigma\times\mathbb{R}^\nu\times(\mathbb{R}^d)^{\{0,1,\ldots,N\}\times\mathbb{N}}\to\mathbb{R}^\varsigma$, $n\in\{1,2,\ldots,N\}$, *be functions, for every function* $z\colon[0,T]\times\mathbb{R}^d\to\mathbb{R}^\delta$ *and every* $n\in\{1,2,\ldots,N\}$, $m\in\mathbb{N}_0$ *let* $\psi_m^{z,n}\colon\mathbb{R}^\varrho\to\mathbb{R}^\nu$ *and* $\Psi_m^{z,n}\colon\mathbb{R}^\varrho\times\mathbb{R}^\nu\to\mathbb{R}^\varrho$ *be functions, for every function* $z\colon[0,T]\times\mathbb{R}^d\to\mathbb{R}^\delta$ *and every* $n\in\{1,2,\ldots,N\}$ *let* $\mathbb{S}^{z,n}\colon\mathbb{N}_0\times\Omega\to\mathbb{R}^\varsigma$ *and* $\Xi^{z,n}\colon\mathbb{N}_0\times\Omega\to\mathbb{R}^\varrho$ *be stochastic processes which satisfy for every* $m\in\mathbb{N}_0$ *that*

$$\mathbb{S}_{m+1}^{z,n}=\mathcal{S}^{z,n}\big(\mathbb{S}_m^{z,n},\Theta_m^{z,n},(\mathcal{Y}_k^{z,n,m,i})_{(k,i)\in\{0,1,\ldots,N\}\times\mathbb{N}}\big),\tag{74}$$

$$\Xi_{m+1}^{z,n}=\Psi_m^{z,n}(\Xi_m^{z,n},\Phi^{z,n,m,\mathbb{S}_{m+1}^{z,n}}(\Theta_m^{z,n}))\quad\text{and}\quad\Theta_{m+1}^{z,n}=\Theta_m^{z,n}-\psi_m^{z,n}(\Xi_{m+1}^{z,n}).\tag{75}$$

In the setting of Framework 2.2 we think under suitable hypothesis for sufficiently large $N\in\mathbb{N}$, for every sufficiently regular function $z\colon[0,T]\times\mathbb{R}^d\to\mathbb{R}^\delta$, for every sufficiently large $m\in\mathbb{N}$, and for every $n\in\{0,1,\ldots N\}$, $x\in\mathbb{R}^d$ of $\mathbb{V}_n^{z,1,\mathbb{S}_m^{z,n}}(\Theta_m^{z,n},x)\colon\Omega\to\mathbb{R}$ as a suitable approximation

$$\mathbb{V}_n^{z,1,\mathbb{S}_m^{z,n}}(\Theta_m^{z,n},x)\approx\mathbb{E}\big[X_{t_n}(x)\,\big|\,Z=z\big]\tag{76}$$

of $\mathbb{E}[X_{t_n}(x)\,|\,Z=z]$ where $X\colon[0,T]\times\mathbb{R}^d\times\Omega\to\mathbb{R}$ is a random field which satisfies for every $t\in[0,T]$, $x\in\mathbb{R}^d$ that $X_t(x)\colon\Omega\to\mathbb{R}$ is $\mathcal{F}_t/\mathcal{B}(\mathbb{R})$-measurable, which satisfies for every $\omega\in\Omega$ that $(X_t(x,\omega))_{(t,x)\in[0,T]\times\mathbb{R}^d}\in C^{0,2}([0,T]\times\mathbb{R}^d,\mathbb{R})$ has at most polynomially growing partial derivatives, and which satisfies that for every $t\in[0,T]$, $x\in\mathbb{R}^d$ it holds $\mathbb{P}$-a.s. that

$$\begin{aligned}X_t(x)&=\varphi(x)+\int_0^t f\big(x,X_s(x),(\nabla X_s)(x)\big)\,ds+\int_0^t\big\langle b\big(x,X_s(x),(\nabla X_s)(x)\big),dZ_s(x)\big\rangle_{\mathbb{R}^\delta}\\&\quad+\int_0^t\Big[\tfrac{1}{2}\operatorname{Trace}\big(\sigma(x)[\sigma(x)]^*(\operatorname{Hess}X_s)(x)\big)+\big\langle\mu(x),(\nabla X_s)(x)\big\rangle_{\mathbb{R}^d}\Big]\,ds\end{aligned}\tag{77}$$

where $\varphi\colon\mathbb{R}^d\to\mathbb{R}$ is a continuous function, where $\mu\colon\mathbb{R}^d\to\mathbb{R}^d$ is a sufficiently regular function, where $\sigma\colon\mathbb{R}^d\to\mathbb{R}^{d\times d}$ is a sufficiently regular and sufficiently non-degenerate function, and where $Z\colon[0,T]\times\mathbb{R}^d\times\Omega\to\mathbb{R}^\delta$ is a sufficiently regular random field (cf. (1), (13), (14), (41), and (42)).

3 Examples

In this section we depict the performance of the algorithm proposed in Subsection 2.8 by providing numerical simulations for four example SPDEs. More precisely, we apply the proposed approximation algorithm to stochastic heat equations with additive noise (cf. Subsection 3.1 below), to stochastic heat equations with multiplicative noise (cf. Subsection 3.2 below), to stochastic Black–Scholes equations with multiplicative noise (cf. Subsection 3.3 below), and to Zakai equations (cf. Subsection 3.4 below). In each of these numerical simulations we employ the general approximation method in Subsection 2.8 in conjunction

with the Milstein approximation scheme (cf. [67, Section 10.3]) and the Adam optimizer (cf. (79) and (80) in Framework 3.1 below and Kingma & Ba [65]) with mini-batches with 64 samples in each iteration step (see Framework 3.1 for a detailed description).

For every sufficiently regular function $z\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ we employ in our implementation N fully-connected feedforward neural networks to represent $\mathbb{V}_n^{z,j,\mathbf{s}}(\theta,x)$ for $n\in\{1,2,\ldots,N\}$, $j\in\{1,2,\ldots,64\}$, $s\in\mathbb{R}^{\varsigma}$, $\theta\in\mathbb{R}^{\nu}$, $x\in\mathbb{R}^d$. Each of these neural networks consists of 4 layers (1 input layer [d-dimensional], 2 hidden layers [both $d+50$ -dimensional], and 1 output layer [1-dimensional]). We employ the tanh activation function as our activation function for the hidden variables. We also employ batch normalization (BN) (see Ioffe & Szegedy [56]) just before the first affine linear transformation (batch normalization for the input) as well as just before every application of the multidimensional version of the tanh activation function (batch normalization for the hidden layers just before activation). All the weights in the network are initialized using a normal or a uniform distribution. Each of the numerical experiments presented below is performed in PYTHON using TENSORFLOW on a NVIDIA GeForce RTX 2080 Ti GPU. The underlying system is an AMD Ryzen 9 3950X CPU with 64 GB DDR4 memory running Tensorflow 2.1 on Ubuntu 19.10. We also refer to Section 4 below for the PYTHON source codes associated to the numerical simulations in Subsections 3.1–3.4 below.

Framework 3.1. *Assume Framework 2.2, let* $\nu=(N+1)[(d+50)(d+1)+(d+50)(d+51)+(d+51)]$ *(cf. E et al. [30, Remark 4.1] and the second paragraph of this section),* $\varepsilon\in(0,\infty)$, $\beta_1=\frac{9}{10}$, $\beta_2=\frac{999}{1000}$, $(\gamma_m)_{m\in\mathbb{N}_0}\subseteq(0,\infty)$, *let* $\mathrm{Pow}_r\colon\mathbb{R}^{\nu}\to\mathbb{R}^{\nu}$, $r\in(0,\infty)$, *satisfy for every* $r\in(0,\infty)$, $x=(x_1,x_2,\ldots,x_{\nu})\in\mathbb{R}^{\nu}$ *that* $\mathrm{Pow}_r(x)=(|x_1|^r,|x_2|^r,\ldots,|x_{\nu}|^r)$, *let* $\varphi\colon\mathbb{R}^d\to\mathbb{R}$, $\mu=(\mu_1,\mu_2,\ldots,\mu_d)\colon\mathbb{R}^d\to\mathbb{R}^d$, *and* $\sigma\colon\mathbb{R}^d\to\mathbb{R}^{d\times d}$ *be functions, let* $X\colon[0,T]\times\mathbb{R}^d\times\Omega\to\mathbb{R}$ *and* $Z\colon[0,T]\times\mathbb{R}^d\times\Omega\to\mathbb{R}^{\delta}$ *be random fields, assume for every* $t\in[0,T]$, $x\in\mathbb{R}^d$ *that* $X_t(x)\colon\Omega\to\mathbb{R}$ *is* $\mathcal{F}_t/\mathcal{B}(\mathbb{R})$*-measurable, assume for every* $x\in\mathbb{R}^d$ *that* $(Z_t(x))_{t\in[0,T]}\colon[0,T]\times\Omega\to\mathbb{R}^{\delta}$ *is an* $(\mathcal{F}_t)_{t\in[0,T]}$*-Itô process, assume for every* $\omega\in\Omega$ *that* $(X_t(x,\omega))_{(t,x)\in[0,T]\times\mathbb{R}^d}\in C^{0,2}([0,T]\times\mathbb{R}^d,\mathbb{R})$ *has at most polynomially growing partial derivatives, assume that for every* $t\in[0,T]$, $x\in\mathbb{R}^d$ *it holds* $\mathbb{P}$*-a.s. that*

$$\begin{aligned}X_t(x)&=\varphi(x)+\int_0^t f\big(x,X_s(x),(\nabla X_s)(x)\big)\,ds+\int_0^t\big\langle b\big(x,X_s(x),(\nabla X_s)(x)\big),dZ_s(x)\big\rangle_{\mathbb{R}^{\delta}}\\&\quad+\int_0^t\Big[\tfrac{1}{2}\,\mathrm{Trace}\big(\sigma(x)[\sigma(x)]^*(\mathrm{Hess}\,X_s)(x)\big)+\big\langle\mu(x),(\nabla X_s)(x)\big\rangle_{\mathbb{R}^d}\Big]\,ds,\end{aligned}\tag{78}$$

assume for every $n\in\{1,2,\ldots,N\}$, $m\in\mathbb{N}_0$, $i\in\{0,1,\ldots,N\}$ *that* $J_m=64$, $t_i=\frac{iT}{N}$, *and* $\varrho=2\nu$, *and assume for every* $m\in\mathbb{N}_0$, $x=(x_1,x_2,\ldots,x_{\nu})$, $y=(y_1,y_2,\ldots,y_{\nu})\in\mathbb{R}^{\nu}$, $\eta=(\eta_1,\eta_2,\ldots,\eta_{\nu})\in\mathbb{R}^{\nu}$ *that*

$$\Psi_m^n(x,y,\eta)=(\beta_1x+(1-\beta_1)\eta,\beta_2y+(1-\beta_2)\,\mathrm{Pow}_2(\eta))\tag{79}$$

and

$$\psi_m^n(x,y) = \left(\left[\sqrt{\tfrac{|y_1|}{1-(\beta_2)^m}} + \varepsilon \right]^{-1} \frac{\gamma_m x_1}{1-(\beta_1)^m}, \ldots, \left[\sqrt{\tfrac{|y_\nu|}{1-(\beta_2)^m}} + \varepsilon \right]^{-1} \frac{\gamma_m x_\nu}{1-(\beta_1)^m} \right). \tag{80}$$

Note that (79) and (80) in Framework 3.1 describe the Adam optimizer (cf. Kingma & Ba [65], e.g., E et al. [30, (32)–(33) in Section 4.2 and (90)–(91) in Section 5.2], and line 108–110 in PYTHON code 1 in Section 4 below).

3.1 Stochastic heat equations with additive noise

In this subsection we apply the approximation algorithm in Framework 2.2 to the stochastic heat equations with additive noise in (81) below.

Assume Framework 3.1, assume that $T = 1$, $N = 5$, $M = 8000$, $d \in \{1, 5, 10, 20, 50\}$, $\delta = 1$, and $\varepsilon = 10^{-8}$, let $W \colon [0,T] \times \Omega \to \mathbb{R}$ be a standard $(\mathcal{F}_t)_{t\in[0,T]}$-Brownian motion, and assume for every $s, t \in [0,T]$, $x, w \in \mathbb{R}^d$, $u \in \mathbb{R}$, $m \in \mathbb{N}_0$ that $H(t,s,x,w) = x + \sqrt{2}w$, $f(x,u,w) = 0$, $b(x,u,w) = 1$, $\mu(x) = 0$, $\sigma(x)w = \sqrt{2}w$, $\varphi(x) = \|x\|_{\mathbb{R}^d}^2$, $Z_t(x) = W_t$, $\gamma_m = 10^{-1}\mathbb{1}_{[0,2000]}(m) + 10^{-2}\mathbb{1}_{(2000,4000]}(m) + 10^{-3}\mathbb{1}_{(4000,6000]}(m) + 10^{-4}\mathbb{1}_{(6000,8000]}(m)$, and $\mathcal{H}_n(x,u,w,z) = u + z$ (cf., for instance, Kloeden & Platen [67, Section 10.3]). Note that (78) ensures that for every $t \in [0,T]$, $x \in \mathbb{R}^d$ it holds $\mathbb{P}$-a.s. that

$$X_t(x) = \|x\|_{\mathbb{R}^d}^2 + \int_0^t (\Delta_x X_s)(x)\, ds + W_t. \tag{81}$$

Next we depict our numerical simulation results for the stochastic heat equations with additive noise in (81). In Table 1 we present numerical approximations for the relative L^2-errors $\left(\mathbb{E}\left[|X_T(0)|^{-2}|\mathbb{V}_N^{0,1,\mathbb{S}_m^{0,N}}(\Theta_m^{0,N},0) - X_T(0)|^2\right]\right)^{1/2}$ for $d \in \{1,5,10,20,50\}$ (cf. (75) and (76)). In our approximative computations for the relative L^2-errors, the exact solutions of the SPDEs in (81) have been approximately computed by means of the well-known result in Lemma 3.2 below.

Lemma 3.2. *Let $T \in (0,\infty)$, $d \in \mathbb{N}$, let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, let $W \colon [0,T] \times \Omega \to \mathbb{R}$ be a stochastic process with continuous sample paths, and let $X \colon [0,T] \times \mathbb{R}^d \times \Omega \to \mathbb{R}$ satisfy for every $t \in [0,T]$, $x \in \mathbb{R}^d$ that*

$$X_t(x) = \|x\|_{\mathbb{R}^d}^2 + 2td + W_t. \tag{82}$$

Then

(i) it holds for every $\omega \in \Omega$ that $([0,T] \times \mathbb{R}^d \ni (t,x) \mapsto X_t(x,\omega) \in \mathbb{R}) \in C^{0,2}([0,T] \times \mathbb{R}^d, \mathbb{R})$ and

(ii) it holds for every $t \in [0,T]$, $x \in \mathbb{R}^d$ that

$$X_t(x) = \|x\|_{\mathbb{R}^d}^2 + \int_0^t (\Delta_x X_s)(x)\, ds + W_t. \tag{83}$$

Proof of Lemma 3.2. Throughout this proof let $\mathfrak{C} \in \mathbb{R}^{d\times d}$ satisfy for every $x \in \mathbb{R}^d$ that $\mathfrak{C}x = 2x$ and let $v\colon [0,T] \times \mathbb{R}^d \to \mathbb{R}$ satisfy for every $t \in [0,T]$, $x \in \mathbb{R}^d$ that

$$v(t,x) = \|x\|_{\mathbb{R}^d}^2 + t\,\mathrm{Trace}(\mathfrak{C}). \tag{84}$$

Note that Lemma 3.3 (applied with $T \curvearrowleft T$, $d \curvearrowleft d$, $\mathfrak{C} \curvearrowleft \mathfrak{C}$, $u \curvearrowleft v$ in the notation of Lemma 3.3) and (84) ensure that

(a) it holds that $v \in C^\infty([0,T] \times \mathbb{R}^d, \mathbb{R})$ and

(b) it holds for every $t \in [0,T]$, $x \in \mathbb{R}^d$ that $v(0,x) = \|x\|_{\mathbb{R}^d}^2$ and

$$(\tfrac{\partial}{\partial t} v)(t,x) = \tfrac{1}{2}\,\mathrm{Trace}\big(\mathfrak{C}(\mathrm{Hess}_x\, v)(t,x)\big) = (\Delta_x v)(t,x). \tag{85}$$

Moreover, note that (84) and the fact that $\mathrm{Trace}(\mathfrak{C}) = 2d$ prove that for every $t \in [0,T]$, $x \in \mathbb{R}^d$ it holds that

$$v(t,x) = \|x\|_{\mathbb{R}^d}^2 + 2td. \tag{86}$$

Combining this, item (a), and (82) hence ensures that for every $\omega \in \Omega$ it holds that $([0,T] \times \mathbb{R}^d \ni (t,x) \mapsto X_t(x,\omega) \in \mathbb{R}) \in C^{0,\infty}([0,T] \times \mathbb{R}^d, \mathbb{R})$. This establishes item (i). Moreover, note that (86), the fundamental theorem of calculus, (82), and items (a) and (b) ensure that for every $t \in [0,T]$, $x \in \mathbb{R}^d$ it holds that

$$\begin{aligned} X_t(x) &= \|x\|_{\mathbb{R}^d}^2 + 2td + W_t = v(t,x) + W_t \\ &= \|x\|_{\mathbb{R}^d}^2 + \int_0^t (\tfrac{\partial}{\partial s} v)(s,x)\,ds + W_t \\ &= \|x\|_{\mathbb{R}^d}^2 + \int_0^t (\Delta_x v)(s,x)\,ds + W_t = \|x\|_{\mathbb{R}^d}^2 + \int_0^t (\Delta_x X_s)(x)\,ds + W_t. \end{aligned} \tag{87}$$

This completes the proof of Lemma 3.2. □

3.2 Stochastic heat equations with multiplicative noise

In this subsection we apply the approximation algorithm in Framework 2.2 to the stochastic heat equations with multiplicative noise in (88) below.

Assume Framework 3.1, assume that $T = 0.5$, $N = 25$, $M = 12000$, $d \in \{1,5,10,20,50\}$, $\delta = 1$, and $\varepsilon = 10^{-8}$, let $W\colon [0,T] \times \Omega \to \mathbb{R}$ be a standard $(\mathcal{F}_t)_{t\in[0,T]}$-Brownian motion, and assume for every $s,t \in [0,T]$, $x,w \in \mathbb{R}^d$, $u,z \in \mathbb{R}$, $n \in \{1,2,\ldots,N\}$, $m \in \mathbb{N}_0$ that $H(t,s,x,w) = x + \sqrt{2}w$, $f(x,u,w) = 0$, $b(x,u,w) = u$, $\mu(x) = 0$, $\sigma(x)w = \sqrt{2}w$, $\varphi(x) = \|x\|_{\mathbb{R}^d}^2$, $Z_t(x) = W_t$, $\gamma_m = 10^{-1}\mathbb{1}_{[0,5000]}(m) + 10^{-2}\mathbb{1}_{(5000,7000]}(m) + 10^{-3}\mathbb{1}_{(7000,10000]}(m) + 10^{-4}\mathbb{1}_{(10000,12000]}(m)$, and $\mathcal{H}_n(x,u,w,z) = u\big(1 + z + \frac{1}{2}z^2 - \frac{1}{2}(t_n - t_{n-1})\big)$ (cf., for instance,

d	Result of the approx. algorithm	Runtime in seconds	Reference solution	Relative pathwise error	Relative L^2-error
1	2.018	80.04	2.035	0.0084	
1	4.590	79.26	4.561	0.0064	
1	3.039	79.07	3.020	0.0063	0.0060
1	2.323	78.81	2.322	0.0006	
1	1.482	79.18	1.489	0.0053	
5	9.529	79.94	9.550	0.0022	
5	9.903	80.55	9.922	0.0019	
5	10.764	80.44	10.701	0.0059	0.0040
5	11.682	80.43	11.624	0.0050	
5	9.259	79.54	9.230	0.0032	
10	18.841	80.10	18.970	0.0068	
10	21.157	80.08	21.078	0.0038	
10	20.899	80.27	20.766	0.0064	0.0050
10	21.763	80.40	21.734	0.0013	
10	20.105	80.91	20.009	0.0048	
20	40.119	79.94	40.183	0.0016	
20	40.158	80.14	40.024	0.0034	
20	40.316	80.19	40.166	0.0037	0.0031
20	40.032	80.26	39.891	0.0035	
20	39.159	79.87	39.059	0.0026	
50	98.139	79.36	98.762	0.0063	
50	100.318	79.79	101.261	0.0093	
50	100.458	80.71	100.997	0.0053	0.0063
50	99.777	80.84	100.196	0.0042	
50	99.812	80.01	100.330	0.0052	

Table 1: Numerical simulations for the stochastic heat equations with additive noise in (81).

Kloeden & Platen [67, Section 10.3]). Note that (78) ensures that for every $t \in [0,T]$, $x \in \mathbb{R}^d$ it holds $\mathbb{P}$-a.s. that

$$X_t(x) = \|x\|_{\mathbb{R}^d}^2 + \int_0^t (\Delta_x X_s)(x)\, ds + \int_0^t X_s(x)\, dW_s. \tag{88}$$

Next we depict our numerical simulation results for the stochastic heat equations with multiplicative noise in (88). In Table 2 we present numerical approximations for the relative L^2-errors $\big(\mathbb{E}\big[|X_T(0)|^{-2}|\mathbb{V}_N^{0,1,\mathbb{S}_m^{0,N}}(\Theta_m^{0,N},0) - X_T(0)|^2\big]\big)^{1/2}$ for $d \in \{1,5,10,20,50\}$ (cf. (75) and (76)). In our approximative computations for the relative L^2-errors, the exact solutions of the SPDEs in (88) have been approximately computed by means of the well-known result in Lemma 3.4 below. Our proof of Lemma 3.4 employs the well-known result in Lemma 3.3 below.

Lemma 3.3. *Let $T \in (0,\infty)$, $d \in \mathbb{N}$, let $\mathfrak{C} \in \mathbb{R}^{d\times d}$ be a strictly positive definite symmetric matrix, and let $u\colon [0,T]\times\mathbb{R}^d \to \mathbb{R}$ satisfy for every $t \in [0,T]$, $x \in \mathbb{R}^d$ that*

$$u(t,x) = \|x\|_{\mathbb{R}^d}^2 + t\,\mathrm{Trace}(\mathfrak{C}). \tag{89}$$

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 $u(0,x) = \|x\|_{\mathbb{R}^d}^2$ and

$$(\tfrac{\partial}{\partial t}u)(t,x) = \tfrac{1}{2}\,\mathrm{Trace}\big(\mathfrak{C}(\mathrm{Hess}_x\, u)(t,x)\big). \tag{90}$$

Proof of Lemma 3.3. Observe that, e.g., [3, Lemma 3.2] (applied with $C \curvearrowleft \mathfrak{C}$ in the notation of [3, Lemma 3.2]) establishes items (i) and (ii). This completes the proof of Lemma 3.3. □

Lemma 3.4. *Let $T \in (0,\infty)$, $d \in \mathbb{N}$, let $(\Omega,\mathcal{F},\mathbb{P})$ be a probability space, let $W\colon [0,T]\times\Omega\to\mathbb{R}$ be a standard Brownian motion, let $\mathfrak{C} \in \mathbb{R}^{d\times d}$ be a strictly positive definite symmetric matrix, and let $X\colon [0,T]\times\mathbb{R}^d\times\Omega\to\mathbb{R}$ satisfy for every $t \in [0,T]$, $x \in \mathbb{R}^d$ that*

$$X_t(x) = \exp\big(W_t - \tfrac{t}{2}\big)\big(t\,\mathrm{Trace}(\mathfrak{C}) + \|x\|_{\mathbb{R}^d}^2\big). \tag{91}$$

Then for every $t \in [0,T]$, $x \in \mathbb{R}^d$ it holds $\mathbb{P}$-a.s. that

$$X_t(x) = \|x\|_{\mathbb{R}^d}^2 + \int_0^t \tfrac{1}{2}\,\mathrm{Trace}\big(\mathfrak{C}(\mathrm{Hess}\, X_s)(x)\big)\, ds + \int_0^t X_s(x)\, dW_s. \tag{92}$$

Proof of Lemma 3.4. Throughout this proof let $v\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ satisfy for every $t\in[0,T]$, $x\in\mathbb{R}^d$ that

$$v(t,x)=\|x\|_{\mathbb{R}^d}^2+t\,\mathrm{Trace}(\mathfrak{C}). \tag{93}$$

Observe that Itô's formula and (93) ensure that for every $t\in[0,T]$, $x\in\mathbb{R}^d$ it holds $\mathbb{P}$-a.s. that

$$\begin{aligned}
X_t(x)&=\exp\!\big(W_t-\tfrac{t}{2}\big)\big(t\,\mathrm{Trace}(\mathfrak{C})+\|x\|_{\mathbb{R}^d}^2\big)=\exp\!\big(W_t-\tfrac{t}{2}\big)\,v(t,x)\\
&=\|x\|_{\mathbb{R}^d}^2+\int_0^t\exp\!\big(W_s-\tfrac{s}{2}\big)\,(\tfrac{\partial}{\partial s}v)(s,x)\,ds+\int_0^t\exp\!\big(W_s-\tfrac{s}{2}\big)\,v(s,x)\,dW_s\\
&\quad+\int_0^t\big[-\tfrac12\big]\exp\!\big(W_s-\tfrac{s}{2}\big)\,v(s,x)\,ds+\tfrac12\int_0^t\exp\!\big(W_s-\tfrac{s}{2}\big)\,v(s,x)\,ds\\
&=\|x\|_{\mathbb{R}^d}^2+\int_0^t\exp\!\big(W_s-\tfrac{s}{2}\big)\,(\tfrac{\partial}{\partial s}v)(s,x)\,ds+\int_0^t\exp\!\big(W_s-\tfrac{s}{2}\big)\,v(s,x)\,dW_s.
\end{aligned}\tag{94}$$

Lemma 3.3 hence ensures that for every $t\in[0,T]$, $x\in\mathbb{R}^d$ it holds $\mathbb{P}$-a.s. that

$$\begin{aligned}
X_t(x)&=\|x\|_{\mathbb{R}^d}^2+\int_0^t\exp\!\big(W_s-\tfrac{s}{2}\big)\,\tfrac12\,\mathrm{Trace}\big(\mathfrak{C}(\mathrm{Hess}_x\,v)(s,x)\big)\,ds+\int_0^t X_s(x)\,dW_s\\
&=\|x\|_{\mathbb{R}^d}^2+\int_0^t\tfrac12\,\mathrm{Trace}\big(\mathfrak{C}(\mathrm{Hess}\,X_s)(x)\big)\,ds+\int_0^t X_s(x)\,dW_s.
\end{aligned}\tag{95}$$

This completes the proof of Lemma 3.4. □

3.3 Stochastic Black–Scholes equations with multiplicative noise

In this subsection we apply the approximation algorithm in Framework 2.2 to the stochastic Black–Scholes equations with multiplicative noise in (97) below.

Assume Framework 3.1, assume that $T=0.5$, $N=20$, $M=10000$, $d\in\{1,5,10,20\}$, $\delta=1$, and $\varepsilon=10^{-8}$, let $r=\frac{1}{50}$, $\mu_1=\frac{\sin(d)+1}{d}$, $\mu_2=\frac{\sin(2d)+1}{d}$, …, $\mu_d=\frac{\sin(d^2)+1}{d}$, $\sigma_1=\frac{1}{4d}$, $\sigma_2=\frac{2}{4d}$, …, $\sigma_d=\frac{d}{4d}$, let $W\colon[0,T]\times\Omega\to\mathbb{R}$ be a standard $(\mathcal{F}_t)_{t\in[0,T]}$-Brownian motion, and assume for every $s,t\in[0,T]$, $x=(x_1,x_2,\ldots,x_d)$, $w=(w_1,w_2,\ldots,w_d)\in\mathbb{R}^d$, $u\in\mathbb{R}$, $m\in\mathbb{N}_0$ that $f(x,u,w)=0$, $b(x,u,w)=u$, $\langle\mu(x),w\rangle_{\mathbb{R}^d}=\sum_{i=1}^d\mu_i x_i w_i$, $\sigma(x)w=(\sigma_1x_1w_1,\sigma_2x_2w_2,\ldots,\sigma_dx_dw_d)$, $\varphi(x)=\exp(-rT)\max\{[\max_{i\in\{1,2,\ldots,d\}}x_i]-100,0\}$, $Z_t(x)=W_t$, $\gamma_m=10^{-1}\mathbb{1}_{[0,4000]}(m)+10^{-2}\mathbb{1}_{(4000,6000]}(m)+10^{-3}\mathbb{1}_{(6000,8000]}(m)+10^{-4}\mathbb{1}_{(8000,10000]}(m)$, $\mathcal{H}_n(x,u,w,z)=u\big(1+z+\frac12z^2-\frac12(t_n-t_{n-1})\big)$ (cf., for instance, Kloeden & Platen [67, Section 10.3]), and

$$\begin{aligned}
&H(t,s,x,w)\\
&=\Big(x_1\exp\!\big((\mu_1-\tfrac{|\sigma_1|^2}{2})(t-s)+\sigma_1w_1\big),\ldots,x_d\exp\!\big((\mu_d-\tfrac{|\sigma_d|^2}{2})(t-s)+\sigma_dw_d\big)\Big).
\end{aligned}\tag{96}$$

d	Result of the approx. algorithm	Runtime in seconds	Reference solution	Relative pathwise error	Relative L^2-error
1	2.801	668.63	2.796	0.0019	
1	0.742	667.01	0.720	0.0303	
1	5.334	667.87	5.272	0.0117	0.0196
1	0.647	667.41	0.644	0.0052	
1	0.299	666.31	0.290	0.0288	
5	1.034	675.16	1.023	0.0108	
5	1.593	673.21	1.587	0.0038	
5	3.381	675.06	3.366	0.0044	0.0101
5	8.005	674.57	7.859	0.0186	
5	5.388	672.14	5.405	0.0032	
10	36.542	679.55	36.150	0.0109	
10	8.705	679.59	8.553	0.0178	
10	27.374	679.59	26.860	0.0191	0.0136
10	3.384	678.01	3.362	0.0066	
10	3.437	678.62	3.407	0.0088	
20	22.041	666.78	22.047	0.0003	
20	24.669	667.58	24.187	0.0199	
20	15.597	666.35	15.328	0.0176	0.0154
20	3.551	664.36	3.493	0.0167	
20	45.559	665.52	44.910	0.0145	
50	68.935	665.67	68.553	0.0056	
50	28.652	666.81	28.250	0.0142	
50	37.778	665.94	37.770	0.0002	0.0140
50	23.248	664.37	22.803	0.0195	
50	14.534	666.92	14.263	0.0190	

Table 2: Numerical simulations for the stochastic heat equations with multiplicative noise in (88).

Note that (78) ensures that for every $t \in [0,T]$, $x \in \mathbb{R}^d$ it holds $\mathbb{P}$-a.s. that

$$X_t(x) = \varphi(x) + \int_0^t \left[\tfrac{1}{2} \sum_{i=1}^{d} |\sigma_i|^2 |x_i|^2 (\tfrac{\partial^2}{\partial x_i^2} X_s)(x) + \sum_{i=1}^{d} \mu_i x_i (\tfrac{\partial}{\partial x_i} X_s)(x) \right] ds + \int_0^t X_s(x)\, dW_s. \tag{97}$$

Next we depict our numerical simulation results for the stochastic Black–Scholes equations with multiplicative noise in (97). In Table 3 we present numerical approximations for the relative L^2-errors $\big(\mathbb{E}\big[|X_T(0)|^{-2} |\mathbb{V}_N^{0,1,\mathbb{S}_m^{0,N}}(\Theta_m^{0,N}, 0) - X_T(0)|^2 \big] \big)^{1/2}$ for $d \in \{1, 5, 10, 20, 50\}$ (cf. (75) and (76)). In our approximative computations for the relative L^2-errors, the exact solutions of the SPDEs in (97) have been approximately computed by means of the well-known result in Lemma 3.7 below. Our proof of Lemma 3.7 employs the well-known result in Lemma 3.5 below.

Lemma 3.5. *Let $d \in \mathbb{N}$, $T, \sigma_1, \sigma_2, \ldots, \sigma_d \in (0,\infty)$, $\mu_1, \mu_2, \ldots, \mu_d \in \mathbb{R}$, let $\varphi \in C(\mathbb{R}^d, \mathbb{R})$ be at most polynomially growing, and let $v \colon [0,T] \times \mathbb{R}^d \to \mathbb{R}$ satisfy for every $t \in (0,T]$, $x = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d$ that $v(0,x) = \varphi(x)$ and*

$$\begin{aligned} v(t,x) = {} & \tfrac{1}{(2\pi t)^{d/2}} \int_{\mathbb{R}} \int_{\mathbb{R}} \cdots \int_{\mathbb{R}} \left[\exp\!\left(- \frac{\sum_{i=1}^{d} |y_i|^2}{2t} \right) \right. \\ & \left. \varphi\!\left(x_1 \exp\!\left(\sigma_1 y_1 + (\mu_1 - \tfrac{|\sigma_1|^2}{2}) t \right), \ldots, x_d \exp\!\left(\sigma_d y_d + (\mu_d - \tfrac{|\sigma_d|^2}{2}) t \right) \right) \right] dy_1\, dy_2 \ldots dy_d. \end{aligned} \tag{98}$$

Then

(i) there exists a unique at most polynomially growing viscosity solution $u \in C([0,T] \times \mathbb{R}^d, \mathbb{R})$ of

$$(\tfrac{\partial}{\partial t} u)(t,x) = \tfrac{1}{2} \sum_{i=1}^{d} |\sigma_i|^2 |x_i|^2 (\tfrac{\partial^2}{\partial x_i^2} u)(t,x) + \sum_{i=1}^{d} \mu_i x_i (\tfrac{\partial}{\partial x_i} u)(t,x) \tag{99}$$

with $u(0,x) = \varphi(x)$ for $(t,x) = (t, x_1, x_2, \ldots, x_d) \in (0,T) \times \mathbb{R}^d$ and

(ii) it holds for every $t \in [0,T]$, $x \in \mathbb{R}^d$ that $u(t,x) = v(t,x)$.

Proof of Lemma 3.5. Throughout this proof let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space and let $W = (W^{(1)}, W^{(2)}, \ldots, W^{(d)}) \colon [0,\infty) \times \Omega \to \mathbb{R}^d$ be a standard Brownian motion. Note that the assumption that $\varphi \in C(\mathbb{R}^d, \mathbb{R})$ is at most polynomially growing assures that there exists a unique at most polynomially growing viscosity solution $u \in C([0,T] \times \mathbb{R}^d, \mathbb{R})$ of

$$(\tfrac{\partial}{\partial t} u)(t,x) = \tfrac{1}{2} \sum_{i=1}^{d} |\sigma_i|^2 |x_i|^2 (\tfrac{\partial^2}{\partial x_i^2} u)(t,x) + \sum_{i=1}^{d} \mu_i x_i (\tfrac{\partial}{\partial x_i} u)(t,x) \tag{100}$$

with $u(0,x) = \varphi(x)$ for $(t,x) = (t, x_1, x_2, \ldots, x_d) \in (0,T) \times \mathbb{R}^d$ (cf., e.g., Beck et al. [5, Corollary 3.9] and Hairer et al. [47, Corollary 4.17]). Moreover, observe that the Feynman-Kac formula ensures that for every $t \in [0,T]$, $x = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d$ it holds that

$$u(t,x) = \mathbb{E}\Big[\varphi\Big(x_1 \exp\Big(\sigma_1 W_t^{(1)} + (\mu_1 - \tfrac{|\sigma_1|^2}{2})t\Big), \ldots, x_d \exp\Big(\sigma_d W_t^{(d)} + (\mu_d - \tfrac{|\sigma_d|^2}{2})t\Big)\Big)\Big] \tag{101}$$

(cf., e.g., Beck et al. [5, Corollary 3.9] and Hairer et al. [47, Corollary 4.17]). Hence, we obtain that for every $t \in [0,T]$, $x = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d$ it holds that

$$\begin{aligned}
&u(t,x) \\
&= \mathbb{E}\Big[\varphi\Big(x_1 \exp\Big(\sigma_1 W_t^{(1)} + (\mu_1 - \tfrac{|\sigma_1|^2}{2})t\Big), \ldots, x_d \exp\Big(\sigma_d W_t^{(d)} + (\mu_d - \tfrac{|\sigma_d|^2}{2})t\Big)\Big)\Big] \\
&= \tfrac{1}{(2\pi t)^{d/2}} \int_{\mathbb{R}} \int_{\mathbb{R}} \cdots \int_{\mathbb{R}} \Bigg[\exp\Bigg(-\frac{\sum_{i=1}^d |y_i|^2}{2t}\Bigg) \\
&\quad \varphi\Big(x_1 \exp\Big(\sigma_1 y_1 + (\mu_1 - \tfrac{|\sigma_1|^2}{2})t\Big), \ldots, x_d \exp\Big(\sigma_d y_d + (\mu_d - \tfrac{|\sigma_d|^2}{2})t\Big)\Big)\Bigg] dy_1\, dy_2 \ldots dy_d \\
&= v(t,x).
\end{aligned} \tag{102}$$

This completes the proof of Lemma 3.5. □

Lemma 3.6. *Let $d \in \mathbb{N}$, $T, \sigma_1, \sigma_2, \ldots, \sigma_d \in (0,\infty)$, $\mu_1, \mu_2, \ldots, \mu_d \in \mathbb{R}$, let $\varphi \in C^2(\mathbb{R}^d, \mathbb{R})$ have at most polynomially growing derivatives, and let $v\colon [0,T] \times \mathbb{R}^d \to \mathbb{R}$ satisfy for every $t \in (0,T]$, $x = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d$ that $v(0,x) = \varphi(x)$ and*

$$\begin{aligned}
v(t,x) &= \tfrac{1}{(2\pi t)^{d/2}} \int_{\mathbb{R}} \int_{\mathbb{R}} \cdots \int_{\mathbb{R}} \Bigg[\exp\Bigg(-\frac{\sum_{i=1}^d |y_i|^2}{2t}\Bigg) \\
&\varphi\Big(x_1 \exp\Big(\sigma_1 y_1 + (\mu_1 - \tfrac{|\sigma_1|^2}{2})t\Big), \ldots, x_d \exp\Big(\sigma_d y_d + (\mu_d - \tfrac{|\sigma_d|^2}{2})t\Big)\Big)\Bigg] dy_1\, dy_2 \ldots dy_d.
\end{aligned} \tag{103}$$

Then

(i) it holds that $v \in C^{1,2}([0,T] \times \mathbb{R}^d, \mathbb{R})$ and

(ii) it holds for every $t \in [0,T]$, $x = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d$ that

$$(\tfrac{\partial}{\partial t} v)(t,x) = \tfrac{1}{2} \sum_{i=1}^d |\sigma_i|^2 |x_i|^2 (\tfrac{\partial^2}{\partial x_i^2} v)(t,x) + \sum_{i=1}^d \mu_i x_i (\tfrac{\partial}{\partial x_i} v)(t,x). \tag{104}$$

Proof of Lemma 3.6. Throughout this proof let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, let $W = (W^{(1)}, W^{(2)}, \ldots, W^{(d)})\colon [0,\infty)\times\Omega \to \mathbb{R}^d$ be a standard Brownian motion, let $S = (S^{(1)}, S^{(2)}, \ldots, S^{(d)})\colon [0,\infty)\times\Omega \to \mathbb{R}^d$ satisfy for every $i \in \{1,2,\ldots,d\}$, $t \in [0,\infty)$ that

$$S_t^{(i)} = \exp\big(\sigma_i W_t^{(i)} + (\mu_i - \tfrac{|\sigma_i|^2}{2})t\big), \tag{105}$$

and let $V\colon [0,T]\times\mathbb{R}^d\times\Omega\to\mathbb{R}$ satisfy for every $t\in[0,T]$, $x=(x_1,x_2,\ldots,x_d)\in\mathbb{R}^d$ that

$$V(t,x) = \varphi\big(x_1 S_t^{(1)}, x_2 S_t^{(2)}, \ldots, x_d S_t^{(d)}\big). \tag{106}$$

Note that (103), (105), and (106) ensure that for every $t\in(0,T]$, $x=(x_1,x_2,\ldots,x_d)\in\mathbb{R}^d$ it holds that

$$\begin{aligned}
&v(t,x)\\
&= \tfrac{1}{(2\pi t)^{d/2}}\int_{\mathbb{R}}\int_{\mathbb{R}}\cdots\int_{\mathbb{R}}\Bigg[\exp\Bigg(-\frac{\sum_{i=1}^d |y_i|^2}{2t}\Bigg)\\
&\quad \varphi\Bigg(x_1\exp\Big(\sigma_1 y_1 + (\mu_1 - \tfrac{|\sigma_1|^2}{2})t\Big),\ldots,x_d\exp\Big(\sigma_d y_d + (\mu_d - \tfrac{|\sigma_d|^2}{2})t\Big)\Bigg)\Bigg]\,dy_1\,dy_2\,\ldots dy_d\\
&= \mathbb{E}\Bigg[\varphi\Bigg(x_1\exp\Big(\sigma_1 W_t^{(1)} + (\mu_1 - \tfrac{|\sigma_1|^2}{2})t\Big),\ldots,x_d\exp\Big(\sigma_d W_t^{(d)} + (\mu_d - \tfrac{|\sigma_d|^2}{2})t\Big)\Bigg)\Bigg]\\
&= \mathbb{E}\Big[\varphi\big(x_1 S_t^{(1)}, x_2 S_t^{(2)},\ldots,x_d S_t^{(d)}\big)\Big]\\
&= \mathbb{E}\big[V(t,x)\big].
\end{aligned} \tag{107}$$

Moreover, note that (105) and Itô's formula assure that for every $i\in\{1,2,\ldots,d\}$, $t\in[0,\infty)$, $x\in\mathbb{R}$ it holds $\mathbb{P}$-a.s. that

$$\begin{aligned}
x S_t^{(i)} &= x\exp\Big(\sigma_i W_t^{(i)} + (\mu_i - \tfrac{|\sigma_i|^2}{2})t\Big)\\
&= x + \int_0^t x\exp\Big(\sigma_i W_s^{(i)} + (\mu_i - \tfrac{|\sigma_i|^2}{2})s\Big)\sigma_i\,dW_s^{(i)}\\
&\quad + \int_0^t x\exp\Big(\sigma_i W_s^{(i)} + (\mu_i - \tfrac{|\sigma_i|^2}{2})s\Big)(\mu_i - \tfrac{|\sigma_i|^2}{2})\,ds\\
&\quad + \tfrac{1}{2}\int_0^t x\exp\Big(\sigma_i W_s^{(i)} + (\mu_i - \tfrac{|\sigma_i|^2}{2})s\Big)|\sigma_i|^2\,ds\\
&= x + \int_0^t \sigma_i x S_s^{(i)}\,dW_s^{(i)} + \int_0^t \mu_i x S_s^{(i)}\,ds.
\end{aligned} \tag{108}$$

Combining this, (106), and Itô's formula ensures that for every $t\in[0,\infty)$, $x=(x_1,x_2,\ldots,x_d)\in$

$\mathbb{R}^d$ it holds $\mathbb{P}$-a.s. that

$$\begin{aligned} V(t,x) &= \varphi\big(x_1 S_t^{(1)}, x_2 S_t^{(2)}, \ldots, x_d S_t^{(d)}\big) \\ &= \varphi(x) + \sum_{i=1}^{d} \int_0^t \big(\tfrac{\partial}{\partial x_i}\varphi\big)\big(x_1 S_s^{(1)}, x_2 S_s^{(2)}, \ldots, x_d S_s^{(d)}\big) \sigma_i x_i S_s^{(i)} \, dW_s^{(i)} \\ &\quad + \sum_{i=1}^{d} \int_0^t \big(\tfrac{\partial}{\partial x_i}\varphi\big)\big(x_1 S_s^{(1)}, x_2 S_s^{(2)}, \ldots, x_d S_s^{(d)}\big) \mu_i x_i S_s^{(i)} \, ds \\ &\quad + \tfrac{1}{2} \sum_{i=1}^{d} \int_0^t \big(\tfrac{\partial^2}{\partial x_i^2}\varphi\big)\big(x_1 S_s^{(1)}, x_2 S_s^{(2)}, \ldots, x_d S_s^{(d)}\big) |\sigma_i|^2 |x_i|^2 \big|S_s^{(i)}\big|^2 \, ds. \end{aligned} \tag{109}$$

Moreover, note that (105) and the fact that $\varphi \in C^2(\mathbb{R}^d, \mathbb{R})$ has at most polynomially growing derivatives assure that for every $p \in (0, \infty)$, $i \in \{1, 2, \ldots, d\}$ it holds that

$$\sup_{t \in [0,p]} \sup_{x_1, x_2, \ldots, x_d \in [-p,p]} \mathbb{E}\Big[\big| \big(\tfrac{\partial}{\partial x_i}\varphi\big)\big(x_1 S_t^{(1)}, x_2 S_t^{(2)}, \ldots, x_d S_t^{(d)}\big) \mu_i x_i S_t^{(i)} \big|^p \Big] < \infty. \tag{110}$$

In addition, observe that (105) and the fact that $\varphi \in C^2(\mathbb{R}^d, \mathbb{R})$ has at most polynomially growing derivatives ensure that for every $p \in (0, \infty)$, $i \in \{1, 2, \ldots, d\}$ it holds that

$$\sup_{t \in [0,p]} \sup_{x_1, x_2, \ldots, x_d \in [-p,p]} \mathbb{E}\Big[\big| \big(\tfrac{\partial^2}{\partial x_i^2}\varphi\big)\big(x_1 S_t^{(1)}, x_2 S_t^{(2)}, \ldots, x_d S_t^{(d)}\big) |\sigma_i|^2 |x_i|^2 \big|S_t^{(i)}\big|^2 \big|^p \Big] < \infty. \tag{111}$$

Furthermore, note that (105) and the fact that $\varphi \in C^2(\mathbb{R}^d, \mathbb{R})$ has at most polynomially growing derivatives assure that for every $i \in \{1, 2, \ldots, d\}$, $t \in [0, \infty)$, $x = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d$ it holds that

$$\mathbb{E}\bigg[\int_0^t \big| \big(\tfrac{\partial}{\partial x_i}\varphi\big)\big(x_1 S_s^{(1)}, x_2 S_s^{(2)}, \ldots, x_d S_s^{(d)}\big) \sigma_i x_i S_s^{(i)} \big|^2 \, ds \bigg] < \infty. \tag{112}$$

This, (110), (111), (109), (107), and Fubini's theorem imply that for every $t \in [0, T]$, $x = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d$ it holds that

$$\begin{aligned} v(t,x) &= \mathbb{E}\big[V(t,x)\big] \\ &= \varphi(x) + \sum_{i=1}^{d} \mathbb{E}\bigg[\int_0^t \big(\tfrac{\partial}{\partial x_i}\varphi\big)\big(x_1 S_s^{(1)}, x_2 S_s^{(2)}, \ldots, x_d S_s^{(d)}\big) \mu_i x_i S_s^{(i)} \, ds \bigg] \\ &\quad + \tfrac{1}{2} \sum_{i=1}^{d} \mathbb{E}\bigg[\int_0^t \big(\tfrac{\partial^2}{\partial x_i^2}\varphi\big)\big(x_1 S_s^{(1)}, x_2 S_s^{(2)}, \ldots, x_d S_s^{(d)}\big) |\sigma_i|^2 |x_i|^2 \big|S_s^{(i)}\big|^2 \, ds \bigg] \\ &= \varphi(x) + \sum_{i=1}^{d} \int_0^t \mathbb{E}\Big[\big(\tfrac{\partial}{\partial x_i}\varphi\big)\big(x_1 S_s^{(1)}, x_2 S_s^{(2)}, \ldots, x_d S_s^{(d)}\big) \mu_i x_i S_s^{(i)} \Big] \, ds \\ &\quad + \tfrac{1}{2} \sum_{i=1}^{d} \int_0^t \mathbb{E}\Big[\big(\tfrac{\partial^2}{\partial x_i^2}\varphi\big)\big(x_1 S_s^{(1)}, x_2 S_s^{(2)}, \ldots, x_d S_s^{(d)}\big) |\sigma_i|^2 |x_i|^2 \big|S_s^{(i)}\big|^2 \Big] \, ds. \end{aligned} \tag{113}$$

Moreover, observe that Lemma 3.5 assures that $v \in C([0,T] \times \mathbb{R}^d, \mathbb{R})$. Combining (110), (111), (113), the de la Vallée Poussin theorem (cf., e.g., Klenke [66, Corollary 6.21]), and the Vitali convergence theorem (cf., e.g., Klenke [66, Theorem 6.25]) with the fundamental theorem of calculus hence ensures that

(a) it holds for every $x \in \mathbb{R}^d$ that $([0,T] \ni t \mapsto v(t,x) \in \mathbb{R}) \in C^1([0,T], \mathbb{R})$ and

(b) it holds that $([0,T] \times \mathbb{R}^d \ni (t,x) \mapsto (\frac{\partial}{\partial t} v)(t,x) \in \mathbb{R}) \in C([0,T] \times \mathbb{R}^d, \mathbb{R})$.

Next note that (106) shows that for every $i \in \{1, 2, \ldots, d\}$, $t \in [0, \infty)$, $x = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d$ it holds that

$$(\tfrac{\partial}{\partial x_i} V)(t, x_1, x_2, \ldots, x_d) = S_t^{(i)} \big(\tfrac{\partial}{\partial x_i} \varphi\big)\big(x_1 S_t^{(1)}, x_2 S_t^{(2)}, \ldots, x_d S_t^{(d)}\big). \tag{114}$$

Moreover, observe that (105) and the fact that $\varphi \in C^2(\mathbb{R}^d, \mathbb{R})$ has at most polynomially growing derivatives assure that for every $p \in (0, \infty)$, $i \in \{1, 2, \ldots, d\}$ it holds that

$$\sup_{t \in [0,p]} \sup_{x_1, x_2, \ldots, x_d \in [-p,p]} \mathbb{E}\Big[\big| S_t^{(i)} \big(\tfrac{\partial}{\partial x_i} \varphi\big)\big(x_1 S_t^{(1)}, x_2 S_t^{(2)}, \ldots, x_d S_t^{(d)}\big) \big|^p \Big] < \infty. \tag{115}$$

Combining this with (114) demonstrates that for every $p \in (0, \infty)$, $i \in \{1, 2, \ldots, d\}$ it holds that

$$\sup_{t \in [0,p]} \sup_{x_1, x_2, \ldots, x_d \in [-p,p]} \mathbb{E}\Big[\big| (\tfrac{\partial}{\partial x_i} V)(t, x_1, x_2, \ldots, x_d) \big|^p \Big] < \infty. \tag{116}$$

This, (107), the de la Vallée Poussin theorem (cf., e.g., Klenke [66, Corollary 6.21]), the Vitali convergence theorem (cf., e.g., Klenke [66, Theorem 6.25]), and the fundamental theorem of calculus imply that

(I) it holds for every $t \in [0,T]$ that $(\mathbb{R}^d \ni x \mapsto v(t,x) \in \mathbb{R}) \in C^1(\mathbb{R}^d, \mathbb{R})$,

(II) it holds for every $i \in \{1, 2, \ldots, d\}$ that

$$\big([0,T] \times \mathbb{R}^d \ni (t,x) \mapsto (\tfrac{\partial}{\partial x_i} v)(t,x) \in \mathbb{R}\big) \in C([0,T] \times \mathbb{R}^d, \mathbb{R}), \tag{117}$$

and

(III) it holds for every $t \in [0,T]$, $x \in \mathbb{R}^d$ that

$$(\tfrac{\partial}{\partial x} v)(t,x) = \mathbb{E}\big[(\tfrac{\partial}{\partial x} V)(t,x)\big]. \tag{118}$$

In addition, observe that (106) ensures that for every $i, j \in \{1, 2, \ldots, d\}$, $t \in [0, \infty)$, $x = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d$ it holds that

$$(\tfrac{\partial^2}{\partial x_i \partial x_j} V)(t, x_1, x_2, \ldots, x_d) = S_t^{(i)} S_t^{(j)} \big(\tfrac{\partial^2}{\partial x_i \partial x_j} \varphi\big)\big(x_1 S_t^{(1)}, x_2 S_t^{(2)}, \ldots, x_d S_t^{(d)}\big). \tag{119}$$

Moreover, note that (105) and the fact that $\varphi \in C^2(\mathbb{R}^d, \mathbb{R})$ has at most polynomially growing derivatives assure that for every $p \in (0, \infty)$, $i, j \in \{1, 2, \ldots, d\}$ it holds that

$$\sup_{t\in[0,p]} \sup_{x_1,x_2,\ldots,x_d\in[-p,p]} \mathbb{E}\Big[\big|S_t^{(i)} S_t^{(j)} \big(\tfrac{\partial^2}{\partial x_i \partial x_j}\varphi\big)\big(x_1 S_t^{(1)}, x_2 S_t^{(2)}, \ldots, x_d S_t^{(d)}\big)\big|^p\Big] < \infty. \tag{120}$$

Combining this with (119) demonstrates that for every $p \in (0, \infty)$, $i \in \{1, 2, \ldots, d\}$ it holds that

$$\sup_{t\in[0,p]} \sup_{x_1,x_2,\ldots,x_d\in[-p,p]} \mathbb{E}\Big[\big|(\tfrac{\partial^2}{\partial x_i \partial x_j} V)(t, x_1, x_2, \ldots, x_d)\big|^p\Big] < \infty. \tag{121}$$

This, item (I), item (II), item (III), the de la Vallée Poussin theorem (cf., e.g., Klenke [66, Corollary 6.21]), the Vitali convergence theorem (cf., e.g., Klenke [66, Theorem 6.25]), and the fundamental theorem of calculus imply that

(A) it holds for every $t \in [0, T]$ that $(\mathbb{R}^d \ni x \mapsto v(t, x) \in \mathbb{R}) \in C^2(\mathbb{R}^d, \mathbb{R})$ and

(B) it holds for every $i, j \in \{1, 2, \ldots, d\}$ that

$$\big([0, T] \times \mathbb{R}^d \ni (t, x) \mapsto (\tfrac{\partial^2}{\partial x_i \partial x_j} v)(t, x) \in \mathbb{R}\big) \in C([0, T] \times \mathbb{R}^d, \mathbb{R}). \tag{122}$$

Moreover, observe that item (a), item (b), and the fact that $v \in C([0, T] \times \mathbb{R}^d, \mathbb{R})$ imply that $v \in C^{1,0}([0, T] \times \mathbb{R}^d, \mathbb{R})$. This, item (A), item (B), and item (II) demonstrate that $v \in C^{1,2}([0, T] \times \mathbb{R}^d, \mathbb{R})$. This establishes item (i). Furthermore, observe that item (i) and Lemma 3.5 establish item (ii). This completes the proof of Lemma 3.6. □

Lemma 3.7. *Let $d \in \mathbb{N}$, $T, \sigma_1, \sigma_2, \ldots, \sigma_d \in (0, \infty)$, $\mu_1, \mu_2, \ldots, \mu_d \in \mathbb{R}$, let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, let $W \colon [0, T] \times \Omega \to \mathbb{R}$ be a standard Brownian motion, let $\varphi \in C^2(\mathbb{R}^d, \mathbb{R})$ have at most polynomially growing derivatives, let $v \colon [0, T] \times \mathbb{R}^d \to \mathbb{R}$ satisfy for every $t \in (0, T]$, $x = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d$ that $v(0, x) = \varphi(x)$ and*

$$\begin{aligned} v(t, x) = \tfrac{1}{(2\pi t)^{d/2}} \int_{\mathbb{R}} \int_{\mathbb{R}} \cdots \int_{\mathbb{R}} \Bigg[\exp\Bigg(-\frac{\sum_{i=1}^d |y_i|^2}{2t} \Bigg) \\ \varphi\Bigg(x_1 \exp\Big(\sigma_1 y_1 + (\mu_1 - \tfrac{|\sigma_1|^2}{2})t\Big), \ldots, x_d \exp\Big(\sigma_d y_d + (\mu_d - \tfrac{|\sigma_d|^2}{2})t\Big)\Bigg)\Bigg] dy_1\, dy_2 \ldots dy_d, \end{aligned} \tag{123}$$

and let $X \colon [0, T] \times \mathbb{R}^d \times \Omega \to \mathbb{R}$ satisfy for every $t \in [0, T]$, $x \in \mathbb{R}^d$ that

$$X_t(x) = \exp\big(W_t - \tfrac{t}{2}\big)\, v(t, x). \tag{124}$$

Then

(i) for every $\omega \in \Omega$ it holds that $([0, T] \times \mathbb{R}^d \ni (t, x) \mapsto X_t(x, \omega) \in \mathbb{R}) \in C^{0,2}([0, T] \times \mathbb{R}^d, \mathbb{R})$ and

(ii) for every $t \in [0,T]$, $x \in \mathbb{R}^d$ it holds $\mathbb{P}$-a.s. that

$$X_t(x) = \varphi(x) + \int_0^t \left[\tfrac{1}{2} \sum_{i=1}^{d} |\sigma_i|^2 |x_i|^2 (\tfrac{\partial^2}{\partial x_i^2} X_s)(x) + \sum_{i=1}^{d} \mu_i x_i (\tfrac{\partial}{\partial x_i} X_s)(x) \right] ds + \int_0^t X_s(x)\, dW_s. \tag{125}$$

Proof of Lemma 3.7. Observe that the hypothesis that $\varphi \in C^2(\mathbb{R}^d, \mathbb{R})$ has at most polynomially growing derivatives and Lemma 3.6 assure that $v \in C^{1,2}([0,T] \times \mathbb{R}^d, \mathbb{R})$. Combining this and (124) proves that for every $\omega \in \Omega$ it holds that $([0,T] \times \mathbb{R}^d \ni (t,x) \mapsto X_t(x,\omega) \in \mathbb{R}) \in C^{0,2}([0,T] \times \mathbb{R}^d, \mathbb{R})$. This establishes item (i). Moreover, observe that the fact that $v \in C^{1,2}([0,T] \times \mathbb{R}^d, \mathbb{R})$, Itô's formula, the assumption that for every $x \in \mathbb{R}^d$ it holds that $v(0,x) = \varphi(x)$, and (124) ensure that for every $t \in [0,T]$, $x \in \mathbb{R}^d$ it holds $\mathbb{P}$-a.s. that

$$\begin{aligned} X_t(x) &= \exp\!\big(W_t - \tfrac{t}{2}\big)\, v(t,x) \\ &= v(0,x) + \int_0^t \exp\!\big(W_s - \tfrac{s}{2}\big)\, (\tfrac{\partial}{\partial s} v)(s,x)\, ds + \int_0^t \exp\!\big(W_s - \tfrac{s}{2}\big)\, v(s,x)\, dW_s \\ &\quad + \int_0^t \big[-\tfrac{1}{2}\big] \exp\!\big(W_s - \tfrac{s}{2}\big)\, v(s,x)\, ds + \tfrac{1}{2} \int_0^t \exp\!\big(W_s - \tfrac{s}{2}\big)\, v(s,x)\, ds \\ &= \varphi(x) + \int_0^t \exp\!\big(W_s - \tfrac{s}{2}\big)\, (\tfrac{\partial}{\partial s} v)(s,x)\, ds + \int_0^t \exp\!\big(W_s - \tfrac{s}{2}\big)\, v(s,x)\, dW_s. \end{aligned} \tag{126}$$

Lemma 3.6, (124), and item (i) hence assure that for every $t \in [0,T]$, $x = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d$ it holds $\mathbb{P}$-a.s. that

$$\begin{aligned} X_t(x) &= \varphi(x) + \int_0^t \exp\!\big(W_s - \tfrac{s}{2}\big) \left[\tfrac{1}{2} \sum_{i=1}^{d} |\sigma_i|^2 |x_i|^2 (\tfrac{\partial^2}{\partial x_i^2} v)(s,x) + \sum_{i=1}^{d} \mu_i x_i (\tfrac{\partial}{\partial x_i} v)(s,x) \right] ds \\ &\quad + \int_0^t X_s(x)\, dW_s \\ &= \varphi(x) + \int_0^t \left[\tfrac{1}{2} \sum_{i=1}^{d} |\sigma_i|^2 |x_i|^2 (\tfrac{\partial^2}{\partial x_i^2} X_s)(x) + \sum_{i=1}^{d} \mu_i x_i (\tfrac{\partial}{\partial x_i} X_s)(x) \right] ds + \int_0^t X_s(x)\, dW_s. \end{aligned} \tag{127}$$

This completes the proof of Lemma 3.7. □

3.4 Zakai equations

In this subsection we apply the approximation algorithm in Framework 2.2 to the Zakai equations in (133) below.

Assume Framework 3.1, let $\alpha = 2\pi$, $\beta = 0.25$, and $\gamma = 0.1$, assume that $T = 0.5$, $N = 25$, $M = 12000$, $d \in \{1, 5, 10, 20, 50\}$, $\delta = d$, and $\varepsilon = 10^{-8}$, let $h = (h_1, h_2, \ldots, h_d) \in C(\mathbb{R}^d, \mathbb{R}^d)$, let $W \colon [0,T] \times \Omega \to \mathbb{R}^d$ be a standard $(\mathcal{F}_t)_{t \in [0,T]}$-Brownian motion, assume for every $s, t \in [0,T]$, $x = (x_1, x_2, \ldots, x_d)$, $w = (w_1, w_2, \ldots, w_d) \in \mathbb{R}^d$, $u \in \mathbb{R}$, $z =$

d	Result of the approx. algorithm	Runtime in seconds	Reference solution	Relative pathwise error	Relative L^2-error
1	139.171	448.275	139.809	0.0046	
1	173.195	450.996	172.210	0.0057	
1	63.231	448.398	63.588	0.0056	0.0044
1	41.285	442.110	41.161	0.0030	
1	116.699	439.910	116.847	0.0013	
5	80.088	455.842	78.773	0.0167	
5	32.612	455.521	31.954	0.0206	
5	77.905	456.283	77.766	0.0018	0.0137
5	24.200	442.570	23.843	0.0150	
5	34.496	443.422	34.610	0.0033	
10	22.367	445.009	22.187	0.0082	
10	69.423	446.532	68.919	0.0073	
10	14.542	452.487	14.596	0.0037	0.0087
10	11.286	455.380	11.285	0.0001	
10	28.276	455.372	27.839	0.0157	
20	4.963	443.438	4.923	0.0081	
20	17.222	442.955	16.951	0.0160	
20	50.882	454.271	50.099	0.0156	0.0138
20	11.090	455.924	10.915	0.0161	
20	18.192	454.760	17.986	0.0115	

Table 3: Numerical simulations for the stochastic Black–Scholes equations with multiplicative noise in (97).

$(z_1, z_2, \ldots, z_d) \in \mathbb{R}^d$, $m \in \mathbb{N}_0$ that $\varphi(x) = \left(\frac{\alpha}{2\pi}\right)^{d/2} \exp(-\frac{\alpha}{2}\|x\|^2_{\mathbb{R}^d})$, $h(x) = \beta x$, $\mu(x) = \gamma x[1+\|x\|^2_{\mathbb{R}^d}]^{-1}$, $\sigma(x)w = d^{-1/2}(\sum_{i=1}^d w_i, \sum_{i=1}^d w_i, \ldots, \sum_{i=1}^d w_i)$, $H(t,s,x,w) = x+\mu(x)(t-s)+\sigma(x)w$, $f(x,u,w) = -\sum_{i=1}^d u\big(\frac{\partial}{\partial x_i}\mu_i\big)(x)$, $b(x,u,w) = uh(x)$, $\gamma_m = 10^{-2}\mathbb{1}_{[0,5000]}(m) + 10^{-3}\mathbb{1}_{(5000,10000]}(m) + 10^{-4}\mathbb{1}_{(10000,12000]}(m)$, and

$$\begin{aligned}\mathcal{H}_n(x,u,w,z) = u - \left[\sum_{i=1}^d u\big(\tfrac{\partial}{\partial x_i}\mu_i\big)(x)\right](t_n - t_{n-1}) + u\langle h(x), z\rangle_{\mathbb{R}^d} \\ + \tfrac{u}{2}\left[\sum_{i,j=1}^d h_i(x)h_j(x)z_iz_j\right] - \tfrac{u(t_n-t_{n-1})}{2}\left[\sum_{i=1}^d |h_i(x)|^2\right],\end{aligned} \tag{128}$$

let $Y\colon [0,T]\times\Omega \to \mathbb{R}^d$ be an $(\mathcal{F}_t)_{t\in[0,T]}$-adapted stochastic process with continuous sample

paths which satisfies that for every $t \in [0,T]$ it holds $\mathbb{P}$-a.s. that

$$Y_t = Y_0 + \int_0^t \mu(Y_s)\,ds + \int_0^t \sigma(Y_s)\,dW_s \tag{129}$$

(signal process/state process/system process), assume for every $A \in \mathcal{B}(\mathbb{R}^d)$ that $\mathbb{P}(Y_0 \in A) = \int_A \varphi(x)\,dx$, let $V\colon [0,T]\times\Omega \to \mathbb{R}^d$ be a standard $(\mathcal{F}_t)_{t\in[0,T]}$-Brownian motion, assume that V and W are independent, and assume that for every $t \in [0,T]$, $x \in \mathbb{R}^d$ it holds $\mathbb{P}$-a.s. that

$$Z_t(x) = \int_0^t h(Y_s)\,ds + V_t \tag{130}$$

(observation process). Note that (78) and the hypothesis that for every $w = (w_1, w_2, \ldots, w_d) \in \mathbb{R}^d$ it holds that $\sigma(x)w = d^{-1/2}(\sum_{i=1}^d w_i, \sum_{i=1}^d w_i, \ldots, \sum_{i=1}^d w_i)$ ensure that for every $t \in [0,T]$, $x = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d$ it holds $\mathbb{P}$-a.s. that

$$\begin{aligned}
X_t(x) &= \varphi(x) + \int_0^t f\big(x, X_s(x), (\nabla X_s)(x)\big)\,ds + \int_0^t \big\langle b\big(x, X_s(x), (\nabla X_s)(x)\big), dZ_s(x)\big\rangle_{\mathbb{R}^\delta} \\
&\quad + \int_0^t \Big[\tfrac{1}{2}\operatorname{Trace}\big(\sigma(x)[\sigma(x)]^*(\operatorname{Hess} X_s)(x)\big) + \big\langle \mu(x), (\nabla X_s)(x)\big\rangle_{\mathbb{R}^d}\Big]\,ds \\
&= \varphi(x) + \int_0^t -\sum_{i=1}^d \big[X_s(x)\big(\tfrac{\partial}{\partial x_i}\mu_i\big)(x)\big]\,ds + \textstyle\int_0^t X_s(x)\,\langle h(x), dZ_s\rangle_{\mathbb{R}^d} \\
&\quad + \int_0^t \Big[\tfrac{1}{2}\operatorname{Trace}\big(\sigma(x)[\sigma(x)]^*(\operatorname{Hess} X_s)(x)\big) - \big\langle \mu(x), (\nabla X_s)(x)\big\rangle_{\mathbb{R}^d}\Big]\,ds \\
&= \varphi(x) + \int_0^t -\sum_{i=1}^d \big[X_s(x)\big(\tfrac{\partial}{\partial x_i}\mu_i\big)(x)\big]\,ds + \textstyle\int_0^t X_s(x)\,\langle h(x), dZ_s\rangle_{\mathbb{R}^d} \\
&\quad + \int_0^t \Big[\tfrac{1}{2}\Big[\sum_{i,j=1}^d \big(\tfrac{\partial^2}{\partial x_i \partial x_j} X_s\big)(x)\Big] - \big\langle \mu(x), (\nabla X_s)(x)\big\rangle_{\mathbb{R}^d}\Big]\,ds.
\end{aligned} \tag{131}$$

The fact that for every $s \in [0,T]$, $x = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d$ it holds that

$$\sum_{i=1}^d \big[X_s(x)\big(\tfrac{\partial}{\partial x_i}\mu_i\big)(x)\big] + \big\langle \mu(x), (\nabla X_s)(x)\big\rangle_{\mathbb{R}^d} = \sum_{i=1}^d \tfrac{\partial}{\partial x_i}\big(\mu_i(x) X_s(x)\big) \tag{132}$$

hence proves that for every $t \in [0,T]$, $x = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d$ it holds $\mathbb{P}$-a.s. that

$$\begin{aligned}
&X_t(x) \\
&= \varphi(x) + \int_0^t \Big[\tfrac{1}{2}\Big[\sum_{i,j=1}^d \big(\tfrac{\partial^2}{\partial x_i \partial x_j} X_s\big)(x)\Big] - \Big[\sum_{i=1}^d \tfrac{\partial}{\partial x_i}\big(\mu_i(x) X_s(x)\big)\Big]\Big]\,ds + \textstyle\int_0^t X_s(x)\,\langle h(x), dZ_s\rangle_{\mathbb{R}^d}.
\end{aligned} \tag{133}$$

In the next step we depict our numerical simulation results for the Zakai equations described in (133) above. In Table 4 we present numerical approximations for the relative L^2-errors $\big(\mathbb{E}\big[|X_T(0)|^{-2}|\mathbb{V}_N^{0,1,\mathbb{S}_m^{0,N}}(\Theta_m^{0,N}, 0) - X_T(0)|^2\big]\big)^{1/2}$ for $d \in \{1, 5, 10, 20, 50\}$ (cf. (75) and (76)).

d	Result of the approx. algorithm	Runtime in seconds	Reference solution	Relative pathwise error	Relative L^2-error
1	0.4699	830.03	0.4812	0.0236	
1	0.4574	827.84	0.4781	0.0433	
1	0.4719	827.02	0.4800	0.0167	0.0274
1	0.4681	828.02	0.4798	0.0243	
1	0.4681	828.28	0.4783	0.0214	
5	0.1984	942.68	0.2063	0.0382	
5	0.2044	942.60	0.2076	0.0155	
5	0.1983	944.02	0.2058	0.0363	0.0266
5	0.2027	942.89	0.2072	0.0216	
5	0.2042	942.89	0.2055	0.0066	
10	0.1233	944.60	0.1271	0.0301	
10	0.1246	943.11	0.1264	0.0142	
10	0.1250	942.84	0.1266	0.0130	0.0165
10	0.1268	941.63	0.1279	0.0088	
10	0.1269	942.62	0.1271	0.0016	
20	0.0691	959.35	0.0695	0.0053	
20	0.0699	961.43	0.0714	0.0215	
20	0.0726	962.18	0.0732	0.0073	0.0117
20	0.0699	959.87	0.0707	0.0119	
20	0.0754	962.79	0.0753	0.0016	
50	0.0283	957.60	0.0283	0.0011	
50	0.0255	958.08	0.0263	0.0279	
50	0.0307	955.14	0.0297	0.0341	0.0209
50	0.0257	958.00	0.0256	0.0032	
50	0.0310	957.00	0.0305	0.0149	

Table 4: Numerical simulations for the Zakai equations in (133).

4 PYTHON source codes for the proposed approximation algorithm

In Subsections 4.1–4.4 below we present the PYTHON source codes associated to the numerical simulations in Subsections 3.1–3.4 above. The following PYTHON source code, PYTHON code 1 below, is employed in the case of each of the PYTHON source codes in Subsections 4.1–4.4 below.

```
import tensorflow as tf
import os
from glob import glob
from tensorflow.python.training.moving_averages \
    import assign_moving_average


def neural_net(y, neurons, name, is_training,
               reuse=None, decay=0.9, dtype=tf.float32):

    def batch_normalization(x):
        beta = tf.compat.v1.get_variable(
            'beta', [x.get_shape()[-1]], dtype,
            tf.zeros_initializer())
        gamma = tf.compat.v1.get_variable(
            'gamma', [x.get_shape()[-1]], dtype,
            tf.ones_initializer())
        mv_mean = tf.compat.v1.get_variable(
            'mv_mean', [x.get_shape()[-1]], dtype=dtype,
            initializer=tf.zeros_initializer(), trainable=False)
        mv_var = tf.compat.v1.get_variable(
            'mv_var', [x.get_shape()[-1]], dtype=dtype,
            initializer=tf.ones_initializer(), trainable=False)
        mean, variance = tf.nn.moments(x, [0], name='moments')
        tf.compat.v1.add_to_collection(
            tf.compat.v1.GraphKeys.UPDATE_OPS,
            assign_moving_average(mv_mean, mean, decay,
                                  zero_debias=True))
        tf.compat.v1.add_to_collection(
            tf.compat.v1.GraphKeys.UPDATE_OPS,
            assign_moving_average(mv_var, variance, decay,
                                  zero_debias=False))

        if is_training:
            return tf.nn.batch_normalization(x, mean, variance,
                                             beta, gamma, 1e-6)
        else:
            return tf.nn.batch_normalization(x, mv_mean, mv_var,
                                             beta, gamma, 1e-6)

```

```
    def layer(x, out_size, activation):
        w = tf.compat.v1.get_variable(
            'weights', [x.get_shape().as_list()[-1], out_size],
            dtype, tf.initializers.glorot_uniform())
        return activation(batch_normalization(tf.matmul(x, w)))

    with tf.compat.v1.variable_scope(name, reuse=reuse):
        y = batch_normalization(y)
        for i in range(len(neurons) - 1):
            with tf.compat.v1.variable_scope('layer_%i_' % (i + 1)):
                y = layer(y, neurons[i], tf.nn.tanh)
        with tf.compat.v1.variable_scope('layer_%i_' % len(neurons)):
            return layer(y, neurons[-1], tf.identity)


def splitting_model(y, z, t, n, phi, h, net,
                    neurons, batch_size, dtype=tf.float32):

    v_n = None

    _y = y[:, :, 1]
    _z = z[:, net]
    if net == 0:
        v_i = phi(_y)
    else:
        v_i = neural_net(_y, neurons, 'v_%i_' % net,
                         False, dtype=dtype)
    grad_v = tf.gradients(v_i, _y)

    _z = tf.reshape(tf.constant(_z, dtype=tf.float32), (1, 1))

    if net == n - 1:
        v_n = tf.compat.v1.get_variable(
            'v_%i_' % (net + 1), [], dtype,
            tf.random_uniform_initializer())
        v_j = tf.ones([batch_size, 1], dtype) * v_n
    else:
        v_j = neural_net(y[:, :, 0], neurons, 'v_%i_' % (net + 1),
                         True, dtype=dtype)

    loss = (v_j - tf.stop_gradient(
        h(_y, v_i, grad_v[0], _z, t / n))) ** 2

    return tf.reduce_mean(loss), v_n


def simulate(t, n, d, sde, phi, h, z, neurons, train_steps,
             batch_size, lr_boundaries, lr_values,
             path, epsilon=1e-8):
```

```
90
91      for i in range(n):
92
93          tf.compat.v1.reset_default_graph()
94
95          y = sde(d, n - i - 1)
96          loss, v_n = splitting_model(y, z, t, n, phi, h, i,
97                                      neurons, batch_size)
98
99          global_step = tf.compat.v1.get_variable(
100             'global_step_%i_' % (i + 1), [], tf.int32,
101             tf.zeros_initializer(), trainable=False)
102
103         learning_rate = tf.compat.v1.train.piecewise_constant(
104             global_step, lr_boundaries, lr_values)
105         update_ops = tf.compat.v1.get_collection(
106             tf.compat.v1.GraphKeys.UPDATE_OPS, 'v_%i_' % (i + 1))
107         with tf.control_dependencies(update_ops):
108             train_op = tf.compat.v1.train.AdamOptimizer(
109                 learning_rate, epsilon=epsilon).minimize(
110                 loss, global_step=global_step)
111
112         with tf.compat.v1.Session() as sess:
113
114             sess.run(tf.compat.v1.global_variables_initializer())
115             var_list_n = tf.compat.v1.get_collection(
116                 tf.compat.v1.GraphKeys.GLOBAL_VARIABLES,
117                 'v_%i_' % (i + 1))
118             saver_n = tf.compat.v1.train.Saver(var_list=var_list_n)
119
120             if i > 0:
121                 saver_p = tf.compat.v1.train.Saver(
122                     var_list=tf.compat.v1.get_collection(
123                         tf.compat.v1.GraphKeys.GLOBAL_VARIABLES,
124                         'v_%i_' % i))
125                 saver_p.restore(sess, os.path.join(
126                     path, 'model_%i_' % i))
127
128             for _ in range(train_steps):
129                 sess.run(train_op)
130
131             saver_n.save(sess, os.path.join(
132                 path, 'model_%i_' % (i + 1)))
133             try:
134                 for filename in glob(os.path.join(
135                         path, 'model_%i_*' % (i - 1))):
136                     os.remove(filename)
137             except OSError:
138                 pass
```

```
139
140             if i == n - 1:
141                 return sess.run(v_n)
```

PYTHON code 1: *common.py*

4.1 A PYTHON source code associated to the numerical simulations in Subsection 3.1

```
1  import tensorflow as tf
2  import numpy as np
3  import os
4  import time
5  import shutil
6  from common import simulate
7
8
9  def phi(x):
10     return tf.reduce_sum(x ** 2, 1, keepdims=True)
11
12
13 def h(x, u, w, z, dt):
14     return u + z
15
16
17 def sde(_d, n):
18     x = [tf.compat.v1.random_normal(
19          [batch_size, _d, 1], stddev=np.sqrt(2. * n * T / N)),
20          tf.compat.v1.random_normal(
21          [batch_size, _d, 1], stddev=np.sqrt(2. * T / N))]
22     return tf.cumsum(tf.concat(x, axis=2), axis=2)
23
24
25 tf.compat.v1.disable_eager_execution()
26 batch_size = 1024
27 train_steps = 8000
28 lr_boundaries = [2000, 4000, 6000]
29 lr_values = [0.1, 0.01, 0.001, 0.0001]
30
31 T = 1.
32 N = 5
33
34 path = '/tmp/heat'
35 _file = open('HeatEquationAdd.csv', 'w')
36 _file.write('d, T, N, run, value, time, ref, pc\n')
37
```

```
for d in [1, 5, 10, 20, 50]:

    neurons = [d + 50, d + 50, 1]

    for run in range(5):

        if os.path.exists(path):
            shutil.rmtree(path)
        os.mkdir(path)

        t_0 = time.time()
        z = np.random.normal(0., np.sqrt(T / N), (1, N))

        xi = np.zeros((d, ))

        v_n = simulate(T, N, d, sde, phi, h, z, neurons, train_steps,
                       batch_size, lr_boundaries, lr_values, path)
        t_1 = time.time()

        b1 = np.cumsum(z, 1)
        u_reference = b1[:, -1] + 2. * T * d + np.sum(xi ** 2, 0)
        _file.write('%i, %f, %i, %i, %f, %f, %f, %f\n'
                    % (d, T, N, run, v_n, t_1 - t_0, u_reference,
                       abs(v_n - u_reference) / u_reference))
        _file.flush()

_file.close()
```

PYTHON code 2: *heat_equation_add.py*

4.2 A PYTHON source code associated to the numerical simulations in Subsection 3.2

```
import tensorflow as tf
import numpy as np
import os
import time
import shutil
from common import simulate


def phi(x):
    return tf.reduce_sum(x ** 2, 1, keepdims=True)


def h(x, u, w, z, dt):
```

```
    return u * (1. + z + 0.5 * z ** 2 - 0.5 * dt)


def sde(_d, n):
    x = [tf.compat.v1.random_normal(
         [batch_size, _d, 1], stddev=np.sqrt(2. * n * T / N)),
         tf.compat.v1.random_normal(
         [batch_size, _d, 1], stddev=np.sqrt(2. * T / N))]
    return tf.cumsum(tf.concat(x, axis=2), axis=2)


tf.compat.v1.disable_eager_execution()
batch_size = 2048
train_steps = 12000
lr_boundaries = [5000, 7000, 10000]
lr_values = [0.1, 0.01, 0.001, 0.0001]

T = 0.5
N = 25

path = '/tmp/heat'
_file = open('HeatEquationMult.csv', 'w')
_file.write('d, T, N, run, value, time, ref, pc\n')

for d in [1, 5, 10, 20, 50]:

    neurons = [d + 50, d + 50, 1]

    for run in range(5):

        if os.path.exists(path):
            shutil.rmtree(path)
        os.mkdir(path)

        t_0 = time.time()
        z = np.random.normal(0., np.sqrt(T / N), (1, N))

        xi = np.zeros((d, 1))

        v_n = simulate(T, N, d, sde, phi, h, z, neurons, train_steps,
                       batch_size, lr_boundaries, lr_values, path)
        t_1 = time.time()

        b1 = np.cumsum(z, 1)
        u_reference = np.exp(b1[:, -1] - T / 2.) * (2. * T * d)

        _file.write('%i, %f, %i, %i, %f, %f, %f, %f\n'
                    % (d, T, N, run, v_n, t_1 - t_0, u_reference,
                       abs(v_n - u_reference) / u_reference))
```

```
        _file.flush()

_file.close()
```

PYTHON code 3: *heat_eqation_mul.py*

4.3 A PYTHON source code associated to the numerical simulations in Subsection 3.3

```
import tensorflow as tf
import numpy as np
import os
import time
import shutil
from common import simulate


def phi(x):
    return np.exp(-1. / 50. * T) * tf.maximum(
        tf.reduce_max(x, 1, keepdims=True) - 100., 0.)


def h(x, u, w, z, dt):
    return u * (1. + z + 0.5 * z ** 2 - 0.5 * dt)


def sde(_d, n):
    x = [tf.compat.v1.random_normal(
         [batch_size, _d, 1], stddev=np.sqrt(n * T / N)),
         tf.compat.v1.random_normal(
         [batch_size, _d, 1], stddev=np.sqrt(T / N))]
    t = tf.reshape(np.array([n * T / N, (n + 1) * T / N],
                            dtype=np.float32), [1, 1, 2])
    return tf.exp((mu - sigma ** 2 / 2.) * t +
                  sigma * tf.cumsum(tf.concat(x, axis=2), axis=2)) \
           * tf.ones([1, _d, 1]) * 100.


def mc(_d):
    y = tf.compat.v1.random_normal([batch_size, _d, 1],
                                   stddev=np.sqrt(T))
    x = tf.exp((mu - sigma ** 2 / 2.) * T + sigma * y) \
        * tf.ones([1, _d, 1]) * 100.
    x = phi(x)
    return tf.reduce_mean(x)

```

```
38
39 tf.compat.v1.disable_eager_execution()
40 batch_size = 1024
41 train_steps = 10000
42 lr_boundaries = [4000, 6000, 8000]
43 lr_values = [0.1, 0.01, 0.001, 0.0001]
44
45 T = 0.5
46 N = 20
47
48 path = '/tmp/bs'
49 _file = open('BlackScholes.csv', 'w')
50 _file.write('d, T, N, run, value, time, ref, pc\n')
51
52 for d in [1, 5, 10, 20]:
53
54     neurons = [d + 50, d + 50, 1]
55
56     for run in range(5):
57
58         if os.path.exists(path):
59             shutil.rmtree(path)
60         os.mkdir(path)
61
62         t_0 = time.time()
63
64         mu = np.reshape((np.sin(np.linspace(d * 1., 1. * d * d, d))
65                          + 1.) / (1. * d), (1, d, 1))
66         sigma = np.reshape(np.linspace(1., 1. * d, d) / (4. * d),
67                            (1, d, 1))
68
69         z = np.random.normal(0., np.sqrt(T / N), (1, N))
70
71         v_n = simulate(T, N, d, sde, phi, h, z, neurons, train_steps,
72                        batch_size, lr_boundaries, lr_values, path)
73         t_1 = time.time()
74
75         b1 = np.cumsum(z, 1)
76         u_reference = np.exp(b1[:, -1] - T / 2.)
77
78         tf.compat.v1.reset_default_graph()
79         ref_sol = mc(d)
80         mc_val = 0.
81         with tf.compat.v1.Session() as sess:
82             for _ in range(1000):
83                 mc_val += sess.run(ref_sol)
84
85         u_reference = u_reference * mc_val / 1000.
86
```

```
        _file.write('%i, %f, %i, %i, %f, %f, %f, %f\n'
                    % (d, T, N, run, v_n, t_1 - t_0, u_reference,
                       abs(v_n - u_reference) / u_reference))
        _file.flush()

_file.close()
```

PYTHON code 4: *black_scholes.py*

4.4 A PYTHON source code associated to the numerical simulations in Subsection 3.4

```
import tensorflow as tf
import numpy as np
import os
import time
import shutil
from common import simulate


def phi(x, _d):
    return ((alpha / 2. / np.pi) ** (_d / 2.)) \
           * tf.exp(-alpha / 2. * tf.reduce_sum(x ** 2, axis=1,
                                                keepdims=True))


def h(x, u, w, z, dt, _d):
    sum_x2 = tf.reduce_sum(x ** 2, axis=1, keepdims=True)
    tmp0 = d / (1. + sum_x2) + 2. * sum_x2 / (1. + sum_x2) ** 2
    hz_sum = tf.reduce_sum(beta * x * z, axis=1, keepdims=True)
    tmp1 = u * hz_sum
    tmp2 = u / 2. * tf.reduce_sum(beta * x * z * hz_sum, axis=1,
                                  keepdims=True)
    tmp3 = u * T / N / 2. * tf.reduce_sum(beta * x * beta * x,
                                          axis=1, keepdims=True)
    return u - u * tmp0 * gamma + tmp1 + tmp2 - tmp3


def sde(_d, n):
    y = [tf.constant(np.zeros((batch_size, _d), dtype=np.float32))]
    for n_ in range(n+1):
        mu = gamma * y[-1] / (1. + tf.reduce_sum(y[-1] ** 2, axis=1,
                                                 keepdims=True))
        sigma = tf.ones((batch_size, _d)) * tf.reduce_sum(
            tf.compat.v1.random_normal((batch_size, _d),
            stddev=np.sqrt(T / N)), axis=1, keepdims=True)
```

```
        y.append(y[-1] + mu * T / N + sigma / np.sqrt(_d))
    return tf.stack(y[n:n + 2], axis=2)


def example():
    w = np.random.normal(0., np.sqrt(T / N), (d, N))
    v = np.random.normal(0., np.sqrt(T / N), (d, N))
    y = [np.zeros((d, ))]
    z = [np.zeros((d, ))]
    for i in range(N):
        z.append(z[-1] + beta * T / N * y[-1] + v[:, i])
        y.append(y[-1] + gamma * T/N * y[-1]
                 / (1. + np.sum(y[-1] ** 2))
                 + np.sum(w[:, i]) / np.sqrt(d))

    return v, y, z


def ref_solution(v, y):
    v = tf.cumsum(tf.expand_dims(tf.cast(v, tf.float32), axis=0),
                  axis=2)
    y = tf.cast(tf.expand_dims(tf.stack(y, axis=1), axis=0),
                tf.float32)
    x_0 = tf.zeros((batch_size, d)) + 0.
    w = tf.compat.v1.random_normal((batch_size, d, N),
                                   stddev=np.sqrt(T / N))
    yy = [x_0]
    BB = 0.
    fact = 0.5
    for i in range(N):
        vv = v[:, :, N - i - 1]
        vv_sum = tf.reduce_sum(v[:, :, N - i - 1], axis=1,
                               keepdims=True)
        yy_norm = tf.reduce_sum(yy[-1] ** 2, axis=1, keepdims=True)
        BB += fact * T/N * tf.reduce_sum(0.5 * vv_sum
            * tf.ones((1, d)) * vv * beta ** 2, axis=1,
                                         keepdims=True)
        BB += fact * T/N * tf.reduce_sum(yy[-1]
            * y[:, :, N - i - 1] * beta ** 2
            - 0.5 * (beta * yy[-1]) ** 2
            - beta * gamma * vv * yy[-1] / (1. + yy_norm)
            - gamma * (1. + yy_norm - 2. * yy[-1] ** 2)
            / ((1. + yy_norm) ** 2), axis=1, keepdims=True)
        yy.append(yy[-1] + T/N * (beta * vv_sum * tf.ones((1, d))
            - gamma * yy[-1] / (1. + yy_norm))
            + tf.reduce_sum(w[:, :, i], axis=1, keepdims=True)
            / np.sqrt(d))
        fact = 1.
    vv = v[:, :, 0] * 0.
```

```
    vv_sum = tf.reduce_sum(v[:, :, 0] * 0., axis=1, keepdims=True)
    yy_norm = tf.reduce_sum(yy[-1] ** 2, axis=1, keepdims=True)
    BB += 0.5 * T / N * tf.reduce_sum(0.5 * vv_sum * tf.ones((1, d))
            * vv * beta ** 2, axis=1, keepdims=True)
    BB += 0.5 * T / N * tf.reduce_sum(yy[-1] * y[:, :, 0] * beta ** 2
            - 0.5 * (beta * yy[-1]) ** 2
            - beta * gamma * vv * yy[-1] / (1. + yy_norm)
            - gamma * (1. + yy_norm - 2. * yy[-1] ** 2)
            / ((1. + yy_norm) ** 2), axis=1, keepdims=True)
    return tf.reduce_mean(((alpha / 2. / np.pi) ** (d / 2))
            * tf.exp(BB - alpha / 2. * yy_norm),
                     axis=0, keepdims=True)


tf.compat.v1.disable_eager_execution()
batch_size = 2048
train_steps = 12000
lr_boundaries = [5000, 10000]
lr_values = [0.01, 0.001, 0.0001]

alpha, beta, gamma = 2. * np.pi, 0.25, 0.1
T = 0.5
N = 25

path = '/tmp/zakai'
_file = open('Zakai.csv', 'w')
_file.write('d, T, N, run, value, time, ref, pc\n')

for d in [1, 5, 10, 20, 50]:

    neurons = [d + 50, d + 50, 1]

    for run in range(5):

        if os.path.exists(path):
            shutil.rmtree(path)
        os.mkdir(path)

        t_0 = time.time()

        tf.compat.v1.reset_default_graph()
        v, y, z = example()

        z = np.stack(z, axis=1)
        z = np.diff(z, axis=1)

        v_mean = ref_solution(v, y)

        sum = 0.
```

```
        with tf.compat.v1.Session() as sess:
            for _ in range(1000):
                sum += sess.run(v_mean)

            sum /= 1000.

        b1 = np.cumsum(v, 1)
        u_reference = sum * np.exp(np.sum(beta * 0. * b1[:, -1]))

        tf.compat.v1.reset_default_graph()

        v_n = simulate(T, N, d, sde, lambda x: phi(x, d),
                       lambda x, u, w, z, dt: h(x, u, w, z, dt, d),
                       z, neurons, train_steps, batch_size,
                       lr_boundaries, lr_values, path)

        t_1 = time.time()

        _file.write('%i, %f, %i, %i, %f, %f, %f, %f\n'
                    % (d, T, N, run, v_n, t_1 - t_0, u_reference,
                       abs(v_n - u_reference) / u_reference))
        _file.flush()

_file.close()
```

PYTHON code 5: *zakai_equation.py*

Acknowledgments

This work has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy EXC 2044-390685587, Mathematics Münster: Dynamics-Geometry-Structure, by the Schweizerischer Nationalfonds (SNF, Swiss National Science Foundation) through the research project 200020_175699 "Higher order numerical approximation methods for stochastic partial differential equations", and by the Nanyang Assistant Professorship Grant (NAP Grant) "Machine Learning based Algorithms in Finance and Insurance".

References

[1] BAYER, C., AND OBERHAUSER, H. Splitting methods for SPDEs: From robustness to financial engineering, optimal control, and nonlinear filtering. In *Splitting Methods in Communication, Imaging, Science, and Engineering*. Springer, 2016, pp. 499–539.

[2] Beck, C., Becker, S., Cheridito, P., Jentzen, A., and Neufeld, A. Deep splitting method for parabolic PDEs. *arXiv:1907.03452* (2019), 40 pages.

[3] Beck, C., Becker, S., Grohs, P., Jaafari, N., and Jentzen, A. Solving stochastic differential equations and Kolmogorov equations by means of deep learning. *arXiv:1806.00421* (2018), 56 pages.

[4] Beck, C., E, W., and Jentzen, A. Machine learning approximation algorithms for high-dimensional fully nonlinear partial differential equations and second-order backward stochastic differential equations. *Journal of Nonlinear Science 29*, 4 (2019), 1563–1619.

[5] Beck, C., Hutzenthaler, M., and Jentzen, A. On nonlinear Feynman–Kac formulas for viscosity solutions of semilinear parabolic partial differential equations. *arXiv:2004.03389* (2020), 54 pages.

[6] Becker, S., Cheridito, P., Jentzen, A., and Welti, T. Solving high-dimensional optimal stopping problems using deep learning. *arXiv:1908.01602* (2019), 42 pages.

[7] Bengio, Y. Learning deep architectures for AI. *Foundations and Trends in Machine Learning 2*, 1 (2009), 1–127.

[8] Bensoussan, A. Splitting up method in the context of stochastic PDE. In *Stochastic partial differential equations and their applications (Charlotte, NC, 1991)*, vol. 176 of *Lect. Notes Control Inf. Sci.* Springer, Berlin, 1992, pp. 22–31.

[9] Bensoussan, A., Glowinski, R., and Rascanu, A. Approximation of the Zakai equation by the splitting up method. *SIAM Journal on Control and Optimization 28*, 6 (1990), 1420–1431.

[10] Bensoussan, A., Glowinski, R., and Răscanu, A. Approximation of some stochastic differential equations by the splitting up method. *Appl. Math. Optim. 25*, 1 (1992), 81–106.

[11] Berg, J., and Nyström, K. A unified deep artificial neural network approach to partial differential equations in complex geometries. *Neurocomputing 317* (2018), 28–41.

[12] Bessaih, H., Brzeźniak, Z., and Millet, A. Splitting up method for the 2d stochastic Navier–Stokes equations. *Stochastic Partial Differential Equations: Analysis and Computations 2*, 4 (2014), 433–470.

[13] Birnir, B. The Kolmogorov–Obukhov statistical theory of turbulence. *Journal of Nonlinear Science 23*, 4 (2013), 657–688.

[14] Birnir, B. *The Kolmogorov-Obukhov Theory of Turbulence: A mathematical theory of turbulence*. Springer Briefs in Mathematics. Springer, New York, 2013.

[15] Blömker, D., and Romito, M. Term structure models driven by Wiener processes and poisson measures: existence and positivity. *Jahresber. Dtsch. Math.-Ver. 117 4* (2014), 233–286.

[16] Brigo, D., and Hanzon, B. On some filtering problems arising in mathematical finance. *Insurance: Mathematics and Economics 22*, 3 (1998), 53–64.

[17] Brzeźniak, Z., Carelli, E., and Prohl, A. Finite-element-based discretizations of the incompressible Navier–Stokes equations with multiplicative random forcing. *IMA Journal of Numerical Analysis 33*, 3 (2013), 771–824.

[18] Budman, H., Holcomb, T., and Morari, M. Pls-based robust inferential control for a packed-bed reactor. *1991 American Control Conference* (1991), 256–261.

[19] Buehner, M., McTaggart-Cowan, R., and Heilliette, S. An ensemble Kalman filter for numerical weather prediction based on variational data assimilation: Varenkf. *Monthly Weather Review 145*, 2 (2017), 617–635.

[20] Cassola, F., and Burlando, M. Wind speed and wind energy forecast through Kalman filtering of numerical weather prediction model output. *Applied energy 99* (2012), 154–166.

[21] Ceci, C., and Colaneri, K. Recent advances in nonlinear filtering with a financial application to derivatives hedging under incomplete information. *Bayesian Inference, Javier Prieto Tejedor (Ed.), InTech, DOI: 10.5772/intechopen.70060.* (2017).

[22] Chan-Wai-Nam, Q., Mikael, J., and Warin, X. Machine learning for semi linear PDEs. *Journal of Scientific Computing 79*, 3 (2019), 1667–1712.

[23] Che, Y., Peng, X., Delle Monache, L., Kawaguchi, T., and Xiao, F. A wind power forecasting system based on the weather research and forecasting model and Kalman filtering over a wind-farm in japan. *Journal of Renewable and Sustainable Energy 8*, 1 (2016), 013302.

[24] Chen, C.-Y., and Sun, C.-C. Adaptive inferential control of packed-bed reactors. *Chemical Engineering Science 46*, 4 (1991), 1041–1054.

[25] Coculescu, D., Geman, H., and Jeanblanc, M. Valuation of default sensitive claims under imperfect information. *Finance and Stochastics 12*, 2 (2008), 195–218.

[26] Debussche, A. Weak approximation of stochastic partial differential equations: the nonlinear case. *Mathematics of Computation 80*, 273 (2011), 89–117.

[27] Deck, T., and Kruse, S. Parabolic differential equations with unbounded coefficients—a generalization of the parametrix method. *Acta Appl. Math. 74*, 1 (2002), 71–91.

[28] Duc, L., Kuroda, T., Saito, K., and Fujita, T. Ensemble Kalman filter data assimilation and storm surge experiments of tropical cyclone nargis. *Tellus A: Dynamic Meteorology and Oceanography 67*, 1 (2015), 25941.

[29] Duffie, D., and Lando, D. Term structures of credit spreads with incomplete accounting information. *Econometrica 69* (2001), 633–664.

[30] E, W., Han, J., and Jentzen, A. Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. *Communications in Mathematics and Statistics 5* (2017), 349–380.

[31] E, W., Han, J., and Jentzen, A. Algorithms for solving high dimensional PDEs: From nonlinear Monte Carlo to machine learning. *arXiv:2008.13333* (2020), 40 pages.

[32] E, W., and Yu, B. The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems. *Communications in Mathematics and Statistics 6*, 1 (2018), 1–12.

[33] Falissard, F. Genuinely multi-dimensional explicit and implicit generalized Shapiro filters for weather forecasting, computational fluid dynamics and aeroacoustics. *Journal of Computational Physics 253* (2013), 344–367.

[34] Filipović, D., Tappe, S., and Teichmann, J. Term structure models driven by Wiener processes and poisson measures: existence and positivity. *SIAM Journal on Financial Mathematics 1*, 1 (2010), 523–554.

[35] Florchinger, P., and Gland, F. l. Time-discretization of the Zakai equation for diffusion processes observed in correlated noise. *Stochastics: An International Journal of Probability and Stochastic Processes 35*, 4 (1991), 233–256.

[36] Frey, R., and Runggaldier, W. Pricing credit derivatives under incomplete information: a nonlinear-filtering approach. *Finance and Stochastics 14*, 4 (2010), 495–526.

[37] Frey, R., and Schmidt, T. Pricing and hedging of credit derivatives via the innovations approach to nonlinear filtering. *Finance and Stochastics 16*, 1 (2012), 105–133.

[38] Geissert, M., Kovács, M., and Larsson, S. Rate of weak convergence of the finite element method for the stochastic heat equation with additive noise. *BIT Numerical Mathematics 49*, 2 (2009), 343–356.

[39] Grecksch, W., and Kloeden, P. E. Time-discretised Galerkin approximations of parabolic stochastic PDEs. *Bull. Austral. Math. Soc. 54*, 1 (1996), 79–85.

[40] Grecksch, W., and Lisei, H. Approximation of stochastic nonlinear equations of Schrödinger type by the splitting method. *Stoch. Anal. Appl. 31*, 2 (2013), 314–335.

[41] Gyöngy, I. Lattice approximations for stochastic quasi-linear parabolic partial differential equations driven by space-time white noise II. *Potential Analysis 11*, 1 (1999), 1–37.

[42] Gyöngy, I. Approximations of stochastic partial differential equations. lecture notes in pure and appl. math. *Stochastic partial differential equations and applications,Dekker, New York 227* (2002), 287–307.

[43] Gyöngy, I., and Krylov, N. On the rate of convergence of splitting-up approximations for SPDEs. In *Stochastic inequalities and applications*, vol. 56 of *Progr. Probab.* Birkhäuser, Basel, 2003, pp. 301–321.

[44] Gyöngy, I., and Krylov, N. On the splitting-up method and stochastic partial differential equations. *Ann. Probab. 31*, 2 (2003), 564–591.

[45] Gyöngy, I., and Martínez, T. On numerical solution of stochastic partial differential equations of elliptic type. *Stochastics: An International Journal of Probability and Stochastics Processes 78*, 4 (2006), 213–231.

[46] Hairer, M. Solving the KPZ equation. *Ann. of Math. 178*, 2 (2013), 559–664.

[47] Hairer, M., Hutzenthaler, M., and Jentzen, A. Loss of regularity for Kolmogorov equations. *Ann. Probab. 43*, 2 (2015), 468–527.

[48] Han, J., Jentzen, A., and E, W. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences 115*, 34 (2018), 8505–8510.

[49] Harms, P., Stefanovits, D., Teichmann, J., and Wüthrich, M. V. Consistent recalibration of yield curve models. *Mathematical Finance 28*, 3 (2018), 757–799.

[50] Hausenblas, E. Numerical analysis of semilinear stochastic evolution equations in Banach spaces. *Journal of Computational and Applied Mathematics 147*, 2 (2002), 485–516.

[51] Hausenblas, E. Approximation for semilinear stochastic evolution equations. *Potential Analysis 18*, 2 (2003), 141–186.

[52] Hausenblas, E. Finite element approximation of stochastic partial differential equations driven by Poisson random measures of jump type. *SIAM Journal on Numerical Analysis 46*, 1 (2008), 437–471.

[53] Henry-Labordere, P. Deep primal-dual algorithm for BSDEs: Applications of machine learning to CVA and IM. *SSRN 3071506* (2017), 36 pages.

[54] Huré, C., Pham, H., and Warin, X. Some machine learning schemes for high-dimensional nonlinear PDEs. *arXiv:1902.01599* (2019), 33 pages.

[55] Hutzenthaler, M., Jentzen, A., and Salimova, D. Strong convergence of full-discrete nonlinearity-truncated accelerated exponential Euler-type approximations for stochastic Kuramoto-Sivashinsky equations. *Communications in Mathematical Sciences 16*, 6 (2018), 1489–1529.

[56] 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), 2015.

[57] Jacquier, A. J., and Oumgari, M. Deep PPDEs for rough local stochastic volatility. *arXiv:1906.02551* (2019), 21 pages.

[58] Jentzen, A., Kloeden, P., and Winkel, G. Efficient simulation of nonlinear parabolic SPDEs with additive noise. *The Annals of Applied Probability 21*, 3 (2011), 908–950.

[59] Jentzen, A., and Kloeden, P. E. The numerical approximation of stochastic partial differential equations. *Milan Journal of Mathematics 77*, 1 (2009), 205–244.

[60] Jentzen, A., and Kloeden, P. E. Overcoming the order barrier in the numerical approximation of stochastic partial differential equations with additive space–time noise. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* (2009), vol. 465, The Royal Society, pp. 649–667.

[61] Jentzen, A., and Kloeden, P. E. *Taylor approximations for stochastic partial differential equations*, vol. 83. SIAM, 2011.

[62] Jentzen, A., and Röckner, M. A Milstein scheme for SPDEs. *Foundations of Computational Mathematics 15*, 2 (2015), 313–362.

[63] Kallianpur, G., and Xiong, J. Stochastic models of environmental pollution. *Advances in Applied Probability 26*, 2 (1994), 377–403.

[64] Katsoulakis, M. A., Kossioris, G. T., and Lakkis, O. Noise regularization and computations for the 1-dimensional stochastic Allen-Cahn problem. *arXiv:1111.6312* (2011), 28 pages.

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

[66] Klenke, A. *Probability theory*, second ed. Universitext. Springer, London, 2014. A comprehensive course.

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

[68] Kloeden, P. E., and Shott, S. Linear-implicit strong schemes for Itô-Galkerin approximations of stochastic pdes. *International Journal of Stochastic Analysis 14*, 1 (2001), 47–53.

[69] Kouritzin, M. A., and Long, H. Convergence of markov chain approximations to stochastic reaction-diffusion equations. *The Annals of Applied Probability 12*, 3 (2002), 1039–1070.

[70] Kovács, M., Larsson, S., and Saedpanah, F. Finite element approximation of the linear stochastic wave equation with additive noise. *SIAM Journal on Numerical Analysis 48*, 2 (2010), 408–427.

[71] Kruse, R. Consistency and stability of a Milstein–Galerkin finite element scheme for semilinear SPDE. *Stochastic Partial Differential Equations: Analysis and Computations 2*, 4 (2014), 471–516.

[72] Kruse, R. *Strong and weak approximation of semilinear stochastic evolution equations.* Springer, 2014.

[73] Krylov, N. V. *Lectures on elliptic and parabolic equations in Hölder spaces*, vol. 12 of *Graduate Studies in Mathematics.* American Mathematical Society, Providence, RI, 1996.

[74] Krylov, N. V. On Kolmogorov's equations for finite-dimensional diffusions. In *Stochastic PDE's and Kolmogorov equations in infinite dimensions (Cetraro, 1998)*, vol. 1715 of *Lecture Notes in Math.* Springer, Berlin, 1999, pp. 1–63.

[75] Kushner, H. J. On the differential equations satisfied by conditional probablitity densities of markov processes, with applications. *Journal of the Society for Industrial and Applied Mathematics, Series A: Control 2*, 1 (1964), 106–119.

[76] Le Gland, F. Splitting-up approximation for SPDEs and SDEs with application to nonlinear filtering. In *Stochastic partial differential equations and their applications (Charlotte, NC, 1991)*, vol. 176 of *Lect. Notes Control Inf. Sci.* Springer, Berlin, 1992, pp. 177–187.

[77] LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *Nature 521* (2015), 436–444.

[78] Liu, W., and Röckner, M. *Stochastic partial differential equations: an introduction*. Springer, 2015.

[79] Lord, G. J., and Shardlow, T. Postprocessing for stochastic parabolic partial differential equations. *SIAM Journal on Numerical Analysis 45*, 2 (2007), 870–889.

[80] Lord, G. J., and Tambue, A. A modified semi–implict Euler-Maruyama scheme for finite element discretization of SPDEs with additive noise. *arXiv:1004.1998* (2010), 28 pages.

[81] Lord, G. J., and Tambue, A. Stochastic exponential integrators for a finite element discretisation of SPDEs with additive noise. *Applied Numerical Mathematics 136* (2019), 163–182.

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

[83] Millet, A., and Morien, P.-L. On implicit and explicit discretization schemes for parabolic SPDEs in any dimension. *Stochastic Processes and their Applications 115*, 7 (2005), 1073–1106.

[84] Milstein, G. N., and Tretyakov, M. V. Solving parabolic stochastic partial differential equations via averaging over characteristics. *Math. Comp. 78*, 268 (2009), 2075–2106.

[85] Mourrat, J.-C., and Weber, H. Convergence ot the two-dimensional dynamic Ising-Kac model to Φ^4_2. *Communications on Pure and Applied Mathematics 70*, 4 (2017), 717–812.

[86] Mueller-Gronbach, T., Ritter, K., and Wagner, T. Optimal pointwise approximation of infinite-dimensional Ornstein–Uhlenbeck processes. *Stochastics and Dynamics 8*, 03 (2008), 519–541.

[87] Mukam, J. D., and Tambue, A. A note on exponential Rosenbrock-Euler method for the finite element discretization of a semilinear parabolic partial differential equation. *Computers & Mathematics with Applications 76*, 7 (2018), 1719–1738.

[88] Müller-Gronbach, T., and Ritter, K. An implicit Euler scheme with non-uniform time discretization for heat equations with multiplicative noise. *BIT Numerical Mathematics 47*, 2 (2007), 393–418.

[89] Pelosi, A., Medina, H., Van den Bergh, J., Vannitsem, S., and Chirico, G. B. Adaptive Kalman filtering for postprocessing ensemble numerical weather predictions. *Monthly Weather Review 145*, 12 (2017), 4837–4854.

[90] Pettersson, R., and Signahl, M. Numerical approximation for a white noise driven spde with locally bounded drift. *Potential Analysis 22*, 4 (2005), 375–393.

[91] Raissi, M. Deep hidden physics models: Deep learning of nonlinear partial differential equations. *The Journal of Machine Learning Research 19*, 1 (2018), 932–955.

[92] Reisinger, C., and Wang, Z. Stability and error analysis of an implicit Milstein finite difference scheme for a two-dimensional Zakai SPDE. *BIT Numerical Mathematics 59*, 4 (2019), 987–1029.

[93] Roth, C. A combination of finite difference and Wong-Zakai methods for hyperbolic stochastic partial differential equations. *Stochastic analysis and applications 24*, 1 (2006), 221–240.

[94] Rutzler, W. Nonlinear and adaptive parameter estimation methods for tubular reactors. *Ind. Eng. Chem. Res. 26*, 2 (1987), 325–333.

[95] Seinfeld, J. H., Gavalas, G. R., and Hwang, M. Nonlinear filtering in distributed parameter systems. *J. Dyn. Sys., Meas., Control 93*, 3 (1971), 157–163.

[96] Shardlow, T. Numerical methods for stochastic parabolic PDEs. *Numer. Funct. Anal. Optim. 20*, 1-2 (1999), 121–145.

[97] Sirignano, J., and Spiliopoulos, K. DGM: a deep learning algorithm for solving partial differential equations. *Journal of computational physics 375* (2018), 1339–1364.

[98] Soliman, M. A., and Ray, W. H. Non-linear filtering for distributed parameter systems having a small parameter. *International Journal of Control 30*, 5 (1979), 757–772.

[99] Walsh, J. B. Finite element methods for parabolic stochastic PDE's. *Potential Anal. 23*, 1 (2005), 1–43.

[100] Walsh, J. B. On numerical solutions of the stochastic wave equation. *Illinois J. Math. 50*, 1-4 (2006), 991–1018.

[101] Wang, X., Gan, S., and Tang, J. Higher order strong approximations of semilinear stochastic wave equation with additive space-time white noise. *SIAM Journal on Scientific Computing 36*, 6 (2014), A2611–A2632.

[102] Wang, X., and Qi, R. A note on an accelerated exponential Euler method for parabolic SPDEs with additive noise. *Applied Mathematics Letters 46* (2015), 31–37.

[103] Windes, L. C., Cinar, A., and Ray, W. H. Dynamic estimation of temperature and concentration profiles in a packed bed reactor. *Chemical Engineering Science 44*, 10 (1989), 2087–2106.

[104] Yan, Y. Galerkin finite element methods for stochastic parabolic partial differential equations. *SIAM journal on numerical analysis 43*, 4 (2005), 1363–1384.

[105] Zakai, M. On the optimal filtering of diffusion processes. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete 11*, 3 (1969), 230–243.

[106] Zhang, D., Guo, L., and Karniadakis, G. E. Learning in modal space: Solving time-dependent stochastic PDEs using physics-informed neural networks. *SIAM Journal on Scientific Computing 42*, 2 (2020), A639–A665.