CIRJE-F-1178

Deep Asymptotic Expansion: Application to Financial Mathematics

Yuga Iguchi
MUFG Bank
UCL London

Riu Naito
Japan Post Insurance
Hitotsubashi University

Yusuke Okano
SMBC Nikko Securities

Akihiko Takahashi
The University of Tokyo

Toshihiro Yamada
Hitotsubashi University
Japan Science and Technology Agency

November 2021; Revised in February 2022

Deep Asymptotic Expansion: Application to Financial Mathematics

Yuga Iguchi
MUFG Bank
Tokyo, Japan
& UCL London, UK
yuga.iguchi@gmail.com

Riu Naito
Japan Post Insurance
& Hitotsubashi University
Tokyo, Japan
riu.naito@gmail.com

Yusuke Okano
SMBC Nikko Securities
Tokyo, Japan
yusukensw@gmail.com

Akihiko Takahashi
University of Tokyo
Tokyo, Japan
akihikot@e.u-tokyo.ac.jp

Toshihiro Yamada
Hitotsubashi University
& JST
Tokyo, Japan
toshihiro.yamada@r.hit-u.ac.jp

***Abstract*—The paper proposes a new computational scheme for diffusion semigroups based on an asymptotic expansion with weak approximation and deep learning algorithm to solve high-dimensional Kolmogorov partial differential equations (PDEs). In particular, we give a spatial approximation for the solution of d-dimensional PDEs on a range $[a, b]^d$ without suffering from the curse of dimensionality.**

***Index Terms*—Deep learning, Asymptotic expansion, Weak approximation, Kolmogorov PDEs, Malliavin calculus, Curse of dimensionality**

I. Introduction

Kolmogorov partial differential equations (PDEs) are widely used in various fields such as physics, engineering and financial mathematics. In general there are no closed form solutions except for a few special cases. Hence, numerical methods are usually required to solve Kolmogorov PDEs.

As classical schemes for solving Kolmogorov PDEs, finite element and finite difference methods are well known. These spatial approximation schemes work only for lower (typically from 1 to 3) dimensions since the computational complexity grows exponentially in the dimension of target Kolmogorov PDEs. In other words, finite element/difference methods suffer from the curse of dimensionality.

Instead, Monte Carlo methods can be applied to high dimensional cases due to the advantage of overcoming the curse of dimensionality. In perspective of solving Kolmogorov PDEs, some discretization methods (weak and strong approximations) of stochastic differential equations are used with Monte Carlo methods. However, Monte Carlo method provides an approximation at a fixed single point for the solution, that is, it does not give a spatial approximation.

We also point out that there exist closed form approximations for solutions of Kolmogorov PDEs such as asymptotic expansion methods. In particular, an expectation of a diffusion and a PDE solution at a single point are efficiently approximated with a probabilistic method [34]. For instance, see [19] [20] [21] [28] [29] [31] [36]. Moreover, some extended expansion methods such as [26] [32] [33] are proposed with discretization (weak approximation) schemes and Monte Carlo methods. Further, pure weak approximation schemes based on the concept of asymptotic expansion are obtained by [15] [23] [35].

Recently, deep learning-based methods for solving high dimensional PDEs have been developed by [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [27] [30], where the deep learning algorithms are used in a crucial step and then new tools for approximations of the solutions to high dimensional PDEs are obtained.

In this paper, we propose a new spatial approximation for the solution of high dimensional Kolmogorov PDEs by applying an asymptotic expansion and weak approximation scheme with a deep learning algorithm. The proposed scheme is inspired by the work of Beck et al. (2018) [1], and we provide an accurate deep learning-based approximation for PDEs without suffering from the curse of dimensionality. Particularly, we extend the work of [32] in the sense that the current work provides a new efficient second order weak approximation through a second order asymptotic expansion. More precisely, for a function $u^\varepsilon : [0, T] \times \mathbb{R}^d \to \mathbb{R}$ with a small parameter ε given by $u^\varepsilon(t, x) = E[f(X_T^{t,x,\varepsilon})]$ where f is a continuous function $f : \mathbb{R}^d \to \mathbb{R}$ and X^ε is a d-dimensional diffusion process, satisfying a Kolmogorov PDE

$$(\partial_t + \mathcal{L}_t^\varepsilon) u^\varepsilon(t, x) = 0, \quad u^\varepsilon(T, x) = f(x), \tag{1}$$

where $\mathcal{L}_t^\varepsilon$ is a second order differential operator, we construct a spatial approximation $Q_{t,t_1}^\varepsilon \circ \cdots \circ Q_{t_{n-1},t_n}^\varepsilon f$ for $u^\varepsilon(t, \cdot)$ on a certain domain $[a, b]^d \subset \mathbb{R}^d$ for a fixed $t < T$, as follows:

$$\sup_{x \in [a,b]^d} |u^\varepsilon(t, x) - Q_{t,t_1}^\varepsilon \circ \cdots \circ Q_{t_{n-1},t_n}^\varepsilon f(x)| = O\left(\frac{\varepsilon^2}{n^2}\right), \tag{2}$$

and approximate the function $Q_{t,t_1}^\varepsilon \circ \cdots \circ Q_{t_{n-1},t_n}^\varepsilon f$ by means of deep learning. We call this approximation the *deep asymptotic expansion* (Deep AE) for short.

The paper is organized as follows. In the next section, we introduce an asymptotic expansion approach for solving Kolmogorov PDEs with weak approximation. Section III describes a deep learning-algorithm for our asymptotic expansion method. Section IV shows numerical results to demonstrate the efficiency of the proposed method. Appendix provides proofs for propositions in the main text.

II. Asymptotic expansion and weak approximation

On a filtered probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_t, P)$, let $W = \{W_t\}_{t \geq 0}$ be a d-dimensional $\{\mathcal{F}_t\}$–Brownian motion. For $t \geq$

0 and $T>t$, let $X_s^{t,x,\varepsilon}$, $s\in[t,T]$, $x\in\mathbb{R}^d$ be the solution of

$$X_s^{t,x,\varepsilon}=x+\int_t^s\mu(r,X_r^{t,x,\varepsilon})dr+\varepsilon\sum_{i=1}^d\int_t^s\sigma_i(r,X_r^{t,x,\varepsilon})dW_r^i,$$

where $\varepsilon\in(0,1]$ and $\mu:[0,T]\times\mathbb{R}^d\to\mathbb{R}^d$, $\sigma_i:[0,T]\times\mathbb{R}^d\to\mathbb{R}^d$, $i=1,\ldots,d$ are continuous and bounded in t and continuously differentiable in x with bounded derivatives of any order. Let $\{P_{t,s}^\varepsilon\}_{s\geq t}$ be a two-parameter semigroup of linear operators given by

$$(P_{t,s}^\varepsilon f)(x)=E[f(X_s^{t,x,\varepsilon})],\quad s\geq t,\quad x\in\mathbb{R}^d, \tag{3}$$

for a continuous function $f:\mathbb{R}^d\to\mathbb{R}$. The aim of this paper is to show an approximation scheme for the function $x\mapsto(P_{t,T}^\varepsilon f)(x)$ where $f:\mathbb{R}^d\to\mathbb{R}$ is a continuous function. The d-dimensional process $X^{t,x,\varepsilon}=(X^{t,x,\varepsilon,1},\ldots,X^{t,x,\varepsilon,d})$ can be expanded as follows: for $i=1,\ldots,d$,

$$X_s^{t,x,\varepsilon,i}\sim X_s^{t,x,0,i}+\varepsilon X_{1,s}^{t,x,i}+\varepsilon^2X_{2,s}^{t,x,i}+\cdots \tag{4}$$

in Malliavin sense, where $X_s^{t,x,0,i}$ is the solution of $X_s^{t,x,0,i}=x+\int_t^s\mu^i(r,X_r^{t,x,0})dr$, and $X_{k,s}^{t,x,i}$, $k\in\mathbb{N}$ given by $X_{k,s}^{t,x,i}=\frac{1}{k!}\frac{\partial^k}{\partial\varepsilon^k}X_s^{t,x,\varepsilon,i}|_{\varepsilon=0}$. Let us define $\overline{X}_s^{t,x,\varepsilon}=X_s^{t,x,0}+\varepsilon X_{1,s}^{t,x}$ for $s\leq T$, where $X_{1,s}^{t,x}$ is explicitly obtained as the following Wiener integral: $X_{1,s}^{t,x}=\sum_{i=1}^d\int_t^sJ_{t\to s}^{x,0}(J_{t\to r}^{x,0})^{-1}\sigma_i(r,X_r^{t,x,0})dW_r^i$, with $J_{t\to r}^{x,0}=\partial/\partial xX_r^{t,x,0}$, $r\geq t$.

We introduce an expansion of $P_{t,T}^\varepsilon f$ with respect to the parameter ε and then give a second-order discretization with respect to the number of time-steps n. Here we only use polynomials of the Gaussian random variable $X_{1,t_{i+1}}^{t_i,x}$ up to the third order on each subinterval $[t_i,t_{i+1}]$, $i=0,1,\ldots,n-1$, where $t_i=t+i(T-t)/n$, $i=0,1,\ldots,n$ are the time-grids of the uniform partition on $[t,T]$. Let $\{Q_{t,s}^\varepsilon\}_{s\geq t}$ be linear operators given by

$$(Q_{t,s}^\varepsilon f)(x)=E[f(\overline{X}_s^{t,x,\varepsilon})\mathcal{W}_s^{t,x,\varepsilon}],\quad s>t,\quad x\in\mathbb{R}^d, \tag{5}$$

for a continuous and bounded function $f:\mathbb{R}^d\to\mathbb{R}$, where $\mathcal{W}_s^{t,x,\varepsilon}$ is a Malliavin weight given in Appendix.

Theorem 1. *Then, there exists $C>0$ such that*

$$\left\|P_{t,T}^\varepsilon f-Q_{t,t_1}^\varepsilon\circ\cdots\circ Q_{t_{n-1},t_n}^\varepsilon f\right\|_\infty\leq\varepsilon^2C\|f\|_\infty\frac{1}{n^2}, \tag{6}$$

for any $\varepsilon>0$, $n\geq1$ and continuous bounded function $f:\mathbb{R}^d\to\mathbb{R}$.

Proof of Theorem 1. See Section V □

Here, the approximation can be expressed as

$$(Q_{t,t_1}^\varepsilon\circ\cdots\circ Q_{t_{n-1},t_n}^\varepsilon f)(x)=E[f(\bar{X}_T^{t,x,(n)})\prod_{i=1}^n\mathcal{W}_{t_i}^{t_{i-1},\bar{X}_{t_{i-1}}^{t,x,(n)},\varepsilon}],$$

for $x\in\mathbb{R}^d$, where $\bar{X}_{t_i}^{t,x,(n)}=\overline{X}_{t_i}^{t_{i-1},\bar{X}_{t_{i-1}}^{t,x,(n)},\varepsilon}$, $i=1,\ldots,n$. Solutions of Kolmogorov PDEs are approximated in the following way. Let $u^\varepsilon:[0,T]\times\mathbb{R}^d\to\mathbb{R}$ be a function given by $u^\varepsilon(t,x)=E[f(X_T^{t,x,\varepsilon})]$ with a continuous function $f:\mathbb{R}^d\to\mathbb{R}$ of polynomial growth order, which satisfies

$$(\partial_t+\mathcal{L}_t^\varepsilon)u^\varepsilon(t,x)=0,\quad u^\varepsilon(T,x)=f(x), \tag{7}$$

where

$$\mathcal{L}_t^\varepsilon=\sum_{j=1}^d\mu^j(t,\cdot)\frac{\partial}{\partial x_j}+\frac{\varepsilon^2}{2}\sum_{i,j_1,j_2=1}^d\sigma_i^{j_1}(t,\cdot)\sigma_i^{j_2}(t,\cdot)\frac{\partial^2}{\partial x_{j_1}\partial x_{j_2}}.$$

Then, there exist $C>0$ and $q>0$ such that

$$\left|u^\varepsilon(t,x)-E[f(\bar{X}_T^{t,x,(n)})\prod_{i=1}^n\mathcal{W}_{t_i}^{t_{i-1},\bar{X}_{t_{i-1}}^{t,x,(n)},\varepsilon}]\right|\leq\varepsilon^2C(1+|x|^q)\frac{1}{n^2},$$

for any $\varepsilon>0$, $n\geq1$ and $x\in\mathbb{R}^d$.

III. Deep learning-based approximation

Let $a\in\mathbb{R}$, $b\in(a,\infty)$, $t>0$, $T>t$, $n\in\mathbb{N}$ and $\xi:\Omega\to[a,b]^d$ be a $\mathcal{F}_0/\mathcal{B}([a,b]^d)$-measurable uniformly distributed random variable. Let $f:\mathbb{R}^d\to\mathbb{R}$ be a continuous function with polynomial growth. We define $\mathbb{X}_T^{t,(n)}=\bar{X}_T^{t,\xi,(n)}$. Then, the following holds.

$$v^*=\operatorname{argmin}_{v\in C([a,b]^d)}E\Big[\Big|v(\xi)-f(\mathbb{X}_T^{t,(n)})\prod_{i=1}^n\mathcal{W}_{t_i}^{t_{i-1},\mathbb{X}_{t_{i-1}}^{t,(n)},\varepsilon}\Big|^2\Big],$$

and it holds that for all $x\in[a,b]^d$,

$$v^*(x)=Q_{t,t_1}^\varepsilon\circ\cdots\circ Q_{t_{n-1},t_n}^\varepsilon f(x). \tag{8}$$

With the above representation, the function $P_{t,T}^\varepsilon f$ can be approximated using deep learning. Let $\mathbb{R}^r\ni x\mapsto\mathcal{L}_r(x)\in\mathbb{R}^r$ be the Rectified Linear Unit (ReLU) activation function given by

$$\mathcal{L}_r(x)=(\max\{x_1,0\},\ldots,\max\{x_r,0\}),\quad x\in\mathbb{R}^r, \tag{9}$$

and for $p,\ell\in\mathbb{N}$, $q\in\{0\}\cup\mathbb{N}$, $\theta=(\theta_1,\ldots,\theta_\nu)\in\mathbb{R}^\nu$ with $\nu\in\mathbb{N}$ such that $q+\ell(p+1)\leq\nu$, let $A_{p,\ell}^{\theta,q}:\mathbb{R}^p\to\mathbb{R}^\ell$ be a function given by

$$A_{p,\ell}^{\theta,q}(x)=\begin{pmatrix}\theta_{q+1}&\cdots&\theta_{q+p}\\\vdots&\ddots&\vdots\\\theta_{q+(\ell-1)p+1}&\cdots&\theta_{q+\ell p}\end{pmatrix}\begin{pmatrix}x_1\\\vdots\\x_p\end{pmatrix}+\begin{pmatrix}\theta_{q+\ell p+1}\\\vdots\\\theta_{q+\ell p+\ell}\end{pmatrix}.$$

Let $s\in\{3,4,5,\ldots\}$ such that $\sum_{k=1}^sd_k(d_{k-1}+1)\leq\nu$ for $d_0=d$, $d_s=1$, $d_1,\ldots,d_{s-1}\in\mathbb{N}$. Then, we have

$$P_{t,T}^\varepsilon f\approx Q_{t,T}^{\varepsilon,[n],\theta^*}f, \tag{10}$$

where $Q_{t,T}^{\varepsilon,[n],\theta^*}f$ is given by

$$Q_{t,T}^{\varepsilon,[n],\theta}f(x)=(A_{d_{s-1},d_s}^{\theta,\sum_{k=1}^{s-1}d_k(d_{k-1}+1)}\circ\mathcal{L}_{d_{s-1}}\circ A_{d_{s-2},d_{s-1}}^{\theta,\sum_{k=1}^{s-2}d_k(d_{k-1}+1)}\circ\cdots\circ\mathcal{L}_{d_2}\circ A_{d_1,d_2}^{\theta,d_1(d_0+1)}\circ\mathcal{L}_{d_1}\circ A_{d_0,d_1}^{\theta,0})(x),\ x\in\mathbb{R}^d$$

with θ^* satisfying

$$\theta^*=\operatorname{argmin}_\theta E\Big[\Big|Q_{t,T}^{\varepsilon,[n],\theta}f(\xi)-f(\mathbb{X}_T^{t,(n)})\prod_{i=1}^n\mathcal{W}_{t_i}^{t_{i-1},\mathbb{X}_{t_{i-1}}^{t,(n)},\varepsilon}\Big|^2\Big].$$

The function $Q_{t,T}^{\varepsilon,[n],\theta^*}f:\mathbb{R}^d\to\mathbb{R}$ with θ^* represents an artificial neural network with $s+1$ layers (1 input layer with d neurons, k-th hidden layers with d_k neurons for each $k=1,\ldots,s-1$, and 1 output layer with 1 neuron).

IV. NUMERICAL EXAMPLES FOR FINANCIAL MATHEMATICS

In the section, we apply the proposed method to the following d-dimensional Kolmogorov PDE:

$$(\partial_t + \mathcal{L}_t^\varepsilon) u^\varepsilon(t,x) = 0, \quad u^\varepsilon(T,x) = f(x), \tag{11}$$

where $\mathcal{L}_t^\varepsilon$ is a second order differential operator given by

$$\mathcal{L}_t^\varepsilon \varphi(x) = \sum_{j=1}^{d} r x_j \frac{\partial \varphi(x)}{\partial x_j} + \frac{\varepsilon^2}{2} \sum_{i,j_1,j_2=1}^{d} \sigma_i^{j_1} x_{j_1} \sigma_i^{j_2} x_{j_2} \frac{\partial^2 \varphi(x)}{\partial x_{j_1} \partial x_{j_2}}, \tag{12}$$

for a smooth function φ and $x \in \mathbb{R}^d$, and f is a continuous function which is specified in the following subsections.

A. Numerical error of Deep AE

First, we evaluate the error for an entire region $[a,b]^d$. Let $d = 10$, $a = 99.0$, $b = 101.0$, $r = 0.01$, $\sigma_i^j = \delta_i^j$, $\varepsilon = 0.2$, $T = 5.0$, and $f_d : \mathbb{R}^d \to \mathbb{R}$ be a function $y \mapsto \max\{\max\{y_1 - K, 0\}, \ldots, \max\{y_d - K, 0\}\}$ with $K = 100.0$, where δ_i^j is the Kronecker delta. As an example, we apply the proposed second order asymptotic expansion with second order scheme (Deep AE) and continuous uniformly distributed random variable $\xi : \Omega \to [a,b]^d$. For comparison, we use Deep EM scheme, the standard deep learning-based splitting method of Beck et al. (2018). Here, we use 1 input layer, 2 hidden layers, 1 output layer with neurons $(d, d+50, d+50, 1)$ in the deep learning computation. Also, the batch size M, the train steps J and learning rate $\gamma(j)$, $j \le J$ in the stochastic gradient descent method are taken as $M = 4096$, $J = 50000$ and $\gamma(j) = 10^{-2}\mathbf{1}_{[0,\ 0.2J]}(j) + 10^{-3}\mathbf{1}_{(0.2J,\ 0.6J]}(j) + 10^{-4}\mathbf{1}_{(0.6J,\ J]}(j)$, $j = 0, 1, \ldots, J$ for both schemes. As an error analysis after the solution of Kolmogorov PDE is estimated by each scheme, we compute $\max_{x \in \{y_0,\ldots,y_k\}} \left| \frac{\mathrm{Ref}(x) - \mathrm{Deep\ AE}(x,n)}{\mathrm{Ref}(x)} \right|$ and $\max_{x \in \{y_0,\ldots,y_k\}} \left| \frac{\mathrm{Ref}(x) - \mathrm{Deep\ EM}(x,n)}{\mathrm{Ref}(x)} \right|$, where $y_i = (a + (b-a)i/k, \ldots, a + (b-a)i/k) \in \mathbb{R}^d$, $k = 20$, $i \le k$, Deep $\mathrm{AE}(x,n)$ and Deep $\mathrm{EM}(x)$, $x \in \{y_0, \ldots, y_k\}$ represent numerical values of Deep AE and Deep EM, respectively, and $\mathrm{Ref}(x)$, $x \in \{y_0, \ldots, y_k\}$ are computed by Monte Carlo simulations with the number of paths 10^8 and the explicit solution of X^x obtained by Itô formula. The table below shows that the convergence of our scheme is faster than that of deep EM as spatial approximation.

TABLE I
THE NUMERICAL ERROR FOR SPATIAL APPROXIMATIONS (DEEP AE AND DEEP EM)

Number of train steps	Error for Deep AE ($n = 2^1$)	Runtime for Deep AE ($n = 2^1$)	Error for Deep EM ($n = 2^6$)	Runtime (s) for Deep EM ($n = 2^6$)
50000	0.00609	459.45s	0.00671	2636.21s

B. Weak convergence

Next, we check the rate of weak convergence based on the theoretical estimate (6). As in the previous subsection, we compare the accuracies of the proposed scheme with those of Deep EM of Beck et al. (2018). In the experiments, we first estimate the function $Q_{t,t_1}^\varepsilon \circ \cdots \circ Q_{t_{n-1},t_n}^\varepsilon f$ on a region $[a,b]^d$ with continuous uniformly distributed random variable $\xi : \Omega \to [a,b]^d$, and compute $Q_{t,t_1}^\varepsilon \circ \cdots \circ Q_{t_{n-1},t_n}^\varepsilon f(x)$ at $x \in [a,b]^d$. Then, we check the numerical error, where our reference value is computed by a Monte Carlo simulation with the number of paths 10^8.

Figure 1 shows the result for $d = 10$, $a = 99.0$, $b = 101.0$, $r = 0.015$, $\sigma_i^j = \delta_i^j$, $\varepsilon = 0.3$, $T = 2.0$, $f : \mathbb{R}^d \to \mathbb{R}$ given by $y \mapsto \max\{\max\{y_1 - K, 0\}, \ldots, \max\{y_d - K, 0\}\}$ with $K = 100.0$, and $x = (100.0, \ldots, 100.0) \in [a,b]^d$, where the relative errors are plotted. Here, we use 1 input layer, 2 hidden layers, 1 output layer with neurons $(d, d+50, d+50, 1)$ in the deep learning computation. Also, the batch size M, the train steps J and learning rate $\gamma(j)$, $j \le J$ in the stochastic gradient descent method are taken as $M = 8192$, $J = 50000$ and $\gamma(j) = 10^{-2}\mathbf{1}_{[0,\ 0.2J]}(j) + 10^{-3}\mathbf{1}_{(0.2J,\ 0.6J]}(j) + 10^{-4}\mathbf{1}_{(0.6J,\ J]}(j)$, $j = 0, 1, \ldots, J$. In Table II, numerical errors and runtimes of the schemes are shown.

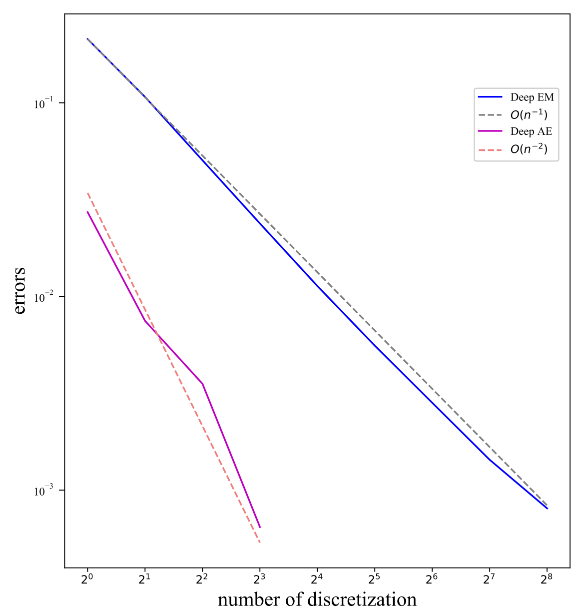

Fig. 1. Weak convergence ($d = 10$)

TABLE II

THE NUMERICAL ERROR AT $x = (100.0, \dots, 100.0) \in [a, b]^d$ ($d = 10$)

Number of train steps	Error for Deep AE ($n = 2^3$)	Runtime (s) for Deep AE ($n = 2^3$)	Error for Deep EM ($n = 2^8$)	Runtime (s) for Deep EM ($n = 2^8$)
50000	0.00064	1375.25s	0.00080	19350.10s

Figure 2 shows the example for $d = 100$, $a = 99.0$, $b = 101.0$, $r = 0.015$, $\sigma_i^j = \delta_i^j$, $\varepsilon = 0.2$, $T = 0.5$, $f : \mathbb{R}^d \to \mathbb{R}$ given by $y \mapsto \max\{\max\{y_1 - K, 0\}, \dots, \max\{y_d - K, 0\}\}$ with $K = 100.0$, and $x = (100.0, \dots, 100.0) \in [a, b]^d$, where the relative errors are plotted. Here, we use 1 input layer, 2 hidden layers, 1 output layer with neurons $(d, d+50, d+50, 1)$ in the deep learning computation. In the stochastic gradient descent method, the batch size M, the train steps J and learning rate $\gamma(j)$, $j \le J$ are taken as $M = 1024$, $J = 25000$ and $\gamma(j) = 5 \times 10^{-2} \mathbf{1}_{[0,\ 0.2J]}(j) + 5 \times 10^{-3} \mathbf{1}_{(0.2J,\ 0.6J]}(j) + 5 \times 10^{-4} \mathbf{1}_{(0.6J,\ J]}(j)$, $j = 0, 1, \dots, J$. Table III shows numerical errors and runtimes of the schemes.

Those figures and tables demonstrate that our Deep AE gives more accurate approximations than Deep EM and provides high performance in terms of runtime to achieve the same level of accuracies.

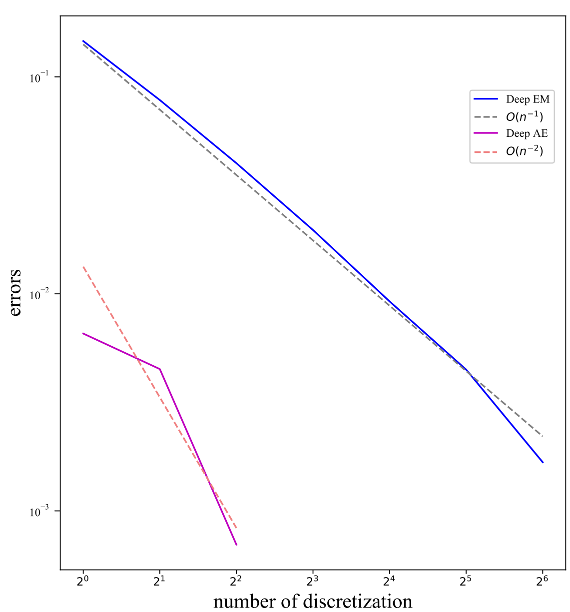

Fig. 2. Weak convergence ($d = 100$)

TABLE III

THE NUMERICAL ERROR AT $x = (100.0, \dots, 100.0) \in [a, b]^d$ ($d = 100$)

Number of train steps	Error for Deep AE ($n = 2^1$)	Runtime (s) for Deep AE ($n = 2^1$)	Error for Deep AE ($n = 2^2$)	Runtime (s) for Deep AE ($n = 2^2$)
25000	0.00450	298.61s	0.00070	463.20s

Number of train steps	Error for Deep EM ($n = 2^5$)	Runtime (s) for Deep EM ($n = 2^5$)	Error for Deep EM ($n = 2^6$)	Runtime (s) for Deep EM ($n = 2^6$)
25000	0.00448	1544.26s	0.00168	3273.39s

Throughout the numerical experiments, we have checked that the proposed scheme works as a spacial approximation in high-dimensional PDE models and the numerical results are consistent with theoretical parts given in Section II and III.

V. PROOF OF THEOREM 1

We prepare some notations on Malliavin calculus. Let $\mathbb{D}^\infty$ be the space of smooth Wiener functionals $F : C([0,T]; \mathbb{R}^d) \to \mathbb{R}$ in the sense of Malliavin. For a *nondegenerate* $F \in (\mathbb{D}^\infty)^d$, $G \in \mathbb{D}^\infty$ and a multi-index γ, there exists $H_\gamma(F, G) \in \mathbb{D}^\infty$ such that

$$\text{(IBP)} \qquad E[\partial^\gamma \varphi(F) G] = E[\varphi(F) H_\gamma(F, G)] \tag{13}$$

for all $\varphi \in C_b^\infty(\mathbb{R}^d)$. See Chapter V.8-10 in Ikeda and Watanabe (1989) [15] and Chapter 1-2 in Nualart (2006) [25] for the details.

The following lemma is useful for the proof of Theorem 1.

Lemma 1. *Let $0 \le t < s$, $k \ge 3$, $\Delta_{t,s} := \{(t_1, \dots, t_k) \in \mathbb{R}^k;\ t \le t_1 < \cdots < t_k \le s\}$ and $\alpha \in \{0, 1, \cdots, d\}^k$ be a multi-index. Let $h : \Delta_{t,s} \to \mathbb{R}$ be a bounded function. There exists $C > 0$ such that*

$$\sup_{x \in \mathbb{R}^d} \left| E[g(\bar{X}_s^{t,x,\varepsilon}) \int_{t < t_1 < \cdots < t_k < s} h(t_1, \dots, t_k) dW_{t_1}^{\alpha_1} \cdots dW_{t_k}^{\alpha_k}] \right|$$
$$\le C \varepsilon^{\#\{\ell;\ \alpha_\ell \neq 0\}} \|\nabla^{\#\{\ell;\ \alpha_\ell \neq 0\}} g\|_\infty (s - t)^k, \tag{14}$$

for all $\varepsilon \in (0, 1]$, $g \in C_b^\infty(\mathbb{R}^d, \mathbb{R})$ and $t < s \le T$.

Proof of Lemma 1. Use the duality formula in Malliavin calculus. $\square$

In the first step, we expand $P_{t,s}^\varepsilon \varphi$ for $\varphi \in C_b^\infty(\mathbb{R}^d)$ as follows:

$$P_{t,s}^\varepsilon \varphi(x) = E[\varphi(\bar{X}_s^{t,x,\varepsilon})] + \sum_{k=1}^{5} \sum_{\ell=1}^{k} \sum_{\substack{k_1 + \cdots + k_\ell = k + \ell,\\ k_p \ge 2}} \sum_{\alpha^{(\ell)} \in \{1, \cdots, d\}^\ell} \frac{1}{\ell!}$$
$$E\Big[\partial_{\alpha^{(\ell)}} \varphi(\bar{X}_s^{t,x,\varepsilon}) \prod_{p=1}^{\ell} \varepsilon^{k_p} X_{k_p,s}^{t,x,\alpha_p}\Big] + \mathscr{R}_{1,s}^{t,\varepsilon} \varphi(x), \quad x \in \mathbb{R}^d. \tag{15}$$

Here, $\mathscr{R}_{1,s}^{t,\varepsilon}\varphi$ satisfies $\sup_x |\mathscr{R}_{1,s}^{t,\varepsilon}\varphi(x)| \le C\varepsilon^6 \|\varphi\|_\infty (s - t)^3$, where the constant $C > 0$ does not depend on φ and t, s. Using the integration by parts on the Wiener space with Lemma 1, we get

$$P^{\varepsilon}_{t,s}\varphi(x)-Q^{\varepsilon}_{t,s}\varphi(x)=\mathscr{R}^{t,\varepsilon}_{1,s}\varphi(x)+\mathscr{R}^{t,\varepsilon}_{2,s}\varphi(x),\quad x\in\mathbb{R}^d, \tag{16}$$

where $\mathscr{R}^{t,\varepsilon}_{2,s}\varphi$ satisfies $\|\mathscr{R}^{t,\varepsilon}_{2,s}\varphi\|_\infty \leq C\sum_{e=0}^{q}\varepsilon^{2+e}\|\nabla^e\varphi\|_\infty(s-t)^3$ for some $C>0$ which does not depend on φ and t,s.

We now estimate the global error. Note that the following decomposition holds: for $x\in\mathbb{R}^d$,

$$\begin{aligned}&P^{\varepsilon}_{t,T}f(x)-Q^{\varepsilon}_{t,t_1}\circ\cdots\circ Q^{\varepsilon}_{t_{n-1},t_n}f(x)\\&=\sum_{i=0}^{n-1}Q^{\varepsilon}_{t,t_1}\circ\cdots\circ Q^{\varepsilon}_{t_{i-1},t_i}(P^{\varepsilon}_{t_i,t_{i+1}}-Q^{\varepsilon}_{t_i,t_{i+1}})P^{\varepsilon}_{t_{i+1},T}f(x)\\&=\sum_{i=0}^{n-1}\sum_{\ell=1}^{2}Q^{\varepsilon}_{t,t_1}\circ\cdots\circ Q^{\varepsilon}_{t_{i-1},t_i}\mathscr{R}^{t_i,\varepsilon}_{\ell,t_{i+1}}P^{\varepsilon}_{t_{i+1},T}f(x).\end{aligned} \tag{17}$$

For $Q^{\varepsilon}_{t,t_1}\circ\cdots\circ Q^{\varepsilon}_{t_{i-1},t_i}\mathscr{R}^{t_i,\varepsilon}_{1,t_{i+1}}P^{\varepsilon}_{t_{i+1},T}f$, we immediately have

$$\begin{aligned}&\left\|Q^{\varepsilon}_{t,t_1}\circ\cdots\circ Q^{\varepsilon}_{t_{i-1},t_i}\mathscr{R}^{t_i,\varepsilon}_{1,t_{i+1}}P^{\varepsilon}_{t_{i+1},T}f\right\|_\infty\\&\leq c\|\mathscr{R}^{t_i,\varepsilon}_{1,t_{i+1}}P^{\varepsilon}_{t_{i+1},T}f\|_\infty\leq C\varepsilon^6\|f\|_\infty\frac{1}{n^3}.\end{aligned} \tag{18}$$

We next estimate the bound of $Q^{\varepsilon}_{t,t_1}\circ\cdots\circ Q^{\varepsilon}_{t_{i-1},t_i}\mathscr{R}^{t_i,\varepsilon}_{2,t_{i+1}}P^{\varepsilon}_{t_{i+1},T}f$. For $0\leq i\leq[n/2]-1$,

$$\begin{aligned}&\|Q^{\varepsilon}_{t,t_1}\circ\cdots\circ Q^{\varepsilon}_{t_{i-1},t_i}\mathscr{R}^{t_i,\varepsilon}_{2,t_{i+1}}P^{\varepsilon}_{t_{i+1},T}f\|_\infty\\&\leq c\sum_{e=0}^{q}\varepsilon^{2+e}\|\nabla^eP^{\varepsilon}_{t_{i+1},T}f\|_\infty\frac{1}{n^3}\leq\varepsilon^2C\|f\|_\infty\frac{1}{n^3},\end{aligned} \tag{19}$$

where we used the estimate $\sup_i\|\nabla^eP^{\varepsilon}_{t_{i+1},T}f\|_\infty\leq\varepsilon^{-e}\|f\|_\infty C$ for some $C>0$ independent of f and n. For $[n/2]\leq i\leq n-1$, we apply the integration by parts on the Wiener space:

$$\begin{aligned}&Q^{\varepsilon}_{t,t_1}\circ\cdots\circ Q^{\varepsilon}_{t_{i-1},t_i}\mathscr{R}^{t_i,\varepsilon}_{2,t_{i+1}}P^{\varepsilon}_{t_{i+1},T}f(x)\\&=\sum_{e=1}^{q}\varepsilon^{2+e}E[P^{\varepsilon}_{t_{i+1},T}f(\bar{X}^{t,x,(n)}_{t_{i+1}})M^{t,x,(n),\varepsilon}_{e,t_{i+1}}],\end{aligned} \tag{20}$$

where $M^{t,x,(n),\varepsilon}_{e,t_{i+1}}$ satisfies for $p\geq1$, $\sup_{[n/2]\leq i\leq n-1}\|M^{t,x,(n),\varepsilon}_{e,t_{i+1}}\|_p\leq\varepsilon^{-e}Cn^{-3}$ with some $C>0$ independent of x and n, and get

$$\|Q^{\varepsilon}_{t,t_1}\circ\cdots\circ Q^{\varepsilon}_{t_{i-1},t_i}\mathscr{R}^{t_i,\varepsilon}_{2,t_{i+1}}P^{\varepsilon}_{t_{i+1},T}f\|_\infty\leq C\varepsilon^2\|f\|_\infty\frac{1}{n^3}. \tag{21}$$

Then, by (18), (19) and (21), we have the assertion:

$$\left\|P^{\varepsilon}_{t,T}f-Q^{\varepsilon}_{t,t_1}\circ\cdots\circ Q^{\varepsilon}_{t_{n-1},t_n}f\right\|_\infty\leq\varepsilon^2C\|f\|_\infty\frac{1}{n^2}.\quad\square \tag{22}$$

VI. Conclusions

In the paper, we introduced a computational scheme for diffusion semigroups based on an asymptotic expansion with weak approximation and deep learning algorithm to solve high-dimensional Kolmogorov PDEs. In particular, we provided a spatial approximation for the solution of d-dimensional PDEs on a hypercube $[a,b]^d$ without suffering from the curse of dimensionality. It can be regarded as an extension of classical finite element method of PDEs. Numerical experiments demonstrated the validity and the effectiveness of the proposed scheme.

Appendix

We give the formula of Malliavin weight in the following:

$$\begin{aligned}\mathcal{W}^{t,x,\varepsilon}_s&=1+\sum_{i_1,i_2,i_3=1}^{d}H_{(i_1,i_2,i_3)}(X^{t,x}_{1,s},1)A_{i_1,i_2,i_3}(t,s,x)\\&+\sum_{i_1,i_2=1}^{d}H_{(i_1,i_2)}(X^{t,x}_{1,s},1)A_{i_1,i_2}(t,s,x)\\&+\sum_{i_1=1}^{d}H_{(i_1)}(X^{t,x}_{1,s},1)A_{i_1}(t,s,x),\end{aligned} \tag{23}$$

with $A_{i_1,i_2,i_3}(t,s,x)=\sum_{j_1,k_1,k_2=1}^{d}C^{(1),k_1,k_2}_{i_1,i_2,i_3,j_1}(t,s,x)+\sum_{j_1,j_2,k_1,k_2=1}^{d}C^{(2),k_1,k_2}_{i_1,i_2,i_3,j_1,j_2}(t,s,x)$, $A_{i_1,i_2}(t,s,x)=\sum_{j_1,j_2,k_1,k_2=1}^{d}C^{(3),k_1,k_2}_{i_1,i_2,j_1,j_2}(t,s,x)+\sum_{j_1,j_2,k_1,k_2,k_3=1}^{d}C^{(4),k_1,k_2,k_3}_{i_1,i_2,j_1,j_2}(t,s,x)$ and $A_{i_1}(t,s,x)=\sum_{j_1,j_2,k_1,k_2=1}^{d}C^{(5),k_1,k_2}_{i_1,j_1,j_2}(t,s,x)$,

$$\begin{aligned}&C^{(1),k_1,k_2}_{i_1,i_2,i_3,j_1}(t,s,x)\\&=\varepsilon\int_t^s\int_t^{t_1}a^{i_3}_{k_2}(t_2,s,x)a^{i_2}_{k_1}(t_1,s,x)b^{i_1,j_1}_{k_1}(t_1,s,x)a^{j_1}_{k_2}(t_2,t_1,x)dt_2dt_1,\end{aligned}$$

$$\begin{aligned}&C^{(2),k_1,k_2}_{i_1,i_2,i_3,j_1,j_2}(t,s,x)\\&=\varepsilon\int_t^s\int_t^{t_1}\int_t^{t_2}a^{i_3}_{k_1}(t_3,s,x)a^{i_2}_{k_2}(t_2,s,x)\\&\qquad c^{i_1,j_1,j_2}(t_1,s,x)a^{j_1}_{k_1}(t_3,t_1,x)a^{j_2}_{k_2}(t_2,t_1,x)dt_3dt_2dt_1,\end{aligned}$$

$$\begin{aligned}&C^{(3),k_1,k_2}_{i_1,i_2,j_1,j_2}(t,s,x)=\varepsilon^2\frac{1}{2}\\&\times\int_t^s\int_t^{t_1}a^{j_1}_{k_1}(t_2,t_1,x)b^{i_1,j_1}_{k_2}(t_1,s,x)a^{j_2}_{k_1}(t_2,t_1,x)b^{i_2,j_2}_{k_2}(t_1,s,x)dt_2dt_1,\end{aligned}$$

$$\begin{aligned}&C^{(4),k_1,k_2,k_3}_{i_1,i_2,j_1,j_2}(t,s,x)\\&=\varepsilon^2\frac{1}{2}1_{k_1=k_2}\int_t^s\int_t^{t_1}a^{i_2}_{k_3}(t_1,s,x)d^{i_1,j_1,j_2}_{k_3}(t_1,s,x)\\&\qquad a^{j_1}_{k_1}(t_2,t_1,x)a^{j_2}_{k_2}(t_2,t_1,x)dt_2dt_1,\end{aligned}$$

$$\begin{aligned}&C^{(5),k_1,k_2}_{i_1,j_1,j_2}(t,s,x)\\&=\varepsilon\frac{1}{2}1_{k_1=k_2}\int_t^s\int_t^{t_1}c^{i_1,j_1,j_2}(t_1,s,x)a^{j_2}_{k_2}(t_2,t_1,x)a^{j_1}_{k_1}(t_2,t_1,x)dt_2dt_1,\end{aligned}$$

and

$$a^i_k(v,u,x):=\sum_{j_1,j_2=1}^{d}[J^{x,0}_{t\to u}]^i_{j_1}[(J^{x,0}_{t\to v})^{-1}]^{j_1}_{j_2}\sigma^{j_2}_k(v,X^{t,x,0}_v), \tag{24}$$

$$b^{i,j_3}_k(v,u,x):=\sum_{j_1,j_2=1}^{d}[J^{x,0}_{t\to u}]^i_{j_1}[(J^{x,0}_{t\to v})^{-1}]^{j_1}_{j_2}\partial_{j_3}\sigma^{j_2}_k(v,X^{t,x,0}_v), \tag{25}$$

$$c^{i,j_3,j_4}(v,u,x):=\sum_{j_1,j_2=1}^{d}[J^{x,0}_{t\to u}]^i_{j_1}[(J^{x,0}_{t\to v})^{-1}]^{j_1}_{j_2}[\partial^2\mu^{j_2}(v,X^{t,x,0}_v)]^{j_3}_{j_4}, \tag{26}$$

$$d^{i,j_3,j_4}_k(v,u,x):=\sum_{j_1,j_2=1}^{d}[J^{x,0}_{t\to u}]^i_{j_1}[(J^{x,0}_{t\to v})^{-1}]^{j_1}_{j_2}[\partial^2\sigma^{j_2}_k(s,X^{t,x,0}_s)]^{j_3}_{j_4}. \tag{27}$$

Acknowledgements

We appreciate the comments from the two anonymous referees. The authors thank Prof. Shigeo Kusuoka (University of Tokyo) for grateful advice for the method of the paper. This work is supported by JST PRESTO (Grant Number JPMJPR2029), Japan.

References

[1] C. Beck, S. Becker, P. Grohs, N. Jaafari and A. Jentzen, Solving stochastic differential equations and Kolmogorov equations by means of deep learning, *Journal of Scientific Computing*, 3 (2021)

[2] C. Beck, S. Becker, P. Cheridito, A. Jentzen and A. Neufeld, Deep splitting method for parabolic PDEs, *SIAM Journal on Scientific Computing*, 43(5), 3135-3154 (2021)

[3] J. Berner, P. Grohs and A. Jentzen, Analysis of the generalization error: Empirical risk minimization over deep artificial neural networks overcomes the curse of dimensionality in the numerical approximation of Black–Scholes partial differential equations, *SIAM Journal on Mathematics of Data Science*, 2(3), 631-657 (2020)

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

[5] Weinan E, J. Han and A. Jentzen, Algorithms for Solving High Dimensional PDEs: From Nonlinear Monte Carlo to Machine Learning, *Nonlinearity*, to appear (2020)

[6] D. Elbrächter, P. Grohs, A. Jentzen and C. Schwab, DNN expression rate analysis of high-dimensional PDEs: Application to option pricing, *Constr. Approx.*, to appear (2021)

[7] M. Fujii, A. Takahashi and M. Takahashi, Asymptotic expansion as prior knowledge in deep learning method for high dimensional BSDEs, *Asia-Pacific Financial Markets*, 26, 391-408 (2019)

[8] P. Grohs, A. Jentzen and D. Salimova, Deep neural network approximations for Monte Carlo algorithms, *Partial Differ. Equ. Appl.*, to appear (2019)

[9] P. Grohs, F. Hornung, A. Jentzen and P. Zimmermann, Space-time error estimates for deep neural network approximations for differential equations, arXiv (2019)

[10] J. Han and J. Long, Convergence of the Deep BSDE method for coupled FBSDEs, *Probability, Uncertainty and Quantitative Risk*, 5(5) (2020)

[11] J. Han, J. Lu and M. Zhou, Solving high-dimensional eigenvalue problems using deep neural networks: A diffusion Monte Carlo like approach, *Journal of Computational Physics*, Volume 423, 15, December 2020, 109792 (2020)

[12] J. Han, L. Zhang and Weinan E, Solving many-electron Schrödinger equation using deep neural networks, *Journal of Computational Physics*, Volume 399, 15, December 2019, 108929 (2019)

[13] F. Hornung, A. Jentzen and D. Salimova, Space-time deep neural network approximations for high-dimensional partial differential equations, arXiv (2020)

[14] N. Ikeda and S. Watanabe, Stochastic Differential Equations and Diffusion Processes, 2nd ed., North-Holland, Amsterdam, Kodansha, Tokyo (1989)

[15] Y. Iguchi and T. Yamada, Operator splitting around Euler-Maruyama scheme and high order discretization of heat kernels, *ESAIM: Mathematical Modelling and Numerical Analysis*, vol 55, 323-367 (2021)

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

[17] S. Kusuoka and D. Stroock, Applications of the Malliavin calculus Part I, *Stochastic Analysis* (Katata/Kyoto 1982), 271-306 (1984)

[18] P.E. Kloeden and E. Platen, *Numerical Solution of Stochastic Differential Equations*, Springer (1999)

[19] N. Kunitomo and A. Takahashi, The asymptotic expansion approach to the valuation of interest rate contingent claims, *Mathematical Finance*, 11, 117-151 (2001)

[20] N. Kunitomo and A. Takahashi, On validity of the asymptotic expansion approach in contingent claim analysis, *Annals of Applied Probability*, 13(3), 914-952 (2003)

[21] P. Malliavin and A. Thalmaier, *Stochastic Calculus of Variations in Mathematical Finance*, Springer (2006)

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

[23] R. Naito and T. Yamada, A third-order weak approximation of multidimensional Itô stochastic differential equations, *Monte Carlo Methods and Applications*, vol 25 (2), 97-120 (2019)

[24] R. Naito and T. Yamada, A machine learning solver for high-dimensional integrals: Solving Kolmogorov PDEs by stochastic weighted minimization and stochastic gradient descent through a high-order weak approximation scheme of SDEs with Malliavin weights, arXiv (2020)

[25] D. Nualart, *The Malliavin Calculus and Related Topics*, Springer (2006)

[26] Y. Okano and T. Yamada, A control variate method for weak approximation of SDEs via discretization of numerical error of asymptotic expansion, *Monte Carlo Methods and Applications*, 25(3) (2019)

[27] J. Sirignano and K. Spiliopoulos, DGM: A deep learning algorithm for solving partial differential equations, *Journal of Computational Physics*, Vol 375, 1339-1364 (2018)

[28] A. Takahashi, An asymptotic expansion approach to pricing financial contingent claims, *Asia-Pacific Financial Markets*, 6(2), 115-151 (1999)

[29] A. Takahashi, Asymptotic expansion approach in finance, *Large Deviations and Asymptotic Methods in Finance* (P. Friz, J. Gatheral, A. Gulisashvili, A. Jacquier and J. Teichmann ed.), Springer Proceedings in Mathematics & Statistics (2015)

[30] A. Takahashi, Y. Tsuchida and T. Yamada, A new efficient approximation scheme for solving high-dimensional semilinear PDEs: control variate method for Deep BSDE solver, arXiv (2021)

[31] A. Takahashi and T. Yamada, An asymptotic expansion with push-down of Malliavin weights, *SIAM Journal on Financial Mathematics*, 3, 95-136 (2012)

[32] A. Takahashi and T. Yamada, A weak approximation with asymptotic expansion and multidimensional Malliavin weights, *Annals of Applied Probability*, 26(2), 818-856 (2016)

[33] A. Takahashi and N. Yoshida, Monte Carlo simulation with asymptotic method, *Journal of the Japan Statistical Society* 35(2), 171-203 (2005)

[34] S. Watanabe, Analysis of Wiener functionals (Malliavin calculus) and its applications to heat kernels, *Annals of Probability*, 15, 1-39 (1987)

[35] T. Yamada, An arbitrary high order weak approximation of SDE and Malliavin Monte Carlo: application to probability distribution functions, *SIAM Journal on Numerical Analysis*, 57(2), 563-591 (2019)

[36] N. Yoshida, Asymptotic expansions of maximum likelihood estimators for small diffusions via the theory of Malliavin-Watanabe, *Probability Theory and Related Fields*, 92, 275-311 (1992)