Learning the random variables in Monte Carlo simulations with stochastic gradient descent: Machine learning for parametric PDEs and financial derivative pricing

Journal Article

Author(s):
Becker, Sebastian; Jentzen, Arnulf; Müller, Marvin S.; von Wurstemberger, Philippe

Publication date:
2024-01

Permanent link:
https://doi.org/10.3929/ethz-b-000627368

Rights / license:
Creative Commons Attribution 4.0 International

Originally published in:
Mathematical Finance 34(1), https://doi.org/10.1111/mafi.12405

Received: 22 February 2022 | Accepted: 9 May 2023

DOI: 10.1111/mafi.12405

SPECIAL ISSUE ARTICLE

MATHEMATICAL FINANCE An International Journal of Mathematics, Statistics and Financial Economics

WILEY

Learning the random variables in Monte Carlo simulations with stochastic gradient descent: Machine learning for parametric PDEs and financial derivative pricing

Sebastian Becker[1] | **Arnulf Jentzen**[2,3] | **Marvin S. Müller**[4] | **Philippe von Wurstemberger**[1,5]

[1]Risklab, Department of Mathematics, ETH Zurich, Zurich, Switzerland

[2]School of Data Science and Shenzhen Research Institute of Big Data, The Chinese University of Hong Kong, Shenzhen, Shenzhen, China

[3]Applied Mathematics: Institute for Analysis and Numerics, University of Münster, Münster, Germany

[4]2Xideas Switzerland AG, Küsnacht, Switzerland

[5]School of Data Science, The Chinese University of Hong Kong, Shenzhen, Shenzhen, China

Correspondence
Philippe von Wurstemberger, Risklab, Department of Mathematics, ETH Zurich, Zurich, Switzerland.
Email: philippe.vonwurstemberger@math.ethz.ch

Funding information
Deutsche Forschungsgemeinschaft, Grant/Award Number: EXC2044-390685587; National Natural Science Foundation of China, Grant/Award Number: 12250610192

Abstract

In financial engineering, prices of financial products are computed approximately many times each trading day with (slightly) different parameters in each calculation. In many financial models such prices can be approximated by means of Monte Carlo (MC) simulations. To obtain a good approximation the MC sample size usually needs to be considerably large resulting in a long computing time to obtain a single approximation. A natural deep learning approach to reduce the computation time when new prices need to be calculated as quickly as possible would be to train an artificial neural network (ANN) to learn the function which maps parameters of the model and of the financial product to the price of the financial product. However, empirically it turns out that this approach leads to approximations with unacceptably high errors, in particular when the error is measured in the L^∞-norm, and it seems that ANNs are not capable to closely approximate prices of financial products in dependence on the model and product parameters in real life applications. This is not entirely surprising given the high-dimensional nature of the problem and the fact that it has recently been proved for a large class of algorithms, including the deep learning approach outlined above, that such methods are in general not capable to overcome the curse of dimensionality for such approximation problems in the L^∞-norm. In this article we introduce a new numerical approximation strategy for parametric approximation problems including the parametric financial pricing problems described above and we illustrate by means of several numerical experiments that the introduced approximation strategy achieves a very high accuracy for a variety of high-dimensional parametric approximation problems, even in the L^∞-norm. A central aspect of the approximation strategy proposed in this article is to combine MC algorithms with machine learning techniques to, roughly speaking, *learn the random variables* (LRV) in MC simulations. In other words, we employ stochastic gradient descent (SGD) optimization methods not to train parameters of standard ANNs but instead to learn random variables appearing in MC approximations. In that sense, the proposed LRV strategy has strong links to *Quasi-Monte Carlo* (QMC) methods as well as to the field of algorithm learning. Our numerical simulations strongly indicate that the LRV strategy might indeed be capable to overcome the curse of dimensionality in the L^∞-norm in several cases where the standard deep learning approach has been proven not to be able to do so. This is not a contradiction to the established lower bounds mentioned above because this new LRV strategy is outside of the class of algorithms for which lower bounds have been established in the scientific literature. The proposed LRV strategy is of general nature and not only restricted to the parametric financial pricing problems described above, but applicable to a large class of approximation problems. In this article we numerically test the LRV strategy in the case of the pricing of European call options in the Black-Scholes model with one underlying asset, in the case of the pricing of European worst-of basket put options in the Black-Scholes model with three underlying assets, in the case of the pricing of European average put options in the Black-Scholes model with three underlying assets and knock-in barriers, as well as in the case of stochastic Lorentz equations. For these examples the LRV strategy produces highly convincing numerical results when compared with standard MC simulations, QMC simulations using Sobol sequences, SGD-trained shallow ANNs, and SGD-trained deep ANNs.

1 INTRODUCTION

Many computational problems from engineering and science can be cast as certain parametric approximation problems (cf., e.g., Berner et al., 2020; Bhattacharya et al., 2021; Chkifa et al., 2015; Cohen & DeVore, 2015; Flandoli et al., 2021; Heinrich & Sindambiwe, 1999; Khoo et al., 2021; Kutyniok et al., 2021; O'Leary-Roseberry et al., 2022; Vidales et al., 2018 and references mentioned therein). In particular, parametric PDEs are of fundamental importance in various applications, where one is not only interested in an approximation of the solution of the approximation problem at one fixed (space-time) point but where one is interested to evaluate the approximative solution again and again as, for instance, in financial engineering where prices of financial products are computed approximately many times each trading day with (slightly) different parameters in each calculation. Moreover, the problems appearing in such financial applications are often high-dimensional, as the dimension usually corresponds to the number of assets/financial contracts in the considered trading portfolio.

It is a widespread issue that the majority of algorithms for such parametric approximation problems suffer from the *curse of dimensionality* (cf., e.g., Bellman, 2010 and Novak & Woźniakowski, 2008b, Chapter 1) in the sense that the computational effort of the approximation methods grows exponentially in the dimension of the approximation problem or in the required approximation accuracy, making them useless for high-dimensional problems. In the *information based complexity* literature there are fundamental lower bounds which generally reveal the impossibility to approximate the solutions of certain classes of high-dimensional approximation problems without the curse of dimensionality among general classes of approximation algorithms; see, for example, Grohs and Voigtlaender (2021), Heinrich (2006), Heinrich and Sindambiwe (1999), and Novak and Woźniakowski (2008a). Developing methods which produce good approximations for high-dimensional problems is thus an exceedingly hard task and, among the general classes of algorithms considered in the above named references, essentially impossible.

In this article we present a new method to tackle high-dimensional parametric approximation problems. Roughly speaking, our strategy is based on the idea to *combine Monte Carlo* (MC) *algorithms* (e.g., standard MC methods, multilevel Monte Carlo (MLMC) methods, or multilevel Picard (MLP) methods) *with stochastic gradient descent* (SGD) *optimization methods* by viewing the employed realizations of random variables in the MC approximation as training parameters for the SGD optimization method. In other words, in this approach we intend to employ SGD optimization methods not to train standard artificial neural networks (ANNs) but to learn random variables appearing in MC approximations.

To make this idea more concrete, we now sketch this *learning the random variables* (LRV) approximation strategy in the context of a basic example of a parametric approximation problem. Let $\mathfrak{p} \in \mathbb{N}$, suppose that we intend to approximate a target function $u : [0,1]^{\mathfrak{p}} \to \mathbb{R}$, let $\phi : [0,1]^{\mathfrak{p}} \times \mathbb{R} \to \mathbb{R}$ be measurable and bounded, let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, let $\mathcal{W} : \Omega \to \mathbb{R}$ be a random variable, and suppose that $u : [0,1]^{\mathfrak{p}} \to \mathbb{R}$ admits the probabilistic representation that for all $p \in [0,1]^{\mathfrak{p}}$ we have that

$$u(p) = \mathbb{E}\left[\phi(p, \mathcal{W})\right] \tag{1}$$

(parametric integration problem; cf., e.g., Cohen & DeVore, 2015 and Heinrich & Sindambiwe, 1999). We note that we only chose the random variable $\mathcal{W}$ to be 1-dimensional for simplicity and refer to Section 2 for the case when $\mathcal{W}$ is a possibly high-dimensional random variable. Our first step to derive the proposed approximation algorithm is to recall standard MC approximations for the parametric expectation in (1). Let $\mathfrak{M} \in \mathbb{N}$, let $W^{m,\mathfrak{m}} : \Omega \to \mathbb{R}$, $m, \mathfrak{m} \in \mathbb{N}_0$, be i.i.d. random variables which satisfy for all $A \in \mathcal{B}(\mathbb{R})$ that $\mathbb{P}(W^{0,0} \in A) = \mathbb{P}(\mathcal{W} \in A)$, and observe for all $p \in [0,1]^{\mathfrak{p}}$ that

$$u(p) = \mathbb{E}\left[\phi(p, \mathcal{W})\right] = \mathbb{E}\left[\phi(p, W^{0,1})\right] \approx \tfrac{1}{\mathfrak{M}}\Big[\textstyle\sum_{\mathrm{m}=1}^{\mathfrak{M}} \phi(p, W^{0,\mathrm{m}})\Big]. \tag{2}$$

In the next step we introduce a parametric function on $[0,1]^{\mathfrak{p}}$ with parameter set $\mathbb{R}^{\mathfrak{M}}$ to reformulate the standard MC approximation in (2) in such a way that the random variables $W^{0,m} : \Omega \to \mathbb{R}$, $m \in \{1, 2, \dots, \mathfrak{M}\}$, correspond to the parameters of the parametric function. More precisely, let $\mathcal{N} : [0,1]^{\mathfrak{p}} \times \mathbb{R}^{\mathfrak{M}} \to \mathbb{R}$ satisfy for all $p \in [0,1]^{\mathfrak{p}}$, $\theta = (\theta_1, \dots, \theta_{\mathfrak{M}}) \in \mathbb{R}^{\mathfrak{M}}$ that $\mathcal{N}(p, \theta) = \frac{1}{\mathfrak{M}}[\sum_{\mathrm{m}=1}^{\mathfrak{M}} \phi(p, \theta_{\mathrm{m}})]$ and note that (2) suggests for all $p \in [0,1]^{\mathfrak{p}}$ that

$$u(p) \approx \tfrac{1}{\mathfrak{M}}\Big[\textstyle\sum_{\mathrm{m}=1}^{\mathfrak{M}} \phi(p, W^{0,\mathrm{m}})\Big] = \mathcal{N}\big(p, (W^{0,1}, W^{0,2}, \dots, W^{0,\mathfrak{M}})\big). \tag{3}$$

Next we employ the SGD optimization method to train the right hand side of (3) in search of “better random realizations” to approximate the target function u than those provided by the random variables $W^{0,m} : \Omega \to \mathbb{R}$, $m \in \{1, 2, \dots, \mathfrak{M}\}$, in the standard MC approximation in (3). The random variables $W^{0,m} : \Omega \to \mathbb{R}$, $m \in \{1, 2, \dots, \mathfrak{M}\}$, on the right hand side of (3) then only supply the initial guess in the SGD training procedure. More specifically, let $P_{m,\mathbf{m}} : \Omega \to [0,1]^{\mathfrak{p}}$, $m, \mathbf{m} \in \mathbb{N}$, be i.i.d. random variables with continuous uniform distribution, let $\mathbf{M} \in \mathbb{N}$, for every $m \in \mathbb{N}$ let $F_m : \mathbb{R}^{\mathfrak{M}} \times \Omega \to \mathbb{R}$ satisfy for all $\theta = (\theta_1, \dots, \theta_{\mathfrak{M}}) \in \mathbb{R}^{\mathfrak{M}}$ that

$$F_m(\theta) = \tfrac{1}{\mathbf{M}}\Big[\textstyle\sum_{\mathbf{m}=1}^{\mathbf{M}} \left|\phi\left(P_{m,\mathbf{m}}, W^{m,\mathbf{m}}\right) - \mathcal{N}(P_{m,\mathbf{m}}, \theta)\right|^2\Big], \tag{4}$$

assume for all $p \in [0,1]^{\mathfrak{p}}$ that $(\mathbb{R} \ni w \mapsto \phi(p, w) \in \mathbb{R}) \in C^1(\mathbb{R}, \mathbb{R})$, for every $m \in \mathbb{N}$ let $G_m : \mathbb{R}^{\mathfrak{M}} \times \Omega \to \mathbb{R}^{\mathfrak{M}}$ satisfy for all $\theta \in \mathbb{R}^{\mathfrak{M}}$, $\omega \in \Omega$ that $G_m(\theta, \omega) = (\nabla_\theta F_m)(\theta, \omega)$, let $(\gamma_m)_{m \in \mathbb{N}} \subseteq$

$(0,\infty)$, and let $\Theta : \mathbb{N}_0 \times \Omega \to \mathbb{R}^{\mathfrak{M}}$ satisfy for all $m \in \mathbb{N}$ that

$$\Theta_0 = (W^{0,1}, W^{0,2}, \ldots, W^{0,\mathfrak{M}}) \qquad \text{and} \qquad \Theta_m = \Theta_{m-1} - \gamma_m G_m(\Theta_{m-1}). \tag{5}$$

Note that the recursion in (5) describes nothing else but the standard SGD optimization method with the learning rate schedule $\mathbb{N} \ni m \mapsto \gamma_m \in \mathbb{R}$. For every sufficiently large $m \in \mathbb{N}$ we then propose to employ the random function

$$[0,1]^{\mathfrak{p}} \times \Omega \ni (p,\omega) \mapsto \mathcal{N}(p, \Theta_m(\omega)) \in \mathbb{R} \tag{6}$$

as an approximation for the target function $[0,1]^{\mathfrak{p}} \ni p \mapsto u(p) \in \mathbb{R}$. Note that the set $[0,1]^{\mathfrak{p}} \ni p \mapsto \mathcal{N}(p,\theta) \in \mathbb{R}$, $\theta \in \mathbb{R}^{\mathfrak{M}}$, of potential approximating functions for $u : [0,1]^{\mathfrak{p}} \to \mathbb{R}$ does not consist of standard fully-connected feedforward ANNs but is a very problem specific class of approximating functions determined by the MC method in (2). In light of this and of the fact that the expression in (3) resembles the definition of single layer fully-connected feedforward ANNs we refer to this class of approximating functions as *MC neural networks*. Moreover, observe that through (3) the MC method also naturally specifies a favorable initializing law for the SGD training procedure in (5).

In the special case of the LRV strategy illustrated in (2)–(6) above, the standard MC approximation method serves as a *proposal algorithm* for the SGD training procedure. More generally, the LRV strategy can, in principle, be used on any high-dimensional approximation problem as soon as there is a reasonable stochastic proposal algorithm for the considered approximation problem available. The LRV strategy thereby naturally specifies the compositional architecture of the involved approximating functions and also naturally specifies the initializing law in the SGD training procedure for each specific approximation problem. In particular, the LRV strategy can be used with the standard MC method (see Section 2), the MC-Euler-Maruyama method (see Section 3), or the MLMC method (see Section 4) as the proposal algorithms to approximate solutions of stochastic differential equations (SDEs) and Kolmogorov PDEs, respectively, and the LRV strategy can be used with the MLP method (see Section 5) as the proposal algorithm to approximate solutions of semilinear PDEs.

We now illustrate the effectiveness of the LRV strategy on a simple but famous numerical example, where the LRV strategy leads to impressively precise approximations. In Table 1 we present results for the approximative computation of prices of European call options in the Black-Scholes model by means of the deep learning method induced by Beck, Becker, Grohs et al. (2021) with 140000 Adam (see Kingma & Ba, 2014) training steps (rows 2–5 in Table 1), by means of the standard MC method with 32768 MC samples (row 6 in Table 1), by means of the antithetic MC method with 32768 MC samples (row 7 in Table 1), by means of the Quasi-Monte Carlo (QMC) method using Sobol sequences with 32768 QMC samples (row 8 in Table 1), and by means of the antithetic QMC method using Sobol sequences with 32768 QMC samples (row 9 in Table 1), by means of the LRV strategy with the standard MC method as the proposal algorithm with 140000 Adam training steps (row 10 in Table 1), and by means of the LRV strategy with the antithetic MC method as the proposal algorithm with 140000 Adam training steps (row 11 in Table 1). In Table 1 the L^2-error and the L^∞-error have been computed approximately on the region

$$(\xi, T, r, \sigma, K) \in [90, 110] \times \left[\tfrac{1}{100}, 1\right] \times \left[-\tfrac{1}{10}, \tfrac{1}{10}\right] \times \left[\tfrac{1}{100}, \tfrac{1}{2}\right] \times [90, 100] \tag{7}$$

TABLE 1 Pricing European call options in the Black-Scholes model.

Approximation method	Number of trainable parameters	Number of MC/QMC samples	L^2-approx. error	L^∞-approx. error	Training time in seconds	Evaluation time in seconds
ANNs (2 hidden layers, 64 neurons each)	4609	-	0.006479	0.266781	181.95	0.78
ANNs (2 hidden layers, 128 neurons each)	17409	-	0.004411	0.197840	191.05	0.74
ANNs (3 hidden layers, 64 neurons each)	8769	-	0.002791	0.124099	196.73	0.81
ANNs (3 hidden layers, 128 neurons each)	33921	-	0.001813	0.097105	199.35	0.84
MC	-	32768	0.081914	1.079885	0	17.12
MC (antithetic)	-	32768	0.045043	0.593046	0	35.07
QMC	-	32768	0.002585	0.011879	0	15.48
QMC (antithetic)	-	32768	0.000466	0.002769	0	33.65
LRV	8192	8192	0.000134	0.002138	1773.54	4.17
LRV (antithetic)	8192	8192	0.000132	0.002210	2799.30	7.73

Abbreviations: ANN, artificial neural network; LRV, learn the random variable; MC, Monte Carlo; QMC, Quasi-Monte Carlo.

TABLE 2 Numerical simulations for the LRV strategy in case of the Black-Scholes model for European call options on one underlying described in Subsection 7.2 (5-dimensional approximation problem) trained with batch size $\mathbf{M} = 8192$ and approximate reference solutions ($e = 0$) based on $\mathscr{M} = 4096$ antithetic MC samples (cf. yellow dots and lines in Figures 1 and 2).

$\mathfrak{M}$	a	Number of trainable parameters	L^1-approx. error	L^2-approx. error	L^∞-approx. error	Training time in seconds	Evaluation time for 8 192 000 evaluations in seconds
32	0	32	0.003183	0.005893	0.068957	728.00	0.60
32	1	32	0.001504	0.003016	0.029396	731.45	0.62
64	0	64	0.001200	0.002350	0.024142	726.46	0.62
64	1	64	0.000639	0.001268	0.012222	735.63	0.68
128	0	128	0.000494	0.000977	0.009113	739.05	0.57
128	1	128	0.000281	0.000555	0.005386	749.49	0.62
256	0	256	0.000228	0.000436	0.004850	742.17	0.61
256	1	256	0.000147	0.000270	0.002682	774.72	0.60
512	0	512	0.000122	0.000239	0.006127	768.41	0.59
512	1	512	0.000085	0.000153	0.001942	818.77	0.65
1024	0	1024	0.000099	0.000208	0.007090	811.28	0.66
1024	1	1024	0.000068	0.000127	0.002126	950.48	0.89
2048	0	2048	0.000080	0.000146	0.002312	944.56	0.87
2048	1	2048	0.000060	0.000120	0.001818	1241.45	2.32
4096	0	4096	0.000061	0.000123	0.002020	1237.02	2.32
4096	1	4096	0.000068	0.000136	0.001901	1782.10	4.20
8192	0	8192	0.000065	0.000134	0.002138	1773.54	4.17
8192	1	8192	0.000064	0.000132	0.002210	2799.30	7.73

using 8 192 000 evaluations of the considered approximation method. In (7) we have that ξ stands for the initial price, that T stands for the time of maturity, that r stands for the drift rate, that σ stands for the volatility, and that K stands for the strike price; see, for example, Becker et al. (2021, Lemma 4.4) (with $\mathrm{c} = 0$ in the notation of Becker et al., 2021, Lemma 4.4). The reference solution values to compute the errors in Table 1 have been computed with the famous Black-Scholes formula (see, e.g., Becker et al., 2021, Lemma 4.4 or (116) in Subsection 7.2). The evaluation time corresponds to the time required to compute 8 192 000 evaluations. We note that the training and evaluation times[1] of the LRV strategy are significantly longer when compared to the deep learning method induced by Beck, Becker, Grohs et al. (2021) even when the considered MC neural networks involve less arithmetic operations than the considered ANNs. This is likely attributed to the efficient implementation and parallelization of standard feedforward ANNs in Tensorflow and our own implementation of MC neural networks, which may be slightly less computationally efficient. The numbers in Table 1 are taken from Tables 2, 3, 4, and 5 in Subsection 7.2. We refer to Subsection 7.2 for more details on the results in Table 1.

[1] The numerical experiments have been performed in TensorFlow 2.12 running on a system equipped with an NVIDIA GeForce RTX 4090 GPU with 24 GB Graphics RAM.

TABLE 3 Numerical simulations for the deep learning method induced by Beck, Becker, Grohs et al. (2021) in case of the Black-Scholes model European call options on one underlying described in Subsection 7.2 (5-dimensional approximation problem).

Number of hidden layers	Number of neurons on each hidden layer	Number of trainable parameters	L^1-approx. error	L^2-approx. error	L^∞-approx. error	Training time in seconds	Evaluation time for 8 192 000 evaluations in seconds
2	8	129	0.068139	0.099111	1.308262	182.59	0.91
2	16	385	0.019551	0.029058	0.626703	184.90	0.81
2	32	1281	0.006874	0.010994	0.294981	182.46	0.73
2	64	4609	0.003855	0.006479	0.266781	181.95	0.78
2	128	17409	0.002519	0.004411	0.197840	191.05	0.74
3	8	201	0.043577	0.064298	1.055705	210.32	0.96
3	16	657	0.011687	0.017968	0.505980	201.72	0.77
3	32	2337	0.003797	0.006030	0.262204	205.58	0.76
3	64	8769	0.001761	0.002791	0.124099	196.73	0.81
3	128	33921	0.001244	0.001813	0.097105	199.35	0.84

TABLE 4 Numerical simulations for the standard and the antithetic MC method in case of the Black-Scholes model for European call options on one underlying described in Subsection 7.2 (5-dimensional approximation problem).

Number of MC samples	MC Method 0: standard 1: antithetic	L^1-approx. error	L^2-approx. error	L^∞-approx. error	Evaluation time for 8 192 000 evaluations in seconds
32	0	1.631470	2.620134	41.858883	1.69
32	1	0.857496	1.443121	21.981831	0.65
64	0	1.154554	1.851235	28.730389	0.62
64	1	0.607091	1.019940	13.166348	0.60
128	0	0.816507	1.308833	17.123699	0.59
128	1	0.429601	0.721680	8.736799	0.66
256	0	0.578006	0.925793	12.759777	0.63
256	1	0.303619	0.510076	6.837311	0.66
512	0	0.408652	0.654518	8.377937	0.63
512	1	0.214879	0.360810	5.057564	0.67
1024	0	0.289187	0.463213	6.491411	0.63
1024	1	0.151856	0.254823	3.114460	1.09
2048	0	0.204313	0.327113	3.708378	0.94
2048	1	0.107378	0.180250	2.206034	2.57
4096	0	0.144438	0.231285	2.808947	2.56
4096	1	0.075998	0.127575	1.839296	4.87
8192	0	0.102109	0.163498	1.972134	4.67
8192	1	0.053718	0.090149	1.181227	9.20
16384	0	0.072263	0.115714	1.562056	8.89
16384	1	0.037983	0.063745	0.771879	17.78
32768	0	0.051127	0.081914	1.079885	17.12
32768	1	0.026839	0.045043	0.593046	35.07

Note that Table 1 indicates that the algorithm obtained by the LRV strategy not only produces very accurate prices in the L^2-norm, but even has a very high accuracy in the uniform L^∞-norm. Concretely, for this 5-dimensional approximation problem, this strongly suggests that for any choice of parameters in the region considered in (7), the LRV strategy offers an approximation with an error smaller than $\frac{2}{1000}$. Based on this, on the other numerical results presented in this article, as well as on preliminary analytic investigations (cf., e.g., Gonon et al., 2021, Lemma 2.16) we conjecture that the LRV strategy can overcome the curse of dimensionality for certain classes of parametric PDE problems in the L^∞-norm. We would like to emphasize that large classes of algorithms for such approximation problems have been shown not to be able to overcome the curse of dimensionality, see, for example, Heinrich and Sindambiwe (1999, Theorem 2.4), Heinrich (2006, Theorem 1), and Grohs and Voigtlaender (2021). However, our conjecture does not contradict the general lower bounds established in the above mentioned references, due to the fact that the LRV strategy does not belong to the class of algorithms considered in the above mentioned references: roughly speaking, in the LRV strategy there are two stages of computational

TABLE 5 Numerical simulations for the standard and the antithetic QMC method with Sobol sequences in case of the Black-Scholes model for European call options on one underlying described in Subsection 7.2 (5-dimensional approximation problem).

Number of MC samples	QMC Method 0: standard 1: antithetic	L^1-approx. error	L^2-approx. error	L^∞-approx. error	Evaluation time for 8 192 000 evaluations in seconds
32	0	0.857061	1.117799	4.344944	0.85
32	1	0.237763	0.366264	1.820515	0.64
64	0	0.493400	0.642453	2.530792	0.64
64	1	0.122386	0.188951	0.944592	0.64
128	0	0.276729	0.360582	1.431679	0.69
128	1	0.062681	0.097161	0.494709	0.64
256	0	0.152859	0.199531	0.797966	0.66
256	1	0.032036	0.049889	0.257656	0.75
512	0	0.083264	0.109023	0.448029	0.71
512	1	0.016332	0.025576	0.135283	0.78
1024	0	0.044996	0.059125	0.249985	0.75
1024	1	0.008340	0.013135	0.070560	1.08
2048	0	0.024177	0.031904	0.137550	0.93
2048	1	0.004245	0.006730	0.037239	2.55
4096	0	0.012906	0.017105	0.074234	2.35
4096	1	0.002163	0.003453	0.019505	4.70
8192	0	0.006863	0.009143	0.040550	4.23
8192	1	0.001101	0.001769	0.010147	8.65
16384	0	0.003636	0.004868	0.022209	7.70
16384	1	0.000561	0.000908	0.005302	16.68
32768	0	0.001920	0.002585	0.011879	15.48
32768	1	0.000286	0.000466	0.002769	33.65

procedures, the *main computational procedure* in which the “best random variables” are learned through SGD (corresponding to the 6th column in Table 1) and the *evaluation procedure* where the computed approximation of the target function is evaluated (corresponding to the 7th column in Table 1). In the LRV strategy we consider the situation where it is allowed to perform function evaluations both during the main computational procedure *and* the evaluation procedure while the lower bounds in Heinrich and Sindambiwe (1999), Heinrich (2006), Grohs and Voigtlaender (2021) consider the situation where function evaluations are only allowed during the main computational procedure *but not* during the evaluation procedure. The LRV strategy being outside of the classes of algorithms considered in the above named references is thus not constrained by the established lower bounds and hence holds the potential to overcome the curse of dimensionality, even in the L^∞-norm. We note that in practically relevant situations, just as in the considered derivative pricing problem, it is often possible to perform function evaluations also in the evaluation procedure, thus making the LRV strategy an applicable method for practically relevant approximation problems.

We now compare the proposed LRV strategy to existing algorithms and computational methods in the scientific literature. As the LRV strategy employs SGD-type methods to “learn” parametric functions it is related to deep learning methods, where instead of random variables, optimal weights of ANNs are “learned”. From this point of view, the LRV strategy can be seen as a machine learning approach where instead of employing generic ANNs, very problem specific parametric functions are used, which contain a lot of human insight about the problem at hand. There is a plethora of deep learning methods for the approximation of PDEs, which have been developed recently, and seem to be very effective for the approximation of high-dimensional PDEs: cf., for example, E et al. (2017), Han et al. (2018), Sirignano and Spiliopoulos (2018), E and Yu (2018), Fujii et al. (2019), Dissanayake and Phan-Thien (1994), Lagaris et al. (1998), Jianyu et al. (2003), Chen, Huang et al. (2020), Chan-Wai-Nam et al. (2019), Germain et al. (2020), Han and Long (2020), Han, Lu et al. (2020), Henry-Labordere (2017), Huré et al. (2020), Jacquier and Oumgari (2019), Ji et al. (2020), Kremsner et al. (2020), Pereira et al. (2019), Pham et al. (2019), Raissi (2018), Beck, Becker, Cheridito et al. (2021), Beck, Becker, Grohs et al. (2021), Berner et al. (2020), Han, Nica et al. (2020), Nüsken and Richter (2021a), Samaniego et al. (2020), Becker et al. (2019), Becker et al. (2021), Chen and Wan (2019), Goudenège et al. (2019), Lyu et al. (2020), Lu et al. (2019), Zang et al. (2020), Han and Hu (2020), Lye et al. (2020), Zhu et al. (2019), Longo et al. (2020), Nüsken and Richter (2021b), Martin et al. (2020). We refer to the survey articles Beck et al. (2023), Blechschmidt and Ernst (2021), Germain et al. (2021), E et al. (2022) for a more detailed overview. For methods which are specifically designed for parametric PDEs we refer to, for example, Vidales et al. (2018), Khoo et al. (2021), Bhattacharya et al. (2021), Berner et al. (2020), O’Leary-Roseberry et al. (2022). In addition, we want to highlight a connection between the LRV strategy and the deep learning method for Kolmogorov PDEs developed in Beck, Becker, Grohs et al. (2021) and further specialized to parametric Kolmogorov PDEs in Berner et al. (2020). Very roughly speaking, the work Berner et al. (2020) is concerned with approximating a function $u : \Gamma \to \mathbb{R}$ given for all $\gamma \in \Gamma$ by $u(\gamma) = \mathbb{E}[\varphi_\gamma(S_\gamma)]$ where Γ is a parameter set, where $d \in \mathbb{N}$, where $(\Omega, \mathcal{F}, \mathbb{P})$ is a probability space, where $\varphi_\gamma : \mathbb{R}^d \to \mathbb{R}$, $\gamma \in \Gamma$, are parametric functions, and where for every $\gamma \in \Gamma$ the random variable $S_\gamma : \Omega \to \mathbb{R}^d$ is the solution of an SDE parameterized by the parameter γ. In Berner et al. (2020), they propose to approximate u by looking for a function $f : \Gamma \to \mathbb{R}$ within a certain class of ANNs which aims to minimize $\mathbb{E}[(f(\Lambda) - \varphi_\Lambda(Y_\Lambda))^2]$ where $\Lambda : \Omega \to \Gamma$ is a random variable and for every $\gamma \in \Gamma$ we have that $Y_\gamma : \Omega \to \mathbb{R}^d$ is an approximation of S_γ (e.g., an Euler-Maruyama approximation). If the class of ANNs in which an optimal function f is looked for is replaced by parametric functions induced by a proposal algorithm this becomes a special case of the LRV framework presented in this article.

Furthermore, the LRV strategy can be associated with a broad subcategory of machine learning called *algorithm learning* (cf., e.g., Borgerding et al., 2016; Chen, Li et al., 2020; Chen et al., 2018; Chen, Zhang et al., 2020; Gregor & LeCun, 2010; Mensch & Blondel, 2018; Towell & Shavlik, 1994; Wilder et al., 2019; Yoon et al., 2013). Roughly speaking, algorithm learning refers to the idea to employ existing algorithms with known empirical or theoretical qualities as a basis to construct or extend ANNs or more general parametric function families. In many cases the employed algorithm relies on certain hyper-parameters (e.g., the learning rates in case of SGD-type methods: cf., e.g., Chen, Zhang et al., 2020) which typically are added to the set of trainable parameters of the ANN. It is in this point that the LRV strategy differs from existing algorithm learning methods since the LRV strategy considers the random variables and not the hyper-parameters of the proposal algorithm as learnable parameters. It thereby has the advantage that the initialization of all trainable parameters is implicitly given through the proposal algorithm. However, the LRV strategy could very well be combined with ideas of algorithm learning. We leave this task open for future research.

Another tranche of literature connected to the LRV strategy are QMC methods (cf., e.g., Caflisch, 1998; Drovandi & Tran, 2018; Morokoff & Caflisch, 1995). Roughly speaking, the idea of QMC methods is to replace uniformly distributed random variables of the MC method by a more suitable sequence of deterministic points to obtain a higher rate of convergence. The idea to improve the choice of random variables in MC methods is a common feature of QMC methods with the LRV strategy. A key difference between the two methods is that QMC methods construct new integration points which have good properties for a wide class of integrands while the LRV strategy aims to "learn" new integration points which are specific to each considered integrand.

Next we illustrate a link between the LRV strategy and quantization methods (cf., e.g., Altmayer et al., 2014; Dereich et al., 2013; Frikha & Sagna, 2012; Müller-Gronbach & Ritter, 2013; Pagès, 1998; Pagès, 2015; Pagès et al., 2004; Pagès & Printems, 2003, 2005; Pham et al., 2005; Rudd et al., 2017). Quantization methods are concerned with approximating a continuously distributed random variable by a random variable with a finite image. Quantization was first introduced for signal processing and information theory (cf., e.g., Graf & Luschgy, 2000; Pagès & Printems, 2003) but can also be used for the numerical approximation of expectations involving the original random variable (cf., e.g., Pagès, 1998; Pagès & Printems, 2003). For the latter, the original random variable in the expectation is replaced by its quantization, resulting in an expected value which can easily be computed by a finite sum of weighted function evaluations. A good quantization is typically found with optimization methods such as, the Newton method in low dimensions or SGD-type optimization methods in high-dimensions (cf., e.g., Pagès, 2015; Pagès et al., 2004; Pagès & Printems, 2003; Pham et al., 2005). To make this more concrete we now roughly illustrate this in the context of the setting described above. For all $\theta = (\theta_1, \dots, \theta_{\mathfrak{M}}) \in \mathbb{R}^{\mathfrak{M}}$, $C_1, C_2, \dots, C_{\mathfrak{M}} \in \mathcal{B}(\mathbb{R})$ with $\forall\, m, n \in \{1, 2, \dots, \mathfrak{M}\}, m \neq n \,:\, C_m \cap C_n = \{\}$ and $\cup_{m=1}^{\mathfrak{M}} C_m = \mathbb{R}$ we consider a quantization $Q_{\theta, C_1, C_2, \dots, C_{\mathfrak{M}}} : \Omega \to \mathbb{R}^{\mathfrak{M}}$ of $\mathcal{W}$ given by $Q_{\theta, C_1, C_2, \dots, C_{\mathfrak{M}}} = \sum_{\mathrm{m}=1}^{\mathfrak{M}} \theta_{\mathrm{m}} \mathbb{1}_{C_{\mathrm{m}}}(\mathcal{W})$. A good quantization is found by minimizing an error between the quantization and the original random variable, for example, the squared L^2-error given for all $\theta = (\theta_1, \dots, \theta_{\mathfrak{M}}) \in \mathbb{R}^{\mathfrak{M}}$, $C_1, C_2, \dots, C_{\mathfrak{M}} \in \mathcal{B}(\mathbb{R})$ by

$$\mathbb{E}\left[\| \mathcal{W} - Q_{\theta, C_1, C_2, \dots, C_{\mathfrak{M}}} \|^2\right], \tag{8}$$

with an SGD-type optimization method. Once appropriate $\theta = (\theta_1, \dots, \theta_{\mathfrak{M}}) \in \mathbb{R}^{\mathfrak{M}}$, $C_1, C_2, \dots, C_{\mathfrak{M}} \in \mathcal{B}(\mathbb{R})$ with $\forall\, m, n \in \{1, 2, \dots, \mathfrak{M}\}, m \neq n \,:\, C_m \cap C_n = \{\}$ and $\cup_{m=1}^{\mathfrak{M}} C_m = \mathbb{R}$ have been found, they can be employed to approximate the expectation in (1) for every $p \in \mathfrak{P}$ by

$$u(p) = \mathbb{E}[\phi(p, \mathcal{W})] \approx \mathbb{E}\left[\phi(p, Q_{\theta, C_1, C_2, \dots, C_{\mathfrak{M}}})\right] = \textstyle\sum_{\mathrm{m}=1}^{\mathfrak{M}} \phi(p, \theta_{\mathrm{m}}) \mathbb{P}(\mathcal{W} \in C_{\mathrm{m}}). \tag{9}$$

Note that this approximation has a similar form as the proposed LRV approximation in (6) and in both cases the points at which ϕ is evaluated are found through an SGD-type optimization method. The main difference is that the optimization problem in (8), which is used to determine the evaluation points in (9), only depends on the random variable $\mathcal{W}$ whereas the optimization problem (see (4)) to determine the evaluation points in (6) in the LRV strategy depends on the random variable $\mathcal{W}$ and the function ϕ.

The concept of optimizing random MC samples, a central aspect of the LRV strategy, also appears in the Bayesian statistics literature in the context of inducing points in Bayesian learning (cf., e.g., Betancourt, 2017; Dellaportas & Titsias, 2019; Ranganath et al., 2014; Snelson & Ghahramani, 2005). Such inducing points are employed to find sparse representation for large data sets. One way to find good inducing points is to first sample them randomly and then optimize them for example with gradient based methods (cf., e.g., Snelson & Ghahramani, 2005), which bears some resemblance to the LRV strategy.

The reminder of this article is structured as follows. In Sections 2, 3, 4, and 5 we present the LRV strategy for increasingly complex approximation problems and proposal algorithms. As proposal algorithms we consider the standard MC method in Section 2, the MC-Euler method in Section 3, the MLMC method in Section 4, and the MLP method in Section 5. In Section 6 we present the most general case of a generic proposal algorithm, which includes all previous sections as special cases. The results of numerical experiments for the LRV approximation strategy are presented in Section 7. Specifically, we consider 1-dimensional Black-Scholes equations for European call options (resulting in a 5-dimensional parametric approximation problem) in Subsection 7.2, we consider 3-dimensional Black-Scholes equations for worst-of basket put options (resulting in a 15-dimensional parametric approximation problem) in Subsection 7.3, we consider 3-dimensional Black-Scholes equations for average basket put options with knock-in barriers (resulting in a 16-dimensional parametric approximation problem) in Subsection 7.4, and we consider stochastic Lorentz equations (resulting in a 10-dimensional parametric approximation problem) in Subsection 7.5.

2 | LEARNING THE RANDOM VARIABLES (LRV) STRATEGY IN THE CASE OF MONTE CARLO APPROXIMATIONS

In this section we employ the LRV strategy for the approximation of parametric expectations (see (10) in Subsection 2.1). This includes as a special case the approximative pricing of European options in the Black-Scholes model, as illustrated in Subsection 2.2. The LRV strategy based on standard MC averages as proposal algorithms for the general parametric approximation problem of Subsection 2.1 is elaborated in Subsections 2.3, 2.4, 2.5, and 2.6. The resulting method is summarized in a single framework in Subsection 2.7.

2.1 | Parametric expectations involving vector valued random variables

Let $\mathfrak{p}, \mathbf{d} \in \mathbb{N}$, let $\mathfrak{P} \subseteq \mathbb{R}^{\mathfrak{p}}$ be measurable, let $u : \mathfrak{P} \to \mathbb{R}$ be a function, let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, let $\mathcal{W} : \Omega \to \mathbb{R}^{\mathbf{d}}$ be a random variable, let $\phi : \mathfrak{P} \times \mathbb{R}^{\mathbf{d}} \to \mathbb{R}$ be measurable, assume for all $p \in \mathfrak{P}$ that $\mathbb{R}^{\mathbf{d}} \ni w \mapsto \phi(p, w) \in \mathbb{R}$ is continuously differentiable, and assume for all $p \in \mathfrak{P}$ that $\mathbb{E}[|\phi(p, \mathcal{W})|] < \infty$ and

$$u(p) = \mathbb{E}[\phi(p, \mathcal{W})]. \tag{10}$$

The goal of this section is to derive an algorithm to approximately compute the function $u : \mathfrak{P} \to \mathbb{R}$ given through the parametric expectation in (10).

2.2 | Approximative pricing of European options in the Black-Scholes model

Lemma 2.1. *Let $\xi, T, \sigma \in (0, \infty)$, $r \in \mathbb{R}$, let $\mathfrak{N} : \mathbb{R} \to \mathbb{R}$ satisfy for all $x \in \mathbb{R}$ that $\mathfrak{N}(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} \exp(-\frac{y^2}{2})\, \mathrm{d}y$, let $(\Omega, \mathcal{F}, (\mathbb{F}_t)_{t\in[0,T]}, \mathbb{P})$ be a filtered probability space which satisfies the usual conditions, let $W : [0,T] \times \Omega \to \mathbb{R}$ be a standard $(\mathbb{F}_t)_{t\in[0,T]}$-Brownian motion with continuous sample paths, and let $X : [0,T] \times \Omega \to \mathbb{R}$ be an $(\mathbb{F}_t)_{t\in[0,T]}$-adapted stochastic process with continuous sample paths which satisfies that for all $t \in [0,T]$ it holds $\mathbb{P}$-a.s. that*

$$X_t = \xi + r \int_0^t X_s \, \mathrm{d}s + \sigma \int_0^t X_s \, \mathrm{d}W_s. \tag{11}$$

Then

(i) it holds for all $t \in [0,T]$ that $\mathbb{P}(X_t = \exp([r - \frac{\sigma^2}{2}]t + \sigma W_t)\xi) = 1$ and
(ii) it holds for all $K \in \mathbb{R}$ that

$$\begin{aligned} &\mathbb{E}[\exp(-rT)\max\{X_T - K, 0\}] \\ &= \begin{cases} \xi\, \mathfrak{N}\left(\frac{\left(r+\frac{\sigma^2}{2}\right)T+\ln(\xi/K)}{\sigma\sqrt{T}}\right) - K\exp(-r\,T)\,\mathfrak{N}\left(\frac{\left(r-\frac{\sigma^2}{2}\right)T+\ln(\xi/K)}{\sigma\sqrt{T}}\right) & : K > 0 \\ \xi - K\exp(-r\,T) & : K \leq 0. \end{cases} \end{aligned} \tag{12}$$

Proof of Lemma 2.1. Observe that, for example, Becker et al. (2021, item (i) in Proposition 4.3) implies that for all $t \in [0,T]$ it holds $\mathbb{P}$-a.s. that

$$X_t = \exp\left(\left[r - \frac{\sigma^2}{2}\right]t + \sigma W_t\right)\xi = \exp\left(\left(r - \frac{\sigma^2}{2}\right)t + \ln(\xi) + \sigma W_t\right). \tag{13}$$

This establishes item (i). Moreover, note that, for example, Becker et al. (2021, Lemma 4.4) and (13) show for all $K \in \mathbb{R}$ that

$$\begin{aligned} &\mathbb{E}[\exp(-rT)\max\{X_T - K, 0\}] = \exp(-rT)\,\mathbb{E}\left[\max\left\{\exp\left(\left(r - \frac{\sigma^2}{2}\right)T + \ln(\xi) + \sigma W_T\right) - K, 0\right\}\right] \\ &= \begin{cases} \xi\, \mathfrak{N}\left(\frac{\left(r+\frac{\sigma^2}{2}\right)T+\ln(\xi/K)}{\sigma\sqrt{T}}\right) - K\exp(-r\,T)\,\mathfrak{N}\left(\frac{\left(r-\frac{\sigma^2}{2}\right)T+\ln(\xi/K)}{\sigma\sqrt{T}}\right) & : K > 0 \\ \xi - K\exp(-r\,T) & : K \leq 0 \end{cases}. \end{aligned} \tag{14}$$

The proof of Lemma 2.1 is thus complete. □

In the case where $\mathfrak{p} = 5$, $\mathbf{d} = 1$, and $\mathfrak{P} = [90, 110] \times [0.01, 1] \times [-0.1, 0.1] \times [0.01, 0.5] \times [90, 110]$, where for all $p = (\xi, T, r, \sigma, K) \in \mathfrak{P}$, $w \in \mathbb{R}$ it holds that $\phi(p, w) = \exp(-rT) \max\{\exp([r - \frac{\sigma^2}{2}]T + T^{1/2}\sigma w)\xi - K, 0\}$, where $\mathcal{W}$ is a standard normal random variable, and where $\mathfrak{N} : \mathbb{R} \to \mathbb{R}$ satisfies for all $x \in \mathbb{R}$ that $\mathfrak{N}(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} \exp(-\frac{y^2}{2})\, \mathrm{d}y$ observe that

(i) it holds for all $p = (\xi, T, r, \sigma, K) \in \mathfrak{P}$ that

$$\begin{aligned} u(p) &= \mathbb{E}\Big[\exp(-rT)\max\Big\{\exp\Big(\Big[r - \tfrac{\sigma^2}{2}\Big]T + T^{1/2}\sigma\mathcal{W}\Big)\xi - K, 0\Big\}\Big] \\ &= \xi\,\mathfrak{N}\left(\frac{\left(r+\frac{\sigma^2}{2}\right)T + \ln(\xi/K)}{\sigma\sqrt{T}}\right) - K\exp(-r\,T)\,\mathfrak{N}\left(\frac{\left(r-\frac{\sigma^2}{2}\right)T + \ln(\xi/K)}{\sigma\sqrt{T}}\right) \end{aligned} \tag{15}$$

(cf. (10) and Lemma 2.1) and

(ii) it holds for all $p = (\xi, T, r, \sigma, K) \in \mathfrak{P}$ that $u(p) \in [0, \infty)$ is the price of an European call option in the Black-Scholes model with initial underlying price ξ, time of maturity T, drift rate r, volatility σ, and strike price K.

2.3 | Monte Carlo approximations

Let $\mathfrak{M} \in \mathbb{N}$ and let $W^{m,\mathbf{m}} : \Omega \to \mathbb{R}^{\mathbf{d}}$, $m, \mathbf{m} \in \mathbb{N}_0$, be i.i.d. random variables which satisfy for all $A \in \mathcal{B}(\mathbb{R}^{\mathbf{d}})$ that $\mathbb{P}(W^{0,0} \in A) = \mathbb{P}(\mathcal{W} \in A)$. Observe that (10) suggests that for all $p \in \mathfrak{P}$ it holds that

$$u(p) = \mathbb{E}[\phi(p, \mathcal{W})] = \mathbb{E}\big[\phi(p, W^{0,1})\big] \approx \frac{1}{\mathfrak{M}}\left[\sum_{\mathbf{m}=1}^{\mathfrak{M}} \phi(p, W^{0,\mathbf{m}})\right]. \tag{16}$$

2.4 | Replacing the random variables in Monte Carlo approximations

Let $\mathcal{N} : \mathfrak{P} \times \mathbb{R}^{\mathfrak{M}\mathbf{d}} \to \mathbb{R}$ satisfy for all $p \in \mathfrak{P}$, $\theta = (\theta_1, \dots, \theta_{\mathfrak{M}\mathbf{d}}) \in \mathbb{R}^{\mathfrak{M}\mathbf{d}}$ that

$$\mathcal{N}(p, \theta) = \frac{1}{\mathfrak{M}}\left[\sum_{\mathbf{m}=1}^{\mathfrak{M}} \phi\big(p, (\theta_{(\mathbf{m}-1)\mathbf{d}+k})_{k\in\{1,2,\dots,\mathbf{d}\}}\big)\right]. \tag{17}$$

Note that (16) and (17) suggest that for all $p \in \mathfrak{P}$ it holds that

$$u(p) \approx \frac{1}{\mathfrak{M}}\left[\sum_{\mathbf{m}=1}^{\mathfrak{M}} \phi\big(p, W^{0,\mathbf{m}}\big)\right] = \mathcal{N}\big(p, (W^{0,1}, W^{0,2}, \dots, W^{0,\mathfrak{M}})\big). \tag{18}$$

2.5 | Random loss functions for fixed random variables in Monte Carlo approximations

Let $\mathbf{M} \in \mathbb{N}$, let $P_{m,\mathbf{m}} : \Omega \to \mathfrak{P}$, $m, \mathbf{m} \in \mathbb{N}$, be i.i.d. random variables, for every $m \in \mathbb{N}$ let $F_m : \mathbb{R}^{\mathfrak{M}\mathbf{d}} \times \Omega \to \mathbb{R}$ satisfy for all $\theta = (\theta_1, \dots, \theta_{\mathfrak{M}\mathbf{d}}) \in \mathbb{R}^{\mathfrak{M}\mathbf{d}}$ that

$$\begin{aligned} F_m(\theta) &= \frac{1}{\mathbf{M}}\left[\sum_{\mathbf{m}=1}^{\mathbf{M}} \left|\phi\left(P_{m,\mathbf{m}}, W^{m,\mathbf{m}}\right) - \mathcal{N}(P_{m,\mathbf{m}}, \theta)\right|^2\right] \\ &= \frac{1}{\mathbf{M}}\left[\sum_{\mathbf{m}=1}^{\mathbf{M}} \left|\phi\left(P_{m,\mathbf{m}}, W^{m,\mathbf{m}}\right) - \frac{1}{\mathfrak{M}}\left[\sum_{\mathbf{m}=1}^{\mathfrak{M}} \phi\big(P_{m,\mathbf{m}}, (\theta_{(\mathbf{m}-1)\mathbf{d}+k})_{k\in\{1,2,\dots,\mathbf{d}\}}\big)\right]\right|^2\right], \end{aligned} \tag{19}$$

and for every $m \in \mathbb{N}$ let $G_m : \mathbb{R}^{\mathfrak{M}\mathbf{d}} \times \Omega \to \mathbb{R}^{\mathfrak{M}\mathbf{d}}$ satisfy for all $\theta \in \mathbb{R}^{\mathfrak{M}\mathbf{d}}$, $\omega \in \Omega$ that

$$G_m(\theta, \omega) = (\nabla_\theta F_m)(\theta, \omega). \tag{20}$$

2.6 | Learning the random variables with stochastic gradient descent

Let $(\gamma_m)_{m\in\mathbb{N}} \subseteq (0, \infty)$ and let $\Theta : \mathbb{N}_0 \times \Omega \to \mathbb{R}^{\mathfrak{M}\mathbf{d}}$ satisfy for all $m \in \mathbb{N}$ that $\Theta_0 = (W^{0,1}, W^{0,2}, \dots, W^{0,\mathfrak{M}})$ and

$$\Theta_m = \Theta_{m-1} - \gamma_m G_m(\Theta_{m-1}). \tag{21}$$

For every sufficiently large $m \in \mathbb{N}$ we propose to employ the random function $\mathfrak{P} \times \Omega \ni (p, \omega) \mapsto \mathscr{N}(p, \Theta_m(\omega)) \in \mathbb{R}$ as an approximation for the target function $\mathfrak{P} \ni p \mapsto u(p) \in \mathbb{R}$ in (10).

2.7 | Description of the proposed approximation algorithm

Framework 2.2. *Let $\mathfrak{p}, \mathbf{d}, \mathfrak{M}, \mathbf{M} \in \mathbb{N}$, $(\gamma_m)_{m\in\mathbb{N}} \subseteq (0, \infty)$, let $\mathfrak{P} \subseteq \mathbb{R}^{\mathfrak{p}}$ be measurable, let $\phi \in C(\mathfrak{P} \times \mathbb{R}^{\mathbf{d}}, \mathbb{R})$, let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, let $P_{m,\mathbf{m}} : \Omega \to \mathfrak{P}$, $m, \mathbf{m} \in \mathbb{N}$, be i.i.d. random variables, let $W^{m,\mathfrak{m}} : \Omega \to \mathbb{R}^{\mathbf{d}}$, $m, \mathfrak{m} \in \mathbb{N}_0$, be i.i.d. random variables, assume that $(P_{m,\mathbf{m}})_{(m,\mathbf{m})\in\mathbb{N}^2}$ and $(W^{m,\mathfrak{m}})_{(m,\mathfrak{m})\in\mathbb{N}^2}$ are independent, assume for all $p \in \mathfrak{P}$ that $\mathbb{R}^{\mathbf{d}} \ni w \mapsto \phi(p, w) \in \mathbb{R}$ is continuously differentiable, for every $m \in \mathbb{N}$ let $F_m : \mathbb{R}^{\mathfrak{M}\mathbf{d}} \times \Omega \to \mathbb{R}$ satisfy for all $\theta = (\theta_1, \dots, \theta_{\mathfrak{M}\mathbf{d}}) \in \mathbb{R}^{\mathfrak{M}\mathbf{d}}$ that*

$$F_m(\theta) = \frac{1}{\mathbf{M}} \left[\sum_{\mathbf{m}=1}^{\mathbf{M}} \left| \phi\big(P_{m,\mathbf{m}}, W^{m,\mathbf{m}}\big) - \frac{1}{\mathfrak{M}} \left[\sum_{\mathfrak{m}=1}^{\mathfrak{M}} \phi\Big(P_{m,\mathbf{m}}, \big(\theta_{(\mathfrak{m}-1)\mathbf{d}+k}\big)_{k\in\{1,2,\dots,\mathbf{d}\}}\Big) \right] \right|^2 \right], \tag{22}$$

for every $m \in \mathbb{N}$ let $G_m : \mathbb{R}^{\mathfrak{M}\mathbf{d}} \times \Omega \to \mathbb{R}^{\mathfrak{M}\mathbf{d}}$ satisfy for all $\theta \in \mathbb{R}^{\mathfrak{M}\mathbf{d}}$, $\omega \in \Omega$ that $G_m(\theta, \omega) = (\nabla_\theta F_m)(\theta, \omega)$, and let $\Theta : \mathbb{N}_0 \times \Omega \to \mathbb{R}^{\mathfrak{M}\mathbf{d}}$ satisfy for all $m \in \mathbb{N}$ that $\Theta_0 = (W^{0,1}, W^{0,2}, \dots, W^{0,\mathfrak{M}})$ and

$$\Theta_m = \Theta_{m-1} - \gamma_m G_m(\Theta_{m-1}). \tag{23}$$

3 | LRV STRATEGY IN THE CASE OF MONTE CARLO-EULER-MARUYAMA APPROXIMATIONS

In the previous section we derived the LRV strategy for the approximation of parametric expectations involving finite-dimensional random variables. In this section we consider parametric expectations involving standard Brownian motions as random variables (see (24) in Subsection 3.1). To derive an LRV algorithm for these approximation problems we first discretize the Brownian motions in Subsection 3.2 to obtain parametric expectations involving only finite-dimensional random variables. In Subsections 3.4, 3.5, 3.6, and 3.7 the LRV strategy is then subsequently applied to these discretized parametric expectations as in Section 2 above (cf. Subsections 2.3, 2.4, 2.5, and 2.6). In Subsection 3.3 we illustrate a special instance of the parametric expectations in (24) and of associated discretizations in the situation where the parametric expectations in (24) involve solutions of SDEs. Finally, in Subsection 3.8 we summarize the algorithm derived in this section in one single framework.

3.1 | Parametric expectations involving vector valued stochastic processes

Let $\mathfrak{p}, d \in \mathbb{N}$, $T \in (0, \infty)$, let $\mathfrak{P} \subseteq \mathbb{R}^{\mathfrak{p}}$ be measurable, let $u : \mathfrak{P} \to \mathbb{R}$ be a function, let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, let $\mathcal{W} : [0, T] \times \Omega \to \mathbb{R}^d$ be a standard Brownian motion with continuous sample paths, let $\phi : \mathfrak{P} \times C([0, T], \mathbb{R}^d) \to \mathbb{R}$ be measurable, and assume for all $p \in \mathfrak{P}$ that $\mathbb{E}[|\phi(p, \mathcal{W})|] < \infty$ and

$$u(p) = \mathbb{E}[\phi(p, \mathcal{W})]. \tag{24}$$

The goal of this section is to derive an algorithm to approximately compute the function $u : \mathfrak{P} \to \mathbb{R}$ given through the parametric expectation in (24).

3.2 | Temporal discretizations of the Brownian motion

Let $N \in \mathbb{N}$, let $\mathfrak{W} : \Omega \to \mathbb{R}^{Nd}$ satisfy $\mathfrak{W} = \sqrt{N/T}\,(\mathcal{W}_{T/N} - \mathcal{W}_0, \mathcal{W}_{2T/N} - \mathcal{W}_{T/N}, \ldots, \mathcal{W}_T - \mathcal{W}_{(N-1)T/N})$, let $\Phi : \mathfrak{P} \times \mathbb{R}^{Nd} \to \mathbb{R}$ satisfy for all $p \in \mathfrak{P}$ that $\mathbb{R}^{Nd} \ni w \mapsto \Phi(p, w) \in \mathbb{R}$ is continuously differentiable, and assume for all $p \in \mathfrak{P}$ that $\mathbb{E}[|\Phi(p, \mathfrak{W})|] < \infty$. We think of $\Phi : \mathfrak{P} \times \mathbb{R}^{Nd} \to \mathbb{R}$ as a suitable approximation of $\phi : \mathfrak{P} \times C([0, T], \mathbb{R}^d) \to \mathbb{R}$ in the sense that for all $p \in \mathfrak{P}$ it holds that

$$\phi(p, \mathcal{W}) \approx \Phi\Big(p, \sqrt{N/T}\,(\mathcal{W}_{T/N} - \mathcal{W}_0, \mathcal{W}_{2T/N} - \mathcal{W}_{T/N}, \ldots, \mathcal{W}_T - \mathcal{W}_{(N-1)T/N})\Big) = \Phi(p, \mathfrak{W}). \tag{25}$$

3.3 | Euler-Maruyama approximations for parametric stochastic differential equations

In the case where $\mathfrak{p} = 1 + d$, where $B \in (0, \infty)$, where $\mathfrak{P} = [0, T] \times [-B, B]^d$, where $\mu : \mathbb{R}^d \to \mathbb{R}^d$ is globally Lipschitz continuous, where $X = (X_t^{\xi,w})_{(t,\xi,w) \in [0,T] \times \mathbb{R}^d \times C([0,T],\mathbb{R}^d)} : [0, T] \times \mathbb{R}^d \times C([0, T], \mathbb{R}^d) \to \mathbb{R}^d$ satisfies for all $t \in [0, T]$, $\xi \in \mathbb{R}^d$, $w = (w_s)_{s \in [0,T]} \in C([0, T], \mathbb{R}^d)$ that

$$X_t^{\xi,w} = \xi + \int_0^t \mu\left(X_s^{\xi,w}\right) \mathrm{d}s + w_t, \tag{26}$$

where $g : [0, T] \times \mathbb{R}^d \to \mathbb{R}$ satisfies for all $t \in [0, T]$ that $\mathbb{R}^d \ni x \mapsto g(t, x) \in \mathbb{R}$ is continuously differentiable, where it holds for all $p = (t, \xi) \in \mathfrak{P}$, $w \in C([0, T], \mathbb{R}^d)$ that $\phi(p, w) = g(t, X_t^{\xi,w})$, where $\mathcal{X}^{\xi,\theta} = (\mathcal{X}_t^{\xi,\theta})_{t \in [0,T]} : [0, T] \to \mathbb{R}^d$, $\theta \in \mathbb{R}^{Nd}$, $\xi \in \mathbb{R}^d$, satisfy for all $\theta \in \mathbb{R}^{Nd}$, $\xi \in \mathbb{R}^d$, $n \in$

$\{1, 2, \dots N\}$, $t \in [\frac{(n-1)T}{N}, \frac{nT}{N}]$ that $\mathcal{X}_0^{\xi,\theta} = \xi$ and

$$\mathcal{X}_t^{\xi,\theta} = \mathcal{X}_{(n-1)T/N}^{\xi,\theta} + \left(\frac{tN}{T} - n + 1\right)\left(\frac{T}{N}\mu\left(\mathcal{X}_{(n-1)T/N}^{\xi,\theta}\right) + \frac{\sqrt{T}}{\sqrt{N}}(\theta_{(n-1)d+k})_{k\in\{1,2,\dots,d\}}\right), \tag{27}$$

and where it holds for all $p \in \mathfrak{P}$, $\theta \in \mathbb{R}^{Nd}$ that $\Phi(p, \theta) = g(t, \mathcal{X}_t^{\xi,\theta})$ observe that

(i) it holds for all $p = (t, \xi) \in \mathfrak{P}$ that

$$u(p) = u(t, \xi) = \mathbb{E}\left[g\left(t, X_t^{\xi,\mathcal{W}}\right)\right] \tag{28}$$

is the expectation of the test function $\mathbb{R}^d \ni x \mapsto g(t, x) \in \mathbb{R}$ evaluated at time t of the solution process $(X_s^{\xi,\mathcal{W}})_{s\in[0,T]}$ of the additive noise driven SDE in (26) and

(ii) it holds for all $p = (t, \xi) \in \mathfrak{P}$ that $\Phi(p, \mathfrak{W})$ is an approximation

$$\Phi(p, \mathfrak{W}) = g\left(t, \mathcal{X}_t^{\xi,\mathfrak{W}}\right) \approx g\left(t, X_t^{\xi,\mathcal{W}}\right) = \phi(p, \mathcal{W}) \tag{29}$$

of $\phi(p, \mathcal{W})$ based on linearly interpolated Euler-Maruyama approximations $(\mathcal{X}_s^{\xi,\mathfrak{W}})_{s\in[0,T]}$ with N timesteps of the solution $(X_s^{\xi,\mathcal{W}})_{s\in[0,T]}$ of the SDE in (26).

3.4 | Monte Carlo approximations

Let $\mathfrak{M} \in \mathbb{N}$ and let $W^{m,\mathfrak{m}} : \Omega \to \mathbb{R}^{Nd}$, $m, \mathfrak{m} \in \mathbb{N}_0$, be i.i.d. standard normal random vectors. Observe that (24) and (25) suggest that for all $p \in \mathfrak{P}$ it holds that

$$u(p) = \mathbb{E}[\phi(p, \mathcal{W})] \approx \mathbb{E}[\Phi(p, \mathfrak{W})] = \mathbb{E}[\Phi(p, W^{0,1})] \approx \frac{1}{\mathfrak{M}}\left[\sum_{\mathfrak{m}=1}^{\mathfrak{M}} \Phi\big(p, W^{0,\mathfrak{m}}\big)\right]. \tag{30}$$

3.5 | Replacing the random variables in Monte Carlo approximations

Let $\mathcal{N} : \mathfrak{P} \times \mathbb{R}^{\mathfrak{M}Nd} \to \mathbb{R}$ satisfy for all $p \in \mathfrak{P}$, $\theta = (\theta_1, \dots, \theta_{\mathfrak{M}Nd}) \in \mathbb{R}^{\mathfrak{M}Nd}$ that

$$\mathcal{N}(p, \theta) = \frac{1}{\mathfrak{M}}\left[\sum_{\mathfrak{m}=1}^{\mathfrak{M}} \Phi\big(p, (\theta_{(\mathfrak{m}-1)Nd+k})_{k\in\{1,2,\dots,Nd\}}\big)\right]. \tag{31}$$

Note that (30) and (31) suggest that for all $p \in \mathfrak{P}$ it holds that

$$u(p) \approx \frac{1}{\mathfrak{M}}\left[\sum_{\mathfrak{m}=1}^{\mathfrak{M}} \Phi\big(p, W^{0,\mathfrak{m}}\big)\right] = \mathcal{N}\left(p, (W^{0,1}, W^{0,2}, \dots, W^{0,\mathfrak{M}})\right). \tag{32}$$

3.6 | Random loss functions for fixed random variables in Monte Carlo approximations

Let $\mathbf{M} \in \mathbb{N}$, let $P_{m,\mathbf{m}} : \Omega \to \mathfrak{P}$, $m, \mathbf{m} \in \mathbb{N}$, be i.i.d. random variables, for every $m \in \mathbb{N}$ let $F_m : \mathbb{R}^{\mathfrak{M}Nd} \times \Omega \to \mathbb{R}$ satisfy for all $\theta = (\theta_1, \dots, \theta_{\mathfrak{M}Nd}) \in \mathbb{R}^{\mathfrak{M}Nd}$ that

$$\begin{aligned} F_m(\theta) &= \frac{1}{\mathbf{M}}\left[\sum_{\mathbf{m}=1}^{\mathbf{M}} \left|\Phi\left(P_{m,\mathbf{m}}, W^{m,\mathbf{m}}\right) - \mathcal{N}(P_{m,\mathbf{m}}, \theta)\right|^2\right] \\ &= \frac{1}{\mathbf{M}}\left[\sum_{\mathbf{m}=1}^{\mathbf{M}} \left|\Phi\left(P_{m,\mathbf{m}}, W^{m,\mathbf{m}}\right) - \frac{1}{\mathfrak{M}}\left[\sum_{\mathfrak{m}=1}^{\mathfrak{M}} \Phi\big(P_{m,\mathbf{m}}, (\theta_{(\mathfrak{m}-1)Nd+k})_{k\in\{1,2,\dots,Nd\}}\big)\right]\right|^2\right], \end{aligned} \tag{33}$$

and for every $m \in \mathbb{N}$ let $G_m : \mathbb{R}^{\mathfrak{M}Nd} \times \Omega \to \mathbb{R}^{\mathfrak{M}Nd}$ satisfy for all $\theta \in \mathbb{R}^{\mathfrak{M}Nd}$, $\omega \in \Omega$ that

$$G_m(\theta, \omega) = (\nabla_\theta F_m)(\theta, \omega). \tag{34}$$

3.7 | Learning the random variables with stochastic gradient descent

Let $(\gamma_m)_{m\in\mathbb{N}} \subseteq (0,\infty)$ and let $\Theta : \mathbb{N}_0 \times \Omega \to \mathbb{R}^{\mathfrak{M}Nd}$ satisfy for all $m \in \mathbb{N}$ that $\Theta_0 = (W^{0,1}, W^{0,2}, \dots, W^{0,\mathfrak{M}})$ and

$$\Theta_m = \Theta_{m-1} - \gamma_m G_m(\Theta_{m-1}). \tag{35}$$

For every sufficiently large $m \in \mathbb{N}$ we propose to employ the random function $\mathfrak{P} \times \Omega \ni (p, \omega) \mapsto \mathcal{N}(p, \Theta_m(\omega)) \in \mathbb{R}$ as an approximation for the target function $\mathfrak{P} \ni p \mapsto u(p) \in \mathbb{R}$ in (24).

3.8 | Description of the proposed approximation algorithm

Framework 3.1. *Let $\mathfrak{p}, d, N, \mathfrak{M}, \mathbf{M} \in \mathbb{N}$, $(\gamma_m)_{m\in\mathbb{N}} \subseteq (0,\infty)$, let $\mathfrak{P} \subseteq \mathbb{R}^{\mathfrak{p}}$ be measurable, let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, let $P_{m,\mathbf{m}} : \Omega \to \mathfrak{P}$, $m, \mathbf{m} \in \mathbb{N}$, be i.i.d. random variables, let $W^{m,\mathfrak{m}} : [0,T] \times \Omega \to \mathbb{R}^d$, $m, \mathfrak{m} \in \mathbb{N}_0$, be i.i.d. standard Brownian motions, assume that $(P_{m,\mathbf{m}})_{(m,\mathbf{m})\in\mathbb{N}^2}$ and $(W^{m,\mathfrak{m}})_{(m,\mathfrak{m})\in\mathbb{N}^2}$ are independent, let $\Phi : \mathfrak{P} \times \mathbb{R}^{Nd} \to \mathbb{R}$ satisfy for all $p \in \mathfrak{P}$ that $\mathbb{R}^{Nd} \ni w \mapsto \Phi(p, w) \in \mathbb{R}$ is continuously differentiable, for every $m \in \mathbb{N}$ let $F_m : \mathbb{R}^{\mathfrak{M}Nd} \times \Omega \to \mathbb{R}$ satisfy for all $\theta = (\theta_1, \dots, \theta_{\mathfrak{M}Nd}) \in \mathbb{R}^{\mathfrak{M}Nd}$ that*

$$\begin{aligned} F_m(\theta) = \frac{1}{\mathbf{M}}\Bigg[\sum_{\mathbf{m}=1}^{\mathbf{M}} \Bigg|&\Phi\Big(P_{m,\mathbf{m}}, \sqrt{N/T}\left(W^{m,\mathbf{m}}_{T/N} - W^{m,\mathbf{m}}_0, W^{m,\mathbf{m}}_{2T/N} - W^{m,\mathbf{m}}_{T/N}, \dots, W^{m,\mathbf{m}}_T - W^{m,\mathbf{m}}_{(N-1)T/N}\right)\Big) \\ &- \frac{1}{\mathfrak{M}}\left[\sum_{\mathfrak{m}=1}^{\mathfrak{M}} \Phi\big(P_{m,\mathbf{m}}, (\theta_{(\mathfrak{m}-1)Nd+k})_{k\in\{1,2,\dots,Nd\}}\big)\right]\Bigg|^2\Bigg], \end{aligned} \tag{36}$$

for every $m \in \mathbb{N}$ let $G_m : \mathbb{R}^{\mathfrak{M}Nd} \times \Omega \to \mathbb{R}^{\mathfrak{M}Nd}$ satisfy for all $\theta \in \mathbb{R}^{\mathfrak{M}Nd}$, $\omega \in \Omega$ that $G_m(\theta, \omega) = (\nabla_\theta F_m)(\theta, \omega)$, and let $\Theta = (\Theta^{(1)}, \dots, \Theta^{(\mathfrak{M}Nd)}) : \mathbb{N}_0 \times \Omega \to \mathbb{R}^{\mathfrak{M}Nd}$ satisfy for all $m \in \mathbb{N}$, $k \in \{1, 2, \dots, \mathfrak{M}\}$, $n \in \{1, 2, \dots, N\}$ that $(\Theta_0^{([(k-1)N+n-1]d+1)}, \Theta_0^{([(k-1)N+n-1]d+2)}, \dots, \Theta_0^{([(k-1)N+n-1]d+d)}) = \sqrt{N/T}(W^{0,k}_{nT/N} - W^{0,k}_{(n-1)T/N})$ and $\Theta_m = \Theta_{m-1} - \gamma_m G_m(\Theta_{m-1})$.

4 | LRV STRATEGY IN THE CASE OF MULTILEVEL MONTE CARLO APPROXIMATIONS

In this section we consider the problem of approximating a parametric expectation involving a general measure space-valued random variable for which a sequence of finite-dimensional approximations is available (see Subsections 4.1 and 4.2). We illustrate such a situation in Subsection 4.4 in which the original random variable is a Brownian motion driving an SDE and the finite-dimensional approximations consist of Euler discretizations of the SDE with decreasing step sizes. The employed proposal algorithm for the LRV strategy in this section is a form of MLMC method (cf. Heinrich, 1998 and Giles, 2008 and cf., e.g., Creutzig et al., 2009; Giles, 2015; Heinrich, 2001; Heinrich & Sindambiwe, 1999) as described in Subsection 4.3. Based on this proposal algorithm, the LRV strategy is applied to the considered approximation problem in Subsections 4.5, 4.6, and 4.7 and, thereafter, summarized in one single framework in Subsection 4.8.

4.1 | Parametric expectations involving measure space valued random variables

Let $\mathfrak{p} \in \mathbb{N}$, let $\mathfrak{P} \subseteq \mathbb{R}^{\mathfrak{p}}$ be measurable, let $(V, \mathcal{V})$ be a measurable space, let $\phi : \mathfrak{P} \times V \to \mathbb{R}$ be continuous, let $u : \mathfrak{P} \to \mathbb{R}$ be a function, let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, let $\mathcal{W} : \Omega \to V$ be a random variable, and assume for all $p \in \mathfrak{P}$ that $\mathbb{E}[|\phi(p, \mathcal{W})|] < \infty$ and

$$u(p) = \mathbb{E}[\phi(p, \mathcal{W})]. \tag{37}$$

The goal of this section is to derive an algorithm to approximately compute the function $u : \mathfrak{P} \to \mathbb{R}$ given through the parametric expectation in (37).

4.2 | Finite-dimensional approximations

Let $\mathbf{d} = (\mathbf{d}_l)_{l \in \mathbb{N}_0} : \mathbb{N}_0 \to \mathbb{N}$ be non-decreasing, let $\mathcal{P}_l : \mathbb{R}^{\mathbf{d}_l} \to \mathbb{R}^{\mathbf{d}_{l-1}}$, $l \in \mathbb{N}$, be continuously differentiable, let $\mathfrak{W}_l : \Omega \to \mathbb{R}^{\mathbf{d}_l}$, $l \in \mathbb{N}_0$, be random variables which satisfy for all $l \in \mathbb{N}$ that $\mathcal{P}_l(\mathfrak{W}_l) = \mathfrak{W}_{l-1}$, let $\Phi_l : \mathfrak{P} \times \mathbb{R}^{\mathbf{d}_l} \to \mathbb{R}$, $l \in \mathbb{N}_0$, satisfy for all $l \in \mathbb{N}_0$, $p \in \mathfrak{P}$ that $\mathbb{R}^{\mathbf{d}_l} \ni w \mapsto \Phi_l(p, w) \in \mathbb{R}$ is continuously differentiable, and assume for all $p \in \mathfrak{P}$, $l \in \mathbb{N}_0$ that $\mathbb{E}[|\Phi_l(p, \mathfrak{W}_l)|] < \infty$. We think of $\mathfrak{W}_l$, $l \in \mathbb{N}_0$, and Φ_l, $l \in \mathbb{N}_0$, as suitable approximations of $\mathcal{W}$ and ϕ in the sense that for all $p \in \mathfrak{P}$ and all sufficiently large $l \in \mathbb{N}_0$ it holds that

$$\phi(p, \mathcal{W}) \approx \Phi_l(p, \mathfrak{W}_l). \tag{38}$$

4.3 | Multilevel Monte Carlo approximations

Let $L \in \mathbb{N}_0$, $\mathfrak{M}_0, \mathfrak{M}_1, \dots, \mathfrak{M}_L \in \mathbb{N}$ satisfy $\mathfrak{M}_0 \geq \mathfrak{M}_1 \geq \dots \geq \mathfrak{M}_L$, let $W_l^{m,\mathfrak{m}} : \Omega \to \mathbb{R}^{\mathbf{d}_l}$, $l, m, \mathfrak{m} \in \mathbb{N}_0$, be independent random variables, and assume for all $l, m, \mathfrak{m} \in \mathbb{N}_0$, $A \in \mathcal{B}(\mathbb{R}^{\mathbf{d}_l})$ that $\mathbb{P}(W_l^{m,\mathfrak{m}} \in A) = \mathbb{P}(\mathfrak{W}_l \in A)$. Observe that (37) and (38) suggest that for all $p \in \mathfrak{P}$ it holds that

$$\begin{aligned} u(p) = \mathbb{E}[\phi(p, \mathcal{W})] \approx \mathbb{E}[\Phi_L(p, \mathfrak{W}_L)] &= \mathbb{E}[\Phi_0(p, \mathfrak{W}_0)] + \sum_{l=1}^{L} \mathbb{E}[\Phi_l(p, \mathfrak{W}_l) - \Phi_{l-1}(p, \mathcal{P}_l(\mathfrak{W}_l))] \\ &\approx \frac{1}{\mathfrak{M}_0}\left[\sum_{m=1}^{\mathfrak{M}_0} \Phi_0\big(p, W_0^{0,m}\big)\right] + \sum_{l=1}^{L} \frac{1}{\mathfrak{M}_l}\left[\sum_{m=1}^{\mathfrak{M}_l} \Phi_l\big(p, W_l^{0,m}\big) - \Phi_{l-1}\big(p, \mathcal{P}_l\big(W_l^{0,m}\big)\big)\right]. \end{aligned} \tag{39}$$

4.4 | Multilevel Monte Carlo approximations for parametric stochastic differential equations

In the case where $d \in \mathbb{N}$, $T, B \in (0, \infty)$, where $\mathfrak{p} = 1 + d$, $\mathfrak{P} = [0, T] \times [-B, B]^d$, and $(V, \mathcal{V}) = (C([0, T], \mathbb{R}^d), \mathcal{B}(C([0, T], \mathbb{R}^d)))$, where $\mu : \mathbb{R}^d \to \mathbb{R}^d$ is globally Lipschitz continuous, where $X = (X_t^{\xi,w})_{(t,\xi,w) \in [0,T] \times \mathbb{R}^d \times C([0,T],\mathbb{R}^d)} : [0, T] \times \mathbb{R}^d \times C([0, T], \mathbb{R}^d) \to \mathbb{R}^d$ satisfies for all $t \in [0, T]$, $\xi \in \mathbb{R}^d$, $w = (w_s)_{s \in [0,T]} \in C([0, T], \mathbb{R}^d)$ that

$$X_t^{\xi,w} = \xi + \int_0^t \mu\Big(X_s^{\xi,w}\Big)\,\mathrm{d}s + w_t, \tag{40}$$

where $g : [0, T] \times \mathbb{R}^d \to \mathbb{R}$ satisfies for all $t \in [0, T]$ that $\mathbb{R}^d \ni x \mapsto g(t, x) \in \mathbb{R}$ is continuously differentiable, where it holds for all $p = (t, \xi) \in \mathfrak{P}$, $w \in C([0, T], \mathbb{R}^d)$ that $\phi(p, w) = g(t, X_t^{\xi,w})$, where $\mathcal{W} : \Omega \to C([0, T], \mathbb{R}^d)$ is a standard Brownian motion, where for all $l \in \mathbb{N}_0$ it holds that $\mathbf{d}_l = 2^l d$, where for all $l \in \mathbb{N}_0$ it holds that

$$\mathfrak{W}_l = \sqrt{2^l/T}\,\big(\mathcal{W}_{T/2^l} - \mathcal{W}_0, \mathcal{W}_{2T/2^l} - \mathcal{W}_{T/2^l}, \dots, \mathcal{W}_T - \mathcal{W}_{(2^l-1)T/2^l}\big), \tag{41}$$

where $\mathcal{X}^{\xi,\theta,N} = (\mathcal{X}_t^{\xi,\theta,N})_{t \in [0,T]} : [0, T] \to \mathbb{R}^d$, $N \in \mathbb{N}$, $\theta \in \mathbb{R}^{Nd}$, $\xi \in \mathbb{R}^d$, satisfy for all $N \in \mathbb{N}$, $\theta \in \mathbb{R}^{Nd}$, $\xi \in \mathbb{R}^d$, $n \in \{1, 2, \dots N\}$, $t \in [\frac{(n-1)T}{N}, \frac{nT}{N}]$ that $\mathcal{X}_0^{\xi,\theta,N} = \xi$ and

$$\mathcal{X}_t^{\xi,\theta,N} = \mathcal{X}_{(n-1)T/N}^{\xi,\theta,N} + \left(\frac{tN}{T} - n + 1\right)\left(\frac{T}{N}\mu\left(\mathcal{X}_{(n-1)T/N}^{\xi,\theta,N}\right) + \frac{\sqrt{T}}{\sqrt{N}}(\theta_{(n-1)d+k})_{k \in \{1,2,\dots,d\}}\right), \tag{42}$$

where for all $l \in \mathbb{N}_0$, $p = (t, \xi) \in \mathfrak{P}$, $\theta \in \mathbb{R}^{2^l d}$ it holds that $\Phi_l(p, \theta) = g(t, \mathcal{X}_t^{\xi,\theta,2^l})$, and where for all $l \in \{0, 1, \dots, L\}$ it holds that $\mathfrak{M}_l = 2^{L-l}$ observe that

(i) it holds for all $l \in \mathbb{N}$, $\theta_1, \theta_2 \dots, \theta_{2^l}, \vartheta_1, \vartheta_2 \dots, \vartheta_{2^{l-1}} \in \mathbb{R}^d$ with $\forall\, n \in \{1, 2, \dots, 2^{l-1}\} : \vartheta_n = \frac{1}{\sqrt{2}}(\theta_{2n} + \theta_{2n-1})$ that

$$\mathcal{P}_l(\theta_1, \theta_2 \dots, \theta_{2^l}) = (\vartheta_1, \vartheta_2 \dots, \vartheta_{2^{l-1}}), \tag{43}$$

(ii) it holds for all $p = (t, \xi) \in \mathfrak{P}$ that

$$u(p) = u(t, \xi) = \mathbb{E}\left[g\left(t, X_t^{\xi,\mathcal{W}}\right)\right] \tag{44}$$

is the expectation of the test function $\mathbb{R}^d \ni x \mapsto g(t, x) \in \mathbb{R}$ evaluated at time t of the solution process $(X_s^{\xi,\mathcal{W}})_{s\in[0,T]}$ of the additive noise driven SDE in (41),

(iii) it holds for all $l \in \mathbb{N}_0$, $p = (t, \xi) \in \mathfrak{P}$ that $\Phi_l(p, \mathfrak{W}_l)$ is an approximation

$$\Phi_l(p, \mathfrak{W}_l) = g\left(t, \mathcal{X}_t^{\xi,\mathfrak{W}_l,2^l}\right) \approx g\left(t, X_t^{\xi,\mathcal{W}}\right) = \phi(p, \mathcal{W}) \tag{45}$$

of $\phi(p, \mathcal{W})$ based on linearly interpolated Euler-Maruyama approximations $(\mathcal{X}_s^{\xi,\mathfrak{W}_l,2^l})_{s\in[0,T]}$ with 2^l timesteps of the solution $(X_s^{\xi,\mathcal{W}})_{s\in[0,T]}$ of the SDE in (41), and

(iv) it holds for all $p = (t, \xi) \in \mathfrak{P}$ that

$$\begin{aligned}
&\frac{1}{\mathfrak{M}_0}\left[\sum_{m=1}^{\mathfrak{M}_0} \Phi_0\left(p, W_0^{0,m}\right)\right] + \sum_{l=1}^{L} \frac{1}{\mathfrak{M}_l}\left[\sum_{m=1}^{\mathfrak{M}_l} \Phi_l\left(p, W_l^{0,m}\right) - \Phi_{l-1}\left(p, \mathcal{P}_l\left(W_l^{0,m}\right)\right)\right] \\
&= \frac{1}{2^L}\left[\sum_{m=1}^{2^L} g\left(t, \mathcal{X}_t^{\xi,W_0^{0,m},1}\right)\right] + \sum_{l=1}^{L} \frac{1}{2^{L-l}}\left[\sum_{m=1}^{2^{L-l}} g\left(t, \mathcal{X}_t^{\xi,W_l^{0,m},2^l}\right) - g\left(t, \mathcal{X}_t^{\xi,\mathcal{P}_l\left(W_l^{0,m}\right),2^{l-1}}\right)\right]
\end{aligned} \tag{46}$$

is a MLMC approximation of $u(t, \xi)$ as proposed in Giles (2008).

4.5 | Replacing the random variables in multilevel Monte Carlo approximations

Let $\mathcal{N} : \mathfrak{P} \times \mathbb{R}^{\mathfrak{M}_0\mathbf{d}_0+\mathfrak{M}_1\mathbf{d}_1+\cdots+\mathfrak{M}_L\mathbf{d}_L} \to \mathbb{R}$ satisfy for all $p \in \mathfrak{P}$, $\theta = (\theta_1, \ldots, \theta_{\mathfrak{M}_0\mathbf{d}_0+\mathfrak{M}_1\mathbf{d}_1+\cdots+\mathfrak{M}_L\mathbf{d}_L}) \in \mathbb{R}^{\mathfrak{M}_0\mathbf{d}_0+\mathfrak{M}_1\mathbf{d}_1+\cdots+\mathfrak{M}_L\mathbf{d}_L}$ that

$$\begin{aligned}
\mathcal{N}(p, \theta) = {} & \frac{1}{\mathfrak{M}_0}\left[\sum_{m=1}^{\mathfrak{M}_0} \Phi_0\left(p, (\theta_{(m-1)\mathbf{d}_0+k})_{k\in\{1,2,\ldots,\mathbf{d}_0\}}\right)\right] \\
& + \sum_{l=1}^{L} \frac{1}{\mathfrak{M}_l}\left[\sum_{m=1}^{\mathfrak{M}_l} \Phi_l\left(p, (\theta_{\mathfrak{M}_0\mathbf{d}_0+\mathfrak{M}_1\mathbf{d}_1+\cdots+\mathfrak{M}_{l-1}\mathbf{d}_{l-1}+(m-1)\mathbf{d}_l+k})_{k\in\{1,2,\ldots,\mathbf{d}_l\}}\right)\right. \\
& \left. - \Phi_{l-1}\left(p, \mathcal{P}_l((\theta_{\mathfrak{M}_0\mathbf{d}_0+\mathfrak{M}_1\mathbf{d}_1+\cdots+\mathfrak{M}_{l-1}\mathbf{d}_{l-1}+(m-1)\mathbf{d}_l+k})_{k\in\{1,2,\ldots,\mathbf{d}_l\}})\right)\right].
\end{aligned} \tag{47}$$

Note that (39) and (47) suggest that for all $p \in \mathfrak{P}$ it holds that

$$\begin{aligned}
u(p) &\approx \frac{1}{\mathfrak{M}_0}\left[\sum_{m=1}^{\mathfrak{M}_0} \Phi_0\left(p, W_0^{0,m}\right)\right] + \sum_{l=1}^{L} \frac{1}{\mathfrak{M}_l}\left[\sum_{m=1}^{\mathfrak{M}_l} \Phi_l\left(p, W_l^{0,m}\right) - \Phi_{l-1}\left(p, \mathcal{P}_l\left(W_l^{0,m}\right)\right)\right] \\
&= \mathcal{N}\left(p, \left(W_0^{0,1}, W_0^{0,2}, \ldots, W_0^{0,\mathfrak{M}_1}, W_1^{0,1}, W_1^{0,2}, \ldots, W_1^{0,\mathfrak{M}_2}, \ldots, W_L^{0,1}, W_L^{0,2}, \ldots, W_L^{0,\mathfrak{M}_L}\right)\right).
\end{aligned} \tag{48}$$

4.6 | Random loss functions for fixed random variables in multilevel Monte Carlo approximations

Let $\mathbf{M} \in \mathbb{N}$, $\mathcal{L} \in \mathbb{N}_0$, let $P_{m,\mathbf{m}} : \Omega \to \mathfrak{P}$, $m, \mathbf{m} \in \mathbb{N}$, be i.i.d. random variables, assume that $(P_{m,\mathbf{m}})_{(m,\mathbf{m})\in\mathbb{N}^2}$ and $(W_l^{m,\mathbf{m}})_{(l,m,\mathbf{m})\in(\mathbb{N}_0)^3}$ are independent, for every $m \in \mathbb{N}$ let $F_m : \mathbb{R}^{\mathfrak{M}_0\mathbf{d}_0+\mathfrak{M}_1\mathbf{d}_1+\cdots+\mathfrak{M}_L\mathbf{d}_L} \times \Omega \to \mathbb{R}$ satisfy for all $\theta = (\theta_1, \dots, \theta_{\mathfrak{M}_0\mathbf{d}_0+\mathfrak{M}_1\mathbf{d}_1+\cdots+\mathfrak{M}_L\mathbf{d}_L}) \in \mathbb{R}^{\mathfrak{M}_0\mathbf{d}_0+\mathfrak{M}_1\mathbf{d}_1+\cdots+\mathfrak{M}_L\mathbf{d}_L}$ that

$$F_m(\theta) = \frac{1}{\mathbf{M}} \left[\sum_{\mathbf{m}=1}^{\mathbf{M}} \left| \Phi_{\mathcal{L}}\big(P_{m,\mathbf{m}}, W_{\mathcal{L}}^{m,\mathbf{m}}\big) - \mathcal{N}(P_{m,\mathbf{m}}, \theta) \right|^2 \right], \tag{49}$$

and for every $m \in \mathbb{N}$ let $G_m : \mathbb{R}^{\mathfrak{M}_0\mathbf{d}_0+\mathfrak{M}_1\mathbf{d}_1+\cdots+\mathfrak{M}_L\mathbf{d}_L} \times \Omega \to \mathbb{R}^{\mathfrak{M}_0\mathbf{d}_0+\mathfrak{M}_1\mathbf{d}_1+\cdots+\mathfrak{M}_L\mathbf{d}_L}$ satisfy for all $\theta \in \mathbb{R}^{\mathfrak{M}_0\mathbf{d}_0+\mathfrak{M}_1\mathbf{d}_1+\cdots+\mathfrak{M}_L\mathbf{d}_L}$, $\omega \in \Omega$ that

$$G_m(\theta, \omega) = (\nabla_\theta F_m)(\theta, \omega). \tag{50}$$

4.7 | Learning the random variables with stochastic gradient descent

Let $(\gamma_m)_{m\in\mathbb{N}} \subseteq (0, \infty)$ and let $\Theta : \mathbb{N}_0 \times \Omega \to \mathbb{R}^{\mathfrak{M}_0\mathbf{d}_0+\mathfrak{M}_1\mathbf{d}_1+\cdots+\mathfrak{M}_L\mathbf{d}_L}$ satisfy for all $m \in \mathbb{N}$ that $\Theta_0 = (W_0^{0,1}, W_0^{0,2}, \dots, W_0^{0,\mathfrak{M}_1}, W_1^{0,1}, W_1^{0,2}, \dots, W_1^{0,\mathfrak{M}_2}, \dots, W_L^{0,1}, W_L^{0,2}, \dots, W_L^{0,\mathfrak{M}_L})$ and

$$\Theta_m = \Theta_{m-1} - \gamma_m G_m(\Theta_{m-1}). \tag{51}$$

For every sufficiently large $m \in \mathbb{N}$ we propose to employ the random function $\mathfrak{P} \times \Omega \ni (p, \omega) \mapsto \mathcal{N}(p, \Theta_m(\omega)) \in \mathbb{R}$ as an approximation for the target function $\mathfrak{P} \ni p \mapsto u(p) \in \mathbb{R}$ in (37).

4.8 | Description of the proposed approximation algorithm

Framework 4.1. *Let $\mathfrak{p}, \mathbf{M} \in \mathbb{N}$, $L, \mathcal{L} \in \mathbb{N}_0$, $\mathfrak{M}_0, \mathfrak{M}_1, \dots, \mathfrak{M}_L \in \mathbb{N}$, $(\mathbf{d}_l)_{l\in\mathbb{N}_0} \subseteq \mathbb{N}$, $(\gamma_m)_{m\in\mathbb{N}} \subseteq (0, \infty)$, let $\mathfrak{P} \subseteq \mathbb{R}^{\mathfrak{p}}$ be measurable, let $\Phi_l : \mathfrak{P} \times \mathbb{R}^{\mathbf{d}_l} \to \mathbb{R}$, $l \in \mathbb{N}_0$, satisfy for all $l \in \mathbb{N}_0$, $p \in \mathfrak{P}$ that $\mathbb{R}^{\mathbf{d}_l} \ni w \mapsto \Phi_l(p, w) \in \mathbb{R}$ is continuously differentiable, let $\mathcal{P}_l : \mathbb{R}^{\mathbf{d}_l} \to \mathbb{R}^{\mathbf{d}_{l-1}}$, $l \in \{1, 2, \dots, L\}$, be continuously differentiable, let $\mathcal{N} : \mathfrak{P} \times \mathbb{R}^{\mathfrak{M}_0\mathbf{d}_0+\mathfrak{M}_1\mathbf{d}_1+\cdots+\mathfrak{M}_L\mathbf{d}_L} \to \mathbb{R}$ satisfy for all $p \in \mathfrak{P}$, $\theta = (\theta_1, \dots, \theta_{\mathfrak{M}_0\mathbf{d}_0+\mathfrak{M}_1\mathbf{d}_1+\cdots+\mathfrak{M}_L\mathbf{d}_L}) \in \mathbb{R}^{\mathfrak{M}_0\mathbf{d}_0+\mathfrak{M}_1\mathbf{d}_1+\cdots+\mathfrak{M}_L\mathbf{d}_L}$ that*

$$\begin{aligned}\mathcal{N}(p, \theta) = {} & \frac{1}{\mathfrak{M}_0} \left[\sum_{m=1}^{\mathfrak{M}_0} \Phi_0\big(p, (\theta_{(m-1)\mathbf{d}_0+k})_{k\in\{1,2,\dots,\mathbf{d}_0\}}\big) \right] \\ & + \sum_{l=1}^{L} \frac{1}{\mathfrak{M}_l} \left[\sum_{m=1}^{\mathfrak{M}_l} \Phi_l\big(p, (\theta_{\mathfrak{M}_0\mathbf{d}_0+\mathfrak{M}_1\mathbf{d}_1+\cdots+\mathfrak{M}_{l-1}\mathbf{d}_{l-1}+(m-1)\mathbf{d}_l+k})_{k\in\{1,2,\dots,\mathbf{d}_l\}}\big) \right. \\ & \left. - \Phi_{l-1}\big(p, \mathcal{P}_l((\theta_{\mathfrak{M}_0\mathbf{d}_0+\mathfrak{M}_1\mathbf{d}_1+\cdots+\mathfrak{M}_{l-1}\mathbf{d}_{l-1}+(m-1)\mathbf{d}_l+k})_{k\in\{1,2,\dots,\mathbf{d}_l\}})\big) \right],\end{aligned} \tag{52}$$

let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, let $W_l^{m,\mathfrak{m}} : \Omega \to \mathbb{R}^d$, $l, m, \mathfrak{m} \in \mathbb{N}_0$, be independent random variables which satisfy for all $l, m, \mathfrak{m} \in \mathbb{N}_0$, $A \in \mathcal{B}(\mathbb{R}^{\mathbf{d}_l})$ that $\mathbb{P}(W_l^{m,\mathfrak{m}} \in A) = \mathbb{P}(W_l^{0,0} \in A)$, let $P_{m,\mathbf{m}} : \Omega \to \mathfrak{P}$, $m, \mathbf{m} \in \mathbb{N}$, be i.i.d. random variables, assume that $(P_{m,\mathbf{m}})_{(m,\mathbf{m}) \in \mathbb{N}^2}$ and $(W_l^{m,\mathfrak{m}})_{(l,m,\mathfrak{m}) \in (\mathbb{N}_0)^3}$ are independent, for every $m \in \mathbb{N}$ let $F_m : \mathbb{R}^{\mathfrak{M}_0 \mathbf{d}_0 + \mathfrak{M}_1 \mathbf{d}_1 + \cdots + \mathfrak{M}_L \mathbf{d}_L} \times \Omega \to \mathbb{R}$ satisfy for all $\theta \in \mathbb{R}^{\mathfrak{M}_0 \mathbf{d}_0 + \mathfrak{M}_1 \mathbf{d}_1 + \cdots + \mathfrak{M}_L \mathbf{d}_L}$ that

$$F_m(\theta) = \frac{1}{\mathbf{M}} \left[\sum_{\mathbf{m}=1}^{\mathbf{M}} \left| \Phi_{\mathcal{L}}\left(P_{m,\mathbf{m}}, W_{\mathcal{L}}^{m,\mathbf{m}}\right) - \mathcal{N}(P_{m,\mathbf{m}}, \theta) \right|^2 \right], \tag{53}$$

for every $m \in \mathbb{N}$ let $G_m : \mathbb{R}^{\mathfrak{M}_0 \mathbf{d}_0 + \mathfrak{M}_1 \mathbf{d}_1 + \cdots + \mathfrak{M}_L \mathbf{d}_L} \times \Omega \to \mathbb{R}^{\mathfrak{M}_0 \mathbf{d}_0 + \mathfrak{M}_1 \mathbf{d}_1 + \cdots + \mathfrak{M}_L \mathbf{d}_L}$ satisfy for all $\theta \in \mathbb{R}^{\mathfrak{M}_0 \mathbf{d}_0 + \mathfrak{M}_1 \mathbf{d}_1 + \cdots + \mathfrak{M}_L \mathbf{d}_L}$, $\omega \in \Omega$ that $G_m(\theta, \omega) = (\nabla_\theta F_m)(\theta, \omega)$, and let $\Theta : \mathbb{N}_0 \times \Omega \to \mathbb{R}^{\mathfrak{M}_0 \mathbf{d}_0 + \mathfrak{M}_1 \mathbf{d}_1 + \cdots + \mathfrak{M}_L \mathbf{d}_L}$ satisfy for all $m \in \mathbb{N}$ that $\Theta_0 = (W_0^{0,1}, W_0^{0,2}, \ldots, W_0^{0,\mathfrak{M}_1}, W_1^{0,1}, W_1^{0,2}, \ldots, W_1^{0,\mathfrak{M}_2}, \ldots, W_L^{0,1}, W_L^{0,2}, \ldots, W_L^{0,\mathfrak{M}_L})$ and

$$\Theta_m = \Theta_{m-1} - \gamma_m G_m(\Theta_{m-1}). \tag{54}$$

5 | LRV STRATEGY IN THE CASE OF MULTILEVEL PICARD APPROXIMATIONS

In this section we consider the case where the target function to be approximated with the LRV strategy is given as the solution of a suitable stochastic fixed point equation (SFPE); see (55) in Subsection 5.1. The setup in Subsection 5.1 includes, as important special cases, the solutions of several semilinear parabolic PDEs. Two examples of such semilinear parabolic PDEs (heat PDEs with Lipschitz nonlinearities and Black-Scholes PDEs with Lipschitz nonlinearities) are presented in Subsection 5.2. As proposal algorithms for the LRV strategy we present in Subsection 5.3 a slight generalization of the MLP algorithm for semilinear PDEs in Hutzenthaler, Jentzen, Kruse et al. (2020) (cf. also E et al., 2019, 2021). In analogy to the previous sections, in Subsections 5.4, 5.5, and 5.6 an algorithm for the considered approximation problem is derived based on the LRV strategy with this MLP algorithm as proposal algorithm. Finally, the problem and the algorithm are summarized in one single framework in Subsection 5.7.

5.1 | Parametric solutions of stochastic fixed point equations

Let $\mathfrak{p}, \mathbf{d} \in \mathbb{N}$, let $\mathfrak{P} \subseteq \mathbb{R}^{\mathfrak{p}}$ be measurable, let $\phi : \mathfrak{P} \times \mathbb{R}^{\mathbf{d}} \times \mathbb{R} \to \mathbb{R}$, $\mathcal{P} : \mathfrak{P} \times \mathbb{R}^{\mathbf{d}} \to \mathfrak{P}$, and $u : \mathfrak{P} \to \mathbb{R}$ be measurable, let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, let $\mathcal{W} : \Omega \to \mathbb{R}^{\mathbf{d}}$ be a random variable, and assume for all $p \in \mathfrak{P}$ that $\mathbb{E}[|\phi(p, \mathcal{W}, u(\mathcal{P}(p, \mathcal{W})))|] < \infty$ and

$$u(p) = \mathbb{E}[\phi(p, \mathcal{W}, u(\mathcal{P}(p, \mathcal{W})))]. \tag{55}$$

The goal of this section is to derive an algorithm to approximately compute the function $u : \mathfrak{P} \to \mathbb{R}$ given through the SFPE in (55).

5.2 | Approximations for parametric semilinear partial differential equations

5.2.1 | Heat partial differential equations with Lipschitz nonlinearities

In the case where $T \in (0,\infty)$, $d \in \mathbb{N}$, where $\mathfrak{p} = \mathbf{d} = 1 + d$ and $\mathfrak{P} = [0,T] \times \mathbb{R}^d$, where $u \in C([0,T] \times \mathbb{R}^d, \mathbb{R})$ is at most polynomially growing, where $f, g \in C(\mathbb{R}^d, \mathbb{R})$ are at most polynomially growing, where for all $p = (t,x) \in \mathfrak{P}$, $w = (w_1, w_2) \in \mathbb{R}^d \times [0,1]$, $v \in \mathbb{R}$ it holds that $\phi(p,w,v) = g(x + (T-t)^{1/2} w_1) + (T-t) f(v)$ and $\mathcal{P}(p,w) = (t + w_2(T-t), x + (w_2(T-t))^{1/2} w_1)$, where $N : \Omega \to \mathbb{R}^d$ is a standard normal random vector, where $R : \Omega \to [0,1]$ is a continuous uniformly distributed random variable on [0,1] (is a $\mathcal{U}_{[0,1]}$-distributed random variable), where N and R are independent, and where $\mathcal{W} = (N, R)$ observe that

(i) it holds for all $p = (t,x) \in \mathfrak{P}$ that

$$u(p) = u(t,x) = \mathbb{E}\big[g(x + (T-t)^{1/2}N) + (T-t) f\big(u(t + R(T-t), x + (R(T-t))^{1/2}N)\big)\big]$$
$$= \mathbb{E}\left[g(x + (T-t)^{1/2}N) + \int_t^T f\big(u(s, x + (s-t)^{1/2}N)\big)\, \mathrm{d}s\right] \tag{56}$$

and

(ii) it holds that u is a viscosity solution of

$$\left(\tfrac{\partial u}{\partial t}\right)(t,x) + \tfrac{1}{2}(\Delta_x u)(t,x) + f(u(t,x)) = 0 \tag{57}$$

with $u(T,x) = g(x)$ for $(t,x) \in (0,T) \times \mathbb{R}^d$ (cf. Beck, Hutzenthaler, Jentzen et al., 2021, Theorem 1.1).

5.2.2 | Black-Scholes partial differential equations with Lipschitz nonlinearities

In the case where $T \in (0,\infty)$, $d \in \mathbb{N}$, where $\mathfrak{p} = \mathbf{d} = 1 + 3d$ and $\mathfrak{P} = [0,T] \times \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}^d$, where for all $\alpha, \beta \in \mathbb{R}^d$ it holds that $[0,T] \times \mathbb{R}^d \ni (t,x) \mapsto u(t,x,\alpha,\beta) \in \mathbb{R}$ is continuous and at most polynomially growing, where $X = (X^{(i)})_{i \in \{1,2,\ldots,d\}} : [0,T] \times \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ satisfies for all $i \in \{1,2,\ldots,d\} t \in [0,T]$, $x = (x_1,\ldots,x_d), \alpha = (\alpha_1,\ldots,\alpha_d), \beta = (\beta_1,\ldots,\beta_d), n = (n_1,\ldots,n_d) \in \mathbb{R}^d$ that

$$X^{(i)}(t,x,\alpha,\beta,n) = x_i \exp\left(\left(\alpha_i - \tfrac{|\beta_i|^2}{2}\right)t + t^{1/2}\beta_i n_i\right), \tag{58}$$

where $f, g \in C(\mathbb{R}^d, \mathbb{R})$ are at most polynomially growing, where for all $p = (t, x, \alpha, \beta) \in \mathfrak{P}$, $w = (w_1, w_2) \in \mathbb{R}^d \times [0, 1]$, $v \in \mathbb{R}$ it holds that $\phi(p, w, v) = g(X(T - t, x, \alpha, \beta, w_1)) + (T - t)f(v)$ and $\mathcal{P}(p, w) = (t + w_2(T - t), X(w_2(T - t), x, \alpha, \beta, w_1))$, where $N : \Omega \to \mathbb{R}^d$ is a standard normal random vector, where $R : \Omega \to [0, 1]$ is an $\mathcal{U}_{[0,1]}$-distributed random variable, where N and R are independent, and where $\mathcal{W} = (N, R)$ observe that

(i) it holds for all $p = (t, x, \alpha, \beta) \in \mathfrak{P}$ that

$$\begin{aligned} u(p) &= u(t, x, \alpha, \beta) \\ &= \mathbb{E}[g(X(T - t, x, \alpha, \beta, N)) + (T - t)f(u(t + R(T - t), X(R(T - t), x, \alpha, \beta, N)))] \\ &= \mathbb{E}\left[g(X(T - t, x, \alpha, \beta, N)) + \int_t^T f(u(s, X(s - t, x, \alpha, \beta, N)))\,\mathrm{d}s\right] \end{aligned} \tag{59}$$

and

(ii) it holds for all $\alpha = (\alpha_1, \dots, \alpha_d), \beta = (\beta_1, \dots, \beta_d) \in \mathbb{R}^d$ that $[0, T] \times \mathbb{R}^d \ni (t, x) \mapsto u(t, x, \alpha, \beta) \in \mathbb{R}$ is a viscosity solution of

$$\begin{aligned} \left(\tfrac{\partial u}{\partial t}\right)(t, x, \alpha, \beta) + \left[\sum_{i=1}^d \tfrac{|\beta_i|^2 |x_i|^2}{2} \left(\tfrac{\partial^2 u}{\partial (x_i)^2}\right)(t, x, \alpha, \beta)\right] \\ + \left[\sum_{i=1}^d \alpha_i x_i \left(\tfrac{\partial u}{\partial x_i}\right)(t, x, \alpha, \beta)\right] + f(u(t, x, \alpha, \beta)) = 0 \end{aligned} \tag{60}$$

with $u(T, x) = g(x)$ for $(t, x) = (t, x_1, x_2, \dots, x_d) \in (0, T) \times \mathbb{R}^d$ (cf., e.g., Beck, Hutzenthaler, Jentzen et al., 2021, Theorem 1.1).

5.3 | Multilevel Picard approximations

Let $\mathfrak{M} \in \mathbb{N}$, $\mathbf{I} = \cup_{n=1}^{\infty} \mathbb{Z}^n$, let $W^{\mathbf{i}} : \Omega \to \mathbb{R}^{\mathbf{d}}$, $\mathbf{i} \in \mathbf{I}$, be i.i.d. random variables which satisfy for all $A \in \mathcal{B}(\mathbb{R}^{\mathbf{d}})$ that $\mathbb{P}(W^0 \in A) = \mathbb{P}(\mathcal{W} \in A)$, and let $V_n^{\mathbf{i}} : \mathfrak{P} \times \Omega \to \mathbb{R}$, $n \in \mathbb{Z}$, $\mathbf{i} \in \mathbf{I}$, satisfy[2] for all $n \in \mathbb{Z}$, $\mathbf{i} \in \mathbf{I}$, $p \in \mathfrak{P}$ that

$$\begin{aligned} V_n^{\mathbf{i}}(p) = \sum_{k=0}^{n-1} \frac{1}{\mathfrak{M}^{n-k}} \sum_{\mathfrak{m}=1}^{\mathfrak{M}^{n-k}} \Big[\phi\Big(p, W^{(\mathbf{i},k,\mathfrak{m})}, V_k^{(\mathbf{i},k,\mathfrak{m})}\big(\mathcal{P}\big(p, W^{(\mathbf{i},k,\mathfrak{m})}\big)\big)\Big) \\ - \mathbb{1}_{\mathbb{N}}(k)\phi\Big(p, W^{(\mathbf{i},k,\mathfrak{m})}, V_{k-1}^{(\mathbf{i},k-1,-\mathfrak{m})}\big(\mathcal{P}\big(p, W^{(\mathbf{i},k,\mathfrak{m})}\big)\big)\Big)\Big]. \end{aligned} \tag{61}$$

[2] Note that for all $n \in \mathbb{N}$, $\mathbf{i} = (\mathbf{i}_1, \dots, \mathbf{i}_n) \in \mathbb{Z}^n$, $k, m \in \mathbb{Z}$ we denote by $(\mathbf{i}, k, m) \in \mathbb{Z}^{n+2}$ the vector given by $(\mathbf{i}, k, m) = (\mathbf{i}_1, \mathbf{i}_2, \dots, \mathbf{i}_n, k, m)$.

Note that under suitable integrability conditions (61) implies for all $n \in \mathbb{N}$, $p \in \mathfrak{P}$, $\mathbf{i} \in \mathbf{I}$ that $V^{\mathbf{i}}_0(p) = 0$ and

$$\begin{aligned}\mathbb{E}\big[V^{\mathbf{i}}_n(p)\big] &= \sum_{k=0}^{n-1} \frac{1}{\mathfrak{M}^{n-k}} \sum_{\mathfrak{m}=1}^{\mathfrak{M}^{n-k}} \Big[\mathbb{E}\Big[\phi\Big(p, W^{(\mathbf{i},k,\mathfrak{m})}, V^{(\mathbf{i},k,\mathfrak{m})}_k\big(\mathcal{P}\big(p, W^{(\mathbf{i},k,\mathfrak{m})}\big)\big)\Big)\Big] \\ &\quad -\mathbb{1}_{\mathbb{N}}(k)\mathbb{E}\Big[\phi\Big(p, W^{(\mathbf{i},k,\mathfrak{m})}, V^{(\mathbf{i},k-1,-\mathfrak{m})}_{k-1}\big(\mathcal{P}\big(p, W^{(\mathbf{i},k,\mathfrak{m})}\big)\big)\Big)\Big]\Big] \\ &= \sum_{k=0}^{n-1} \big[\mathbb{E}\big[\phi\big(p, W^0, V^0_k\big(\mathcal{P}(p, W^0)\big)\big)\big] - \mathbb{1}_{\mathbb{N}}(k)\mathbb{E}\big[\phi\big(p, W^0, V^0_{k-1}\big(\mathcal{P}(p, W^0)\big)\big)\big]\big] \\ &= \mathbb{E}\big[\phi\big(p, W^0, V^0_{n-1}\big(\mathcal{P}(p, W^0)\big)\big)\big].\end{aligned} \tag{62}$$

Induction thus shows that for every $n \in \mathbb{N}$ the identically distributed random functions $V^{\mathbf{i}}_n : \mathfrak{P} \times \Omega \to \mathbb{R}$, $\mathbf{i} \in \mathbf{I}$, correspond in expectation to the n-th fixed point iterate of the fixed point equation in (55). For every $n \in \mathbb{N}$ the recursive definition of $(V^{\mathbf{i}}_n)_{\mathbf{i}\in\mathbf{I}}$ in (61) thus represents an approximated fixed point iteration step in which the expectation of the fixed point iteration is approximated by a MLMC sum over previously computed approximated fixed point iterates $(V^{\mathbf{i}}_k)_{(\mathbf{i},k)\in\mathbf{I}\times\{0,1,\dots,n-1\}}$. Under suitable assumptions (see, e.g., Hutzenthaler, Jentzen, von Wurstemberger, 2020 for precise assumptions and a more detailed derivation of MLP algorithms in the case of semilinear PDEs) we expect for sufficiently large $n \in \mathbb{N}$ that

$$u(p) \approx V^0_n(p). \tag{63}$$

5.4 | Replacing the random variables in multilevel Picard approximations

Let $(C_n)_{n\in\mathbb{Z}} \subseteq \mathbb{N}_0$ satisfy for all $n \in \mathbb{Z}$ that

$$C_n = \sum_{k=0}^{n-1} \mathfrak{M}^{n-k}(1 + C_k + C_{k-1}), \tag{64}$$

for every $n, \mathfrak{m} \in \mathbb{N}$, $l \in \{1, 2, \dots, n\}$ let $c_{n,l,\mathfrak{m}} \in \mathbb{N}$ satisfy

$$c_{n,l,\mathfrak{m}} = \left[\sum_{k=0}^{l-1} \mathfrak{M}^{n-k}(1 + C_k + C_{k-1})\right] + (\mathfrak{m} - 1)(1 + C_l + C_{l-1}) + 1, \tag{65}$$

let $\mathcal{N}_n : \mathfrak{P} \times \mathbb{R}^{C_n\mathbf{d}}$, $n \in \mathbb{N}_0$, satisfy for all $n \in \mathbb{N}$, $p \in \mathfrak{P}$, $\theta_1, \theta_2, \dots, \theta_{C_n} \in \mathbb{R}^{\mathbf{d}}$ that $\mathcal{N}_0(p) = 0$ and

$$\mathcal{N}_n(p, (\theta_r)_{r\in\{1,2,\dots,C_n\}}) = \left[\frac{1}{\mathfrak{M}^n} \sum_{\mathfrak{m}=1}^{\mathfrak{M}^n} \phi(p, \theta_{\mathfrak{m}}, 0)\right]$$

$$+\sum_{k=1}^{n-1}\frac{1}{\mathfrak{M}^{n-k}}\sum_{\mathfrak{m}=1}^{\mathfrak{M}^{n-k}}\Big[\phi\Big(p,\theta_{c_{n,k,\mathfrak{m}}},\mathcal{N}_k\Big(\mathcal{P}(p,\theta_{c_{n,k,\mathfrak{m}}}),(\theta_{c_{n,k,\mathfrak{m}}+r})_{r\in\{1,2,\dots,C_k\}}\Big)\Big)$$

$$-\phi\Big(p,\theta_{c_{n,k,\mathfrak{m}}},\mathcal{N}_{k-1}\Big(\mathcal{P}(p,\theta_{c_{n,k,\mathfrak{m}}}),(\theta_{c_{n,k,\mathfrak{m}}+C_k+r})_{r\in\{1,2,\dots,C_{k-1}\}}\Big)\Big)\Big], \tag{66}$$

let $N\in\mathbb{N}$, assume for all $p\in\mathfrak{P}$ that $\mathbb{R}^{C_N\mathbf{d}}\ni\theta\mapsto\mathcal{N}(p,\theta)\in\mathbb{R}$ is continuously differentiable, and let $\mathfrak{W}_k:\Omega\to\mathbb{R}^{\mathbf{d}}$, $k\in\{1,2,\dots,C_N\}$, be i.i.d. random variables which satisfy for all $B\in\mathcal{B}(\mathbb{R}^{\mathbf{d}})$ that $\mathbb{P}(\mathfrak{W}_1\in B)=\mathbb{P}(\mathcal{W}\in B)$. Observe that induction shows that for all $n\in\mathbb{N}_0$, $p\in\mathfrak{P}$, $\mathbf{i}\in\mathbf{I}$ the number $C_n\in\mathbb{N}$ corresponds, roughly speaking, to the number of realizations of $\mathbf{d}$-dimensional random variables required to compute one random realization of $V_n^{\mathbf{i}}(p)$. Moreover, note that (61), (64), (65), and (66) assure that for all $p\in\mathfrak{P}$, $B\in\mathcal{B}(\mathbb{R})$ it holds that

$$\mathbb{P}\big(\mathcal{N}_N\big(p,(\mathfrak{W}_1,\mathfrak{W}_2,\dots,\mathfrak{W}_{C_N})\big)\in B\big)=\mathbb{P}(V_N^0(p)\in B). \tag{67}$$

Combining this and (63) suggests that for all $p\in\mathfrak{P}$ it holds that

$$\mathcal{N}_N(p,\mathfrak{W})\approx u(p). \tag{68}$$

5.5 | Random loss functions for fixed random variables in multilevel Picard approximations

Let $\mathbf{M},\mathcal{N}\in\mathbb{N}$, let $P_{m,\mathbf{m}}:\Omega\to\mathfrak{P}$, $m,\mathbf{m}\in\mathbb{N}$, be i.i.d. random variables, assume that $(P_{m,\mathbf{m}})_{(m,\mathbf{m})\in\mathbb{N}^2}$, $(W^{\mathbf{i}})_{\mathbf{i}\in\mathbf{I}}$, and $(\mathfrak{W}_k)_{k\in\{1,2,\dots,C_N\}}$ are independent, for every $m\in\mathbb{N}$ let $F_m:\mathbb{R}^{C_N\mathbf{d}}\times\Omega\to\mathbb{R}$ satisfy for all $\theta\in\mathbb{R}^{C_N\mathbf{d}}$ that

$$F_m(\theta)=\frac{1}{\mathbf{M}}\left[\sum_{\mathbf{m}=1}^{\mathbf{M}}\left|V_{\mathcal{N}}^{(m,\mathbf{m})}(P_{m,\mathbf{m}})-\mathcal{N}_N(P_{m,\mathbf{m}},\theta)\right|^2\right], \tag{69}$$

and for every $m\in\mathbb{N}$ let $G_m:\mathbb{R}^{C_N\mathbf{d}}\times\Omega\to\mathbb{R}^{C_N\mathbf{d}}$ satisfy for all $\theta\in\mathbb{R}^{C_N\mathbf{d}}$, $\omega\in\Omega$ that

$$G_m(\theta,\omega)=(\nabla_\theta F_m)(\theta,\omega). \tag{70}$$

5.6 | Learning the random variables with stochastic gradient descent

Let $(\gamma_m)_{m\in\mathbb{N}}\subseteq(0,\infty)$ and let $\Theta:\mathbb{N}_0\times\Omega\to\mathbb{R}^{C_N\mathbf{d}}$ satisfy for all $m\in\mathbb{N}$ that $\Theta_0=(\mathfrak{W}_1,\mathfrak{W}_2,\dots,\mathfrak{W}_{C_N})$ and

$$\Theta_m=\Theta_{m-1}-\gamma_m G_m(\Theta_{m-1}). \tag{71}$$

For every sufficiently large $m\in\mathbb{N}$ we propose to employ the random function $\mathfrak{P}\times\Omega\ni(p,\omega)\mapsto\mathcal{N}_N(p,\Theta_m(\omega))\in\mathbb{R}$ as an approximation for the target function $\mathfrak{P}\ni p\mapsto u(p)\in\mathbb{R}$ in (55).

5.7 | Description of the proposed approximation algorithm

Framework 5.1. *Let $\mathfrak{p}, \mathbf{d}, \mathfrak{M}, N, \mathcal{N}, \mathbf{M} \in \mathbb{N}$, $(\gamma_m)_{m\in\mathbb{N}} \subseteq (0,\infty)$, let $\mathfrak{P} \subseteq \mathbb{R}^{\mathfrak{p}}$ be measurable, let $\phi : \mathfrak{P} \times \mathbb{R}^{\mathbf{d}} \times \mathbb{R} \to \mathbb{R}$ and $\mathcal{P} : \mathfrak{P} \times \mathbb{R}^{\mathbf{d}} \to \mathfrak{P}$ be measurable, let $(C_n)_{n\in\mathbb{Z}} \subseteq \mathbb{N}_0$ satisfy for all $n \in \mathbb{Z}$ that*

$$C_n = \sum_{k=0}^{n-1} \mathfrak{M}^{n-k}(1 + C_k + C_{k-1}), \tag{72}$$

for every $n, \mathfrak{m} \in \mathbb{N}$, $l \in \{1, 2, \dots, n\}$ let $c_{n,l,\mathfrak{m}} \in \mathbb{N}$ satisfy

$$c_{n,l,\mathfrak{m}} = \left[\sum_{k=0}^{l-1} \mathfrak{M}^{n-k}(1 + C_k + C_{k-1})\right] + (\mathfrak{m} - 1)(1 + C_l + C_{l-1}) + 1, \tag{73}$$

let $\mathcal{N}_n : \mathfrak{P} \times \mathbb{R}^{C_n\mathbf{d}}$, $n \in \mathbb{N}_0$, satisfy for all $n \in \mathbb{N}$, $p \in \mathfrak{P}$, $\theta_1, \theta_2, \dots, \theta_{C_n} \in \mathbb{R}^{\mathbf{d}}$ that $\mathcal{N}_0(p) = 0$ and

$$\mathcal{N}_n(p, (\theta_1, \theta_2, \dots, \theta_{C_n})) = \left[\frac{1}{\mathfrak{M}^n} \sum_{\mathfrak{m}=1}^{\mathfrak{M}^n} \phi(p, \theta_{\mathfrak{m}}, 0)\right] \tag{74}$$

$$+ \sum_{k=1}^{n-1} \frac{1}{\mathfrak{M}^{n-k}} \sum_{\mathfrak{m}=1}^{\mathfrak{M}^{n-k}} \Big[\phi\Big(p, \theta_{c_{n,k,\mathfrak{m}}}, \mathcal{N}_k\Big(\mathcal{P}(p, \theta_{c_{n,k,\mathfrak{m}}}), (\theta_{c_{n,k,\mathfrak{m}}+r})_{r\in\{1,2,\dots,C_k\}}\Big)$$

$$- \phi\Big(p, \theta_{c_{n,k,\mathfrak{m}}}, \mathcal{N}_{k-1}\Big(\mathcal{P}(p, \theta_{c_{n,k,\mathfrak{m}}}), (\theta_{c_{n,k,\mathfrak{m}}+C_k+r})_{r\in\{1,2,\dots,C_{k-1}\}}\Big)\Big)\Big], \tag{74}$$

assume for all $p \in \mathfrak{P}$ that $\mathbb{R}^{C_N\mathbf{d}} \ni \theta \mapsto \mathcal{N}(p, \theta) \in \mathbb{R}$ is continuously differentiable, let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, let $\mathcal{W} : \Omega \to \mathbb{R}^{\mathbf{d}}$ be a random variable, let $\mathfrak{W}_k : \Omega \to \mathbb{R}^{\mathbf{d}}$, $k \in \{1, 2, \dots, C_N\}$, be i.i.d. random variables, let $W_k^{m,\mathbf{m}} : \Omega \to \mathbb{R}^{\mathbf{d}}$, $m, \mathbf{m} \in \mathbb{N}$, $k \in \{1, 2, \dots, C_{\mathcal{N}}\}$, be i.i.d. random variables, assume for all $B \in \mathcal{B}(\mathbb{R}^{\mathbf{d}})$ that $\mathbb{P}(\mathfrak{W}_1 \in B) = \mathbb{P}(\mathcal{W} \in B) = \mathbb{P}(W_1^{1,1} \in B)$, let $P_{m,\mathbf{m}} : \Omega \to \mathfrak{P}$, $m, \mathbf{m} \in \mathbb{N}$, be i.i.d. random variables, assume that $(P_{m,\mathbf{m}})_{(m,\mathbf{m})\in\mathbb{N}^2}$, $(W_k^{m,\mathbf{m}})_{(m,\mathbf{m},k)\in\mathbb{N}^2\times\{1,2,\dots,C_{\mathcal{N}}\}}$, and $(\mathfrak{W}_k)_{k\in\{1,2,\dots,C_N\}}$ are independent, for every $m \in \mathbb{N}$ let $F_m : \mathbb{R}^{C_N\mathbf{d}} \times \Omega \to \mathbb{R}$ satisfy for all $\theta \in \mathbb{R}^{C_N\mathbf{d}}$ that

$$F_m(\theta) = \frac{1}{\mathbf{M}}\left[\sum_{\mathbf{m}=1}^{\mathbf{M}} \left|\mathcal{N}_{\mathcal{N}}\Big(P_{m,\mathbf{m}}, \Big(W_1^{m,\mathbf{m}}, W_2^{m,\mathbf{m}}, \dots, W_{C_{\mathcal{N}}}^{m,\mathbf{m}}\Big)\Big) - \mathcal{N}_N\big(P_{m,\mathbf{m}}, \theta\big)\right|^2\right], \tag{75}$$

for every $m \in \mathbb{N}$ let $G_m : \mathbb{R}^{C_N\mathbf{d}} \times \Omega \to \mathbb{R}^{C_N\mathbf{d}}$ satisfy for all $\theta \in \mathbb{R}^{C_N\mathbf{d}}$, $\omega \in \Omega$ that $G_m(\theta, \omega) = (\nabla_\theta F_m)(\theta, \omega)$, and let $\Theta : \mathbb{N}_0 \times \Omega \to \mathbb{R}^{C_N\mathbf{d}}$ satisfy for all $m \in \mathbb{N}$ that $\Theta_0 = (\mathfrak{W}_1, \mathfrak{W}_2, \dots, \mathfrak{W}_{C_N})$ and $\Theta_m = \Theta_{m-1} - \gamma_m G_m(\Theta_{m-1})$.

6 | LRV STRATEGY IN THE CASE OF A GENERAL PROPOSAL ALGORITHM

In this section we derive and formulate the LRV strategy in its most general form, which contains all the algorithms derived in the previous sections as special cases. Roughly speaking, we want to approximate a target function (cf. $u : \mathfrak{P} \to \mathbb{R}^k$ in Subsection 6.1) for which we already have a generic stochastic approximation algorithm (cf. $\Psi : \mathfrak{P} \times \mathbb{R}^{\mathfrak{d}} \to \mathbb{R}^k$ and $\mathfrak{W} : \Omega \to \mathbb{R}^{\mathfrak{d}}$ in (76) in Subsection 6.1). We refer to this algorithm as *proposal algorithm*. Moreover, we assume that we are able to generate random reference solutions which approximate the target function at every point in expectation (cf. $\mathfrak{P} \times \Omega \ni (p, \omega) \mapsto \Xi(p, \mathbf{W}(\omega)) \in \mathbb{R}^k$ in (77) in Subsection 6.1).

In the next few sentences we briefly sketch in words the LRV strategy in this general case. The first step of the LRV strategy is to consider the random variables in the stochastic approximation algorithm as parameters for a parametric family of functions (corresponding to $\mathfrak{P} \ni p \mapsto \Psi(p, \theta) \in \mathbb{R}^k$, $\theta \in \mathbb{R}^{\mathfrak{d}}$, in Subsection 6.1); see Subsection 6.2. The goal of the LRV strategy is then to "learn" parameters whose corresponding function yields a good approximation of the target function $u : \mathfrak{P} \to \mathbb{R}^k$. Taking this into account, the second step of the LRV strategy is to minimize a loss function (cf. (78) in Subsection 6.2) measuring the distance between the approximating function and the approximate reference solutions with an SGD-type optimization method; see Subsection 6.3. As initial guess for the SGD-type learning procedure we suggest to randomly choose the parameters according to the distribution of the random variables appearing in the proposal algorithm, since we know that this already results in a passable approximation. This feature of the LRV strategy is an important advantage when compared to standard deep learning methods in the sense that the LRV strategy has already in the beginning of the training procedure a relatively small loss function. The entire approach is presented in one single framework in Subsection 6.4.

One of the differences between this section and the previous sections is that in the previous sections we only used, for simplicity, the plain vanilla SGD method, however in this section we allow for various more sophisticated SGD-type optimization methods (cf. (80) in Subsection 6.3). Some of these more sophisticated SGD-type methods are presented in Subsection 6.5 as special cases of the framework in Subsection 6.4.

6.1 | Stochastic approximations for general target functions related to parametric expectations

Let $\mathfrak{p}, \mathfrak{d}, k, \mathbf{d} \in \mathbb{N}$, let $\mathfrak{P} \subseteq \mathbb{R}^{\mathfrak{p}}$ be measurable, let $\Psi : \mathfrak{P} \times \mathbb{R}^{\mathfrak{d}} \to \mathbb{R}^k$, $\Xi : \mathfrak{P} \times \mathbb{R}^{\mathbf{d}} \to \mathbb{R}^k$, and $u : \mathfrak{P} \to \mathbb{R}^k$ be measurable, assume for all $p \in \mathfrak{P}$ that $\mathbb{R}^{\mathfrak{d}} \ni w \mapsto \Psi(p, w) \in \mathbb{R}^k$ is continuously differentiable, let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, let $\mathfrak{W} : \Omega \to \mathbb{R}^{\mathfrak{d}}$ and $\mathbf{W} : \Omega \to \mathbb{R}^{\mathbf{d}}$ be independent random variables, and assume for all $p \in \mathfrak{P}$ that $\mathbb{E}[\|\Xi(p, \mathbf{W})\|] < \infty$. We think of $(\Psi, \mathfrak{W})$ as a stochastic approximation algorithm for $u : \mathfrak{P} \to \mathbb{R}^k$ in the sense that for all $p \in \mathfrak{P}$ it holds that

$$u(p) \approx \Psi(p, \mathfrak{W}) \tag{76}$$

and for every $p \in \mathfrak{P}$ we think of $\mathbb{E}[\Xi(p, \mathbf{W})]$ as a suitable approximation

$$u(p) \approx \mathbb{E}[\Xi(p, \mathbf{W})] \tag{77}$$

of $u(p)$. The goal of this section is to derive an algorithm to approximately compute the function $u : \mathfrak{P} \to \mathbb{R}$.

6.2 | Random loss functions for fixed random variables in stochastic approximations

Let $\mathbf{M} \in \mathbb{N}$, let $\mathfrak{C} : \mathbb{R}^k \times \mathbb{R}^k \to [0, \infty)$ continuously differentiable, let $P_{m,\mathbf{m}} : \Omega \to \mathfrak{P}$, $m, \mathbf{m} \in \mathbb{N}$, be i.i.d. random variables, let $W^{m,\mathfrak{m}} : \Omega \to \mathbb{R}^{\mathbf{d}}$, $m, \mathfrak{m} \in \mathbb{N}$, be i.i.d. random variables which satisfy for all $A \in \mathcal{B}(\mathbb{R}^d)$ that $\mathbb{P}(W^{0,0} \in A) = \mathbb{P}(\mathbf{W} \in A)$, assume that $(P_{m,\mathbf{m}})_{(m,\mathbf{m})\in\mathbb{N}^2}$ and $(W^{m,\mathfrak{m}})_{(m,\mathfrak{m})\in\mathbb{N}^2}$ are independent, and for every $m \in \mathbb{N}$ let $F_m : \mathbb{R}^{\mathfrak{d}} \times \Omega \to \mathbb{R}$ satisfy for all $\theta \in \mathbb{R}^{\mathfrak{d}}$ that

$$F_m(\theta) = \frac{1}{\mathbf{M}} \left[\sum_{\mathbf{m}=1}^{\mathbf{M}} \mathfrak{C}\big(\Xi\big(P_{m,\mathbf{m}}, W^{m,\mathbf{m}}\big), \Psi(P_{m,\mathbf{m}}, \theta)\big) \right], \tag{78}$$

and for every $m \in \mathbb{N}$ let $G_m : \mathbb{R}^{\mathfrak{d}} \times \Omega \to \mathbb{R}^{\mathfrak{d}}$ satisfy for all $\theta \in \mathbb{R}^{\mathfrak{d}}$, $\omega \in \Omega$ that

$$G_m(\theta, \omega) = (\nabla_\theta F_m)(\theta, \omega). \tag{79}$$

6.3 | Learning the random variables with stochastic gradient descent-type methods

Let $\psi_m : \mathbb{R}^{m\mathfrak{d}} \to \mathbb{R}^{\mathfrak{d}}$, $m \in \mathbb{N}$, be functions and let $\Theta : \mathbb{N}_0 \times \Omega \to \mathbb{R}^{\mathfrak{d}}$ satisfy for all $m \in \mathbb{N}$ that $\Theta_0 = \mathfrak{W}$ and

$$\Theta_m = \Theta_{m-1} - \psi_m((G_1(\Theta_0), G_2(\Theta_1), \dots, G_m(\Theta_{m-1}))). \tag{80}$$

For every sufficiently large $m \in \mathbb{N}$ we propose to employ the random function $\mathfrak{P} \times \Omega \ni (p, \omega) \mapsto \Psi(p, \Theta_m(\omega)) \in \mathbb{R}$ as an approximation for the target function $\mathfrak{P} \ni p \mapsto u(p) \in \mathbb{R}$.

6.4 | Description of the proposed approximation algorithm

Framework 6.1. *Let $\mathfrak{p}, \mathfrak{d}, \mathbf{d}, k, \mathbf{M} \in \mathbb{N}$, let $\mathfrak{P} \subseteq \mathbb{R}^{\mathfrak{p}}$ be measurable, let $\Psi : \mathfrak{P} \times \mathbb{R}^{\mathfrak{d}} \to \mathbb{R}^k$, $\Xi : \mathfrak{P} \times \mathbb{R}^{\mathbf{d}} \to \mathbb{R}^k$, and $\mathfrak{C} : \mathbb{R}^k \times \mathbb{R}^k \to [0, \infty)$ be functions, let $\psi_m = (\psi_m^{(1)}, \dots, \psi_m^{(\mathfrak{d})}) : \mathbb{R}^{m\mathfrak{d}} \to \mathbb{R}^{\mathfrak{d}}$, $m \in \mathbb{N}$, be functions, let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, let $\mathfrak{W} : \Omega \to \mathbb{R}^{\mathfrak{d}}$ be a random variable, let $P_{m,\mathbf{m}} : \Omega \to \mathfrak{P}$, $m, \mathbf{m} \in \mathbb{N}$, be i.i.d. random variables, let $W^{m,\mathfrak{m}} : \Omega \to \mathbb{R}^{\mathbf{d}}$, $m, \mathfrak{m} \in \mathbb{N}$, be i.i.d. random variables, assume that $\mathfrak{W}$, $(P_{m,\mathbf{m}})_{(m,\mathbf{m})\in\mathbb{N}^2}$, and $(W^{m,\mathfrak{m}})_{(m,\mathfrak{m})\in\mathbb{N}^2}$ are independent, for every $m \in \mathbb{N}$ let $F_m : \mathbb{R}^{\mathfrak{d}} \times \Omega \to \mathbb{R}$ satisfy for all $\theta \in \mathbb{R}^{\mathfrak{d}}$ that*

$$F_m(\theta) = \frac{1}{\mathbf{M}} \left[\sum_{\mathbf{m}=1}^{\mathbf{M}} \mathfrak{C}\big(\Xi\big(P_{m,\mathbf{m}}, W^{m,\mathbf{m}}\big), \Psi(P_{m,\mathbf{m}}, \theta)\big) \right], \tag{81}$$

for every $m \in \mathbb{N}$ let $G_m = (G_m^{(1)}, \dots, G_m^{(\mathfrak{d})}) : \mathbb{R}^{\mathfrak{d}} \times \Omega \to \mathbb{R}^{\mathfrak{d}}$ satisfy for all $\omega \in \Omega$, $\theta \in \{v \in \mathbb{R}^{\mathfrak{d}} : F_m(\cdot, \omega)$ is differentiable at $v\}$ that $G_m(\theta, \omega) = (\nabla_\theta F_m)(\theta, \omega)$, and let $\Theta = (\Theta^{(1)}, \dots, \Theta^{(\mathfrak{d})}) : \mathbb{N}_0 \times \Omega \to \mathbb{R}^{\mathfrak{d}}$ be a stochastic process which satisfies for all $m \in \mathbb{N}$ that $\Theta_0 = \mathfrak{W}$ and

$$\Theta_m = \Theta_{m-1} - \psi_m(G_1(\Theta_0), G_2(\Theta_1), \dots, G_m(\Theta_{m-1})). \tag{82}$$

6.5 | Explicit descriptions of some popular stochastic gradient descent-type methods

6.5.1 | Standard stochastic gradient descent (SGD)

Lemma 6.2. *Assume Framework 6.1, let $(\gamma_m)_{m\in\mathbb{N}} \subseteq (0,\infty)$, and assume for all $m \in \mathbb{N}$, $g_1, g_2, \ldots, g_m \in \mathbb{R}^{\mathfrak{d}}$ that $\psi_m(g_1, g_2, \ldots, g_m) = \gamma_m g_m$. Then it holds for all $m \in \mathbb{N}$ that*

$$\Theta_m = \Theta_{m-1} - \gamma_m G_m(\Theta_{m-1}). \tag{83}$$

6.5.2 | Stochastic gradient descent with momentum (SGD with momentum)

Lemma 6.3. *Assume Framework 6.1, let $(\gamma_m)_{m\in\mathbb{N}} \subseteq (0,\infty)$, $\alpha \in (0,1)$, and assume for all $m \in \mathbb{N}$, $g_1, g_2, \ldots, g_m \in \mathbb{R}^{\mathfrak{d}}$ that $\psi_m(g_1, g_2, \ldots, g_m) = \gamma_m \sum_{k=1}^{m} \alpha^{m-k}(1-\alpha) g_k$. Then there exists $\mathbf{m} : \mathbb{N}_0 \times \Omega \to \mathbb{R}^d$ such that for all $m \in \mathbb{N}$ it holds that*

$$\mathbf{m}_0 = 0, \qquad \mathbf{m}_m = \alpha \mathbf{m}_{m-1} + (1-\alpha) G_m(\Theta_{m-1}), \qquad \textit{and} \qquad \Theta_n = \Theta_{n-1} - \gamma_n \mathbf{m}_n. \tag{84}$$

6.5.3 | Adaptive stochastic gradient descent (Adagrad)

Lemma 6.4. *Assume Framework 6.1, let $(\gamma_m)_{m\in\mathbb{N}} \subseteq (0,\infty)$, $\varepsilon \in (0,\infty)$, and assume for all $m \in \mathbb{N}$, $i \in \{1, 2, \ldots, \mathfrak{d}\}$, $g_1 = (g_1^{(1)}, \ldots, g_1^{(\mathfrak{d})}), g_2 = (g_2^{(1)}, \ldots, g_2^{(\mathfrak{d})}), \ldots, g_m = (g_m^{(1)}, \ldots, g_m^{(\mathfrak{d})}) \in \mathbb{R}^{\mathfrak{d}}$ that*

$$\psi_m^{(i)}(g_1, g_2, \ldots, g_m) = \left[\frac{\gamma_m}{\left(\varepsilon + \sum_{k=1}^{m} |g_k^{(i)}|^2\right)^{1/2}}\right] g_m^{(i)}. \tag{85}$$

Then it holds for all $m \in \mathbb{N}$, $i \in \{1, 2, \ldots, \mathfrak{d}\}$ that

$$\Theta_m^{(i)} = \Theta_{m-1}^{(i)} - \left[\frac{\gamma_m}{\left(\varepsilon + \sum_{k=1}^{m} |G_k^{(i)}(\Theta_{k-1})|^2\right)^{1/2}}\right] G_m^{(i)}(\Theta_{m-1}). \tag{86}$$

6.5.4 | Root mean square error propagation stochastic gradient descent (RMSprop)

Lemma 6.5. *Assume Framework 6.1, let $(\gamma_m)_{m\in\mathbb{N}} \subseteq (0,\infty)$, $\varepsilon \in (0,\infty)$, $\beta \in (0,1)$, and assume for all $m \in \mathbb{N}$, $i \in \{1, 2, \ldots, \mathfrak{d}\}$, $g_1 = (g_1^{(1)}, \ldots, g_1^{(\mathfrak{d})}), g_2 = (g_2^{(1)}, \ldots, g_2^{(\mathfrak{d})}), \ldots, g_m = (g_m^{(1)}, \ldots, g_m^{(\mathfrak{d})}) \in \mathbb{R}^{\mathfrak{d}}$ that*

$$\psi_m^{(i)}(g_1, g_2, \ldots, g_m) = \left[\frac{\gamma_m}{\left(\varepsilon + \sum_{k=1}^{m} \beta^{m-k}(1-\beta)|g_k^{(i)}|^2\right)^{1/2}}\right] g_m^{(i)}. \tag{87}$$

Then there exists $\mathbb{M} = (\mathbb{M}^{(1)}, \dots, \mathbb{M}^{(\mathfrak{d})}) : \mathbb{N}_0 \times \Omega \to \mathbb{R}^{\mathfrak{d}}$ *such that for all* $m \in \mathbb{N}$, $i \in \{1, 2, \dots, \mathfrak{d}\}$ *it holds that*

$$\mathbb{M}_0 = 0, \qquad \mathbb{M}_m^{(i)} = \beta \mathbb{M}_{m-1}^{(i)} + (1-\beta)|G_m^{(i)}(\Theta_{m-1})|^2, \tag{88}$$

$$\text{and} \qquad \Theta_m^{(i)} = \Theta_{m-1}^{(i)} - \left[\frac{\gamma_m}{(\varepsilon + \mathbb{M}_m^{(i)})^{1/2}}\right] G_m^{(i)}(\Theta_{m-1}). \tag{89}$$

6.5.5 | Adadelta stochastic gradient descent (Adadelta)

Lemma 6.6. *Assume Framework* 6.1*, let* $\varepsilon \in (0, \infty)$*,* $\beta, \delta \in (0, 1)$*, and assume for all* $m \in \mathbb{N}$*,* $i \in \{1, 2, \dots, \mathfrak{d}\}$*,* $\mathrm{g}_1 = (\mathrm{g}_1^{(1)}, \dots, \mathrm{g}_1^{(\mathfrak{d})}), \mathrm{g}_2 = (\mathrm{g}_2^{(1)}, \dots, \mathrm{g}_2^{(\mathfrak{d})}), \dots, \mathrm{g}_m = (\mathrm{g}_m^{(1)}, \dots, \mathrm{g}_m^{(\mathfrak{d})}) \in \mathbb{R}^{\mathfrak{d}}$ *that*

$$\psi_m^{(i)}(\mathrm{g}_1, \mathrm{g}_2, \dots, \mathrm{g}_m) = \left[\frac{\varepsilon + \sum_{k=1}^{m-1} \delta^{m-1-k}(1-\delta)|\psi_k^{(i)}(\mathrm{g}_1, \mathrm{g}_2, \dots, \mathrm{g}_k)|^2}{\varepsilon + \sum_{k=1}^{m} \beta^{m-k}(1-\beta)|\mathrm{g}_k^{(i)}|^2}\right]^{1/2} \mathrm{g}_m^{(i)}. \tag{90}$$

Then there exist $\mathbb{M} = (\mathbb{M}^{(1)}, \dots, \mathbb{M}^{(d)})$ *and* $\Delta = (\Delta^{(1)}, \dots, \Delta^{(d)}) : \mathbb{N}_0 \times \Omega \to \mathbb{R}^d$ *such that for all* $m \in \mathbb{N}$*,* $i \in \{1, 2, \dots, d\}$ *it holds that*

$$\mathbb{M}_0 = 0, \qquad \Delta_0 = 0, \qquad \mathbb{M}_m^{(i)} = \beta\, \mathbb{M}_{m-1}^{(i)} + (1-\beta)\left|G_m^{(i)}(\Theta_{m-1})\right|^2, \tag{91}$$

$$\Theta_m^{(i)} = \Theta_{m-1}^{(i)} - \left[\frac{\varepsilon + \Delta_{m-1}^{(i)}}{\varepsilon + \mathbb{M}_m^{(i)}}\right]^{1/2} G_m^{(i)}(\Theta_{m-1}), \qquad \text{and} \qquad \Delta_m^{(i)} = \delta \Delta_{m-1}^{(i)} + (1-\delta)\left|\Theta_m^{(i)} - \Theta_{m-1}^{(i)}\right|^2. \tag{92}$$

6.5.6 | Adamax stochastic gradient descent (Adamax)

Lemma 6.7. *Assume Framework* 6.1*, let* $\varepsilon \in (0, \infty)$*,* $\alpha, \beta \in (0, 1)$*, and assume for all* $m \in \mathbb{N}$*,* $i \in \{1, 2, \dots, \mathfrak{d}\}$*,* $\mathrm{g}_1 = (\mathrm{g}_1^{(1)}, \dots, \mathrm{g}_1^{(\mathfrak{d})}), \mathrm{g}_2 = (\mathrm{g}_2^{(1)}, \dots, \mathrm{g}_2^{(\mathfrak{d})}), \dots, \mathrm{g}_m = (\mathrm{g}_m^{(1)}, \dots, \mathrm{g}_m^{(\mathfrak{d})}) \in \mathbb{R}^{\mathfrak{d}}$ *that*

$$\psi_m^{(i)}(\mathrm{g}_1, \mathrm{g}_2, \dots, \mathrm{g}_m) = \gamma_m \left[\frac{\sum_{k=1}^{m} \alpha^{m-k}(1-\alpha)\mathrm{g}_k^{(i)}}{1-\alpha^m}\right] \left[\varepsilon + \max\left\{\left|\mathrm{g}_m^{(i)}\right|, \beta\left|\mathrm{g}_{m-1}^{(i)}\right|, \dots, \beta^{m-1}\left|\mathrm{g}_1^{(i)}\right|\right\}\right]^{-1}. \tag{93}$$

Then there exist $\mathbf{m} = (\mathbf{m}^{(1)}, \dots, \mathbf{m}^{(d)})$ *and* $\mathbb{M} = (\mathbb{M}^{(1)}, \dots, \mathbb{M}^{(d)}) : \mathbb{N}_0 \times \Omega \to \mathbb{R}^d$ *such that for all* $m \in \mathbb{N}$*,* $i \in \{1, 2, \dots, d\}$ *it holds that*

$$\mathbf{m}_0 = 0, \qquad \mathbf{m}_m = \alpha \mathbf{m}_{m-1} + (1-\alpha) G_m(\Theta_{m-1}), \tag{94}$$

$$\mathbb{M}_0 = 0, \qquad \mathbb{M}_m^{(i)} = \max\left\{\beta\, \mathbb{M}_{m-1}^{(i)}, \left|G_m^{(i)}(\Theta_{m-1})\right|^2\right\}, \tag{95}$$

$$\text{and} \qquad \Theta_m^{(i)} = \Theta_{m-1}^{(i)} - \gamma_m \left[\frac{\mathbf{m}_m^{(i)}}{1-\alpha^m}\right] \left[\varepsilon + \mathbb{M}_m^{(i)}\right]^{-1}. \tag{96}$$

6.5.7 | Adaptive moment estimation stochastic gradient descent (Adam)

Lemma 6.8. *Assume Framework 6.1, let* $\varepsilon \in (0,\infty)$, $\alpha,\beta \in (0,1)$, *and assume for all* $m \in \mathbb{N}$, $i \in \{1,2,\dots,\mathfrak{d}\}$, $g_1 = (g_1^{(1)},\dots,g_1^{(\mathfrak{d})}), g_2 = (g_2^{(1)},\dots,g_2^{(\mathfrak{d})}),\dots,g_m = (g_m^{(1)},\dots,g_m^{(\mathfrak{d})}) \in \mathbb{R}^{\mathfrak{d}}$ *that*

$$\psi_m^{(i)}(g_1,g_2,\dots,g_m) = \gamma_m \left[\frac{\sum_{k=1}^m \alpha^{m-k}(1-\alpha)g_k^{(i)}}{1-\alpha^m}\right]\left[\varepsilon + \left[\frac{\sum_{k=1}^m \beta^{m-k}(1-\beta)|g_k^{(i)}|^2}{1-\beta^m}\right]^{1/2}\right]^{-1}. \tag{97}$$

Then there exist $\mathbf{m} = (\mathbf{m}^{(1)},\dots,\mathbf{m}^{(\mathfrak{d})}) : \mathbb{N}_0 \times \Omega \to \mathbb{R}^{\mathfrak{d}}$ *and* $\mathbb{M} = (\mathbb{M}^{(1)},\dots,\mathbb{M}^{(\mathfrak{d})}) : \mathbb{N}_0 \times \Omega \to \mathbb{R}^{\mathfrak{d}}$ *such that for all* $m \in \mathbb{N}$, $i \in \{1,2,\dots,\mathfrak{d}\}$ *it holds that*

$$\mathbf{m}_0 = 0, \qquad \mathbf{m}_m = \alpha\mathbf{m}_{m-1} + (1-\alpha)G_m(\Theta_{m-1}), \tag{98}$$

$$\mathbb{M}_0 = 0, \qquad \mathbb{M}_m^{(i)} = \beta\,\mathbb{M}_{m-1}^{(i)} + (1-\beta)|G_m^{(i)}(\Theta_{m-1})|^2, \tag{99}$$

$$\text{and} \qquad \Theta_m^{(i)} = \Theta_{m-1}^{(i)} - \gamma_m\left[\frac{\mathbf{m}_m^{(i)}}{1-\alpha^m}\right]\left[\varepsilon + \left(\frac{\mathbb{M}_m^{(i)}}{1-\beta^m}\right)^{1/2}\right]^{-1}. \tag{100}$$

7 | NUMERICAL EXAMPLES

In this section we apply the LRV strategy to different parametric approximation problems from the literature. Specifically, we consider the classical parametric Black-Scholes model for the pricing of European call options in Subsection 7.2, we consider a parametric Black-Scholes model for the pricing of worst-of basket put options on three underlying assets in Subsection 7.3, we consider a parametric Black-Scholes model for the pricing of average basket put options on three underlying assets with knock-in barriers in Subsection 7.4, and we consider a parametric stochastic Lorentz equation in Subsection 7.5. In the literature there are already a number of simulation results for SGD-based deep learning methods regarding the high-dimensional pricing of financial derivative contracts. In particular we refer to, for example, Berner et al. (2020), Beck, Becker, Grohs et al. (2021), Lokeshwar et al. (2022), Ferguson and Green (2018), Germain et al. (2021), Biagini et al. (2021) for parametric pricing results for European options, we refer to, for example, Andersson and Oosterlee (2021), Salvador et al. (2020), Becker et al. (2019), Becker et al. (2020), Lind (2022), Chen and Wan (2021), Lapeyre and Lelong (2021), Gaspar et al. (2020), Ye and Zhang (2019), Becker et al. (2021) for the pricing of American options, and we refer, for example, to Ruf and Wang (2020) for further references.

In Subsection 7.1 we briefly recall the antithetic MC method (cf., e.g., Glasserman, 2004) and some well-known properties of it. This is a variance reduction technique for MC methods which we will employ in some of the proposal algorithms for the LRV strategy in case of some of the above mentioned approximation problems. In each of the considered numerical examples we also compare the LRV strategy with existing approximation techniques from the literature such as the deep learning method induced by Beck, Becker, Grohs et al. (2021), MC methods, and QMC methods.

All the simulations in this section were run in PYTHON using TENSORFLOW 2.12 on remote machines from `https://vast.ai` equipped with a single NVIDIA GEFORCE RTX 4090 GPU with 24 GB Graphics RAM. The PYTHON source codes which were employed to produce all the results in this section can be downloaded as part of the sources of the arXiv version of this article at https://arxiv.org/e-print/2202.02717. Specifically the codes in the folder *1_BS1* were employed to produce all the results in Subsection 7.2, the codes in the folder *2_BS_eur_put_basket* were employed to produce all the results in Subsection 7.3, the codes in the folder *3_BS_barrier_put_basket_avg* were employed to produce all the results in Subsection 7.4, and the codes in the folder *4_Lorentz* were employed to produce all the results in Subsection 7.5.

7.1 | Antithetic Monte Carlo approximations

In this section we recall a special case of antithetic variates for MC methods (cf., e.g., Glasserman, 2004, Section 4.2) when the distribution of the sampled random variables is symmetric around the origin. The following result, Lemma 7.1 below, shows that the resulting antithetic MC method achieves a higher or equal L^2-accuracy than the standard MC method when the same number of MC samples are used for both methods. The subsequent result, Corollary 7.2 below, then provides a sufficient condition for the antithetic MC method to achieve a *strictly* higher L^2-accuracy than the standard MC method when the same number of MC samples are used for both methods.

Lemma 7.1. *Let $d, \mathfrak{M} \in \mathbb{N}$, let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, let $X_m : \Omega \to \mathbb{R}^d$, $m \in \{1, 2, \dots, \mathfrak{M}\}$, be i.i.d. random variables, assume for all $B \in \mathcal{B}(\mathbb{R}^d)$ that $\mathbb{P}(X_1 \in B) = \mathbb{P}(-X_1 \in B)$, let $f : \mathbb{R}^d \to \mathbb{R}$ be measurable, assume $\mathbb{E}[|f(X_1)|^2] < \infty$, and let $M : \Omega \to \mathbb{R}$ and $A : \Omega \to \mathbb{R}$ satisfy*

$$M = \left[\frac{1}{\mathfrak{M}} \sum_{m=1}^{\mathfrak{M}} f(X_m)\right] \qquad \textit{and} \qquad A = \frac{1}{2\mathfrak{M}} \left[\sum_{m=1}^{\mathfrak{M}} (f(X_m) + f(-X_m))\right]. \tag{101}$$

Then

(i) it holds that $\mathbb{E}[|M - \mathbb{E}[f(X_1)]|^2] = \frac{\operatorname{Var}(f(X_1))}{\mathfrak{M}}$ *and*

(ii) it holds that $\mathbb{E}[|A - \mathbb{E}[f(X_1)]|^2] = \frac{\operatorname{Var}(f(X_1)) + \operatorname{Cov}(f(X_1), f(-X_1))}{2\mathfrak{M}} \leq \frac{\operatorname{Var}(f(X_1))}{\mathfrak{M}}$.

Proof of Lemma 7.1. First, note that the fact that X_m, $m \in \{1, 2, \dots, \mathfrak{M}\}$, are i.i.d. implies that for all $m, n \in \{1, 2, \dots, \mathfrak{M}\}$ with $m \neq n$ it holds that

$$\operatorname{Cov}(f(X_m), f(X_n)) = 0 \qquad \text{and} \qquad \operatorname{Cov}(f(X_m) + f(-X_m), f(X_n) + f(-X_n)) = 0 \tag{102}$$

(cf., e.g., Klenke, 2014, Theorem 5.4). This, the fact that $\mathbb{E}[M] = \mathbb{E}[f(X_1)]$, and the Bienaymé formula (cf., e.g., Klenke, 2014, Theorem 5.7) assure that

$$\begin{aligned}\mathbb{E}\left[|M - \mathbb{E}[f(X_1)]|^2\right] &= \operatorname{Var}(M) = \frac{1}{\mathfrak{M}^2} \operatorname{Var}\left(\sum_{m=1}^{\mathfrak{M}} f(X_m)\right) \\ &= \frac{1}{\mathfrak{M}^2}\left(\left[\sum_{m=1}^{\mathfrak{M}} \operatorname{Var}(f(X_m))\right] + \sum_{m,n \in \{1,2,\dots,\mathfrak{M}\}, m \neq n} \operatorname{Cov}(f(X_m), f(X_n))\right) \\ &= \frac{1}{\mathfrak{M}^2} \sum_{m=1}^{\mathfrak{M}} \operatorname{Var}(f(X_1)) = \frac{\operatorname{Var}(f(X_1))}{\mathfrak{M}}.\end{aligned} \tag{103}$$

This proves item (i). Next observe that the fact that $X_1, X_2, \dots, X_{\mathfrak{M}}, -X_1, -X_2, \dots, -X_{\mathfrak{M}}$ are identically distributed implies that

$$\mathbb{E}[A] = \frac{1}{2\mathfrak{M}} \sum_{m=1}^{\mathfrak{M}} (\mathbb{E}[f(X_m)] + \mathbb{E}[f(-X_m)]) = \frac{1}{2\mathfrak{M}} \sum_{m=1}^{\mathfrak{M}} 2\mathbb{E}[f(X_1)] = \mathbb{E}[f(X_1)]. \tag{104}$$

This, (102), the fact that $X_1, X_2, \dots, X_{\mathfrak{M}}, -X_1, -X_2, \dots, -X_{\mathfrak{M}}$ are identically distributed, and the Bienaymé formula (cf., e.g., Klenke, 2014, Theorem 5.7) assure that

$$\begin{aligned}
\mathbb{E}\big[|A - \mathbb{E}[f(X_1)]|^2\big] &= \operatorname{Var}(A) = \frac{1}{(2\mathfrak{M})^2} \operatorname{Var}\left(\textstyle\sum_{m=1}^{\mathfrak{M}} f(X_m) + f(-X_m)\right) \\
&= \frac{1}{4\mathfrak{M}^2}\bigg(\Big[\textstyle\sum_{m=1}^{\mathfrak{M}} \operatorname{Var}(f(X_m) + f(-X_m))\Big] \\
&\quad + \Big[\textstyle\sum_{m,n \in \{1,2,\dots,\mathfrak{M}\}, m \neq n} \operatorname{Cov}(f(X_m) + f(-X_m), f(X_n) + f(-X_n))\Big]\bigg) \\
&= \frac{1}{4\mathfrak{M}^2} \textstyle\sum_{m=1}^{\mathfrak{M}} \operatorname{Var}(f(X_1) + f(-X_1)) \\
&= \frac{\operatorname{Var}(f(X_1) + f(-X_1))}{4\mathfrak{M}} \\
&= \frac{\operatorname{Var}(f(X_1)) + 2\operatorname{Cov}(f(X_1), f(-X_1)) + \operatorname{Var}(f(-X_1))}{4\mathfrak{M}} \\
&= \frac{\operatorname{Var}(f(X_1)) + \operatorname{Cov}(f(X_1), f(-X_1))}{2\mathfrak{M}}.
\end{aligned} \tag{105}$$

Moreover, note that the fact that X_1 and $-X_1$ are identically distributed and the Cauchy-Schwarz inequality (cf., e.g., Klenke, 2014, Theorem 5.8) assure that

$$(\operatorname{Cov}(f(X_1), f(-X_1)))^2 \leq \operatorname{Var}(f(X_1)) \operatorname{Var}(f(-X_1)) = (\operatorname{Var}(f(X_1)))^2. \tag{106}$$

Combining this with (105) demonstrates that

$$\begin{aligned}
\mathbb{E}\big[|A - \mathbb{E}[f(X_1)]|^2\big] &= \frac{\operatorname{Var}(f(X_1)) + \operatorname{Cov}(f(X_1), f(-X_1))}{2\mathfrak{M}} \\
&\leq \frac{\operatorname{Var}(f(X_1)) + \operatorname{Var}(f(X_1))}{2\mathfrak{M}} = \frac{\operatorname{Var}(f(X_1))}{\mathfrak{M}}.
\end{aligned} \tag{107}$$

This establishes item (ii). The proof of Lemma 7.1 is thus complete. □

Corollary 7.2. *Let $d, \mathfrak{M} \in \mathbb{N}$, let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, let $X_m : \Omega \to \mathbb{R}^d$, $m \in \{1, 2, \dots, \mathfrak{M}\}$, be i.i.d. random variables, assume for all $B \in \mathcal{B}(\mathbb{R}^d)$ that $\mathbb{P}(X_1 \in B) = \mathbb{P}(-X_1 \in B)$, let $f : \mathbb{R}^d \to \mathbb{R}$ be measurable, assume for all $a, b, c \in \mathbb{R}$ that $\mathbb{E}[|f(X_1)|^2] < \infty$ and $\mathbb{P}(af(X_1) + bf(-X_1) + c = 0) < 1$, and let $M : \Omega \to \mathbb{R}$ and $A : \Omega \to \mathbb{R}$ satisfy*

$$M = \left[\frac{1}{\mathfrak{M}} \sum_{m=1}^{\mathfrak{M}} f(X_m)\right] \quad \text{and} \quad A = \frac{1}{2\mathfrak{M}}\left[\sum_{m=1}^{\mathfrak{M}} (f(X_m) + f(-X_m))\right]. \tag{108}$$

Then

(i) it holds that $\mathbb{E}[|M - \mathbb{E}[f(X_1)]|^2] = \frac{\mathrm{Var}(f(X_1))}{\mathfrak{M}}$ *and*
(ii) it holds that $\mathbb{E}[|A - \mathbb{E}[f(X_1)]|^2] = \frac{\mathrm{Var}(f(X_1)) + \mathrm{Cov}(f(X_1), f(-X_1))}{2\mathfrak{M}} < \frac{\mathrm{Var}(f(X_1))}{\mathfrak{M}}$.

Proof of Corollary 7.2. First, note item (i) in Lemma 7.1 establishes item (i). Moreover, observe that the assumption that for all $a, b, c \in \mathbb{R}$ it holds that $\mathbb{P}(af(X_1) + bf(-X_1) + c = 0) < 1$, the fact that X_1 and $-X_1$ are identically distributed, and the strict Cauchy-Schwarz inequality (cf., e.g., Klenke Klenke, 2014, Theorem 5.8) assure that

$$(\mathrm{Cov}\,(f(X_1), f(-X_1)))^2 < \mathrm{Var}\,(f(X_1))\,\mathrm{Var}\,(f(-X_1)) = (\mathrm{Var}\,(f(X_1)))^2. \tag{109}$$

Combining this with item (ii) in Lemma 7.1 demonstrates that

$$\begin{aligned} \mathbb{E}\big[|A - \mathbb{E}[f(X_1)]|^2\big] &= \frac{\mathrm{Var}\,(f(X_1)) + \mathrm{Cov}\,(f(X_1), f(-X_1))}{2\mathfrak{M}} \\ &< \frac{\mathrm{Var}\,(f(X_1)) + \mathrm{Var}\,(f(X_1))}{2\mathfrak{M}} = \frac{\mathrm{Var}\,(f(X_1))}{\mathfrak{M}}. \end{aligned} \tag{110}$$

This establishes item (ii). The proof of Corollary 7.2 is thus complete. □

7.2 | Parametric Black-Scholes partial differential equations for European call options

In this section we apply the LRV strategy to the problem of approximating the fair price of an European call option in the classical Black-Scholes model (cf. Black & Scholes, 1973 and Merton, 1973). A brief summary of the numerical results of this subsection can be found in Table 1 in the introduction. We start by introducing the Black-Scholes model in the context of Framework 6.1.

Assume Framework 6.1, assume $\mathfrak{p} = 5$ and

$$\mathfrak{P} = [90, 110] \times [0.01, 1] \times [-0.1, 0.1] \times [0.01, 0.5] \times [90, 110], \tag{111}$$

let $\mathfrak{N} : \mathbb{R} \to \mathbb{R}$ satisfy for all $z \in \mathbb{R}$ that $\mathfrak{N}(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} \exp(-\frac{y^2}{2})\,\mathrm{d}y$, and let $u : \mathfrak{P} \to \mathbb{R}$ satisfy for all $p = (\xi, T, r, \sigma, K) \in \mathfrak{P}$ that

$$u(p) = \xi\,\mathfrak{N}\left(\frac{\left(r + \frac{\sigma^2}{2}\right)T + \ln(\xi/K)}{\sigma\sqrt{T}}\right) - \exp(-rT)K\,\mathfrak{N}\left(\frac{\left(r - \frac{\sigma^2}{2}\right)T + \ln(\xi/K)}{\sigma\sqrt{T}}\right). \tag{112}$$

Note that (112) corresponds to the famous Black-Scholes formula for European call options. In the economic interpretation of the Black-Scholes model, for every $p = (\xi, T, r, \sigma, K) \in \mathfrak{P}$ the number $u(p) \in \mathbb{R}$ thus corresponds to the fair price of a European call option with initial price ξ, time of maturity T, drift rate r, volatility σ, and strike price K.

We now specify the mathematical objects in the LRV strategy appearing in Framework 6.1 to approximately calculate the target function $u : \mathfrak{P} \to \mathbb{R}$ in (112). Specifically, in addition to the assumptions above, let $\mathfrak{M}, \mathscr{M} \in \mathbb{N}$, $a, e \in \{0, 1\}$, assume $\mathfrak{d} = \mathfrak{M}$, $\mathbf{d} = \mathscr{M}$, $k = 1$, for every $\xi, r, \sigma, w \in \mathbb{R}$, $T \in [0, \infty)$ let $X_T^{r,\sigma,\xi,w} \in \mathbb{R}$ satisfy

$$X_T^{r,\sigma,\xi,w} = \xi \exp\left(\left(r - \frac{\sigma^2}{2}\right)T + \sigma\sqrt{T}w\right), \tag{113}$$

let $\phi_{\mathfrak{a}} : \mathfrak{P} \times \mathbb{R}^{\mathbf{d}} \to \mathbb{R}$, $\mathfrak{a} \in \{0, 1\}$, satisfy for all $p = (\xi, T, r, \sigma, K) \in \mathfrak{P}$, $w \in \mathbb{R}$ that

$$\phi_0(p, w) = \exp(-rT) \max\left\{X_T^{r,\sigma,\xi,w} - K, 0\right\} \qquad \text{and} \tag{114}$$

$$\phi_1(p, w) = \frac{\phi_0(p,w)+\phi_0(p,-w)}{2} = \frac{\exp(-rT)}{2}\left(\max\left\{X_T^{r,\sigma,\xi,w} - K, 0\right\} + \max\left\{X_T^{r,\sigma,\xi,-w} - K, 0\right\}\right), \tag{115}$$

assume for all $p = (\xi, T, r, \sigma, K) \in \mathfrak{P}$, $w = (w_1, w_2, \dots, w_{\mathscr{M}}) \in \mathbb{R}^{\mathscr{M}}$, $\theta = (\theta_1, \dots, \theta_{\mathfrak{M}}) \in \mathbb{R}^{\mathfrak{M}}$ that

$$\Psi(p, \theta) = \frac{1}{\mathfrak{M}}\left[\sum_{\mathrm{m}=1}^{\mathfrak{M}} \phi_a(p, \theta_{\mathrm{m}})\right] \qquad \text{and} \qquad \Xi(p, w) = \frac{\mathbb{1}_{\{0\}}(e)}{\mathscr{M}}\left[\sum_{\mathrm{m}=1}^{\mathscr{M}} \phi_1(p, w_{\mathrm{m}})\right] + \mathbb{1}_{\{1\}}(e)u(p), \tag{116}$$

assume for all $x, y \in \mathbb{R}$ that $\mathfrak{C}(x, y) = |x - y|^2$, let $\mathbf{a} = \frac{9}{10}$, $\mathbf{b} = \frac{999}{1000}$, $\varepsilon \in (0, \infty)$, let $(\gamma_m)_{m\in\mathbb{N}} \subseteq (0, \infty)$ satisfy for all $j \in \{1, 2, \dots, 7\}$, $m \in \mathbb{N} \cap (20000(j - 1), 20000j]$ that $\gamma_m = 10^{-j}$, assume for all $m \in \mathbb{N}$, $i \in \{1, 2, \dots, \mathfrak{d}\}$, $g_1 = (g_1^{(1)}, \dots, g_1^{(\mathfrak{d})})$, $g_2 = (g_2^{(1)}, \dots, g_2^{(\mathfrak{d})})$, $\dots$, $g_m = (g_m^{(1)}, \dots, g_m^{(\mathfrak{d})}) \in \mathbb{R}^{\mathfrak{d}}$ that

$$\psi_m^{(i)}(g_1, g_2, \dots, g_m) = \gamma_m\left[\frac{\sum_{k=1}^m \mathbf{a}^{m-k}(1 - \mathbf{a})g_k^{(i)}}{1 - \mathbf{a}^m}\right]\left[\varepsilon + \left[\frac{\sum_{k=1}^m \mathbf{b}^{m-k}(1 - \mathbf{b})|g_k^{(i)}|^2}{1 - \mathbf{b}^m}\right]^{1/2}\right]^{-1}, \tag{117}$$

assume that $\mathfrak{W}$ is a standard normal random vector, assume that $P_{1,1}$ is $\mathcal{U}_{\mathfrak{P}}$-distributed, and assume that $W^{1,1}$ is a standard normal random variable.

Let us add some comments regarding the setup introduced above. Observe that (112), (114), (115), and Lemma 2.1 assure that for all $p = (\xi, T, r, \sigma, K) \in \mathfrak{P}$ it holds that

$$u(p) = \mathbb{E}\left[\exp(-rT) \max\left\{X_T^{r,\sigma,\xi,W^{1,1}} - K, 0\right\}\right] = \mathbb{E}\left[\phi_0(p, W^{1,1})\right] = \mathbb{E}\left[\phi_1(p, W^{1,1})\right]. \tag{118}$$

Moreover, note that in the case $a = 0$ the proposal algorithm on the left hand side of (116) corresponds to the standard MC method with $\mathfrak{M}$ samples and that in the case $a = 1$ the proposal algorithm on the left hand side of (116) corresponds to the antithetic MC method with $\mathfrak{M}$ samples (cf. Subsection 7.1). In addition, observe that in the case $e = 0$ the reference solutions on the right hand side of (116) correspond to antithetic MC approximations with $\mathscr{M}$ samples and that in the case $e = 1$ the reference solutions on the right hand side of (116) are given by the exact solution. Furthermore, observe that (117) describes the Adam optimizer in the setup of Framework 6.1 (cf. Kingma & Ba, 2014 and Subsection 6.5.7).

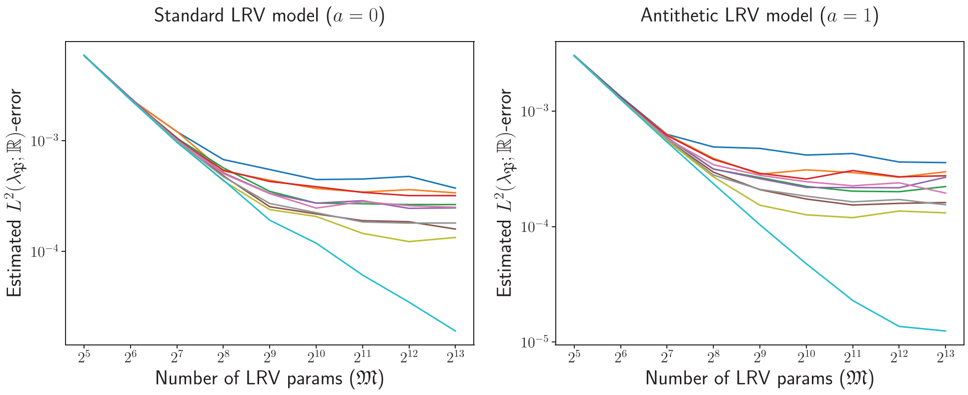

FIGURE 1 Numerical simulations for the LRV strategy in case of the Black-Scholes model for European call options on one underlying described in Subsection 7.2 (5-dimensional approximation problem). See Figure 2 below for the legend. [Color figure can be viewed at wileyonlinelibrary.com]

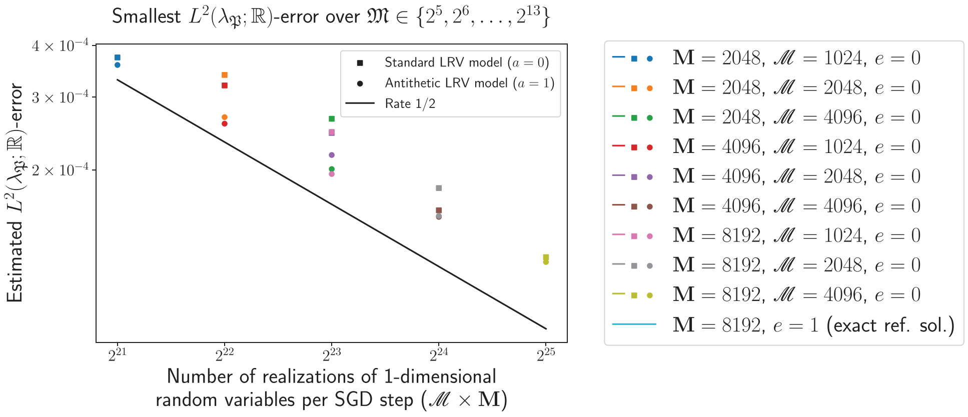

FIGURE 2 Smallest estimated $L^2(\lambda_{\mathfrak{P}}; \mathbb{R})$-errors over $\mathfrak{M}$ for different choices of training parameters. Each dot corresponds to the minimum of a line of the same color in Figure 1. The convergence rate 1/2 is inspired by the convergence rate 1/2 of the MC method which is strongly related to SGD. [Color figure can be viewed at wileyonlinelibrary.com]

In Figures 1 and 2 and Tables 2 and 6 we approximately present for different choices of $\mathfrak{M} \in \{2^5, 2^6, \ldots, 2^{13}\}$, $\mathscr{M} \in \{2^{10}, 2^{11}, 2^{12}\}$, $\mathbf{M} \in \{2^{11}, 2^{12}, 2^{13}\}$, $a, e \in \{0, 1\}$ random realizations of the $L^1(\lambda_{\mathfrak{P}}; \mathbb{R})$-approximation error

$$\int_{\mathfrak{P}} |u(p) - \Psi(p, \Theta_{140000})| \, \mathrm{d}p \tag{119}$$

(3rd column in Tables 2 and 6), random realizations of the $L^2(\lambda_{\mathfrak{P}}; \mathbb{R})$-approximation error

$$\left[\int_{\mathfrak{P}} |u(p) - \Psi(p, \Theta_{140000})|^2 \, \mathrm{d}p \right]^{1/2} \tag{120}$$

TABLE 6 Numerical simulations for the LRV strategy in case of the Black-Scholes model for European call options on one underlying described in Subsection 7.2 (5-dimensional approximation problem) trained with batch size $\mathbf{M} = 8192$ and exact ($e = 1$) reference solutions (cf. bright blue lines in Figures 1 and 2).

$\mathfrak{M}$	a	Number of trainable parameters	L^1-approx. error	L^2-approx. error	L^∞-approx. error	Training time in seconds	Evaluation time for 8 192 000 evaluations in seconds
32	0	32	0.003186	0.005893	0.065741	149.13	0.58
32	1	32	0.001502	0.003013	0.028984	153.95	0.62
64	0	64	0.001211	0.002352	0.023104	148.42	0.64
64	1	64	0.000635	0.001262	0.012718	156.98	0.61
128	0	128	0.000482	0.000971	0.009106	152.34	0.60
128	1	128	0.000273	0.000541	0.005257	152.41	0.60
256	0	256	0.000206	0.000441	0.007746	154.84	0.61
256	1	256	0.000118	0.000235	0.002228	170.88	0.68
512	0	512	0.000089	0.000192	0.003382	163.79	0.66
512	1	512	0.000052	0.000104	0.000969	179.32	0.67
1024	0	1024	0.000049	0.000119	0.003899	184.64	0.71
1024	1	1024	0.000024	0.000048	0.000420	305.44	0.91
2048	0	2048	0.000027	0.000061	0.002447	304.91	0.87
2048	1	2048	0.000012	0.000023	0.000206	656.63	2.36
4096	0	4096	0.000015	0.000035	0.001529	652.68	2.34
4096	1	4096	0.000008	0.000014	0.000122	1199.47	4.20
8192	0	8192	0.000010	0.000019	0.000862	1187.81	4.18
8192	1	8192	0.000007	0.000012	0.000145	2201.75	7.84

(Figures 1 and 2 and 4th column in Tables 2 and 6), random realizations of the $L^\infty(\lambda_{\mathfrak{P}};\mathbb{R})$-approximation error

$$\sup_{p\in\mathfrak{P}} |u(p) - \Psi(p,\Theta_{140000})|, \tag{121}$$

(5th column in Tables 2 and 6), the time to compute Θ_{140000} (6th column in Tables 2 and 6), and the time to compute 8 192 000 evaluations of the function $\mathfrak{P} \ni p \mapsto \Psi(p,\Theta_{140000}) \in \mathbb{R}$ (7th column in Tables 2 and 6). We approximated the integrals in (119) and (120) with the MC method based on 8 192 000 samples and we approximated the suprema in (121) based on 8 192 000 random samples (cf., e.g., Beck, Becker, Grohs et al., 2021, Lemma 3.5 and Beck et al., 2022, Section 3.3).

To compare the LRV strategy with existing approximation techniques from the literature, we also employ several other methods to approximate the function $u : \mathfrak{P} \to \mathbb{R}$ in (112). Specifically, in Table 3 we present numerical simulations for the deep learning method induced by Beck, Becker, Grohs et al. (2021) (with training values given by the exact solution, Adam 140000 training steps, batch size 8192, learning rate schedule $\mathbb{N} \ni j \mapsto \mathbb{1}_{(0,20000]}(m)10^{-2} + \mathbb{1}_{(20000,50000]}(m)10^{-3} + \mathbb{1}_{(50000,80000]}(m)10^{-4} + \mathbb{1}_{(80000,100000]}(m)10^{-5} + \mathbb{1}_{(100000,120000]}(m)10^{-6} + \mathbb{1}_{(120000,140000]}(m)10^{-7}$, and GELU activation function), in Table 4 we present numerical simulations for the standard and the antithetic MC method, and in Table 5 we present numerical simulations for the standard and

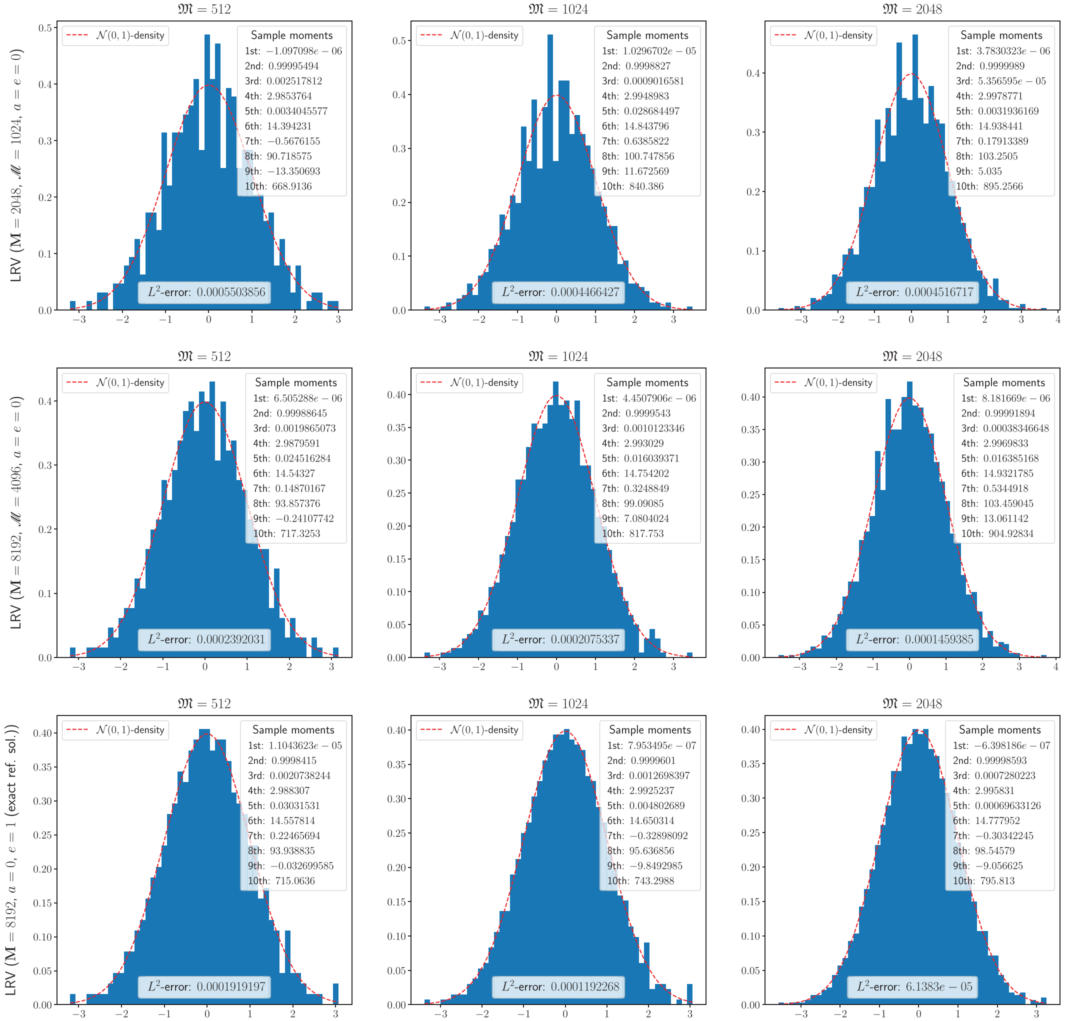

FIGURE 3 Histograms and sample moments of realizations of *learned random variables* $\Theta^{(1)}_{140000}, \Theta^{(2)}_{140000}, \dots, \Theta^{(\mathfrak{M})}_{140000}$ in the LRV strategy in case of the Black-Scholes model for European call options on one underlying described in Subsection 7.2. See Figure 4 for the moments of the standard normal distribution. [Color figure can be viewed at wileyonlinelibrary.com]

the antithetic QMC method with Sobol sequences. In Tables 3, 4, and 5 we have approximated the $L^1(\lambda_{\mathfrak{P}}; \mathbb{R})$-approximation errors of the respective approximation methods with the MC method based on 8 192 000 samples, we have approximated the $L^2(\lambda_{\mathfrak{P}}; \mathbb{R})$-approximation errors of the respective approximation methods with the MC method based on 8 192 000 samples, and we have approximated the $L^\infty(\lambda_{\mathfrak{P}}; \mathbb{R})$-approximation errors of the respective approximation methods based on 8 192 000 random samples (cf., e.g., Beck, Becker, Grohs et al., 2021, Lemma 3.5 and Beck et al., 2022, Section 3.3).

Next we discuss the empirical distributions of the random variables learned by the LRV methodology and compare them to empirical distributions of MC and QMC samples. Specifically, in Figure 3 we visualize for $\mathfrak{M} \in \{2^9, 2^{10}, 2^{11}\}$, $a = 0$ realizations of empirical distributions of *learned random variables*$\Theta^{(1)}_{140000}, \Theta^{(2)}_{140000}, \dots, \Theta^{(\mathfrak{M})}_{140000}$ in the case of three different kinds of training procedures and in Figure 4 we visualize realizations of empirical distributions of random variables in the MC method and in QMC method based on Sobol sequences. Very roughly speaking, it seems that with increasingly precise reference solutions and thereby smaller L^2-errors, the LRV method produces learned random variables whose histograms approximate the density of the normal distribution more closely, in particular more closely than the realizations of the standard MC samples with which the LRV method is initialized. This suggests that the LRV strategy is learning random variables which in some sense try to approximate the normal distribution. On the other hand, we note that the QMC samples seem to have the most regular histograms, but still have much worse L^2-errors when compared to the learned random variables. One explanation for this could be that even though the histograms of the QMC method seem to approximate the normal density very accurately, the empirical moments of the QMC samples are a worse approximation of the moments of the normal distribution than the empirical moments of the learned random variables, and so the QMC samples effectively do not approximate the normal distribution as well as the learned random variables.

7.3 | Parametric Black-Scholes partial differential equations for multi-asset worst-of basket put options

In this section we apply the LRV strategy for the approximation of the price of European worst-of basket put options in the Black-Scholes option pricing model (cf. Black & Scholes, 1973 and Merton, 1973). We start by introducing the Black-Scholes model for this pricing problem in the context of Framework 6.1.

Assume Framework 6.1, let $d = 3$, $e_1 = (1,0,0)$, $e_2 = (0,1,0)$, $e_3 = (0,0,1)$, let $Q : \mathbb{R}^d \to \mathbb{R}^{d\times d}$ satisfy for all $\rho = (\rho_1, \rho_2, \rho_3) \in \mathbb{R}^d$ that

$$Q(\rho) = \begin{pmatrix} 1 & \rho_1 & \rho_2 \\ \rho_1 & 1 & \rho_3 \\ \rho_2 & \rho_3 & 1 \end{pmatrix}, \tag{122}$$

let $R \subseteq \mathbb{R}^d$ satisfy $R = \{\rho = (\rho_1, \rho_2, \rho_3) \in [-\frac{95}{100}, \frac{95}{100}]^d : 1 - |\rho_2|^2 - \frac{(\rho_3 - \rho_1\rho_2)^2}{(1-|\rho_1|^2)^{1/2}} \geq 0\}$, assume $\mathfrak{p} = 15$ and

$$\mathfrak{P} = [90, 110]^3 \times \left[\frac{1}{100}, 1\right] \times \left[-\frac{1}{20}, \frac{1}{20}\right] \times \left[0, \frac{1}{10}\right]^3 \times \left[\frac{1}{100}, \frac{1}{2}\right]^3 \times R \times [90, 110], \tag{123}$$

let $L : R \to \mathbb{R}^{d\times d}$ satisfy for all $\rho = (\rho_1, \rho_2, \rho_3) \in R$ that

$$L(\rho) = \begin{pmatrix} 1 & 0 & 0 \\ \rho_1 & (1 - |\rho_1|^2)^{1/2} & 0 \\ \rho_2 & \frac{\rho_3 - \rho_1\rho_2}{(1-|\rho_1|^2)^{1/2}} & \left(1 - |\rho_2|^2 - \frac{(\rho_3-\rho_1\rho_2)^2}{1-|\rho_1|^2}\right)^{1/2} \end{pmatrix}, \tag{124}$$

for every $\xi = (\xi_1, \xi_2, \xi_3) \in \mathbb{R}^d$, $T \in [0, \infty)$, $r \in \mathbb{R}$, $\delta = (\delta_1, \delta_2, \delta_3) \in \mathbb{R}^d$, $\sigma = (\sigma_1, \sigma_2, \sigma_3) \in \mathbb{R}^d$, $\rho \in R$, $w \in \mathbb{R}^d$ let $X_T^{\xi,r,\delta,\sigma,\rho,w} = (X_T^{r,\xi,\delta,\sigma,\rho,w,1}, X_T^{r,\xi,\delta,\sigma,\rho,w,2}, X_T^{r,\xi,\delta,\sigma,\rho,w,3}) \in \mathbb{R}^d$ satisfy[3] for all $i \in \{1,2,3\}$

[3] Note that for all $n \in \mathbb{N}$, $v = (v_1, \dots, v_n)$, $w = (w_1, \dots, w_n) \in \mathbb{R}^n$ it holds that $\langle v, w \rangle = \sum_{i=1}^n v_i w_i$.

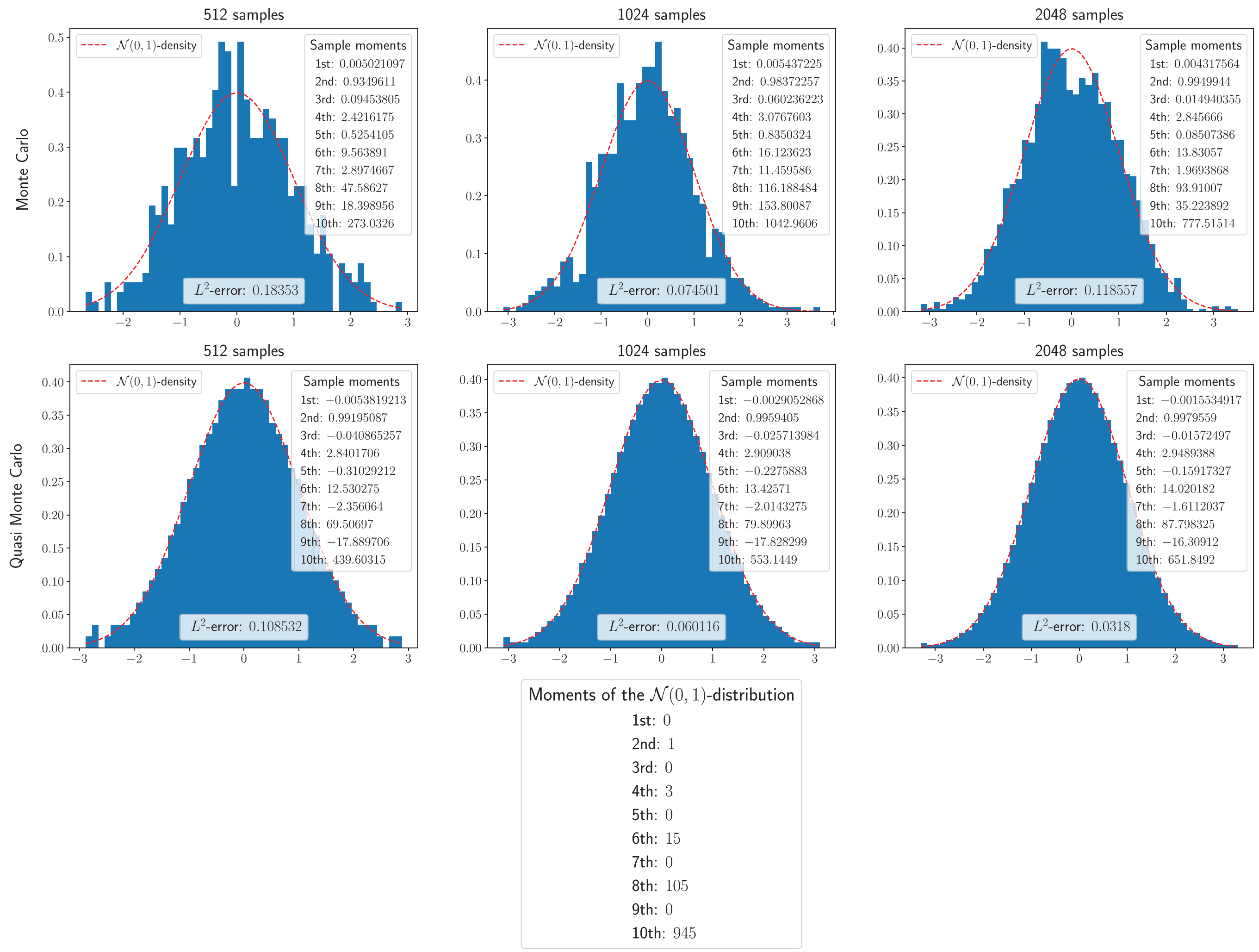

FIGURE 4 Histograms and sample moments of realizations of random variables appearing in the MC method and QMC method in case of the Black-Scholes model for European call options on one underlying described in Subsection 7.2. [Color figure can be viewed at wileyonlinelibrary.com]

that

$$X_T^{\xi,r,\delta,\sigma,\rho,w,i} = \xi_i \exp\left(\left[r - \delta_i - \tfrac{|\sigma_i|^2}{2}\right]T + \sqrt{T}\sigma_i\langle L(\rho)w, e_i\rangle\right), \tag{125}$$

let $\mathbf{l} : \mathbb{R}^d \to \mathbb{R}$ satisfy for all $x = (x_1, x_2, x_3) \in \mathbb{R}^d$ that $\mathbf{l}(x) = \min\{x_1, x_2, x_3\}$, let $\phi : \mathfrak{P} \times \mathbb{R}^d \to \mathbb{R}$ satisfy for all $p = (\xi, T, r, \delta, \sigma, \rho, K) \in \mathfrak{P}$, $w \in \mathbb{R}^d$ that

$$\phi(p, w) = \exp(-rT) \max\left\{K - \mathbf{l}\left(X_T^{\xi,r,\delta,\sigma,\rho,w}\right), 0\right\}, \tag{126}$$

let $\mathcal{W} : \Omega \to \mathbb{R}^d$ be a standard normal random vector, and let $u : \mathfrak{P} \to \mathbb{R}$ satisfy for all $p \in \mathfrak{P}$ that

$$u(p) = \mathbb{E}[\phi(p, \mathcal{W})]. \tag{127}$$

In the economic interpretation of the Black-Scholes model for every $p = (\xi, T, r, \delta, \sigma, \rho, K) \in \mathfrak{P}$ the number $u(p) \in \mathbb{R}$ corresponds to the fair price of a worst-of basekt put option on three underlying assets with initial prices ξ, time of maturity T, risk free rate r, dividend yields of the respective underlying assets δ, volatilities of the respective underlying assets σ, covariance matrix of the Brownian motions $Q(\rho)$, and strike price K.

We now specify the mathematical objects in the LRV strategy appearing in Framework 6.1 to approximately calculate the target function $u : \mathfrak{P} \to \mathbb{R}$ in (127). Specifically, in addition to the assumptions above, let $\mathfrak{M} \in \mathbb{N}$, $\mathscr{M} = 8192$, assume $\mathfrak{d} = \mathfrak{M}d$, $\mathbf{d} = \mathscr{M}d$, $k = 1$, $\mathbf{M} = 4096$, assume for all $p = (\xi, T, r, \delta, \sigma, \rho, K) \in \mathfrak{P}$, $w = (w_1, \dots, w_{\mathscr{M}d}) \in \mathbb{R}^{\mathscr{M}d}$, $\theta = (\theta_1, \dots, \theta_{\mathfrak{M}d}) \in \mathbb{R}^{\mathfrak{M}d}$ that

$$\Xi(p, w) = \frac{1}{\mathscr{M}}\left[\sum_{\mathrm{m}=1}^{\mathscr{M}} \phi\big(p, (w_{(\mathrm{m}-1)d+k})_{k \in \{1,2,3\}}\big)\right] \text{ and } \Psi(p, \theta) = \frac{1}{\mathfrak{M}}\left[\sum_{\mathrm{m}=1}^{\mathfrak{M}} \phi\big(p, (\theta_{(\mathrm{m}-1)d+k})_{k \in \{1,2,3\}}\big)\right], \tag{128}$$

assume for all $x, y \in \mathbb{R}$ that $\mathfrak{C}(x, y) = |x - y|^2$, let $\mathbf{a} = \frac{9}{10}$, $\mathbf{b} = \frac{999}{1000}$, $\varepsilon \in (0, \infty)$, let $(\gamma_m)_{m \in \mathbb{N}} \subseteq (0, \infty)$ satisfy for all $m \in \{1, 2, \dots, 30000\}$ that $\gamma_m = \mathbb{1}_{(0,15000]}(m)10^{-3} + \mathbb{1}_{(15000,25000]}(m)10^{-4} + \mathbb{1}_{(25000,30000]}(m)10^{-5}$, assume for all $m \in \mathbb{N}$, $i \in \{1, 2, \dots, \mathfrak{d}\}$, $g_1 = (g_1^{(1)}, \dots, g_1^{(\mathfrak{d})})$, $g_2 = (g_2^{(1)}, \dots, g_2^{(\mathfrak{d})})$, $\dots$, $g_m = (g_m^{(1)}, \dots, g_m^{(\mathfrak{d})}) \in \mathbb{R}^{\mathfrak{d}}$ that

$$\psi_m^{(i)}(g_1, g_2, \dots, g_m) = \gamma_m \left[\frac{\sum_{k=1}^m \mathbf{a}^{m-k}(1 - \mathbf{a})g_k^{(i)}}{1 - \mathbf{a}^m}\right]\left[\varepsilon + \left[\frac{\sum_{k=1}^m \mathbf{b}^{m-k}(1 - \mathbf{b})|g_k^{(i)}|^2}{1 - \mathbf{b}^m}\right]^{1/2}\right]^{-1}, \tag{129}$$

assume that $\mathfrak{W}$ is a standard normal random vector, assume that $P_{1,1}$ is $\mathcal{U}_{\mathfrak{P}}$-distributed, and assume that $W^{1,1}$ is a standard normal random vector.

Let us add some comments regarding the setup introduced above. Note that the proposal algorithm in (127) corresponds to the MC method. Furthermore, observe that (129) describes the Adam optimizer in the setup of Framework 6.1 (cf. Kingma & Ba, 2014 and Subsection 6.5.7).

In Table 7 we approximately present for $\mathfrak{M} \in \{2^5, 2^6, \dots, 2^{13}\}$ one random realization of the $L^1(\lambda_{\mathfrak{P}}; \mathbb{R})$-approximation error

$$\textstyle\int_{\mathfrak{P}} |u(p) - \Psi(p, \Theta_{30000})| \,\mathrm{d}p \tag{130}$$

TABLE 7 Numerical simulations for the LRV strategy in case of the Black-Scholes model for European worst-of basket put options on three underlyings described in Subsection 7.3 (15-dimensional approximation problem).

$\mathfrak{M}$	Number of trainable parameters	L^1-approx. error	L^2-approx. error	L^∞-approx. error	Training time in seconds
32	96	0.072564	0.097067	0.693916	311.69
64	192	0.035654	0.047869	0.335918	313.22
128	384	0.018049	0.024193	0.169165	314.48
256	768	0.009603	0.012866	0.089918	318.79
512	1536	0.005317	0.007125	0.064743	337.47
1024	3072	0.003129	0.004171	0.029337	369.79
2048	6144	0.002132	0.002827	0.021137	443.43
4096	12288	0.001594	0.002108	0.014027	594.14
8192	24576	0.001242	0.001642	0.011484	900.10

(3rd column in Table 7), one random realization of the $L^2(\lambda_{\mathfrak{P}};\mathbb{R})$-approximation error

$$\left[\int_{\mathfrak{P}} |u(p) - \Psi(p,\Theta_{30000})|^2\,\mathrm{d}p\right]^{1/2} \tag{131}$$

(4th column in Table 7), one random realization of the $L^\infty(\lambda_{\mathfrak{P}};\mathbb{R})$-approximation error

$$\sup\nolimits_{p\in\mathfrak{P}} |u(p) - \Psi(p,\Theta_{30000})|, \tag{132}$$

(5th column in Table 7), and the time to compute Θ_{30000} (6th column in Table 7). For every $\mathfrak{M} \in \{2^5, 2^6, \dots, 2^{13}\}$ we approximated the integrals in (130) and (131) with the MC method based on 128000 samples and we approximated the supremum in (132) based on 128000 random samples (cf., e.g., Beck, Becker, Grohs et al., 2021, Lemma 3.5 and Beck et al., 2022, Section 3.3). In our approximations of (130), (131), and (132) we have approximately computed for all required sample points $p \in \mathfrak{P}$ the value $u(p)$ of the unknown exact solution by means of an MC approximation with 209715200 MC samples.

Besides the LRV strategy we also employed the standard MC method to approximate the function $u : \mathfrak{P} \to \mathbb{R}$ in (127). In Table 8 we present the corresponding numerical simulation results. In Table 8 we have approximated the $L^1(\lambda_{\mathfrak{P}};\mathbb{R})$-approximation error of the MC method with the MC method based on 128000 samples, we have approximated the $L^2(\lambda_{\mathfrak{P}};\mathbb{R})$-approximation errors of the MC method with the MC method based on 128000 samples, and we have approximated the $L^\infty(\lambda_{\mathfrak{P}};\mathbb{R})$-approximation errors of the MC method based on 128000 random samples (cf., e.g., Beck, Becker, Grohs et al., 2021, Lemma 3.5 and Beck et al., 2022, Section 3.3). In our approximations of the above mentioned approximation errors we have approximately computed for all required sample points $p \in \mathfrak{P}$ the value $u(p)$ of the unknown exact solution by means of an MC approximation with 209715200 MC samples.

TABLE 8 Numerical simulations for the standard MC method in case of the Black-Scholes model for European worst-of basket put options on three underlyings described in Subsection 7.3 (15-dimensional approximation problem).

Number of MC samples	Number of scalar random variables per evaluation	L^1-approx. error	L^2-approx. error	L^∞-approx. error
32	96	1.543287	2.080271	12.685066
64	192	1.091341	1.470045	9.742924
128	384	0.770266	1.038834	6.845669
256	768	0.544692	0.736095	4.272978
512	1536	0.386441	0.521668	2.919704
1024	3072	0.272983	0.369065	2.299841
2048	6144	0.192779	0.260462	1.656979
4096	12288	0.136122	0.183617	1.268164
8192	24576	0.096641	0.130191	0.825016
16384	49152	0.068338	0.092281	0.577732
32768	98304	0.048270	0.065142	0.452351

7.4 | Parametric Black-Scholes partial differential equations for multi-asset average put options with knock-in barriers

In this section we apply the LRV strategy to a more complicated and practically relevant Black-Scholes option pricing problem. Specifically, we approximate the fair price of a European average basket put option on three underlyings with a knock-in barrier in the Black-Scholes model (cf. Black & Scholes, 1973 and Merton, 1973). We start by introducing the Black-Scholes model for this pricing problem in the context of Framework 6.1.

Assume Framework 6.1, let $d = 3$, $e_1 = (1,0,0)$, $e_2 = (0,1,0)$, $e_3 = (0,0,1)$, let $Q : \mathbb{R}^d \to \mathbb{R}^{d \times d}$ satisfy for all $\rho = (\rho_1, \rho_2, \rho_3) \in \mathbb{R}^d$ that

$$Q(\rho) = \begin{pmatrix} 1 & \rho_1 & \rho_2 \\ \rho_1 & 1 & \rho_3 \\ \rho_2 & \rho_3 & 1 \end{pmatrix}, \tag{133}$$

let $R \subseteq \mathbb{R}^d$ satisfy $R = \{\rho = (\rho_1, \rho_2, \rho_3) \in [-\frac{95}{100}, \frac{95}{100}]^d : 1 - |\rho_2|^2 - \frac{(\rho_3 - \rho_1\rho_2)^2}{(1-|\rho_1|^2)^{1/2}} \geq 0\}$, assume $\mathfrak{p} = 16$ and

$$\mathfrak{P} = [90, 110]^3 \times \left[\tfrac{1}{2}, 1\right] \times \left[-\tfrac{1}{20}, \tfrac{1}{20}\right] \times \left[0, \tfrac{1}{10}\right]^3 \times \left[\tfrac{1}{100}, \tfrac{1}{2}\right]^3 \times R \times [90, 110] \times [70, 80], \tag{134}$$

let $L : R \to \mathbb{R}^{d \times d}$ satisfy for all $\rho = (\rho_1, \rho_2, \rho_3) \in R$ that

$$L(\rho) = \begin{pmatrix} 1 & 0 & 0 \\ \rho_1 & (1 - |\rho_1|^2)^{1/2} & 0 \\ \rho_2 & \frac{\rho_3 - \rho_1\rho_2}{(1-|\rho_1|^2)^{1/2}} & \left(1 - |\rho_2|^2 - \frac{(\rho_3 - \rho_1\rho_2)^2}{1 - |\rho_1|^2}\right)^{1/2} \end{pmatrix}, \tag{135}$$

let $\mathcal{W} : \Omega \to C([0,\infty),\mathbb{R}^d)$ be a standard Brownian motion with continuous sample paths, for every $\xi = (\xi_1,\xi_2,\xi_3) \in \mathbb{R}^d$, $r \in \mathbb{R}$, $\delta = (\delta_1,\delta_2,\delta_3) \in \mathbb{R}^d$, $\sigma = (\sigma_1,\sigma_2,\sigma_3) \in \mathbb{R}^d$, $\rho \in \mathbb{R}^d$ let $X^{\xi,r,\delta,\sigma,\rho} = (X^{r,\xi,\delta,\sigma,\rho,1}, X^{r,\xi,\delta,\sigma,\rho,2}, X^{r,\xi,\delta,\sigma,\rho,3}) : C([0,\infty),\mathbb{R}^d) \to C([0,\infty),\mathbb{R}^d)$ satisfy for all $w \in C([0,\infty),\mathbb{R}^d)$, $t \in [0,\infty)$, $i \in \{1,2,3\}$ that

$$X_t^{\xi,r,\delta,\sigma,\rho,i}(w) = \xi_i \exp\left(\left[r - \delta_i - \tfrac{|\sigma_i|^2}{2}\right]t + \sigma_i \langle L(\rho)w_t, e_i\rangle\right), \tag{136}$$

let $\mathbf{l} : \mathbb{R}^d \to \mathbb{R}$ and $\mathbf{m} : \mathbb{R}^d \to \mathbb{R}$ satisfy for all $x = (x_1,x_2,x_3) \in \mathbb{R}^d$ that $\mathbf{l}(x) = \min\{x_1,x_2,x_3\}$ and $\mathbf{m}(x) = \frac{x_1+x_2+x_3}{3}$, for every $T \in [0,\infty)$, $B \in \mathbb{R}$ let $\mathfrak{B}_{T,B} \subseteq C([0,\infty),\mathbb{R}^d)$ satisfy

$$\mathfrak{B}_{T,B} = \left\{w \in C([0,\infty),\mathbb{R}^d) : (\exists\, t \in [0,T] : \mathbf{l}(w_t) < B)\right\}, \tag{137}$$

let $\phi : \mathfrak{P} \times C([0,\infty),\mathbb{R}^d) \to \mathbb{R}$ satisfy for all $p = (\xi,T,r,\delta,\sigma,\rho,K,B) \in \mathfrak{P}$, $w \in C([0,\infty),\mathbb{R}^d)$ that

$$\phi(p,w) = \mathbb{1}_{\mathfrak{B}_{T,B}}(X^{\xi,r,\delta,\sigma,\rho}(w)) \exp(-rT) \max\left\{K - \mathbf{m}(X_T^{\xi,r,\delta,\sigma,\rho}(w)), 0\right\}, \tag{138}$$

and let $u : \mathfrak{P} \to \mathbb{R}$ satisfy for all $p \in \mathfrak{P}$ that

$$u(p) = \mathbb{E}[\phi(p,\mathcal{W})]. \tag{139}$$

In the economic interpretation of the Black-Scholes model for every $p = (\xi,T,r,\delta,\sigma,\rho,K,B) \in \mathfrak{P}$ the number $u(p) \in \mathbb{R}$ corresponds to the fair price of a average basekt put option on three underlying assets with initial prices ξ, time of maturity T, risk free rate r, dividend yields of the respective underlying assets δ, volatilities of the respective underlying assets σ, covariance matrix of the Brownian motions $Q(\rho)$, strike price K, and knock-in barrier B.

We now specify the mathematical objects in the LRV strategy appearing in Framework 6.1 to approximately calculate the target function $u : \mathfrak{P} \to \mathbb{R}$ in (139). Specifically, in addition to the assumptions above, let $N = 10$, $\mathcal{M} = 1024$, $\mathfrak{M} \in \mathbb{N}$, assume $\mathfrak{d} = \mathfrak{M}Nd$, $\mathbf{d} = \mathcal{M}Nd$, $k = 1$, $\mathbf{M} = 1024$, for every $p = (\xi,T,r,\delta,\sigma,\rho,K,B) \in \mathfrak{P}$, $w = (w_1,\dots,w_N) \in \mathbb{R}^{Nd}$ let $\mathcal{X}^{p,w} = (\mathcal{X}^{p,w,1},\mathcal{X}^{p,w,2},\mathcal{X}^{p,w,3}) : \{0,1,\dots,N\} \to \mathbb{R}^d$ satisfy for all $n \in \{1,2,\dots,N\}$, $i \in \{1,2,3\}$ that $\mathcal{X}_0^{p,w} = \xi$ and

$$\mathcal{X}_n^{p,w,i} = \mathcal{X}_{n-1}^{p,w,i} \exp\left(\frac{T}{N}\left[r - \delta_i - \frac{|\sigma_i|^2}{2}\right] + \left[\frac{T}{N}\right]^{1/2} \sigma_i \langle L(\rho)w_n, e_i\rangle\right), \tag{140}$$

for every $p = (\xi,T,r,\delta,\sigma_1,\sigma_2,\sigma_3,\rho,K,B) \in \mathfrak{P}$ let $\mathcal{T}_p = (\mathcal{T}_{p,1},\mathcal{T}_{p,2},\mathcal{T}_{p,3}) : (0,\infty)^d \times (0,\infty)^d \to \mathbb{R}^d$, $\mathcal{U}_p : (0,\infty)^d \times (0,\infty)^d \to [0,1]$, $\mathcal{L}_p : (0,\infty)^d \times (0,\infty)^d \to [0,1]$, and $\mathcal{P}_p : ((0,\infty)^d)^{N+1} \to [0,1]$ satisfy for all $x = (x_1,x_2,x_3)$, $y = (y_1,y_2,y_3) \in (0,\infty)^d$, $\mathbf{x}_0,\mathbf{x}_1,\dots,\mathbf{x}_N \in (0,\infty)^d$, $i \in \{1,2,3\}$ that

$$\mathcal{T}_{p,i}(x,y) = \begin{cases} 1 & : \min\{x,y\} < B \\ \exp\left(-\frac{2\ln(x_i/B)\ln(y_i/B)}{(\sigma_i)^2 T/N}\right) & : \min\{x,y\} \geq B, \end{cases} \tag{141}$$

$$\mathscr{U}_p(x,y) = 1 - \max\left(\cup_{j\in\{1,2,3\}}\{\mathscr{T}_{p,j}(x,y)\}\right), \qquad \mathscr{L}_p(x,y) = \max\left\{1 - \textstyle\sum_{j=1}^{3} \mathscr{T}_{p,j}(x,y), 0\right\}, \tag{142}$$

$$\text{and} \qquad \mathscr{P}_p(\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_N) = \tfrac{1}{2}\left(2 - \left[\prod_{n=1}^{N} \mathscr{U}_p(\mathbf{x}_{n-1}, \mathbf{x}_n)\right] - \left[\prod_{n=1}^{N} \mathscr{L}_p(\mathbf{x}_{n-1}, \mathbf{x}_n)\right]\right), \tag{143}$$

let $\Phi : \mathfrak{P} \times \mathbb{R}^{Nd} \to \mathbb{R}$ satisfy for all $p = (\xi, T, r, \delta, \sigma, \rho, K, B) \in \mathfrak{P}$, $w \in \mathbb{R}^{Nd}$ that

$$\Phi(p,w) = \mathscr{P}_p\left(\mathcal{X}_0^{p,w}, \mathcal{X}_1^{p,w}, \dots, \mathcal{X}_N^{p,w}\right) \exp(-rT) \max\left\{K - \mathbf{m}\left(\mathcal{X}_N^{p,w}\right), 0\right\}, \tag{144}$$

assume for all $p = (\xi, T, r, \delta, \sigma, \rho, K, B) \in \mathfrak{P}$, $w = (w_1, \dots, w_{\mathscr{M}Nd}) \in \mathbb{R}^{\mathscr{M}Nd}$, $\theta = (\theta_1, \dots, \theta_{\mathfrak{M}Nd}) \in \mathbb{R}^{\mathfrak{M}Nd}$ that

$$\Xi(p,w) = \frac{1}{\mathscr{M}}\left[\sum_{\mathfrak{m}=1}^{\mathscr{M}} \Phi\big(p, (w_{(\mathfrak{m}-1)Nd+k})_{k\in\{1,2,\dots,Nd\}}\big)\right] \tag{145}$$

$$\text{and} \qquad \Psi(p,\theta) = \frac{1}{\mathfrak{M}}\left[\sum_{\mathfrak{m}=1}^{\mathfrak{M}} \Phi\big(p, (\theta_{(\mathfrak{m}-1)Nd+k})_{k\in\{1,2,\dots,Nd\}}\big)\right], \tag{146}$$

assume for all $x, y \in \mathbb{R}$ that $\mathfrak{C}(x,y) = |x-y|^2$, let $\mathbf{a} = \frac{9}{10}$, $\mathbf{b} = \frac{999}{1000}$, $\varepsilon \in (0,\infty)$, let $(\gamma_m)_{m\in\mathbb{N}} \subseteq (0,\infty)$ satisfy for all $m \in \{1, 2, \dots, 40000\}$ that $\gamma_m = \mathbb{1}_{(0,20000]}(m)10^{-3} + \mathbb{1}_{(20000,30000]}(m)10^{-4} + \mathbb{1}_{(30000,40000]}(m)10^{-5}$, assume for all $m \in \mathbb{N}$, $i \in \{1, 2, \dots, \mathfrak{d}\}$, $g_1 = (\mathrm{g}_1^{(1)}, \dots, \mathrm{g}_1^{(\mathfrak{d})})$, $g_2 = (\mathrm{g}_2^{(1)}, \dots, \mathrm{g}_2^{(\mathfrak{d})})$, $\dots$, $g_m = (\mathrm{g}_m^{(1)}, \dots, \mathrm{g}_m^{(\mathfrak{d})}) \in \mathbb{R}^{\mathfrak{d}}$ that

$$\psi_m^{(i)}(g_1, g_2, \dots, g_m) = \gamma_m \left[\frac{\sum_{k=1}^{m} \mathbf{a}^{m-k}(1-\mathbf{a})\mathrm{g}_k^{(i)}}{1-\mathbf{a}^m}\right]\left[\varepsilon + \left[\frac{\sum_{k=1}^{m} \mathbf{b}^{m-k}(1-\mathbf{b})|\mathrm{g}_k^{(i)}|^2}{1-\mathbf{b}^m}\right]^{1/2}\right]^{-1}, \tag{147}$$

assume that $\mathfrak{W}$ is a standard normal random vector, assume that $P_{1,1}$ is $\mathcal{U}_{\mathfrak{P}}$-distributed, and assume that $W^{1,1}$ is a standard normal random vector.

Let us add some comments regarding the setup introduced above. The functions $\mathscr{P}_p : ((0,\infty)^d)^{N+1} \to [0,1]$, $p \in \mathfrak{P}$, in (140) are employed to estimate crossing probabilities of Brownian bridges as proposed in Shevchenko (2003) (see also, e.g., Gobet, 2009). Specifically, note that (Gobet, 2009, Displays (12), (13)) suggests that for all $p = (\xi, T, r, \delta, \sigma, \rho, K, B) \in \mathfrak{P}$ we have $\mathbb{P}$-a.s. that

$$\begin{aligned} &\mathscr{P}_p\left(X_0^{\xi,r,\delta,\sigma,\rho}(\mathcal{W}), X_{T/N}^{\xi,r,\delta,\sigma,\rho}(\mathcal{W}), \dots, X_T^{\xi,r,\delta,\sigma,\rho}(\mathcal{W})\right) \\ &\approx \mathbb{P}\Big(\exists\, t \in [0,T] : \mathbf{l}\left(X_t^{\xi,r,\delta,\sigma,\rho}(\mathcal{W})\right) < B \,\big|\, (\mathcal{W}_0, \mathcal{W}_{T/N}, \dots, \mathcal{W}_T)\Big). \end{aligned} \tag{148}$$

TABLE 9 Numerical simulations for the LRV strategy in case of the Black-Scholes model for European average put options with knock-in barriers described in Subsection 7.4 (16-dimensional approximation problem).

$\mathfrak{M}$	Number of trainable parameters	L^1-approx. error	L^2-approx. error	L^∞-approx. error	Training time in seconds
32	960	0.102353	0.144240	0.989026	1092.81
64	1920	0.059373	0.082772	0.600640	1092.24
128	3840	0.034955	0.049141	0.410478	1101.67
256	7680	0.020976	0.029907	0.222575	1123.90
512	15360	0.013157	0.018535	0.139017	1102.52
1024	30720	0.008828	0.012484	0.095372	1428.70
2048	61440	0.006280	0.008905	0.057179	2289.94
4096	122880	0.004984	0.007025	0.054062	4380.77
8192	245760	0.004056	0.005799	0.050417	8742.66

Combining this with (136), (140), (144), the tower property for conditional expectations, and the fact that Brownian motions have independent increments suggests for all $p = (\xi, T, r, \delta, \sigma, \rho, K, B) \in \mathfrak{P}$ that

$$\begin{aligned}
\mathbb{E}[\Phi(p, \mathfrak{W})] &= \mathbb{E}\Big[\Phi\Big(p, \sqrt{N/T}\,(\mathcal{W}_{T/N} - \mathcal{W}_0, \mathcal{W}_{2T/N} - \mathcal{W}_{T/N}, \dots, \mathcal{W}_T - \mathcal{W}_{(N-1)T/N})\Big)\Big] \\
&= \mathbb{E}\Big[\mathscr{P}_p\left(X_0^{\xi,r,\delta,\sigma,\rho}(\mathcal{W}), X_{T/N}^{\xi,r,\delta,\sigma,\rho}(\mathcal{W}), \dots, X_T^{\xi,r,\delta,\sigma,\rho}(\mathcal{W})\right) \\
&\qquad \exp(-rT)\max\left\{K - \mathbf{m}\left(X_T^{\xi,r,\delta,\sigma,\rho}(\mathcal{W})\right), 0\right\}\Big] \\
&\approx \mathbb{E}\Big[\mathbb{P}\Big(\exists\, t \in [0,T] : \mathbf{l}\left(X_t^{\xi,r,\delta,\sigma,\rho}(\mathcal{W})\right) < B \,\big|\, (\mathcal{W}_0, \mathcal{W}_{T/N}, \dots, \mathcal{W}_T)\Big) \\
&\qquad \exp(-rT)\max\left\{K - \mathbf{m}\left(X_T^{\xi,r,\delta,\sigma,\rho}(\mathcal{W})\right), 0\right\}\Big] \\
&= \mathbb{E}\Big[\mathbb{P}\big(X^{\xi,r,\delta,\sigma,\rho}(\mathcal{W}) \in \mathfrak{B}_{T,B} \,|\, \mathcal{W}\big)\exp(-rT)\max\left\{K - \mathbf{m}\left(X_T^{\xi,r,\delta,\sigma,\rho}(\mathcal{W})\right), 0\right\}\Big] \\
&= \mathbb{E}\Big[\mathbb{1}_{\mathfrak{B}_{T,B}}(\mathcal{W})\exp(-rT)\max\left\{K - \mathbf{m}\left(X_T^{\xi,r,\delta,\sigma,\rho}(\mathcal{W})\right), 0\right\}\Big] = \mathbb{E}[\phi(p, \mathcal{W})] = u(p).
\end{aligned} \tag{149}$$

The proposal algorithm in (139) thus corresponds to the MC method based on approximated MC samples. Furthermore, observe that (147) describes the Adam optimizer in the setup of Framework 6.1 (cf. Kingma & Ba, 2014 and Subsection 6.5.7).

In Table 9 we approximately present for $\mathfrak{M} \in \{2^5, 2^6, \dots, 2^{13}\}$ one random realization of the $L^1(\lambda_{\mathfrak{P}}; \mathbb{R})$-approximation error

$$\int_{\mathfrak{P}} |u(p) - \Psi(p, \Theta_{40000})|\, \mathrm{d}p \tag{150}$$

TABLE 10 Numerical simulations for the standard MC method in case of the Black-Scholes model for European average put options with knock-in barriers described in Subsection 7.4 (16-dimensional approximation problem).

Number of MC samples	Number of scalar random variables per evaluation	L^1-approx. error	L^2-approx. error	L^∞-approx. error
32	960	1.131512	1.550692	8.077627
64	1920	0.780559	1.080267	6.354096
128	3840	0.555417	0.762485	4.352852
256	7680	0.392276	0.540796	3.551247
512	15360	0.280314	0.385225	2.428810
1024	30720	0.197006	0.270320	1.669893
2048	61440	0.139361	0.192542	1.172441
4096	122880	0.098571	0.135519	0.899519
8192	245760	0.070336	0.097068	0.597193
16384	491520	0.049755	0.068624	0.537441

(3rd column in Table 9), one random realization of the $L^2(\lambda_{\mathfrak{P}};\mathbb{R})$-approximation error

$$\left[\int_{\mathfrak{P}} |u(p) - \Psi(p, \Theta_{40000})|^2 \, \mathrm{d}p\right]^{1/2} \tag{151}$$

(4th column in Table 9), one random realization of the $L^\infty(\lambda_{\mathfrak{P}};\mathbb{R})$-approximation error

$$\sup\nolimits_{p\in\mathfrak{P}} |u(p) - \Psi(p, \Theta_{40000})|, \tag{152}$$

(5th column in Table 9), and the time to compute Θ_{40000} (6th column in Table 9). For every $\mathfrak{M} \in \{2^5, 2^6, \ldots, 2^{13}\}$ we approximated the integrals in (150) and (151) with the MC method based on 12800 samples and we approximated the supremum in (152) based on 12800 random samples (cf., e.g., Beck, Becker, Grohs et al., 2021, Lemma 3.5 and Beck et al., 2022, Section 3.3). In our approximations of (150), (151), and (152) we have approximately computed for all required sample points $p \in \mathfrak{P}$ the value $u(p)$ of the unknown exact solution by means of an MC approximation with 52428800 MC samples.

Besides the LRV strategy we also employed the standard MC method to approximate the function $u : \mathfrak{P} \to \mathbb{R}$ in (139). In Table 10 we present the corresponding numerical simulation results. In Table 10 we have approximated the $L^1(\lambda_{\mathfrak{P}};\mathbb{R})$-approximation error of the MC method with the MC method based on 12800 samples, we have approximated the $L^2(\lambda_{\mathfrak{P}};\mathbb{R})$-approximation errors of the MC method with the MC method based on 12800 samples, and we have approximated the $L^\infty(\lambda_{\mathfrak{P}};\mathbb{R})$-approximation errors of the MC method based on 12800 random samples (cf., e.g., Beck, Becker, Grohs et al., 2021, Lemma 3.5 and Beck et al., 2022, Section 3.3). In our approximations of the above mentioned approximation errors we have approximately computed for all required sample points $p \in \mathfrak{P}$ the value $u(p)$ of the unknown exact solution by means of an MC approximation with 52428800 MC samples.

7.5 | Parametric stochastic Lorentz equations

In this section we apply the LRV strategy to the parametric stochastic Lorentz equation (cf., e.g, Schmallfuß, 1997 and Hutzenthaler & Jentzen, 2015, Section 4.4). A brief summary of the numerical results of this subsection can be found in Table 11 below. We first introduce the parametric stochastic Lorentz equation in the context of Framework 6.1.

Assume Framework 6.1, let $d = 3$, let $g : \mathbb{R}^d \to \mathbb{R}$ satisfy for all $x \in \mathbb{R}^d$ that $g(x) = \|x\|^2$, for every let $\alpha = (\alpha_1, \alpha_2, \alpha_3) \in \mathbb{R}^d$ let $\mu_\alpha : \mathbb{R}^d \to \mathbb{R}^d$ satisfy for all $x = (x_1, x_2, x_3) \in \mathbb{R}^d$ that

$$\mu_\alpha(x) = (\alpha_1(x_2 - x_1), \alpha_2 x_1 - x_2 - x_1 x_3, x_1 x_2 - \alpha_3 x_3), \tag{153}$$

and for every $\alpha = (\alpha_1, \alpha_2, \alpha_3), \beta = (\beta_1, \beta_2, \beta_3) \in \mathbb{R}^d$ let $u_{\alpha,\beta} = (u_{\alpha,\beta}(t,x))_{(t,x)\in[0,\infty)\times\mathbb{R}^d} \in C^{1,2}([0,\infty)\times\mathbb{R}^d, \mathbb{R})$ be an at most polynomially growing function which satisfies for all $t \in [0,\infty)$, $x \in \mathbb{R}^d$ that

$$\begin{aligned}\left(\frac{\partial u_{\alpha,\beta}}{\partial t}\right)(t,x) = \alpha_1(x_2 - x_1)\left(\frac{\partial u_{\alpha,\beta}}{\partial x_1}\right)(t,x) + (\alpha_2 x_1 - x_2 - x_1 x_3)\left(\frac{\partial u_{\alpha,\beta}}{\partial x_2}\right)(t,x)\\ + (x_1 x_2 - \alpha_3 x_3)\left(\frac{\partial u_{\alpha,\beta}}{\partial x_3}\right)(t,x) + \sum_{i=1}^{d} \frac{(\beta_i)^2}{2}\left(\frac{\partial^2 u_{\alpha,\beta}}{\partial (x_i)^2}\right)(t,x)\end{aligned} \tag{154}$$

and $u_{\alpha,\beta}(0,x) = g(x)$.

We now specify the mathematical objects in the LRV strategy appearing in Framework 6.1 to approximately calculate $(u_{\alpha,\beta})_{(\alpha,\beta)\in\mathbb{R}^d\times\mathbb{R}^d}$. Specifically, in addition to the assumptions above, let $\odot : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ satisfy for all $x = (x_1, x_2, x_3)$, $y = (y_1, y_2, y_3)$ that $x \odot y = (x_1 y_1, x_2 y_2, x_3 y_3)$, let $N = 25$, $\mathscr{M} = 512$, $\mathfrak{M} \in \mathbb{N}$, $a \in \{0,1\}$, assume $\mathfrak{p} = 10$, $\mathfrak{d} = \mathfrak{M}Nd$, $\mathbf{d} = \mathscr{M}Nd$, $k = 1$, $\mathbf{M} = 512$, and

$$\mathfrak{P} = [0.01, 1] \times ([9, 11] \times [13, 15] \times [1, 2]) \times [0.05, 0.25]^3 \times ([0.5, 2.5] \times [8, 10] \times [10, 12]), \tag{155}$$

for every $p = (T, \alpha, \beta, x) \in \mathfrak{P}$, $w = (w_1, \dots, w_N) \in \mathbb{R}^{Nd}$ let $\mathcal{X}^{p,w} : \{0, 1, \dots, N\} \to \mathbb{R}^d$ satisfy for all $n \in \{1, 2, \dots, N\}$ that $\mathcal{X}_0^{p,w} = x$ and

$$\mathcal{X}_n^{p,w} = \mathcal{X}_{n-1}^{p,w} + \sqrt{T/N}(\beta \odot w_n) + \tfrac{T}{2N}\Big(\mu_\alpha\big(\mathcal{X}_{n-1}^{p,w}\big) + \mu_\alpha\Big(\mathcal{X}_{n-1}^{p,w} + \tfrac{T}{N}\mu_\alpha\big(\mathcal{X}_{n-1}^{p,w}\big) + \sqrt{T/N}(\beta \odot w_n)\Big)\Big), \tag{156}$$

let $\Phi_k : \mathfrak{P} \times \mathbb{R}^{Nd} \to \mathbb{R}$, $k \in \{0,1\}$, satisfy for all $p \in \mathfrak{P}$, $w \in \mathbb{R}^{Nd}$ that

$$\Phi_0(p,w) = g\big(\mathcal{X}_N^{p,w}\big) \qquad \text{and} \qquad \Phi_1(p,w) = \tfrac{1}{2}\big[g\big(\mathcal{X}_N^{p,w}\big) + g\big(\mathcal{X}_N^{p,-w}\big)\big], \tag{157}$$

assume for all $p \in \mathfrak{P}$, $w = (w_1, \dots, w_{\mathscr{M}Nd}) \in \mathbb{R}^{\mathscr{M}Nd}$, $\theta = (\theta_1, \dots, \theta_{\mathfrak{M}Nd}) \in \mathbb{R}^{\mathfrak{M}Nd}$ that

$$\Xi(p,w) = \frac{1}{\mathscr{M}}\left[\sum_{\mathfrak{m}=1}^{\mathscr{M}} \Phi_a\big(p, (w_{(\mathfrak{m}-1)Nd+k})_{k\in\{1,2,\dots,Nd\}}\big)\right] \tag{158}$$

$$\text{and} \qquad \Psi(p,\theta) = \frac{1}{\mathfrak{M}}\left[\sum_{\mathfrak{m}=1}^{\mathfrak{M}} \Phi_a\big(p, (\theta_{(\mathfrak{m}-1)Nd+k})_{k\in\{1,2,\dots,Nd\}}\big)\right], \tag{159}$$

TABLE 11 Parametric stochastic Lorentz equation.

Approximation method	Number of trainable parameters	Number of MC samples	L^2-approx. error	L^∞-approx. error	Training time in seconds
ANNs (2 hidden layers, 128 neurons each)	18049	-	13.618965	377.537964	1869.14
ANNs (2 hidden layers, 256 neurons each)	68865	-	9.487827	106.434830	1882.02
ANNs (3 hidden layers, 128 neurons each)	34561	-	15.087503	200.595459	1903.12
ANNs (3 hidden layers, 256 neurons each)	134657	-	110.419151	313.946106	1891.98
Monte Carlo	-	32768	0.014373	0.114243	0
Monte Carlo (antithetic)	-	32768	0.000457	0.006233	0
LRV	76800	1024	0.000958	0.015106	327.63
LRV (antithetic)	76800	1024	0.000223	0.004013	518.68

assume for all $x, y \in \mathbb{R}$ that $\mathfrak{C}(x, y) = |x - y|^2$, let $\mathbf{a} = \frac{9}{10}$, $\mathbf{b} = \frac{999}{1000}$, $\varepsilon \in (0, \infty)$, let $(\gamma_m)_{m \in \mathbb{N}} \subseteq (0, \infty)$ satisfy for all $m \in \{1, 2, \dots, 10000\}$ that $\gamma_m = \mathbb{1}_{(0,5000]}(m) 10^{-3} + \mathbb{1}_{(5000,8000]}(m) 10^{-4} + \mathbb{1}_{(8000,10000]}(m) 10^{-5}$, assume for all $m \in \mathbb{N}$, $i \in \{1, 2, \dots, \mathfrak{d}\}$, $\mathfrak{g}_1 = (\mathfrak{g}_1^{(1)}, \dots, \mathfrak{g}_1^{(\mathfrak{d})})$, $\mathfrak{g}_2 = (\mathfrak{g}_2^{(1)}, \dots, \mathfrak{g}_2^{(\mathfrak{d})})$, $\dots$, $\mathfrak{g}_m = (\mathfrak{g}_m^{(1)}, \dots, \mathfrak{g}_m^{(\mathfrak{d})}) \in \mathbb{R}^{\mathfrak{d}}$ that

$$\psi_m^{(i)}(\mathfrak{g}_1, \mathfrak{g}_2, \dots, \mathfrak{g}_m) = \gamma_m \left[\frac{\sum_{k=1}^m \mathbf{a}^{m-k}(1 - \mathbf{a}) \mathfrak{g}_k^{(i)}}{1 - \mathbf{a}^m} \right] \left[\varepsilon + \left[\frac{\sum_{k=1}^m \mathbf{b}^{m-k}(1 - \mathbf{b}) \left| \mathfrak{g}_k^{(i)} \right|^2}{1 - \mathbf{b}^m} \right]^{1/2} \right]^{-1}, \tag{160}$$

assume that $\mathfrak{W}$ is a standard normal random vector, assume that $P_{1,1}$ is $\mathcal{U}_{\mathfrak{P}}$-distributed, and assume that $W^{1,1}$ is a standard normal random vector.

Let us add some comments regarding the setup introduced above. Observe that (156) corresponds to a Heun discretization (cf., e.g., Kloeden & Platen, 1992, (1.4) in Section 15) of the SDE associated to the Kolmogorov PDE in (154). In the case $a = 0$ the proposal algorithm in (159) thus corresponds to the MC-Heun method and in the case $a = 1$ the proposal algorithm in (159) thus corresponds to the antithetic MC-Heun method. In particular, we remark that for all $p = (T, \alpha, \beta, x) \in \mathfrak{P}$ we have that

$$\frac{1}{\mathfrak{M}} \left[\sum_{\mathfrak{m}=1}^{\mathfrak{M}} \Phi_0\big(p, W^{0,\mathfrak{m}}\big) \right] \approx u_{\alpha,\beta}(T, x) \approx \frac{1}{\mathfrak{M}} \left[\sum_{\mathfrak{m}=1}^{\mathfrak{M}} \Phi_1\big(p, W^{0,\mathfrak{m}}\big) \right]. \tag{161}$$

Furthermore, observe that (160) describes the Adam optimizer in the setup of Framework 6.1 (cf. Kingma & Ba, 2014 and Subsection 6.5.7).

In Table 12 we approximately present for $\mathfrak{M} \in \{2^5, 2^6, \dots, 2^{13}\}$, $a \in \{0, 1\}$ one random realization of the $L^1(\lambda_{\mathfrak{P}}; \mathbb{R})$-approximation error

$$\int_{\mathfrak{P}} |u_{\alpha,\beta}(T, x) - \Psi((T, \alpha, \beta, x), \Theta_{10000})| \, \mathrm{d}(T, \alpha, \beta, x) \tag{162}$$

(3rd column in Table 12), one random realization of the $L^2(\lambda_{\mathfrak{P}}; \mathbb{R})$-approximation error

$$\left[\int_{\mathfrak{P}} |u_{\alpha,\beta}(T, x) - \Psi((T, \alpha, \beta, x), \Theta_{10000})|^2 \, \mathrm{d}(T, \alpha, \beta, x) \right]^{1/2} \tag{163}$$

(4th column in Table 12), one random realization of the $L^\infty(\lambda_{\mathfrak{P}}; \mathbb{R})$-approximation error

$$\sup_{(T,\alpha,\beta,x) \in \mathfrak{P}} |u_{\alpha,\beta}(T, x) - \Psi((T, \alpha, \beta, x), \Theta_{10000})| \tag{164}$$

(5th column in Table 12), and the time to compute Θ_{10000} (6th column in Table 12). For every $\mathfrak{M} \in \{2^5, 2^6, \dots, 2^{13}\}$, $a \in \{0, 1\}$ we approximated the integrals in (162) and (163) with the MC method based on 12800 samples and we approximated the supremum in (164) based on 12800 random samples (cf., e.g., Beck, Becker, Grohs et al., 2021, Lemma 3.5 and Beck et al., 2022, Section 3.3). In our approximations of (162), (163), and (164) we have approximately computed for all required sample points $(T, \alpha, \beta, x) \in \mathfrak{P}$ the value $u_{\alpha,\beta}(T, x)$ of the unknown exact solution by means of an antithetic MC approximation with 8 196 000 MC samples.

TABLE 12 Numerical simulations for the LRV strategy in case of the parametric stochastic Lorentz equation described in Subsection 7.5 (10-dimensional approximation problem).

$\mathfrak{M}$	a	Number of trainable parameters	L^1-approx. error	L^2-approx. error	L^∞-approx. error	Training time in seconds
32	0	2400	0.001903	0.003234	0.042107	321.32
32	1	2400	0.000376	0.000714	0.011292	481.52
64	0	4800	0.001706	0.003016	0.042892	323.65
64	1	4800	0.000296	0.000587	0.010147	481.12
128	0	9600	0.001648	0.003062	0.062836	322.32
128	1	9600	0.000276	0.000589	0.011658	480.80
256	0	19200	0.000801	0.001198	0.015732	320.24
256	1	19200	0.000166	0.000328	0.009003	480.83
512	0	38400	0.000732	0.001265	0.020386	322.96
512	1	38400	0.000173	0.000376	0.008942	482.99
1024	0	76800	0.000537	0.000958	0.015106	327.63
1024	1	76800	0.000112	0.000223	0.004013	518.68
2048	0	153600	0.000721	0.001086	0.014084	399.33
2048	1	153600	0.000069	0.000118	0.001572	968.62
4096	0	307200	0.000627	0.000844	0.007401	836.97
4096	1	307200	0.000090	0.000155	0.001984	2174.38
8192	0	614400	0.000177	0.000282	0.004089	1924.66
8192	1	614400	0.000085	0.000144	0.001419	4571.75

TABLE 13 Numerical simulations for the deep learning method induced by Beck, Becker, Grohs et al. (2021) in case of the parametric stochastic Lorentz equation described in Subsection 7.5 (10-dimensional approximation problem).

Number of hidden layers	Number of neurons on each hidden layer	Number of trainable parameters	L^1-approx. error	L^2-approx. error	L^∞-approx. error	Training time in seconds
2	32	1441	2.299140	5.982908	104.470703	1876.86
2	64	4929	4.730990	16.753924	199.115005	1863.79
2	128	18049	4.884268	13.618965	377.537964	1869.14
2	256	68865	3.218077	9.487827	106.434830	1882.02
3	32	2497	3.712238	11.086662	693.645264	1880.95
3	64	9089	28.689849	161.865602	3000.016357	1881.48
3	128	34561	5.824964	15.087503	200.595459	1903.12
3	256	134657	93.630541	110.419151	313.946106	1891.98

To compare the LRV strategy with existing approximation techniques from the literature, we also employ several other methods to approximate the function $\mathfrak{P} \ni (T, \alpha, \beta, x) \mapsto u_{\alpha,\beta}(T, x) \in \mathbb{R}$ in (154). Specifically, in Table 13 we present numerical simulations for the deep learning method induced by Beck, Becker, Grohs et al. (2021) (with Adam 160000 training steps, batch size 256, learning rate schedule $\mathbb{N} \ni j \mapsto \mathbb{1}_{(0,50000]}(m)10^{-1} + \mathbb{1}_{(50000,100000]}(m)10^{-2} + \mathbb{1}_{(100000,120000]}(m)10^{-3} + \mathbb{1}_{(120000,140000]}(m)10^{-4} + \mathbb{1}_{(140000,160000]}(m)10^{-5}$, and GELU activation

TABLE 14 Numerical simulations for the standard and the antithetic MC method in case of the parametric stochastic Lorentz equation described in Subsection 7.5 (10-dimensional approximation problem).

Number of MC samples	MC Method 0: standard 1: antithetic	Number of scalar random variables per evaluation	L^1-approx. error	L^2-approx. error	L^∞-approx. error
32	0	2400	0.383773	0.534779	4.735626
32	1	2400	0.008103	0.016280	0.210976
64	0	4800	0.275408	0.375781	3.149811
64	1	4800	0.005694	0.011187	0.183968
128	0	9600	0.204422	0.281741	2.723969
128	1	9600	0.003942	0.007754	0.103683
256	0	19200	0.145995	0.200546	1.234207
256	1	19200	0.002641	0.005119	0.073952
512	0	38400	0.093797	0.127197	1.015808
512	1	38400	0.001872	0.003753	0.071739
1024	0	76800	0.073213	0.101349	0.916580
1024	1	76800	0.001386	0.002828	0.041504
2048	0	153600	0.048225	0.067596	0.737793
2048	1	153600	0.000961	0.001928	0.030483
4096	0	307200	0.034988	0.048361	0.357437
4096	1	307200	0.000664	0.001317	0.020393
8192	0	614400	0.025400	0.034397	0.242668
8192	1	614400	0.000449	0.000922	0.016327
16384	0	1228800	0.018963	0.026513	0.256882
16384	1	1228800	0.000357	0.000715	0.010361
32768	0	2457600	0.010552	0.014373	0.114243
32768	1	2457600	0.000241	0.000457	0.006233

function), and in Table 14 we present numerical simulations for the standard and the antithetic MC method. In Tables 13 and 14 we have approximated the $L^1(\lambda_{\mathfrak{P}}; \mathbb{R})$-approximation errors of the respective approximation methods with the MC method based on 12800 samples, we have approximated the $L^2(\lambda_{\mathfrak{P}}; \mathbb{R})$-approximation errors of the respective approximation methods with the MC method based on 12800 samples, and we have approximated the $L^\infty(\lambda_{\mathfrak{P}}; \mathbb{R})$-approximation errors of the respective approximation methods based on 12800 random samples (cf., e.g., Beck, Becker, Grohs et al., 2021, Lemma 3.5 and Beck et al., 2022, Section 3.3). In our approximations of the above mentioned approximation errors we have approximately computed for all required sample points $(T, \alpha, \beta, x) \in \mathfrak{P}$ the value $u_{\alpha,\beta}(T, x)$ of the unknown exact solution by means of an antithetic MC approximation with 8 196 000 MC samples

ACKNOWLEDGMENTS

Nor Jaafari is gratefully acknowledged for his useful assistance regarding some numerical simulations. The second author gratefully acknowledges the Cluster of Excellence EXC 2044-390685587, Mathematics Münster: Dynamics-Geometry-Structure funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation). This work has been partially funded by the National Science Foundation of China (NSFC) under grant number 12250610192.

Open access funding provided by Eidgenossische Technische Hochschule Zurich.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Arnulf Jentzen https://orcid.org/0000-0002-9840-3339
Marvin S. Müller https://orcid.org/0000-0001-7579-4114
Philippe von Wurstemberger https://orcid.org/0000-0003-3881-7684

REFERENCES

Altmayer, M., Dereich, S., Li, S., Müller-Gronbach, T., Neuenkirch, A., Ritter, K., & Yaroslavtseva, L. (2014). Constructive quantization and multilevel algorithms for quadrature of stochastic differential equations. In *Extraction of Quantifiable Information from Complex Systems*, vol. 102 of *Lecture Notes in Computational Science and Engineering* (pp. 109–132). Springer.

Andersson, K., & Oosterlee, C. W. (2021). A deep learning approach for computations of exposure profiles for high-dimensional Bermudan options. *Applied Mathematics and Computation*, *408*, Paper No. 126332, 21.

Beck, C., Becker, S., Cheridito, P., Jentzen, A., & Neufeld, A. (2021). Deep splitting method for parabolic PDEs. *SIAM Journal of Scientific Computing*, *43*(5), A3135–A3154.

Beck, C., Becker, S., Grohs, P., Jaafari, N., & Jentzen, A. (2021). Solving the Kolmogorov PDE by means of deep learning. *Journal of Scientific Computing*, *88*(3), Paper No. 73, 28.

Beck, C., Hutzenthaler, M., & Jentzen, A. (2021). On nonlinear Feynman–Kac formulas for viscosity solutions of semilinear parabolic partial differential equations. *Stochastics and Dynamics*, *21*(8), Paper No. 2150048, 68.

Beck, C., Hutzenthaler, M., Jentzen, A., & Kuckuck, B. (2023). An overview on deep learning-based approximation methods for partial differential equations. *Discrete and Continuous Dynamical Systems – Series B*, *28*(6), 3697–3746.

Beck, C., Jentzen, A., & Kuckuck, B. (2022). Full error analysis for the training of deep neural networks. *Infinite Dimensional Analysis, Quantum Probability and Related Topics*, *25*(2), Paper No. 2150020, 76.

Becker, S., Cheridito, P., & Jentzen, A. (2019). Deep optimal stopping. *The Journal of Machine Learning Research*, *20*, Paper No. 74, 25.

Becker, S., Cheridito, P., & Jentzen, A. (Jul, 2020). Pricing and hedging American-style options with deep learning. *Journal of Risk and Financial Management*, *13*(7), 158.

Becker, S., Cheridito, P., Jentzen, A., & Welti, T. (2021). Solving high-dimensional optimal stopping problems using deep learning. *European Journal of Applied Mathematics*, *32*(3), 470–514.

Bellman, R. (2010). *Dynamic programming*. Princeton landmarks in mathematics. Princeton University Press. Reprint of the 1957 edition, With a new introduction by Stuart Dreyfus.

Berner, J., Dablander, M., & Grohs, P. (2020). Numerically solving parametric families of high-dimensional Kolmogorov partial differential equations via deep learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems* (Vol. 33, pp. 16615–16627) Curran Associates, Inc.

Betancourt, M. (2017). A conceptual introduction to Hamiltonian Monte Carlo. *arXiv:1701.02434*.

Bhattacharya, K., Hosseini, B., Kovachki, N. B., & Stuart, A. M. (2021). Model reduction and neural networks for parametric PDEs. *SMAI Journal of Computational Mathematics*, *7*, 121–157.

Biagini, F., Gonon, L., & Reitsam, T. (2021). Neural network approximation for superhedging prices. *arXiv:2107.14113*.

Black, F., & Scholes, M. (1973). The pricing of options and corporate liabilities. *Journal of Political Economy*, *81*(3), 637–654.

Blechschmidt, J., & Ernst, O. G. (2021). Three ways to solve partial differential equations with neural networks – A review. *arXiv:2102.11802*.

Borgerding, M., Schniter, P., & Rangan, S. (2016). AMP-inspired deep networks for sparse linear inverse problems. *arXiv:1612.01183*.

Caflisch, R. E. (Jan, 1998). Monte Carlo and quasi-Monte Carlo methods. *Acta Numerica*, *7*, 1–49.

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

Chen, F., Huang, J., Wang, C., & Yang, H. (2020). Friedrichs learning: Weak solutions of partial differential equations via deep learning. *arXiv:2012.08023*.

Chen, X., Li, Y., Umarov, R., Gao, X., & Song, L. (2020). RNA secondary structure prediction by learning unrolled algorithms. In Proceedings of the *International Conference on Learning Representations*.

Chen, X., Liu, J., Wang, Z., & Yin, W. (2018). Theoretical linear convergence of unfolded ISTA and its practical weights and thresholds. *arXiv:1808.10038*.

Chen, X., Zhang, Y., Reisinger, C., & Song, L. (2020). Understanding deep architectures with reasoning layer. *arXiv:2006.13401*.

Chen, Y., & Wan, J. W. L. (2019). Deep neural network framework based on backward stochastic differential equations for pricing and hedging American options in high dimensions. *arXiv:1909.11532*.

Chen, Y., & Wan, J. W. L. (2021). Deep neural network framework based on backward stochastic differential equations for pricing and hedging American options in high dimensions. *Quantitative Finance*, *21*(1), 45–67.

Chkifa, A., Cohen, A., & Schwab, C. (2015). Breaking the curse of dimensionality in sparse polynomial approximation of parametric PDEs. *Journal des Mathematiques Pures et Appliquees (9)*, *103*(2), 400–428.

Cohen, A., & DeVore, R. (2015). Approximation of high-dimensional parametric PDEs. *Acta Numerica*, *24*, 1–159.

Creutzig, J., Dereich, S., Müller-Gronbach, T., & Ritter, K. (2009). Infinite-dimensional quadrature and approximation of distributions. *Foundations of Computational Mathematics*, *9*(4), 391–429.

Dellaportas, P., & Titsias, M. K. (2019) Gradient-based adaptive Markov chain Monte Carlo. In Proceedings of Advances in Neural Information Processing Systems.

Dereich, S., Scheutzow, M., & Schottstedt, R. (2013). Constructive quantization: Approximation by empirical measures. *Annales de l'IHP Probabilités et statistiques*, *49*(4), 1183–1203.

Dissanayake, M. W. M. G., & Phan-Thien, N. (1994). Neural-network-based approximations for solving partial differential equations. *Communications in Numerical Methods in Engineering*, *10*(3), 195–201.

Drovandi, C. C., & Tran, M.-N. (2018). Improving the efficiency of fully bayesian optimal design of experiments using randomised quasi-Monte Carlo. *Bayesian Analysis*, *13*(1), 139–162.

E, W., Han, J., & Jentzen, A. (2017). Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. *Communications in Mathematics and Statistics*, *5*(4), 349–380.

E, W., Han, J., & Jentzen, A. (2022). Algorithms for solving high dimensional PDEs: From nonlinear Monte Carlo to machine learning. *Nonlinearity*, *35*, 278–310.

E, W., Hutzenthaler, M., Jentzen, A., & Kruse, T. (2019). On multilevel Picard numerical approximations for high-dimensional nonlinear parabolic partial differential equations and high-dimensional nonlinear backward stochastic differential equations. *Journal of Scientific Computing*, *79*(3), 1534–1571.

E, W., Hutzenthaler, M., Jentzen, A., & Kruse, T. (2021). Multilevel Picard iterations for solving smooth semilinear parabolic heat equations. *Partial Differential Equations and Applications*, *2*(6), 80.

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

Ferguson, R., & Green, A. D. (2018). Deeply learning derivatives. Available at *SSRN: 3244821*.

Flandoli, F., Luo, D., & Ricci, C. (2021). A numerical approach to Kolmogorov equation in high dimension based on Gaussian analysis. *Journal of Mathematical Analysis and Applications*, *493*(1), Paper No. 124505, 29.

Frikha, N., & Sagna, A. (2012). Quantization based recursive importance sampling. *Monte Carlo Methods and Applications*, *18*(4), 287–326.

Fujii, M., Takahashi, A., & Takahashi, M. (Sept 2019). Asymptotic expansion as prior knowledge in deep learning method for high dimensional BSDEs. *Asia-Pacific Financial Markets*, *26*(3), 391–408.

Gaspar, R. M., Lopes, S. D., & Sequeira, B. (2020). Neural network pricing of American put options. *Risks*, *8*(3), 73.

Germain, M., Pham, H., & Warin, X. (2020). Approximation error analysis of some deep backward schemes for nonlinear PDEs. *arXiv:2006.01496*.

Germain, M., Pham, H., & Warin, X. (2021). Neural networks-based algorithms for stochastic control and PDEs in finance. *arXiv:2101.08068*.

Giles, M. B. (2008). Multilevel Monte Carlo path simulation. *Operations Research*, *56*(3), 607–617.

Giles, M. B. (2015). Multilevel Monte Carlo methods. *Acta Numerica*, *24*, 259–328.

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

Gobet, E. (2009). Advanced Monte Carlo methods for barrier and related exotic options. In *Special volume: Mathematical modeling and numerical methods in finance* (pp. 497–528). Elsevier.

Gonon, L., Grohs, P., Jentzen, A., Kofler, D., & Šiška, D. (2021). Uniform error estimates for artificial neural network approximations for heat equations. *IMA Journal of Numerical Analysis, 42(3), 1991-2054*.

Goudenège, L., Molent, A., & Zanette, A. (2019). Variance reduction applied to machine learning for pricing Bermudan/American options in high dimension. *arXiv:1903.11275*.

Graf, S., & Luschgy, H. (2000). *Foundations of quantization for probability distributions*, Vol. 1730 of Lecture notes in mathematics. Springer-Verlag.

Gregor, K., & LeCun, Y. (2010). Learning fast approximations of sparse coding. In *Proceedings of the 27th International Conference on International Conference onMachine Learning* (pp. 399–406).

Grohs, P., & Voigtlaender, F. (2021). Proof of the theory-to-practice gap in deep learning via sampling complexity bounds for neural network approximation spaces. *arXiv:2104.02746*.

Han, J., & Hu, R. (2020). Deep fictitious play for finding Markovian Nash equilibrium in multi-agent games. In *Proceedings of The First Mathematical and Scientific Machine Learning Conference (MSML)* (Vol. 107, pp. 221–245).

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

Han, J., & Long, J. (2020). Convergence of the deep BSDE method for coupled FBSDEs. *Probability, Uncertainty and Quantitative Risk*, *5*, Paper No. 5, 33.

Han, J., Lu, J., & Zhou, M. (2020). Solving high-dimensional eigenvalue problems using deep neural networks: A diffusion Monte Carlo like approach. *Journal of Computational Physics*, *423*, 109792.

Han, J., Nica, M., & Stinchcombe, A. R. (2020). A derivative-free method for solving elliptic partial differential equations with deep neural networks. *Journal of Computational Physics*, *419*, 109672.

Heinrich, S. (1998). Monte Carlo complexity of global solution of integral equations. *Journal of Complexity*, *14*(2), 151–175.

Heinrich, S. (2001). Multilevel Monte Carlo methods. In *Large-scale scientific computing* (pp. 58–67). Springer.

Heinrich, S. (2006). The randomized information complexity of elliptic PDE. *Journal of Complexity*, *22*(2), 220–249. https://link.springer.com/book/10.1007/978-3-662-12616-5

Heinrich, S., & Sindambiwe, E. (1999). Monte Carlo complexity of parametric integration. *Journal of Complexity*, *15*(3), 317–341.

Henry-Labordere, P. (2017). Deep primal-dual algorithm for BSDEs: Applications of machine learning to CVA and IM. *Available at SSRN 3071506*.

Huré, C., Pham, H., Pham, H., & Warin, X. (2020). Deep backward schemes for high-dimensional nonlinear PDEs. *Mathematics of Computation*, *89*(324), 1547–1579.

Hutzenthaler, M., & Jentzen, A. (2015). Numerical approximations of stochastic differential equations with non-globally Lipschitz continuous coefficients. *Memoirs of the American Mathematical Society*, *236*, 1112, v+99.

Hutzenthaler, M., Jentzen, A., Kruse, T., Nguyen, T. A., & von Wurstemberger, P. (2020). Overcoming the curse of dimensionality in the numerical approximation of semilinear parabolic partial differential equations. *Proceedings of the Royal Society A*, *476*(2244), 20190630, 25.

Hutzenthaler, M., Jentzen, A., & von Wurstemberger, P. (2020). Overcoming the curse of dimensionality in the approximative pricing of financial derivatives with default risks. *Electronic Journal of Probability*, *25*, Paper No. 101, 73.

Jacquier, A., & Oumgari, M. (2019). Deep curve-dependent PDEs for affine rough volatility. *arXiv:1906.02551*.

Ji, S., Peng, S., Peng, Y., & Zhang, X. (2020). Three algorithms for solving high-dimensional fully coupled FBSDEs through deep learning. *IEEE Intelligent Systems*, *35*(3), 71–84.

Jianyu, L., Siwei, L., Yingjian, Q., & Yaping, H. (2003). Numerical solution of elliptic partial differential equation using radial basis function neural networks. *Neural Networks*, *16*(5-6), 729–734.

Khoo, Y., Lu, J., & Ying, L. (2021). Solving parametric PDE problems with artificial neural networks. *European Journal of Applied Mathematics*, *32*(3), 421–435.

Kingma, D. P., & Ba, J. (Dec, 2014). Adam: A method for stochastic pptimization. *arXiv:1412.6980*.

Klenke, A. (2014). *Probability theory* (2nd ed.). Universitext. Springer-Verlag London Ltd.

Kloeden, P. E., & Platen, E. (1992). *Numerical solution of stochastic differential equations*, Vol. 23 of *Applications of mathematics (New York)*. Springer-Verlag.

Kremsner, S., Steinicke, A., & Szölgyenyi, M. (2020). A deep neural network algorithm for semilinear elliptic PDEs with applications in insurance mathematics. *Risks*, *8*(4), 136.

Kutyniok, G., Petersen, P., Raslan, M., & Schneider, R. (2021). A theoretical analysis of deep neural networks and parametric PDEs. *Constructive Approximation, 55, 73-125*.

Lagaris, I. E., Likas, A., & Fotiadis, D. I. (1998). Artificial neural networks for solving ordinary and partial differential equations. *IEEE Transactions on Neural Networks*, *9*(5), 987–1000.

Lapeyre, B., & Lelong, J. (2021). Neural network regression for Bermudan option pricing. *Monte Carlo Methods and Applications*, *27*(3), 227–247.

Lind, P. P. (2022). Overcoming the feature selection issue in the pricing of American options. *Available at SSRN: 4014593*.

Lokeshwar, V., Bharadwaj, V., & Jain, S. (2022). Explainable neural network for pricing and universal static hedging of contingent claims. *Applied Mathematics and Computation*, *417*, Paper No. 126775, 20.

Longo, M., Mishra, S., Rusch, T. K., & Schwab, C. (2020). Higher-order quasi-Monte Carlo training of deep neural networks. *arXiv:2009.02713*.

Lu, L., Meng, X., Mao, Z., & Karniadakis, G. E. (2019). Deepxde: A deep learning library for solving differential equations. *arXiv:1907.04502*.

Lye, K. O., Mishra, S., & Ray, D. (2020). Deep learning observables in computational fluid dynamics. *Journal of Computational Physics*, *410*, 109339, 26.

Lyu, L., Zhang, Z., Chen, M., & Chen, J. (2020). Mim: A deep mixed residual method for solving high-order partial differential equations. *arXiv:2006.04146*.

Martin, C., Zhang, H., Costacurta, J., Nica, M., & Stinchcombe, A. R. (Aug 2020). Solving elliptic equations with Brownian motion: Bias reduction and temporal difference learning. *arXiv:2008.00144*.

Mensch, A., & Blondel, M. (2018). Differentiable dynamic programming for structured prediction and attention. In J. Dy, & A. Krause (Eds.), *Proceedings of the 35th International Conference on Machine Learning*, Vol. 80 of Proceedings of Machine Learning Research (pp. 3462–3471). PMLR.

Merton, R. C. (1973). Theory of rational option pricing. The Bell Journal of Economics and Management Science, 4(1), 141–183.

Morokoff, W. J., & Caflisch, R. E. (Dec, 1995). Quasi-Monte Carlo integration. *Journal of Computational Physics*, *122*(2), 218–230.

Müller-Gronbach, T., & Ritter, K. (2013). A local refinement strategy for constructive quantization of scalar SDEs. *Foundations of Computational Mathematics*, *13*(6), 1005–1033.

Novak, E., & Woźniakowski, H. (2008a). *Tractability of multivariate problems: Standard information for functionals* (Vol. 12.). European Mathematical Society.

Novak, E., & Woźniakowski, H. (2008b). *Tractability of multivariate problems. Vol. 1: Linear information*, Vol. 6 of EMS Tracts in Mathematics. European Mathematical Society (EMS).

Nüsken, N., & Richter, L. (2021a). Solving high-dimensional Hamilton–Jacobi–Bellman PDEs using neural networks: Perspectives from the theory of controlled diffusions and measures on path space. *Partial Differential Equations and Applications*, *2*(4), Paper No. 48.

Nüsken, N., & Richter, L. (2021b). Interpolating between BSDEs and PINNs – Deep learning for elliptic and parabolic boundary value problems. *arXiv:2112.03749*.

O'Leary-Roseberry, T., Villa, U., Chen, P., & Ghattas, O. (2022). Derivative-informed projected neural networks for high-dimensional parametric maps governed by PDEs. *Computer Methods in Applied Mechanics and Engineering*, *388*, Paper No. 114199, 24.

Pagès, G. (1998). A space quantization method for numerical integration. *Journal of Computational and Applied Mathematics*, *89*(1), 1–38.

Pagès, G. (2015). Introduction to vector quantization and its applications for numerics. *ESAIM: Proceedings and Surveys*, *48*, 29–79.

Pagès, G., Pham, H., & Printems, J. (2004). Optimal quantization methods and applications to numerical problems in finance. In *Handbook of computational and numerical methods in finance* (pp. 253–297). Springer.

Pagès, G., & Printems, J. (2003). Optimal quadratic quantization for numerics: The Gaussian case. *Monte Carlo Methods and Applications*, *9*(2), 135–165.

Pagès, G., & Printems, J. (2005). Functional quantization for numerics with an application to option pricing. *Monte Carlo Methods and Applications*, *11*(4), 407–446.

Pereira, M. A., Wang, Z., Exarchos, I., & Theodorou, E. A. (2019). Learning deep stochastic optimal control policies using forward-backward SDEs. In Proceedings of *Robotics: Science and systems*.

Pham, H., Runggaldier, W., & Sellami, A. (2005). Approximation by quantization of the filter process and applications to optimal stopping problems under partial observation. *Monte Carlo Methods and Applications*, *11*(1), 57–81.

Pham, H., Warin, X., & Germain, M. (2019). Neural networks-based backward scheme for fully nonlinear PDEs. *arXiv:1908.00412*.

Raissi, M. (2018). Forward-backward stochastic neural networks: Deep learning of high-dimensional partial differential equations. *arXiv:1804.07010*.

Ranganath, R., Gerrish, S., & Blei, D. (2014). Black box variational inference. In *Artificial intelligence and statistics* (pp. 814–822). PMLR.

Rudd, R., McWalter, T., Kienitz, J., & Platen, E. (2017). Fast quantization of stochastic volatility models. *Available at SSRN: 2956168*.

Ruf, J., & Wang, W. (2020). Neural networks for option pricing and hedging: a literature review. *Journal of Computational Finance, Forthcoming*.

Salvador, B., Oosterlee, C. W., & van der Meer, R. (2020). Financial option valuation by unsupervised learning with artificial neural networks. *Mathematics*, *9*(1), 46.

Samaniego, E., Anitescu, C., Goswami, S., Nguyen-Thanh, V., Guo, H., Hamdia, K., Zhuang, X., & Rabczuk, T. (2020). An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications. *Computer Methods in Applied Mechanics and Engineering*, *362*, 112790.

Schmallfuß, B. (1997). The random attractor of the stochastic Lorenz system. *Zeitschrift für angewandte Mathematik und Physik*, *48*(6), 951.

Shevchenko, P. (2003). Addressing the bias in Monte Carlo pricing of multi-asset options with multiple barriers through discrete sampling. *The Journal of Computational Finance*, *6*(3), 1–20.

Sirignano, J., & Spiliopoulos, K. (2018). DGM: A deep learning algorithm for solving partial differential equations. *Journal of Computational Physics*, *375*, 1339–1364.

Snelson, E., & Ghahramani, Z. (2005). Sparse Gaussian processes using pseudo-inputs. *Advances in Neural Information Processing Systems*, *18*, 1257–1264.

Towell, G. G., & Shavlik, J. W. (1994). Knowledge-based artificial neural networks. *Artificial Intelligence*, *70*(1–2), 119–165.

Vidales, M. S., Siska, D., & Szpruch, L. (2018). Unbiased deep solvers for parametric PDEs. *arXiv:1810.05094*.

Wilder, B., Ewing, E., Dilkina, B., & Tambe, M. (2019). End to end learning and optimization on graphs. *arXiv:1905.13732*.

Ye, T., & Zhang, L. (2019). Derivatives pricing via machine learning. *Journal of Mathematical Finance, 9, 561-589*.

Yoon, H., Park, C.-S., Kim, J. S., & Baek, J.-G. (2013). Algorithm learning based neural network integrating feature selection and classification. *Expert Systems with Applications*, *40*(1), 231–241.

Zang, Y., Bao, G., Ye, X., & Zhou, H. (2020). Weak adversarial networks for high-dimensional partial differential equations. *Journal of Computational Physics*, *411*, 109409.

Zhu, Y., Zabaras, N., Koutsourelakis, P.-S., & Perdikaris, P. (2019). Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. *Journal of Computational Physics*, *394*, 56–81.

How to cite this article: Becker, S., Jentzen, A., Müller, M. S., & von Wurstemberger, P. (2024). Learning the random variables in Monte Carlo simulations with stochastic gradient descent: Machine learning for parametric PDEs and financial derivative pricing. *Mathematical Finance*, *34*, 90–150. https://doi.org/10.1111/mafi.12405