AIMS Mathematics, 9(5): 11975–11991.
DOI: 10.3934/math.2024585
Received: 16 January 2024
Revised: 12 March 2024
Accepted: 20 March 2024
Published: 27 March 2024

http://www.aimspress.com/journal/Math

Research article

An efficient data-driven approximation to the stochastic differential equations with non-global Lipschitz coefficient and multiplicative noise

Xiao Qi, Tianyao Duan and Huan Guo*

School of Artificial Intelligence, Jianghan University, 430056 Wuhan, China

* **Correspondence:** Email: guohuan.2007@163.com.

Abstract: This paper studied the numerical approximation of the stochastic differential equations driven by non-global Lipschitz drift coefficient and multiplicative noise. An efficient data-driven method, called extended continuous latent process flow, was proposed for the underlying problem. Compared with the piecewise construction of a variational posterior process used in the classical continuous latent process flow developed by Deng et al. [13], the principle idea of our method was to derive a variational lower bound by constructing a posterior latent process conditional on all information over the whole time interval to maximize the log-likelihood generated by the observations, which reduces the computational cost and, thus, provides a convenient way to approximate the considered equation. Particularly, our new method showed a better approximation to the underlying equation than the classical drift-θ discretization scheme through numerical error comparison. Numerical experiments were finally reported to demonstrate the effectiveness and generalization performance of the proposed method.

Keywords: non-global Lipschitz; drift-θ Milstein scheme; neural stochastic differential equations; extended continuous latent process flow
Mathematics Subject Classification: 62M45, 60H35, 82C32

1. Introduction

Stochastic differential equations (SDEs) are widely applied in scientific areas such as phase transitions, electromagnetic scattering, tumor growth, neural dynamics, and many other branches of science. However, in most cases, analytical solutions of nonlinear SDEs are not available, which makes the research on numerical methods of SDEs receive widespread attention in practice. In general, imposing global Lipschitz conditions on the coefficient functions of SDEs facilitates the numerical analysis of SDEs; see, e.g., [28, 34] and references therein. Nevertheless, SDEs arising from practical applications rarely follow this traditional but restrictive condition, such as the $\frac{3}{2}$-volatility

model [25], Ait-Sahalia interest rate model [1] and so on [29, 39], which are SDEs with non-global Lipschitz coefficients. As an active research field, a great amount of work has been performed and a wide literature may be found on the numerical approximation for the SDEs under non-global Lipschitz conditions. For example, Chassagneux et al. [10] presented a modified explicit Euler-Maruyama discretization scheme for approximating the one-dimensional SDEs with non-Lipschitz drift or diffusion coefficients. Zhang et al. [43] showed an explicit balanced scheme that is order-preserving for SDEs with superlinear growth coefficients, and deduced the strong convergence rate of the corresponding scheme. Andersson et al. [3] first proved a strong convergence rate for a multistep method applied to equations with superlinearly growing drift and diffusion coefficient functions. Beyn et al. [7, 8] analyzed the mean-square convergence error of Euler-type and Milstein-type schemes using the concepts of "C-stability" and "B-consistency" for SDEs with a global monotonicity condition. Zong et al. [44] studied the convergence and stability of the two classes of theta-Milstein scheme for SDEs with non-global Lipschitz continuous coefficients. Kumar et al. [24] proposed a new explicit Milstein schemes for SDEs with superlinearly growing drift and diffusion coefficients and derived its optimal convergence in L^p sense. Gan et al. [15] introduced a family of explicit tamed stochastic Runge-Kutta methods for commutative SDEs with superlinearly growing drift and diffusion coefficients. Kelly et al. [22] developed an explicit adaptive Milstein method for SDEs with no commutativity condition and proved that the scheme is strongly convergent of order one. Wang et al. [39] investigated a class of implicit Milstein-type methods for SDEs with non-globally Lipschitz drift and diffusion coefficients, and offered upper mean-square error bounds for the proposed schemes under certain assumptions. Of course, the works mentioned above are not, by far, an exhaustive list of research touching the subject. We refer to the papers and manuscripts in [2, 11, 16, 18, 19, 21, 30–33, 35, 38, 41, 42] and references therein as a complementary for the relevant respect. Although great progress has been made in the analysis and calculation of numerical discretization schemes for SDEs with non-global Lipschitz coefficients, not much work has been done on numerically solving SDEs using stochastic neural networks. In this regard, Raja et al. [4] respectively used Adams method and Kloeden-Platen-Schurz methods to numerically solve the deterministic and stochastic vector-borne plant epidemic model. Additionally, Raja et al. [5, 6] employed separately a novel neuro-stochastic adaptive processing as well as stochastic predictive computing networks to investigate the dynamics of the SIS (Susceptible-Infectious-Susceptible) with a vaccination impact-based epidemic model represented by nonlinear SDEs.

The purpose of this work is to attempt to develop a data-driven approach called extended continuous latent process flows (ECLPF) to numerically approximate the SDEs with non-global Lipschitz drift coefficients and multiplicative noise. To the best of our knowledge, there has been no work on the numerical study of SDEs with non-global Lipschitz coefficients using this ECLPF network. The basic construction of this new method is inspired by the classical CLPF approach proposed in paper [13], which is a typical generative time series model and has shown remarkable potential in enhancing the representation power and variational inference quality of the existing time series models. It is worth mentioning that in paper [13], researchers employed a variational approximation based on the piecewise construction of the posterior latent process, but at the expense of a lot of time in actual calculations. In the current work, the variational posterior process on the ECLPF is constructed using a neural SDE (NSDE) conditioned on the information over the whole time interval, which has the advantage of reducing computational cost. In particular, such posterior NSDE is related to the

derivation of the variational lower bound [37] that will be used together with the stochastic adjoint sensitivity method, originally proposed by Li et al. [26], to maximize the log-likelihood function induced by partial observations. Besides an efficient network architecture, the initial input data is supposed to be provided as accurately as possible to better numerically approximate the underlying problem. In this work, we will make use of a temporal discretization method called drift-θ Milstein scheme [39, 44] to compute the initial inputs of ECLPF.

The main contributions/novelties of this paper are summarized as follows:

• The well-posedness of the underlying problem is established. That is, the existence, uniqueness, and the stability of the solution are provided.

• The structure of ECLPF consists of three major components: (i) a prior SDE that acts as a variational autoencoder to describe the continuous latent dynamics of the observed data, (ii) a continuously normalizing flow serving as a time-dependent decoder, and (iii) a posterior process conditional on the information over the entire time interval, which is used to derive the variational lower bound so that the maximum of the observed log-likelihood function can be approximated conveniently. Notably, the network architecture of our new method is different from that of the classic CLPF in paper [13], which we will explain in detail in Section 3. The approximation behavior of the ECLPF will be presented through a series of numerical experiments to demonstrate the effectiveness of our proposed method.

The rest of the paper is organized as follows. In Section 2, we establish the well-posedness of the considered problem under certain standard assumptions and introduce the drift-θ Milstein scheme for time discretization. The architecture of the ECLPF as well as the training and inference procedures are described in Section 3. In Section 4, several numerical experiments are provided to demonstrate the approximation and generalization capabilities of the ECLPF method.

2. Problem description and temporal discretization

In this section, we first introduce the problem to be studied and state the main assumptions on the well-posedness of the considered model, then a time-stepping scheme called the drift-θ Milstein method is made use of to discretize the underlying problem.

Let $T \in (0, \infty)$, $d, m \in \mathbb{N}$, and $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t\in[0,T]}, \mathbb{P})$ be a filtered probability space with a normal filtration $\{\mathcal{F}_t\}_{t\in[0,T]}$. Denote by $W^k(t) : [0, T] \times \Omega \to \mathbb{R}, k \in \{1, \cdots, m\}$, an independent family of real-valued standard $\{\mathcal{F}_t\}_{t\in[0,T]}$-adapted Brownian motions on probability space $(\Omega, \mathcal{F}, \mathbb{P})$. We are interested in the problem written as the following SDE: For $u(t, \omega) : [0, T] \times \Omega \to \mathbb{R}^d$,

$$\begin{aligned} du(t) &= f(u(t))dt + \sum_{k=1}^{m} g_k(u(t))dW^k(t),\ 0 < t \le T, \\ u(0) &= u_0, \end{aligned} \tag{2.1}$$

where $f : \mathbb{R}^d \to \mathbb{R}^d$ represents the drift coefficient function and $g_k : \mathbb{R}^d \to \mathbb{R}^d, k \in \{1, \cdots, m\}$, stands for the diffusion coefficient function. Let $\langle \cdot, \cdot \rangle$ be the Euclidean inner product, and the corresponding norm on $\mathbb{R}^d$ is $|\cdot|$. Assume the noise term appearing in (2.1) to be commutative [28, 40] and the initial condition $u_0 \in L^p(\Omega; \mathbb{R}^d)$ for $p \in [2, \infty)$. Throughout the paper, we use c to mean generic positive constants, which may not be the same at different occurrences.

In order to establish the well-posedness of the problem (2.1), some assumptions imposed on the drift and the diffusion coefficient functions are needed, which are collected below:

Suppose the coefficient functions f and $g_k, k \in \{1, \cdots, m\}$, are first-order continuously Fréchet differentiable, and there exist constants $c > 0$ such that: For all $v_1, v_2 \in \mathbb{R}^d$,

$$\langle v_1 - v_2, f(v_1) - f(v_2)\rangle \leq c|v_1 - v_2|^2, \tag{2.2}$$

$$\sum_{k=1}^{m} |g_k(v_1) - g_k(v_2)|^2 \leq c|v_1 - v_2|^2. \tag{2.3}$$

Note that according to (2.2) and (2.3), one deduces readily that there exists a constant c such that for every $k \in \{1, \cdots, m\}$,

$$\langle v, f(v)\rangle \vee |g_k(v)|^2 \leq c(1 + |v|^2), \quad \forall v \in \mathbb{R}^d, \tag{2.4}$$

where $a \vee b$ represents $\max\{a, b\}$.

Remark 2.1. *Assumption* (2.2) *on the drift f is known as the one-sided Lipschitz condition. Assumption* (2.3) *on the diffusions $\{g_k\}_{k=1,\cdots,m}$ is the global Lipschitz condition, and assumption* (2.4) *is the one-sided linear growth condition. These assumptions are often applied in proving the well-posedness of SDEs with non-global Lipschitz drift terms; see, e.g., [29, 44].*

We are now in a position to establish the well-posedness of the problem (2.1). Let $\mathcal{H}_T^2$ be the $\mathbb{R}^d$-valued predictable processes $\{v(t) : t \in [0, T]\}$ such that

$$\|v\|_{\mathcal{H}_T^2} := \sup_{0\leq t\leq T} \mathbb{E}[|v|^2]^{\frac{1}{2}} < \infty$$

with $\mathbb{E}[\cdot]$ being the expectation on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$. According to the known result reported on [29, Theorem 3.6], there exists a unique solution $u(t) \in \mathcal{H}_T^2$ to the problem (2.1). Precisely, there exists an $\{\mathcal{F}_t\}_{t\in[0,T]}$-adapted stochastic process $u(t, \omega) : [0, T] \times \Omega \to \mathbb{R}^d$, which satisfies $\mathbb{P}$-almost surely the integral equation

$$u(t) = u_0 + \int_0^t f(u(\tau))d\tau + \sum_{k=1}^{m} \int_0^t g_k(u(\tau))dW^k(\tau). \tag{2.5}$$

Furthermore, by [17, Lemma 3.2], one can obtain the following stability inequality for $u(t)$:

$$\sup_{0\leq t\leq T} \mathbb{E}[|u(t)|^p] \leq c(1 + \mathbb{E}[|u_0|^p]), \quad p \in [2, \infty).$$

Next, we focus on the time discretization to the problem (2.1). The numerical approach used in the current work is called the drift-θ Milstein scheme that has been applied to solve SDEs, see, e.g., [39,44] and references therein. However, this paper appears to be the first attempt to combine this temporal discretization with a CLPF-type network to numerically approximate SDEs. Below we briefly describe this temporal scheme.

Define time step $\Delta t := T/N$, with N being a positive integer. For a fixed $k \in \{1, \cdots, m\}$, let $g_k(\boldsymbol{x}) := [g_{1k}(\boldsymbol{x}), \cdots, g_{dk}(\boldsymbol{x})]^T \in \mathbb{R}^d$, $\forall \boldsymbol{x} = [x_1, \cdots, x_d]^T \in \mathbb{R}^d$. Denote

$$L^j g_k(\boldsymbol{x}) := \sum_{q=1}^{d} g_{q,j}(\boldsymbol{x}) \frac{\partial g_k(\boldsymbol{x})}{\partial x_q}, \quad j, k \in \{1, \cdots, m\}.$$

The drift-θ Milstein scheme reads: Find a $\mathbb{R}^d$-valued approximation $\{y_i\}_{i=1,\cdots,N}$, such that for $\theta \in (\frac{1}{2}, 1]$,

$$\begin{aligned} y_{i+1} &= y_i + (\theta f(y_{i+1}) + (1-\theta) f(y_i))\Delta t + g(y_i)\Delta W_i + \frac{1}{2}\sum_{j,k=1}^{m} L^j g_k(y_i)(\Delta W_i^j \Delta W_i^k - \delta_{kj}\Delta t), \\ y_0 &= u_0, \end{aligned} \tag{2.6}$$

where y_i is the approximation to $u(t_i)$, $\Delta W_i = W((i+1)\Delta t) - W(i\Delta t)$, $\Delta W_i^j = W^j((i+1)\Delta t) - W^j(i\Delta t)$, $\delta_{kj} = 1$ for $k = j$, and $\delta_{kj} = 0$ for $k \neq j$.

Remark 2.2. *Since the problem* (2.1) *has the commutative noise, the scheme* (2.6) *does not involve the Lévy area [27, 28], thus avoiding extra computational effort. Furthermore, as an implicit scheme, the well-posedness of the scheme* (2.6) *is straightforward to establish; see, e.g., [44] for details. In particular, when* $\theta = 1$, (2.6) *becomes the drift-implicit Milstein scheme investigated in [35].*

We further assume that $f, \{g_k\}_{k=1,\cdots,m}$ are twice continuously Fréchet differentiable and there exists a constant c such that for any $v, v_1, v_2 \in \mathbb{R}^d$ and $j, k \in \{1, \cdots, m\}$,

$$\begin{aligned} &|L^j g_k(v_1) - L^j g_k(v_2)|^2 \leq c|v_1 - v_2|^2, \\ &|f'(v)| \vee |f''(v)| \leq c(1 + |v|^q), \quad q \geq 2, \\ &|g_k''(v)| \leq c. \end{aligned}$$

It is known from [44, Theorem 4.1] that the strong convergence rate of the numerical approximation y_i to the solution $u(t_i)$ satisfies: For $i = 1, \cdots, N$,

$$\mathbb{E}[|y_i - u(t_i)|^2]^{\frac{1}{2}} \leq c\Delta t. \tag{2.7}$$

Here we show this convergence rate intuitively through a numerical example. For instance, set $d = m = 2$, $u = [u_1, u_2]^T$, $f(u) = [u_1, -u_2]^T$, $g_1(u) = [u_1, 0]^T$, $g_2(u) = [0, 2u_2]^T$, $u_0 = [1, 1]^T$. The strong convergence rate is measured in terms of mean-square approximation errors at the endpoint $T = 1$. In actual calculations, we will use the reference solution computed in the fine time mesh size as the exact solution. To be specific, we take the reference solution calculated by $\Delta t = 2 \times 10^{-4}$ for the temporal accuracy test. Additionally, we employ the mean of 3000 samples to approximate the expected value of the numerical error, i.e., the strong convergence error $\mathbb{E}[|y_N - u(T)|^2]^{\frac{1}{2}}$ is computed approximately by

$$\mathbb{E}[|y_N - u(T)|^2]^{\frac{1}{2}} \approx \Big(\frac{1}{3000}\sum_{j=1}^{3000} |u_{\text{ref}}^j - y_N^j|^2\Big)^{\frac{1}{2}} =: u_{\text{error}},$$

where u_{ref}^j and y_N^j denote separately the reference solution and the numerical solution of the j-th sample. Under different θ, we compute u_{error} with different time steps $\Delta t = 8\times10^{-3}, 4\times10^{-3}, 2\times10^{-3}, 1\times10^{-3}$. The error behavior is presented in Figure 1, also given in Table 1 for specific convergence rate data, from which we see that the strong convergence rate for scheme (2.6) with different θ is consistent with the theoretical first-order convergence result shown in (2.7).

Note that the scheme (2.6) provides a convenient way to generate an observed time series associated with the solution $u(t)$, which we will use as the initial input to the ECLPF network introduced in the next section.

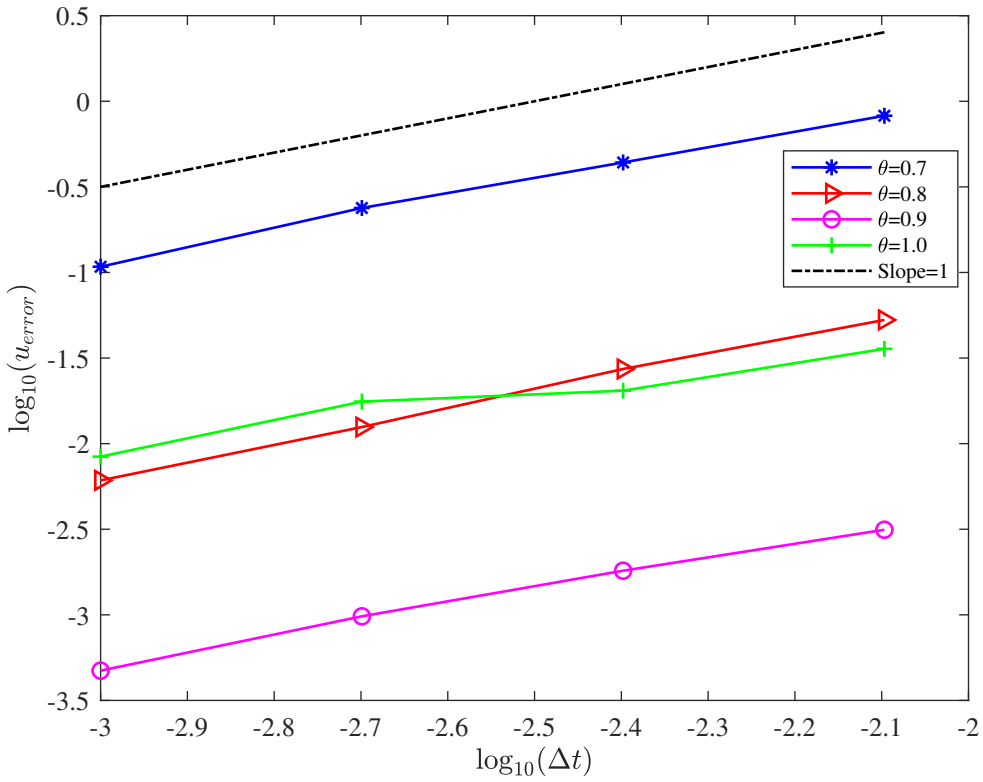


Figure 1. Temporal convergence rate test of scheme (2.6) on log-log scale.

Table 1. Strong error and convergence rate with different θ.

Time step Δt	$\theta = 0.7$		$\theta = 0.8$		$\theta = 0.9$		$\theta = 1.0$	
	u_{error}	Order	u_{error}	Order	u_{error}	Order	u_{error}	Order
8.00E-3	8.23E-1	–	5.28E-2	–	3.14E-2	–	3.58E-2	–
4.00E-3	4.39E-1	0.91	2.73E-2	0.95	1.81E-2	0.79	2.04E-2	0.81
2.00E-3	2.38E-1	0.88	1.25E-2	1.13	9.80E-3	0.89	1.76E-2	1.02
1.00E-3	1.08E-1	1.14	6.10E-3	1.04	4.72E-3	1.05	8.40E-3	1.07

3. Extended CLPF

This section is devoted to describing the architecture of the ECLPF. It consists of three parts: (i) a prior SDE, (ii) a normalizing flow, and (iii) a posterior SDE, which will be presented below individually. Before that, we first denote by $\{\tilde{u}_{t_i}\}_{i=1,\cdots,N}$ a sequence of d-dimensional observations sampled at time t_i. These observations are related to the solution of the problem (2.1) and are assumed to be partial realizations of a stochastic process $X(t) \in \mathbb{R}^d$ that we use to approximate $u(t)$. Our training goal is to maximize the observational log-likelihood function defined by

$$\mathcal{L} = \log \mathbb{P}(\tilde{u}_{t_1}, \cdots, \tilde{u}_{t_N}), \tag{3.1}$$

where $\mathbb{P}(\tilde{u}_{t_1}, \cdots, \tilde{u}_{t_N})$ represents the probability of observing $\{\tilde{u}_{t_i}\}_{i=1,\cdots,N}$.

3.1. Latent dynamics and decoding

We use an NSDE driven by m-dimensional Wiener process $W(t)$ to model the evolution of an d-dimensional time-continuous latent state $Z(t)$. Specifically, $Z(t)$ satisfies:

$$\begin{aligned} dZ(t) &= \mu_\gamma(Z(t))dt + \sigma_\gamma(Z(t))dW(t), \ \ 0 < t \le T, \\ Z(0) &= z_0, \ \ z_0 \in \mathbb{R}^d, \end{aligned} \tag{3.2}$$

where γ stands for the learnable parameters of the drift $\mu_\gamma \in \mathbb{R}^d$ and diffusion $\sigma_\gamma \in \mathbb{R}^{d\times m}$. Notice that the system (3.2) is usually called a prior SDE, and μ_γ as well as σ_γ can be implemented by using deep neural networks; see, e.g., [13, 23, 26] for more details. Motivated by [13], the latent process $Z(t)$ is commonly decoded into a continuous trajectory of the stochastic process $X(t)$ by exploiting the dynamic normalizing flows [9, 14]. To be precise, $X(t)$ can be modeled by

$$\begin{aligned} X(t) &= F_\xi(O(t), Z(t)), \\ X(0) &= x_0, \ \ x_0 \in \mathbb{R}^d, \end{aligned} \tag{3.3}$$

where $O(t)$ is the d-dimensional base process generated by Ornstein-Uhlenbeck process, and it has closed-form transition density. $F_\xi(O(t), Z(t))$ is a normalizing flow with parameter ξ for any $Z(t)$, which plays an important role in decoding each sample path of $Z(t)$ into a trajectory satisfying the distribution on $X(t)$. Let $\{z_{t_i}\}_{i=1,\cdots,N}$ be the samples of $\{Z(t_i)\}_{i=1,\cdots,N}$.

3.2. Training and inference

With the model specified, we are now in a position to focus on the training and inference. It involves the calculation and optimization of the observational log-likelihood function (3.1) induced by a time-dependent decoding of (3.3), which seems to be subtle and challenging since only few SDEs have closed-form transition densities. Here we resort to an appropriate method of approximating the log-likelihood (3.1) based on the variational posterior latent process conditioned on all information in the interval $[0, T]$. Below we present this method in detail.

The log-likelihood function $\mathcal{L}$ can be expressed as a conditional expectation with respect to the latent process $Z(t)$, i.e., we have

$$\mathcal{L} = \log \mathbb{P}(\tilde{u}_{t_1}, \cdots, \tilde{u}_{t_N}) = \log \mathbb{E}[\mathbb{P}(\tilde{u}_{t_1}, \cdots, \tilde{u}_{t_N}|Z(t))], \ \ 0 \le t \le T,$$

where $\mathbb{P}(\tilde{u}_{t_1}, \cdots, \tilde{u}_{t_N}|Z(t))$ denotes the probability of observing $\{\tilde{u}_{t_i}\}_{i=1,\cdots,N}$ conditioned on the trajectory of the latent process $Z(t)$ in the interval $[0, T]$. Inspired by [26], we then use the variational bound to approximate $\mathcal{L}$, for which we need to introduce a posterior SDE defined by: For $\mu_\phi \in \mathbb{R}^d$,

$$\begin{aligned} d\tilde{Z}(t) &= \mu_\phi(\tilde{Z}(t))dt + \sigma_\gamma(\tilde{Z}(t))dW(t), \ \ 0 < t \le T, \\ \tilde{Z}(0) &= \tilde{z}_0, \ \ \tilde{z}_0 \in \mathbb{R}^d, \end{aligned} \tag{3.4}$$

where, except for drift μ_ϕ, $W(t)$ and diffusion σ_γ are the same as those shown in (3.2). Note that the parameter ϕ is the output of an encoder Long Short-Term Memory (LSTM) that takes the sequence of

observations $\{\tilde{u}_{t_i}\}_{i=1,\cdots,N}$ and the sampled latent states $\{z_{t_i}\}_{i=1,\cdots,N}$ as inputs. The system (3.4) is called the posterior SDE, which plays an important role in variational approximation; see, e.g., [37] and references therein.

Suppose there exists a mapping $h(v) : \mathbb{R}^d \to \mathbb{R}^m,\ \forall v \in \mathbb{R}^d$, such that $\sigma_\gamma(\cdot)h(\cdot) = \mu_\phi(\cdot) - \mu_\gamma(\cdot)$ and h satisfies Novikov's condition [36, Chapter 8.6], then we obtain: For $t \in [0, T]$,

$$\mathcal{L} = \log \mathbb{P}(\tilde{u}_{t_1}, \cdots, \tilde{u}_{t_N}) = \log \mathbb{E}[\mathbb{P}(\tilde{u}_{t_1}, \cdots, \tilde{u}_{t_N}|Z(t))] = \log \mathbb{E}[\mathbb{P}(\tilde{u}_{t_1}, \cdots, \tilde{u}_{t_N}|\tilde{Z}(t))M_T]$$

with $M_T := \exp\big(-\int_0^T \frac{1}{2}|h(\tilde{Z}(s))|^2 ds - \int_0^T h(\tilde{Z}(s))^T dW(s)\big)$ representing a weighting between (3.2) and (3.4); see, e.g., [26] for details. Furthermore, applying Jensen's inequality yields:

$$\begin{aligned} \mathcal{L} = \log \mathbb{E}[\mathbb{P}(\tilde{u}_{t_1}, \cdots, \tilde{u}_{t_N}|\tilde{Z}(t))M_T] &\geq \mathbb{E}[\log\big(\mathbb{P}(\tilde{u}_{t_1}, \cdots, \tilde{u}_{t_N}|\tilde{Z}(t))M_T\big)] \\ &\geq \mathbb{E}[\log M_T + \log \mathbb{P}(\tilde{u}_{t_1}, \cdots, \tilde{u}_{t_N}|\tilde{Z}(t))], \end{aligned} \tag{3.5}$$

where $\mathbb{E}[\log M_T + \log \mathbb{P}(\tilde{u}_{t_1}, \cdots, \tilde{u}_{t_N}|\tilde{Z}(t))]$ is known as an evidence lower bound [37]. One more thing to mention is that Li et al. [26] has successfully realized the optimization of the log-likelihood function $\mathcal{L}$ through this lower bound using the adjoint sensitivity method, and we will also use this adjoint approach for calculations in the current work.

Note that parameter ϕ is a function of latent states $\{z_{t_i}\}_{i=1,\cdots,N}$, and $\{z_{t_i}\}_{i=1,\cdots,N}$ is a function of parameter γ, so the parameter ϕ can also be characterized by parameter γ. In summary, we implement the relevant calculation in the following steps:

i) Use the adjoint sensitivity method to calculate the gradient of the expected value $\mathbb{E}[\log M_T + \log \mathbb{P}(\tilde{u}_{t_1}, \cdots, \tilde{u}_{t_N}|\tilde{Z}(t))]$ on the right side of (3.5) (i.e., approximately computing the maximum value of the log-likelihood function $\mathcal{L}$), thereby obtaining the parameter γ;

ii) Compute latent state $Z(t)$ by (3.2);

iii) Employ dynamic normalizing flow to compute $X(t)$ appeared in (3.3), which will be used as the approximation to the solution $u(t)$ shown in (2.5).

Remark 3.1. *Our ECLPF can actually be viewed as a hybridization of the classical CLPF [13] and latent SDE [26], but it is different from both. In fact, compared with the latent SDE [26], our ECLPF additionally adds a normalizing flow structure* (3.3)*, which helps to improve the stability in computation; see, e.g., [13, Section 3.1]. Moreover, compared to the classical CLPF model [13], instead of constructing a piecewise posterior process, our ECLPF model uses a variational posterior process based on all information in the interval* $[0, T]$ *to derive the variational lower bound, which reduces the computational cost.*

4. Numerical experiments

In this section, we take several SDEs with non-global Lipschitz nonlinearity as test examples to show the effectiveness of the ECLPF method. More details on hyperparameter settings and implementation of the ECLPF are motivated by papers [13, 26]. Our code is available at https://github.com/JHUNAI/ECLPF.

Example 4.1. The first example is called the stochastic Ginzburg-Landau equation, which is often used to describe phase transition [20] and is expressed as:

$$\begin{aligned} du(t) &= \frac{u-u^3}{0.01}dt + \frac{u}{2}dW(t),\ 0 < t \le T, \\ u(0) &= u_0 = 1. \end{aligned} \tag{4.1}$$

To begin, we concentrate on the strong and weak errors at endpoint $t = T$. These two types of errors are, respectively, presented as follows:

$$\mathbb{E}[|u(T) - X(T)|^2]^{\frac{1}{2}} =: e_{\text{strong}}, \qquad \Big|\mathbb{E}[u(T)] - \mathbb{E}[X(T)]\Big| =: e_{\text{weak}}, \tag{4.2}$$

where $u(T)$ and $X(T)$ denote separately the exact solution of the problem (4.1) and the solution generated by ECLPF method. In actual calculation, since it is difficult to get the explicit exact solution, we will use the reference solution calculated at fine time steps $\Delta t_{\text{ref}} = 5 \times 10^{-4}$ instead of the exact solution for error measurement. The observed input data $\{\tilde{u}_{t_i}\}_{i=1,\cdots,N}$ is obtained through the calculation of the scheme (2.6) with $\Delta t = \frac{T}{N} = 0.01$ (i.e., $N = 100$). Additionally, we employ the mean of $M \in \mathbb{N}_+$ samples to approximate the expected value of the numerical errors, i.e., the strong error e_{strong} and the weak error e_{weak} can be computed approximately by:

$$\begin{aligned} e_{\text{strong}} &\approx \Big(\frac{1}{M}\sum_{j=1}^{M} |u_j^{\text{ref}} - X_j(T)|^2\Big)^{\frac{1}{2}} =: u_{\text{strong}}, \\ e_{\text{weak}} &\approx \Big|\frac{1}{M}\sum_{j=1}^{M} (u_j^{\text{ref}} - X_j(T))\Big| =: u_{\text{weak}}, \end{aligned}$$

where u_j^{ref} represents the j-th sample of the reference solution, and $X_j(T)$ stands for the j-th sample of $X(T)$. Under different values of θ, we compute u_{strong} and u_{weak} with different M by taking $T = 1$ and compare these errors to those computed by scheme (2.6). The error behavior is shown in Table 2, from which we see that the strong and weak errors of the ECLPF method are smaller than those of the scheme (2.6) for $\theta = 0.6, 0.8, 1.0$. It means that the proposed new method shows a better approximation effect to the equation (4.1) compared to the scheme (2.6).

Next, we pay attention to the fitting and generalization performance of the ECLPF method to the reference solution. To be specific, we randomly select one sample path with $\theta = 1$ to simulate the fitting effect, as shown in (a) in Figure 2, where we observe that the time evolution trajectory of the ECLPF solution is basically able to fit the reference solution. We then investigate the generalization performance of the ECLPF method on the time interval $[T, 2T]$. Precisely, we take the mean of 10 sample paths of the reference solution and the ECLPF solution, and let

$$\bar{u}^{\text{ref}}(T + i\Delta t) := \frac{1}{10}\sum_{j=1}^{10} u_j^{\text{ref}}(T + i\Delta t), \ \ \bar{X}(T + i\Delta t) := \frac{1}{10}\sum_{j=1}^{10} X_j(T + i\Delta t), \ i = 1, \cdots, N,$$

where $u_j^{\text{ref}}(T + i\Delta t)$ and $X_j(T + i\Delta t)$ separately represent the j-th sample of the reference solution and the ECLPF solution at time $T + i\Delta t$. We calculate the generalization relative error e_g by

$$e_g := \frac{\Big(\sum_{i=1}^{N} (\bar{u}^{\text{ref}}(T + i\Delta t) - \bar{X}(T + i\Delta t))^2 \Delta t\Big)^{\frac{1}{2}}}{\big(\sum_{i=1}^{N} \bar{u}^{\text{ref}}(T + i\Delta t)^2 \Delta t\big)^{\frac{1}{2}}} \approx 0.01633. \tag{4.3}$$

This is a number less than 5%, indicating that our new method shows acceptable generalization performance. Furthermore, we plot the evolution trajectories of the sample mean of the reference solution and the ECLPF solution in (b) in Figure 2, from which we see that the proposed new method exhibits effective generalization behavior.

Example 4.2. In this example, we consider the Duffing-van der Pol system as a test model, which reads:

$$\begin{aligned} du_1(t) &= u_2(t)dt + u_2(t)dW(t), \\ du_2(t) &= \big(- u_2(1 + u_1^2) + (u_1 - u_1^3)\big)dt + u_1 dW(t), \end{aligned} \tag{4.4}$$

where the term $-u_2(1 + u_1^2)$ stands for nonlinear dissipation and the term $(u_1 - u_1^3)$ denotes potential conservative force. This model is typically used to simulate the unstable limit cycles in two-dimensional phase space; see, e.g., [12] for more detail.

Let $\boldsymbol{u}(t) := [u_1(t), u_2(t)]^T$. Denote the ECLPF solution by $\boldsymbol{X}(t) := [X_1(t), X_2(t)]^T$ with $X_k(t)$, $k = 1, 2$, being the k-th component. Like the Example 4.1, we use the reference solution instead of exact solution to approximately compute the strong error e_{strong} and weak error e_{weak} at endpoint $t = T$, as below:

$$e_{\text{strong}} \approx \Big(\frac{1}{M}\sum_{j=1}^{M}\big(|u_{1,j}^{\text{ref}} - X_{1,j}(T)|^2 + |u_{2,j}^{\text{ref}} - X_{2,j}(T)|^2\big)\Big)^{\frac{1}{2}} =: \boldsymbol{u}_{\text{strong}},$$

$$e_{\text{weak}} \approx \frac{1}{M}\sum_{j=1}^{M}\big(|u_{1,j}^{\text{ref}} - X_{1,j}(T)| + |u_{2,j}^{\text{ref}} - X_{2,j}(T)|\big) =: \boldsymbol{u}_{\text{weak}},$$

where $u_{k,j}^{\text{ref}}$ and $X_{k,j}(T)$, $k = 1, 2$, respectively represent the j-th sample of the k-th component of the reference solution and the CLPF solution. Set $\theta = 1$, $T = 1$, and the initial value $\boldsymbol{u}(0) = [u_1(0), u_2(0)]^T = [\frac{1}{2}, 0]^T$. We calculate $\boldsymbol{u}_{\text{strong}}$ and $\boldsymbol{u}_{\text{weak}}$ with different sample numbers and compare these errors with those computed by scheme (2.6). The numerical results are listed in Table 3, from which we find that the strong and weak errors of the ECLPF method are smaller than those of the scheme (2.6). This means that our proposed new method can approximate model (4.4) more accurately than scheme (2.6).

We next focus on the ability of the fitting and generalization of the ECLPF method to the model (4.4). Similar to the Example 4.1, we randomly take one sample path for fitting, and the numerical behavior of the time evolution of each component of the CLPF solution and the reference solution is shown in (a) and (b) of Figure 3, respectively. We observe that the ECLPF solution exhibits good fitting capabilities. Now, we consider the generalization performance of the ECLPF method on the time interval $[T, 2T]$. As like in Example 4.1, we first take the mean of 10 sample paths of the reference solution and the ECLPF solution, and define

$$\bar{\boldsymbol{u}}^{\text{ref}}(T + i\Delta t) := \frac{1}{10}\sum_{j=1}^{10}\boldsymbol{u}_j^{\text{ref}}(T + i\Delta t), \quad \bar{\boldsymbol{X}}(T + i\Delta t) := \frac{1}{10}\sum_{j=1}^{10}\boldsymbol{X}_j(T + i\Delta t), \quad i = 1, \cdots, N,$$

where $\boldsymbol{u}_j^{\text{ref}}(T + i\Delta t) := [u_{1,j}^{\text{ref}}(T + i\Delta t), u_{2,j}^{\text{ref}}(T + i\Delta t)]^T$ and $\boldsymbol{X}_j(T + i\Delta t) := [X_{1,j}(T + i\Delta t), X_{2,j}(T + i\Delta t)]^T$. Once again, we calculate the generalization relative error $\boldsymbol{e}_g$ by

$$\boldsymbol{e}_g := \frac{\Big(\sum_{i=1}^{N}\big\|\bar{\boldsymbol{u}}^{\text{ref}}(T + i\Delta t) - \bar{\boldsymbol{X}}(T + i\Delta t)\big\|_2^2\Delta t\Big)^{\frac{1}{2}}}{\big(\sum_{i=1}^{N}\big\|\bar{\boldsymbol{u}}^{\text{ref}}(T + i\Delta t)\big\|_2^2\Delta t\big)^{\frac{1}{2}}} \approx 0.03723 < 5\%, \tag{4.5}$$

which shows a satisfactory generalization performance of our ECLPF method. Moreover, we individually plot the time evolution of the sample mean of the reference solution and the ECLPF solution in (c) and (d) of Figure 3, from which we see that the proposed new method exhibits effective generalization behavior.

Example 4.3. In the last example, we will use an SDE with a Nagumo potential function, which is commonly used to model the dynamics of activation and deactivation of spiking neurons [28], to demonstrate the computational performance of our proposed ECLPF method. The specific test model is as follows:

$$\begin{aligned} du(t) &= u(1-u)(u+\frac{1}{2})dt + udW(t),\ 0 < t \le T, \\ u(0) &= u_0 = 1. \end{aligned} \tag{4.6}$$

It is first devoted to the comparison for the calculation efficiency of the ECLPF and the classical CLPF methods. The code of classical CLPF has been provided by Deng et al. [13]; see "https://github.com/BorealisAI/continuous-latent-process-flows" for details. All the solvers are implemented using Python. The numerical experiments are performed on a computer with 64GB memory and CPU (central processing unit) "AMD-7950X". Let $T = 1$. We employ, respectively, the ECLPF and CLPF methods to calculate the CPU time consumption under different sample numbers and time steps. The relevant numerical results are listed in Table 4, where the left one is the result with $\Delta t = 5 \times 10^{-4}$ and the right one is the result when the sample number is fixed at 1000. We see from this Table that the ECLPF method saves a lot of time compared with the classical CLPF method. Therefore, the computational cost of our proposed method is lower than the classical CLPF method.

Next, we use the calculation rules shown in Example 4.1 to compute the strong and weak errors of problem (4.6) again using the ECLPF method and the scheme (2.6) individually. The numerical errors are given in Table 5, where we find that the strong and weak errors of the ECLPF method are smaller than those computed by scheme (2.6). Therefore, the ECLPF method shows a better approximation effect to the underlying model (4.6) compared to the scheme (2.6).

Finally, we compute the generalization error of the ECLPF method on the time interval $[T, 2T]$. Similar to the Example 4.1, we take the mean of 10 sample paths of the reference solution and the ECLPF solution, and let

$$\bar{u}^{\text{ref}}(T + i\Delta t) := \frac{1}{10}\sum_{j=1}^{10} u_j^{\text{ref}}(T + i\Delta t),\ \ \bar{X}(T + i\Delta t) := \frac{1}{10}\sum_{j=1}^{10} X_j(T + i\Delta t),\ i = 1, \cdots, N,$$

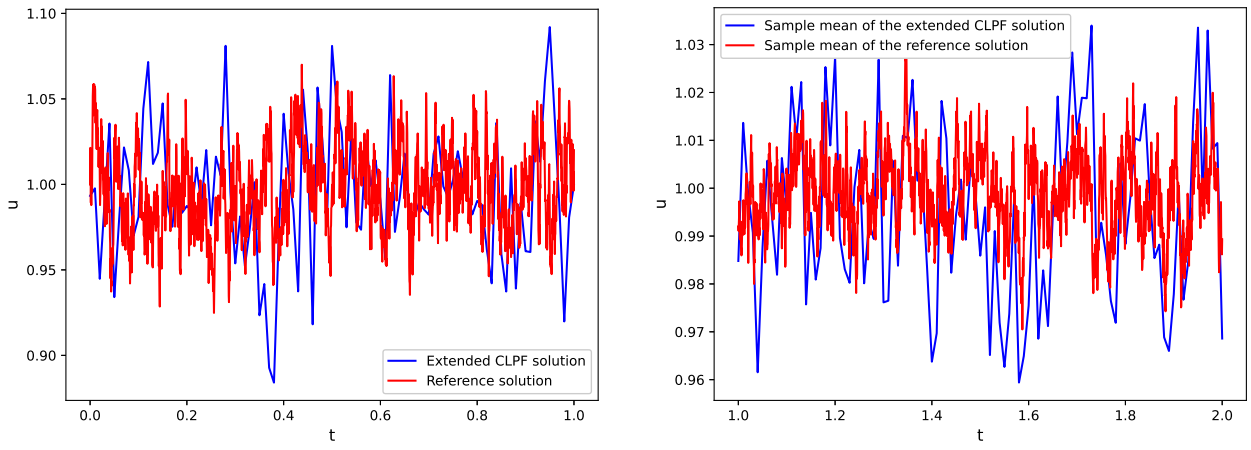
where $u_j^{\text{ref}}(T + i\Delta t)$ and $X_j(T + i\Delta t)$ separately represent the j-th sample of the reference solution and the ECLPF solution at time $T + i\Delta t$. The generalization relative error e_g satisfies:

$$e_g := \frac{\left(\sum_{i=1}^{N} (\bar{u}^{\text{ref}}(T + i\Delta t) - \bar{X}(T + i\Delta t))^2 \Delta t\right)^{\frac{1}{2}}}{\left(\sum_{i=1}^{N} \bar{u}^{\text{ref}}(T + i\Delta t)^2 \Delta t\right)^{\frac{1}{2}}} \approx 0.02997,$$

which is a small number less than 5%. Once again, it implies that our ECLPF method exhibits good generalization performance for the problem (4.6).

Table 2. Error comparison. Upper: $\theta = 0.6$; Middle: $\theta = 0.8$; Lower: $\theta = 1.0$.

θ-value	Sample number M	ECLPF method		Scheme (2.6)	
		u_{strong}	u_{weak}	u_{strong}	u_{weak}
$\theta = 0.6$	100	6.9862E-2	5.5620E-2	7.6799E-2	6.4412E-2
	200	6.9136E-2	5.3978E-2	7.8656E-2	6.3444E-2
	500	6.3581E-2	5.2564E-2	7.6496E-2	6.2053E-2
	800	6.3812E-2	5.1136E-2	7.5558E-2	6.1835E-2
	1000	6.3061E-2	5.0718E-2	7.5544E-2	6.1225E-2
$\theta = 0.8$	100	6.9843E-2	5.5602E-2	9.1942E-2	7.3036E-2
	200	6.9116E-2	5.3966E-2	9.0315E-2	7.2722E-2
	500	6.7042E-2	5.2549E-2	9.0001E-2	7.2137E-2
	800	6.3796E-2	5.1122E-2	8.8917E-2	7.1399E-2
	1000	6.3563E-2	5.0703E-2	8.8803E-2	7.0612E-2
$\theta = 1.0$	100	7.3940E-2	5.8470E-2	1.0304E-1	8.1380E-2
	200	6.9090E-2	5.5310E-2	1.0404E-1	8.0470E-2
	500	6.7710E-2	5.4340E-2	1.0275E-1	8.0880E-2
	800	6.6830E-2	5.4160E-2	1.0129E-1	7.9690E-2
	1000	6.5890E-2	5.3310E-2	1.0135E-1	7.9240E-2



(a) (b)

Figure 2. Fitting and generalization demonstration of the the ECLPF method. (a): Fitting performance. (b): Generalization performance.

Table 3. Strong and weak error comparison.

Sample number M	ECLPF method		Scheme (2.6)	
	$\boldsymbol{u}_{\text{strong}}$	$\boldsymbol{u}_{\text{weak}}$	$\boldsymbol{u}_{\text{strong}}$	$\boldsymbol{u}_{\text{weak}}$
100	4.948E-1	5.117E-1	5.682E-1	5.859E-1
200	5.282E-1	5.571E-1	6.384E-1	6.464E-1
500	4.998E-1	5.218E-1	5.913E-1	5.977E-1
800	4.600E-1	4.552E-1	5.960E-1	6.023E-1
1000	4.883E-1	4.687E-1	6.157E-1	6.202E-1

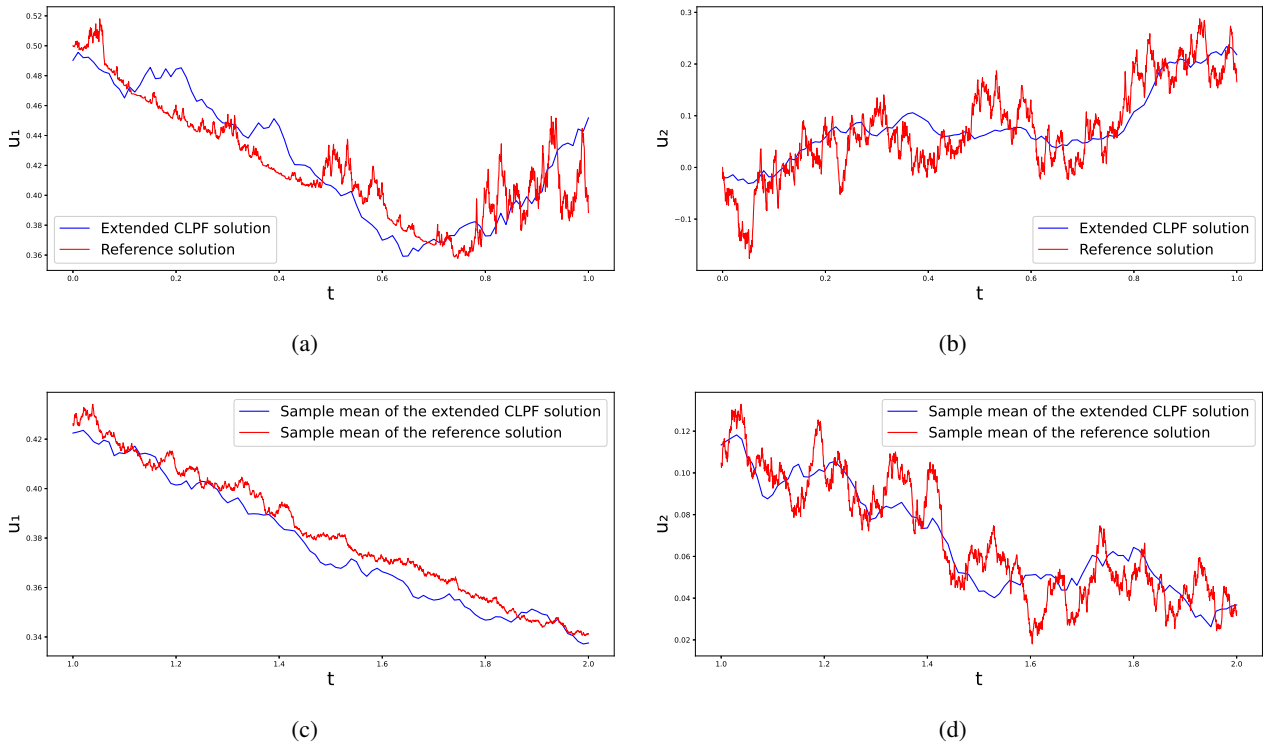


Figure 3. Fitting and generalization demonstration of the the ECLPF method. (a) and (b): Fitting performance of $u_1(t)$ and $u_2(t)$. (c) and (d): Generalization performance of sample mean of $u_1(t)$ and $u_2(t)$.

Table 4. CPU time under different sample numbers (left) and time steps (right).

Sample number M	ECLPF method	CLPF method	Time step Δt	ECLPF method	CLPF method
	CPU time(s)	CPU time(s)		CPU time(s)	CPU time(s)
100	0.62	16.59	1/100	0.065	7.17
200	0.63	30.97	1/200	0.12	13.65
500	0.64	74.92	1/500	0.29	34.53
800	0.66	120.11	1/800	0.46	55.28
1000	0.69	140.65	1/1000	0.61	70.73

Table 5. Strong and weak error comparison.

Sample number M	ECLPF method		Scheme (2.6)	
	u_{strong}	u_{weak}	u_{strong}	u_{weak}
100	3.8262E-1	2.9902E-1	5.8525E-1	4.4244E-1
200	3.5462E-1	2.6334E-1	6.2152E-1	4.7477E-1
500	3.0238E-1	2.3591E-1	5.8920E-1	4.4990E-1
800	3.1958E-1	2.5090E-1	6.1063E-1	4.6905E-1
1000	3.0220E-1	2.2940E-1	6.2339E-1	4.8226E-1

5. Conclusions

This paper studied a data-driven approximation for the SDE driven by multiplicative noise and non-global Lipschitz drift coefficient. The well-posedness of the underlying equation was elaborated. An efficient neural network method called the ECLPF approach was proposed. Some typical numerical experiments are provided, including strong/weak error calculations, fitting performance simulations, and generalization performance comparisons, to illustrate the effectiveness of this new method.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant No. 71601085, the Key Research and Development Project of Hubei Province, China under Grant 2020BCA084, and the Ministry of Education Humanities and Social Sciences Research Project in China under Grant No.23YJCZH062.

Conflict of interest

The authors declare no competing interests.

References

1. Y. Ait-Sahalia, Testing continuous-time models of the spot interest rate, *Rev. Financ. Stud.,* **9** (1996), 385–426. https://doi.org/10.1093/rfs/9.2.385
2. A. Alfonsi, Strong order one convergence of a drift implicit Euler scheme: Application to the CIR process, *Stat. Probabil. Lett.,* **83** (2013), 602–607. https://doi.org/10.1016/j.spl.2012.10.034
3. A. Andersson, R. Kruse, Mean-square convergence of the BDF2-Maruyama and backward Euler schemes for SDE satisfying a global monotonicity condition, *BIT,* **57** (2017), 21–53. https://doi.org/10.1007/s10543-016-0624-y

4. N. Anwar, I. Ahmad, A. K. Kiani, M. Shoaib, M. A. Z. Raja, Euler-Maruyama and Kloeden-Platen-Schurz computing paradigm for stochastic vector-borne plant epidemic model, *Waves Random Complex,* (2023), 1–23. https://doi.org/10.1080/17455030.2022.2152908

5. N. Anwar, I. Ahmad, A. K. Kiani, M. Shoaib, M. A. Z. Raja, Novel intelligent predictive networks for analysis of chaos in stochastic differential SIS epidemic model with vaccination impact, *Math. Comput. Simul.,* **219** (2024), 251–283. https://doi.org/10.1016/j.matcom.2023.12.024

6. N. Anwar, I. Ahmad, A. K. Kiani, M. Shoaib, M. A. Z. Raja, Novel neuro-stochastic adaptive supervised learning for numerical treatment of nonlinear epidemic delay differential system with impact of double diseases, *Int. J. Model. Simul.,* (2024), 1–23. https://doi.org/10.1080/02286203.2024.2303577

7. W. Beyn, E. Isaak, R. Kruse, Stochastic C-stability and B-consistency of explicit and implicit Euler-type schemes, *J. Sci. Comput.,* **67** (2016), 955–987. https://doi.org/10.1007/s10915-015-0114-4

8. W. Beyn, E. Isaak, R. Kruse, Stochastic C-stability and B-consistency of explicit and implicit Milstein-type schemes, *J. Sci. Comput.,* **70** (2017), 1042–1077. https://doi.org/10.1007/s10915-016-0290-x

9. A. Caterini, R. Cornish, D. Sejdinovic, A. Doucet, Variational inference with continuouslyindexed normalizing flows, *Uncertainty in Artificial Intelligence*, pages 44–53, PMLR, 2021. Available from: https://proceedings.mlr.press/v161/caterini21a.html

10. J. Chassagneux, A. Jacquier, I. Mihaylov, An explicit Euler scheme with strong rate of convergence for financial SDEs with non-Lipschitz coefficients, *SIAM J. Financ. Math.,* **7** (2016), 993–1021. https://doi.org/10.1137/15M1017788

11. J. Cui, J. Hong, D. Sheng, Convergence in density of splitting AVF scheme for stochastic Langevin equation, *arXiv preprint arXiv:1906.03439,* (2019). https://doi.org/10.48550/arXiv.1906.03439

12. M. B. Dadfar, J. Geer, C. M. Andersen, Perturbation analysis of the limit cycle of the free van der Pol equation, *SIAM J. Appl. Math.,* **44** (1984), 881–895. https://doi.org/10.1137/0144063

13. R. Deng, M. A. Brubaker, G. Mori, A. Lehrmann, Continuous latent process flows, *Adv. Neural Inf. Process. Syst.*, **34** (2021), 5162–5173. Available from: https://proceedings.neurips.cc/paper/2021/hash/2983e3047c0c730d3b7c02258471 7f3f-Abstract.html

14. R. Deng, B. Chang, M. A. Brubaker, G. Mori, A. Lehrmann, Modeling continuous stochastic processes with dynamic normalizing flows, *Advances in Neural Information Processing Systems*, 33: 7805–7815, 2020. Available from: https://proceedings.neurips.cc/paper/2020/hash/58c54802a9fb9526cd0923353a34a7ae-Abstract.html

15. S. Gan, Y. He, X. Wang, Tamed Runge-Kutta methods for SDEs with super-linearly growing drift and diffusion coefficients, *Appl. Numer. Math.,* **152** (2020), 379–402. https://doi.org/10.1016/j.apnum.2019.11.014

16. Q. Guo, W. Liu, X. Mao, R. Yue, The truncated Milstein method for stochastic differential equations with commutative noise, *J. Comput. Appl. Math.,* **338** (2018), 298–310. https://doi.org/10.1016/j.cam.2018.01.014

17. D. J. Higham, X. Mao, A. M. Stuart, Strong convergence of Euler-type methods for nonlinear stochastic differential equations, *SIAM J. Numer. Anal.,* **40** (2002), 1041–1063. https://doi.org/10.1137/S0036142901389530

18. M. Hutzenthaler, A. Jentzen, *Numerical approximations of stochastic differential equations with non-globally Lipschitz continuous coefficients*, American Mathematical Society, 2015. https://doi.org/10.1090/memo/1112

19. M. Hutzenthaler, A. Jentzen, On a perturbation theory and on strong convergence rates for stochastic ordinary and partial differential equations with nonglobally monotone coefficients, *Ann. Probab.,* **48** (2020), 53–93. https://www.jstor.org/stable/26922909

20. M. Hutzenthaler, A. Jentzen, P. E. Kloeden, Strong and weak divergence in finite time of Euler's method for stochastic differential equations with non-globally Lipschitz continuous coefficients, *P. Roy. Soc. A-Math. Phy.,* **467** (2011), 1563–1576. https://doi.org/10.1098/rspa.2010.0348

21. M. Hutzenthaler, A. Jentzen, X. Wang, Exponential integrability properties of numerical approximation processes for nonlinear stochastic differential equations, *Math. Comput.,* **87** (2018), 1353–1413. http://dx.doi.org/10.1090/mcom/3146

22. C. Kelly, Gabriel. J. Lord, F. Sun, Strong convergence of an adaptive time-stepping Milstein method for SDEs with monotone coefficients, *BIT,* **63** (2023), 33. https://doi.org/10.1007/s10543-023-00969-9

23. P. Kidger, On neural differential equations, *arXiv preprint arXiv:2202.02435, 2022,* https://doi.org/10.48550/arXiv.2202.02435

24. C. Kumar, S. Sabanis, On Milstein approximations with varying coefficients: the case of superlinear diffusion coefficients, *BIT,* **59** (2023), 929–968. https://doi.org/10.1007/s10543-019-00756-5

25. A. L. Lewis, *Option valuation under stochastic volatility ii*, Finance Press, 2009. Available from: `https://financepress.com/wp-content/uploads/2016/06/Lewis.Vol2_.TOC_.pdf`

26. X. Li, T. K. Wong, R. Chen, D. K. Duvenaud, *Scalable gradients and variational inference for stochastic differential equations*, In Symposium on Advances in Approximate Bayesian Inference, pages 1–28. PMLR, 2020. Available from: `https://proceedings.mlr.press/v118/li20a`

27. X. Li, G. Yin, Explicit Milstein schemes with truncation for nonlinear stochastic differential equations: Convergence and its rate, *J. Comput. Appl. Math.,* **374** (2020), 112771. https://doi.org/10.1016/j.cam.2020.112771

28. G. J. Lord, C. E. Powell, T. Shardlow, *An introduction to computational stochastic PDEs,* Cambridge University Press, 2014. https://doi.org/10.1017/CBO9781139017329

29. X. Mao, *Stochastic differential equations and applications*, Elsevier, 2007.

30. X. Mao, The truncated Euler–Maruyama method for stochastic differential equations, *J. Comput. Appl. Math.,* **290** (2015), 370–384. https://doi.org/10.1016/j.cam.2015.06.002

31. X. Mao, Convergence rates of the truncated Euler–Maruyama method for stochastic differential equations, *J. Comput. Appl. Math.,* **296** (2016), 362–375. https://doi.org/10.1016/j.cam.2015.09.035

32. X. Mao, L. Szpruch, Strong convergence and stability of implicit numerical methods for stochastic differential equations with non-globally Lipschitz continuous coefficients, *J. Comput. Appl. Math.,* **238** (2013), 14–28. https://doi.org/10.1016/j.cam.2012.08.015
33. X. Mao, L. Szpruch, Strong convergence rates for backward Euler–Maruyama method for non-linear dissipative-type stochastic differential equations with super-linear diffusion coefficients, *Stochastics,* **85** (2013), 144–171. https://doi.org/10.1080/17442508.2011.651213
34. G. N. Milstein, M. V. Tretyakov, *Stochastic numerics for mathematical physics*, volume 39. Springer, 2004. https://doi.org/10.1007/978-3-030-82040-4
35. A. Neuenkirch, L. Szpruch, First order strong approximations of scalar SDEs defined in a domain, *Numer. Math.,* **128** (2014), 103–136. https://doi.org/10.1007/s00211-014-0606-4
36. B. Oksendal, *Stochastic differential equations: An introduction with applications*, Springer Science & Business Media, 2013.
37. M. Opper, Variational inference for stochastic differential equations, *Ann. Phys.-Berlin,* **531** (2019), 1800233. https://doi.org/10.1002/andp.201800233
38. M. V. Tretyakov, Z. Zhang, A fundamental mean-square convergence theorem for SDEs with locally lipschitz coefficients and its applications, *SIAM J. Numer. Anal.,* **51** (2013), 3135–3162. https://doi.org/10.1137/120902318
39. X. Wang, Mean-square convergence rates of implicit Milstein type methods for SDEs with nonLipschitz coefficients, *Adv. Comput. Math.,* **49** (2023), 37. https://doi.org/10.1007/s10444-023-10034-2
40. X. Wang, S. Gan, The tamed Milstein method for commutative stochastic differential equations with non-globally Lipschitz continuous coefficients, *J. Differ. Equ. Appl.,* **19** (2013), 466–490. https://doi.org/10.1080/10236198.2012.656617
41. X. Wang, J. Wu, B. Dong, Mean-square convergence rates of stochastic theta methods for SDEs under a coupled monotonicity condition, *BIT,* **60** (2020), 759–790. https://doi.org/10.1007/s10543-019-00793-0
42. J. Yao, S. Gan, Stability of the drift-implicit and double-implicit Milstein schemes for nonlinear SDEs, *Appl. Math. Comput.,* **339** (2018), 294–301. https://doi.org/10.1016/j.amc.2018.07.026
43. Z. Zhang, H. Ma, Order-preserving strong schemes for SDEs with locally Lipschitz coefficients, *Appl. Numer. Math.,* **112** (2017), 1–16. https://doi.org/10.1016/j.apnum.2016.09.013
44. X. Zong, F. Wu, G. Xu, Convergence and stability of two classes of theta-Milstein schemes for stochastic differential equations, *J. Comput. Appl. Math.,* **336** (2018), 8–29. https://doi.org/10.1016/j.cam.2017.12.025