THE UNIVERSITY
of EDINBURGH

Monte-Carlo based numerical methods for a class of non-local deterministic PDEs and several random PDE systems

Shuren Tan

THE UNIVERSITY
of EDINBURGH

Thesis submitted in fulfilment of
the requirements for the degree of
Doctor of Philosophy
to the
University of Edinburgh — 2019

Declaration

I declare that this thesis has been composed solely by myself and that it has not been submitted, either in whole or in part, in any previous application for a degree. Except where otherwise acknowledged, the work presented is entirely my own.

Shuren

October 2019

Lay Summary

PDE systems appear in modeling many different phenomena. Monte Carlo simulation provides a general methodology to represent their solutions approximately. This thesis concerns not only the theoretical numerical analysis but the efficiency of algorithms in practice, through numerous techniques in stochastic analysis, PDE theory, and machine learning.

Abstract

In this thesis, we will investigate McKean-Vlasov SDEs (McKV-SDEs) on $\mathbb{R}^d$,

$$\begin{cases} dX_t &= b[X_t, \mu_t^X]\, dt + \sigma[X_t, \mu_t^X]\, dW_t, \quad X_0 \sim \mu_0 \in \mathcal{P}_2(\mathbb{R}^d), \\ \mu_t^X &= \mathrm{Law}(X_t), \quad t \in [0,T], \end{cases}$$

where coefficient functions b and σ satisfy sufficient regularity conditions and $\{W_t\}_{t\in[0,T]}$ is a Wiener process. These SDEs correspond to a class of deterministic non-local PDEs

$$\partial_t \langle \mu_t, P\rangle = \langle \mu_t, \frac{1}{2}\sum_{i,j=1}^{d} \partial^2_{x_i,x_j} P(\cdot)\big(\sigma\sigma^T\big)_{ij}[\cdot,\mu_t] + \sum_{i=1}^{d} \partial_{x_i} P(\cdot) b_i[\cdot,\mu_t]\rangle,$$

where $\langle m, F\rangle := \int_{\mathbb{R}^d} F(y)\, m(dy)$.

The principal aim of the first part is to present Multilevel Monte Carlo (MLMC) schemes for the McKV-SDEs. To overcome challenges due to the dependence of coefficients on the measure, we work with Picard iteration. There are two different ways to proceed. The first way is to address the McKV-SDEs with interacting kernels directly by combining MLMC and Picard. The MLMC is used to represent the empirical densities of the mean-fields at each Picard step. This iterative MLMC approach reduces the computational complexity of calculating expectations by an order of magnitude.

However, we can also link the McKV-SDEs with interacting kernels to that with non-interacting kernels by projection and then iteratively solve the simpler by MLMC method. In each Picard iteration, the MLMC estimator can approximate a few mean-fields directly. This iterative MLMC approach via projection reduces the computational complexity of calculating expectations hugely by three orders of magnitude.

In the second part, the main purpose is to demonstrate the plausibility of applying deep learning technique to several types of random linear PDE systems. We design learning algorithms by using probabilistic representation and Feynman-Kac formula is crucial for deriving the recursive relationships on constructing the loss function in each training session.

Acknowledgements

I would first like to thank my supervisor, Dr. Lukasz Szpruch, for his constant encouragement and suggestions throughout my Ph.D. study. He taught me not only research but also professional skills, and I greatly appreciate the extra efforts he made to assist me. I would also like to thank my second supervisor, Goncalo dos Reis for his support and encouragement over the past years.

For all the research and non-research discussions, I would like to thank my research group member namely: Alvin Tse - for his collaboration on my very first paper, Denis Belomestny - for his participation in the project of projected MLMC, Marc Sabate Vidales, Aretha Teckentrup and David Siska - for their help on my last paper about deep learning. I would also like to express my special thanks to my friend Xiangrui Meng, for discussions on how to use the Eddie system.

Also, I give my gratitude to the financial support of the China Scholarship Council (CSC) and the School of Mathematics at the University of Edinburgh.

Finally, I am forever indebted to my parents (Yamin Tan and Yunhua Deng) and my family for their support and encouragement right through my study.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The main subject of this thesis is Monte-Carlo based methods. One is the Multilevel Monte Carlo (MLMC) for a class of deterministic PDEs with mean-fields and another is deep neural networks for several random linear PDEs. The mean field limits of systems of interacting diffusions (also called stochastic interacting particle systems (SIPS)) have been intensively studied since McKean [49] as they pave a way to probabilistic representations for many important nonlinear/nonlocal PDEs. The fact that particles are not independent render classical variance reduction techniques not directly applicable and consequently make simulations of interacting diffusions prohibitive.

In the third chapter, we include the published article [67] that provides an alternative iterative particle representation, inspired by the fixed point argument by Sznitman [65] after the presentation of the least background material and some frequently used results in the second chapter. The representation enjoys suitable conditional independence property that is leveraged in our analysis. We establish weak convergence of iterative particle system to the McKean-Vlasov SDEs (McKV-SDEs) with interacting kernels. One of the immediate advantages

of iterative particle system is that it can be combined with the Multilevel Monte Carlo (MLMC) approach for the simulation of McKV-SDEs. We proved that the MLMC approach reduces the computational complexity of calculating expectations by an order of magnitude. Another perspective on this work is that we analyse the error of nested Multilevel Monte Carlo estimators, which is of independent interest. Furthermore, we work with state dependent functionals, unlike scalar outputs which are common in literature on MLMC. The error analysis is carried out in uniform, and what seems to be new, weighted norms.

In the fourth chapter, we establish the framework for the McKV-SDEs with non-interacting kernels and present a generic methodology for efficient numerical approximation of the density function of the McKean-Vlasov SDEs with interacting kernels. Our approach is based on a combination of iterative MLMC particle method and projection estimation of particle densities. This allows to exploit smoothness of the coefficients for McKean-Vlasov SDEs. In the best-case scenario (i.e C^∞ for the coefficients), we improve the computational efficiency by three orders of magnitude.

In the final chapter, we study high-dimensional random Partial Differential Equations (PDEs). These might naturally induce the curse of dimensionality if using traditional numerical methods. We show how to construct deep neural network surrogates for high-dimensional random PDEs by extension of the construction in [62] and their effectiveness in numerous numerical experiments. Notably, the architecture of deep network surrogates is feed-forward fully-connected. The framework is mesh-free and can avoid the issue from the irregular computational domains. Variants of the Stochastic Gradient Descent (SGD) algorithm are applied to determine the parameters of the approximating deep neural network iteratively. Moreover, we study the efficiency of the deep net for the inverse problem of recovering the value of a parameter in the PDE given observations of the PDE solution. Finally, we investigate a particular random PDE system and demonstrate there is

a possibility that applying the approach of the weak-enforcement of the boundary condition would perform better than that of the hard.

Chapter 2

Preliminaries

In this chapter, we present some general background and briefly summarize some basic results from the theory of probability and stochastic differential equations which lay the mathematical foundation for later chapters. What we cover here is only small part of the comprehensive theory and they can be found in many standard textbooks such as [37], [41] and [55]. Then the later sections are

- section 2.1: we give the least required background of probability theory;
- section 2.2: we define the stochastic process, and especially Wiener process which lay the foundation for a class of stochastic integral in rest chapters;
- section 2.3: we outline inequalities and statements which will be used through whole thesis;

2.1 Basic Probability Background

Definition 2.1. *Let Ω be a set. Then a σ-algebra (or σ-field) $\mathscr{F}$ is a collection of subsets of such that*

1. $\Omega \in \mathscr{F}$.

2. *For every* $A_1, \ldots, A_i, \ldots$ *such that* $A_i \in \mathscr{F}$*, then* $\bigcup_i A_i \in \mathscr{F}$*.*

3. *If* $A \in \mathscr{F}$*, then* $A^c \in \mathscr{F}$*, where* $A^c = \Omega \backslash A$*.*

Definition 2.2. *A probability measure* $\mathbb{P}(\cdot)$ *defined on a sigma-algebra* $\mathscr{F}$ *of* Ω *is a functional mapping from* $\mathscr{F}$ *to* $[0,1]$ *and satisfies the following two properties:*

1. $\mathbb{P}(\Omega) = 1$*.*

2. *For every sequence of* $A_1, \ldots, A_i, \ldots$ *such that* $A_i \in \mathscr{F}$ *and* $A_i \cap A_j = \emptyset$ *for all* $i \neq j$*, then* $\mathbb{P}(\bigcup_i^{\infty} A_i) = \sum_{i=1}^{\infty} \mathbb{P}(A_i)$*.*

Definition 2.3. *The Polish space* Ω *is a complete separable metric space with metric* $\rho(x,y)$*. The closed ball of radius* r *centered at* x *is*

$$B_r(x) = \{y : \rho(x,y) \leq r\}.$$

Definition 2.4. *The Borel* σ*-field* $\mathscr{B}(\Omega)$ *is the smallest* σ*-field of subsets of* Ω *containing all closed balls. Elements of* $\mathscr{B}(\Omega)$ *are called Borel sets.*

Definition 2.5. *If* Ω_1 *and* Ω_2 *are Polish spaces, and* $f : \Omega_1 \to \Omega_2$ *, then the function* f *is called a Borel function if*

$$f^{-1}(B) := \{x : f(x) \in B\} \in \mathscr{B}(\Omega_1)$$

Definition 2.6. *Let* $(\Omega, \mathscr{F}, \mathbb{P})$ *be a probability space. An event* $E \in \mathscr{F}$ *happens almost surely (i.e. a.s) if* $\mathbb{P}(E) = 1$*.*

Definition 2.7. *Let* $(\Omega, \mathscr{F}, \mathbb{P})$ *be a probability space. Then, a d-dimensional real-valued function* $X : \Omega \to D \subset \mathbb{R}^d$ *is called random variable if* $\sigma(X) := X^{-1}(\mathscr{B}(D)) := \{X^{-1}(B) : B \in \mathscr{B}(D)\} \subset \mathscr{F}$*, where* D *is closed.*

Definition 2.8. *Let* X *be a real-valued random variable defined on the probability space* $(\Omega, \mathscr{F}, \mathbb{P})$ *and at least one of the integrals*

$$\int_\Omega X_+(w)\mathbb{P}(dw), \quad \int_\Omega X_-(w)\mathbb{P}(dw)$$

($X_\pm := (|X| \pm X)/2$) is finite, then the expectation of X is defined as

$$\mathbb{E}[X] := \int_\Omega X(w)\mathbb{P}(dw) := \int_\Omega X_+(w)\mathbb{P}(dw) - \int_\Omega X_-(w)\mathbb{P}(dw).$$

This expectation operator $\mathbb{E}[\cdot]$ has various properties such as *linearity*, i.e. $\mathbb{E}[X+Y] = \mathbb{E}[X] + \mathbb{E}[Y]$ if both $\mathbb{E}[X]$ and $\mathbb{E}[Y]$ are finite, and *monotonicity*, i.e. $\mathbb{E}[X] \leq \mathbb{E}[Y]$ if $X \leq Y$ $(a.s)$ and both $\mathbb{E}[X]$ and $\mathbb{E}[Y]$ exist.

Moreover, the following assertions for $\mathbb{E}[\cdot]$ hold:

Theorem 2.9 (Monotone Convergence Theorem)**.** *Consider a non-decreasing sequence of random variables $\{X_i\}_{i=1}^\infty$ defined on the probability space $(\Omega, \mathscr{F}, \mathbb{P})$, i.e. for every $i \in \mathbb{Z}^+ := \{1, 2, \ldots\}$, $X_i \leq X_{i+1}$ $(a.s)$ and $\lim_{i\to\infty} X_i = X$ $(a.s)$, then*

$$\lim_{i\to\infty} \mathbb{E}[X_i] = \mathbb{E}[X].$$

Theorem 2.10 (Dominated Convergence Theorem)**.** *Consider a sequence of random variables $\{X_i\}_{i=1}^\infty$ defined on the probability space $(\Omega, \mathscr{F}, \mathbb{P})$. Suppose for every $i \in \mathbb{Z}^+$, $|X_i| \leq Y$ $(a.s)$ and $\mathbb{E}[Y] < \infty$ and the limit $\lim_{i\to\infty} X_i := X$ exists, then*

$$\lim_{i\to\infty} \mathbb{E}[X_i] = \mathbb{E}[X].$$

Next, it is necessary to differentiate several modes of convergence for the infinite sequence of random variables $\{X_i\}_{i=1}^\infty$ defined on the probability space $(\Omega, \mathscr{F}, \mathbb{P})$.

Definition 2.11. *A sequence of random variables $\{X_i\}_{i=1}^\infty$* ***converges almost surely*** *(i.e.* ***converges a.s****) if $\mathbb{P}(\{w : \lim_{i\to\infty} X_i(w) = X(w)\}) = 1$. This is also called* ***convergence with probability one****.*

Definition 2.12. *A sequence of random variables $\{X_i\}_{i=1}^\infty$* ***converges in probability*** *to X if*

$$\lim_{n\to\infty} \mathbb{P}\big(|X_n - X| > \varepsilon\big) = 0, \quad \forall \varepsilon > 0.$$

Definition 2.13. *Suppose F_i and F are the cumulative distribution functions of real-valued random variables X_i and X, respectively. The sequence $\{X_i\}_{i=1}^\infty$ is said*

to ***converge in law*** *to* X *if* $\lim_{i\to\infty} F_i(x) = F(x)$*, for every* $x \in \mathbb{R}$ *at which* F *is continuous. This is also called* ***convergence in distribution****.*

2.2 Stochastic Processes and Stopping time

Definition 2.14. *A d-dimensional real-valued stochastic process in continuous time is a collection of random variables* $X := \{X_t : 0 \leq t < \infty\}$ *on the probability space* $(\Omega, \mathscr{F}, \mathbb{P})$ *, where we conveniently interpret the index* $t \in [0, \infty)$ *of* X_t *as time. For a fixed sample point* $w \in \Omega$*, the function* $t \to X_t(w); t \geq 0$ *is the realization of the process* X *with* w*. Note that we always assume* $X : [0, \infty) \times \Omega \to D \subset \mathbb{R}^d$*, where* D *is closed, is measurable in* (t, w) *with respect to the product* σ*-field* $\mathscr{F} \otimes \mathscr{B}(D)$*.*

Definition 2.15. *A d-dimensional real-valued stochastic process* X *with index* t *defined on* $[0, T] \subset [0, \infty)$ *is continuous if all its realizations are continuous functions on* $[0, T]$*.*

Definition 2.16. *A d-dimensional real-valued stochastic process* X *with index* t *defined on* $[0, T] \subset [0, \infty)$ *is càdlàg if all its realizations belongs to the set of all real-valued right-continuous functions on [0,T] which have finite left limits at each point* $t \in [0, T]$*.*

An important class of stochastic processes that can be use as a building block for other processes is Brownian motion (or Wiener process) and characterized as follows:

Definition 2.17 (Brownian motion)**.** *A d-dimensional real-valued continuous stochastic process* $W := \{W_t : t \in [0, \infty)\}$ *defined on the probability space* $(\Omega, \mathscr{F}, \mathbb{P})$ *is Brownian motion if it has the properties*

1. $W_0 = 0$, *a.s;*

2. *for every* $t_1, t_2 \geq 0$, *the increment* $B_{t_1+t_2} - B_{t_1}$ *is independent of* $\{B_u : 0 \leq u \leq t_1\}$, *and has a Gaussian distribution with mean 0 and variance* t_2.

Also, Brownian motion is an example of a more general class of stochastic process named by martingale:

Definition 2.18 (Martingale)**.** *A stochastic process* X_t *defined on the probability space* $(\Omega, \mathscr{F}, \mathbb{P})$ *equipped with the filtration* $\{\mathscr{F}_t\}_{t\geq 0}$ *is called a martingale with respect to* $\mathscr{F}_t$ *if it satisfies the following conditions:*

1. X_t *is* $\mathscr{F}_t$*-adapted;*

2. $\mathbb{E}|X_t| < \infty$ *for all* $t > 0$*;*

3. $\mathbb{E}[X_t|X_s] = X_s$ *for every* $t \geq s \geq 0$, *a.s.* .

Finally, we introduce definitions of stopping time and local martingale:

Definition 2.19 (Stopping time)**.** *A random time* τ *defined on the probability space* $(\Omega, \mathscr{F}, \mathbb{P})$ *equipped with the filtration* $\{\mathscr{F}_t\}_{t\geq 0}$ *is called a stopping time with respect to* $\mathscr{F}_t$ *if the event* $\{\tau \leq t\}$ *belongs to the* σ*-field* $\mathscr{F}_t$ *for every* $t \geq 0$.

Definition 2.20 (Local martingale)**.** *A stochastic process* X_t *defined on the probability space* $(\Omega, \mathscr{F}, \mathbb{P})$ *equipped with the filtration* $\{\mathscr{F}_t\}_{t\geq 0}$ *is called a local martingale with respect to* $\mathscr{F}_t$ *if there exists a sequence of* $\mathscr{F}_t$*-stopping times such that* $\tau_i \to \infty$, *a.s and* $X_{t\wedge\tau_i}$ *is a martingale with respect to* $\mathscr{F}_{t\wedge\tau_i}$ *for every* $i \in \mathbb{Z}^+$.

2.3 Necessary inequalities and statements

Theorem 2.21 (Hölder's inequality)**.** *For any real-valued random variables* X *and* Y *defined on the probability space* $(\Omega, \mathscr{F}, \mathbb{P})$, *then*

$$\mathbb{E}|XY| \leq \left(\mathbb{E}\left[|X|^p\right]\right)^{\frac{1}{p}} \left(\mathbb{E}\left[|Y|^q\right]\right)^{\frac{1}{q}},$$

where $1/p + 1/q = 1$ *if* $p \in (1, \infty)$.

Theorem 2.22 (Cauchy-Schwarz inequality)**.** *For any real-valued random variables* X *and* Y *defined on the probability space* $(\Omega, \mathscr{F}, \mathbb{P})$*, then*

$$|\mathbb{E}[XY]| \leq \left(\mathbb{E}\left[|X|^2\right]\right)^{\frac{1}{2}} \left(\mathbb{E}\left[|Y|^2\right]\right)^{\frac{1}{2}}.$$

Theorem 2.23 (Jensen's inequality)**.** *If* X *is an integrable* (*i.e.* $\mathbb{E}[X] < \infty$) *random variable defined on the probability space* $(\Omega, \mathscr{F}, \mathbb{P})$ *and* φ *is a convex function , then*

$$\varphi\left(\mathbb{E}[X]\right) \leq \mathbb{E}\left[\varphi(X)\right].$$

Lemma 2.24 (Conditional Jensen's inequality)**.** *If* X *is an integrable random variable defined on the probability space* $(\Omega, \mathscr{F}, \mathbb{P})$*,* φ *is a convex function and* $\mathscr{G} \subset \mathscr{F}$ *is a* σ*-field, then*

$$\varphi\left(\mathbb{E}[X|\mathscr{G}]\right) \leq \mathbb{E}\left[\varphi(X)|\mathscr{G}\right], a.s.$$

Theorem 2.25 (Burkholder-Davis-Gundy inequality)**.** *For any* $1 \leq p < \infty$ *there exist positive constants* c_p, C_p *that only depend on the choice of* p *such that for all local martingales* M *with* $M_0 = 0$*, maximum denoted by* $M_t^* := \sup_{s \leq t} |M_s|$ *and stopping times* τ *, the following inequality holds:*

$$c_p \mathbb{E}\left[[M]_\tau^{p/2}\right] \leq \mathbb{E}\left[(M_\tau^*)^p\right] \leq C_p \mathbb{E}\left[[M]_\tau^{p/2}\right],$$

where $[M]$ *denotes the quadratic variation of a process* M*.*

Lemma 2.26 (Gronwall's lemma)**.** *Let* $T > 0$ *and* α*,* β *and* u *be real-valued functions defined on* $[0, T]$*. Assume that* α *and* u *are continuous and that the negative part of* β *is integrable on every closed and bounded subinterval of* $[0, T]$*.*

If α is non-negative and if u satisfies the integral inequality

$$u(t) \leq \beta(t) + \int_0^t \alpha(s)u(s)ds, \quad \forall t \in [0,T],$$

then

$$u(t) \leq \beta(t) + \int_0^t \alpha(s)\beta(s)\exp(\int_s^t \alpha(r)dr)ds, \quad \forall t \in [0,T].$$

Theorem 2.27 (Conditional Dominated Convergence Theorem)**.** *Let X be an integrable random variable defined on the probability space $(\Omega, \mathscr{F}, \mathbb{P})$ and $\mathscr{G} \subset \mathscr{F}$ is a σ-field. Consider a sequence of random variables $\{X_i\}_{i=1}^{\infty}$ such that $|X_i| \leq Y$, a.s where $\mathbb{E}[Y] < \infty$ and $\lim_{i\to\infty} X_i = X$, a.s, then*

$$\lim_{i\to\infty} \mathbb{E}[X_i|\mathscr{G}] = \mathbb{E}[X|\mathscr{G}], a.s.$$

Theorem 2.28 (Fubini's theorem)**.** *Suppose Ω_1 and Ω_2 are σ-finite measure spaces, and suppose that $\Omega_1 \times \Omega_2$ is given the product measure (which is unique as Ω_1 and Ω_2 are σ-finite). If $f(w_1, w_2)$ is a measurable function on $\Omega_1 \times \Omega_2$ such that*

$$\int_{\Omega_1\times\Omega_2} |f(w_1,w_2)|\,\mu(w_1,w_2) < \infty,$$

then

$$\int_{\Omega_1}\left(\int_{\Omega_2} f(w_1,w_2)\,\mu_{\Omega_2}(w_2)\right)\mu_{\Omega_1}(w_1) = \int_{\Omega_2}\left(\int_{\Omega_1} f(w_1,w_2)\,\mu_{\Omega_1}(w_1)\right)\mu_{\Omega_2}(w_2).$$

Lemma 2.29. *Let $(\Omega_1, \mathscr{B}_1)$ and $(\Omega_2, \mathscr{B}_2)$ be complete separable metric spaces. Let X and Y be random variables in the same probability space $(\Omega, \mathscr{F}, \mathbb{P})$, with X taking values in $(\Omega_1, \mathscr{B}_1)$ and Y taking values in $(\Omega_2, \mathscr{B}_2)$. Let $\mathscr{G} \subseteq \mathscr{F}$ be a sub σ-algebra. Suppose that Y is $\mathscr{G}$-measurable. Then, for every Borel function $h : \Omega_1 \times \Omega_2 \to \mathbb{R}$ such that $\mathbb{E}|h(X,Y)| < +\infty$, there exists (a.s.) a probability measure $\mu_{X|\mathscr{G}} : \mathscr{B}_1 \times \Omega \to [0,1]$ (called the **regular conditional probability***

measure of X ***given*** $\mathscr{G}$*) such that*

$$\big(\mathbb{E}[h(X,Y)|\mathscr{G}]\big)(\omega) = \int_{M_1} h(x, Y(\omega))\, \mu_{X|\mathscr{G}}(dx, \omega).$$

Suppose that, in addition, X *is independent of* $\mathscr{G}$*. Then,* $\mu_{X|\mathscr{G}}(\cdot, \omega) = Law(X)$ *for* ω*-almost surely.*

Theorem 2.30 (Feynman-Kac formula)**.** *Suppose* f *is a continuous function,* c *is continuous in* (x,t) *and bounded in* x*,* σ *and* b *are Lipschitz continuous in* x*, continuous in* t*, and grow at most linearly in* x *and* $u : [0,T] \times \mathbb{R}^d \to \mathbb{R}$ *satisfies*

$$\max_{0\le t\le T} |u(t,x)| \le Ce^{p|x|^2}$$

for some constant $p > 0$ *sufficiently small, and the second-order linear PDE*

$$\begin{cases} \dfrac{\partial u}{\partial t} + \dfrac{1}{2}\displaystyle\sum_{i,j=1}^{d} a_{ij}(t,x)\frac{\partial^2 u}{\partial x_i \partial x_j} + \sum_{j=1}^{d} b_j(t,x)\frac{\partial u}{\partial x_j} + c(x,t)u = 0, \quad (t,x) \in [0,T] \times \mathbb{R}^d, \\ u(T,x) = f(x), \end{cases} \tag{2.1}$$

where a *given by* $\sigma(t,x)\sigma(t,x)^T$ *is uniformly positive definite. Then* u *admits the stochastic representation*

$$u(t,x) = \mathbb{E}[f(X_T^{x,t})e^{\int_t^T c(X_s^{x,t},s)ds}],$$

where the stochastic flow is determined by the dynamics

$$dX_s^{x,t} = b(X_s^{x,t}, s)ds + \sigma(X_s^{x,t}, s)dW_s, \quad s \in [t,T] \text{ and } X_t^{x,t} = x \in \mathbb{R}^d.$$

Chapter 3

Iterative Multilevel Particle Approximation for McKean-Vlasov SDEs with Interacting Kernels

3.1 Introduction

The theory of mean field interacting particle systems was pioneered by the work of H. McKean [49], where he gave a probabilistic interpretation of a class of nonlinear (due to the dependence on the coefficients of the solution itself) nonlocal PDEs. Probabilistic representation has an advantage, as it paves a way to Monte-Carlo approximation methods which are efficient in high dimensions. Fix $T > 0$. Let $\{W_t\}_{t\in[0,T]}$ be an r-dimensional Brownian motion on a filtered probability space $(\Omega, \{\mathcal{F}_t\}_t, \mathcal{F}, \mathbb{P})$. Consider continuous functions $b : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$, $\sigma : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^{d\otimes r}$ and their corresponding non-linear (in the sense of McKean)

stochastic differential equation (McKV-SDE with interacting kernel) given by

$$\begin{cases} dX_t &= b[X_t, \mu_t^X]\, dt + \sigma[X_t, \mu_t^X]\, dW_t, \\ \mu_t^X &= \text{Law}(X_t), \quad t \in [0,T], \end{cases} \tag{3.1}$$

where $X_0 \sim \mu_0 \in \mathcal{P}_2(\mathbb{R}^d)$ and $G[x,m] := \int_{\mathbb{R}^d} G(x,y)\, m(dy)$, for any $x \in \mathbb{R}^d$ and $m \in \mathcal{P}_2(\mathbb{R}^d)$ (square-integrable laws on $\mathbb{R}^d$). Notice that $\{X_t\}_{t\in[0,T]}$ is not necessarily a Markov process and hence it is not immediate what the corresponding backward Kolmogorov equation looks like. Nonetheless using Itô's formula with $P \in C_b^2(\mathbb{R}^d)$, one can derive corresponding nonlinear Kolmogorov-Fokker-Planck equation

$$\partial_t \langle \mu_t, P\rangle = \langle \mu_t, \frac{1}{2}\sum_{i,j=1}^{d} \partial^2_{x_i,x_j} P(\cdot)\big(\sigma\sigma^T\big)_{ij}[\cdot,\mu_t] + \sum_{i=1}^{d} \partial_{x_i} P(\cdot) b_i[\cdot,\mu_t]\rangle, \tag{3.2}$$

where $\langle m, F\rangle := \int_{\mathbb{R}^d} F(y)\, m(dy)$, [2, 19, 65]. The theory of propagation of chaos, [65], states that (3.1) arises as a limiting equation of the system of interacting diffusions $\{Y_t^{i,N}\}_{i=1,\dots,N}$ on $(\mathbb{R}^d)^N$ given by

$$\begin{cases} dY_t^{i,N} &= b[Y_t^{i,N}, \mu_t^{Y,N}]dt + \sigma[Y_t^{i,N}, \mu_t^{Y,N}]dW_t^i, \\ \mu_t^{Y,N} &:= \frac{1}{N}\sum_{i=1}^N \delta_{Y_t^{i,N}}, \quad t \geq 0, \end{cases} \tag{3.3}$$

where $\{Y_0^{i,N}\}_{i=1,\dots,N}$ are i.i.d samples with law μ_0 and $\{W_t^i\}_{i=1,\dots,N}$ are independent Brownian motions. It can be shown, under sufficient regularity conditions on the coefficients, that $\mu^{Y,N} \in \mathcal{P}_2(C([0,T],\mathbb{R}^d))$ converges in law to μ^X, see [52]. This is a not trivial result as the particles are not independent. Moreover, (3.3) can be interpreted as a first step towards numerical schemes for (3.1). To obtain a fully implementable algorithm one needs to study time discretisation of (3.1). As in seminal papers by Bossy and Talay [11, 12] we work with an Euler scheme. Take partition $\{t_k\}_k$ of $[0,T]$, with $t_k - t_{k-1} = h$ and define $\eta(t) := t_k$ if $t \in [t_k, t_{k+1})$.

The continuous Euler scheme reads

$$\overline{Y}_t^{i,N} = \overline{Y}_{t_k}^{i,N} + b[\overline{Y}_{\eta(t)}^{i,N}, \mu_{\eta(t)}^{\overline{Y},N}](t - t_k) + \sigma[\overline{Y}_{\eta(t)}^{i,N}, \mu_{\eta(t)}^{\overline{Y},N}](W_t^i - W_{t_k}^i)\,. \tag{3.4}$$

Note that due to interactions between discretised diffusions, implementation of (3.4) requires N^2 arithmetic operations at each step t_k of the scheme. This makes simulations of (3.4) very costly, but should not come as a surprise as the aim is to approximate non linear/non local PDEs (3.2) for which the deterministic schemes based on space discretisation, typically, are also computationally very demanding [13]. It has been proven that the empirical distribution function of N particles (3.4) converges, in a weak sense, to the distribution of the corresponding McKean-Vlasov limiting equation with the rate $O((\sqrt{N})^{-1} + h)$, see [2, 9, 10, 12]. Hence the computational cost of achieving a mean-square-error (see Theorem 3.19 for the definition) of order $\epsilon^2 > 0$ using this direct approach is $\mathcal{O}(\epsilon^{-5})$.

The lack of independence among interacting diffusions and the fact that the statistical error coming from approximating a measure creates a bias in the approximation, render applications of variance reduction techniques non-trivial. In fact, we are not aware of any rigorous work on variance reduction techniques for McKV-SDEs. In this chapter, we develop an iterated particle system that allows decomposing the statistical error and bias. We also provide an error analysis for a general class of McKV-SDEs. Finally, we deploy the MLMC method of Giles-Heinrich [29, 33] (see also 2-level MC of Kebaier [38]). In Section 3.2.1, we show that a direct application of MLMC to (3.3) fails. It is worth pointing out that the idea of combining an iterative method with MLMC to solve non-linear PDEs has very recently been proposed in [25]. However, their interest is on BSDEs and their connections to semi-linear PDEs.

The key technical part of this chapter is weak convergence analysis of the time discretisation that allows for iteration of the error in a suitable norms. It is

well know, at least since the work [68] that weak error analysis relies on the corresponding PDE theory. However as we already stated the solution to (3.1) is not Markovian on $\mathbb{R}^d$. To overcome we work with forward backward system

$$\begin{cases} X_t^{0,X_0} &= \xi + \int_0^t b[X_s^{0,X_0}, \mu_s^{X^{0,X_0}}]\, ds + \int_0^t \sigma[X_s^{0,X_0}, \mu_s^{X^{0,X_0}}]\, dW_s, \\ \mu_t^{X^{0,X_0}} &= \text{Law}(X_t^{0,X_0}), \end{cases}$$

and note that $X_t^{0,X_0} \neq X_t^{0,x}\big|_{x=X_0}$ in general (see [16]). This makes building of standard PDE theory on $[0,T] \times \mathbb{R}^d$ problematic and lead to theory of PDEs on measure spaces proposed by P. Lions in his lectures in Collège de France ([46]) and further developed in [16, 19]. Here we work with

$$\mathcal{X}_t^{0,x} = x + \int_0^t b[\mathcal{X}_s^{0,x}, \mu_s^{X^{0,\xi}}]\, ds + \int_0^t \sigma[\mathcal{X}_s^{0,x}, \mu_s^{X^{0,\xi}}] dW_s. \tag{3.5}$$

Notice that (3.5), unlike (3.1), is a Markov process. Furthermore, if (3.1) has a unique (weak) solution, then $\mathcal{X}_t^{0,x}|_{x=X_0} = X_t^{0,X_0}$. This means that

$$\int_{\mathbb{R}^d} \mathbb{E}\big[P(\mathcal{X}_t^{0,x})\big]\, \mu_0(dx) = \mathbb{E}\big[\mathbb{E}[P(X_t)|X_0]\big].$$

It can be shown that $v(0,x) = \mathbb{E}\big[P(\mathcal{X}_t^{0,x})\big]$ is a solution to backward Kolmogorov equation on $[0,T] \times \mathbb{R}^d$ which we will explore in this chapter.

3.1.1 Main assumptions on the McKean-Vlasov SDE

Here we state the assumptions needed for the analysis of equation (3.1).

Assumption 3.1.

(Ker-Reg) *The kernels b and σ belong to the sets $C_{b,b}^{2,1}(\mathbb{R}^d \times \mathbb{R}^d, \mathbb{R}^d) \cap C_{b,p}^{0,2}(\mathbb{R}^d \times \mathbb{R}^d, \mathbb{R}^d)$ and $C_{b,b}^{2,1}(\mathbb{R}^d \times \mathbb{R}^d, \mathbb{R}^{d\otimes r}) \cap C_{b,p}^{0,2}(\mathbb{R}^d \times \mathbb{R}^d, \mathbb{R}^{d\otimes r})$ respectively.*

(μ_0-L_p) *The initial law $\mu_0 := \mu_0^X$ satisfies the following condition: for any $p \geq 1$, $\mu_0 \in L^p(\Omega; \mathbb{R}^d)$, i.e.*

$$\int_{\mathbb{R}^d} |x|^p \mu_0(dx) < \infty.$$

Note that if **(Ker-Reg)** holds, then

(Lip) the kernels b and σ are globally Lipschitz, i.e. for all $x_1, x_2, y_1, y_2 \in \mathbb{R}^d$, there exists a constant L such that

$$|b(x_1, y_1) - b(x_2, y_2)| + \|\sigma(x_1, y_1) - \sigma(x_2, y_2)\| \leq L(|x_1 - x_2| + |y_1 - y_2|).$$

If **(Ker-Reg)** and **(μ_0-L_p)** hold, then a weak solution to (3.1) exists and pathwise uniqueness holds (see [65]). In other words $\{X_t\}_{t\geq 0}$ induces a unique probability measure on $C([0,T], \mathbb{R}^d)$. Furthermore it has a property that

$$\sup_{0\leq t\leq T} \mathbb{E}|X_t|^p < \infty. \tag{3.6}$$

The additional smoothness stipulated in **(Ker-Reg)** is needed in the analysis of weak approximation errors.

3.1.2 Iterated particle method

The main idea is to approximate (3.1) with a sequence of classical SDEs defined as

$$dX_t^m = b[X_t^m, \mu_t^{X^{m-1}}]dt + \sigma[X_t^m, \mu_t^{X^{m-1}}]dW_t^m, \quad \mu_0^{X^m} = \mu_0^X, \tag{3.7}$$

where (W^m, X_0^m) are independent for all $m \in \mathbb{N}$ as well as (W^m, X_0^m) and (W^n, X_0^n) $m \neq n \in \mathbb{N}$, are independent. The conditional independence across iterations is the key difference of our approach from the proof of existence of

solutions by Sznitman [65], where the same Brownian motion and initial condition are used at every iteration. The Euler scheme with $\mu_0^{\overline{X}^m} = \mu_0^X$ reads

$$d\overline{X}_t^m = b[\overline{X}_{\eta(t)}^m, \mu_{\eta(t)}^{\overline{X}^{m-1}}]dt + \sigma[\overline{X}_{\eta(t)}^m, \mu_{\eta(t)}^{\overline{X}^{m-1}}]dW_t^m\,. \tag{3.8}$$

To implement (3.8) at every step of the scheme, one needs to compute the integral with respect to the measure from the previous iteration $m-1$. This integral is calculated by approximating measure $\mu_{\eta(t)}^{\overline{X}^{m-1}}$ by the empirical measure with N_{m-1} samples. Consequently, we take $\mu_0^{\overline{Y}^{i,m}} = \mu_0^X$ and define, for $m \in \mathbb{N}$ and $1 \le i \le N_m$,

$$d\overline{Y}_t^{i,m} = b[\overline{Y}_{\eta(t)}^{i,m}, \mu_{\eta(t)}^{\overline{Y}^{m-1},N_{m-1}}]dt + \sigma[\overline{Y}_{\eta(t)}^{i,m}, \mu_{\eta(t)}^{\overline{Y}^{m-1},N_{m-1}}]dW_t^{i,m}, \tag{3.9}$$

and call it an *iterative particle system*. As above, we require that $W^{i,m}$, $1 \le i \le N_m$, $m \in \mathbb{N}$, and $\overline{Y}_0^{i,m}$, $1 \le i \le N_m$, $m \in \mathbb{N}$, are independent. By this construction, the particles $(\overline{Y}_t^{i,m})_i$ are independent upon conditioning on $\sigma\Big(\{\overline{Y}_t^{i,m-1}\}_{1\le i\le N_{m-1}} : t \in [0,T]\Big)$. The error analysis of (3.9) is presented in Theorem (3.19) and (3.20). From there one can deduce that optimal computational cost is achieved when $\{N_m\}_m$ is increasing and the computational complexity of computing expectations with (3.9) is of the same order as the original particle system, i.e. ϵ^{-5}.

3.1.3 Main result of the iterative MLMC algorithm

To reduce the computational cost, we combine the MLMC method with Picard iteration (3.7). Fix m and L. Let $\Pi^\ell = \{0 = t_0^\ell, \dots, t_k^\ell, \dots, T = t_{2^\ell}^\ell\}$, $\ell = 0, \dots, L$, be a family of time grids such that $t_k^\ell - t_{k-1}^\ell = h_\ell = T2^{-\ell}$. To simulate (3.8) at Picard step m and for all discretisation levels ℓ we need to have an approximation of the relevant expectations with respect to the law of the process at the previous

Picard step $m-1$ and the time grid Π^L, i.e.

$$\left(\mathbb{E}[b(x,\overline{X}_0^{m-1})],\dots,\mathbb{E}[b(x,\overline{X}_{t_k^L}^{m-1})],\dots,\mathbb{E}[b(x,\overline{X}_T^{m-1})]\right),$$
$$\left(\mathbb{E}[\sigma(x,\overline{X}_0^{m-1})],\dots,\mathbb{E}[\sigma(x,\overline{X}_{t_k^L}^{m-1})],\dots,\mathbb{E}[\sigma(x,\overline{X}_T^{m-1})]\right).$$

By approximating these expectations with the MLMC (signed) measure $\mathcal{M}^{(m-1)}$ (see Section 3.2.2 for its exact definition), we arrive at the *iterative MLMC particle* method defined as

$$dY_t^{i,m,\ell} = \langle \mathcal{M}^{(m-1)}_{\eta_\ell(t)}, b(Y^{i,m,\ell}_{\eta_\ell(t)},\cdot)\rangle\, dt + \langle \mathcal{M}^{(m-1)}_{\eta_\ell(t)}, \sigma(Y^{i,m,\ell}_{\eta_\ell(t)},\cdot)\rangle\, dW_t^{i,m}, \tag{3.10}$$

where $Y^{i,0,\ell} = X_0$. Under the assumptions listed in Section 3.1.1, the main result of this chapter gives precise error bounds for (3.10).

Theorem 3.2. *Assume* ***(Ker-Reg)*** *and* ***(μ_0-L_p)***. *Fix $M > 0$ and let $P \in C_b^2(\mathbb{R}^d)$. Define $MSE_t^{(M)}(P) := \mathbb{E}[(\langle \mathcal{M}_t^{(M)}, P\rangle - \mathbb{E}[P(X_t)])^2]$. Then there exists a constant $c>0$ (independent of the choices of M, L and $\{N_{m,\ell}\}_{m,\ell}$) such that for every $t\in[0,T]$,*

$$MSE^{(M)}_{\eta_L(t)}(P) \le c\left\{h_L^2 + \sum_{m=1}^{M}\frac{c^{M-m}}{(M-m)!}\cdot\sum_{\ell=0}^{L}\frac{h_\ell}{N_{m,\ell}} + \frac{c^{M-1}}{M!}\right\}.$$

The proof can be found in Section 3.4.1. The first term in the above error comes from the analysis of weak convergence for the Euler scheme. The second contains the usual MLMC variance and shows that computational effort should be increasing with with iteration m (rather than equally distributed across iterations). Finally the last term is an extra error due to iterations. Using this result, we prove in Theorem 3.17 that the overall complexity of the algorithm is of order $\epsilon^{-4}|\log\epsilon|^3$ (i.e. one order of magnitude better than the direct approach). We remark that the MLMC measure acts on functionals that depend on spatial

variables. We work with uniform norms as in [30, 33], but also introduce suitable weighted norms, which seems new in MLMC literature.

We remark that, the analysis of stochastic particles systems is of independent interest, as it is used as models in molecular dynamics; physical particles in fluid dynamics [57]; behaviour of interacting agents in economics or social networks [18] or interacting neurons in biology [22]. It is also used in modelling networks of neurons (see [21]) and modelling altruism (see [25]).

3.1.4 Convention of notations

We use $\|A\|$ to denote the Hilbert-Schmidt norm while $|\mathbf{v}|$ is used to denote the Euclidean norm. For any stochastic process $R = \{R_t\}_{t\in I}$, the law of R_t at any time point $t \in I$ is denoted by μ_t^R. $\mathcal{P}_2(E)$ denotes the set of square-integrable probability measures on any Polish space E. On the other hand, $\mathcal{P}_2^s(E)$ denotes, on any Polish space E, the set of random signed measures that are square-integrable almost surely.

Moreover, we denote by $C_{b,p}^{0,2}(\mathbb{R}^m \times \mathbb{R}^n, \mathbb{R})$ the set of functions P from $\mathbb{R}^m \times \mathbb{R}^n$ to $\mathbb{R}$ that are continuously twice-differentiable in the second argument, for which there exists a constant L such that for each $x \in \mathbb{R}^m$, $y \in \mathbb{R}^n$, $i, j \in \{1, \dots, n\}$,

$$|\partial_{y_i} P(x,y)| \le L(1+|y|^p), \qquad |\partial^2_{y_i,y_j} P(x,y)| \le L(1+|y|^p),$$

where ∂_{y_i} and $\partial^2_{y_i,y_j}$ denote respectively the first and second order partial derivatives w.r.t. the second argument. Finally, we denote by $C_{b,b}^{p,q}(\mathbb{R}^m \times \mathbb{R}^n, \mathbb{R})$ the set of functions from $\mathbb{R}^m \times \mathbb{R}^n$ to $\mathbb{R}$ that are continuously p times differentiable in the first argument and continuously q times differentiable in the second argument

such that the partial derivatives (up to the respective orders, excluding the "zeroth" order derivative) are bounded.

3.2 The iterative MLMC algorithm

3.2.1 Direct application of MLMC to interacting diffusions

There are two issues pertaining to the direct application of MLMC methodology to (3.4): i) the telescopic property needed for MLMC identity [29] does not hold in general; ii) a small number of simulations (particles) on fine time steps (a reason for the improved computational cost in MLMC setting) would lead to a poor approximation of the measure, leading to a high bias. To show that telescopic sum does not hold in general, consider a collection of discretisations of $[0,T]$ with different resolutions. To this end, we fix $L \in \mathbb{N}$. Then Y_T^{i,ℓ,N_ℓ}, $\ell = 1,\dots,L$, denotes for each i a particle corresponding to (3.4) with time-step h_ℓ, where N_ℓ is the total number of particles. Let $P : \mathbb{R}^d \to \mathbb{R}$ be any Borel-measurable function. With a direct application of MLMC in time for (3.4), we replace the standard Monte-Carlo estimator on the left-hand side by an MLMC estimator on the right-hand side as follows.

$$\begin{aligned} &\frac{1}{N_L}\sum_{i=1}^{N_L} P(Y_t^{i,L,N_L}) \\ \approx\ &\frac{1}{N_0}\sum_{i=1}^{N_0} P(Y_t^{i,0,N_0}) + \sum_{\ell=0}^{L}\frac{1}{N_\ell}\sum_{i=1}^{N_\ell}\Big[P(Y_t^{i,\ell,N_\ell}) - P(Y_t^{i,\ell-1,N_\ell})\Big]. \end{aligned} \tag{3.11}$$

However, we observe that such a direct application is not possible, since, in general,

$$\mathbb{E}\left[P(Y_t^{1,\ell,N_\ell})\right] \neq \mathbb{E}\left[P(Y_t^{1,\ell,N_{\ell+1}})\right],$$

which means that we do not have equality in expectation on both sides of (3.11). On the contrary, if we required the number of particles for all the levels to be the same, then the telescopic sum would hold, but clearly, there would be no computational gain from doing MLMC. We are aware of two articles that tackle the aforementioned issue. The case of linear coefficients is treated in [59], in which particles from all levels are used to approximate the mean field at the final (most accurate) approximation level. It is not clear how this approach could be extended to general McKean-Vlasov equations. A numerical study of a "multi-cloud" approach is presented in [31]. The algorithm resembles the MLMC approach to the nested simulation problem in [1, 30, 17, 45]. Their approach is very natural, but because particles within each cloud are not independent, one faces similar challenges as with the classical particle system.

3.2.2 Construction of the iterative MLMC algorithm

We approximate each of the expectations by the MLMC method, but only have access to samples at grid points Π^ℓ that correspond to $(Y^{i,m-1,\ell})_{i,\ell}$. Consequently, for $\ell < \ell'$, the empirical measure $\frac{1}{N}\sum_{i=1}^N \delta_{Y_t^{i,m-1,\ell}}$ is only defined at every timepoint in Π^ℓ, but not $\Pi^{\ell'}$ and one cannot build MLMC telescopic sum across all discretisation levels. For that reason (as in original development of MLMC by Heinrich [33]), we introduce a linear-interpolated measure (in time) $\widetilde{\mu}_t^{Y^{m-1,\ell},N}$

given by

$$\widetilde{\mu}_t^{Y^{m-1,\ell},N} := \begin{cases} \frac{1}{N}\sum_{i=1}^N \delta_{Y_t^{i,m-1,\ell}} & , t \in \Pi^\ell, \\ \left[\frac{t-\eta_\ell(t)}{h_\ell}\right]\widetilde{\mu}_{\eta_\ell(t)+h_\ell}^{Y^{m-1,\ell},N} + \left[1-\frac{t-\eta_\ell(t)}{h_\ell}\right]\widetilde{\mu}_{\eta_\ell(t)}^{Y^{m-1,\ell},N} & , t \notin \Pi^\ell\,, \end{cases} \tag{3.12}$$

where $\eta_\ell(t) := t_k^\ell, \quad \text{if } t \in [t_k^\ell, t_{k+1}^\ell)$. For any continuous function $P : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ and any $x \in \mathbb{R}^d$, we define the MLMC signed measure $\mathcal{M}_t^{(m-1)}$ by

$$\langle \mathcal{M}_t^{m-1}, P(x,\cdot)\rangle := \langle \sum_{\ell=0}^{L} (\widetilde{\mu}_t^{Y^{m-1,\ell},N_{m-1,\ell}} - \widetilde{\mu}_t^{Y^{m-1,\ell-1},N_{m-1,\ell}}), P(x,\cdot)\rangle\,, \tag{3.13}$$

where $\widetilde{\mu}_t^{Y^{m-1,-1},N_{m,0}} := 0$. We interpret the MLMC operator in a componentwise sense. We then define the particle system $\{Y^{i,m,\ell}\}$ as in (3.10). As usual for MLMC estimators, at each level ℓ, we use the same Brownian motion to simulate particle systems $(Y^{i,m,\ell}, Y^{i,m,\ell-1})_i$ to ensure that the variance of the overall estimator is reduced. As for the iterative particle system, we require that $W^{i,m}$, $1 \le i \le N_{m,\ell}$, $m \in \mathbb{N}$, and $Y_0^{i,m,\ell}$, $1 \le i \le N_{m,\ell}$, $1 \le \ell \le L$, $m \in \mathbb{N}$, are independent.

3.3 Abstract framework for MLMC analysis

To streamline the analysis of the iterated MLMC estimator, we introduce an abstract framework corresponding to one iteration. This simplifies the notation and also may be useful for future developments of MLMC algorithms.

Let $\overline{b} : \mathbb{R}^d \times \mathcal{P}_2^s(\mathbb{R}^d) \to \mathbb{R}^d$ and $\overline{\sigma} : \mathbb{R}^d \times \mathcal{P}_2^s(\mathbb{R}^d) \to \mathbb{R}^{d\otimes r}$ be measurable functions. Also, $\mathcal{V} \in \mathcal{P}_2^s(C([0,T],\mathbb{R}^d))$ is fixed (the precise conditions that we impose on $\overline{b}$, $\overline{\sigma}$ and $\mathcal{V}$ will be presented in Section 3.3.1). We consider SDEs with *random*

coefficients of the form

$$dU_t = \bar{b}(U_t, \mathcal{V}_t)dt + \bar{\sigma}(U_t, \mathcal{V}_t)dW_t, \quad \mu_0^U = \mu_0^X. \tag{3.14}$$

The solution of this SDE is well-defined under the assumptions in Section 3.3.1, by [41]. For $\ell = 1, \dots, L$, the corresponding Euler approximation of (3.14) at level ℓ is given by

$$dZ_t^\ell = \bar{b}(Z_{\eta_\ell(t)}^\ell, \mathcal{V}_{\eta_\ell(t)})dt + \bar{\sigma}(Z_{\eta_\ell(t)}^\ell, \mathcal{V}_{\eta_\ell(t)})dW_t, \quad \mu_0^{Z^\ell} = \mu_0^X. \tag{3.15}$$

We require that $\mathcal{V}$ does not depend on ℓ and that $(W_t)_{t\in[0,T]}$ is independent of $\mathcal{V}$. Subsequently, we define a particle system $\{Z^{i,\ell}\}$ as follows,

$$dZ_t^{i,\ell} = \bar{b}(Z_{\eta_\ell(t)}^{i,\ell}, \mathcal{V}_{\eta_\ell(t)})dt + \bar{\sigma}(Z_{\eta_\ell(t)}^{i,\ell}, \mathcal{V}_{\eta_\ell(t)})dW_t^i, \qquad \mu_0^{Z^{i,\ell}} = \mu_0^X. \tag{3.16}$$

3.3.1 Analysis of the abstract framework

Using the notation defined in the previous section, we formulate the conditions needed to study the convergence of the iterated particle system. Recall that $\mathcal{V} \in \mathcal{P}_2^s(C([0,T], \mathbb{R}^d))$ is given and we consider equations (3.15) and (3.16). We assume the following.

Assumption 3.3.

($\mathcal{V}$-bound) *The random measure $\mathcal{V}$ is independent of W^i and $Z_0^{i,\ell}$. For each $p \geq 1$,*

$$\sup_{0\leq s\leq T} \mathbb{E}\left| \int_{\mathbb{R}^d} |y|^p \mathcal{V}_s(dy) \right| < \infty.$$

($\mathcal{V}$-Reg) *There exists a constant c such that*

$$\sup_{x\in\mathbb{R}^d}\sup_{0\le s\le t\le T}\mathbb{E}\big[\big|\bar{b}(x,\mathcal{V}_t)-\bar{b}(x,\mathcal{V}_s)\big|^2+\big\|\overline{\sigma}(x,\mathcal{V}_t)-\overline{\sigma}(x,\mathcal{V}_s)\big\|^2\big]\le c(t-s).$$

($\mathcal{V}$-Lip) *There exists a constant c such that for each $t\in[0,T]$ and $x,y\in\mathbb{R}^d$,*

$$|\bar{b}(x,\mathcal{V}_t)-\bar{b}(y,\mathcal{V}_t)|+\|\overline{\sigma}(x,\mathcal{V}_t)-\overline{\sigma}(y,\mathcal{V}_t)\|\le c|x-y| \tag{3.17}$$

$$|\bar{b}(x,\mathcal{V}_t)|+\|\overline{\sigma}(x,\mathcal{V}_t)\|\le c\left(1+|x|+\left|\int_{\mathbb{R}^d}|y|\,\mathcal{V}_t(dy)\right|\right). \tag{3.18}$$

Analysis of conditional MLMC variance For the rest of this section, we denote by c a generic constant that depends on T, but not on ℓ or N_ℓ. We first consider the integrability of process (3.15).

Lemma 3.4. *Let Z^ℓ be defined as in (3.15). Assume **($\mathcal{V}$-Lip)** and **(μ_0-L_p)** . Then for any $p\ge 2$ and $\ell\ge 0$, there exists a constant c such that*

$$\mathbb{E}\left[\sup_{t\in[0,T]}|Z_t^\ell|^p\right]\le c\left(1+\mathbb{E}\left[\int_0^T\left|\int_{\mathbb{R}^d}|y|^p\mathcal{V}_{\eta_\ell(s)}(dy)\right|ds\right]\right).$$

Proof. Given any ℓ, let us define a sequence of stopping times $\tau_M:=\inf\{t\ge 0:|Z_t^\ell-Z_0^\ell|\ge M\}$. For any $t\in[0,T]$, we consider the stopped process $Z^\ell_{t\wedge\tau_M}$ and compute by the Burkholder-Davis-Gundy and Hölder inequalities and assumptions **($\mathcal{V}$-Lip)** and **(μ_0-L_p)** to obtain that

$$\begin{aligned}\mathbb{E}\left[\sup_{0\le u\le t}|Z^\ell_{u\wedge\tau_M}|^p\right] &\le c\left(\mathbb{E}[|Z_0^\ell|^p]+t^{p-1}\mathbb{E}\left[\int_0^t|\bar{b}(Z^\ell_{\eta(s)\wedge\tau_M},\mathcal{V}_{\eta(s)})|^p ds\right]\right.\\ &\quad\left.+t^{\frac{p}{2}-1}\mathbb{E}\left[\int_0^t\|\overline{\sigma}(Z^\ell_{\eta(s)\wedge\tau_M},\mathcal{V}_{\eta(s)})\|^p ds\right]\right).\\ &\le c\left(1+\mathbb{E}\left[\int_0^t\left|\int_{\mathbb{R}^d}|y|^p\mathcal{V}_{\eta(s)}(dy)\right|ds\right]\right.\end{aligned}$$

$$+\int_0^t \mathbb{E}\Big[\sup_{0\le u\le s}|Z^\ell_{u\wedge\tau_M}|^p\Big]\,ds\Big).$$

Note that, by $(\mu_0\text{-}L_p)$,

$$\mathbb{E}\Big[\sup_{0\le u\le s}|Z^\ell_{u\wedge\tau_M}|^p\Big] \le c\Big(\mathbb{E}\Big[\sup_{0\le u\le s}|Z^\ell_{u\wedge\tau_M}-Z^\ell_0|^p\Big]+\mathbb{E}|Z^\ell_0|^p\Big) < +\infty.$$

By Gronwall's lemma,

$$\mathbb{E}\Big[\sup_{0\le u\le t}|Z^\ell_{u\wedge\tau_M}|^p\Big] \le c\,\Big(1+\mathbb{E}\Big[\int_0^T\Big|\int_{\mathbb{R}^d}|y|^p\mathcal{V}_{\eta(s)}(dy)\Big|\,ds\Big]\Big).$$

Furthermore, since $\sup_{0\le t\le T}|Z^\ell_{t\wedge\tau_M}|^p$ is a non-decreasing sequence (in M) converging pointwise to $\sup_{0\le t\le T}|Z^\ell_t|^p$, the lemma follows from the monotone convergence theorem. □

The following two lemmas focus on the regularity of Z^ℓ_t in time and its strong convergence property. The first lemma bounds the difference in Z^ℓ_t over two time points, at a fixed level ℓ. The second lemma bounds the difference in Z^ℓ_t over adjacent levels, at a fixed time t. Their proofs follow from standard estimates in the theory of SDE and are therefore omitted.

Lemma 3.5 (Regularity of Z^ℓ_t)**.** *Let Z^ℓ be defined as in* (3.15)*. Assume* ***($\mathcal{V}$-Lip)*** *and* ***($\mathcal{V}$-bound)*** *. Then, for $p \ge 1$, $0 \le u \le s \le T$,*

$$\Big(\mathbb{E}[|Z^\ell_s - Z^\ell_u|^p]\Big)^{\frac{1}{p}} \le c(s-u)^{\frac{1}{2}}.$$

Lemma 3.6 (Strong convergence of Z^ℓ_t)**.** *Assume* ***($\mathcal{V}$-Lip)***, ***($\mathcal{V}$-bound)*** *and* ***($\mathcal{V}$-Reg)*** *. Then for any $\ell \in \{1,2,\dots,L\}$, there exists a constant $c>0$ such that*

$$\mathbb{E}\Big[\sup_{0\le t\le T}|Z_t^\ell - Z_t^{\ell-1}|^2\Big] \le ch_\ell.$$

We define the interpolated empirical measures $\widetilde{\mu}_t^{Z^\ell,N}$ exactly as in (3.12) and the corresponding MLMC operator $\mathcal{M}_t$ (corresponding to (3.13), but for one Picard iteration) as

$$\langle \mathcal{M}_t, P(x,\cdot)\rangle = \left\langle \sum_{\ell=0}^{L}\left(\widetilde{\mu}_t^{Z^\ell,N_\ell} - \widetilde{\mu}_t^{Z^{\ell-1},N_\ell}\right), P(x,\cdot)\right\rangle, \qquad \widetilde{\mu}_t^{Z^{-1},N_0} := 0.$$

We also define σ-algebra $\mathcal{F}_t^{\mathcal{V}} = \{\sigma(\mathcal{V}_s)_{0\le s\le t}\}$. Since samples $\{Z^{i,\ell}_{\eta_L(t)}\}$, $i = 1,\dots,N_\ell$, $\ell = 0,\dots,L$, conditioned on $\mathcal{F}_T^{\mathcal{V}}$ are independent, we can bound the conditional MLMC variance as follows.

Lemma 3.7. *Assume* ***($\mathcal{V}$-Lip)***, ***($\mathcal{V}$-bound)*** *and* ***($\mathcal{V}$-Reg)*** *hold. Let $\mu \in \mathcal{P}_2\big(C([0,T],\mathbb{R}^d)\big)$. Then for any Lipschitz function $P:\mathbb{R}^d\times\mathbb{R}^d\to\mathbb{R}$, there exists a constant c such that*

$$\sup_{0\le t\le T}\int_{\mathbb{R}^d}\mathbb{E}\left[\mathrm{Var}\left(\langle \mathcal{M}_{\eta_L(t)}, P(x,\cdot)\rangle\middle|\mathcal{F}_T^{\mathcal{V}}\right)\right]\mu_t(dx) \le c\sum_{\ell=0}^{L}\frac{h_\ell}{N_\ell}. \tag{3.19}$$

Proof. The independence condition in **($\mathcal{V}$-bound)** implies that

$$\begin{aligned}&\mathbb{E}\left[\mathrm{Var}\left(\langle \mathcal{M}_{\eta_L(t)}, P(x,\cdot)\rangle\middle|\mathcal{F}_T^{\mathcal{V}}\right)\right]\\ =\ &\sum_{i=1}^{N_0}\frac{1}{N_0^2}\mathbb{E}\left[\mathrm{Var}\left[P^{i,0}_{\eta_L(t)}\middle|\mathcal{F}_T^{\mathcal{V}}\right]\right] + \sum_{\ell=1}^{L}\sum_{i=1}^{N_\ell}\frac{1}{N_\ell^2}\mathbb{E}\left[\mathrm{Var}\left[P^{i,\ell}_{\eta_L(t)} - P^{i,\ell-1}_{\eta_L(t)}\middle|\mathcal{F}_T^{\mathcal{V}}\right]\right],\end{aligned}$$

where

$$P^{i,\ell}_{\eta_L(t)} := (1-\lambda_t^\ell)P(x, Z^{i,\ell}_{\eta_\ell(\eta_L(t))}) + \lambda_t^\ell P(x, Z^{i,\ell}_{\eta_\ell(\eta_L(t))+h_\ell}), \tag{3.20}$$

$\lambda_t^\ell = \frac{\eta_L(t)-\eta_\ell(\eta_L(t))}{h_\ell} \in [0,1]$. Using the fact that $\mathbb{E}[\mathrm{Var}(X|\mathcal{G})] \le \mathrm{Var}(X) \le \mathbb{E}[X^2]$,

we obtain the bound

$$\mathbb{E}\left[\mathrm{Var}\left(\langle \mathcal{M}_{\eta_L(t)}, P(x,\cdot)\rangle \middle| \mathcal{F}_T^{\mathcal{V}}\right)\right] \le \sum_{i=1}^{N_0} \frac{1}{N_0^2}\mathbb{E}\left|P^{i,0}_{\eta_L(t)}\right|^2 + \sum_{\ell=1}^{L}\sum_{i=1}^{N_\ell} \frac{1}{N_\ell^2}\mathbb{E}\left|P^{i,\ell}_{\eta_L(t)} - P^{i,\ell-1}_{\eta_L(t)}\right|^2.$$

Since P is Lipschitz, it has linear growth. By Lemma 3.4, it follows that

$$\mathbb{E}\left|P^{i,0}_{\eta_L(t)}\right|^2 \le c \sup_{0\le t\le T}\int_{\mathbb{R}^d}\left(x^2 + \mathbb{E}|Z^{i,0}_{\eta_0(\eta_L(t))}|^2 + \mathbb{E}|Z^{i,0}_{\eta_0(\eta_L(t))+h_0}|^2\right)\mu_t(dx) < +\infty.$$

Next, we consider levels $\ell \in \{1,\dots,L\}$. Recall from (3.20) that

$$\begin{aligned}
P^{i,\ell}_{\eta_L(t)} &= (1-\lambda_t^\ell)P(x, Z^{i,\ell}_{\eta_\ell(\eta_L(t))}) + \lambda_t^\ell P(x, Z^{i,\ell}_{\eta_\ell(\eta_L(t))+h_\ell}),\\
P^{i,\ell-1}_{\eta_L(t)} &= (1-\lambda_t^{\ell-1})P(x, Z^{i,\ell-1}_{\eta_{\ell-1}(\eta_L(t))}) + \lambda_t^{\ell-1} P(x, Z^{i,\ell-1}_{\eta_{\ell-1}(\eta_L(t))+h_{\ell-1}}).
\end{aligned}$$

We decompose the error as follows.

$$\begin{aligned}
&|P^{i,\ell}_{\eta_L(t)} - P^{i,\ell-1}_{\eta_L(t)}|\\
\le\ & (1-\lambda_t^{\ell-1})\cdot\left|P(x, Z^{i,\ell}_{\eta_\ell(\eta_L(t))}) \pm P(x, Z^{i,\ell-1}_{\eta_\ell(\eta_L(t))}) - P(x, Z^{i,\ell-1}_{\eta_{\ell-1}(\eta_L(t))})\right|\\
&+\lambda_t^{\ell-1}\cdot\left|P(x, Z^{i,\ell}_{\eta_\ell(\eta_L(t))+h_\ell}) \pm P(x, Z^{i,\ell-1}_{\eta_\ell(\eta_L(t))+h_\ell}) - P(x, Z^{i,\ell-1}_{\eta_{\ell-1}(\eta_L(t))+h_{\ell-1}})\right|\\
&+|\lambda_t^\ell - \lambda_t^{\ell-1}|\cdot\left|P(x, Z^{i,\ell}_{\eta_\ell(\eta_L(t))+h_\ell}) - P(x, Z^{i,\ell}_{\eta_\ell(\eta_L(t))})\right|.
\end{aligned}$$

By Lemma 3.6,

$$\mathbb{E}|P(x, Z^{i,\ell}_{\eta_\ell(\eta_L(t))}) - P(x, Z^{i,\ell-1}_{\eta_\ell(\eta_L(t))})|^2 \le ch_\ell, \tag{3.21}$$

$$\mathbb{E}|P(x, Z^{i,\ell}_{\eta_\ell(\eta_L(t))+h_\ell}) - P(x, Z^{i,\ell-1}_{\eta_\ell(\eta_L(t))+h_\ell})|^2 \le ch_\ell. \tag{3.22}$$

Also, by Lemma 3.5,

$$\mathbb{E}|P(x, Z^{i,\ell-1}_{\eta_\ell(\eta_L(t))}) - P(x, Z^{i,\ell-1}_{\eta_{\ell-1}(\eta_L(t))})|^2 \le c(\eta_\ell(\eta_L(t)) - \eta_{\ell-1}(\eta_L(t))) \le ch_\ell, \tag{3.23}$$

$$\mathbb{E}|P(x, Z^{i,\ell-1}_{\eta_\ell(\eta_L(t))+h_\ell}) - P(x, Z^{i,\ell-1}_{\eta_{\ell-1}(\eta_L(t))+h_{\ell-1}})|^2 \le ch_\ell, \tag{3.24}$$

and

$$\mathbb{E}|P(x, Z^{i,\ell}_{\eta_\ell(\eta_L(t))+h_\ell}) - P(x, Z^{i,\ell}_{\eta_\ell(\eta_L(t))})|^2 \leq c h_\ell. \tag{3.25}$$

We obtain (3.19) by combining (3.21), (3.22), (3.23), (3.24) and (3.25). Since t and x are arbitrary, the proof is complete.

□

3.3.2 Weak error analysis

We begin this subsection by defining $\mathcal{X}^{s,x}$ as

$$\mathcal{X}^{s,x}_t = x + \int_s^t b[\mathcal{X}^{s,x}_u, \mu^X_u]\, du + \int_s^t \sigma[\mathcal{X}^{s,x}_u, \mu^X_u]\, dW_u.$$

For $P \in C^{0,2}_{b,b}(\mathbb{R}^d \times \mathbb{R}^d, \mathbb{R})$ and $t \in [0,T]$, we consider the function

$$v_y(s,x) := \mathbb{E}[P(y, \mathcal{X}^{s,x}_t)], \quad y \in \mathbb{R}^d \text{ and } (s,x) \in [0,t] \times \mathbb{R}^d. \tag{3.26}$$

We aim to show that $v_y(s,x) \in C^{1,2}$. The first step is the lemma below.

Lemma 3.8. *Assume* ***(μ_0-L_p)*** *and* ***(Ker-Reg)*** *. Then*

$$b[\cdot, \mu^X_\cdot] \in C^{2,1}_{b,b}(\mathbb{R}^d \times [0,T], \mathbb{R}^d) \text{ and } \sigma[\cdot, \mu^X_\cdot] \in C^{2,1}_{b,b}(\mathbb{R}^d \times [0,T], \mathbb{R}^{d\otimes r}).$$

Proof. For any $x \in \mathbb{R}^d$, $s \in [0,T]$ and $t \in [s,T]$, we apply Itô's formula to each coordinate $k \in \{1, \ldots, d\}$ of b to get

$$\begin{aligned} b_k(x, X_t) = b_k(x, X_s) &+ \int_s^t \sum_{j=1}^d \sum_{i=1}^r \partial_{y_j} b_k(x, X_u) \sigma_{ji}[X_u, \mu^X_u] dW^i_u \\ &+ \int_s^t \sum_{j=1}^d \partial_{y_j} b_k(x, X_u) b_j[X_u, \mu^X_u] du + \frac{1}{2} \int_s^t \sum_{i,j=1}^d \partial^2_{y_i, y_j} b_k(x, X_u) a_{ij}[X_u, \mu^X_u] du, \end{aligned} \tag{3.27}$$

where $a[x,\mu] = \sigma[x,\mu]\sigma[x,\mu]^T$ and $\partial_{y_i} b_k, \partial^2_{y_i,y_j} b_k$ indicate the derivatives w.r.t. the the second argument. Assumptions **(Ker-Reg)** , **(Lip)** , (μ_0-L_p) and (3.6) imply that the above stochastic integral is a martingale. By the fundamental theorem of calculus,

$$\partial_t \mathbb{E}[b_k(x, X_t)] = \mathbb{E}\bigg[\sum_{j=1}^{d} \partial_{y_j} b_k(x, X_t) b_j[X_t, \mu_t^X] + \frac{1}{2}\sum_{i,j=1}^{d} \partial^2_{y_i,y_j} b_k(x, X_t) a_{ij}[X_t, \mu_t^X]\bigg]. \tag{3.28}$$

By **(Ker-Reg)** , $\partial_{y_j} b_k$ and $\partial^2_{y_i,y_j} b_k$ are bounded. Moreover, by **(Lip)** , we know that b and a are respectively of linear and quadratic growth in x. Therefore, by (3.6), we conclude that $\partial_t b_k[x, \mu_t^X]$ is bounded. To conclude, we can apply the same argument to $\sigma[\cdot, \mu_\cdot^X]$. □

Lemma 3.9. *Assume* ***(Ker-Reg)*** *and* ***(μ_0-L_p)*** *. Then for any $(s,x) \in [0,t] \times \mathbb{R}^d$, $(i,j) \in \{1,\dots,d\}^2$ and $P \in C^{0,2}_{b,b}(\mathbb{R}^d \times \mathbb{R}^d, \mathbb{R})$,*

$$\sup_{y \in \mathbb{R}^d} (\|\partial_{x_i} v_y(s,x)\|_\infty + \|\partial^2_{x_i,x_j} v_y(s,x)\|_\infty) \le L. \tag{(**v-diff-Reg+**)}$$

Proof. We only provide a sketch as the argument is standard. By the fact that the first-order spatial derivatives of $b[\cdot, \mu_\cdot^X]$ and $\sigma[\cdot, \mu_\cdot^X]$ are bounded, it is straightforward to deduce that

$$\sup_{x \in \mathbb{R}^d} \sup_{s \in [0,t]} \mathbb{E}\bigg[\Big|\partial_{x_i} (\mathcal{X}_t^{s,x})^{(j)}\Big|^2\bigg] < \infty. \tag{3.29}$$

Theorem 5.5.5 in [27] establishes that

$$\partial_{x_i} v_y(s,x) = \sum_{j=1}^{d} \mathbb{E}\Big[\partial_{y_j} P(y, \mathcal{X}_t^{s,x}) \partial_{x_i} (\mathcal{X}_t^{s,x})^{(j)}\Big]. \tag{3.30}$$

By (3.29), it is clear that the assertion for the first order derivatives in ((**v-diff-Reg+**)) holds if $P \in C^{0,2}_{b,b}(\mathbb{R}^d \times \mathbb{R}^d, \mathbb{R})$. Similarly, we can prove the assertion for the second order derivatives in the same way. □

Lemma 3.10. *Let $\{Q_t\}_{t\in[0,T]}$ be a cadlag square-integrable process adapted to the filtration $\{\mathcal{F}_t\}_{t\in[0,T]}$. Suppose that $\{W_t\}_{t\in[0,T]}$ is a $\{\mathcal{F}_t\}_{t\in[0,T]}$-Brownian motion. Let $\mathcal{G}$ be a σ-algebra such that $\mathcal{G} \subseteq \mathcal{F}_0$. Then the following equalities hold for any $t \in [0,T]$.*

$$\begin{aligned} &(a) \quad \mathbb{E}\left[\int_0^t Q_s dW_s \,\middle|\, \mathcal{G}\right] = 0, \\ &(b) \quad \mathbb{E}\left[\left(\int_0^t Q_s\, dW_s\right)^2 \middle| \mathcal{G}\right] = \mathbb{E}\left[\int_0^t Q_s^2\, ds \middle| \mathcal{G}\right]. \end{aligned}$$

Proof. (a) By the tower property of conditional expectation,

$$\mathbb{E}\left[\int_0^t Q_s dW_s \,\middle|\, \mathcal{G}\right] = \mathbb{E}\left[\mathbb{E}\left[\int_0^t Q_s dW_s \,\middle|\, \mathcal{F}_0\right] \middle| \mathcal{G}\right] = 0.$$

(b) Similarly, by Itô's isometry and the tower property of conditional expectation,

$$\begin{aligned} \mathbb{E}\left[\left(\int_0^t Q_s\, dW_s\right)^2 \middle| \mathcal{G}\right] &= \mathbb{E}\left[\mathbb{E}\left[\left(\int_0^t Q_s\, dW_s\right)^2 \middle| \mathcal{F}_0\right] \middle| \mathcal{G}\right] \\ &= \mathbb{E}\left[\mathbb{E}\left[\int_0^t Q_s^2\, ds \,\middle|\, \mathcal{F}_0\right] \middle| \mathcal{G}\right] \\ &= \mathbb{E}\left[\int_0^t Q_s^2\, ds \middle| \mathcal{G}\right]. \end{aligned}$$

□

By the Feynman-Kac theorem ([40]), it can be shown that $v_y(\cdot,\cdot)$ satisfies the

following Cauchy problem,

$$\begin{cases} \partial_s v_y(s,x) + \frac{1}{2}\sum_{i,j=1}^{d}\Big(\sigma[x,\mu_s^X]\sigma[x,\mu_s^X]^T\Big)_{ij}\partial^2_{x_i,x_j}v_y(s,x) \\ \qquad\qquad + \sum_{j=1}^{d}\Big(b[x,\mu_s^X]\Big)_j\partial_{x_j}v_y(s,x) &= 0, \quad (s,x)\in[0,t]\times\mathbb{R}^d, \\ v_y(t,x) &= P(y,x). \end{cases} \tag{3.31}$$

The following theorem reveals the order of weak convergence of (3.15) to (3.1). We denote by $\mu_t^{Z^\ell|\mathcal{F}_T^{\mathcal{V}}}$ the regular conditional probability measure of Z_t^ℓ given $F_T^{\mathcal{V}}$. (Recall Lemma 2.29 and see Theorem 7.1 in [56] for details.) The existence of regular conditional probability measure follows from the fact that we work on a Polish space with the Borel σ-algebra.

Theorem 3.11. *Let $P \in C_{b,b}^{0,2}(\mathbb{R}^d\times\mathbb{R}^d,\mathbb{R})$ be a Lipschitz continuous function.* [1] *Assume that* ***(Ker-Reg)*** *,* ***(μ_0-L_p)*** *,* ***($\mathcal{V}$-bound)*** *and* ***($\mathcal{V}$-Lip)*** *hold. Then there exists a constant c (independent of the choices of L and $N_1,\dots,N_L$) such that for each $t\in[0,T]$, $\ell\in\{0,\dots,L\}$ and $x\in\mathbb{R}^d$,*

$$\begin{aligned} &\sup_{0\le s\le t}|\mathbb{E}[P(x,Z_s^\ell)]-\mathbb{E}[P(x,X_s)]| \\ &\quad\le c\Bigg(h_\ell + \int_0^t \mathbb{E}\Bigg[\int_{\mathbb{R}^d}\Big|\overline{b}(x,\mathcal{V}_{\eta_\ell(s)})-\mathbb{E}[b(x,X_{\eta_\ell(s)})]\Big|\,\mu_{\eta_\ell(s)}^{Z^\ell|\mathcal{F}_T^{\mathcal{V}}}(dx)\Bigg]ds \\ &\qquad + \int_0^t \mathbb{E}\Bigg[\int_{\mathbb{R}^d}\Big\|\overline{\sigma}(x,\mathcal{V}_{\eta_\ell(s)})-\mathbb{E}[\sigma(x,X_{\eta_\ell(s)})]\Big\|\,\mu_{\eta_\ell(s)}^{Z^\ell|\mathcal{F}_T^{\mathcal{V}}}(dx)\Bigg]ds\Bigg). \end{aligned}$$

Proof. To lighten the notation in this proof, we use t_k, $\eta(s)$ and Z to denote t_k^ℓ, $\eta_\ell(s)$ and Z^ℓ respectively. First, we observe that

$$|\mathbb{E}[P(y,Z_s)]-\mathbb{E}[P(y,X_s)]| \le \mathbb{E}|\mathbb{E}[P(y,Z_s)|\mathcal{F}_T^{\mathcal{V}}]-\mathbb{E}[P(y,X_s)]|.$$

From definition of $v(\cdot,\cdot)$ in (3.26), we compute that

[1] *Note that the regularity of P can be relaxed to $C_{b,p}^{0,2}(\mathbb{R}^d\times\mathbb{R}^d,\mathbb{R})$. We prove the result in a slightly stronger assumption for the sake of simplicity.*

$$\mathbb{E}[v_y(0,X_0)] = \int_{\mathbb{R}^d} v_y(0,x)\,\mu_0(dx) = \int_{\mathbb{R}^d} \mathbb{E}\big[P(y,\mathcal{X}_t^{0,x})\big]\,\mu_0(dx) = \mathbb{E}\big[\mathbb{E}[P(y,X_t)|X_0]\big].$$

The Feynman-Kac theorem, hypothesis **($\mathcal{V}$-bound)** and the fact that $\mu_0^X = \mu_0^Z$ give

$$\begin{aligned}\mathbb{E}[P(y,Z_t)|\mathcal{F}_T^{\mathcal{V}}] - \mathbb{E}[P(y,X_t)] &= \mathbb{E}[v_y(t,Z_t)|\mathcal{F}_T^{\mathcal{V}}] - \mathbb{E}[v_y(0,Z_0)] \\ &= \mathbb{E}[v_y(t,Z_t)|\mathcal{F}_T^{\mathcal{V}}] - \mathbb{E}[v_y(0,Z_0)|\mathcal{F}_T^{\mathcal{V}}] \\ &= \sum_{k=0}^{n-1} \mathbb{E}\big[v_y(t_{k+1},Z_{k+1}) - v_y(t_k,Z_k)\,\big|\,\mathcal{F}_T^{\mathcal{V}}\big],\end{aligned}$$

where $n = t/h_\ell$ [2]. By Itô's formula,

$$\begin{aligned}&\mathbb{E}[v_y(t,Z_t)|\mathcal{F}_T^{\mathcal{V}}] - \mathbb{E}[v_y(0,Z_0)] \\ =\ &\sum_{k=0}^{n-1} \mathbb{E}\Bigg[\int_{t_k}^{t_{k+1}} \Bigg(\partial_t v_y(s,Z_s) + \sum_{j=1}^d \partial_{x_j} v_y(s,Z_s)\bar{b}_j(Z_{\eta(s)},\mathcal{V}_{\eta(s)}) \\ &+\frac{1}{2}\sum_{i,j=1}^d \partial^2_{x_i,x_j} v_y(s,Z_s)\bar{a}_{ij}(Z_{\eta(s)},\mathcal{V}_{\eta(s)})\Bigg)ds \\ &+\int_{t_k}^{t_{k+1}} \sum_{j=1}^d\sum_{i=1}^r \partial_{x_j} v_y(s,Z_s)\bar{\sigma}_{ji}(Z_{\eta(s)},\mathcal{V}_{\eta(s)})dW_s^{(i)}\,\Bigg|\,\mathcal{F}_T^{\mathcal{V}}\Bigg],\end{aligned}$$

where $\bar{a}(x,\mu) = \bar{\sigma}(x,\mu)\bar{\sigma}(x,\mu)^T$. Condition (**(v-diff-Reg+)**), as well as hypotheses **(Lip)** , **(μ_0-L_p)** and **($\mathcal{V}$-bound)** , along with Lemma 3.4 and part (a) of Lemma 3.10 (with the filtration $\{\mathcal{F}_t\}_{t\in[0,T]}$ such that $\mathcal{F}_t = \sigma(\mathcal{F}_T^{\mathcal{V}}, \{W_u\}_{0\le u\le t}, \{Z_u\}_{0\le u\le t})$) imply that

$$\mathbb{E}\Bigg[\int_{t_k}^{t_{k+1}} \sum_{j=1}^d\sum_{i=1}^r \partial_{x_j} v_y(s,Z_s)\bar{\sigma}_{ji}(Z_{\eta(s)},\mathcal{V}_{\eta(s)})dW_s^{(i)}\,\Bigg|\,\mathcal{F}_T^{\mathcal{V}}\Bigg] = 0. \tag{3.32}$$

Subsequently, using the fact that $v(\cdot,\cdot)$ satisfies PDE (3.31), we have

$$\mathbb{E}[v_y(t,Z_t)|\mathcal{F}_T^{\mathcal{V}}] - \mathbb{E}[v_y(0,Z_0)]$$

[2] For simplicity we assume that n is an integer.

$$= \sum_{k=0}^{n-1} \int_{t_k}^{t_{k+1}} \mathbb{E}\Big[\sum_{j=1}^{d} \partial_{x_j} v_y(s, Z_s)(\bar{b}_j(Z_{\eta(s)}, \mathcal{V}_{\eta(s)}) - b_j[Z_s, \mu_s^X]) + \frac{1}{2} \sum_{i,j=1}^{d} \partial^2_{x_i,x_j} v_y(s, Z_s)(\bar{a}_{ij}(Z_{\eta(s)}, \mathcal{V}_{\eta(s)}) - a_{ij}[Z_s, \mu_s^X]) \,\Big|\, \mathcal{F}_T^{\mathcal{V}} \Big] ds.$$

Hence,

$$\mathbb{E}[v_y(t, Z_t)|\mathcal{F}_T^{\mathcal{V}}] - \mathbb{E}[v_y(0, Z_0)] = \sum_{k=0}^{n-1} \int_{t_k}^{t_{k+1}} \mathbb{E}\Big[\sum_{i=1}^{4} R_i(s) \Big| \mathcal{F}_T^{\mathcal{V}} \Big] ds,$$

where

$$R_1(s) := \sum_{j=1}^{d} \partial_{x_j} v_y(s, Z_s)(b_j[Z_{\eta(s)}, \mu^X_{\eta(s)}] - b_j[Z_s, \mu_s^X])$$

$$R_2(s) := \sum_{j=1}^{d} \partial_{x_j} v_y(s, Z_s)(\bar{b}_j(Z_{\eta(s)}, \mathcal{V}_{\eta(s)}) - b_j[Z_{\eta(s)}, \mu^X_{\eta(s)}])$$

$$R_3(s) := \frac{1}{2} \sum_{i,j=1}^{d} \partial^2_{x_i,x_j} v_y(s, Z_s)(a_{ij}[Z_{\eta(s)}, \mu^X_{\eta(s)}] - a_{ij}[Z_s, \mu_s^X])$$

$$R_4(s) := \frac{1}{2} \sum_{i,j=1}^{d} \partial^2_{x_i,x_j} v_y(s, Z_s)(\bar{a}_{ij}(Z_{\eta(s)}, \mathcal{V}_{\eta(s)}) - a_{ij}[Z_{\eta(s)}, \mu^X_{\eta(s)}]).$$

Error R_1: Let $\mathcal{F}_T^Z$ be the sigma-algebra generated by $\{Z_t\}_{t\in[0,T]}$. From part (a) of Lemma 3.10 and the Itô's formula, we have

$$\begin{aligned} &\mathbb{E}[R_1(s)|\mathcal{F}_T^{\mathcal{V}}] \\ = &\sum_{k=1}^{d} \mathbb{E}\Big[\partial_{x_k} v_y(s, Z_s) \mathbb{E}\Big[\int_s^{\eta(s)} \Big[\partial_u b_k[Z_u, \mu_u^X] + \sum_{i=1}^{d} \partial_{x_i} b_k[Z_u, \mu_u^X] \bar{b}_i(Z_{\eta(u)}, \mathcal{V}_{\eta(u)}) + \\ &+ \frac{1}{2} \sum_{i,j=1}^{d} \partial^2_{x_i,x_j} b_k[Z_u, \mu_u^X] \bar{a}_{ij}(Z_{\eta(u)}, \mathcal{V}_{\eta(u)}) \Big] du \,\Big|\, \sigma(\mathcal{F}_T^Z, \mathcal{F}_T^{\mathcal{V}}) \Big] \,\Big|\, \mathcal{F}_T^{\mathcal{V}} \Big]. \end{aligned}$$

Condition (**(v-diff-Reg+)**) and the conditional Jensen inequality imply that

$$\mathbb{E}\big|\mathbb{E}[R_1(s)|\mathcal{F}_T^{\mathcal{V}}]\big|$$

$$\leq \quad c\sum_{k=1}^{d}\Bigg(\int_{\eta(s)}^{s}\mathbb{E}\Big|\partial_u b_k[Z_u,\mu_u^X]+\sum_{i=1}^{d}\partial_{x_i}b_k[Z_u,\mu_u^X]\bar{b}_i(Z_{\eta(u)},\mathcal{V}_{\eta(u)})+$$
$$\frac{1}{2}\sum_{i,j=1}^{d}\partial^2_{x_i,x_j}b_k[Z_u,\mu_u^X]\bar{a}_{ij}(Z_{\eta(u)},\mathcal{V}_{\eta(u)})\Big|du\Bigg). \tag{3.33}$$

Using these two bounds along with Lemma 3.8 and assumption **($\mathcal{V}$-Lip)**, we can see that

$$\mathbb{E}\Big|\mathbb{E}[R_1(s)|\mathcal{F}_T^{\mathcal{V}}]\Big|\leq c\Bigg(\int_{\eta(s)}^{s}1+\sup_{s'\in[0,t]}\mathbb{E}|Z_{s'}|^2+\sup_{s'\in[0,t]}\mathbb{E}\Big|\int_{\mathbb{R}^d}|x|^2\mathcal{V}_{s'}(dx)\Big|\,du\Bigg).$$

Assumptions **(Lip)** , **(μ_0-L_p)** and **($\mathcal{V}$-bound)** allow us to conclude that

$$\sup_{0\leq s\leq t}\mathbb{E}|\mathbb{E}[R_1(s)|\mathcal{F}_T^{\mathcal{V}}]|\leq ch_\ell.$$

Error R_2: Condition (**(v-diff-Reg+)**) implies that

$$\Big|\mathbb{E}[R_2(s)|\mathcal{F}_T^{\mathcal{V}}]\Big|\leq c\,\mathbb{E}\big[|b[Z_{\eta(s)},\mu^X_{\eta(s)}]-\bar{b}(Z_{\eta(s)},\mathcal{V}_{\eta(s)})|\,\big|\mathcal{F}_T^{\mathcal{V}}\big]. \tag{3.34}$$

Using the notation of regular conditional probability measures,

$$\mathbb{E}|\mathbb{E}[R_2(s)|\mathcal{F}_T^{\mathcal{V}}]|\leq c\,\mathbb{E}\Bigg[\int_{\mathbb{R}^d}\Big|\mathbb{E}[b(x,X_{\eta(s)})]-\bar{b}(x,\mathcal{V}_{\eta(s)})\Big|\,\mu_{\eta(s)}^{Z|\mathcal{F}_T^{\mathcal{V}}}(dx)\Bigg].$$

Similarly, by the condition on the second-order derivatives from (**(v-diff-Reg+)**), we can establish that

$$\sup_{0\leq s\leq T}\mathbb{E}|\mathbb{E}[R_3(s)|\mathcal{F}_T^{\mathcal{V}}]|\leq ch_\ell \tag{3.35}$$

and

$$|\mathbb{E}[R_4(s)|\mathcal{F}_T^{\mathcal{V}}]|\leq c\,\mathbb{E}\big[\big\|\sigma[Z_{\eta(s)},\mu^X_{\eta(s)}]-\overline{\sigma}(Z_{\eta(s)},\mathcal{V}_{\eta(s)})\big\|\,\big|\mathcal{F}_T^{\mathcal{V}}\big]. \tag{3.36}$$

□

Next, we introduce an artificial process $\bar{Z}^\ell$ in order to remove the dependence of Z^ℓ on $\mathcal{F}_T^{\mathcal{V}}$. Note that $\mu^{Z^\ell|\mathcal{F}_T^{\mathcal{V}}}_{\eta_\ell(s)}$ is a random measure, whereas $\mu^{\bar{Z}^\ell}_{\eta_\ell(s)}$ is non-random. This is crucial in the iteration that will be discussed in the next section.

Lemma 3.12. *Let $P \in C^{0,2}_{b,b}(\mathbb{R}^d \times \mathbb{R}^d, \mathbb{R})$ be a Lipschitz continuous function. Assume that* ***(Ker-Reg)*** *,* ***(μ_0-L_p)*** *,* ***($\mathcal{V}$-bound)*** *and* ***($\mathcal{V}$-Lip)*** *hold. Then there exists a constant c (independent of the choices of L and $N_1, \dots, N_L$) such that for each $t \in [0,T]$, $\ell \in \{0, \dots, L\}$ and $x \in \mathbb{R}^d$,*

$$\begin{aligned}
&\sup_{0\le s\le t} \mathbb{E}\Big[\big|\mathbb{E}[P(x, Z^\ell_s)|\mathcal{F}_T^{\mathcal{V}}] - \mathbb{E}[P(x, X_s)]\big|^2\Big] \\
\le\ & c\bigg(h_\ell^2 + \int_0^t \bigg[\int_{\mathbb{R}^d} \mathbb{E}|\bar{b}(x, \mathcal{V}_{\eta_\ell(s)}) - \mathbb{E}[b(x, X_{\eta_\ell(s)})]|^2 \mu^{\bar{Z}^\ell}_{\eta_\ell(s)}(dx)\bigg] ds \\
&+ \int_0^t \bigg[\int_{\mathbb{R}^d} \mathbb{E}\Big\|\overline{\sigma}(x, \mathcal{V}_{\eta_\ell(s)}) - \mathbb{E}[\sigma(x, X_{\eta_\ell(s)})]\Big\|^2 \mu^{\bar{Z}^\ell}_{\eta_\ell(s)}(dx)\bigg] ds\bigg),
\end{aligned}$$

where $\bar{Z}^\ell$ is a process defined by

$$d\bar{Z}^\ell_t = \int_{\mathbb{R}^d} b(\bar{Z}^\ell_{\eta_\ell(t)}, y)\, \mu^X_{\eta_\ell(t)}(dy)\, dt + \int_{\mathbb{R}^d} \sigma(\bar{Z}^\ell_{\eta_\ell(t)}, y)\, \mu^X_{\eta_\ell(t)}(dy)\, dW_t.$$

Proof. As in the proof of Theorem 3.11, we use $\eta(s)$, Z and $\bar{Z}$ to denote $\eta_\ell(s)$, Z^ℓ and $\bar{Z}^\ell$ respectively. By **(Lip)** and **($\mathcal{V}$-Lip)**,

$$\begin{aligned}
&\mathbb{E}\Big[\Big|\big(b[Z_{\eta(s)}, \mu^X_{\eta(s)}] - \bar{b}(Z_{\eta(s)}, \mathcal{V}_{\eta(s)})\big) - \big(b[\bar{Z}_{\eta(s)}, \mu^X_{\eta(s)}] - \bar{b}(\bar{Z}_{\eta(s)}, \mathcal{V}_{\eta(s)})\big)\Big|^2 \Big|\mathcal{F}_T^{\mathcal{V}}\Big] \\
&\le c\,\mathbb{E}\big[\big|Z_{\eta(s)} - \bar{Z}_{\eta(s)}\big|^2 \big|\mathcal{F}_T^{\mathcal{V}}\big]. \qquad (3.37)
\end{aligned}$$

We further decompose the error as follows.

$$\begin{aligned}
\mathbb{E}\big[\big|Z_{\eta(s)} - \bar{Z}_{\eta(s)}\big|^2 \big|\mathcal{F}_T^{\mathcal{V}}\big] \ &\le\ 2\bigg(\mathbb{E}\bigg[\bigg|\int_0^s \Big(b[\bar{Z}_{\eta(u)}, \mu^X_{\eta(u)}] - \bar{b}(Z_{\eta(u)}, \mathcal{V}_{\eta(u)})\Big)\, du\bigg|^2 \Big|\mathcal{F}_T^{\mathcal{V}}\bigg] \\
&\quad + \mathbb{E}\bigg[\bigg|\int_0^s \Big(\sigma[\bar{Z}_{\eta(u)}, \mu^X_{\eta(u)}] - \overline{\sigma}(Z_{\eta(u)}, \mathcal{V}_{\eta(u)})\Big)\, dW_u\bigg|^2 \Big|\mathcal{F}_T^{\mathcal{V}}\bigg]\bigg) \\
&=:\ 2(R_{21}(s) + R_{22}(s)).
\end{aligned}$$

By the conditional Fubini's theorem and the Cauchy-Schwarz inequality, there

exists a constant $K > 0$ such that

$$\begin{aligned} & R_{21}(s) \\ \leq\ & c\bigg(\int_0^s \mathbb{E}\bigg[\Big| b[\bar{Z}_{\eta(u)}, \mu^X_{\eta(u)}] - \bar{b}\big(\bar{Z}_{\eta(u)}, \mathcal{V}_{\eta(u)}\big) \Big|^2 \Big| \mathcal{F}^{\mathcal{V}}_T \bigg] \\ & + \mathbb{E}\bigg[\Big| \bar{b}\big(\bar{Z}_{\eta(u)}, \mathcal{V}_{\eta(u)}\big) - \bar{b}\big(Z_{\eta(u)}, \mathcal{V}_{\eta(u)}\big) \Big|^2 \Big| \mathcal{F}^{\mathcal{V}}_T \bigg]\, du \bigg) \\ \leq\ & c\bigg(\int_0^s \mathbb{E}\bigg[\Big| b[\bar{Z}_{\eta(u)}, \mu^X_{\eta(u)}] - \bar{b}\big(\bar{Z}_{\eta(u)}, \mathcal{V}_{\eta(u)}\big) \Big|^2 \Big| \mathcal{F}^{\mathcal{V}}_T \bigg] + \mathbb{E}\big[\big| Z_{\eta(u)} - \bar{Z}_{\eta(u)} \big|^2 \big| \mathcal{F}^{\mathcal{V}}_T \big]\, du \bigg), \end{aligned}$$

where assumption **($\mathcal{V}$-Lip)** is used in the final inequality. Since $\bar{Z}$ is independent of $\mathcal{F}^{\mathcal{V}}_T$ and that $\mu^X_{\eta(u)}$ is a non-random measure, we use the properties of regular conditional distributions as outlined in Theorem 7.1 of [56] to prove that for each $\omega \in \Omega$,

$$\begin{aligned} & \bigg(\mathbb{E}\bigg[\Big| b[\bar{Z}_{\eta(u)}, \mu^X_{\eta(u)}] - \bar{b}\big(\bar{Z}_{\eta(u)}, \mathcal{V}_{\eta(u)}\big) \Big|^2 \Big| \mathcal{F}^{\mathcal{V}}_T \bigg] \bigg)(\omega) \\ =\ & \int_{\mathbb{R}^d} \Big| b[x, \mu^X_{\eta(u)}] - \bar{b}(x, \mathcal{V}_{\eta(u)}(\omega)) \Big|^2 \mu^{\bar{Z}}_{\eta(u)}(dx). \end{aligned}$$

Therefore,

$$R_{21}(s) \leq c\bigg(\int_0^s \bigg[\mathbb{E}\big[\big| Z_{\eta(u)} - \bar{Z}_{\eta(u)} \big|^2 \big| \mathcal{F}^{\mathcal{V}}_T \big] + \int_{\mathbb{R}^d} \Big| b[x, \mu^X_{\eta(u)}] - \bar{b}(x, \mathcal{V}_{\eta(u)}) \Big|^2 \mu^{\bar{Z}}_{\eta(u)}(dx) \bigg]\, du \bigg).$$

We proceed similarly as $R_{22}(s)$ and apply part (b) of Lemma 3.10 (with the filtration $\{\mathcal{F}_t\}_{t\in[0,T]}$ such that $\mathcal{F}_t = \sigma(\mathcal{F}^{\mathcal{V}}_T, \{W_u\}_{0\leq u\leq t}, Z_0)$) to get

$$R_{22}(s) \leq c\bigg(\int_0^s \bigg[\mathbb{E}\big[\big| Z_{\eta(u)} - \bar{Z}_{\eta(u)} \big|^2 \big| \mathcal{F}^{\mathcal{V}}_T \big] + \int_{\mathbb{R}^d} \Big\| \sigma[x, \mu^X_{\eta(u)}] - \bar{\sigma}(x, \mathcal{V}_{\eta(u)}) \Big\|^2 \mu^{\bar{Z}}_{\eta(u)}(dx) \bigg]\, du \bigg).$$

Combining both bounds gives

$$\begin{aligned} \mathbb{E}\big[\big| Z_{\eta(s)} - \bar{Z}_{\eta(s)} \big|^2 \big| \mathcal{F}^{\mathcal{V}}_T \big] \leq\ & c\bigg(\int_0^s \bigg[\mathbb{E}\big[\big| Z_{\eta(u)} - \bar{Z}_{\eta(u)} \big|^2 \big| \mathcal{F}^{\mathcal{V}}_T \big] \\ & + \int_{\mathbb{R}^d} \Big| b[x, \mu^X_{\eta(u)}] - \bar{b}(x, \mathcal{V}_{\eta(u)}) \Big|^2 \mu^{\bar{Z}}_{\eta(u)}(dx) \\ & + \int_{\mathbb{R}^d} \Big\| \sigma[x, \mu^X_{\eta(u)}] - \bar{\sigma}(x, \mathcal{V}_{\eta(u)}) \Big\|^2 \mu^{\bar{Z}}_{\eta(u)}(dx) \bigg]\, du \bigg), \end{aligned}$$

for any $s \in [0, t]$. By Gronwall's lemma and integration from 0 to t in time, we

obtain that

$$\int_0^t \mathbb{E}\big[\big|Z_{\eta(s)} - \bar{Z}_{\eta(s)}\big|^2 \big| \mathcal{F}_T^{\mathcal{V}}\big]\, ds \;\leq\; c\bigg(\int_0^t \bigg[\int_{\mathbb{R}^d} \Big| b[x, \mu^X_{\eta(s)}] - \bar{b}(x, \mathcal{V}_{\eta(s)}) \Big|^2 \mu^{\bar{Z}}_{\eta(s)}(dx) + \int_{\mathbb{R}^d} \Big\| \sigma[x, \mu^X_{\eta(s)}] - \overline{\sigma}(x, \mathcal{V}_{\eta(s)}) \Big\|^2 \mu^{\bar{Z}}_{\eta(s)}(dx) \bigg] ds \bigg).$$

By (3.34) and (3.37), it is clear that

$$\int_0^t |\mathbb{E}[R_2(s)|\mathcal{F}_T^V]|^2\, ds \;\leq\; c\bigg(\int_0^t \mathbb{E}\big[\big|Z_{\eta(s)} - \bar{Z}_{\eta(s)}\big|^2 \big| \mathcal{F}_T^{\mathcal{V}}\big] + \mathbb{E}\big[|b[\bar{Z}_{\eta(s)}, \mu^X_{\eta(s)}] - \bar{b}(\bar{Z}_{\eta(s)}, \mathcal{V}_{\eta(s)})|^2 \, \big| \mathcal{F}_T^{\mathcal{V}}\big]\, ds \bigg).$$

This shows that

$$\int_0^t |\mathbb{E}[R_2(s)|\mathcal{F}_T^{\mathcal{V}}]|^2\, ds \;\leq\; c\bigg(\int_0^t \bigg[\int_{\mathbb{R}^d} \Big| b[x, \mu^X_{\eta(s)}] - \bar{b}(x, \mathcal{V}_{\eta(s)}) \Big|^2 \mu^{\bar{Z}}_{\eta(s)}(dx) + \int_{\mathbb{R}^d} \Big\| \sigma[x, \mu^X_{\eta(s)}] - \overline{\sigma}(x, \mathcal{V}_{\eta(s)}) \Big\|^2 \mu^{\bar{Z}}_{\eta(s)}(dx) \bigg] ds \bigg).$$

We repeat the same argument for $R_4(s)$ and conclude that

$$\int_0^t |\mathbb{E}[R_4(s)|\mathcal{F}_T^{\mathcal{V}}]|^2\, ds \;\leq\; c\bigg(\int_0^t \bigg[\int_{\mathbb{R}^d} \Big| b[x, \mu^X_{\eta(s)}] - \bar{b}(x, \mathcal{V}_{\eta(s)}) \Big|^2 \mu^{\bar{Z}}_{\eta(s)}(dx) + \int_{\mathbb{R}^d} \Big\| \sigma[x, \mu^X_{\eta(s)}] - \overline{\sigma}(x, \mathcal{V}_{\eta(s)}) \Big\|^2 \mu^{\bar{Z}}_{\eta(s)}(dx) \bigg] ds \bigg).$$

□

3.4 Iteration of the MLMC algorithm

Fix $m \geq 1$ and correspond each particle $Z^{i,\ell}$ in the abstract framework with $Y^{i,m,\ell}$ defined in (3.10) and $\mathcal{F}_T^{\mathcal{V}}$ with the sigma-algebra $\mathcal{F}^{m-1}$ generated by all the particles $Y^{i,m-1,\ell}$ in the $(m-1)$th Picard step, $0 \leq \ell \leq L, 1 \leq i \leq N_{m-1,\ell}$. We set $\mathcal{V}_t := \mathcal{M}_t^{(m-1)}$ (defined in (3.13)), $\bar{b}(x,\mu) := b[x,\mu]$ and $\overline{\sigma}(x,\mu) := \sigma[x,\mu]$, so that

$$\bar{b}(x, \mathcal{M}_t^{(m-1)}) = \langle \mathcal{M}_t^{(m-1)}, b(x,\cdot)\rangle \quad \text{and} \quad \overline{\sigma}(x, \mathcal{M}_t^{(m-1)}) = \langle \mathcal{M}_t^{(m-1)}, \sigma(x,\cdot)\rangle,$$

for each $x \in \mathbb{R}^d$. The measure $\mathcal{M}^{(m-1)}$ satisfies the independence criterion in **($\mathcal{V}$-bound)** , since $\{Y^{m-1}\} \perp (W^m, Z_0^m)$. The criteria **($\mathcal{V}$-bound)** , **($\mathcal{V}$-Reg)** and **($\mathcal{V}$-Lip)** are verified below.

In the results of this section, c denotes a generic constant that depends on T, but not on m,ℓ or $N_{m,\ell}$.

Lemma 3.13 (Verification of **($\mathcal{V}$-Lip)**)**.** *Assume **(Lip)** and **(μ_0-L_p)** . Then, for each $t \in [0,T]$, there exists a constant c such that for all $x_1, x_2 \in \mathbb{R}^d$*

$$|\langle \mathcal{M}_t^{(m-1)}, b(x_1,\cdot) - b(x_2,\cdot)\rangle| + \|\langle \mathcal{M}_t^{(m-1)}, \sigma(x_1,\cdot) - \sigma(x_2,\cdot)\rangle\| \le c|x_1 - x_2|,$$

$$|\langle \mathcal{M}_t^{(m-1)} b(x_1,\cdot)\rangle| + \|\langle \mathcal{M}_t^{(m-1)} \sigma(x_1,\cdot)\rangle\| \le c\left(1 + |x_1| + \left|\int_{\mathbb{R}^d} |y| \mathcal{M}_t^{(m-1)}(dy)\right|\right).$$

Proof. For any $t \in [0,T]$ and $x_1, x_2 \in \mathbb{R}^d$, by the definition of $\mathcal{M}_t^{(m-1)}$,

$$\begin{aligned}
&\left|\langle \mathcal{M}_t^{(m-1)}, b(x_1,\cdot)\rangle - \langle \mathcal{M}_t^{(m-1)}, b(x_2,\cdot)\rangle\right| \\
= &\left| \sum_{\ell=1}^{L} \frac{1}{N_{m-1,\ell}} \sum_{i=1}^{N_{m-1,\ell}} \left[\left(\frac{t-\eta_\ell(t)}{h_\ell}\right) \cdot \left(b(x_1, Y^{i,m-1,\ell}_{\eta_\ell(t)+h_\ell}) - b(x_2, Y^{i,m-1,\ell}_{\eta_\ell(t)+h_\ell})\right) \right.\right. \\
&+ \left(1 - \frac{t-\eta_\ell(t)}{h_\ell}\right) \cdot \left(b(x_1, Y^{i,m-1,\ell}_{\eta_\ell(t)}) - b(x_2, Y^{i,m-1,\ell}_{\eta_\ell(t)})\right) \\
&- \left(\frac{t-\eta_{\ell-1}(t)}{h_{\ell-1}}\right) \cdot \left(b(x_1, Y^{i,m-1,\ell-1}_{\eta_{\ell-1}(t)+h_{\ell-1}}) - b(x_2, Y^{i,m-1,\ell-1}_{\eta_{\ell-1}(t)+h_{\ell-1}})\right) \\
&\left. - \left(1 - \frac{t-\eta_{\ell-1}(t)}{h_{\ell-1}}\right) \cdot \left(b(x_1, Y^{i,m-1,\ell-1}_{\eta_{\ell-1}(t)}) - b(x_2, Y^{i,m-1,\ell-1}_{\eta_{\ell-1}(t)})\right) \right] \\
&+ \frac{1}{N_{m-1,0}} \sum_{i=1}^{N_{m-1,0}} \left[\left(\frac{t-\eta_0(t)}{h_0}\right) \cdot \left(b(x_1, Y^{i,m-1,0}_{\eta_0(t)+h_0}) - b(x_2, Y^{i,m-1,0}_{\eta_0(t)+h_0})\right) \right. \\
&\left.\left. + \left(1 - \frac{t-\eta_0(t)}{h_0}\right) \cdot \left(b(x_1, Y^{i,m-1,0}_{\eta_0(t)}) - b(x_2, Y^{i,m-1,0}_{\eta_0(t)})\right) \right] \right|.
\end{aligned}$$

The required bounds follow from **(Lip)** . The corresponding estimates for $\|\overline{\sigma}(x_1, \mathcal{V}_{\eta(t)}) - \overline{\sigma}(x_2, \mathcal{V}_{\eta(t)})\|$ and $\|\overline{\sigma}(x_1, \mathcal{V}_{\eta(t)})\|$ can be obtained in a similar way and are hence omitted. □

Lemma 3.14 (Verification of **($\mathcal{V}$-bound)**)**.** *Assume **(Lip)** and (μ_0-L_p) . Then for any $p \geq 2$, there exists a constant c such that*

$$\sup_{n\in\mathbb{N}\cup\{0\}} \sup_{t\in[0,T]} \mathbb{E}\left| \int_{\mathbb{R}^d} |x|^p \mathcal{M}_t^{(n)}(dx) \right| \leq c.$$

Proof. For simplicity of notation, we rewrite

$$\int_{\mathbb{R}^d} |x|^p \mathcal{M}_t^{(n)}(dx) := \frac{1}{N_0} \sum_{i=1}^{N_0} P_t^{i,0} + \sum_{\ell=1}^{L} \frac{1}{N_\ell} \sum_{i=1}^{N_\ell} \left(P_t^{i,\ell} - P_t^{i,\ell-1} \right),$$

where

$$P_t^{i,\ell} = \left(\frac{t - \eta_\ell(t)}{h_\ell} \right) \left| Y^{i,n,\ell}_{\eta_\ell(t)+h_\ell} \right|^p + \left(1 - \frac{t - \eta_\ell(t)}{h_\ell} \right) \left| Y^{i,n,\ell}_{\eta_\ell(t)} \right|^p.$$

We first fix $\ell \in \{1, \ldots, L\}$ and define

$$\Delta_t^{i,\ell} := \mathbb{E}|P_t^{i,\ell} - P_t^{i,\ell-1}|, \quad i \in \{1, \ldots, N_\ell\}.$$

By exchangeability, there exists a constant c (independent of the Picard step n) such that

$$\mathbb{E}[|\Delta_t^{i,\ell}|] \leq c \sum_{\ell'=\ell-1}^{\ell} (\mathbb{E}|Y^{1,n,\ell'}_{\eta_{\ell'}(t)}|^p + \mathbb{E}|Y^{1,n,\ell'}_{\eta_{\ell'}(t)+h_{\ell'}}|^p).$$

By the triangle inequality,

$$\mathbb{E}\left| \frac{1}{N_\ell} \sum_{i=1}^{N_\ell} \Delta_t^{i,\ell} \right| \leq N_\ell^{-1} \sum_{i=1}^{N_\ell} \mathbb{E}|\Delta_t^{i,\ell}| \leq c \sum_{\ell'=\ell-1}^{\ell} \left(\mathbb{E}|Y^{1,n,\ell'}_{\eta_{\ell'}(t)}|^p + \mathbb{E}|Y^{1,n,\ell'}_{\eta_{\ell'}(t)+h_{\ell'}}|^p \right).$$

Similarly, we can show that

$$\mathbb{E}\left| \frac{1}{N_0} \sum_{i=1}^{N_0} P_t^{i,0} \right| \leq c \left(\mathbb{E}|Y^{1,n,0}_{\eta_0(t)}|^p + \mathbb{E}|Y^{1,n,0}_{\eta_0(t)+h_0}|^p \right).$$

Note that

$$\begin{aligned}\mathbb{E}\left|\int_{\mathbb{R}^d}|x|^p\mathcal{M}_t^{(n)}(dx)\right| &\leq \mathbb{E}\left|\frac{1}{N_0}\sum_{i=1}^{N_0}P_t^{i,0}+\sum_{\ell=1}^{L}\frac{1}{N_\ell}\sum_{i=1}^{N_\ell}\Delta_t^{i,\ell}\right| \\ &\leq c\sum_{\ell=0}^{L}\left(\mathbb{E}|Y^{1,n,\ell}_{\eta_\ell(t)}|^p+\mathbb{E}|Y^{1,n,\ell}_{\eta_\ell(t)+h_\ell}|^p\right).\end{aligned}$$

We can see from the proof of Lemma 3.13 that the constant c in Lemma 3.4 does not depend on the particular Picard step. Therefore, by Lemma 3.4,

$$\sup_{0\leq t\leq T}\mathbb{E}\left|\int_{\mathbb{R}^d}|x|^p\mathcal{M}_t^{(n)}(dx)\right|\leq c\left(1+\int_0^T\sup_{0\leq u\leq s}\mathbb{E}\left|\int_{\mathbb{R}^d}|x|^p\mathcal{M}_u^{(n-1)}(dx)\right|ds\right).$$

By iteration, we conclude that

$$\begin{aligned}\sup_{0\leq t\leq T}\mathbb{E}\left|\int_{\mathbb{R}^d}|x|^p\mathcal{M}_t^{(n)}(dx)\right| &\leq \sum_{r=0}^{n-1}\frac{c(cT)^r}{r!}+\sup_{0\leq t\leq T}\mathbb{E}\left|\int_{\mathbb{R}^d}|x|^p\mathcal{M}_t^{(0)}(dx)\right|\frac{(cT)^n}{n!} \\ &\leq ce^{cT}\left(1+\sup_{0\leq t\leq T}\mathbb{E}\left|\int_{\mathbb{R}^d}|x|^p\mathcal{M}_t^{(0)}(dx)\right|\right)<+\infty.\end{aligned}$$

□

Lemma 3.15 (Verification of **($\mathcal{V}$-Reg)**)**.** *Assume **(Lip)** and **(μ_0-L_p)** . Given any Lipschitz continuous function $C^{0,2}_{b,b}\ni P:\mathbb{R}^d\times\mathbb{R}^d\to\mathbb{R}$ and $n\in\mathbb{N}\cup\{0\}$, there exists a constant c such that*

$$\mathbb{E}\left|\langle\mathcal{M}_t^{(n)},P(x,\cdot)\rangle-\langle\mathcal{M}_s^{(n)},P(x,\cdot)\rangle\right|^2\leq c(t-s), \tag{3.38}$$

for any $x\in\mathbb{R}^d$ and $0\leq s\leq t\leq T$.

Proof. When analysing the regularity of MLMC measure (3.38) one needs to pay attention to the interpolation in time that we used. Pick any $\ell^*\in\{0,1,2,\dots L\}$.

For simplicity of notation, we rewrite $\langle \mathcal{M}_t^{(n)}, P(x,\cdot)\rangle$ as

$$\langle \mathcal{M}_t^{(n)}, P(x,\cdot)\rangle := \frac{1}{N_{n,0}}\sum_{i=1}^{N_{n,0}} P_t^{i,0} + \sum_{\ell=1}^{L}\frac{1}{N_{n,\ell}}\sum_{i=1}^{N_{n,\ell}}\left(P_t^{i,\ell} - P_t^{i,\ell-1}\right), \tag{3.39}$$

where

$$P_t^{i,\ell} = \left(\frac{t-\eta_\ell(t)}{h_\ell}\right)P\big(x, Y^{i,n,\ell}_{\eta_\ell(t)+h_\ell}\big) + \left(1-\frac{t-\eta_\ell(t)}{h_\ell}\right)P\big(x, Y^{i,n,\ell}_{\eta_\ell(t)}\big).$$

Given any $k \in \{0,1,\ldots,2^{\ell_*}-1\}$, we compute

$$\begin{aligned}
&\langle \mathcal{M}^{(n)}_{t^{\ell^*}_{k+1}}, P(x,\cdot)\rangle - \langle \mathcal{M}^{(n)}_{t^{\ell^*}_{k}}, P(x,\cdot)\rangle \\
= \;& \frac{1}{N_{n,0}}\sum_{i=1}^{N_{n,0}}(P^{i,0}_{t^{\ell^*}_{k+1}} - P^{i,0}_{t^{\ell^*}_{k}}) + \sum_{\ell=1}^{L}\frac{1}{N_{n,\ell}}\sum_{i=1}^{N_{n,\ell}}\Big((P^{i,\ell}_{t^{\ell^*}_{k+1}} - P^{i,\ell}_{t^{\ell^*}_{k}}) - (P^{i,\ell-1}_{t^{\ell^*}_{k+1}} - P^{i,\ell-1}_{t^{\ell^*}_{k}})\Big).
\end{aligned}$$

Thus, we only need to consider $P^{i,\ell}_{t^{\ell^*}_{k+1}} - P^{i,\ell}_{t^{\ell^*}_{k}}$, for each $\ell \in \{0,1,\ldots,L\}$. There are two cases depending on the value of ℓ: $\ell < \ell^*$ and $\ell \geq \ell^*$.

For levels $\ell < \ell^*$, at least one of $P^{i,\ell}_{t^{\ell^*}_{k+1}}$ and $P^{i,\ell}_{t^{\ell^*}_{k}}$ is an interpolated value. Then there exist a unique $s \in \{0,1,\ldots,2^\ell-1\}$ (chosen such that $\eta_\ell(t^{\ell^*}_k) = t^\ell_s$) and constants $\lambda \in (0, 1-\frac{h_{\ell^*}}{h_\ell}]$ and $\tilde{\lambda}$, given by

$$\lambda = \frac{t^{\ell^*}_k - t^\ell_s}{h_\ell} \text{ and } \tilde{\lambda} = \frac{t^{\ell^*}_{k+1} - t^\ell_s}{h_\ell},$$

such that

$$P^{i,\ell}_{t^{\ell^*}_k} = (1-\lambda)P(x, Y^{i,n,\ell}_{t^\ell_s}) + \lambda P(x, Y^{i,n,\ell}_{t^\ell_{s+1}}) \text{ and } P^{i,\ell}_{t^{\ell^*}_{k+1}} = (1-\tilde{\lambda})P(x, Y^{i,n,\ell}_{t^\ell_s}) + \tilde{\lambda}P(x, Y^{i,n,\ell}_{t^\ell_{s+1}}).$$

Note that $\tilde{\lambda} - \lambda = \dfrac{h_{\ell^*}}{h_\ell}$. By taking the difference between $P^{i,\ell}_{t^{\ell^*}_{k+1}}$ and $P^{i,\ell}_{t^{\ell^*}_{k}}$, we compute that

$$P^{i,\ell}_{t^{\ell^*}_{k+1}} - P^{i,\ell}_{t^{\ell^*}_{k}} = \frac{h_{\ell^*}}{h_\ell}(P(x, Y^{i,n,\ell}_{t^\ell_{s+1}}) - P(x, Y^{i,n,\ell}_{t^\ell_s})). \tag{3.40}$$

For levels $\ell \geq \ell^*$, both of them are not interpolated. This gives

$$P^{i,\ell}_{t^{\ell^*}_{k+1}} - P^{i,\ell}_{t^{\ell^*}_{k}} = P(x, Y^{i,n,\ell}_{t^{\ell^*}_{k+1}}) - P(x, Y^{i,n,\ell}_{t^{\ell^*}_{k}}). \tag{3.41}$$

By Lemmas 3.14 and 3.13, the hypotheses of Lemma 3.5 are satisfied. By applying Lemma 3.5 to (3.40) and (3.41) along with the global Lipschitz property of P, we have

$$\mathbb{E}|P^{i,\ell}_{t^{\ell^*}_{k+1}} - P^{i,\ell}_{t^{\ell^*}_{k}}|^2 \leq c h_{\ell^*} \quad \forall \ell \in \{0, 1, \ldots, L\}.$$

This shows that

$$\begin{aligned}
&\mathbb{E}\left|\langle \mathcal{M}^{(n)}_{t^{\ell^*}_{k+1}}, P(x,\cdot)\rangle - \langle \mathcal{M}^{(n)}_{t^{\ell^*}_{k}}, P(x,\cdot)\rangle\right|^2 \\
\leq\ & \frac{1}{N_{n,0}} \sum_{i=1}^{N_{n,0}} \mathbb{E}|P^{i,0}_{t^{\ell^*}_{k+1}} - P^{i,0}_{t^{\ell^*}_{k}}|^2 + \sum_{\ell=1}^{L} \frac{2}{N_{n,\ell}} \sum_{i=1}^{N_{n,\ell}} \left(\mathbb{E}|P^{i,\ell}_{t^{\ell^*}_{k+1}} - P^{i,\ell}_{t^{\ell^*}_{k}}|^2 + \mathbb{E}|P^{i,\ell-1}_{t^{\ell^*}_{k+1}} - P^{i,\ell-1}_{t^{\ell^*}_{k}}|^2 \right) \\
\leq\ & c h_{\ell^*}.
\end{aligned}$$

The proof is complete by replacing s and t by $\eta_L(s)$ and $\eta_L(t)$ respectively if any of them (or both) does not belong to Π^L. □

Lemma 3.16 below gives a decomposition of MSE (mean-square-error) for MLMC along one iteration of the particle system (3.10).

Lemma 3.16. *Assume* ***(Ker-Reg)*** *and* ***(μ_0-L_p)*** *. Let $P \in C^{0,2}_{b,b}(\mathbb{R}^d \times \mathbb{R}^d, \mathbb{R})$ be a Lipschitz continuous function. Let*

$$MSE^{(m)}_t\big(P(x,\cdot)\big) := \mathbb{E}\left[\left(\mathbb{E}[P(x, X_t)] - \langle \mathcal{M}^{(m)}_t, P(x,\cdot)\rangle\right)^2\right], \quad t \in [0,T].$$

Then, there exists a constant $c > 0$ (independent of the choices of m, L and $(N_{m,\ell})_{0\leq\ell\leq L}$) such that for every $t \in [0,T]$,

$$\int_{\mathbb{R}^d} MSE^{(m)}_{\eta_L(t)}\big(P(x,\cdot)\big)\ \mu^{\bar{Z}^L}_{\eta_L(t)}(dx)$$

$$\leq c\Bigg(h_L^2 + \int_0^t \Bigg[\int_{\mathbb{R}^d} \mathbb{E}\Big|\langle \mathcal{M}^{(m-1)}_{\eta_L(s)}, b(x,\cdot)\rangle - \mathbb{E}[b(x, X_{\eta_L(s)})]\Big|^2 \mu^{\bar{Z}^L}_{\eta_L(s)}(dx)\Bigg] ds$$
$$+ \int_0^t \Bigg[\int_{\mathbb{R}^d} \mathbb{E}\Big\|\langle \mathcal{M}^{(m-1)}_{\eta_L(s)}, \sigma(x,\cdot)\rangle - \mathbb{E}[\sigma(x, X_{\eta_L(s)})]\Big\|^2 \mu^{\bar{Z}^L}_{\eta_L(s)}(dx)\Bigg] ds + \sum_{\ell=0}^{L} \frac{h_\ell}{N_{m,\ell}}\Bigg).$$

Furthermore, if we assume that the functions b and σ are both bounded, then there exists a constant $c > 0$ (independent of the choices of m, L and $(N_{m,\ell})_{0\leq\ell\leq L}$) such that for every $t \in [0,T]$,

$$\sup_{x\in\mathbb{R}^d} MSE^{(m)}_{\eta_L(t)}\big(P(x,\cdot)\big)$$
$$\leq c\Bigg(h_L^2 + \int_0^t \Bigg[\sup_{x\in\mathbb{R}^d} \mathbb{E}\Big|\langle \mathcal{M}^{(m-1)}_{\eta_L(s)}, b(x,\cdot)\rangle - \mathbb{E}[b(x, X_{\eta_L(s)})]\Big|^2\Bigg] ds$$
$$+ \int_0^t \Bigg[\sup_{x\in\mathbb{R}^d} \mathbb{E}\Big\|\langle \mathcal{M}^{(m-1)}_{\eta_L(s)}, \sigma(x,\cdot)\rangle - \mathbb{E}[\sigma(x, X_{\eta_L(s)})]\Big\|^2\Bigg] ds + \sum_{\ell=0}^{L} \frac{h_\ell}{N_{m,\ell}}\Bigg).$$

Proof. For $x \in \mathbb{R}^d$ and $t \in [0,T]$, we consider

$$\mathbb{E}\Big[\Big(\mathbb{E}[P(x, X_{\eta_L(t)})] - \langle \mathcal{M}^{(m)}_{\eta_L(t)}, P(x,\cdot)\rangle\Big)^2\Big]$$
$$= \mathbb{E}\Big[\Big(\mathbb{E}[P(x, X_{\eta_L(t)})] - \mathbb{E}\Big[\langle \mathcal{M}^{(m)}_{\eta_L(t)}, P(x,\cdot)\rangle\Big|\mathcal{F}^{m-1}\Big]$$
$$+\mathbb{E}\Big[\langle \mathcal{M}^{(m)}_{\eta_L(t)}, P(x,\cdot)\rangle\Big|\mathcal{F}^{m-1}\Big] - \langle \mathcal{M}^{(m)}_{\eta_L(t)}, P(x,\cdot)\rangle\Big)^2\Big].$$

Observe that

$$MSE^{(m)}_{\eta_L(t)}\big(P(x,\cdot)\big)$$
$$= \mathbb{E}\Big[\Big(\mathbb{E}[P(x, X_{\eta_L(t)})] - \mathbb{E}[P(x, Y^{1,m,L}_{\eta_L(t)})|\mathcal{F}^{m-1}]\Big)^2\Big]$$
$$+\mathbb{E}\Big[\Big(\mathbb{E}\Big[\langle \mathcal{M}^{(m)}_{\eta_L(t)}, P(x,\cdot)\rangle\Big|\mathcal{F}^{m-1}\Big] - \langle \mathcal{M}^{(m)}_{\eta_L(t)}, P(x,\cdot)\rangle\Big)^2\Big], \tag{3.42}$$

as $\mathbb{E}\Big[\langle \mathcal{M}^{(m)}_{\eta_L(t)}, P(x,\cdot)\rangle\Big|\mathcal{F}^{m-1}\Big] = \mathbb{E}[P(x, Y^{1,m,L}_{\eta_L(t)})|\mathcal{F}^{m-1}]$ by exchangeability. Next, from Lemma 3.12, there exists a constant c such that

$$\mathbb{E}\Big[\Big(\mathbb{E}[P(x, X_{\eta_L(t)})] - \mathbb{E}[P(x, Y^{1,m,L}_{\eta_L(t)})|\mathcal{F}^{m-1}]\Big)^2\Big]$$

$$\leq c\left(h_L^2+\int_0^t\left[\int_{\mathbb{R}^d}\mathbb{E}\left|\langle\mathcal{M}^{(m-1)}_{\eta_L(s)},b(x,\cdot)\rangle-\mathbb{E}[b(x,X_{\eta_L(s)})]\right|^2\mu^{\bar{Z}^L}_{\eta_L(s)}(dx)\right]ds\right.$$
$$\left.+\int_0^t\left[\int_{\mathbb{R}^d}\mathbb{E}\left\|\langle\mathcal{M}^{(m-1)}_{\eta_L(s)},\sigma(x,\cdot)\rangle-\mathbb{E}[\sigma(x,X_{\eta_L(s)})]\right\|^2\mu^{\bar{Z}^L}_{\eta_L(s)}(dx)\right]ds\right). \quad (3.43)$$

By Lemma 3.7, there exists a constant c such that

$$\int_{\mathbb{R}^d}\mathbb{E}\left[\left(\mathbb{E}\left[\langle\mathcal{M}^{(m)}_{\eta_L(t)},P(x,\cdot)\rangle\middle|\mathcal{F}^{m-1}\right]-\langle\mathcal{M}^{(m)}_{\eta_L(t)},P(x,\cdot)\rangle\right)^2\right]\mu^{\bar{Z}^L}_{\eta_L(t)}(dx)$$
$$=\int_{\mathbb{R}^d}\mathbb{E}\left[\mathrm{Var}\left(\langle\mathcal{M}^{(m)}_{\eta_L(t)},P(x,\cdot)\rangle\middle|\mathcal{F}^{m-1}\right)\right]\mu^{\bar{Z}^L}_{\eta_L(t)}(dx)\leq c\sum_{\ell=0}^{L}\frac{h_\ell}{N_{m,\ell}}. \quad (3.44)$$

Combining (3.42), (3.43) and (3.44) yields the result. □

The complete algorithm consists of a sequence of nested MLMC estimators $\left\{\langle\mathcal{M}^{(m)},P(x,\cdot)\rangle\right\}_{m=1,\dots,M}$ and its error analysis is presented in Theorem 3.2. Note that we iterate the algorithm by replacing P by the component real-valued functions $\{b_i\}_{1\leq i\leq d}$ and $\{\sigma_{i,j}\}_{1\leq i\leq d,1\leq j\leq r}$.

3.4.1 Proof of Theorem 3.2

Proof. First, the assumption that $Y^{i,0,\ell}=X_0$ gives

$$\sup_{0\leq t\leq T}\int_{\mathbb{R}^d}\mathbb{E}\left[\left|\mathbb{E}[b(x,X_{\eta_L(t)})]-\langle\mathcal{M}^{(0)}_{\eta_L(t)},b(x,\cdot)\rangle\right|^2\right. \quad (3.45)$$
$$\left.+\left\|\mathbb{E}[\sigma(x,X_{\eta_L(t)})]-\langle\mathcal{M}^{(0)}_{\eta_L(t)},\sigma(x,\cdot)\rangle\right\|^2\right]\mu^{\bar{Z}^L}_{\eta_L(t)}(dx)\leq c.$$

Fixing $M>0$ and $P\in C_b^2(\mathbb{R}^d)$, we set

$$a_t^{(m)}:=\begin{cases}\mathbb{E}\left[\left(\langle\mathcal{M}^{(m)}_{\eta_L(t)},P\rangle-\mathbb{E}[P(X_{\eta_L(t)})]\right)^2\right], & m=M,\\ \int_{\mathbb{R}^d}\mathbb{E}\left[\left|\langle\mathcal{M}^{(m-1)}_{\eta_L(t)},b(x,\cdot)\rangle-\mathbb{E}[b(x,X_{\eta_L(t)})]\right|^2\right.\\ \qquad\left.+\left\|\langle\mathcal{M}^{(m-1)}_{\eta_L(t)},\sigma(x,\cdot)\rangle-\mathbb{E}[\sigma(x,X_{\eta_L(t)})]\right\|^2\right]\mu^{\bar{Z}^L}_{\eta_L(t)}(dx), & m\leq M-1.\end{cases} \quad (3.46)$$

From Lemma 3.16, we observe that

$$a_t^{(m)} \le c\left(b^{(m)} + \int_0^t a_s^{(m-1)} ds\right), \qquad \forall m \in \{1, 2, \dots, M\}, \tag{3.47}$$

where $b^{(m)} = h_L^2 + \sum_{\ell=0}^{L} \frac{h_\ell}{N_{m,\ell}}$. Then one can easily show that

$$\sup_{0\le t\le T} a_t^M \le \sum_{m=0}^{M-1} b^{(M-m)} \frac{(cT)^m}{m!} + \left(\sup_{0\le s\le T} a_s^{(0)}\right) \cdot \frac{(cT)^M}{M!}. \tag{3.48}$$

Inequalities (3.45) and (3.48) conclude the proof. □

We are now in a position to present the complexity theorem for iterated MLMC estimators of $\{\mathbb{E}[P(X_{\eta_L(t)})]\}_{t\in[0,T]}$.

Theorem 3.17. *Assume* ***(Ker-Reg)*** *and* ***(μ_0-L_p)*** *. Fix $M > 0$ and let $P \in C_b^2(\mathbb{R}^d)$. Then there exists some constant $c > 0$ (independent of the choices of M, L and $\{N_{m,\ell}\}_{m,\ell}$) such that for any $\epsilon < e^{-1}$, there exist M, L and $\{N_{m,\ell}\}_{m,\ell}$ such that for every $t \in [0, T]$,*

$$MSE^{(M)}_{\eta_L(t)}(P) := \mathbb{E}\left[\left(\langle \mathcal{M}^{(M)}_{\eta_L(t)}, P\rangle - \mathbb{E}[P(X_{\eta_L(t)})]\right)^2\right] \le c\,\epsilon^2,$$

and computational complexity is of the order $\epsilon^{-4}|\log\epsilon|^3$.

Proof. The cost of obtaining $\langle \mathcal{M}^{(M)}_{\eta_L(t)}, P\rangle$ involves M iterations. In each iteration, one performs the standard MLMC algorithm, where the cost of approximating the law in the drift and diffusion coefficients is $\sum_{\ell'=0}^{L} N_{m-1,\ell'}$. Hence the overall cost $C := C(M, L, \{N_{m,\ell}\}_{m,\ell})$ of the algorithm is

$$C = \sum_{\ell=0}^{L} h_\ell^{-1} N_{1,\ell} + \sum_{m=2}^{M} \sum_{\ell=0}^{L} \left(h_\ell^{-1} N_{m,\ell} \sum_{\ell'=0}^{L} N_{m-1,\ell'}\right). \tag{3.49}$$

For convenience, we use the notation $x \lesssim y$ to denote that there exists a constant c such that $x \le c\, y$. We shall establish specific values $M^*, L^*, \{N^*_{m,\ell}\}_{m,\ell}$ (depending

on ϵ) such that the mean-square error satisfies

$$\sum_{m=1}^{M^*} \frac{c^{M^*-m}}{(M^*-m)!}\big(h_{L^*}^2 + \sum_{\ell=0}^{L^*} \frac{h_\ell}{N_{m,\ell}^*}\big) + \frac{c^{M^*-1}}{M^*!} \lesssim \epsilon^2 \tag{3.50}$$

and show that corresponding computational complexity is of order $\epsilon^{-4}|\log\epsilon|^3$. Firstly, we define

$$M^* := \big\lfloor \log(\epsilon^{-1}) \big\rfloor \implies c^{M^*-1}(M^*!)^{-1} \lesssim \epsilon^2 \tag{3.51}$$

by Stirling's approximation. For $m \in \{1, \dots, M^*\}$, we define $\epsilon_m^2 := w_m \epsilon^2$, for some sequence $\{w_m\}_{m=1}^{M^*}$ (depending on M^* and ϵ) which satisfies the following conditions:

(C1) Minimum condition: For each m, $w_m \geq w_{M^*} = 1$;

(C2) Weight condition: $\sum_{m=1}^{M^*} \frac{c^{M^*-m}}{(M^*-m)!} w_m \leq K$;

(C3) Cost condition: $\sum_{m=1}^{M^*} w_m^{-1} \leq K$,

for some constant $K > 0$. (See Lemma 3.18 for a concrete example.) Subsequently, we define

$$L^* := \max_{1\leq m\leq M^*} L_m^*, \quad L_m^* := \begin{cases} \big|\, \lfloor \log(\epsilon_m^{-1}) \rfloor \,\big|, & \epsilon_m \leq e, \\ 1, & \epsilon_m > e. \end{cases} \tag{3.52}$$

We also define

$$N_{m,\ell}^* := \big\lceil \epsilon_m^{-2}(L^*+1)h_\ell \big\rceil, \quad \ell \in \{0, \dots, L^*\}, \quad m \in \{1, \dots, M^*\}. \tag{3.53}$$

Note that $h_{L^*} \lesssim \epsilon_m$, for any $m \in \{1, \dots, M^*\}$. To see this, we show that $h_{L_m^*} \lesssim \epsilon_m$ by considering the following three cases.

1. Case I: $\epsilon_m > e$. In this case,

$$h_{L_m^*} = T2^{-L_m^*} = T2^{-1} = \left(\frac{T2^{-1}}{e}\right)e < \left(\frac{T2^{-1}}{e}\right)\epsilon_m.$$

2. Case II: $1 \leq \epsilon_m \leq e$. In this case,

$$h_{L_m^*} = T2^{-L_m^*} = T2^{\lfloor \log(\epsilon_m^{-1})\rfloor} = T2^{-\log(\epsilon_m)} \leq T \leq T\epsilon_m.$$

3. Case III: $0 < \epsilon_m < 1$. Without loss of generality, we assume that $T \leq \frac{1}{2}$. (We can scale T by an appropriate factor if it is greater than $\frac{1}{2}$.) In this case,

$$\log(\epsilon_m) \leq \left(\frac{1}{\log 2}\right)\log(\epsilon_m) - \frac{\log(2T)}{\log 2} = \frac{\log(\frac{\epsilon_m}{2T})}{\log 2} = \log_2\left(\frac{\epsilon_m}{2T}\right),$$

which implies that

$$h_{L_m^*} = T2^{-L_m^*} = T2^{-\lfloor \log(\epsilon_m^{-1})\rfloor} \leq T2^{-\left(\log(\epsilon_m^{-1})-1\right)} = 2T2^{\log(\epsilon_m)} \leq \epsilon_m.$$

We can therefore observe that

$$\begin{aligned}
&\sum_{m=1}^{M^*} \frac{c^{M^*-m}}{(M^*-m)!}\left(h_{L^*}^2 + \sum_{\ell=0}^{L^*}\frac{h_\ell}{N_{m,\ell}^*}\right) \\
\leq\ &\sum_{m=1}^{M^*} \frac{c^{M^*-m}}{(M^*-m)!}\left(h_{L^*}^2 + \sum_{\ell=0}^{L^*}\frac{h_\ell}{\epsilon_m^{-2}(L^*+1)h_\ell}\right) \\
\lesssim\ &\sum_{m=1}^{M^*} \frac{c^{M^*-m}}{(M^*-m)!}\epsilon_m^2 \lesssim \epsilon^2,
\end{aligned}$$

by property (C2). Combining this estimate with (3.51), we conclude that the constraint (3.50) is satisfied.

It remains to compute the complexity of the cost under the values

$M^*, L^*, \{N^*_{m,\ell}\}_{m,\ell}$.

$$
\begin{aligned}
C &= \sum_{\ell=0}^{L^*}\left(h_\ell^{-1}\left\lceil \epsilon_1^{-2}(L^*+1)h_\ell\right\rceil\right) + \sum_{m=2}^{M^*}\sum_{\ell=0}^{L^*}\left(h_\ell^{-1}\left\lceil \epsilon_m^{-2}(L^*+1)h_\ell\right\rceil\right. \\
&\quad \left.\sum_{\ell'=0}^{L^*}\left\lceil \epsilon_{m-1}^{-2}(L^*+1)h_{\ell'}\right\rceil\right) \\
&\lesssim \sum_{\ell=0}^{L^*}\left(h_\ell^{-1}\left(\epsilon_1^{-2}(L^*+1)h_\ell+1\right)\right) + \sum_{m=2}^{M^*}\sum_{\ell=0}^{L^*}\left(h_\ell^{-1}\left(\epsilon_m^{-2}(L^*+1)h_\ell+1\right)\right. \\
&\quad \left.\left(\epsilon_{m-1}^{-2}(L^*+1)+(L^*+1)\right)\right) \\
&\lesssim \epsilon^{-2}(L^*+1)^2 + \sum_{m=2}^{M^*}\left(\epsilon_m^{-2}\epsilon_{m-1}^{-2}(L^*+1)^3 + \epsilon_m^{-2}(L^*+1)^3 + \right. \\
&\quad \left.\epsilon^{-1}(L^*+1)^2\epsilon_{m-1}^{-2} + \epsilon^{-1}(L^*+1)^2\right) \\
&\lesssim \epsilon^{-2}|\log(\epsilon^{-1})|^2 + |\log(\epsilon^{-1})|^3\sum_{m=2}^{M^*}\epsilon_m^{-2}\epsilon_{m-1}^{-2} + |\log(\epsilon^{-1})|^3\sum_{m=2}^{M^*}\epsilon_m^{-2} \\
&\quad +\epsilon^{-1}|\log(\epsilon^{-1})|^2\sum_{m=2}^{M^*}\epsilon_{m-1}^{-2} + \epsilon^{-1}|\log(\epsilon^{-1})|^2 M^*, \qquad (3.54)
\end{aligned}
$$

where, we have used in the last two estimates the bounds $L^* \leq \log(\epsilon^{-1})$ (by property (C1)) and $h_\ell^{-1} = T^{-1}2^\ell \leq T^{-1}2^{L^*} \lesssim 2^{\log(\epsilon^{-1})} \lesssim \epsilon^{-1}$. Finally, by properties (C1) and (C3) of $\{w_m\}_{m=1}^{M^*}$, together with (3.54) and (3.51), we conclude that $C \lesssim \epsilon^{-4}|\log(\epsilon)|^3$. □

Lemma 3.18. *The sequence* $\{w_m\}_{m=1}^{M^*}$ *defined by*

$$
w_m := \begin{cases} \max\left\{\frac{(M^*-m-2)!}{c^{M^*-m-2}}, 1\right\}, & 1 \leq m \leq M^*-2, \\ 1, & M^*-1 \leq m \leq M^*, \end{cases}
$$

satisfies properties (C1) *to* (C3) *stipulated in the proof of Theorem 3.17.*

Proof. First, property (C1) follows easily from the definition of w_m. For property

(C2), we verify that

$$\begin{aligned}
&\sum_{m=1}^{M^*} \frac{c^{M^*-m}}{(M^*-m)!} w_m \\
\leq\ &\sum_{m=1}^{M^*-2} \frac{c^{M^*-m}}{(M^*-m)!}\Big(\frac{(M^*-m-2)!}{c^{M^*-m-2}}+1\Big) + \sum_{m=M^*-1}^{M^*} \frac{c^{M^*-m}}{(M^*-m)!} \\
=\ &\sum_{m=1}^{M^*} \frac{c^{M^*-m}}{(M^*-m)!} + c^2 \sum_{m=1}^{M^*-2} \frac{1}{(M^*-m)(M^*-m-1)} \\
=\ &\sum_{m=1}^{M^*} \frac{c^{M^*-m}}{(M^*-m)!} + c^2\left(1-\frac{1}{M^*-1}\right) \leq e^c + c^2.
\end{aligned}$$

Lastly, we show this sequence satisfies property (C3). Indeed,

$$\sum_{m=1}^{M^*} w_m^{-1} = \sum_{m=1}^{M^*-2} \frac{c^{M^*-m-2}}{(M^*-m-2)!} + 2 \leq e^c + 2.$$

□

3.4.2 Plain iterated particle system

The proof of the following theorem constitutes a special case of Lemma 3.16 and Theorem 3.2.

Theorem 3.19. *Assume* ***(Ker-Reg)*** *and* ***(μ_0-L_p)*** *. Fix $M > 0$ and let $P \in C_b^2(\mathbb{R}^d)$. We define the mean-square error as*

$$MSE_t^{(M)}(P) := \mathbb{E}\left[\left(\frac{1}{N_M}\sum_{i=1}^{N_M} P(\overline{Y}_t^{i,M}) - \mathbb{E}[P(X_t)]\right)^2\right].$$

Then for every $t \in [0,T]$,

$$MSE_{\eta(t)}^{(M)}(P) \leq c\left\{h^2 + \sum_{m=1}^{M} \frac{c^{M-m}}{(M-m)!}\cdot\frac{1}{N_m} + \frac{c^{M-1}}{M!}\right\},$$

for some constant $c > 0$ that does not depend on M or $N_1, \dots, N_M$.

The following theorem concerns the computational complexity in the estimation of $\{\mathbb{E}[P(X_{\eta(t)})]\}_{t\in[0,T]}$, whose proof follows similar procedures as the proof of Theorem 3.17 and is omitted.

Theorem 3.20. *Assume* ***(Ker-Reg)*** *and* ***(μ_0-L_p)*** *. Fix $M > 0$ and let $P \in C_b^2(\mathbb{R}^d)$. Then there exists some constant $c > 0$ (independent of the choices of M and $\{N_m\}_{1\le m\le M}$) such that for any $\epsilon < e^{-1}$, there exist M and $\{N_m\}_{0\le m\le M}$ such that for every $t \in [0, T]$,*

$$MSE^{(M)}_{\eta(t)}(P) := \mathbb{E}\left[\left(\frac{1}{N_M}\sum_{i=1}^{N_M} P(\overline{Y}^{i,M}_{\eta(t)}) - \mathbb{E}[P(X_{\eta(t)})]\right)^2\right] \le c\epsilon^2, \tag{3.55}$$

and computational complexity C is of the order ϵ^{-5}.

3.5 Algorithm for the MLMC particle system

For the purpose of implementation, we outline the iterated MLMC particle system in algorithm 1.

3.6 Numerical results

In this section, we present numerical simulations that confirms that iterative MLMC method achieves one order better computational complexity comparing to classical particle system. Furthermore, numerical experiments indicate that the iterative MLMC method works well even if the coefficients of the McKV-SDEs do not satisfy previously stated regularity and growth assumptions. We compare the following methods

Algorithm 1: Nested MLMC with Picard scheme for (3.1)

Input: Initial measure μ^0 for $Y^{i,0,\ell}$, global Lipschitz payoff function $C_p^2 \ni P : \mathbb{R}^d \to \mathbb{R}$ and accuracy level ϵ

Output: $\langle \mathcal{M}_T^{(M)}, P \rangle$, the approximation for our goal $\mathbb{E}[P(X_T)]$.

1 Fix parameters M (see (3.51)) and L (see (3.52)) that correspond to ϵ;

2 Given $\mu^0 = \text{Law}(Y^{i,0,0})$, sample $\{Y^{i,0,0}_{t_k^L}\}_{k=0,\dots,2^L}$;

3 **for** $m = 1$ *to* $M-1$ **do**

4 During mth Picard step, given samples $\{Y^{i,m-1,\ell}_{t_k^\ell}\}^{\ell=0,\dots,L}_{k=0,\dots,2^\ell}$, take (3.10) and run MLMC to obtain $\{Y^{i,m,\ell}_{t_k^\ell}\}^{\ell=0,\dots,L}_{k=0,\dots,2^\ell}$. This requires calculating

$$\left(\langle \mathcal{M}^{(m-1)}_{t_0^L}, b(x,\cdot) \rangle, \dots, \langle \mathcal{M}^{(m-1)}_{t_{2^L}^L}, b(x,\cdot) \rangle \right),$$

$$\left(\langle \mathcal{M}^{(m-1)}_{t_0^L}, \sigma(x,\cdot) \rangle, \dots, \langle \mathcal{M}^{(m-1)}_{t_{2^L}^L}, \sigma(x,\cdot) \rangle \right),$$

where in place of x, we put particles $\{Y^{i,m,\ell}_{t_k^\ell}\}^{\ell=0,\dots,L}_{k=0,\dots,2^\ell-1}$;

5 Given samples $\{Y^{i,M-1,\ell}_{t_k^\ell}\}^{\ell=0,\dots,L}_{k=0,\dots,2^\ell}$, run standard MLMC (with interpolation) to obtain the final vector of approximations $\left(\langle \mathcal{M}^{(M)}_{t_0^L}, P \rangle, \dots, \langle \mathcal{M}^{(M)}_{t_{2^L}^L}, P \rangle \right)$;

6 **Return** $\langle \mathcal{M}_T^{(M)}, P \rangle$.

- Classical particle system (3.4),
- MC Picard I - *iterative particle system* (3.9) with fixed number of particles N for all Picard steps,
- MC Picard II - *iterative particle system* (3.9) with an increasing sequence of particles $\{N_m\}_{m=1,\dots,M}$ where $N_m = w_m N_M$ (see the choice of w_m in Lemma 3.18),
- Iterated MLMC particle system.

Remark 3.21. The computational time in seconds might differ from the theoretical computational cost because it includes the amount of computational cost induced by linear interpolation operations and additional read-write operations relevant to the empirical densities during Picard iterations.

3.6.1 Kuramoto model

First, we provide a numerical example of a one-dimensional stochastic differential equation derived from the Kuramoto model:

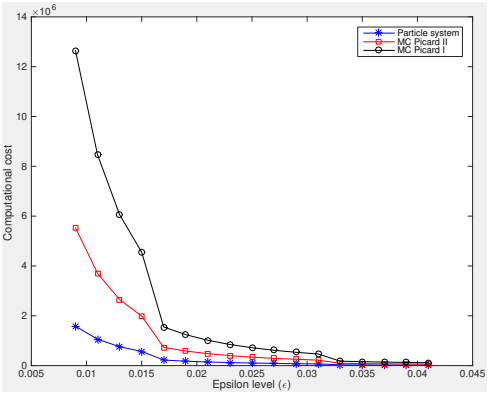
$$\begin{aligned} dX_t &= \int_{\mathbb{R}} \sin(X_t - y)\mu_t^X(dy)\,dt + dW_t, \qquad t \in [0,1], \qquad X_0 = 0, \\ &= \sin(X_t)\int_{\mathbb{R}} \cos(y)\mu_t^X(dy) - \cos(X_t)\int_{\mathbb{R}} \sin(y)\mu_t^X(dy)\,dt + dW_t\,. \end{aligned}$$

For the numerical tests we work with the the bottom representation. We set $P(x) = \sqrt{1+x^2}$. For the initial condition of the iterative algorithm we choose $Y_t^{0,\ell} \sim N(0,t)$.

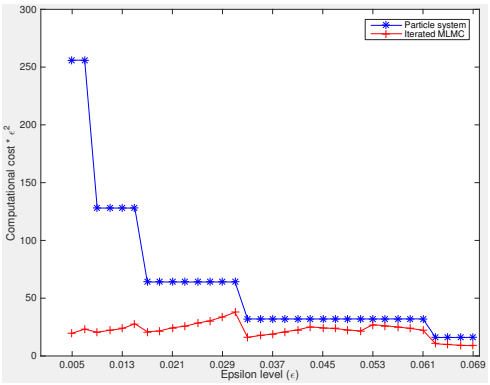
Figure 3.1a shows that both MC Picard I and MC Picard II are less efficient than the classical particle system. In Figure 3.1b, the iterated MLMC particle system achieves computational complexity of order ϵ^{-2} (note that here the cost of simulating particle system is N per Euler step and not N^2 - see Chapter 4).

Figure 3.1c illustrates that the approximation error of iterated methods is within 2ϵ of that of the classical particle system and that it decreases as number of particles increases.

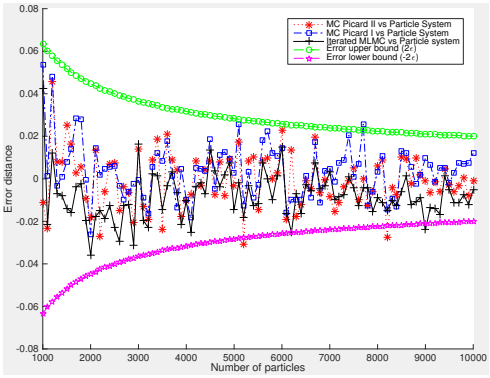
Figure 3.1d depicts $\mathrm{Var}[Y_T^{1,m,\ell}|\mathcal{M}^{(m-1)}]$ and $\mathrm{Var}[Y_T^{1,m,\ell} - Y_T^{1,m,\ell-1}|\mathcal{M}^{(m-1)}]$ (in log scale) for each Picard step across levels ℓ . We see that that the conditional MLMC decays with rate 2. This is higher than the rate given in Lemma 3.6,



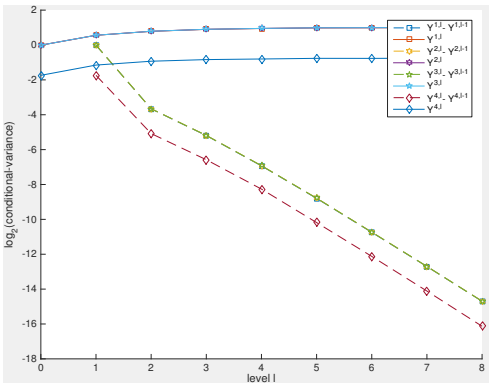
(a) Iterated MC vs Particle method



(b) Iterated MLMC vs Particle method



(c) Approximation error



(d) Variance of $\log_2$ against ℓ for all Picard steps

Figure 3.1: Result of Kuramoto model

since this example treats SDE with constant diffusion coefficient for which Euler scheme achieves higher strong convergence rate.

3.6.2 Polynomial drift

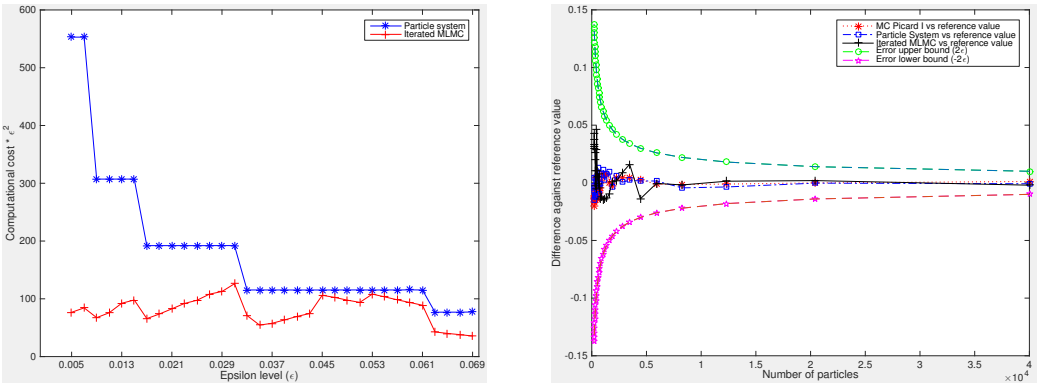
We consider the following McKV-SDE:

$$dX_t = (2X_t + \mathbb{E}[X_t] - X_t\mathbb{E}[X_t^2])dt + X_t dW_t, \quad t \in [0,1], \quad X_0 = 1\,. \tag{3.56}$$

Assumption 3.1 is clearly violated. Note that

$$d\mathbb{E}[X_t] = (3\mathbb{E}[X_t] - \mathbb{E}[X_t]\mathbb{E}[X_t^2])dt \quad \mathbb{E}[X_0] = 1$$
$$d\mathbb{E}[X_t^2] = (5\mathbb{E}[X_t^2] + 2(\mathbb{E}[X_t])^2 - (\mathbb{E}[X_t^2])^2)dt \quad \mathbb{E}[X_0^2] = 1\,.$$

By solving the above system of ODEs with Euler scheme we obtain particle free approximation to the solution of (3.56) that we use as a reference for iterative MLMC method. Figure 3.2a, shows that the iterated MLMC achieves computational complexity of order ϵ^{-2}. Figure 3.2b indicates that the approximation error of iterated methods is within less than 2ϵ of that of the reference value and that it decreases as number of particles increases.



(a) Iterated MLMC vs Particle sytem **(b)** Approximation error

Figure 3.2: Result of Polynomial drift

3.6.3 Viscous Burgers equation

Last, we perform a numerical experiment for the discontinuous case (not Lipschitz) corresponding to the Burgers equation ([13]) given by

$$dX_t = \bar{F}_t(X_t)dt + \frac{1}{4}dW_t, \qquad t \in [0,1], \qquad X_0 = 0, \tag{3.57}$$

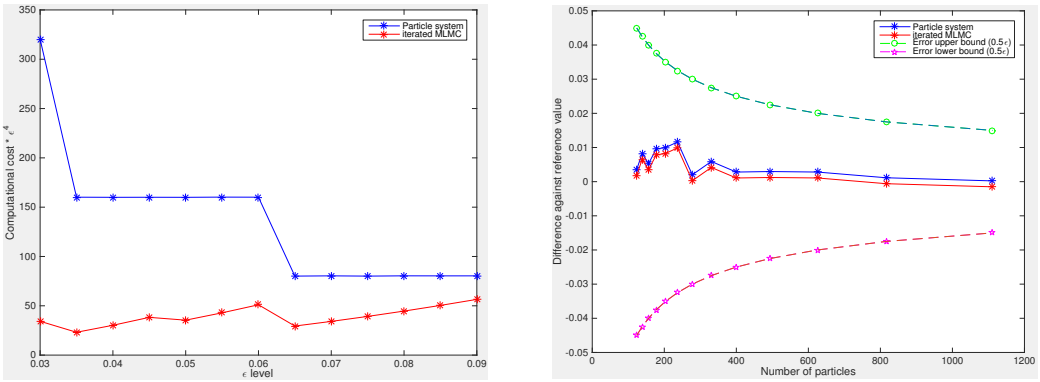
where $\bar{F}_t(x) = \mathbb{P}(X_t \geq x)$. Linking to the Fokker-Planck equation of X_t, it is important to notice that $\bar{F}_t(x)$ is the solution to the viscous Burgers equation:

$$\partial_t v(t,x) = \frac{1}{32}\partial_{xx} v(t,x) - v(t,x)\partial_x v(t,x).$$

where $\bar{F}_0(x) = \mathbf{1}_{\{x \leq 0\}}$ since the initial condition $X_0 = 0$. The Cole-Hopf transformation results in, for any $t \in (0,1]$

$$\bar{F}_t(x) = \frac{\mathcal{N}(\frac{4t-4x}{\sqrt{t}})}{\exp(16x-8t)\mathcal{N}(\frac{4x}{\sqrt{t}}) + \mathcal{N}(\frac{4t-4x}{\sqrt{t}})},$$

where $\mathcal{N}(x) = \int_{-\infty}^{x} \exp(\frac{-y^2}{2})\frac{dy}{\sqrt{2\pi}}$. Then we take $\bar{F}_1(0.5) = 0.5$ as the reference value. In Figure 3.3a, the iterated MLMC achieves computational complexity



(a) Iterated MLMC vs Particle sytem **(b)** Approximation error

Figure 3.3: Result of viscous Burgers equation

of order ϵ^{-4}. Figure 3.3b demonstrates the similar desired behaviour of the approximation error as observed in the case of the polynomial drift.

Chapter 4

Iterative Multilevel Particle Approximation for McKean-Vlasov SDEs with Non-Interacting Kernels

4.1 Introduction

Fix $T > 0$. Let $\{W_t\}_{t\in[0,T]}$ be an r-dimensional Brownian motion on a filtered probability space $(\Omega, \{\mathcal{F}_t\}_t, \mathcal{F}, \mathbb{P})$. Consider continuous functions $b : \mathbb{R}^d \times \mathbb{R}^q \to \mathbb{R}^d$, $\sigma : \mathbb{R}^d \times \mathbb{R}^q \to \mathbb{R}^{d\otimes r}$, $f : \mathbb{R}^d \to \mathbb{R}^q$ and $g : \mathbb{R}^d \to \mathbb{R}^q$. In many applications, only a few expectations rather than the measure affect the interaction in the dynamics (3.1) and it reduces to the corresponding non-linear (in the sense of McKean) stochastic differential equation (McKV-SDEs with non-interacting kernels) given

by

$$dX_t = b\left(X_t, \int_{\mathbb{R}^d} f(y)\,\mu_t^X(dy)\right)dt + \sigma\left(X_t, \int_{\mathbb{R}^d} g(y)\,\mu_t^X(dy)\right)dW_t, \tag{4.1}$$

where $X_0 \sim \mu_0 \in \mathcal{P}_2(\mathbb{R}^d)$. Notice that $\{X_t\}_{t\in[0,T]}$ is still not necessarily a Markov process. Nevertheless applying Itô's formula with $p \in C_{b,\cdot}^{2,1}(\mathbb{R}^d \times [0,T], \mathbb{R})$, the corresponding nonlinear Kolmogorov-Fokker-Planck equation is simplified from (3.2) to

$$\partial_t p + \sum_{i=1}^{d} \partial_{x_i}[p(x,t)b_i(x, M_{1,t}^X)] = \frac{1}{2}\sum_{i,j=1}^{d} \partial^2_{x_i,x_j}[p(x,t)\big(\sigma\sigma^T\big)_{ij}(x, M_{2,t}^X)], \tag{4.2}$$

where M_1 and M_2 are given by respectively

$$M_{1,t}^X := \int_{\mathbb{R}^d} f(y)\,p(y,t)(dy), \quad \textit{and} \quad M_{2,t}^X := \int_{\mathbb{R}^d} g(y)\,p(y,t)(dy).$$

The theory of propagation of chaos, [65] implies the corresponding simplified interacting diffusions $\{Y_t^{i,N}\}_{i=1,\dots,N}$ on $(\mathbb{R}^d)^N$ given by

$$\begin{cases} dY_t^{i,N} &= b(Y_t^{i,N}, M_{1,t}^{Y,N})dt + \sigma(Y_t^{i,N}, M_{2,t}^{Y,N})dW_t^i, \\ M_{1,t}^{Y,N} &:= \frac{1}{N}\sum_{i=1}^N f(Y_t^{i,N}), \quad t \geq 0, \\ M_{2,t}^{Y,N} &:= \frac{1}{N}\sum_{i=1}^N g(Y_t^{i,N}), \quad t \geq 0, \end{cases} \tag{4.3}$$

where $\{Y_0^{i,N}\}_{i=1,\dots,N}$ are i.i.d samples with law μ_0 and $\{W_t^i\}_{i=1,\dots,N}$ are independent Brownian motions. With the same notation we have introduced in Section 3.1, the continuous Euler scheme reads

$$\overline{Y}_t^{i,N} = \overline{Y}_{t_k}^{i,N} + b(\overline{Y}_{\eta(t)}^{i,N}, M_{1,\eta(t)}^{\overline{Y},N})(t-t_k) + \sigma(\overline{Y}_{\eta(t)}^{i,N}, M_{2,\eta(t)}^{\overline{Y},N})(W_t^i - W_{t_k}^i)\,. \tag{4.4}$$

Unlike [31], [59] and [66], we are able to approximate the densities rather than only the expectations. By approximating the density function of (3.1) (we assume it exists) by means of a projection estimator, we obtain reduction of N^2 cost

of the algorithm at every step to $K(\epsilon)N$, where K is the number of terms used in the projection expansion. Next we approximate a projected version of (3.1) further by using (4.4) (let $q = K(\epsilon)$) and a sequence of (linear in a sense of McKean) SDEs obtained via Picard iterations. This allows for MLMC method to be deployed. Projection estimators, are important, as they allow for a computationally efficient mapping of the density approximation between Picard iterations, see [7] for the use of projection type estimators for McKean-Vlasov SDEs. We present fully implementable algorithm that can exploit smoothness of the coefficients in (3.1) and in the best case scenario, that is, if all the coefficient functions are infinitely smooth, the complexity reduces to one in the order of $O(\varepsilon^{-2}|\log(\epsilon)|^3)$ for approximation of expectations and $O(\epsilon^{-2}|\log\epsilon|^4)$ for density estimation.

4.2 Main result of the iterative MLMC algorithm

Again, we combine the MLMC and Picard iteration to improve the computational efficiency. Fix m and L. Take the same family of partitions Π^ℓ, $\ell = 0, \ldots, L$ as we introduce in Section 3.1.3. In this case, we mainly approximate (4.1) by $\{\overline{X}_t^m\}_{t\in[0,T]}$ redefined as

$$d\overline{X}_t^m = b(\overline{X}^m_{\eta_L(t)}, M^{\overline{X}^{m-1}}_{1,\eta_L(t)})dt + \sigma(\overline{X}^m_{\eta_L(t)}, M^{\overline{X}^{m-1}}_{2,\eta_L(t)})dW_t^m\,. \tag{4.5}$$

To simulate (4.5), we only need to approximate the relevant expectations, without dependence on x, with respect to the law of the process at the previous Picard

step $m-1$ and the time grid Π^L, i.e.

$$\left(\mathbb{E}[f(\overline{X}_0^{m-1})], \ldots, \mathbb{E}[f(\overline{X}_{t_k^L}^{m-1})] \ldots, \mathbb{E}[f(\overline{X}_T^{m-1})]\right),$$
$$\left(\mathbb{E}[g(\overline{X}_0^{m-1})], \ldots, \mathbb{E}[g(\overline{X}_{t_k^L}^{m-1})] \ldots, \mathbb{E}[g(\overline{X}_T^{m-1})]\right).$$

By approximating these expectations with the MLMC samples $\mathcal{M}^{(m-1)}$, the *iterative MLMC particle* method reads

$$dY_t^{i,m,\ell} = b(Y_{\eta_\ell(t)}^{i,m,\ell}, \mathcal{M}_{1,\eta_\ell(t)}^{(m-1)})\, dt + \sigma(Y_{\eta_\ell(t)}^{i,m,\ell}, \mathcal{M}_{2,\eta_\ell(t)}^{(m-1)})\, dW_t^{i,m}, \tag{4.6}$$

where $Y^{i,0,\ell} = X_0$ and $\mathcal{M}^{(m-1)}$ is defined as

$$\mathcal{M}_{1,t}^{(m-1)}(f) = \sum_{\ell=0}^{L} \frac{1}{N_{m-1,\ell}} \sum_{i=1}^{N_{m-1,\ell}} f(Y_t^{i,m-1,\ell}) - f(Y_t^{i,m-1,\ell-1}), \tag{4.7}$$

and

$$\mathcal{M}_{2,t}^{(m-1)}(g) = \sum_{\ell=0}^{L} \frac{1}{N_{m-1,\ell}} \sum_{i=1}^{N_{m-1,\ell}} g(Y_t^{i,m-1,\ell}) - g(Y_t^{i,m-1,\ell-1}), \tag{4.8}$$

where $f(Y^{\cdot,-1,\cdot}) = g(Y^{\cdot,-1,\cdot}) = 0$. Under the assumptions listed in Section 3.1.1 with the change of the second argument of functions b and σ from $\mathbb{R}^d$ to $\mathbb{R}^q$, the main result of this chapter gives the same precise error bounds for (4.6) as confirmed in Theorem 3.2 in Chapter 3.

4.3 Abstract framework for MLMC analysis

Compared to Section 3.3, we again introduce an abstract framework corresponding to one iteration with a change for $\mathcal{V} := (\mathcal{V}_1, \mathcal{V}_2)$ to be a $\mathbb{R}^{2q}$-valued random variables from $\mathcal{V} \in \mathcal{P}_2^s(C([0,T], \mathbb{R}^d))$. We consider SDEs with *random* coefficients of the form

$$dU_t = b(U_t, \mathcal{V}_{1,t})dt + \sigma(U_t, \mathcal{V}_{2,t})dW_t, \quad \mu_0^U = \mu_0^X. \tag{4.9}$$

By [41], we also know the solution of this SDE is well-defined under additional assumptions in Section 4.3.1. For $\ell = 1, \dots, L$, the corresponding Euler approximation of (4.9) at level ℓ reads

$$dZ_t^\ell = b(Z^\ell_{\eta_\ell(t)}, \mathcal{V}_{1,\eta_\ell(t)})dt + \sigma(Z^\ell_{\eta_\ell(t)}, \mathcal{V}_{2,\eta_\ell(t)})dW_t, \quad \mu_0^{Z^\ell} = \mu_0^X. \tag{4.10}$$

We still require that $\mathcal{V}$ does not depend on ℓ and that $(W_t)_{t\in[0,T]}$ is independent of $\mathcal{V}$. Subsequently, the particle system $\{Z^{i,\ell}\}$ is defined as follows,

$$dZ_t^{i,\ell} = b(Z^{i,\ell}_{\eta_\ell(t)}, \mathcal{V}_{1,\eta_\ell(t)})dt + \sigma(Z^{i,\ell}_{\eta_\ell(t)}, \mathcal{V}_{2,\eta_\ell(t)})dW_t^i, \qquad \mu_0^{Z^{i,\ell}} = \mu_0^X. \tag{4.11}$$

We remark that this new structure of the abstract framework in terms of $\mathcal{V}$ would allow us to refrain from regular conditional probability measure and thus avoid the additional introduction of the process $\{\overline{Z}_t\}_{t\in[0,T]}$ as needed in the analysis of previous chapter.

4.3.1 Analysis of the abstract framework

With the notation defined in the previous section, the conditions that we need to study the convergence of this iterated particle system is formulated as below. Recall that $\mathcal{V}$ is a square-integrable $\mathbb{R}^{2q}$-valued random variables and we consider equations (4.10) and (4.11). The assumptions are:

Assumption 4.1.

($\mathcal{V}$-bound) *The $\mathbb{R}^{2q}$-valued random variable $\mathcal{V}$, independent of W^i and $Z_0^{i,\ell}$, is integrable for any moment $p \geq 1$.*

($\mathcal{V}$-Reg) *There exists a constant c such that*

$$\sup_{x\in\mathbb{R}^d} \sup_{0\leq s\leq t\leq T} \mathbb{E}\big[\big|b(x,\mathcal{V}_t) - b(x,\mathcal{V}_s)\big|^2 + \big\|\sigma(x,\mathcal{V}_t) - \sigma(x,\mathcal{V}_s)\big\|^2\big] \leq c(t-s).$$

($\mathcal{V}$-Lip) *There exists a constant c such that for each $t \in [0,T]$ and $x, y \in \mathbb{R}^d$,*

$$|b(x,\mathcal{V}_t) - b(y,\mathcal{V}_t)| + \|\sigma(x,\mathcal{V}_t) - \sigma(y,\mathcal{V}_t)\| \leq c|x-y| \tag{4.12}$$

$$|b(x,\mathcal{V}_t)| + \|\sigma(x,\mathcal{V}_t)\| \leq c\Big(1 + |x| + |\mathcal{V}_t|\Big). \tag{4.13}$$

We remark that since all proofs later on follow similar procedures as the proof shown in Section 3.3.1 and Section 3.3.2, we omit those.

Analysis of conditional MLMC variance Using the same notation as we have introduced in Section 3.3.1, we obtain the following integrability of process (4.10) and regularity and strong convergence of process Z_t^ℓ.

Lemma 4.2. *Let Z^ℓ be defined as in (4.10). Assume **($\mathcal{V}$-Lip)** and **(μ_0-L_p)** . Then for any $p \geq 2$ and $\ell \geq 0$, there exists a constant c such that*

$$\mathbb{E}\left[\sup_{t\in[0,T]} |Z_t^\ell|^p\right] \leq c\left(1 + \left[\int_0^T \mathbb{E}|\mathcal{V}_{\eta_\ell(s)}|^p\, ds\right]\right).$$

Lemma 4.3 (Regularity of Z_t^ℓ)**.** *Let Z^ℓ be defined as in (4.10). Assume **($\mathcal{V}$-Lip)** and **($\mathcal{V}$-bound)** . Then, for $p \geq 1$, $0 \leq u \leq s \leq T$,*

$$\Big(\mathbb{E}[|Z_s^\ell - Z_u^\ell|^p]\Big)^{\frac{1}{p}} \leq c(s-u)^{\frac{1}{2}}.$$

Lemma 4.4 (Strong convergence of Z_t^ℓ)**.** *Assume **($\mathcal{V}$-Lip)**, **($\mathcal{V}$-bound)** and **($\mathcal{V}$-Reg)** . Then for any $\ell \in \{1,2,\ldots,L\}$, there exists a constant $c > 0$ such that*

$$\mathbb{E}\left[\sup_{0\leq t\leq T} |Z_t^\ell - Z_t^{\ell-1}|^2\right] \leq ch_\ell.$$

Recalling the definition of σ-algebra $\mathcal{F}_t^{\mathcal{V}} = \{\sigma(\mathcal{V}_s)_{0\le s\le t}\}$, we can finally bound the conditional MLMC variance as follows.

Lemma 4.5. *Assume* ***($\mathcal{V}$-Lip)***, ***($\mathcal{V}$-bound)*** *and* ***($\mathcal{V}$-Reg)*** *hold. Then for any Lipschitz function $P : \mathbb{R}^d \to \mathbb{R}$, there exists a constant c such that*

$$\sup_{0\le t\le T} \mathbb{E}\left[\mathrm{Var}\left(\mathcal{M}_{\eta_L(t)}(P)\middle|\mathcal{F}_T^{\mathcal{V}}\right)\right] \le c\sum_{\ell=0}^{L}\frac{h_\ell}{N_\ell}. \tag{4.14}$$

4.3.2 Weak error analysis

We begin this subsection by defining $\mathcal{X}^{s,x}$ as

$$\mathcal{X}_t^{s,x} = x + \int_s^t b(\mathcal{X}_u^{s,x}, M_{1,u}^X)\,du + \int_s^t \sigma(\mathcal{X}_u^{s,x}, M_{2,u}^X)\,dW_u.$$

For $P \in C_b^2(\mathbb{R}^d, \mathbb{R})$ and $t \in [0,T]$, we consider the function

$$v(s,x) := \mathbb{E}[P(\mathcal{X}_t^{s,x})], \quad (s,x) \in [0,t]\times\mathbb{R}^d. \tag{4.15}$$

We again aim to show that $v(s,x) \in C^{1,2}$. The lemma below gives the first step.

Lemma 4.6. *Assume* ***(μ_0-L_p)*** *and* ***(Ker-Reg)*** *. Then*

$$b(\cdot, M_{1,\cdot}^X) \in C_{b,b}^{2,1}(\mathbb{R}^d\times[0,T],\mathbb{R}^d) \text{ and } \sigma(\cdot, M_{2,\cdot}^X) \in C_{b,b}^{2,1}(\mathbb{R}^d\times[0,T],\mathbb{R}^{d\otimes r}).$$

Next, the following lemma verifies the condition of $v(s,x)$ that is needed:

Lemma 4.7. *Assume* ***(Ker-Reg)*** *and* ***(μ_0-L_p)*** *. Then for any $(s,x) \in [0,t]\times\mathbb{R}^d$, $(i,j) \in \{1,\dots,d\}^2$ and $P \in C_b^2(\mathbb{R}^d,\mathbb{R})$,*

$$\|\partial_{x_i} v(s,x)\|_\infty + \|\partial^2_{x_i,x_j} v(s,x)\|_\infty \le L. \qquad \textbf{((v-diff-Reg+))}$$

Appying the Feynman-Kac theorem ([40]), one can find that $v(\cdot,\cdot)$ satisfies the following Cauchy problem,

$$\begin{cases} \partial_s v(s,x) + \frac{1}{2}\sum_{i,j=1}^{d}\left(\sigma(x,M_{2,s}^X)\sigma(x,M_{2,s}^X)^T\right)_{ij}\partial^2_{x_i,x_j}v(s,x) \\ \qquad\qquad + \sum_{j=1}^{d}\left(b(x,M_{1,s}^X)\right)_j\partial_{x_j}v(s,x) &= 0, \quad (s,x)\in[0,t]\times\mathbb{R}^d, \\ v(t,x) &= P(x). \end{cases} \tag{4.16}$$

We can then reveal the order of weak convergence of (4.10) to (4.1) in the following Theorem 4.8 without addressing the regularity conditional probability measure.

Theorem 4.8. *Let $P\in C_b^2(\mathbb{R}^d,\mathbb{R})$ be a Lipschitz continuous function.* [1] *Assume that* ***(Ker-Reg)*** *,* ***(μ_0-L_p)*** *,* ***($\mathcal{V}$-bound)*** *and* ***($\mathcal{V}$-Lip)*** *hold. Then there exists a constant c (independent of the choices of L and $N_1,\dots,N_L$) such that for each $t\in[0,T]$, $\ell\in\{0,\dots,L\}$ and $x\in\mathbb{R}^d$,*

$$\begin{aligned} &\sup_{0\le s\le t}|\mathbb{E}[P(Z_s^\ell)]-\mathbb{E}[P(X_s)]| \\ &\qquad\le c\left(h_\ell+\int_0^t\mathbb{E}\left|\mathcal{V}_{1,\eta_\ell(s)}-M_{1,\eta_\ell(s)}^X\right|ds+\int_0^t\mathbb{E}\left|\mathcal{V}_{2,\eta_\ell(s)}-M_{2,\eta_\ell(s)}^X\right|ds\right). \end{aligned}$$

and in addition,

$$\begin{aligned} &\sup_{0\le s\le t}|\mathbb{E}[P(x,Z_s^\ell)]-\mathbb{E}[P(x,X_s)]|^2 \\ &\qquad\le c\left(h_\ell^2+\int_0^t\mathbb{E}\left|\mathcal{V}_{1,\eta_\ell(s)}-M_{1,\eta_\ell(s)}^X\right|^2ds+\int_0^t\mathbb{E}\left|\mathcal{V}_{2,\eta_\ell(s)}-M_{2,\eta_\ell(s)}^X\right|^2ds\right). \end{aligned}$$

Proof. Compared to the proof in 3.12, we are now able to use **(Lip)** directly to bound R_2 and R_4. Because $\mathcal{V}_{1,\cdot}$ and $\mathcal{V}_{2,\cdot}$ is measurable with respect to $\mathcal{F}_T^{\mathcal{V}}$, it

[1] *Note that the regularity of P can be relaxed to $C_p^2(\mathbb{R}^d,\mathbb{R})$. We prove the result in a slightly stronger assumption for the sake of simplicity.*

implies

$$\begin{aligned}\mathbb{E}|\mathbb{E}[R_2(s)|\mathcal{F}_T^{\mathcal{V}}]| &\le c\,\mathbb{E}\left[\mathbb{E}\left[|b(Z_{\eta(s)},\mathcal{V}_{1,\eta(s)})-b(Z_{\eta(s)},M^X_{1,\eta(s)})|\middle|\mathcal{F}_T^{\mathcal{V}}\right]\right],\\ &\le c\,\mathbb{E}\left[\mathbb{E}\left[|\mathcal{V}_{1,\eta(s)}-M^X_{1,\eta(s)}|\middle|\mathcal{F}_T^{\mathcal{V}}\right]\right] \le c\,\mathbb{E}|\mathcal{V}_{1,\eta(s)}-M^X_{1,\eta(s)}|,\end{aligned}$$

and

$$\begin{aligned}\mathbb{E}|\mathbb{E}[R_4(s)|\mathcal{F}_T^{\mathcal{V}}]| &\le c\,\mathbb{E}\left[\mathbb{E}\left[\|\sigma(Z_{\eta(s)},\mathcal{V}_{2,\eta(s)})-\sigma(Z_{\eta(s)},M^X_{2,\eta(s)})\|\middle|\mathcal{F}_T^{\mathcal{V}}\right]\right],\\ &\le c\,\mathbb{E}\left[\mathbb{E}\left[|\mathcal{V}_{2,\eta(s)}-M^X_{2,\eta(s)}|\middle|\mathcal{F}_T^{\mathcal{V}}\right]\right] \le c\,\mathbb{E}|\mathcal{V}_{2,\eta(s)}-M^X_{2,\eta(s)}|.\end{aligned}$$

□

4.4 Iteration of the MLMC algorithm

Fix $m \ge 1$ and correspond each particle $Z^{i,\ell}$ in the abstract framework with $Y^{i,m,\ell}$ defined in (4.6) and $\mathcal{F}_T^{\mathcal{V}}$ with the sigma-algebra $\mathcal{F}^{m-1}$ generated by all the particles $Y^{i,m-1,\ell}$ in the $(m-1)$th Picard step, $0 \le \ell \le L, 1 \le i \le N_{m-1,\ell}$. We set $\mathcal{V}_t := \mathcal{M}_t^{(m-1)} = (\mathcal{M}_{1,t}^{(m-1)}, \mathcal{M}_{2,t}^{(m-1)})$ (defined in (4.7) and (4.8)). The $\mathbb{R}^{2q}$-valued random variable $\mathcal{M}^{(m-1)}$ satisfies **($\mathcal{V}$-bound)** , **($\mathcal{V}$-Reg)** and **($\mathcal{V}$-Lip)** which are verified below.

In the results of this section, c denotes a generic constant that depends on T, but not on m,ℓ or $N_{m,\ell}$.

Lemma 4.9 (Verification of **($\mathcal{V}$-Lip)**). *Assume **(Lip)** and **(μ_0-L_p)** . Then, for each $t \in [0,T]$, there exists a constant c such that for all $x_1, x_2 \in \mathbb{R}^d$*

$$|b(x_1,\mathcal{M}_t^{(m-1)})-b(x_2,\mathcal{M}_t^{(m-1)})| + \|\sigma(x_1,\mathcal{M}_t^{(m-1)})-\sigma(x_2,\mathcal{M}_t^{(m-1)})\| \le c|x_1-x_2|,$$

$$|b(x_1, \mathcal{M}_t^{(m-1)})| + \|\sigma(x_1, \mathcal{M}_t^{(m-1)})\| \leq c\left(1 + |x_1| + \left|\mathcal{M}_t^{(m-1)}\right|\right).$$

Lemma 4.10 (Verification of **($\mathcal{V}$-bound)**). *Assume **(Lip)** and (μ_0-L_p) . Then for any $p \geq 2$, there exists a constant c such that*

$$\sup_{n\in\mathbb{N}\cup\{0\}} \sup_{t\in[0,T]} \mathbb{E}\left|\mathcal{M}_t^{(n)}\right|^p \leq c.$$

Lemma 4.11 (Verification of **($\mathcal{V}$-Reg)**). *Assume **(Lip)** and (μ_0-L_p) . Given any Lipschitz continuous function $C_{b,b}^{0,2} \ni P : \mathbb{R}^d \times \mathbb{R}^q \to \mathbb{R}$ and $n \in \mathbb{N}\cup\{0\}$, there exists a constant c such that*

$$\mathbb{E}\left|P(x, \mathcal{M}_t^{(n)}) - P(x, \mathcal{M}_s^{(n)})\right|^2 \leq c(t-s), \tag{4.17}$$

for any $x \in \mathbb{R}^d$ and $0 \leq s \leq t \leq T$.

A decomposition of MSE (mean-square-error) for MLMC along one iteration of the particle system (4.6) is then given by Lemma 4.12 below.

Lemma 4.12. *Assume **(Ker-Reg)** and (μ_0-L_p) . Let $P \in C_b^2(\mathbb{R}^d, \mathbb{R})$ be a Lipschitz continuous function. Let*

$$MSE_t^{(m)}(P) := \mathbb{E}\left[\left(\mathbb{E}[P(X_t)] - \mathcal{M}_t^{(m)}(P)\right)^2\right], \quad t \in [0,T].$$

Then, there exists a constant $c > 0$ (independent of the choices of m, L and $(N_{m,\ell})_{0\leq\ell\leq L}$) such that for every $t \in [0,T]$,

$$\begin{aligned} & MSE_{\eta_L(t)}^{(m)}(P) \\ \leq\ & c\Bigg(h_L^2 + \int_0^t \mathbb{E}\left|\mathcal{M}_{1,\eta_L(s)}^{(m-1)} - \mathbb{E}[f(X_{\eta_L(s)})]\right|^2 ds \\ & + \int_0^t \mathbb{E}\left|\mathcal{M}_{2,\eta_L(s)}^{(m-1)} - \mathbb{E}[g(X_{\eta_L(s)})]\right|^2 ds + \sum_{\ell=0}^{L} \frac{h_\ell}{N_{m,\ell}}\Bigg). \end{aligned}$$

We are now able to present the complexity theorem for iterated MLMC estimators of $\{\mathbb{E}[P(X_{\eta_L(t)})]\}_{t\in[0,T]}$ for the case of non-interacting kernels.

Theorem 4.13. *Assume* ***(Ker-Reg)*** *and* ***(μ_0-L_p)*** *. Fix $M > 0$ and let $P \in C_b^2(\mathbb{R}^d, \mathbb{R})$. Then there exists some constant $c > 0$ (independent of the choices of M, L and $\{N_{m,\ell}\}_{m,\ell}$) such that for any $\epsilon < e^{-1}$, there exist M, L and $\{N_{m,\ell}\}_{m,\ell}$ such that for every $t \in [0,T]$,*

$$MSE^{(M)}_{\eta_L(t)}(P) := \mathbb{E}\left[\left(\mathcal{M}^{(M)}_{\eta_L(t)}(P) - \mathbb{E}[P(X_{\eta_L(t)})]\right)^2\right] \leq c\,\epsilon^2,$$

and computational complexity is of the order $q \cdot \epsilon^{-2}|\log\epsilon|^2$.

Proof. The term $\sum_{\ell=0}^{L} h_\ell^{-1} N_{m,\ell} \sum_{\ell'=0}^{L} N_{m-1,\ell'}$ in (3.49) is replaced by $\sum_{\ell=0}^{L} h_\ell^{-1} N_{m,\ell} + q \cdot \sum_{\ell'=0}^{L} h_{\ell'}^{-1} N_{m-1,\ell'}$. By performing the same computation as in the proof of Theorem 3.17, we can show that the computational complexity is reduced to the order of $q \cdot \epsilon^{-2}|\log\epsilon|^2$. □

4.5 Iterative MLMC algorithm with Projected coefficients

In previous chapter, we address McKV-SDEs (3.1) directly with the iterative MLMC for interacting kernels. However, if there exists a valid reduction method to reduce the interaction of the interacting kernels to a few mean-fields which are non-interacting kernels, we are now in a situation to decrease the complexity from ϵ^{-5} to ϵ^{-2} if ignoring the additional log term. A proposed reduction approach is projection method which is explored by [7].

Projections. Assume that for any $t \geq 0$, the measure $\mu_t(dy)$ of process X_t (3.1) possesses a bounded density $\mu_t(y)$. Let $(\varphi_{\mathbf{k}}, \mathbf{k} \in \mathbb{N}_0^d)$ be a total orthonormal system in $L_2(\mathbb{R}^d, w)$ for some weight function $w(x) > 0$, $x \in \mathbb{R}^d$. Let us formally

write

$$\mu_t(y) = \sum_{\mathbf{k}\in\mathbb{N}_0^d} \gamma_{\mathbf{k}}(t)\varphi_{\mathbf{k}}(y)w(y), \text{ with } \gamma_{\mathbf{k}}(t) = \int_{\mathbb{R}^d} \mu_t(y)\varphi_{\mathbf{k}}(y)\,dy = \mathbb{E}\left[\varphi_{\mathbf{k}}(X_t)\right].$$

Let us introduce the functions

$$\alpha_{\mathbf{k}}(x) := \int b(x,u)\varphi_{\mathbf{k}}(u)\,w(u)\,du, \;\; \beta_{\mathbf{k}}(x) := \int \sigma(x,u)\varphi_{\mathbf{k}}(u)\,w(u)\,du \tag{4.18}$$

so that

$$\int_{\mathbb{R}^d} b(x,y)\mu_t(y)dy = \sum_{\mathbf{k}\in\mathbb{N}_0^d} \alpha_{\mathbf{k}}(x)\gamma_{\mathbf{k}}(t), \;\; \int_{\mathbb{R}^d} \sigma(x,y)\mu_t(y)dy = \sum_{\mathbf{k}\in\mathbb{N}_0^d}^{\infty} \beta_{\mathbf{k}}(x)\gamma_{\mathbf{k}}(t), \tag{4.19}$$

provided $b(x,\cdot), \sigma(x,\cdot) \in L_2(\mathbb{R}^d, w)$ for any $x \in \mathbb{R}^d$, and (3.1) transforms to

$$dX_t = \sum_{\mathbf{k}\in\mathbb{N}_0^d} \alpha_{\mathbf{k}}(x)\gamma_{\mathbf{k}}(t)\,dt + \sum_{\mathbf{k}\in\mathbb{N}_0^d} \beta_{\mathbf{k}}(x)\gamma_{\mathbf{k}}(t)\,dW_t, \tag{4.20}$$

The idea of the projection estimation consists in replacing the infinite sums in (4.20) by finite sums of the first at most $(K+1)^d$ terms, i.e., we consider

$$\begin{aligned} dX_t^K &= \sum_{|\mathbf{k}|\le K} \alpha_{\mathbf{k}}(X_t^K)\gamma_{\mathbf{k}}(t)\,dt + \sum_{|\mathbf{k}|\le K} \beta_{\mathbf{k}}(X_t^K)\gamma_{\mathbf{k}}(t)\,dW_t & (4.21)\\ &= A_K(t,X_t^K)\,dt + B_K(t,X_t^K)\,dW_t & (4.22) \end{aligned}$$

with $|\mathbf{k}| = \max\{k_1,\ldots,k_d\}$ where $\mathbf{k} = (k_1,\ldots,k_d)$. If the coefficients $\gamma_{\mathbf{k}}$ decay fast as $|\mathbf{k}| \to \infty$, one expects that the corresponding approximations $A_{\mathbf{k}}$ and $B_{\mathbf{k}}$ are close to the original coefficients. The associated particle system for (4.21) is

given by

$$\bar{X}_{s+1}^{i,N,K} = \bar{X}_s^{i,N,K} + \sum_{|\mathbf{k}|\leq K} \alpha_{\mathbf{k}}(\bar{X}_s^{i,N,K})\hat{\gamma}_{\mathbf{k}}(t_s)\,(t_{s+1}-t_s) + \sum_{|\mathbf{k}|\leq K} \beta_{\mathbf{k}}(\bar{X}_s^{i,N,K})\hat{\gamma}_{\mathbf{k}}(t_s)\,\Delta W_{s+1}^i, \tag{4.23}$$

where $\hat{\gamma}_{\mathbf{k}}(t) = \frac{1}{N}\sum_{j=1}^N \varphi_{\mathbf{k}}(\bar{X}_t^{j,N,K})$.

Analysis of projections First, we state fairly general conditions that guarantee that the coefficients of projected McKV-SDEs (4.20) are globally Lipschitz and of linear growth and all involved Lipschitz constant decay when $k \to \infty$ (this is crucial for rigorous asymptotic analysis).

(AF) The basis functions $(\varphi_{\mathbf{k}})$ fulfil

$$|\varphi_{\mathbf{k}}(z) - \varphi_{\mathbf{k}}(z')| \leq L_{\mathbf{k},\varphi}\,|z-z'|\,, \quad |\varphi_{\mathbf{k}}(z)| \leq D_{\mathbf{k},\varphi}, \quad \mathbf{k}\in\mathbb{N}_0^d$$

for all $z, z' \in \mathbb{R}^d$ and some constants $L_{\mathbf{k},\varphi}, D_{\mathbf{k},\varphi} > 0$.

(AC) The functions $\alpha_{\mathbf{k}}(x)$, $\beta_{\mathbf{k}}(x)$, $\mathbf{k}\in\mathbb{N}_0^d$ satisfy

$$\sup_{x\in\mathbb{R}^d}|\alpha_{\mathbf{k}}(x)| \leq A_{k,\alpha}(1+|x|) \quad \text{with} \quad \sum_{\mathbf{k}\in\mathbb{N}_0^d} A_{\mathbf{k},\alpha} \leq A_\alpha \text{ and } \sum_{\mathbf{k}\in\mathbb{N}_0^d} L_{\mathbf{k},\varphi}A_{\mathbf{k},\alpha} \leq L_\varphi A_\alpha,$$

$$\sup_{x\in\mathbb{R}^d}|\beta_{\mathbf{k}}(x)| \leq A_{\mathbf{k},\beta}(1+|x|) \quad \text{with} \quad \sum_{\mathbf{k}\in\mathbb{N}_0^d} A_{\mathbf{k},\beta} \leq A_\beta \text{ and } \sum_{\mathbf{k}\in\mathbb{N}_0^d} L_{\mathbf{k},\varphi}A_{\mathbf{k},\beta} \leq L_\varphi A_\beta,$$

and

$$\sup_{x,x'\in\mathbb{R}^d}\frac{|\alpha_{\mathbf{k}}(x)-\alpha_{\mathbf{k}}(x')|}{|x-x'|} \leq B_{\mathbf{k},\alpha} \quad \text{with} \quad \sum_{\mathbf{k}\in\mathbb{N}_0^d} B_{\mathbf{k},\alpha} \leq B_\alpha \text{ and } \sum_{\mathbf{k}\in\mathbb{N}_0^d} D_{\mathbf{k},\varphi}B_{\mathbf{k},\alpha} \leq D_\varphi B_\alpha$$

$$\sup_{x,x'\in\mathbb{R}^d}\sum_{\mathbf{k}\in\mathbb{N}_0^d}^{\infty}\frac{|\beta_{\mathbf{k}}(x)-\beta_{\mathbf{k}}(x')|}{|x-x'|} \leq B_{\mathbf{k},\beta} \quad \text{with} \quad \sum_{\mathbf{k}\in\mathbb{N}_0^d} B_{\mathbf{k},\beta} \leq B_\beta \text{ and } \sum_{\mathbf{k}\in\mathbb{N}_0^d} D_{\mathbf{k},\varphi}B_{\mathbf{k},\beta} \leq D_\varphi B_\beta,$$

for some positive constants $L_\varphi, D_\varphi, A_\alpha, A_\beta, B_\alpha, B_\beta$.

(AM) The density of μ_0 satisfies

$$\mu_0(x) \lesssim \exp\left(-\rho_0|x|^{\rho_1}\right), \quad |x| \to \infty$$

for some $\rho_0 > 0$ and $\rho_1 > 0$.

Example 4.14. *The (normalised) Hermite polynomial of order j is given, for $j \geq 0$, by*

$$\overline{H}_j(x) = c_j(-1)^j e^{x^2} \frac{d^j}{dx^j}(e^{-x^2}), \quad c_j = \left(2^j j!\sqrt{\pi}\right)^{-1/2}.$$

These polynomials are orthonormal with respect to the weight function $w(x) = e^{-x^2}$ and satisfy: $\int_{\mathbb{R}} \overline{H}_j(x)\overline{H}_\ell(x)e^{-x^2}dx = \delta_{j,\ell}$. Set

$$\varphi_{\mathbf{k}}(u) = \overline{H}_{k_1}(u)e^{-u^2/2} \cdot \ldots \cdot \overline{H}_{k_d}(u)e^{-u^2/2} \tag{4.24}$$

where $\mathbf{k} = (k_1, \ldots, k_d) \in \mathbb{N}_0^d$. Then the system $(\varphi_{\mathbf{k}})_{\mathbf{k}\in\mathbb{N}_0^d}$ is a complete orthonormal system in $L_2(\mathbb{R}^d)$ fulfilling the assumption (AF) with $D_{\mathbf{k},\varphi}$ and $L_{\mathbf{k},\varphi}$ uniformly bounded in $\mathbf{k}$. Suppose that $\mu_t \in L_2(\mathbb{R}^d)$ and $b(x,\cdot), \sigma(x,\cdot) \in L_2(\mathbb{R}^d)$. Then the density μ_t can be developed in the Hermite basis $\mu_t(u) = \sum_{\mathbf{j}\in\mathbb{N}_0^d} \gamma_{\mathbf{j}}(t)\varphi_{\mathbf{j}}(u)$ where $\gamma_{\mathbf{j}}(t) = \int_{\mathbb{R}^d} \mu_t(x)\varphi_{\mathbf{j}}(x)dx$. Let us discuss the assumption (AC) by introducing the following notations:

$$T_m(y; f) := \prod_{k=1}^{d} (y_k - \partial_{y_k})^m f(y), \quad y \in \mathbb{R}^d$$

for any smooth function f in $\mathbb{R}^d$.. We also set

$$\tilde{b}(x,u) = \frac{b(x,u)}{\sqrt{1+x^2}}, \quad \tilde{\sigma}(x,u) = \frac{\sigma(x,u)}{\sqrt{1+x^2}}.$$

Theorem 4.15 (Sufficient Condition)**.** *Suppose that for any $x \in \mathbb{R}^d$, it holds that $T_m(\cdot, \tilde{b}(x,\cdot)) \in L_2$ and $T_m(\cdot, \tilde{\sigma}(x,\cdot)) \in L_2$ for some $m \geq 2$. Then the assumption*

(AC) is satisfied and

$$\left\| \sup_{0\le r\le T} \left| X_r^{\cdot,K} - X_r^{\cdot} \right| \right\|_p \le C(d) K^{1-dm/2}, \quad \text{as } K \to \infty, \tag{4.25}$$

where the constant $C(d)$ depends on the dimension d only through the norms $\sup_{x\in\mathbb{R}^d} \|T_m(\cdot, a(x,\cdot))\|_{L_2}$, $\sup_{x\in\mathbb{R}^d} \|T_m(\cdot, b(x,\cdot))\|_{L_2}$ and constants A_α, A_β, L_φ, D_φ, B_α, B_β.

Remark 4.16. In [2] conditions are formulated, guaranteeing that all measures μ_t, $t \ge 0$, possess smooth exponentially decaying densities. In this case we can additionally profit from the decay of the Fourier coefficients $(\gamma_{\mathbf{k}})$. Generally, the smoothness of coefficients b and σ increase as m increases. The smoother the density μ_t is, the faster is the decay rate of its Fourier coefficients $\gamma_{\mathbf{k}}(t)$. Note that the cost of evaluating coefficients is K^d and constant $C(d) \approx d^m$. The convergence then can be expressed in terms of cost as $cost^{1/d-m/2}$. This shows the dependence on the dimension and regularity.

Proof. The proof can be carried out along the same lines as in [7]. □

Iterative MLMC with projections The main idea is to further approximate (4.21) with iterative MLMC. This suggests we need to set both f and g as φ for the iterative MLMC and $\mathcal{M}_{1,t}^{(\cdot)} = \mathcal{M}_{2,t}^{(\cdot)}$. The corresponding Euler scheme adapted from (4.6) reads

$$dY_t^{i,K,m,\ell} = b(Y_{\eta_\ell(t)}^{i,K,m,\ell}, \mathcal{M}_{1,\eta_\ell(t)}^{(m-1)})\, dt + \sigma(Y_{\eta_\ell(t)}^{i,K,m,\ell}, \mathcal{M}_{1,\eta_\ell(t)}^{(m-1)})\, dW_t^{i,m}, \tag{4.26}$$

where $Y^{i,K,0,\ell} = X_0$ and K represents the coefficient is projected up to K-th order. The complete MLMC particle approximation via projection is presented in a schematic form as Algorithm 2.

complexity (Approximation of Expectations) First, we present the complexity theorem for projections w.r.t. the basis (4.24), that is, we consider the particle system (4.23).

Theorem 4.17. *Let $P \in C_b^2(\mathbb{R}^d, \mathbb{R})$ be a Lipschitz continuous function. Assume (AC) and (AF) hold. Then there exists a constant c such that for any $\epsilon < e^{-1}$ and $md > 2$, there exists K, L and N such that*

$$MSE := \sup_{0\le t\le T} \mathbb{E}\left[(\frac{1}{N}\sum_{i=1}^{N} P(\bar{X}^{i,N,K}_{\eta_L(t)}) - \mathbb{E}[P(X_{\eta_L(t)})])^2 \right] < \epsilon^2,$$

and computational complexity C is of order $C(d)^{-\frac{1}{2-md}}\epsilon^{-(3+\frac{2}{md-2})}$.

Proof. From Theorem 4.15 and standard results for particle system, we obtain

$$\sup_{0\le t\le T} \mathbb{E}\left[(\frac{1}{N}\sum_{i=1}^{N} G(\bar{X}^{i,N,K}_{\eta_L(t)}) - \mathbb{E}[G(X_{\eta_L(t)})])^2 \right] \lesssim \left\{ C(d)K^{2-md} + h_L^2 + \frac{1}{N} \right\}.$$

It induces the following optimisation problem

$$\min_{K,L,N} C(K,L,N) = N \cdot K \cdot h_L^{-1} \tag{4.27}$$

$$\text{s.t. } C(d)K^{2-md} + h_L^2 + \frac{1}{N} \lesssim \epsilon^2. \tag{4.28}$$

(4.28) leads to $K \lesssim \frac{\epsilon^{\frac{2}{2-md}}}{C(d)^{\frac{1}{2-md}}}$, $h_L \lesssim \epsilon$ and $N \lesssim \epsilon^{-2}$. Combining these and (4.27) completes the proof. □

Next, we present the complexity theorem for iterated MLMC with projected coefficients approch for the particle system (4.26).

Theorem 4.18. *Assume that **(Ker-Reg)** , (μ_0-L_p) , (AC) and (AF) hold. Let $Y^{\cdot,K,m,\ell}$ be defined in (4.26), $P \in C_b^2(\mathbb{R}^d, \mathbb{R})$ be a Lipschitz continuous function. Then there exists a constant c such that for any $\epsilon < e^{-1}$ there exists M, K, L*

and $\{N_\ell\}_{\ell=0}^L$ *such that*

$$MSE_{M,K} := \sup_{0\le t\le T} \mathbb{E}\left[(\mathcal{M}^{(M,K)}_{\eta_L(t)}(P) - \mathbb{E}[P(X_{\eta_L(t)})])^2\right] < \epsilon^2,$$

where $\mathcal{M}^{(M,K)}_{\eta_L(t)}(P)$ *is given by*

$$\mathcal{M}^{(M,K)}_t(P) = \sum_{\ell=0}^{L} \frac{1}{N_{M,\ell}} \sum_{i=1}^{N_{M,\ell}} P(Y_t^{i,K,M,\ell}) - P(Y_t^{i,K,M,\ell-1}), \quad P(Y_t^{\cdot,\cdot,\cdot,-1}) = 0,$$

the corresponding computational cost being of order $C(d)^{-\frac{1}{2-md}}\epsilon^{-(2+\frac{2}{md-2})}|\log\epsilon|^3$.

Proof. With (AC) and (AF), Theorem 4.15 implies

$$\begin{aligned}\sup_{0\le t\le T} \mathbb{E}\left[(\mathcal{M}^{(M,K)}_{\eta_L(t)}(P) - \mathbb{E}[P(X_{\eta_L(t)})])^2\right] &\lesssim \Big\{|\mathbb{E}[P(X^K_u)] - \mathbb{E}[P(X_{\eta_L(t)})]|^2| \\ &\quad + \mathbb{E}|\mathcal{M}^{(M,K)}_{\eta_L(t)}(P) - \mathbb{E}[P(X^K_u)]|^2\Big\} \\ &\lesssim C(d)K^{2-md} + \mathbb{E}|\mathcal{M}^{(M,K)}_{\eta_L(t)}(P) - \mathbb{E}[P(X^K_u)]|^2.\end{aligned}$$

By **(Ker-Reg)** , **(μ_0-L_p)** , Lemma 4.12 under the setting $q = K+1$ and $f = g := (\phi_0, \dots, \phi_K)$ further gives

$$\begin{aligned}\sup_{0\le t\le T} \mathbb{E}\left[(\mathcal{M}^{(M,K)}_{\eta_L(t)}(P) - \mathbb{E}[P(X_{\eta_L(t)})])^2\right] &\lesssim \Big\{C(d)K^{2-md} + h_L^2 + \sum_{\ell=0}^{L}\frac{h_\ell}{N_\ell} \\ &\quad + \int_0^T \mathbb{E}|[\mathcal{M}^{(M-1,K)}_{\eta_L(t)}(\varphi) - \mathbb{E}[\varphi(X^K_{\eta_L(t)})]|^2 dt\Big\} \\ &\lesssim \cdots \lesssim \Big\{C(d)K^{2-md} + h_L^2 + \sum_{\ell=0}^{L}\frac{h_\ell}{N_\ell} + \frac{1}{M!}\Big\},\end{aligned}$$

where $N_\ell = \max_m\{N_{m,\ell}\}$. It induces the following optimisation problem

$$\min_{M,K,L,\{N_\ell\}_{\ell=0}^L} C(M,K,L,N_0,\dots,N_L) = M\cdot K\cdot\sum_{\ell=0}^{L} h_\ell^{-1}N_\ell \tag{4.29}$$

$$\text{s.t. } C(d)K^{2-md} + h_L^2 + \sum_{\ell=0}^{L} \frac{h_\ell}{N_\ell} + \frac{1}{M!} \lesssim \epsilon^2. \tag{4.30}$$

In a similar way as in Giles's complexity theorem in [30], we see that the only difference is presence of multiplicative constants M and K in (4.29) and additive cost functions $C(d)K^{2-md}$ and $\frac{1}{M!}$ in (4.30). First, we have

$$C(d)K^{2-md} \lesssim \epsilon^2 \implies K \lesssim \frac{\epsilon^{\frac{2}{2-md}}}{C(d)^{\frac{1}{2-md}}}. \tag{4.31}$$

By the Stirling approximation in [60], given by $\sqrt{2\pi}n^{n+\frac{1}{2}}e^{-n} \leq n!, \forall n \geq 1,$, along with (4.31), we obtain

$$\frac{1}{M!} \lesssim \frac{1}{\sqrt{2\pi}(\frac{M}{e})^M} \lesssim \epsilon^2 \implies M \lesssim \log \epsilon^{-1}. \tag{4.32}$$

From [29], we know that $\sum_{\ell=0}^{L} N_\ell h_\ell^{-1} \lesssim \epsilon^{-2}|\log \epsilon|^2$. Combining this, (4.29), (4.31) and (4.32) yields the result.

□

Remark 4.19. Combining remark 4.16 and Theorem 4.18, we observe that if coefficients are infinitely smooth (i.e., $m = \infty$), the complexity is of order $\epsilon^{-2}|\log \epsilon|^3$ for any dimension $d \geq 1$.

complexity (Density estimation) Let us now discuss the estimation of the densities μ_t, $t \geq 0$. Fix some $t > 0$, $K \in \mathbb{N}^d$ and set

$$\widehat{\mu}_t^K(x) := \sum_{|\mathbf{k}| \leq K} \gamma_{\mathbf{k}}^A(t)\varphi_{\mathbf{k}}(x)w(x),$$

where $\gamma_{\mathbf{k}}^A(t)$ is a type of approximation for $\gamma_{\mathbf{k}}$ such as $\hat{\gamma}_{\mathbf{k}}$ in (4.23). We again work with mean-square-error in the form of

$$\begin{aligned} MSE &:= \sup_{0\le t\le T} \mathbb{E} \int |\widehat{\mu}_t^K(x) - \mu_t(x)|^2 w^{-1}(x)\, dx \\ &\lesssim \sum_{|\mathbf{k}|\le K} \sup_{0\le t\le T} \mathbb{E}\left[|\gamma_{\mathbf{k}}^A(t) - \gamma_{\mathbf{k}}(t)|^2\right] + \sum_{|\mathbf{k}|>K} \sup_{0\le t\le T} |\gamma_{\mathbf{k}}(t)|^2, \end{aligned} \tag{4.33}$$

If we assume that the coefficients b and σ are infinitely smooth w.r.t the basis (4.24) in the sense that

$$A_{\mathbf{k},\alpha} \lesssim \exp(-\alpha^\circ |\mathbf{k}|), \quad A_{\mathbf{k},\beta} \lesssim \exp(-\beta^\circ |\mathbf{k}|), \quad |\mathbf{k}| \to \infty$$

for some constants $\alpha^\circ, \beta^\circ > 0$, then it can be shown along the same lines as in the proof of Theorem 2.2 in [2] that

$$|\gamma_{\mathbf{k}}(t)| \lesssim \exp(-\gamma^\circ |\mathbf{k}|), \quad |\mathbf{k}| \to \infty \tag{4.34}$$

for some $\gamma^\circ > 0$, provided that the diffusion coefficient satisfies the Hörmander condition. As shown in Example 4.14, $D_{\mathbf{k},\varphi}$ and $L_{\mathbf{k},\varphi}$ are uniformly bounded in $\mathbf{k}$. To ensure $MSE < \epsilon_0^2$, combing (4.33) and (4.34), together with the property of geometric series implies $K \lesssim |\log(\epsilon_0)|$. Therefore, for any $|\mathbf{k}| < K$, we need

$$\sup_{0\le t\le T} \mathbb{E}\left[|\gamma_{\mathbf{k}}^A(t) - \gamma_{\mathbf{k}}(t)|^2\right] = \sup_{0\le t\le T} \mathbb{E}\left[|\gamma_{\mathbf{k}}^A(t) - \mathbb{E}[\varphi(X_t)]|^2\right] \lesssim \frac{\epsilon_0^2}{K} \lesssim \frac{\epsilon_0^2}{|\log(\epsilon_0)|}$$

It means if we set the accuracy level ϵ^2 as $\frac{\epsilon_0^2}{|\log(\epsilon_0)|}$ in previous complexity theorems for approximation of expectations, we find the optimal parameter setting corresponding to $\gamma_{\mathbf{k}}^A(t)$ for both methods. By computation, the complexities are in the magnitude of

$$\mathcal{O}\left(C(d)^{-\frac{1}{2-md}} \left(\frac{\epsilon_0}{\sqrt{|\log(\epsilon_0)|}}\right)^{-(3+\frac{2}{md-2})}\right)$$

and

$$\mathcal{O}\left(C(d)^{-\frac{1}{2-md}}\left(\frac{\epsilon_0}{\sqrt{|\log(\epsilon_0)|}}\right)^{-(2+\frac{2}{md-2})}|\log\epsilon_0|^3\right),$$

respectively. To conclude, the density estimation results in the additional log term compared to the corresponding approximation of expectations.

4.6 Numerical Experiments

Projection particle method Consider the MVSDE of the form:

$$dX_t = \mathbb{E}_{X'}\left[Q(X_t - X'_t)\right]\,dt + \sigma\,dW_t, \quad t \in [0,1], \quad X_0 = 0.5, \tag{4.35}$$

i.e. of the form (3.1) with $b(x,y) = Q(x-y)$, $\sigma(x,y) = \sigma$. Let us use the Hermite basis (see section 4.5) to approximate the density of X_t for any $t \in [0,1]$. In the case $Q(x) = e^{-x^2/2}$, we explicitly derive

$$\begin{aligned}\int_{\mathbb{R}} e^{-(x-y)^2/2 - x^2/2} H_n(x)\,dx &= \frac{e^{-y^2/4}}{2}\int e^{-(z-y)^2/4} H_n(z/2)\,dz \\ &= \sqrt{\pi}\,\frac{e^{-y^2/4}}{2}\left(\frac{1}{2}\right)^{n-1}(2y)^n.\end{aligned}$$

As a result

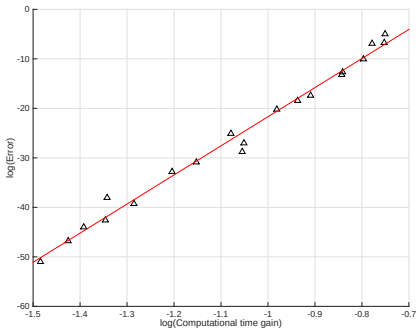
$$\alpha_n(y) = \int e^{-(x-y)^2/2 - x^2/2}\overline{H}_n(x)\,dx = \pi^{1/4}\left(\frac{1}{2}\right)^{n/2}\frac{y^n}{\sqrt{n!}}e^{-y^2/4},$$

where $\overline{H}_n$ stands for the normalised Hermite polynomial of order n. We take $\sigma = 0.1$. Using the Euler scheme (3.8) with time step $h = 1/L = 0.01$, we first simulate $N = 500$ paths of the time discretised process $\bar{X}^{\cdot,N}$. Next, by means of the closed form expressions for α_n, we generate N paths of the projected approximating process $\bar{X}^{\cdot,K,N}$, $K \in \{1,\dots,20\}$ using the same Wiener increments as for $\bar{X}^{\cdot,N}$, so that the approximations $\bar{X}^{\cdot,N}$ and $\bar{X}^{\cdot,K,N}$ are coupled. Finally, we

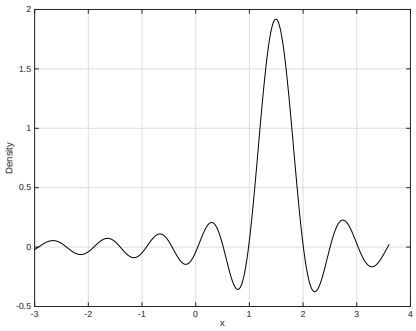
compute the strong approximation error

$$E_{N,K} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\bar{X}_1^{i,K,N} - \bar{X}_1^{i,N}\right)^2}$$

and record times needed to compute approximations $\bar{X}_1^{\cdot,N}$ and $\bar{X}_1^{\cdot,K,N}$, respectively. Figure 4.1 shows the logarithm of $E_{N,K}$ versus the logarithm of the corresponding computational time differences for values $K \in \{1, \dots, 20\}$. As can be seen, the relation between logarithmic strong error and logarithmic computational time gain can be well approximated by a linear function. On the left-hand side of Figure 4.1 we depict the projection estimate for the density of X_1 corresponding to $K = 10$.



(a) Strong Error $E_{500,K}$ between PPM and Chaos



(b) Estimated density of X_1 under $K = 10$

Figure 4.1: Left: Strong error $E_{500,K}$ between the solution of projected (see (4.21)) and non-projected (see (3.4)) time-discretised particle systems versus the difference (gain) in computational time. Right: Estimated density of X_1 using 11 basis functions.

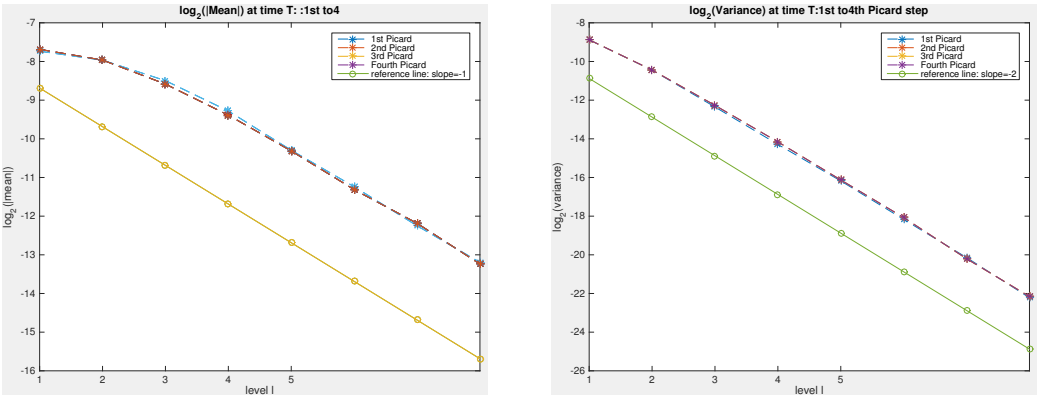
Remark 4.20. Note that in Figure 4.1 (also Figure 4.4) the approximated density takes negative values but for example it can be resolved by a certain shifting transformation (see section 2.7 in [6]).

Iterated MLMC on Kth order Hermite Projected SDE Likely, consider one-dimensional Kth order Hermite Projected SDE (4.36):

$$dX_t = \sum_{k=0}^{K} \mathbb{E}[\varphi_k(X_t)]\alpha_k(X_t)dt + \sigma dW_t, \quad t \in [0,1], \quad X_0 = 0.5, \tag{4.36}$$

where $T = 1, \sigma = 0.1$, $\varphi_k(x) = \bar{H}_k(x)e^{-\frac{x^2}{2}}$ and $\alpha_k(x) = \pi^{1/4}\left(\frac{1}{2}\right)^{k/2}\frac{x^k}{\sqrt{n!}}e^{-x^2/4}$.

Tests of convergence rates As to initialisation, let $X^0 \sim N(0.5, 1)$, $X_0 = 0.5$ and $L = 5$ (recall that $h_\ell = T \cdot 2^{-\ell}$). In this case when $X^0 \sim N(0.5, 1)$ (does no change with time) first Picard X^1 is just an SDE with drift at each time-step being exactly the same. We perform tests across all Picard steps and obtain convergence rates are stable during all Picard iterations as depicted on Figure 4.2. Results are shown in Figure 4.2.



(a) weak rate **(b)** strong rate

Figure 4.2: In those tests, N_m is 100000 for all Picard steps and the terminal $T = 1$. The sequence (a_ℓ) corresponding to weak error of φ_0 turns into $a_\ell := \left|\frac{1}{N}\sum_{i=1}^{N}\varphi_0(Y_t^{i,m,\ell}) - \varphi_0(Y_t^{i,m,\ell-1})\right|$. Similarly, the sequence (b_ℓ) corresponding to strong error of φ_0 becomes $b_\ell := \frac{1}{N}\sum_{i=1}^{N}\left|\varphi_0(Y_t^{i,m,\ell}) - \varphi_0(Y_t^{i,m,\ell-1})\right|^2$. The reference lines in Figure 4.2a and Figure 4.2b indicates $\alpha = 1$ and $\beta = 2$ respectively.

Comparison of all three methods. In this section, we compare complexities among standard Particle system applied to the original MVSDE (4.35), Projected Particle system applied to SDE (4.36) and iterated MLMC (MLMC with Picard) with projected coefficients applied to SDE (4.36). The measurement of computational cost consists of the number of random generations, evaluations of the drift coefficient while ignoring a small amount of constant computational cost induced by linear interpolation operations.

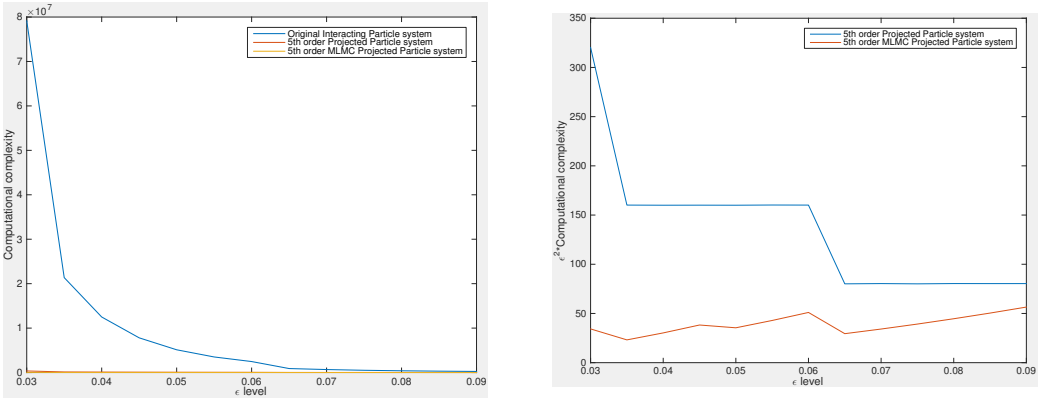
Remark 4.21. We consider the particle system with $5 \cdot 10^5$ particles and 2^{10} timesteps as proxy for the explicit solution of equation (4.35) and it corresponds to the benchmark value 1.4951. The MSEs then have been verified, for example, if we require MSEs to be less than $\epsilon = 0.03$, the values lie in the desired region $(1.4951 \pm 2\epsilon)$.

Complexity. Figure 4.3 presents the comparison of those complexities. Figure 4.3a shows standard Particle system applied to the original MVSDE (4.35) is the worst one. Figure 4.3b indicates that iterated MLMC with projected coefficients can be applied to SDE (4.36) to further improve the Projected Particle method after reducing interactions from N to K.

Estimate density of X_1. Combining Figure 4.3 and Figure 4.4 concludes that iterated MLMC with projected coefficients reduce the computational cost without losing the accuracy in terms of density estimation.

4.7 Conclusions

In this section we presented a novel iterative MLMC estimator with the certain reduction method (projection by using Hermite polynomials) for the challenging problem of simulating McKean-Vlasov SDEs (3.1). As in the classical stochastic

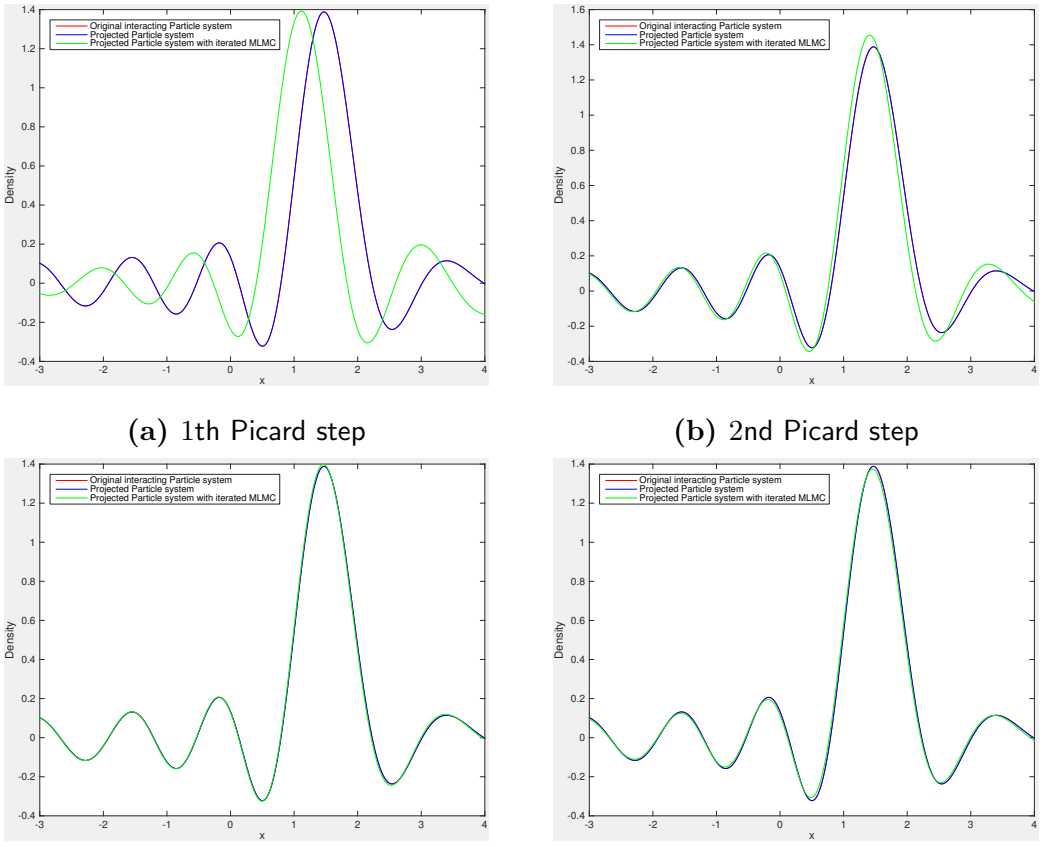


(a) Chaos vs PPM vs Iterated MLMC with Projected coefficients

(b) PPM vs Iterated MLMC with Projected coefficients

Figure 4.3: Chaos stands for standard particle system method applied to the original interacting MVSDE; PPM means Projected Particle method; From Figure 4.3a , we see standard particle system has much higher order than ϵ^{-2} and also observe from Figure 4.3b that MLMC with projected coefficients is close to $O(\epsilon^{-2})$ while projections close to $O(\epsilon^{-3})$ which coincides with the Theorems 4.17 and 4.18 for complexity.

particle, approximation particles are not independent and the bias and statistical error are in a nonlinear relationship, classical methods fails. Nonetheless in the case when coefficients of MVSDEs are smooth enough our approach recovers computational complexity $\varepsilon^{-2}|\log(\epsilon)|^3$. This is a very promising strand of research and we have addressed the complete rigorous analysis of this fact. Furthermore, our approach easily extends to other reduction approximation methods and also allows for MLMC treatment of approximating densities. Finally, we believe that the idea of approximating a complex/non-linear/high-dimensional models with manageable/simplified models and then applying iterative MLMC approach can be fruitful in other application areas and we anticipate interest in this approach from various research communities.



(a) 1th Picard step

(b) 2nd Picard step

(c) 3th Picard step

(d) 4th Picard step

Figure 4.4: Estimated density of X_1 using 10 basis functions both for Projected particle system and iterated MLMC with projected coefficients during all Picard steps.

Algorithm 2: Nested MLMC with Picard scheme for (4.1)

Input: Initial measure μ^0 for X^0, MSE level ϵ, payoff function $P(\cdot) : \mathbb{R}^d \to \mathbb{R}$ and M, the number of Picard steps.

Output: $\mathcal{M}_T^M(P)$, the approximation for the quantity $\mathbb{E}[P(X_T)]$.

Find $K(\epsilon)$ and set $q = K(\epsilon)$;

Fix L_0. Apply standard MLMC to the process X^0 to calculate the approximation matrix of size $q \times (1 + 2^{L_0})$

$$\bar{D}^0 := \left(\mathcal{M}_{1,t_0}^{(0)} \quad , \ldots , \quad \mathcal{M}_{1,t_{2^{L_0}}}^{(0)}\right).$$

for $m = 1$ *to* $M - 1$ **do**

  Fix L_m, the largest level in Multilevel in mth Picard step;

  Conditioning on $\bar{D}^{m-1}$ (starting from $\bar{D}^0$), take (4.26) and run MLMC to obtain the matrix of of size $q \times (1 + 2^{L_0})$:

$$\bar{D}^m := \left(\mathcal{M}_{1,t_0}^{(m)} \quad , \ldots , \quad \mathcal{M}_{1,t_{2^{L_0}}}^{(m)}\right).$$

Fix L_M, the largest level in MLMC at Mth Picard step;

Conditioning on $\bar{D}^{M-1}$, run standard MLMC with interpolation to obtain the final vectors of approximations $(\mathcal{M}_{t_0}^{(M,K)}(P), \ldots, \mathcal{M}_{t_{2^{L_M}}}^{(M,K)}(P))$;

Return $\mathcal{M}_{t_{2^{L_M}}}^{(M,K)}(P)$, i.e. $\mathcal{M}_T^{(M,K)}(P)$.

Chapter 5

Monte-Carlo based deep learning methods for several types of random PDEs

5.1 Introduction

It is common to use partial differential equations (PDEs) to describe numerous physical phenomena such as fluid dynamics, quantum mechanics, and elasticity. In order to reliably analyze these phenomena, we need to consider the additional inherent uncertainties in the system of interest (e.g. in their initial or boundary conditions, material properties, external forces) and quantify the impact of these uncertainties on quantities of interest (QoIs). This often involves computing the expectation (and/or higher order moments) of a functional of the solution of a random PDEs system. Realistic models of the uncertainties often require a large number of random parameters, and thus, even if the spatial dimension is typically only two or three, we have to solve PDEs with a high-dimensional input space.

Developing numerical methods for solving these has been a persistent challenge. The aim of this work is to show how deep neural networks can be used to efficiently solve problems in uncertainty quantification in PDE based modelling.

We start with a brief overview of the most common approaches to solving high-dimensional random PDEs. Monte Carlo Sampling (MCS) [26] is one of the simplest approaches for this task. In MCS, we generate independent samples of the random parameters, determined by their given probability distribution. For each sample of the random parameters, we then solve the resulting (deterministis) PDE by a numerical method, such as Finite Element (FE), Finite Difference (FD), and Finite Volume (FV). We then approximate the QoI by the Monte-Carlo average of these deterministic solutions. Although MCS has the desirable property of a dimension-independent convergence rate, the convergence can be quite slow, and recent research has often focussed on improved variants such as Quasi-Monte Carlo sampling [42] and multilevel Monte Carlo sampling [5, 20].

Alternatively, some researchers expand the PDE solution (a random variable determined by the random parameters) as a Taylor series around the means [47], in so-called Perturbation methods. These methods are limited by the magnitude of uncertainty, and are only applicable if the uncertainties are small. Also, it is possible to manipulate the stochastic operators in PDEs, and the classic approaches are the weighted integral method [23] and the Neumann expansion [75]. It also requires small uncertainties, and furthermore, it is restrictive since it would be effective only on time-independent problems. Sometimes QoI is the moments of the random solution to PDE, and the moment equations methods are suitable by constructing equations from the averages of the random PDEs. But they cannot avoid the so-called closure problem [78].

Lastly, there are approaches based on approximating the PDE solution by a surrogate model (or reduced order model). In this class, we find the Generalized

Polynomial Chaos (gPC) approach [28, 72, 74], as well as reduced basis methods [34, 58]. The general idea is to approximate the PDE solution in terms of a set of basis functions in the parameter space. In gPC methods, we use a basis expansion in orthogonal polynomials. Typically, an approximate gPC solution is then computed using stochastic Galerkin [3, 28] or stochastic collocation [4, 8, 24, 73] methods. However, as our PDE takes a highly complicated and non-linear form, it is often a challenge to solve the Galerkin systems efficiently. Moreover, gPC methods do not necessarily avoid the curse of dimensionality [54], and in high-dimensional systems, the approximation errors can be substantial, unless an unreasonably large number of samples is acquired [72]. Reduced basis methods, on the other hand, choose for the basis so-called snapshots of the solution, i.e. PDE solutions at a number of suitably chosen parameter values.

The aim of this work is to study the use of deep neural networks to solve random PDEs in uncertainty quantification. The use of neural and deep neural networks to solve deterministic PDEs has been of large recent interest, and the work generally falls into two categories. In the first category, we employ neural and deep neural networks in association with other traditional methods (e.g. FE, FD and FV) mostly to reduce the computational complexity of the methods. For example, in [44, 77], the neural networks successfully accelerate the computation of difference equations that arise in the finite difference solution for a PDE. Particularly, different neural network architectures are used to represent the linear system of equations derived from the difference equations and the solution to that of equations is obtained by minimizing the neural networks' energy functions. Additionally, the authors in [50, 51] followed an approach of linear B-splines to estimate the solution of ordinary differential equations (ODEs) while using feed-forward neural networks to govern the spline parameters. However, this method is not easily feasible for high dimensions [43]. In [69], aiming at hasting simulation of Navier-Stokes equations in an Eulerian framework, they

establish Convolutional Neural Networks (CNNs) to address the sparse system of linear equations originated from the Poisson equation of pressure. In [64], an hastened (opposed to multi-scale simulation) approach is proposed based on regression analysis. They train the local regression models by using a training dataset that is obtained from pre-analysis or experiments. Those local models connect the local nodal points in a nonlinear sense and each is recognised as a discretized equation for the PDE. For the purpose, they adopt neural networks and polynomial functions as the regression model in their study.

In the second category, researchers utilise neural and deep neural networks as a replacement of the traditional methods. Typically in these methods, a neural or deep neural network is trained to approximate the solution to the deterministic PDE, i.e. it is used as a surrogate model for the (deterministic) PDE solution. For example in [43], they address initial and boundary value problems by using neural networks. There are two parts for the trial solution. The first part satisfies the initial or boundary conditions with no adjustable parameters, and the second part consists of a neural network which is independent of those conditions. To this end, they tune parameters of the neural network by optimising the mean-square-error (MSE) over specified collocation points. An alternative approach, called Deep Galerkin Method (DGM), was proposed in [63], with some crucial differences, specially (1) they forfeit the trail function that strictly enforces the initial and boundary conditions to fulfill weakly these during the neural network training, (2) they start to build relatively deeper neural network architectures and train them by using the advanced mini-batch optimization techniques, and (3) with a certain stopping criteria, they minimise MSE iteratively over a set of randomly sampled points per iteration and this ensure the method is mesh-free. Also in [48], they research a hybrid variation of the approach proposed in [43]. In [61], he introduces the method of constrained backpropagation to train neural networks and these approximate the solution to nonlinear elliptic and parabolic

PDEs. Meanwhile in [32, 35, 62, 71], they propose an approach for solving high-dimensional parabolic PDEs by training a set of deep networks. To achieve these, they adopt the probabilistic representation for the solution to PDE by using backward stochastic differential equations (BSDEs) or the rough theory and train a neural network to approximate it by regarding the gradient of it as the role of policy function linked to the reinforcement learning.

Recently, neural and deep neural networks have also been investigated in [36, 53, 70, 76] for solving random PDEs. [53] follows the idea of DGM to construct MSE with additional sampling on random parameters. Hard and soft assignments have been proposed, but with no comparison, in the numerical experiment though they claim the hard assignments would perform better than the soft. However, we will show at least for the particular set-up of the experiment the hard potentially would behave less efficiently than the soft. Also, we perform additional experiments for the case with randomness on the boundary condition. Meanwhile in [62] they explore the possibility of training deep neural nets via probabilistic representation for the solution of families of PDEs in the context of mathematical finance without boundary condition. Notably, we cover additional types of random PDEs with boundary conditions.

The contributions of this work are to show how deep neural network surrogates can be constructed for high-dimensional random PDEs by expanding on the construction in [62], and to confirm their effectiveness in a range of numerical tests. Of course, one of the benefits of constructing a surrogate model, as opposed to for example Monte Carlo sampling, is that once the surrogate for the PDE solution is constructed, we can use this surrogate for computing as many different QoIs as required. Therefore, it is common to distinguish the so-called offline cost of setting up the surrogate model, from the online cost of actually using the surrogate model to perform the tasks of interest. The offline cost is only acquired once, and can be comparably large, but the online cost is crucially small. In our

numerical experiments, we use a deep neural network to solve the inverse problem of recovering the value of a parameter in the PDE given observations of the PDE solution, i.e. a model calibration problem. Due to the small online cost of the deep neural network, this can be done very efficiently.

5.2 Deep learning for random PDEs

5.2.1 General goal

We aim to approximate the solution $u(x,t,\theta)$ (or $u(x,\theta)$) for the following (partial or ordinary) differential equation

$$\begin{aligned}
\mathcal{L}(x,t,\theta;u(x,t,\theta)) &= 0, \quad t\in[0,T], x\in\mathcal{D}, \theta\in L_2(\Omega,\mathbb{R}^{d_\theta})\sim P_\theta,\\
\mathcal{I}(x,\theta;u(x,0,\theta)) &= 0, \quad x\in\mathcal{D}, \theta\in L_2(\Omega,\mathbb{R}^{d_\theta}),\\
\mathcal{B}(x,t,\theta;u(x,t,\theta)) &= 0, \quad t\in[0,T], x\in\partial\mathcal{D}, \theta\in L_2(\Omega,\mathbb{R}^{d_\theta})\sim P_\theta,
\end{aligned} \tag{5.1}$$

where θ represents the $\mathbb{R}^{d_\theta}$-valued square integrable random parameter governed by the given distribution P_θ, $\mathcal{L}(\cdot)$ is a general differential operator that includes the time derivatives (corresponding to t), spatial derivatives (corresponding to x) and linear and nonlinear terms, x is a position vector lying on a bounded continuous spatial domain $\mathcal{D}\subset\mathbb{R}^d$. Also, $\mathcal{I}(\cdot)$ and $\mathcal{B}(\cdot)$ represents, respectively, the initial and boundary conditions and include the differential, linear, or nonlinear operators.

5.2.2 Practical object

The concrete examples we focus on are:

Parabolic equation with terminal condition We consider the following d-dimensional random PDE:

$$\frac{\partial u}{\partial t}(x,t,\theta) + \nabla \cdot (\kappa(x,\theta)\nabla u(x,t,\theta)) - c(x,t,\theta)u(x,t,\theta) = 0 \qquad \text{in} \qquad (\mathcal{D},[0,T]) \subset (\mathbb{R}^d,\mathbb{R}^+), \tag{5.2}$$

where the terminal condition is known, the function $c : \mathbb{R}^d \times \mathbb{R} \times L_2(\Omega,\mathbb{R}^{d_\theta}) \to \mathbb{R}$ is continuous in (x,t,θ) and bounded in x. The random function $\kappa(x,\theta)$ or its exponent is assumed to have the following form:

$$\sum_{i=1}^{d_\theta} \phi_i(x)\varphi(\theta_i) + \phi_0(x), \tag{5.3}$$

where $||\phi_i||_\infty$ not increases with the index i, φ globally Lipschitz, $\theta \sim \mathbb{P}_\theta$ and $\phi_0(x)$ is sufficiently non-negative.

Elliptic equation with boundary condition Also we address the following d-dimensional random PDE:

$$\nabla \cdot (\kappa(x,\theta)\nabla u(x,\theta)) - c(x,\theta)u(x,\theta) = 0 \qquad \text{in} \qquad \mathcal{D} \subset \mathbb{R}^d, \tag{5.4}$$

where $c : \mathbb{R}^d \times L_2(\Omega,\mathbb{R}^{d_\theta}) \to \mathbb{R}$ is non-negative, the boundary condition is known. The random function $\kappa(x,\theta)$ or its exponent is assumed to have the same form as (5.3).

Poisson's equation Lastly, we investigate this type of d-dimensional random PDE:

$$\nabla \cdot (\kappa(x,\theta)\nabla u(x,\theta)) - c(x,\theta)u(x,\theta) = f(x) \qquad \text{in} \qquad \mathbb{R}^d, \tag{5.5}$$

where $c : \mathbb{R}^d \times L_2(\Omega, \mathbb{R}^{d_\theta}) \to \mathbb{R}$ is positive and bounded from below, $f : \mathbb{R}^d \to \mathbb{R}$ belongs to $\mathcal{C}^2$ that the space contains all continuous derivatives up to the second order and $\kappa(x, \theta)$ or its exponent is assumed to have the same form as (5.3).

We remark that the randomness of (5.2), (5.4) or (5.5) only comes from θ and for each practical problem the set of functions $\{\phi_i\}_{i=1}^N$ is given.

5.2.3 How to treat the additional random parameter θ

Since $\theta \in L_2(\Omega, \mathbb{R}^{d_\theta})$ satisfies the given probability distribution $\mathbb{P}_\theta$, assume we can sample jth vector of parameters

$$\theta^{(j)} := (\theta_1^{(j)}, \theta_2^{(j)}, \dots, \theta_{d_\theta}^{(j)}).$$

Two perspectives Fix $t^* \in [0, T]$. We consider two to approximate (by applying the solver of neural network) $u(x, t^*)$ for all $x \in \mathcal{D}$:

- For fixed $\omega^{(j)} \in \Omega$, this makes $\theta^{(j)}$ deterministic and renders a problem of solving deterministic PDE. The two things have to be mentioned:
  1. for the $\theta^{(j)}$, the output of the network is the approximation of $u_{\theta^{(j)}}(x, t^*)$
  2. if we are interested in quantify $\mathbb{E}[P(u_\theta(x, t^*))]$ where the function $P(\cdot)$ is given, we can apply MLMC in subsampling to improve the computational efficiency.
- Alternatively, we can also add the parameter θ as part of the input and the feed-forward network will give the estimation of $u(x, t^*, \theta)$. More precisely,

we change input from

$$\mathbf{x} := (x_1, x_2, \ldots, x_d) \quad \text{to} \quad (\mathbf{x}, \theta) := (x_1, x_2, \ldots, x_d, \ldots, \theta_1, \theta_2, \ldots, \theta_{d_\theta})$$

and learn a general form of u that vary both in $\mathbf{x}$ and θ . Besides, we are also potentially able to apply MLMC in subsampling to reduce computational cost if aiming at quantifying the expectation of some functionals on $u(x, t^*, \theta)$.

- In our proposed algorithms, we treat θ as an additional argument of the input because this avoids additional interpolations, for example, if we need the solution to perform calibration.

5.2.4 Learning the random PDE solution based on probabilistic representation

We consider three cases separately:

- linear random parabolic PDE (5.2) with terminal condition ($u(x, T, \theta) = g(x)$),
- linear random elliptic PDE (5.4) with boundary condition ($u(x, \theta) = g(x)$ where $x \in \partial\mathcal{D}$),
- Poisson's equation (5.5) with the unbounded domain,

where we assume the function $g : \mathbb{R}^d \to \mathbb{R}$ is Lipschitz continuous in x.

Derivation of recursive relationship for the type (5.2) For a given $(x, t) \in \mathbb{R}^d \times \mathbb{R}^+$, suppose that $b : \mathbb{R}^d \times \mathbb{R}^+ \times L_2(\Omega, \mathbb{R}^{d_\theta}) \to \mathbb{R}^d$ and σ :

$\mathbb{R}^d \times \mathbb{R}^+ \times L_2(\Omega, \mathbb{R}^{d_\theta}) \to \mathbb{R}^{d \times d_w}$ are Lipschitz continuous for all arguments. We define $X_s^{x,t,\theta}$ as

$$dX_s^{x,t,\theta} = b(X_s^{x,t,\theta}, s, \theta)ds + \sigma(X_s^{x,t,\theta}, s, \theta)dW_s, \quad X_t^{x,t,\theta} = x \in \mathbb{R}^d, \tag{5.6}$$

where θ is $\mathcal{F}_0$-measurable (independent of d_w-dimensional Brownian $W_{t\geq 0}$) and satisfies the distribution $\mathbb{P}_\theta$. Under the Lipschitz continuous assumption of the function $g : \mathbb{R}^d \to \mathbb{R}$, we apply Itô formula and product rule to the process defined by

$$H_r^\theta = u(X_r^{x,t,\theta}, t, \theta)e^{-\int_t^r c(X_s^{x,t,\theta}, s, \theta)ds}, \quad r \in [t, T],$$

then let $a = (a_{ij}) := \sigma \cdot \sigma^T$, we have that for $r \in [t, T]$,

$$\begin{aligned} H_r^\theta - H_t^\theta &= u(X_r^{x,t,\theta}, r, \theta)e^{-\int_t^r c(X_s^{x,t,\theta}, s, \theta)ds} - u(X_t^{x,t,\theta}, t, \theta) \\ &= \int_t^r e^{-\int_t^s c(X_\tau^{x,t,\theta}, \tau, \theta)d\tau} \left(\frac{\partial u}{\partial s} + \sum_{i=1}^d b_i \frac{\partial u}{\partial x_i} + \frac{1}{2} \sum_{i=j}^d a_{ij} \frac{\partial u}{\partial x_i x_j} - c(X_s^{x,t,\theta}, s, \theta)u \right) ds \\ &+ \int_t^r e^{-\int_t^s c(X_\tau^{x,t,\theta}, \tau, \theta)d\tau} \sum_{i,j} \frac{\partial u}{\partial x_i} \sigma_{ij}(X_s^{x,t,\theta}, s, \theta)dW_s^{(j)}. \end{aligned} \tag{5.7}$$

Notice that arguments inside the integrals are evaluated at $(X_s^{x,t,\theta}, s, \theta)$ and if we assume u satisfies (5.2), $2\kappa = diagonal(a)$ and $\nabla_x \kappa = b$, we conclude that

$$H_r^\theta - H_t^\theta = \int_t^r e^{-\int_t^s c(X_\tau^{x,t,\theta}, \tau, \theta)d\tau} \sum_{i,j} \frac{\partial u}{\partial x_i} \sigma_{ij}(X_s^{x,t,\theta}, s, \theta)dW_s^{(j)}. \tag{5.8}$$

In order to obtain an implementable algorithm, we would need to discretise the integrals that arise in (5.8). To that end take a partition of $[t, T]$ with N steps denoted by

$$\pi := \{t_0 = t < t_1 < \ldots < t_m < \ldots < t_N = T\} \tag{5.9}$$

and consider an approximation of (5.6) by $X_{t_i}^{\theta,\pi}$. For simplicity we approximate all integrals arising by Riemann sums always taking the left-hand point when approximating the value of the integrand. Also, there are more sophisticated

methods available such as quadrature rules. Since we don't know the exact solution u to the PDE (5.2), we will also have to approximate the gradient $\frac{\partial u}{\partial x}$ by $\mathcal{R}\eta_2$ while approximating the solution u by $\mathcal{R}\eta_1$. To formulate the learning task in the discrete time, we then write it recursively as

$$\begin{aligned} u(X^{\theta,\pi}_{t_N}, t_N, \theta) &= D^\pi(t_N, T) g(X^{\theta,\pi}_{t_N}) \\ D^\pi(t, t_{m+1}) u(X^{\theta,\pi}_{t_{m+1}}, t_{m+1}, \theta) &= D^\pi(t, t_m) u(X^{\theta,\pi}_{t_m}, t_m, \theta) \\ &\quad + \int_{t_m}^{t_{m+1}} D^\pi(t, s) \sum_{i,j} \frac{\partial u}{\partial x_i}(X^{\theta,\pi}_s, s, \theta) \sigma_{ij}(X^{\theta,\pi}_s, s, \theta) dW^{(j)}_s \quad \text{for} \quad t_{m+1} \in \pi, \end{aligned} \tag{5.10}$$

where $D^\pi(t,s) = e^{-\int_t^s c(X^{\theta,\pi}_r, r, \theta) dr}$. Next consider two deep neural network approximations

$$\mathcal{R}\eta_1(x, t_m, \theta) \approx u(x, t_m, \theta) \quad \text{and} \quad \mathcal{R}\eta_2(x, t_m, \theta) \approx \frac{\partial u}{\partial x}(x, t_m, \theta) \quad \text{for } \eta_1, \eta_2 \in \mathcal{DN}. \tag{5.11}$$

These approximations depends on weights $\eta_1 \in \mathbb{R}^{d_{\eta_1}}$ and $\eta_2 \in \mathbb{R}^{d_{\eta_2}}$. We set the learning task as the following optimisation problem:

$$(\eta_1^*, \eta_2^*) = \arg\min_{(\eta_1,\eta_2)} \mathbb{E}\left[\left| g(X^{\theta,\pi}_T) - \mathcal{R}_{\eta_1}(X^\theta_T, t_N, \theta) \right|^2 + \frac{1}{N} \sum_{m=0}^{N-1} |\mathcal{E}^{\pi,(\eta_1,\eta_2)}_{m+1}|^2 \right], \tag{5.12}$$

where $\mathcal{E}^{\pi,(\eta_1,\eta_2)}_{m+1}$ is given by (matrix form)

$$\begin{aligned} \mathcal{E}^{\pi,(\eta_1,\eta_2)}_{m+1} &= D^\pi(t, t_{m+1}) \mathcal{R}_{\eta_1}(X^{\theta,\pi}_{t_{m+1}}, t_{m+1}, \theta) - D^\pi(t, t_m) \mathcal{R}_{\eta_1}(X^{\theta,\pi}_{t_m}, t_m, \theta) \\ &\quad - D^\pi(t, t_m) \mathcal{R}_{\eta_2}(X^{\theta,\pi}_{t_m}, t_m, \theta) \sigma(X^{\theta,\pi}_{t_m}, t_m, \theta)(W_{t_{m+1}} - W_{t_m}), \end{aligned} \tag{5.13}$$

Note that in practice any minimisation algorithm can only be expected to find the approximation $(\eta_1^{\diamond,n}, \eta_2^{\diamond,n})$ of (η_1^*, η_2^*) and n means we have n sampling points in each mini-batch training step. The complete learning method is stated as Algorithm 3.

Algorithm 3: PDE probabilistic representation learning for the type (5.2)

1 Initialize DNN parameters (η_1, η_2) (along the timeline), the size of mini-batch n and the partition π with N steps as (5.9);

2 **for** *j=1:n* **do**

generate $\{X_{t_0}^{\theta^{(j)},\pi}, \theta^{(j)}\}$ and simulate the approximation $X_{t_m}^{\theta^{(j)},\pi}$ of (5.6) for any $t_m \in \pi$.;

for *k=0:N-1* **do**

Compute

$$D^{\pi,j}(t_0, t_k) = e^{-\sum_{m=0}^{k-1} c(X_{t_m}^{\theta^{(j)},\pi}, t_m, \theta^{(j)})(t_{m+1}-t_m)}$$

3 find $(\eta_1^{\diamond,n}, \eta_2^{\diamond,n})$ where

$$(\eta_1^{\diamond,n}, \eta_2^{\diamond,n}) = \widehat{\arg\min}_{(\eta_1,\eta_2)} \frac{1}{n} \sum_{j=1}^{n} \left[\left| g(X_T^{\theta^{(j)},\pi}) - \mathcal{R}_{\eta_1}(X_T^{\theta^{(j)}}, t_N, \theta) \right|^2 + \frac{1}{N} \sum_{m=0}^{N-1} |\mathcal{E}_{m+1}^{\pi,j,(\eta_1,\eta_2)}|^2 \right],$$

where $\mathcal{E}_{m+1}^{\pi,j,(\eta_1,\eta_2)}$ is given by

$$\begin{aligned}\mathcal{E}_{m+1}^{\pi,j,(\eta_1,\eta_2)} =& D^{\pi,j}(t, t_{m+1}) \mathcal{R}_{\eta_1}(X_{t_{m+1}}^{\theta^{(j)},\pi}, t_{m+1}, \theta^{(j)}) - D^{\pi,j}(t, t_m) \mathcal{R}_{\eta_1}(X_{t_m}^{\theta^{(j)},\pi}, t_m, \theta^{(j)}) \\ & - D^{\pi,j}(t, t_m) \mathcal{R}_{\eta_2}(X_{t_m}^{\theta^{(j)},\pi}, t_m, \theta^{(j)}) \sigma(X_{t_m}^{\theta^{(j)},\pi}, t_m, \theta^{(j)})(W_{t_{m+1}}^j - W_{t_m}^j),\end{aligned}$$

4 **Return** $(\eta_1^{\diamond,n}, \eta_2^{\diamond,n})$.

Derivation of recursive relationship for the type (5.4) We begin this derivation by defining $X_t^{x,\theta}$ inside $\mathcal{D} \subset \mathbb{R}^d$ as

$$dX_t^{x,\theta} = b(X_t^{x,\theta}, \theta)dt + \sigma(X_t^{x,\theta}, \theta)dW_t, \quad X_0^{x,\theta} = x \in \mathcal{D}, \tag{5.14}$$

where $b : \mathbb{R}^d \times L_2(\Omega, \mathbb{R}^{d_\theta}) \to \mathbb{R}^d$ and $\sigma : \mathbb{R}^d \times L_2(\Omega, \mathbb{R}^{d_\theta}) \to \mathbb{R}^{d \times d_w}$ are Lipschitz continuous for all arguments, θ is $\mathcal{F}_0$-measurable (independent of d_w-dimensional Brownian $W_{t\geq 0}$) and satisfies the distribution $\mathbb{P}_\theta$. But the trajectories may travel outside the domain $\mathcal{D}$. To overcome this difficulty, we introduce the stopping time $\tau_{\partial\mathcal{D}} = \inf\{t \geq 0 | X_t^{x,\theta} \notin \mathcal{D}\}$. This is the first passage time to the boundary $\partial\mathcal{D}$. With the Lipschitz continuous assumption of the function $g : \mathbb{R}^d \to \mathbb{R}$, we again apply Itô formula and product rule to the process defined by

$$H_t^\theta = u(X_{t\wedge\tau_{\partial\mathcal{D}}}^{x,\theta}, \theta)e^{-\int_0^{t\wedge\tau_{\partial\mathcal{D}}} c(X_s^{x,\theta},\theta)ds}, \quad t \geq 0,$$

where we denote $t \wedge \tau_{\partial\mathcal{D}}$ as $\min(t, \tau_{\partial\mathcal{D}})$. Then still let $a = (a_{ij}) := \sigma \cdot \sigma^T$, we have that for $t \geq 0$,

$$\begin{aligned} H_t^\theta - H_0^\theta &= u(X_{t\wedge\tau_{\partial\mathcal{D}}}^{x,\theta}, \theta)e^{-\int_0^{t\wedge\tau_{\partial\mathcal{D}}} c(X_s^{x,\theta},\theta)ds} - u(X_0^{x,\theta}, \theta) \\ &= \int_0^{t\wedge\tau_{\partial\mathcal{D}}} e^{-\int_0^s c(X_\gamma^{x,\theta},\theta)d\gamma} \left(\sum_{i=1}^d b_i \frac{\partial u}{\partial x_i} + \frac{1}{2}\sum_{i=j}^d a_{ij}\frac{\partial u}{\partial x_i x_j} - c(X_s^{x,\theta},\theta)u \right) ds \\ &+ \int_0^{t\wedge\tau_{\partial\mathcal{D}}} e^{-\int_0^s c(X_\gamma^{x,\theta},\theta)d\gamma} \sum_{i,j} \frac{\partial u}{\partial x_i}\sigma_{ij}(X_s^{x,\theta},\theta)dW_s^{(j)}. \end{aligned} \tag{5.15}$$

With the same reasoning as the first case but assuming the function u satisfies (5.4), we have

$$H_t^\theta - H_0^\theta = \int_0^{t\wedge\tau_{\partial\mathcal{D}}} e^{-\int_0^s c(X_\gamma^{x,\theta},\theta)d\gamma} \sum_{i,j} \frac{\partial u}{\partial x_i}\sigma_{ij}(X_s^{x,\theta},\theta)dW_s^{(j)}. \tag{5.16}$$

To be implementable, we are restricted to $[0, T]$ and consider the partition in the form of (5.9) with N steps where $t_0 = 0$. With the approximation of (5.14) by

$X_{t_i}^{\theta,\pi}$, the discrete recursive relationship of the learning task is formulated as

$$\begin{aligned} u(X_{\tau_{\partial\mathcal{D}}}^{\theta,\pi},\theta) &= g(X_{\tau_{\partial\mathcal{D}}}^{\theta,\pi}) \quad \text{if} \quad t_N \geq \tau_{\partial\mathcal{D}} \\ D^\pi(0,t_{m+1})u(X_{t_{m+1}}^{\theta,\pi},\theta) &= D^\pi(0,t_m)u(X_{t_m}^{\theta,\pi},\theta) \\ &\quad + \int_{t_m}^{t_{m+1}} D^\pi(0,s)\sum_{i,j}\frac{\partial u}{\partial x_i}(X_s^{\theta,\pi},\theta)\sigma_{ij}(X_s^{\theta,\pi},\theta)dW_s^{(j)} \quad \text{for} \quad t_{m+1}\in\pi\leq\tau_{\partial\mathcal{D}}, \end{aligned} \tag{5.17}$$

where $D^\pi(t,s) = e^{-\int_t^s c(X_\gamma^{\theta,\pi},\theta)d\gamma}$. Next we consider two deep neural network approximations

$$\mathcal{R}\eta_1(x,\theta)\approx u(x,\theta) \quad \text{and} \quad \mathcal{R}\eta_2(x,\theta)\approx\frac{\partial u}{\partial x}(x,\theta) \quad \text{for } \eta_1,\eta_2\in\mathcal{DN}. \tag{5.18}$$

Notice that these approximations do not have time dependence compared to (5.11). The learning task is equivalent to solving the minimisation problem:

$$\begin{aligned} (\eta_1^*,\eta_2^*) = \arg\min_{(\eta_1,\eta_2)}\mathbb{E}\Bigg[\Bigg(\Big|g(X_{\tau_{\partial\mathcal{D}}}^{\theta,\pi}) - \mathcal{R}_{\eta_1}(X_{\tau_{\partial\mathcal{D}}}^{\theta},\theta)\Big|^2 + \frac{1}{N_{\tau_{\partial\mathcal{D}}}}\sum_{m=0}^{N_{\tau_{\partial\mathcal{D}}}-1}|\mathcal{E}_{m+1}^{\pi,(\eta_1,\eta_2)}|^2\Bigg)\mathbf{1}_{\tau_{\partial\mathcal{D}}\leq t_N} \\ + \frac{1}{N}\sum_{m=0}^{N-1}|\mathcal{E}_{m+1}^{\pi,(\eta_1,\eta_2)}|^2\mathbf{1}_{\tau_{\partial\mathcal{D}}>t_N}\Bigg], \end{aligned} \tag{5.19}$$

where $N_{\tau_{\partial\mathcal{D}}} = \inf\{m : X_{t_m}^{\theta,\pi}\notin\mathcal{D}\}$ and $\mathcal{E}_{m+1}^{\pi,(\eta_1,\eta_2)}$ is given by

$$\begin{aligned} \mathcal{E}_{m+1}^{\pi,(\eta_1,\eta_2)} &= D^\pi(0,t_{m+1})\mathcal{R}_{\eta_1}(X_{t_{m+1}}^{\theta,\pi},\theta) - D^\pi(0,t_m)\mathcal{R}_{\eta_1}(X_{t_m}^{\theta,\pi},\theta) \\ &\quad - D^\pi(0,t_m)\mathcal{R}_{\eta_2}(X_{t_m}^{\theta,\pi},\theta)\sigma(X_{t_m}^{\theta,\pi},\theta)(W_{t_{m+1}}-W_{t_m}). \end{aligned} \tag{5.20}$$

We remark that in [14] and [15], they propose advanced methods to find the approximated stopping time $t_{N_{\tau_{\partial\mathcal{D}}}}$ rather than the trivial stopping procedure given above. The entire learning method is stated as Algorithm 4.

Algorithm 4: PDE probabilistic representation learning for type (5.4)

1 Initialize (η_1, η_2), the size of mini-batch n and the partition π with N steps as (5.9);

2 **for** *j=1:n* **do**

generate $\{X_{t_0}^{\theta^{(j)},\pi}, \theta^{(j)}\}$ and simulate the approximation $X_{t_m}^{\theta^{(j)},\pi}$ of (5.14) for $t_m \in \pi \leq \tau_{\partial\mathcal{D}}$.;

for *k=0:N-1* **do**

Compute

$$D^{\pi,j}(t_0, t_k) = e^{-\sum_{m=0}^{k-1} c(X_{t_m}^{\theta^{(j)},\pi}, t_m, \theta^{(j)})(t_{m+1}-t_m)}$$

3 find $(\eta_1^{\diamond,n}, \eta_2^{\diamond,n})$ where

$$(\eta_1^{\diamond,n}, \eta_2^{\diamond,n}) = \widehat{\arg\min_{(\eta_1,\eta_2)}} \frac{1}{n} \sum_{j=1}^{n} \left[\frac{1}{n_{\tau_{\partial\mathcal{D}}}} \sum_{j=1}^{n_{\tau_{\partial\mathcal{D}}}} \left(|g(x_\tau^{(i,j)}) - \mathcal{R}_{\eta_1}(x_\tau^{(i,j)}, \theta^{(j)})|^2 + \frac{1}{N_{\tau_{\partial\mathcal{D}}}} \sum_{m=0}^{N_{\tau_{\partial\mathcal{D}}}-1} |\mathcal{E}_{m+1}^{\pi,j,(\eta_1,\eta_2)}|^2 \right) + \frac{1}{n - n_{\tau_{\partial\mathcal{D}}}} \sum_{j=n-n_{\tau_{\partial\mathcal{D}}}+1}^{n} \left(\frac{1}{N} \sum_{m=0}^{N-1} |\mathcal{E}_{m+1}^{\pi,j,(\eta_1,\eta_2)}|^2 \right) \right],$$

where $n_{\tau_{\partial\mathcal{D}}}$ is the total number of approximations $X_{t_m}^{\theta^{(j)},\pi}$ whose simulation is stopped before t_N and $\mathcal{E}_{m+1}^{\pi,j,(\eta_1,\eta_2)}$ is given by

$$\begin{aligned}\mathcal{E}_{m+1}^{\pi,j,(\eta_1,\eta_2)} =& D^{\pi,j}(0, t_{m+1}) \mathcal{R}_{\eta_1}(X_{t_{m+1}}^{\theta^{(j)},\pi}, \theta^{(j)}) - D^{\pi,j}(0, t_m) \mathcal{R}_{\eta_1}(X_{t_m}^{\theta^{(j)},\pi}, \theta^{(j)}) \\ & - D^{\pi,j}(0, t_m) \mathcal{R}_{\eta_2}(X_{t_m}^{\theta^{(j)},\pi}, \theta^{(j)}) \sigma(X_{t_m}^{\theta^{(j)},\pi}, \theta^{(j)})(W_{t_{m+1}}^j - W_{t_m}^j),\end{aligned}$$

4 **Return** $(\eta_1^{\diamond,n}, \eta_2^{\diamond,n})$.

Derivation of recursive relationship for type (5.5) Suppose $X_t^{x,\theta}$ solves the stochastic differential equation

$$dX_t^{x,\theta} = b(X_t^{x,\theta}, \theta)dt + \sigma(X_t^{x,\theta}, \theta)dW_t, \quad X_0^{x,\theta} = x \in \mathbb{R}^d, \quad t > 0, \tag{5.21}$$

where $b : \mathbb{R}^d \times L_2(\Omega, \mathbb{R}^{d_\theta}) \to \mathbb{R}^d$ and $\sigma : \mathbb{R}^d \times L_2(\Omega, \mathbb{R}^{d_\theta}) \to \mathbb{R}^{d \times d_w}$ are Lipschitz continuous for all arguments, θ is $\mathcal{F}_0$-measurable (independent of d_w-dimensional Brownian $W_{t\geq 0}$) and satisfies the distribution $\mathbb{P}_\theta$. Again we apply Itô formula and product rule to the process defined by

$$H_t^\theta = u(X_t^{x,\theta}, \theta)e^{-\int_0^t c(X_s^{x,\theta},\theta)ds}, \quad t \geq 0,$$

Then recalling that $a = (a_{ij}) := \sigma \cdot \sigma^T$, we have that for $t \geq 0$,

$$\begin{aligned} H_t^\theta - H_0^\theta &= u(X_t^{x,\theta}, \theta)e^{-\int_0^t c(X_s^{x,\theta},\theta)ds} - u(X_0^{x,\theta}, \theta) \\ &= \int_0^t e^{-\int_0^s c(X_\gamma^{x,\theta},\theta)d\gamma} \left(\sum_{i=1}^d b_i \frac{\partial u}{\partial x_i} + \frac{1}{2} \sum_{i=j}^d a_{ij} \frac{\partial u}{\partial x_i x_j} - c(X_s^{x,\theta}, \theta)u \right) ds \\ &+ \int_0^t e^{-\int_0^s c(X_\gamma^{x,\theta},\theta)d\gamma} \sum_{i,j} \frac{\partial u}{\partial x_i} \sigma_{ij}(X_s^{x,\theta}, \theta) dW_s^{(j)}. \end{aligned} \tag{5.22}$$

By the same substitution as before but assuming the function u satisfies (5.5), we see

$$H_t^\theta - H_0^\theta = \int_0^t e^{-\int_0^s c(X_\gamma^{x,\theta},\theta)d\gamma} \left(f(X_s^{x,\theta})ds + \sum_{i,j} \frac{\partial u}{\partial x_i} \sigma_{ij}(X_s^{x,\theta}, \theta) dW_s^{(j)} \right). \tag{5.23}$$

To be plausible, we restrict t from $[0, \infty]$ to $[0, T]$ and consider the partition defined in (5.9) with N steps where $t_0 = 0$. By introducing the approximation of

(5.21) by $X_{t_i}^{\theta,\pi}$, we formulate the learning task as the following recursive system:

$$D^{\pi}(0,t_{m+1})u(t_{m+1},X_{t_{m+1}}^{\theta,\pi}) = D^{\pi}(0,t_m)u(t_m,X_{t_m}^{\theta,\pi}) + \int_{t_m}^{t_{m+1}} D^{\pi}(0,s)\left(f(X_s^{\theta,\pi})ds + \sum_{i,j}\frac{\partial u}{\partial x_i}(X_s^{\theta,\pi},\theta)\sigma_{ij}(X_s^{\theta,\pi},\theta)dW_s^{(j)}\right) \quad \text{for} \quad t_{m+1}\in\pi, \tag{5.24}$$

where $D^{\pi}(t,s) = e^{-\int_t^s c(X_\gamma^{\theta,\pi},\theta)d\gamma}$. Next we consider two deep neural network approximations with the same notation as (5.18). The learning task is to solve the minimisation problem:

$$(\eta_1^*,\eta_2^*) = \arg\min_{(\eta_1,\eta_2)}\mathbb{E}\left[\frac{1}{N}\sum_{m=0}^{N-1}|\mathcal{E}_{m+1}^{\pi,(\eta_1,\eta_2)}|^2\right], \tag{5.25}$$

where $\mathcal{E}_{m+1}^{\pi,(\eta_1,\eta_2)}$ is given by

$$\begin{aligned}\mathcal{E}_{m+1}^{\pi,(\eta_1,\eta_2)} =& D^{\pi}(0,t_{m+1})\mathcal{R}_{\eta_1}(X_{t_{m+1}}^{\theta,\pi},\theta) - D^{\pi}(0,t_m)\mathcal{R}_{\eta_1}(X_{t_m}^{\theta,\pi},\theta)\\ &- D^{\pi}(0,t_m)\left(f(X_{t_m}^{\theta,\pi})(t_{m+1}-t_m) + \mathcal{R}_{\eta_2}(X_{t_m}^{\theta,\pi},\theta)\sigma(X_{t_m}^{\theta,\pi},\theta)(W_{t_{m+1}}-W_{t_m})\right).\end{aligned} \tag{5.26}$$

We then propose the Algorithm 5.

5.2.5 Learning the random PDE solution directly

For type (5.4), we need to resolve the additional issue of sampling stopping time as accurately as possible under the limited computational budget. But if the position argument x lies in a bounded domain $\mathcal{D}\subset\mathbb{R}^d$, it is natural to set up the learning algorithm for the solution u of (5.4) by using PDE (5.4) itself as score function. Let $\mathcal{R}\eta$ approximate u for $\eta\in\mathcal{DN}$. We then establish the learning task

Algorithm 5: PDE probabilistic representation learning for type (5.5)

1 Initialize DNN parameters (η_1, η_2) (along the timeline), the size of mini-batch n and the partition π with N steps as (5.9);

2 **for** *j=1:n* **do**

generate $\{X_{t_0}^{\theta^{(j)},\pi}, \theta^{(j)}\}$ and simulate the approximation $X_{t_m}^{\theta^{(j)},\pi}$ of (5.21) for any $t_m \in \pi$.;

for *k=0:N-1* **do**

Compute

$$D^{\pi,j}(t_0, t_k) = e^{-\sum_{m=0}^{k-1} c(X_{t_m}^{\theta^{(j)},\pi}, \theta^{(j)})(t_{m+1}-t_m)}$$

3 find $(\eta_1^{\diamond,n}, \eta_2^{\diamond,n})$ where

$$(\eta_1^{\diamond,n}, \eta_2^{\diamond,n}) = \widehat{\arg\min}_{(\eta_1,\eta_2)} \frac{1}{n} \sum_{j=1}^{n} \left[\frac{1}{N} \sum_{m=0}^{N-1} |\mathcal{E}_{m+1}^{\pi,j,(\eta_1,\eta_2)}|^2 \right],$$

where $\mathcal{E}_{m+1}^{\pi,j,(\eta_1,\eta_2)}$ is given by

$$\mathcal{E}_{m+1}^{\pi,j,(\eta_1,\eta_2)} = D^{\pi,j}(0, t_{m+1}) \mathcal{R}_{\eta_1}(X_{t_{m+1}}^{\theta^{(j)},\pi}, \theta^{(j)}) - D^{\pi,j}(0, t_m) \mathcal{R}_{\eta_1}(X_{t_m}^{\theta^{(j)},\pi}, \theta^{(j)})$$
$$- D^{\pi,j}(0, t_m) \Big(f(X_{t_m}^{\theta,\pi})(t_{m+1} - t_m) + \mathcal{R}_{\eta_2}(X_{t_m}^{\theta^{(j)},\pi}, \theta^{(j)}) \sigma(X_{t_m}^{\theta^{(j)},\pi}, \theta^{(j)})(W_{t_{m+1}}^j - W_{t_m}^j) \Big),$$

4 **Return** $(\eta_1^{\diamond,n}, \eta_2^{\diamond,n})$.

softly as

$$\eta^* = \arg\min_{\eta} ||\nabla\cdot(\kappa(x,\theta)\nabla\mathcal{R}\eta(x,\theta)) - c(x,\theta)\mathcal{R}\eta(x,\theta)||_{\mathcal{D}\times\mathbb{R}^{d_\theta}} + ||\mathcal{R}\eta(x,\theta) - g(x)||_{\partial\mathcal{D}\times\mathbb{R}^{d_\theta}} \tag{5.27}$$

in some appropriate norms $||\cdot||_{\mathcal{D}\times\mathbb{R}^{d_\theta}}$ and $||\cdot||_{\partial\mathcal{D}\times\mathbb{R}^{d_\theta}}$. The first norm measures the error of approximating the PDE equation and the second norm measures the error of approximation of the boundary condition. To investigate, we choose

$$\begin{aligned} ||f(x,\theta)||_{\mathcal{D}\times\mathbb{R}^{d_\theta}} &= \int_{\mathcal{D}}\int_{\mathbb{R}^{d_\theta}} |f(x,\theta)|^2 dx\mathbb{P}_\theta(d\theta), \\ ||f(x,\theta)||_{\partial\mathcal{D}\times\mathbb{R}^{d_\theta}} &= \int_{\partial\mathcal{D}}\int_{\mathbb{R}^{d_\theta}} |f(x,\theta)|^2 dx\mathbb{P}_\theta(d\theta). \end{aligned}$$

Note that it is only practical to find the approximation $\eta^{\diamond,n}$ of η^*. The full procedure is described in Algorithm 6.

Algorithm 6: Softly-direct PDE learning for type (5.4)

1 Initialize DNN parameters η, the size of mini-batch n;

2 **for** *j=1:n* **do**

  sample $(x^{1,(j)}, x^{2,(j)})$ uniformly from $\mathcal{D}\times\partial\mathcal{D}$ and $\theta^{(j)}$ according the distribution $\mathbb{P}_\theta$.

3 find $\eta^{\diamond,n}$ where

$$\eta^{\diamond,n} = \widehat{\arg\min_{\eta}} \frac{1}{n}\sum_{j=1}^{n}\left[\left|\nabla\cdot(\kappa(x^{1,(j)},\theta^{(j)})\nabla\mathcal{R}\eta(x^{1,(j)},\theta^{(j)})) - c(x^{1,(j)},\theta^{(j)})\mathcal{R}\eta(x^{1,(j)},\theta^{(j)})\right|^2 + \left|\mathcal{R}\eta(x^{2,(j)},\theta^{(j)}) - g(x^{2,(j)})\right|^2\right].$$

4 **Return** $\eta^{\diamond,n}$.

Alternatively, we follow the idea from [53], and set up the learning task strongly

for type (5.4) as

$$\eta^* = \arg\min_{\eta} ||\nabla \cdot (\kappa(x,\theta)\nabla U(x,\theta,\mathcal{R}\eta(x,\theta))) - c(x,\theta)U(x,\theta,\mathcal{R}\eta(x,\theta)))||_{\mathcal{D}\times\mathbb{R}^{d_\theta}}, \tag{5.28}$$

where $U(x,\theta,\mathcal{R}\eta(x,\theta))) = u_0(x,\theta) + u_1(x,\theta,\mathcal{R}\eta(x,\theta))$ and $u_0 : \mathbb{R}^d \times \mathbb{R}^{d_\theta} \to \mathbb{R}$ satisfies the boundary condition, and we construct the function $u_1 : \mathbb{R}^d \times \mathbb{R}^{d_\theta} \times \mathbb{R} \to \mathbb{R}$ such that it does not contribute to the boundary condition. All steps are illustrated in Algorithm 7. We remark that in the literature these two algorithms are also called as the soft-assignment and hard-assignment respectively.

Algorithm 7: Strongly-direct PDE learning for type (5.4)

1 Initialize DNN parameters η, the size of mini-batch n;

2 **for** *j=1:n* **do**
  sample $x^{(j)}$ uniformly from $\mathcal{D}$ and $\theta^{(j)}$ according the distribution $\mathbb{P}_\theta$.

3 find $\eta^{\diamond,n}$ where

$$\eta^{\diamond,n} = \widehat{\arg\min_{\eta}} \frac{1}{n}\sum_{j=1}^{n}\left|\nabla \cdot (\kappa(x^{(j)},\theta^{(j)})\nabla U(x^{(j)},\theta^{(j)})) - c(x^{(j)},\theta^{(j)})U(x^{(j)},\theta^{(j)})\right|^2.$$

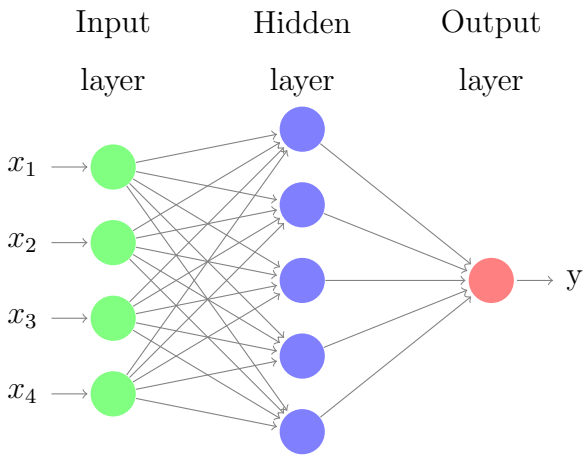
4 **Return** $\eta^{\diamond,n}$.

5.3 Implementation details for the algorithms

This section provides details for the architecture of deep networks via probabilistic representation (algorithm 3 to 5) and that of those through the non-probabilistic (algorithm 6 and 7), hyperparameters, and cost of evaluation of those deep networks.

As to the architecture of deep networks via probabilistic representation, we use

the fully-connected feedforward network and a example of this with one hidden layer of dimension d_h is below:



and $y = \sigma(xW_1 + b_1)W_2 + b_2$ where W_1 and W_2 are weights matrices of size $d_x \times d_h$ and $d_h \times d_y$, b_1 and b_2 are bias vector of size $1 \times d_h$ and $1 \times d_y$, and $\sigma(\cdot)$ is a element-wise non-linear function, commonly known as activation function. Thus, we need to set the number of layers (d_ℓ) including the input and output layers, the number of units (d_u) in each hidden layer, and the activation function before the training session. The cost of evaluation of those deep nets in terms of the number of arithmetic computation is determined by $d_x \cdot d_h + (d_\ell - 2) \cdot d_h^2 + d_y d_h$.

On the other hand, we modify the architecture in [63] (see details in Section 4.2) to construct our deep networks through the non-probabilistic by adding the additional random parameter θ to the original input x in a consistent conception.

5.4 Numerical experiment

In this section, we present numerical simulations that confirm the plausibility of applying the proposed deep learning methods including both the way based

on the probabilistic representation and the non-probabilistic in solving several types of high-dimensional random PDEs. Furthermore, numerical experiments indicate that the softly-direct PDE learning might achieve better computational performance comparing to the strongly-direct PDE learning for the particular random PDE system. We experiment the following methods:

- PDE probabilistic representation learning for the type (5.2) in example 5.4.1:
  - Forward problem: find the DNN approximation satisfying the required accuracy level to the solution to the random PDE;
  - Inverse problem: given a set of synthetic data, calibrate the DNN in terms of θ.
- PDE probabilistic representation learning for the type (5.4) in example 5.4.2 and 5.4.3,
- PDE probabilistic representation learning for the type (5.5) in example 5.4.4 ,
- Softly and Strongly-direct PDE learning for type (5.4) in example 5.4.5,

We remark that if we were able to solve the sample random PDE system by the deep learning methods based on probabilistic representation, we omitted results of those build on the non-probabilistic for avoiding redundancy. The number of layers (d_ℓ) for the deep nets trained in examples 1, 2, 3, 4, 5 are, respectively 8, 10, 14, 18 and 20. Dimensionality of each hidden layer (d_h) is set to 256. $Tanh$ nonlinearities are adopted for each hidden layer as the activation function. The Adam optimization algorithm [39] is used to find the optimal parameters and mini-batch size is set to 50.

5.4.1 Kolmogorov backward equation with the random coefficient

Forward problem In the first example, we consider the diffusion problem in one-dimensional spatial domain with random diffusion $\kappa(\theta)$. Particularly, the random PDE system is given by

$$\frac{\partial u}{\partial t}(x,t,\theta) + \kappa(\theta)\frac{\partial^2 u}{\partial x^2}(x,t,\theta) = 0 \qquad \text{where } (x,t,\theta) \text{ in } (\mathbb{R},[0,1],\mathbb{R}^{d_\theta}), \tag{5.29}$$

with the deterministic terminal condition

$$u(x,T,\theta) = g(x), \quad \text{and} \quad g(x) := 10 \cdot e^{\frac{x^2}{2\cdot 0.1^2}}.$$

The random coefficient $\kappa(\theta)$ independent of spatial domain x is represented by:

$$\kappa(\theta) := \kappa(\theta_1,\theta_2,\ldots,\theta_{d_\theta}) = \sum_{i=1}^{d_\theta} \frac{\theta_i^2}{2}, \tag{5.30}$$

where we show the result of the case as $d_\theta = 50$ and we sample θ from the high-dimensional normal distribution $\mathcal{N}(0, I_{d_\theta \times d_\theta})$. With the Fourier transform, the analytical solution to this random PDE system (5.29) is given by

$$u(x,t,\theta) = \int_{-\infty}^{+\infty} \Phi(x-y,t,\theta)\cdot g(y)dy, \tag{5.31}$$

where $\Phi(x,t,\theta)$ is given by

$$\Phi(x,t,\theta) := \Phi(x,t,\theta_1,\theta_2,\ldots,\theta_{d_\theta}) = \frac{1}{2\cdot\sqrt{\pi\cdot\sum\theta_i^2\cdot(1-t)}}\cdot e^{\frac{x^2}{2\cdot\sqrt{\sum\theta_i^2\cdot(1-t)}}}.$$

Note that in practice, we use $\hat{u}_a(x,t,\theta)$ as the benchmark approximation of $u(x,t,\theta)$ given by

$$\hat{u}_a(x,t,\theta) = F(\Phi(x-y,t,\theta)\cdot g(y)), \tag{5.32}$$

where F is the numerical integration from -5 to 5 with respect to y . For our experiment, we firstly study the number of iterations necessary to meet the requirement of accuracy level ϵ and Figure 5.1 suggests a potentially exponential growth of the number of iterations as ϵ decreases. Next, we aim to explore the stability of the DNN approximation. In the case, we set the required accuracy level in algorithm 3 to be 10^{-5} and after 12563 training steps (the average training time for each step is 0.237s) it gives the DNN approximation $\mathcal{R}_{\eta_1}(x,t,\theta)$. Next, we compare the results of deep learning algorithm 3 to $\hat{u}_a(x,t,\theta)$ rather than the Monte Carlo approximation to avoid the additional error. In Figure 5.2, we compare the solutions at 10 different realizations of θ^* such that $\sum(\theta_i^*)^2 = 1$ and observe that those DNN approximations overlaps under this resolution. It demonstrates the stability of approximation with respect to the random parameter θ at least for this particular problem. Also, it suggests a good agreement between the DNN and the benchmark uniformly both in spatial and time domain. Finally, it shows the potential of applying algorithm 3 to the more complicated random PDE systems. We remark that to ensure the sample θ^* satisfying that condition, we sample the first 49 θ_i and the last element θ_{d_θ} is determined by $\sqrt{1-\sum_{i=1}^{d_\theta-1}(\theta_i^*)^2}$.

Inverse problem Firstly, we notice that the end product of the above forward problem is a trained neural network (deterministic function) $\mathcal{R}_{\eta_1}(x,t,\theta)$ that maps input (state, time plus the additional parameter θ) to output. Then it is natural to raise a question: if we obtain a set of n_u data points $\{\hat{u}_a(x^{(i)},t^{(i)},\theta^*)\}_{i=1}^{n_u}$ (see definition in (5.32)) where the argument θ^* is unknown, whether or not we inversely find the unique optimal $\theta^{\diamond,n_u}$ such that

$$\theta^{\diamond,n_u} = \widehat{\arg\min}_{\theta} \frac{1}{n_u}\sum_{j=1}^{n_u}\left|\mathcal{R}\eta(x^{(j)},t^{(j)},\theta) - \hat{u}_a(x^{(j)},t^{(j)},\theta^*)\right|^2. \tag{5.33}$$

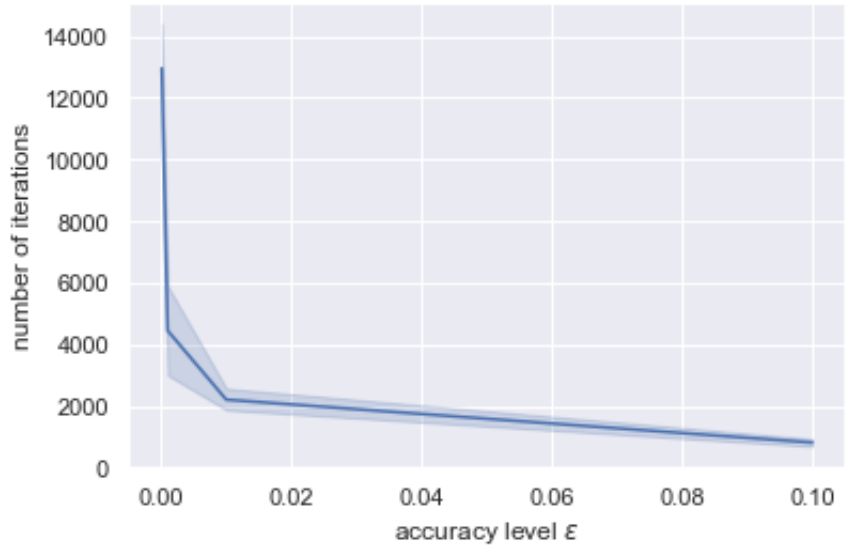


Figure 5.1: Number of optimiser iterations in terms of ϵ for example 5.4.1 and algorithm 3. The faded light-blue area is the 95%-confidence interval.

To answer this, we thus perform an artificially calibration experiment for the DNN approximation $\mathcal{R}\eta$ to the PDE (5.29) with $d_\theta = 1$ (otherwise the inversion is not unique) in the following iterative procedures after initialising θ as $0.1, 0.5, 0.8, 1.5$ and 2 respectively and setting a common required accuracy level as 10^{-3}:

- At n-th step, we generate x and t simultaneously according to standard normal and uniformly distribution on $[0, 1]$ respectively, i.e. $\{x_n^{(i)}, t_n^{(i)}\}_{i=1}^{n_u}$.
- Let θ^* be 1 and collect the data that contains n_u points by computing $\hat{u}_a$ at those sampling points $\{x_n^{(i)}, t_n^{(i)}\}_{i=1}^{n_u}$, i.e. $\{\hat{u}_a(x_n^{(i)}, t_n^{(i)}, \theta^*)\}_{i=1}^{n_u}$,
- We update θ_n for the minimisation problem (5.33) by descent method with adaptive learning rate (η_n) in the form of

$$\theta_{n+1} = \theta_n - \frac{2\eta_n}{n_u} \sum_{j=1}^{n_u} (\mathcal{R}\eta(x^{(j)}, t^{(j)}, \theta) - u(x^{(j)}, t^{(j)}, \theta^*)) \cdot \frac{\partial \mathcal{R}\eta}{\partial \theta}(x^{(j)}, t^{(j)}, \theta)$$

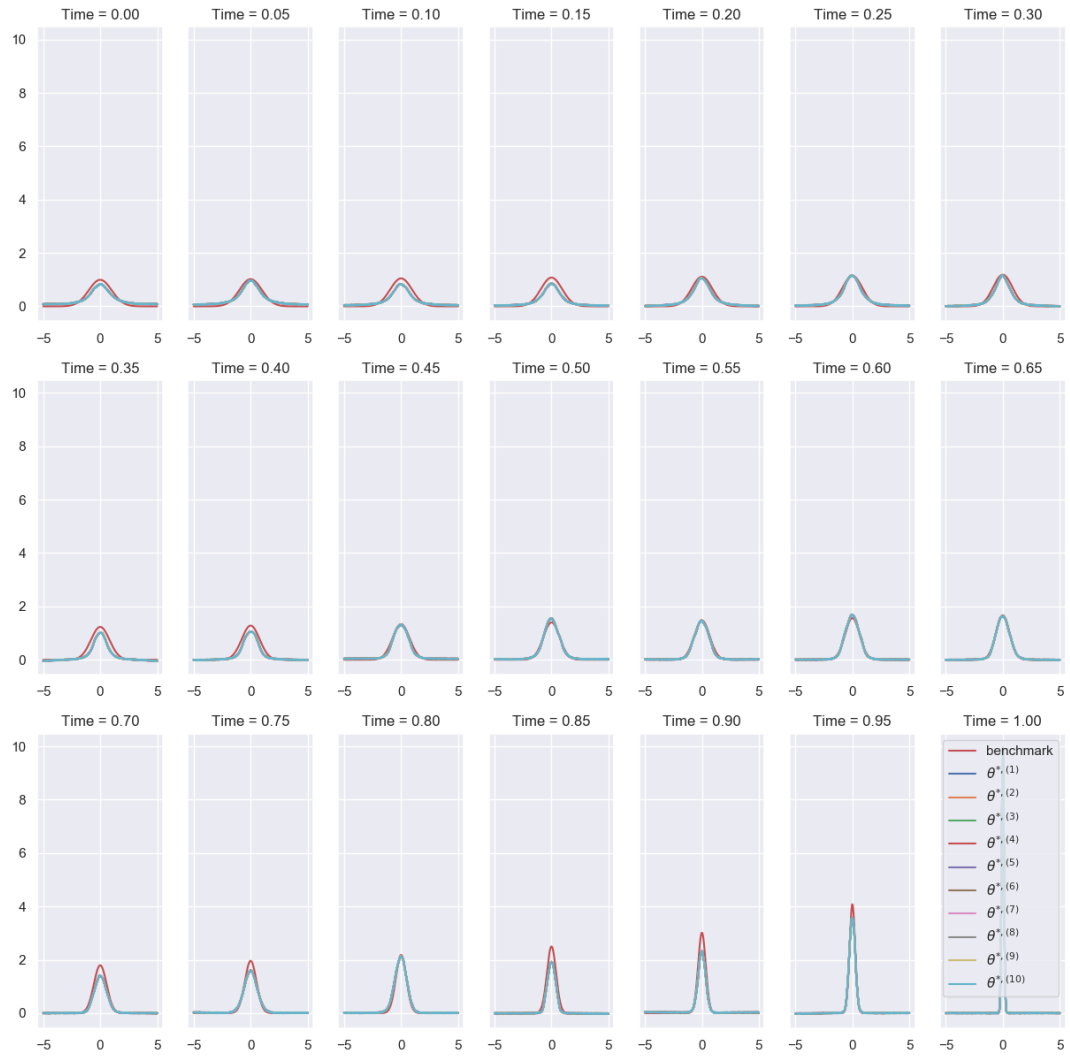


Figure 5.2: We compare a DNN with 10 different instances of θ^* and the corresponding approximated analytical solution $\hat{u}_a(x,t,\theta^*)$ (red curve) as those of θ^* satisfying $\sqrt{\sum(\theta_i^*)^2} = 1$ at 21 time points from 0 to 1. Note that in the last subfigure we compare the DNN to the exact terminal function g. The diffusion coefficient is represented by the equation (5.30). This is for example 5.4.1 and algorithm 3.

- We then check whether it is hitting the threshold and if not, update the ordinal number n by $n+1$ and repeat the above three steps.

The results are $\theta^{\diamond,n_u} = 0.988$ (after 17862 iterations), 0.991 (after 13523 iterations), 1.001 (after 11598 iterations), 1.004 (after 14726 iterations), 1.007 (after 20814 iterations) respectively and all are close to the true value of θ^*, i.e. 1. Also, this inversion is not time-consuming since the neural network is considered as a deterministic function once we obtain it after the training session for the corresponding forward problem. We remark that this type of minimisation problem (5.33) in one-dimension can also simply be solved by Newton's method.

Moreover, we add additional Gaussian noise $\mathcal{N}^{(i)}(0,\sigma^2)$ to each $\hat{u}_a(x^{(i)},t^{(i)},\theta^*)$ for any $i \in \{1,\dots,n_u\}$ and denote them as $\{\tilde{u}_a(x^{(i)},t^{(i)},\theta^*)\}_{i=1}^{n_u}$ where $\tilde{u}_a(x^{(i)},t^{(i)},\theta^*) = \hat{u}_a(x^{(i)},t^{(i)},\theta^*) + \mathcal{N}^{(i)}(0,\sigma^2)$. We are now in a situation to ask whether we can learn the true θ^* from the noisy data set. We follow the same procedures described above as the original data is replaced by two corresponding noisy with different σ (0.1 and 0.5). In the left subfigure of Figure 5.3, it shows that the approximation for θ^* would instead jump around the point estimate of the case without noise if we start from a higher learning rate. As σ increases, the fluctuation of the approximation between two consecutive optimisation steps renders much bigger due to the larger noise. However, in the right subfigure of Figure 5.3, we see that they follow the same trend regardless of the noise if we substantially reduce the learning rate from 0.2 to $4 \cdot 10^{-4}$. It suggests if the data has noise, we need to be much careful about choosing the learning rate for solving the optimisation problem, and that it is still robust due to this type of Gaussian noise with $\sigma \leq 0.5$.

Alternatively, assume θ has uniform prior $p(\theta)$ and the likelihood to be Gaussian. Now we aim to learn the Gaussian posterior $\mathcal{N}(\mu_\theta,\sigma_\theta^2)$ for θ rather than a point estimate. Combing Bayes rules, uniform prior $p(\theta)$ and the conditional

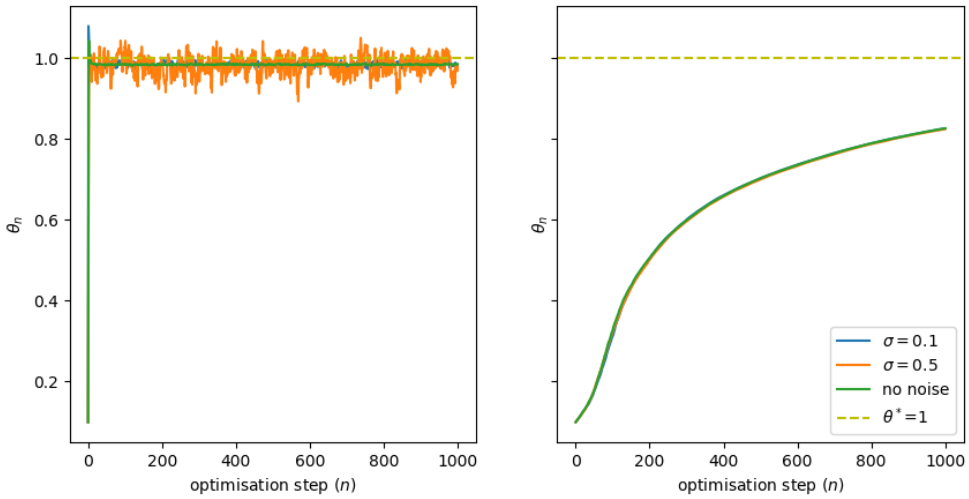


Figure 5.3: Path of θ_n with respect to different σ and learning rates: the left subfigure corresponds to the learning rate starting from 0.2 and right from $4 \cdot 10^{-4}$.

distribution of $\tilde{u}_a$, we know

$$p(\theta|\tilde{u}_a, x^{(i)}, t^{(i)}, \sigma) \propto p(\tilde{u}_a|\theta, x^{(i)}, t^{(i)}, \sigma)p(\theta) = \mathcal{N}(\tilde{u}_a(x^{(i)}, t^{(i)}, \theta^*)|\mathcal{R}\eta(x^{(i)}, t^{(i)}, \theta), \sigma^2)p(\theta)$$
$$= \exp\left(-\frac{|\mathcal{R}\eta(x^{(i)}, t^{(i)}, \theta) - \tilde{u}_a(x^{(i)}, t^{(i)}, \theta^*)|^2}{2\sigma^2}\right), \quad \forall i \in \{1, \dots, n_u\}$$

Applying Laplace approximation, it results in the optimisation problem for the parameter μ_θ in the same form as (5.33) and then

$$\sigma_\theta^2 = \sigma^2[\frac{\partial^2 \mathcal{R}\eta}{\partial x^2}(\mu_\theta)\sum_{i=1}^{n_u}(\mathcal{R}\eta(x^{(i)}, t^{(i)}, \theta) - \tilde{u}_a(x^{(i)}, t^{(i)}, \theta^*)) - n_u(\frac{\partial \mathcal{R}\eta}{\partial x}(\mu_\theta))^2]^{-1}. \tag{5.34}$$

Therefore, our previous point estimates can be used as the estimate for μ_θ and then plug it into (5.34) to obtain the value of variance, i.e. σ_θ^2. Note that changing the distribution for the prior would lead to a optimisation problem different from the form of (5.33).

5.4.2 Dirichlet problem with the random coefficient and boundary condition

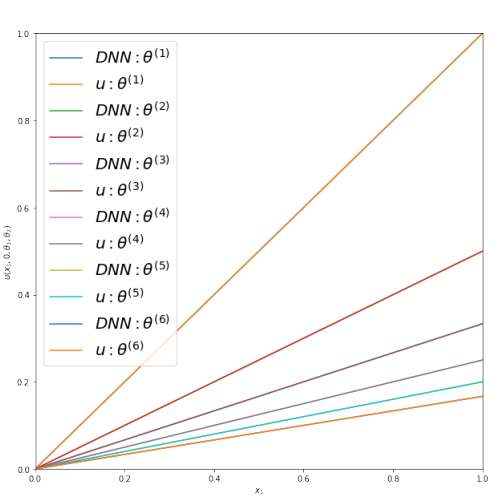
In this example, we consider a simple elliptic PDE system in two-dimensional spatial domain with random diffusion $\kappa(\theta)$

$$\kappa(\theta)\left(\frac{\partial^2 u}{\partial x_1^2}(x_1, x_2, \theta) + \frac{\partial^2 u}{\partial x_2^2}(x_1, x_2, \theta)\right) = 0 \qquad \text{in} \qquad (\mathcal{D}, [0,1]^{d_\theta}), \tag{5.35}$$

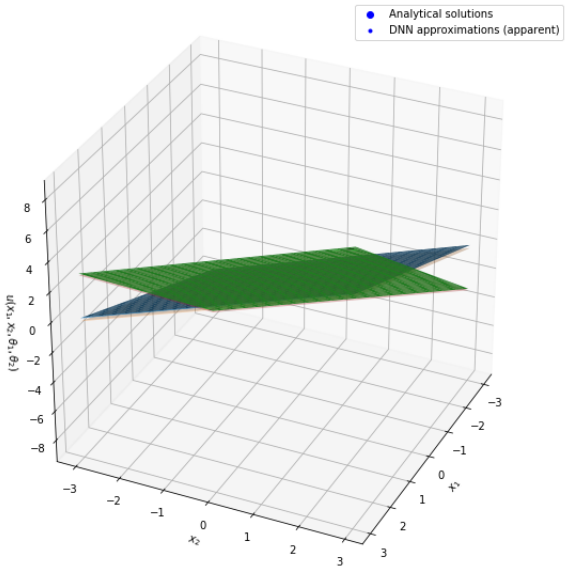
with the random boundary condition

$$u(x_1, x_2, \theta) := u(x_1, x_2, \theta_1, \ldots, \theta_{d_\theta}) = \theta \cdot x \quad \text{if } (x_1, x_2) \in \partial\mathcal{D}$$

where $\mathcal{D} = (0,1) \times (0,1)$ and $\partial\mathcal{D}$ is the boundary of $\mathcal{D}$. We assume the random



(a) 2D comparison: 6 choices of θ as fixing $x_2 = 0$.



(b) 3D comparison: 2 choices of θ

Figure 5.4: A comparison of DNN and the analytical solution $u(x_1, x_2, \theta_1, \theta_2)$ for example 5.4.2 and algorithm 4.

function $\kappa(\theta)$ have the same form as (5.30) and show the result of the case as $d_\theta = 2$ with sampling θ from the high-dimensional uniform distribution $\mathcal{U}[0,1]^{d_\theta}$.

By analysis, we find the unique analytical solution is give by

$$u(x_1, x_2, \theta) = \theta_1 \cdot x_1 + \theta_2 \cdot x_2, \quad (x_1, x_2) \in \mathcal{D} \cup \partial\mathcal{D}.$$

For our experiment, we again set the required accuracy level in algorithm 4 to be 10^{-5} and after 10384 training steps (the average training time for each step is 0.416s), the DNN approximation $\mathcal{R}_{\eta_1}(x_1, x_2, \theta)$ of algorithm 4 is determined. Next we visualise our DNN results together with the benchmark both in 2D and 3D. Figure 5.4 then confirms the validity of representation of deep network of algorithm 4 for the solution to (5.35).

5.4.3 Dirichlet problem with the random coefficient and lower order term

In the third example, we consider a more complicated boundary value problem for $u : \mathbb{R}^2 \to \mathbb{R}$ in two-dimensional spatial domain with random diffusion $\kappa(\theta)$

$$\kappa(\theta)\left(\frac{\partial^2 u}{\partial x_1^2}(x_1, x_2, \theta) + \frac{\partial^2 u}{\partial x_2^2}(x_1, x_2, \theta)\right) = -2 \qquad \text{in} \qquad (\mathcal{D}, [0,1]^{d_\theta}), \tag{5.36}$$

with the deterministic zero-valued boundary condition, i.e.

$$u(x_1, x_2, \theta) = 0 \quad \text{if } (x_1, x_2) \in \partial\mathcal{D},$$

where $\mathcal{D} = (-1, 1) \times (-1, 1)$ and $\partial\mathcal{D}$ is the boundary of $\mathcal{D}$. Assuming the random function $\kappa(\theta)$ have the same form as (5.30), we present the result of the case as $d_\theta = 50$ with θ sampled from the high-dimensional uniform distribution $\mathcal{U}[0, 1]^{d_\theta}$. To obtain our DNN approximation $\mathcal{R}_{\eta_1}$, the required accuracy level to stop in algorithm 4 is 10^{-5}. Following the reasoning from [15], we consider $\cos(\pi x_1(2k_1+1)/2)\sin(\pi x_2(2k_2+1)/2)$ as eigenfunctions and expand the solution

u in those. Then we find

$$u(0,0,\theta) = \frac{1}{k(\theta)} \cdot \frac{128}{\pi^4} \sum_{k_1=0,k_2=0}^{\infty} \frac{(-1)^{k_1+k_2}}{(2k_1+1)(2k_2+1)((2k_1+1)^2+(2k_2+1)^2)}$$
$$\approx \hat{u}_a(\theta) := \frac{1}{k(\theta)} \cdot 0.5893708\ldots$$

Using this particular expression, we measure the error between the DNN approximation $\mathbb{R}_{\eta_1}$ of algorithm 4 at k-th training step over $n_\theta(=100)$ realisations of θ by

$$Error(k) = \frac{1}{n_\theta} \sum_{j=1}^{n_\theta} |\hat{u}_a(\theta^{(j)}) - \mathcal{R}_{\eta_1}^{(k)}(0,0,\theta^{(j)})|, \tag{5.37}$$

which we expects to decrease in the ordinal k and where $\mathcal{R}^{(k)}$ means the DNN approximation obtained after k-th training step. However, behaviour of error at a particular spatial point is not informative because the DNN approximation $\mathcal{R}_{\eta_1}$ learns the general solution in the whole spatial domain. Then we consider the average of $n_\theta(=100) \times n_x(=100)$ errors in the spatial domain $\mathcal{D}$ given by

$$AError(k) = \frac{1}{n_\theta} \sum_{j=1}^{n_\theta} \frac{1}{n_x} \sum_{i=1}^{n_x} |\overline{u}(x_1^{(i)}, x_2^{(i)}, \theta^{(j)}) - \mathcal{R}_{\eta_1}^{(k)}(x_1^{(i)}, x_2^{(i)}.\theta^{(j)})|, \tag{5.38}$$

Note that in (5.38), each $x^{(i)}$ or $\theta^{(j)}$ is sampled uniformly from $\mathcal{D}$ or $[0,1]^{d_\theta}$ respectively and $\overline{u}$ is obtained by application of the standard deterministic PDE library. We obtain the final DNN approximation $\mathcal{R}_{\eta_1}$ around after 12000 training steps while in Figure 5.5 (the average training times for each step are 0.451 and 0.459 respectively) we present behaviours only in the first 6000 steps which have much bigger fluctuation in magnitude compared to the last. Figure 5.5 shows the error at this particular spatial point behave similarly but with larger confidence intervals as the mean-error over the whole domain which consists of spatial and random parameter. Moreover, we define the squared error of i-th

Figure 5.5: Behaviour of errors (5.37) (orange) and (5.38) (blue). The faded areas corresponds to their confidence intervals. This is for example 5.4.3 and algorithm 4.

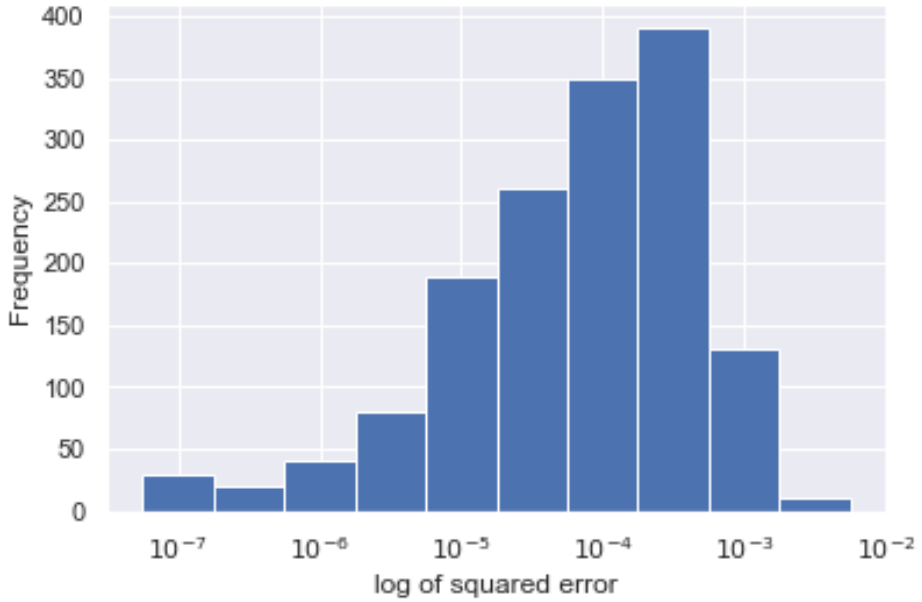


Figure 5.6: Histogram of squared-errors (see definition in (5.39)) on the test data set $\mathcal{S}$. This is for example 5.4.3 and algorithm 4.

instance $(x^{(i)}, \theta^{(i)}) \in \mathcal{S} := \{(x^{(i)}, \theta^{(i)}) : i = 1, 2, \ldots, 1500\}$ as

$$Error(x^{(i)}, \theta^{(i)}) = |\overline{u}(x_1^{(i)}, x_2^{(i)}, \theta^{(i)}) - \mathcal{R}_{\eta_1}(x_1^{(i)}, x_2^{(i)}, \theta^{(i)})|^2, \tag{5.39}$$

where each instance in $\mathcal{S}$ is uniformly drawn from the domain. In Figure 5.6, it displays the error varies significantly and spans from almost 10^{-7} to 10^{-2} though the average is around $4 * 10^{-5}$.

5.4.4 Poisson's equation with the random coefficient

To investigate further, we consider the following example in two-dimensional spatial domain with random diffusion $\kappa(\theta)$

$$\kappa(\theta)\left(\frac{\partial^2 u}{\partial x_1^2}(x_1, x_2, \theta) + \frac{\partial^2 u}{\partial x_2^2}(x_1, x_2, \theta)\right) - u(x_1, x_2, \theta) = f(x_1, x_2) \qquad \text{in} \qquad (\mathbb{R}^2, [0,1]^{d_\theta}), \tag{5.40}$$

where the random function $\kappa(\theta)$ is represented by the equation (5.30). Recalling the derivation of recursive relationship for type (5.5), we take the expectation to both sides of (5.23) and let $t \to \infty$. Because of the lower bound $c(x, \theta) = 1 \geq 1 > 0$, the dominated convergence theorem together with the approximation $X_t^{\theta,\pi}$ with initial state $X_0^{\theta,\pi} = x := (x_1, x_2)$ gives that

$$u(x, \theta) = \mathbb{E}[\int_0^\infty e^{-\int_0^s c(x,\theta)d\gamma} f(X_s^{x,\theta})ds] = \mathbb{E}[\int_0^\infty e^{-s} f(X_s^{x,\theta})ds] \approx \mathbb{E}[\int_0^\infty e^{-s} f(X_s^{\theta,\pi})ds].$$

With truncation at $t = 50$ and numerical integration, we obtain our Monte-Carlo benchmark with $n_{MC}(= 10^6)$ X samples given by

$$\hat{u}_{MC}(x_1, x_2, \theta) = \frac{1}{n_{MC}} \sum_{i=1}^{n_{MC}} F(e^{-s} f(X_s^{i,\theta,\pi}))$$

where F denotes the numerical integration method with respect to s from 0 to 50. Next, we experiment several cases according to the different choices of $f(x_1, x_2)$ including the quadratic function $(x_1+x_2)^2$ and the non-linear function $\sin(x_1+x_2)$ and $\cos(x_1 + x_2)$ and the constant function 1. Before running our algorithm 5, we set the required accuracy level to be 10^{-5} and choose a set of θ^* such that $\kappa(\theta^*) = 1$. Then for all cases, we track the average of errors corresponding to $n_\theta(= 100) \times n_x(= 100)$ samples (x, θ) in terms of training steps by

$$AError(k) = \frac{1}{n_\theta}\sum_{j=1}^{n_\theta}\frac{1}{n_x}\sum_{i=1}^{n_x}|\hat{u}_{MC}(x_1^{(i)}, x_2^{(i)}, \theta^{(j)}) - \mathcal{R}_{\eta_1}^{(k)}(x_1^{(i)}, x_2^{(i)}, \theta^{(j)})| \tag{5.41}$$

where k denotes the k-th training step and $\mathcal{R}^{(k)}$ means the DNN approximation obtained after k-th training step. Next, we obtain the target DNN approximations $\mathcal{R}_{\eta_1}(x, \theta)$ for the four cases after 14453 (*quadratic*), 13875 (*sin*), 13348 (cos) and 10451 (*constant*) respectively and the average training times for each step are 0.613s, 0.544s, 0.547s, 0.409s respectively.

Figure 5.7 accords with the naivety that learning the DNN approximation determined by constant is much easier and stable than other cases.

5.4.5 Surface flow problem with the random coefficient

In the last example, we consider a probabilistic uncertainty quantification problem in surface flow and the governing random PDE system is given by the following second-order elliptic PDE (ODE) in one-dimensional spatial domain

$$\begin{aligned} &\nabla \cdot (\kappa(x, \theta)\nabla u(x, \theta)) = 0 \qquad \text{in} \qquad [0, 1] \times \mathbb{R}^{d_\theta}, \\ &u(0, \theta) = 1, \quad u(1, \theta) = 0, \quad \forall \theta \in \mathbb{R}^{d_\theta} \end{aligned} \tag{5.42}$$

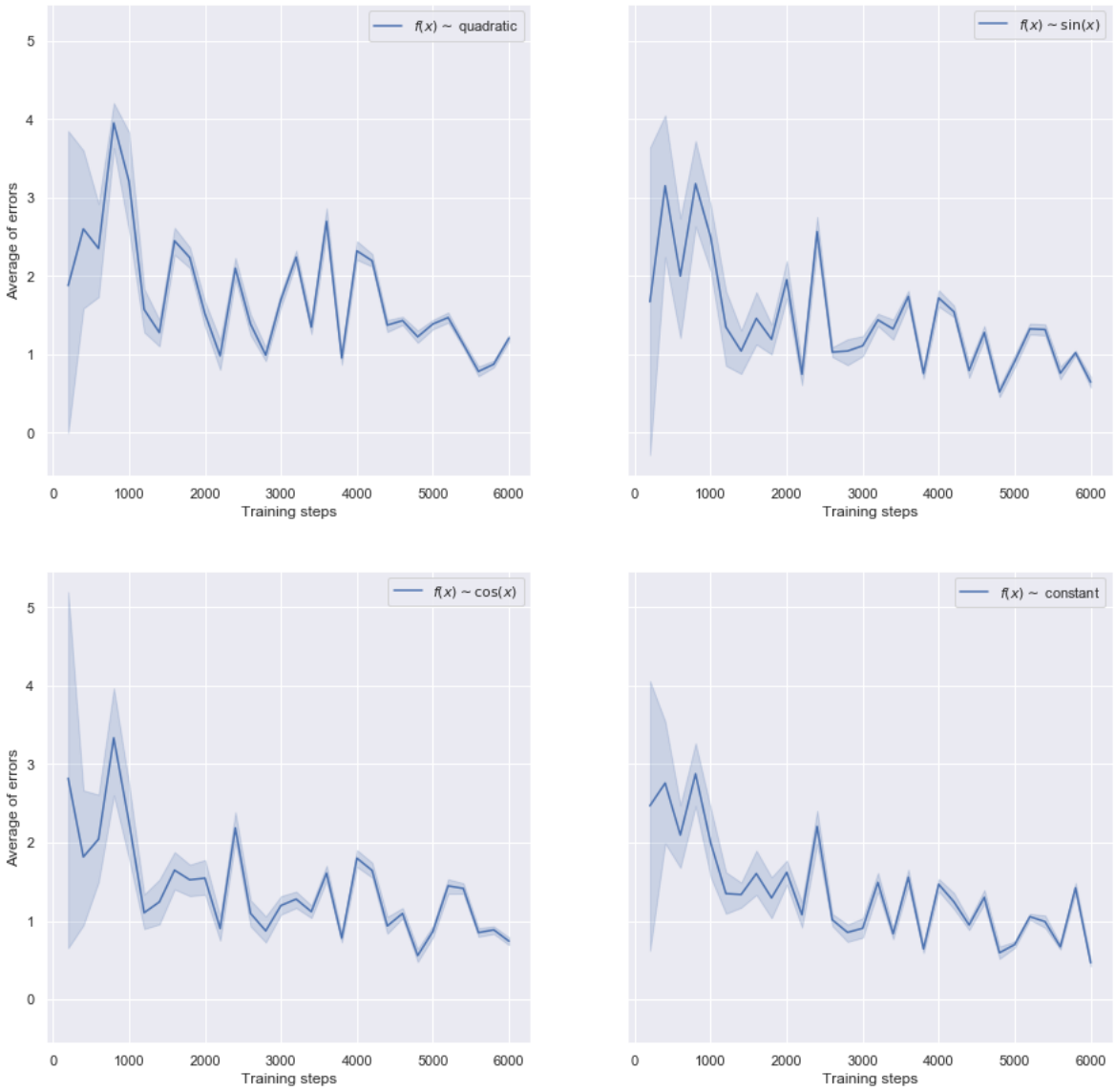


Figure 5.7: Trend of errors (see definition in (5.4.4)) with respect to four different choices of function f in (5.40): the upper left corner is for the case $f(x_1, x_2) = (x_1 + x_2)^2$ and the upper right is for the case $f(x_1, x_2) = \sin(x_1 + x_2)$; the lower left is for that $f(x_1, x_2) = \cos(x_1 + x_2)$ and the lower right is that $f(x_1, x_2) = 1.$. The diffusion coefficient is represented by the equation (5.30). This is for example 5.4.4 and algorithm 5.

where the random coefficient $\kappa(\cdot,\cdot)$ is dependent not only on θ but the spatial variable x and in the form of $\kappa(x,\theta) = e^{Z(x,\theta)}$. This Z can be expanded by a countable set of uncorrelated, zero mean random variables $\{\theta_i\}_{i\in\mathbb{N}}$ such that

$$Z(x,\theta) = \mathbb{E}[Z(x,\theta)] + \sum_{i=1}^{\infty} \sqrt{w_i}\theta_i b_i(x), \tag{5.43}$$

where $\{w_i\}_{i\in\mathbb{N}}$ are the eigenvalues and $\{b_i(x)\}_{i\in\mathbb{N}}$ the normalised eigenfunctions of the covariance operator with kernel function

$$C(x,y) = \exp(\frac{|x-y|}{\lambda}). \tag{5.44}$$

Particularly, we choose $\mathbb{E}[Z(x,\theta)] = 0$ in (5.45). To be implementable for $Z(x,\theta)$, we truncate it in the form of :

$$Z_N(x,\theta) = \sum_{i=1}^{N} \sqrt{w_i}\theta_i b_i(x) \quad (\to Z(x,\theta) \text{ as } N \text{ tend to } \infty)$$

where N denotes the KL-expansion of Z is truncated at which term. From [20], we find the analytical expressions for the eigenpairs of the covariance operator given by

$$\begin{aligned} w_i &= \frac{2\lambda}{\lambda^2 v_i^2 + 1}, \quad i \in \mathbb{N} \\ b_i(x) &= \frac{1}{||\sin(v_i x) + \lambda v_i \cos(v_i x)||_{L_2(0,1)}}(\sin(v_i x) + \lambda v_i \cos(v_i x)), \end{aligned} \tag{5.45}$$

where $\{v_i\}_{i\in\mathbb{N}}$ are the set of solutions of the transcendental equation

$$\tan(v) = \frac{2\lambda v}{\lambda^2 v^2 - 1}. \tag{5.46}$$

Before simulation, we set $\lambda = 0.3$ in (5.44) and $N = 20$ in (5.46) and choose the threshold level for accuracy to be 10^{-4} in algorithm 6. After 7339 training steps (the average training time for each step is 0.492s), we obtain the desired

DNN approximation $\mathcal{R}_{\eta_1}(x,\theta)$. To investigate, we focus expected values of the solution at several specified spatial points, i.e. $\mathbb{E}[u(0,\cdot)]$, $\mathbb{E}[u(0.25,\cdot)]$, $\mathbb{E}[u(0.5,\cdot)]$ and $\mathbb{E}[u(1,\cdot)]$. Given the boundary condition of this random PDE system (5.42), we know $\mathbb{E}[u(1,\cdot)] = 1$ and $\mathbb{E}[u(1,\cdot)] = 0$. Furthermore, we use the Monte Carlo solutions under the constraint of the accuracy level at 10^{-4} as benchmark for the other two. With $n_1 = 5.12 \cdot 10^5$ and $n_2 = 4.93 \cdot 10^5$, $\mathbb{E}[u(0.25,\cdot)] \approx \frac{1}{n_1}\sum_{i=1}^{n_1} \hat{u}(0.25,\theta^{(i)}) = 0.74098$ and $\mathbb{E}[u(0.5,\cdot)] \approx \frac{1}{n_2}\sum_{i=1}^{n_2} \hat{u}(0.5,\theta^{(i)}) = 0.50009$. Note that given $\theta^{(i)}$, each $\hat{u}(\cdot,\theta^{(i)})$ is solved numerically by the standard ODE library. Then we track the error during the training session in terms of training steps by

$$Error(x,k) = |\hat{\mathbb{E}}[u(x,\cdot)] - \frac{1}{n}\sum_{i=1}^{n=10000} \mathcal{R}_{\eta_1}^{(k)}(x,\theta^{(i)})|, \tag{5.47}$$

where $\hat{\mathbb{E}}$ means exactly the expectation if $\mathbb{E}[u(x,\cdot)]$ can be computed analytically otherwise Monte-Carlo average and $\mathcal{R}^{(k)}$ means the DNN approximation obtained after k-th training step. Figure 5.8 shows in the first few training steps, the

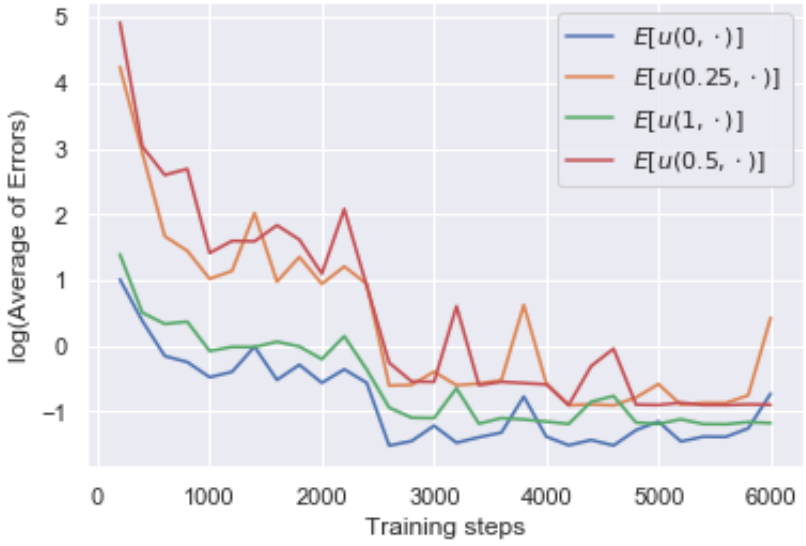


Figure 5.8: Behaviour of errors (5.47) at four specified spatial points.

DNN approximation $\mathcal{R}$ shows the biggest uncertainty (error) in the middle of the

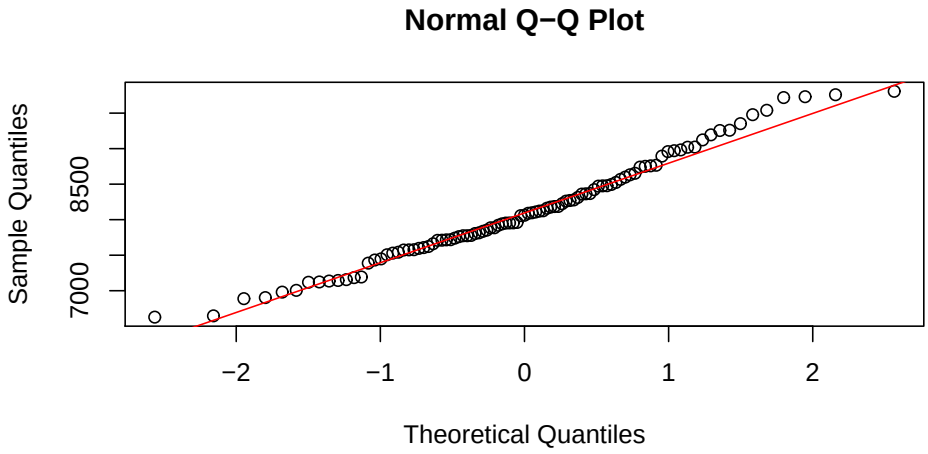
spatial domain but with the potential fastest rate of convergence while the DNN estimates at the two boundary points present a similar and much less unstable trend compared to the other two.

Moreover, we compare the efficiency between algorithm 6 and algorithm 7 in the following procedures. First, we set the same accuracy level requirement, i.e. $threshold = 10^{-4}$ for both two algorithms (algorithm 6 and 7) as the stopping criteria. Second, we choose 100 random seeds to obtain the target data. Note that given a random seed, we record two stopping times τ_1 and τ_2 which are the ordinal number of training step where the algorithm stops. With the limited page, we show the first five out of 100 in Table 5.1.

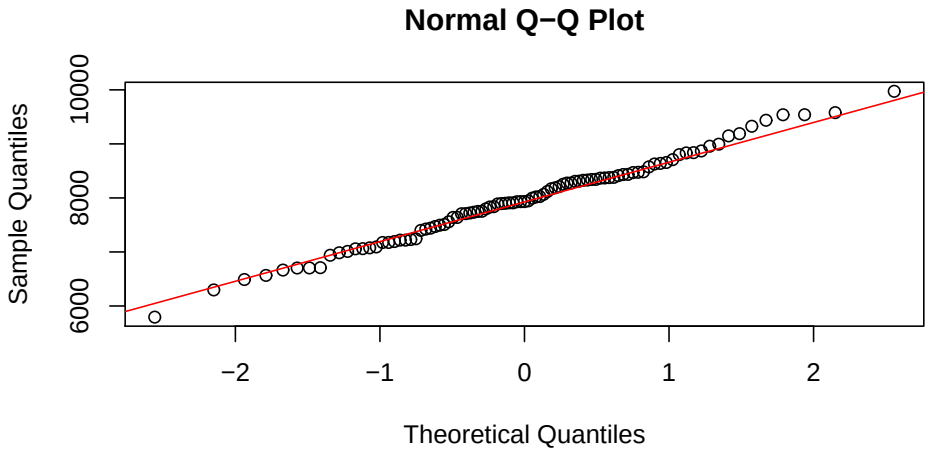
τ_1 for Algorithm 6	τ_2 for Algorithm 7
7339	7144
7589	7954
7240	8161
8169	8225
7244	7156
...	...

Table 5.1: Stopping times

Figure 5.9 suggests the potential normality of the stopping times of our algorithms and it confirms the validity to perform two-sample t-test further for the collected data. Finally, we obtain the p-value= 0.09003 in the Welch two sample t-test which has the hypothesis that the true difference $(\tau_2 - \tau_1)$ in means is equal to 400. This means we cannot reject the hypothesis at the 5%-significance level and it demonstrate the mean of stopping times for the softly-direct (algorithm 6) would be significantly less than the strongly-direct (algorithm 7). Thus it



(a) normality of data for algorithm 6



(b) normality of data for algorithm 7

Figure 5.9: Normality of stopping times

is unwise to apply hard-assignment blindly since we find a situation where the soft-assignment would not perform less efficiently than the hard.

5.5 Conclusions

We propose a novel method for solving high-dimensional random PDEs using a deep network that is based on the probabilistic representation compared to those in [53] based on the non-probabilistic. They are mesh-free and thus might avoid the potential challenge induced from the curse of dimensionality. Also, the DNN solution is a representation for the general solution including all arguments, i.e., spatial, temporal and random parameters, and therefore can be conveniently used to produce samples for the specific uncertainty quantification problem and solve the corresponding inverse problems. But a disadvantage is that the essential backward relationship is highly sensitive to the structure of the PDE system, and this is the reason why we propose different learning methods (algorithm 3 to 5) for different types of random PDE systems. On the one hand, the aim of this chapter is to show that DNNs based on the probabilistic can be used to solve several types of random PDEs numerically with a desired rate of convergence. Moreover, DNNs based on the non-probabilistic with using the weak-enforcement of the boundary condition would perform even better than the strong in a certain circumstance. For the further investigation, we can explore the validity of applying our new proposed DNNs on more complicated random PDE systems especially handling those of high-dimensional with boundary conditions.

Bibliography

[1] Ali, A.L.H. (2012). *Pedestrian Flow in the Mean Field Limit*. Ph.D. thesis, King Abdullah University of Science and Technology (KAUST).

[2] Antonelli, F. and Kohatsu-Higa, A. (2002). Rate of convergence of a particle method to the solution of the McKean–Vlasov equation. *The Annals of Applied Probability*, **12**, 423–476.

[3] Babuska, I., Tempone, R. and Zouraris, G.E. (2004). Galerkin finite element approximations of stochastic elliptic partial differential equations. *SIAM Journal on Numerical Analysis*, **42**, 800–825.

[4] Babuška, I., Nobile, F. and Tempone, R. (2007). A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, **45**, 1005–1034.

[5] Barth, A., Schwab, C. and Zollinger, N. (2011). Multilevel Monte Carlo finite element method for elliptic PDEs with stochastic coefficients. *Numerische Mathematik*, **119**, 123–161.

[6] Belomestny, D. and Reiss, M. (2005). Implementing the spectral calibration of exponential Lévy models.

[7] Belomestny, D. and Schoenmakers, J. (2018). Projected particle methods for

solving McKean–Vlasov stochastic differential equations. *SIAM Journal on Numerical Analysis*, **56**, 3169–3195.

[8] Blatman, G. and Sudret, B. (2010). An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis. *Probabilistic Engineering Mechanics*, **25**, 183–197.

[9] Bossy, M. (2004). Optimal rate of convergence of a stochastic particle method to solutions of 1D viscous scalar conservation laws. *Mathematics of computation*, **73**, 777–812.

[10] Bossy, M. and Jourdain, B. (2002). Rate of convergeance of a particle method for the solution of a 1D viscous scalar conservation law in a bounded interval. *The Annals of Probability*, **30**, 1797–1832.

[11] Bossy, M. and Talay, D. (1996). Convergence rate for the approximation of the limit law of weakly interacting particles: application to the Burgers equation. *The Annals of Applied Probability*, **6**, 818–861.

[12] Bossy, M. and Talay, D. (1997). A stochastic particle method for the McKean-Vlasov and the Burgers equation. *Mathematics of Computation of the American Mathematical Society*, **66**, 157–192.

[13] Bossy, M., Fezoui, L. and Piperno, S. (1997). Comparison of a stochastic particle method and a finite volume deterministic method applied to Burgers equation. *Monte Carlo Methods and Applications*, **3**, 113–140.

[14] Buchmann, F.M. (2005). Simulation of stopped diffusions. *Journal of Computational Physics*, **202**, 446–462.

[15] Buchmann, F.M. and Petersen, W.P. (2006). An exit probability approach to solving high dimensional Dirichlet problems. *SIAM Journal on Scientific Computing*, **28**, 1153–1166.

[16] Buckdahn, R., Li, J., Peng, S. and Rainer, C. (2017). Mean-field stochastic differential equations and associated PDEs. *The Annals of Probability*, **45**, 824–878.

[17] Bujok, K., Hambly, B. and Reisinger, C. (2013). Multilevel simulation of functionals of Bernoulli random variables with application to basket credit derivatives. *Methodology and Computing in Applied Probability*, 1–26.

[18] Carmona, R., Delarue, F. and Lachapelle, A. (2013). Control of McKean–Vlasov dynamics versus mean field games. *Mathematics and Financial Economics*, **7**, 131–166.

[19] Chassagneux, J.F., Crisan, D. and Delarue, F. (2014). A probabilistic approach to classical solutions of the master equation for large population equilibria. *arXiv preprint arXiv:1411.3009*.

[20] Cliffe, K.A., Giles, M.B., Scheichl, R. and Teckentrup, A.L. (2011). Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients. *Computing and Visualization in Science*, **14**, 3.

[21] Delarue, F., Inglis, J., Rubenthaler, S. and Tanré, E. (2015). Global solvability of a networked integrate-and-fire model of Mckean–Vlasov type. *The Annals of Applied Probability*, **25**, 2096–2133.

[22] Delarue, F., Inglis, J., Rubenthaler, S. and Tanré, E. (2015). Particle systems with a singular mean-field self-excitation. Application to neuronal networks. *Stochastic Processes and their Applications*, **125**, 2451–2492.

[23] Deodatis, G. (1991). Weighted integral method. i: stochastic stiffness matrix. *Journal of Engineering Mechanics*, **117**, 1851–1864.

[24] Doostan, A. and Owhadi, H. (2011). A non-adapted sparse approximation of PDEs with stochastic inputs. *Journal of Computational Physics*, **230**, 3015–3034.

[25] E, W., Hutzenthaler, M., Jentzen, A. and Kruse, T. (2016). On full history recursive multilevel Picard approximations and numerical approximations of high-dimensional nonlinear parabolic partial differential equations. *arXiv preprint arXiv:1607.03295*.

[26] Fishman, G.S. (1996). *Monte Carlo: Concepts, Algorithms and Applications*. Springer Verlag, New York, NY, USA.

[27] Friedman, A. (2006). *Stochastic differential equations and applications*. Courier Corporation.

[28] Ghanem, R.G. and Spanos, P.D. (2003). *Stochastic finite elements: a spectral approach*. Courier Corporation.

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

[30] Giles, M.B., Nagapetyan, T. and Ritter, K. (2015). Multilevel Monte Carlo approximation of distribution functions and densities. *SIAM/ASA Journal on Uncertainty Quantification*, **3**, 267–295.

[31] Haji-Ali, A.L. and Tempone, R. (2016). Multilevel and Multi-index Monte Carlo methods for Mckean-Vlasov equations. *arXiv preprint arXiv:1610.09934*.

[32] Han, J., Jentzen, A. and Weinan, E. (2017). Overcoming the curse of dimensionality: Solving high-dimensional partial differential equations using deep learning. *arXiv preprint arXiv:1707.02568*, 1–13.

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

[34] Hesthaven, J.S., Rozza, G., Stamm, B. *et al.* (2016). *Certified reduced basis methods for parametrized partial differential equations*. Springer.

[35] Jacquier, A.J. and Oumgari, M. (2019). Deep PPDEs for rough local stochastic volatility. *Available at SSRN 3400035*.

[36] Jakeman, J.D., Perego, M. and Severa, W.M. (2018). Neural networks as surrogates of nonlinear high-dimensional parameter-to-prediction maps. Tech. rep., Sandia National Lab.(SNL-NM), Albuquerque, NM (United States).

[37] Karatzas, I. and Shreve, S. (2012). *Brownian motion and stochastic calculus*, vol. 113. Springer Science & Business Media.

[38] Kebaier, A. (2005). Statistical Romberg extrapolation: a new variance reduction method and applications to option pricing. *The Annals of Applied Probability*, **15**, 2681–2705.

[39] Kingma, D.P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[40] Krylov, N.V. (1980). *Controlled diffusion processes*, vol. 14 of *Applications of Mathematics*. Springer-Verlag, New York-Berlin, translated from the Russian by A. B. Aries.

[41] Krylov, N.V. (2002). *Introduction to the theory of random processes*, vol. 43. American Mathematical Soc.

[42] Kuo, F.Y., Schwab, C. and Sloan, I.H. (2012). Quasi-Monte Carlo finite element methods for a class of elliptic partial differential equations with random coefficients. *SIAM Journal on Numerical Analysis*, **50**, 3351–3374.

[43] Lagaris, I.E., Likas, A. and Fotiadis, D.I. (1998). Artificial neural networks for solving ordinary and partial differential equations. *IEEE transactions on neural networks*, **9**, 987–1000.

[44] Lee, H. and Kang, I.S. (1990). Neural algorithm for solving differential equations. *Journal of Computational Physics*, **91**, 110–131.

[45] Lemaire, V. and Pagès, G. (2017). Multilevel Richardson–Romberg extrapolation. *Bernoulli*, **23**, 2643–2692.

[46] Lions, P. (2014). Cours au collège de france: Théorie des jeux à champs moyens.

[47] Liu, W.K., Belytschko, T. and Mani, A. (1986). Probabilistic finite elements for nonlinear structural dynamics. *Computer Methods in Applied Mechanics and Engineering*, **56**, 61–81.

[48] Malek, A. and Beidokhti, R.S. (2006). Numerical solution for high order differential equations using a hybrid neural network-optimization method. *Applied Mathematics and Computation*, **183**, 260–271.

[49] McKean Jr, H. (1966). A class of Markov processes associated with nonlinear parabolic equations. *Proceedings of the National Academy of Sciences of the United States of America*, **56**, 1907.

[50] Meade Jr, A.J. and Fernandez, A.A. (1994). The numerical solution of linear ordinary differential equations by feedforward neural networks. *Mathematical and Computer Modelling*, **19**, 1–25.

[51] Meade Jr, A.J. and Fernandez, A.A. (1994). Solution of nonlinear ordinary differential equations by feedforward neural networks. *Mathematical and Computer Modelling*, **20**, 19–44.

[52] Méléard, S. (1996). Asymptotic behaviour of some interacting particle systems; McKean-Vlasov and Boltzmann models. In *Probabilistic models for nonlinear partial differential equations*, 42–95, Springer.

[53] Nabian, M.A. and Meidani, H. (2018). A deep neural network surrogate for high-dimensional random partial differential equations. *arXiv preprint arXiv:1806.02957*.

[54] Nobile, F., Tempone, R. and Webster, C.G. (2008). An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, **46**, 2411–2442.

[55] Øksendal, B. (2003). Stochastic differential equations. In *Stochastic differential equations*, 65–84, Springer.

[56] Parthasarathy, K.R. (1967). *Probability measures on metric spaces*, vol. 352. American Mathematical Soc.

[57] Pope, S.B. (2000). *Turbulent flows*. Cambridge University Press.

[58] Quarteroni, A., Manzoni, A. and Negri, F. (2015). *Reduced basis methods for partial differential equations: an introduction*, vol. 92. Springer.

[59] Ricketson, L. (2015). A multilevel Monte Carlo method for a class of Mckean-Vlasov processes. *arXiv preprint arXiv:1508.02299*.

[60] Robbins, H. (1955). A remark on Stirling's formula. *The American Mathematical Monthly*, **62**, 26–29.

[61] Rudd, K. (2013). *Solving partial differential equations using artificial neural networks*. Ph.D. thesis, Duke University.

[62] Sabate Vidales, M., Siska, D. and Szpruch, L. (2018). Unbiased deep solvers for parametric PDEs. *arXiv e-prints*, arXiv:1810.05094.

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

[64] Suzuki, Y. (2017). Neural network-based discretization of nonlinear differential equations. *Neural Computing and Applications*, 1–16.

[65] Sznitman, A.S. (1991). *Topics in propagation of chaos*. Springer.

[66] Szpruch, Ł. and Tse, A. (2019). Antithetic multilevel particle system sampling method for Mckean-Vlasov SDEs. *arXiv preprint arXiv:1903.07063*.

[67] Szpruch, L., Tan, S. and Tse, A. (2019). Iterative multilevel particle approximation for Mckean-Vlasov SDEs. *Ann. Appl. Probab.*, **29**, 2230–2265.

[68] Talay, D. and Tubaro, L. (1990). Expansion of the global error for numerical schemes solving stochastic differential equations. *Stochastic analysis and applications*, **8**, 483–509.

[69] Tompson, J., Schlachter, K., Sprechmann, P. and Perlin, K. (2017). Accelerating eulerian fluid simulation with convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 3424–3433, JMLR. org.

[70] Tripathy, R.K. and Bilionis, I. (2018). Deep UQ: Learning deep neural network surrogate models for high dimensional uncertainty quantification. *Journal of Computational Physics*, **375**, 565–588.

[71] Weinan, E., Han, J. and 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**, 349–380.

[72] Xiu, D. (2010). *Numerical methods for stochastic computations: a spectral method approach*. Princeton university press.

[73] Xiu, D. and Hesthaven, J.S. (2005). High-order collocation methods for differential equations with random inputs. *SIAM Journal on Scientific Computing*, **27**, 1118–1139.

[74] Xiu, D. and Karniadakis, G.E. (2002). The Wiener–Askey polynomial chaos for stochastic differential equations. *SIAM journal on scientific computing*, **24**, 619–644.

[75] Yamazaki, F., Shinozuka, M. and Dasgupta, G. (1988). Neumann expansion for stochastic finite element analysis. *Journal of Engineering Mechanics*, **114**, 1335–1354.

[76] Yang, Y. and Perdikaris, P. (2019). Adversarial uncertainty quantification in physics-informed neural networks. *Journal of Computational Physics*, **394**, 136–152.

[77] Yentis, R. and Zaghloul, M. (1996). VLSI implementation of locally connected neural network for solving partial differential equations. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, **43**, 687–690.

[78] Zhang, D. (2001). *Stochastic methods for flow in porous media: coping with uncertainties*. Elsevier.