

# NUMERICAL METHODS FOR STOCHASTIC DIFFERENTIAL EQUATIONS BASED ON GAUSSIAN MIXTURE\*

LEI LI<sup>†</sup>, JIANFENG LU<sup>‡</sup>, JONATHAN C. MATTINGLY<sup>§</sup>, AND LIHAN WANG<sup>¶</sup>

**Abstract.** We develop in this work a numerical method for stochastic differential equations (SDEs) with weak second-order accuracy based on Gaussian mixture. Unlike conventional higher order schemes for SDEs based on Itô-Taylor expansion and iterated Itô integrals, the scheme we propose approximates the probability measure  $\mu(X^{n+1} | X^n = x_n)$  using a mixture of Gaussians. The solution at the next time step  $X^{n+1}$  is drawn from the Gaussian mixture with complexity linear in dimension  $d$ . This provides a new strategy to construct efficient high weak order numerical schemes for SDEs.

**Keywords.** Gaussian mixture; stochastic differential equation; second-order scheme; weak convergence.

**AMS subject classifications.** 60H35; 65C30; 65L20.

## 1. Introduction

Stochastic differential equations (SDEs) [23] have been used to model a wide range of phenomena, such as stock prices of financial derivatives [4, 11], and physical systems in contact with heat bath [5, 12, 30]. SDEs have recently also been used for analyzing stochastic gradient descent (SGD) in machine learning [7, 9, 13]. The SDEs are dynamical systems with noise [6, 23] that often represent interactions that are not included in the model but affect the dynamics. For example, in the Langevin equations [5, 24], one considers the evolution of a subsystem, while the rest of the system, consisting of potentially large degrees of freedom, is regarded as a heat bath. The interaction between the heat bath and the subsystem is modelled by noise and dissipation terms.

We will consider general SDEs driven by white noise [6, 24] and in particular SDEs in Itô sense [23, Chap. 5]:

$$dX(t) = b(X(t))dt + \sigma(X(t))dW, \quad X(0) = x, \quad (1.1)$$

where  $X \in \mathbb{R}^d$ ,  $W$  is a standard  $m$ -dimensional Brownian motion,  $b: \mathbb{R}^d \rightarrow \mathbb{R}^d$  is the drift, and  $\sigma: \mathbb{R}^d \rightarrow \mathbb{R}^{d \times m}$  is the diffusion matrix. We are interested in its numerical approximations, and depending on whether our goal is to approximate the sample paths or the distributions, the numerical methods can be classified into strong schemes and weak schemes [6, 10]. The weak schemes approximate the distributions, and we refer the readers to Section 2.2 for more details. Indeed, weak schemes attempt to match the moments of the iterated Itô integrals, and therefore, the key question for designing weak schemes is how to approximate these moments efficiently. The classical Euler-Maruyama scheme (3.1) is known to be a weak first-order scheme. In applications, schemes with higher order accuracy are often desired. The weak second-order schemes, however, are not trivial for SDEs. The traditional second-order schemes, based on Itô-Taylor

---

\*Received: May 04, 2020; Accepted (in revised form): February 01, 2021. Communicated by Grigorios A. Pavliotis.

<sup>†</sup>School of Mathematical Sciences, Institute of Natural Sciences, MOE-LSC, Shanghai Jiao Tong University, Minhang District, Shanghai 200240, China ([leili2010@sjtu.edu.cn](mailto:leili2010@sjtu.edu.cn)).

<sup>‡</sup>Department of Mathematics, Department of Chemistry and Department of Physics, Duke University, Durham, NC 27708, USA ([jianfeng@math.duke.edu](mailto:jianfeng@math.duke.edu)).

<sup>§</sup>Department of Mathematics and Department of Statistical Science, Duke University, Durham, NC 27708, USA ([jonm@math.duke.edu](mailto:jonm@math.duke.edu)).

<sup>¶</sup>Department of Mathematics, Duke University, Durham, NC 27708, USA ([lihan@math.duke.edu](mailto:lihan@math.duke.edu)).

expansion [10, 20], involve evaluations of the spatial derivatives of drift and diffusion coefficients, as well as iterated Itô integrals. The weak second-order schemes date back to Milstein and Talay [18, 28]. The Talay-Tubaro expansion can also yield a weak second-order approximation [29]. In [25, 26], Runge-Kutta methods that achieve arbitrary weak order for scalar noise and weak second-order for general noise are developed. Runge-Kutta schemes can avoid approximating some derivatives of the drift and the diffusion coefficients directly (see for example the Talay-Tubaro scheme), and lead to better stability. A weak trapezoidal second-order method has been developed in [2], which is derivative free and no evaluation of iterated Itô integrals is needed. However, it leverages the structure of a particular, but common, class of equations. In [1], higher order convergence for a class of SDEs is achieved based on solving modified SDEs. In [21, 22], another class of higher order schemes were developed which are often referred to as Lie splitting methods. Like the current methods, they strive for a level of weak accuracy by deriving a condition which guarantees that certain terms vanish in an expansion of the difference of the true density and that of the numerical method.

As will be discussed in Section 2.2, one may approximate the conditional distribution  $\mu(X^{n+1} | X^n = x_n)$  with asymptotic weak local error  $O(h^3)$  to achieve global weak second-order accuracy, where  $\{X^n\}$ 's are the numerical solutions and  $h$  is the step size. In this work, we propose a novel Gaussian mixture method to achieve  $O(h^3)$  weak local error in which  $X^{n+1}$  can be sampled from one of a mixture of Gaussians. Our ansatz is inspired by the expansion of the solution using commutators. Our Gaussian particle ansatz has two attractive properties:

- Since only one of the Gaussians in the mixture is chosen in each step, the cost in each step is minimal and the simulation is fast: we only need to generate  $d$  scalar random variables (with three possible values only) to generate an initial point, and then generate a  $d$ -dimensional multi-variate normal variable whose mean and covariance matrix are related with the  $d$  scalar random variables and obtained by solving an ODE (see Remark 4.4 for some comments on the number of random variables needed). In this sense, the method we construct in this paper can be considered as a random mixture of Euler-Maruyama steps that produces a higher order method, and it is simple to implement.
- Secondly, our scheme does not need the spatial derivatives of the coefficients, which is useful in several contexts.

Numerical simulations show that our Gaussian mixture method is indeed weak second-order for reasonable values of the step size. This agrees with our theoretical results that our methods are asymptotically second-order as the step size goes to zero. For related works about using Gaussian approximations for general distributions, see [3, 14]. In [3], Gaussian processes based on a variational approach are used to approximate posterior measure in path space. In [15], Gaussian approximation is used to approximate transition paths in Langevin dynamics.

This work is primarily interested with accuracy considerations. This is reflected in the use of essentially forward-Euler like steps to build the Gaussian Mixture. It is easy to imagine replacing these with an implicit step to address stability concerns.

The rest of the paper is organized as follows. In Section 2, we give a brief introduction to SDEs and the basic setup of our problem. In particular, the concept and criteria for weak accuracy using test functions with bounded derivatives are introduced. In Section 3, we introduce the idea of Gaussian mixtures for high order weak accuracy and develop an algorithm for one-dimensional SDEs with weak second-order accuracy, where the mean and variance of the Gaussian are computed either based on some ODEs

or construction. In Section 4, we generalize the algorithm for 1D SDEs to SDEs in multi-dimensions. The number of Gaussian beams are exponential in the dimension  $d$ , but we only need  $d$  discrete random variables to determine which beam we choose, so the complexity is linear in  $d$ . In Section 5, we perform several numerical examples to see how our algorithm performs regarding different aspects.

**2. Preliminaries**

In this section, we collect some definitions and notations related to SDEs. Moreover, the notion of weak convergence is introduced in detail, which lays the foundation of our construction of Gaussian mixture methods in later sections.

**2.1. Notations and assumptions.** For the convenience of later discussions, we introduce for any integer  $k$  the following set of functions

$$C_b^k = \left\{ f \in C^k : \|f\|_{C^k} := \sup_{x \in \mathbb{R}^d} \sum_{|\alpha| \leq k} |D^\alpha f| < \infty \right\}. \tag{2.1}$$

Here the subscript  $b$  is used to remind the reader that the functions are bounded in value and all their derivatives up to the specified order. We use  $\mathbb{E}_x$  to denote the expectation under the law of the process  $X(t)$  with  $X(0) = x$ . The notation  $\mathcal{N}(m, \Lambda)$  denotes the normal distribution with mean  $m$  and covariance matrix  $\Lambda$ .

We list out the following assumptions, which will be used throughout this work:

ASSUMPTION 2.1. *The diffusion matrix*

$$\Lambda(x) := \sigma(x)\sigma^T(x) \tag{2.2}$$

*is uniformly positive definite. In other words,*

$$\inf_{x \in \mathbb{R}^d} \min \lambda(\Lambda(x)) \geq \sigma_0^2 > 0$$

*for some  $\sigma_0 > 0$ , where  $\lambda(\Lambda)$  is the set of eigenvalues of  $\Lambda$ .*

ASSUMPTION 2.2. *The coefficients are smooth in the sense that  $b, \sigma \in C_b^6$ .*

Note that Assumption 2.2 implies that the coefficients are Lipschitz continuous, i.e. there exists a constant  $K > 0$  such that for all  $x, y \in \mathbb{R}^d$ ,

$$|b(x) - b(y)| + |\sigma(x) - \sigma(y)| \leq K|x - y|. \tag{2.3}$$

It is well known that Assumption 2.2 ensures the existence of strong solutions to (1.1) [23] and that the moments of the solution are bounded:

$$\sup_{0 \leq t \leq T} \mathbb{E}_x |X(t)|^{2m} \leq C(x, T).$$

Though likely overly restrictive, Assumption 2.1 and Assumption 2.2 will simplify the analysis and make the ideas more transparent. Analysis based on Assumption 2.2 has been pursued in many works (see for example [2]). Compared with Assumption 2.2, some authors relax the coefficients to have polynomial growth (see for example [19]). The current results can be extended to locally Lipschitz coefficients with polynomial growth under appropriate one-sided Lyapunov conditions or simply the arbitrary moment bounds they imply, see for instance [16].

The generator of the diffusion process (1.1) is given by

$$\mathcal{L} = \sum_{i=1}^d b_i \partial_i + \sum_{1 \leq i, j \leq d} \frac{1}{2} \Lambda_{ij} \partial_{ij}. \tag{2.4}$$

The evolution of the law satisfies the Fokker-Planck equation (or the forward Kolmogorov equation)

$$\partial_t p = \mathcal{L}^* p := -\nabla \cdot (bp) + \frac{1}{2} \sum_{ij} \partial_{ij}^2 (\Lambda_{ij} p). \tag{2.5}$$

For a smooth function  $\phi$ , let us define

$$u(x, t) = \mathbb{E}_x \phi(X(t)). \tag{2.6}$$

With regularity Assumptions 2.1 and 2.2,  $u$  satisfies the backward Kolmogorov equation (see, for example, [23, Chap. 8])

$$u_t = \mathcal{L}u. \tag{2.7}$$

Formally, this implies the semigroup expansion

$$u(x, t) = e^{t\mathcal{L}} \phi(x) = \sum_{j=0}^{\infty} \frac{t^j}{j!} \mathcal{L}^j \phi(x). \tag{2.8}$$

Given regularity assumptions on  $\phi$ , the expansion can be rigorously established up to a certain order. We cite a classical result in [8, Chap. XI] for expansion up to  $j=2$ , which has been modified for our purpose.

LEMMA 2.1 ([8, Theorem 11.6.4]). *Under Assumptions 2.1–2.2, there exists a non-negative non-decreasing function  $\rho$ , such that for all  $\phi \in C_b^\infty$ ,*

$$\sup_{x \in \mathbb{R}^d} \left| u(x, h) - \left( \phi(x) + \sum_{j=1}^2 \frac{h^j}{j!} \mathcal{L}^j \phi \right) \right| \leq \rho(\|\phi\|_{C^6}) h^3. \tag{2.9}$$

In a numerical scheme, we generate the approximation sequences for the diffusion process at discrete time steps. Let  $T > 0$  be the terminal time point, and  $N$  the number of numerical steps such that

$$h = T/N. \tag{2.10}$$

We use  $t_n = nh$  ( $n = 0, 1, \dots$ ) to denote the time grid points,  $X^n$  the random variable generated by some numerical method to approximate  $X(t_n)$ , and  $x_n$  a particular realization of the random variable  $X^n$ .

**2.2. Weak convergence.** We only require the law of  $X^n$  to approximate the law of the solution to (1.1). This is described by the notion of weak convergence, which will be the focus of this section.

DEFINITION 2.1. *Fix  $T > 0$ . Let  $N, h$  and  $X^n$  be given as in Section 2.1. We say  $X^n$  converges weakly with order  $r > 0$  to  $X(t_n)$  as  $h \rightarrow 0$  if for any  $\phi \in C_b^\infty$ , there exist  $C > 0$ ,  $h_0 > 0$  that are independent of  $h$  (but may depend on  $T$  and  $\phi$ ) such that*

$$\left| \mathbb{E} \phi(X^n) - \mathbb{E} \phi(X(t_n)) \right| \leq Ch^r, \quad \forall 1 \leq n \leq N, \text{ whenever } h \in (0, h_0). \tag{2.11}$$

Here,  $\mathbb{E}$  represents the expectation under the law of  $X^n$  or  $X(t_n)$ .

REMARK 2.1. Note that the test functions here have bounded derivatives, and those used in [10, Sec. 9.7] and [19] have derivatives with polynomial growth. Test functions with bounded derivatives induce weaker topology but are much easier to handle (see e.g., [2]). The results can be extended to the more general setting with additional work and assumptions to ensure boundedness of moments.

We now move on to the criteria of weak convergence. Suppose the random sequence  $X^n$  is generated by

$$X^{n+1} = X^n + A(X^n, \zeta^n, h), X^0 = x, \tag{2.12}$$

where  $\zeta^n$  is a random vector generated at time  $t_n$  and  $A$  is a function. If  $\{\zeta^n\}$ 's are i.i.d, then  $\{X^n\}$  is a time-homogeneous Markov chain. (For example, in Algorithm 1 below,  $\zeta^n$  is a combination of the  $z$  random variable and the standard 1D normal variable  $\xi$ .)

The following proposition is standard and we provide the proof in Appendix A for reference. We emphasize that the following two results are not new and are part of the standard “folklore” meta-theorems in the subject. We repeat them only to make precise the versions we need and their dependences on parameters.

PROPOSITION 2.1. *Let  $b$  and  $\sigma$  satisfy  $b, \sigma \in C_b^{2(r+1)}$  for  $r > 0$ . If there is a nonnegative and non-decreasing function  $\rho$  such that for all  $\phi \in C_b^\infty$ , we have the local truncation error bounded by*

$$|\mathbb{E}_x \phi(X(h)) - \mathbb{E}_x \phi(X^1)| \leq \rho(\|\phi\|_{C^{2(r+1)}}) h^{r+1}, \forall x \in \mathbb{R}^d,$$

then  $X^n$  converges weakly with order  $r$  to  $X(t_n)$  as  $h \rightarrow 0$ .

As before,  $\mathbb{E}_x$  represents the expectation under the law of the process or Markov chain starting at  $x$ . This proposition basically says that if the local truncation error is  $O(h^{r+1})$ , then the global error is of order  $r$ . We have the following trivial observation by Lemma 2.1 and Proposition 2.1:

COROLLARY 2.1. *Under Assumptions 2.1-2.2, if there exists  $\rho$  that is nonnegative and non-decreasing such that  $\forall \phi \in C_b^\infty$ , we have*

$$\sup_{x \in \mathbb{R}^d} \left| \mathbb{E}_x(\phi(X^1)) - \sum_{j=0}^2 \frac{h^j}{j!} \mathcal{L}^j \phi(x) \right| \leq \rho(\|\phi\|_{C^6}) h^3, \tag{2.13}$$

then the method (2.12) is of weak second-order accuracy.

### 3. Weak second-order Gaussian mixture method

The Euler-Maruyama scheme [10] for SDE (1.1)

$$X^{n+1} = X^n + b(X^n)h + \sigma(X^n)\Delta W_n \tag{3.1}$$

generates Gaussian distributions for  $X^{n+1}$  conditioning on  $X^n = x_n$  but has only weak first-order accuracy. It is well known that constructing a weak second-order scheme is nontrivial, not to mention a weak second-order scheme using Gaussian approximations for the measure  $\mu(X^{n+1}|X^n = x_n)$ . In fact, as we will see, an approximation using a single Gaussian is generally insufficient for weak second-order accuracy. Hence, we aim to use Gaussian mixture to construct higher order schemes.

To start with, let us recall that the law of  $Y(t)$ , the weak solution of the following SDE with additive noise, is a Gaussian distribution if  $Y(0)$  is a Gaussian random variable independent of  $W$ :

$$dY(t) = \mu(t)dt + \sigma(t)dW. \tag{3.2}$$

Here,  $\mu(t)$  and  $\sigma(t)$  only depend on time. The mean and covariance matrix of  $Y(t)$  are given respectively by

$$m(t) = m_0 + \int_0^t \mu(s)ds, \quad S(t) = S_0 + \int_0^t \sigma(s)\sigma^T(s)ds. \tag{3.3}$$

Conversely, if we are given some  $m$  and some positive semidefinite matrix  $S$ , we can recover the normal distribution  $\mathcal{N}(m, S)$  by constructing “paths”  $\{m(t)\}_{t \in [0, h]}$  and  $\{S(t)\}_{t \in [0, h]}$  with  $m(h) = m, S(h) = S$ , and  $\dot{S}$  being positive semi-definite so that the solution to the SDE

$$dY(t) = \dot{m}(t)dt + \sqrt{\dot{S}(t)}dW$$

with  $Y(0) \sim \mathcal{N}(m(0), S(0))$ , independent of  $W$ , will satisfy  $Y(h) \sim \mathcal{N}(m, S)$ . Here  $\dot{m}$  and  $\dot{S}$  denote their respective time derivatives. Now let us consider the time-dependent generator

$$\mathcal{L}(s) = \dot{m}(s) \cdot \nabla_x + \frac{1}{2} \dot{S}_{ij}(s) \partial_{ij}. \tag{3.4}$$

We now assume  $S(0) = 0$  and  $m(0) = x_0$  for some  $x_0$ . By the backward equation (2.7), we have

$$\mathbb{E}\phi(Y(h)) = \exp\left(\int_0^h \mathcal{L}(s)ds\right)\phi(x_0) = \exp\left((m - x_0) \cdot \nabla_x + \frac{1}{2} S_{ij} \partial_{ij}\right)\phi(x_0). \tag{3.5}$$

Here the second equality comes from integrating  $\mathcal{L}$  in time. Therefore, for any random variable  $Y \sim \mathcal{N}(m, S)$ , we can express the expectation of  $\phi$  as

$$\mathbb{E}\phi(Y) = \exp(\mathcal{L}_z)\varphi(z), \quad \forall z \in \mathbb{R}^d, \tag{3.6}$$

where

$$\mathcal{L}_z = (m - z) \partial_x + \frac{1}{2} S_{ij} \partial_{ij}. \tag{3.7}$$

We will use (3.6) to construct our scheme. In this section, we start with  $d = 1$ . Our construction for  $d = 1$  will be used as the building block for constructing our scheme in higher dimensions in Section 4.

**3.1. Conditions for second-order Gaussian mixtures.** First of all, we claim that using a single Gaussian distribution to approximate  $\mu(X^{n+1} | X^n = x_n)$  is generally insufficient for weak second-order accuracy. To start with, we assume  $X^1$  generated by (2.12) conditioning on  $X^0 = x_0$  is a normal distribution with mean  $m(h, x_0)$  and variance  $S(h, x_0)$ :

$$X^1 = x_0 + A(x_0, \xi, h) \sim \tilde{\rho}(x; h, x_0) = \frac{1}{\sqrt{2\pi S(h, x_0)}} \exp\left(-\frac{(x - m(h, x_0))^2}{2S(h, x_0)}\right). \tag{3.8}$$

Here  $X^1 \sim \tilde{\rho}$  means the law of  $X^1$  has a density  $\tilde{\rho}$ . Using (3.6), we desire

$$\exp\left((m(h, x_0) - x_0)\partial_x + \frac{1}{2}S(h, x_0)\partial_{xx}\right)\varphi(x_0) = \varphi(x_0) + h\mathcal{L}\varphi(x_0) + \frac{h^2}{2}\mathcal{L}^2\varphi(x_0) + O(h^3) \tag{3.9}$$

in order to achieve global weak second-order accuracy. Clearly, we need  $m(h, x_0) - x_0 = o(1), S(h, x_0) = o(1)$  as  $h \rightarrow 0$ . Using the semigroup expansion (Lemma 2.1), we infer that

$$\begin{aligned} m(h, x_0) &= x_0 + m_1(x_0)h + \frac{1}{2}m_2(x_0)h^2 + R_1(h, x_0)h^3, \\ S(h, x_0) &= S_0(x_0)h + \frac{1}{2}S_1(x_0)h^2 + R_2(h, x_0)h^3, \end{aligned}$$

where  $R_1, R_2$  are bounded. Detailed calculation shows the following:

**PROPOSITION 3.1.** *For a general multiplicative noise (or equivalently  $\sigma(x)$  is not constant), there exist no  $(m_0, m_1, m_2, S_0, S_1)$  as functions of  $x_0$  such that the constraint (2.13) can be satisfied.*

**REMARK 3.1.** By the proof of Proposition (3.1) in Appendix B, if the noise is additive ( $\sigma$  is independent of  $x$ ), it is possible to construct an approximation with a single Gaussian that yields global weak second-order accuracy.

The proof of Proposition 3.1 is provided in Appendix B. Proposition 3.1 is a strong indication that no approximation with one Gaussian can reach weak second-order accuracy, which forces us to seek Gaussian mixtures. In the derivation below, we use  $R(x)$  to denote a generic function with a bound that depends only on  $\|\cdot\|_{C^6}$  norms of  $b, \sigma$ , and  $\phi$  (the test function), and its concrete meaning may change from line to line. Below, we would first present an informal argument to derive the scheme; the rigorous analysis of the scheme will be deferred to later sections.

As we have mentioned, considering the law of  $X^1$  given the initial position  $X^0 = X(0) = x_0$  is sufficient to determine the whole Markov chain by time homogeneity. Therefore, it suffices to consider that the law of  $X^1$  is given by a mixture of  $M$  Gaussians:

$$X^1 \sim \sum_{i=1}^M w_i \mathcal{N}(m_i(h), S_i(h)). \tag{3.10}$$

Here we abuse notation by letting  $\mathcal{N}(m, S)$  denote the density function of a Gaussian with the given mean and covariance.

Let  $\mathcal{L}_i := (m_i(h) - x_0)\partial_x + \frac{1}{2}S_i(h)\partial_{xx}$ . By (3.6), we have

$$\mathbb{E}\phi(X^1) = \sum_{i=1}^M w_i \exp(\mathcal{L}_i)\phi(x_0). \tag{3.11}$$

Here, the dependence on  $x_0$  in the coefficients is not written out explicitly for simplicity. Since after time  $h$ , the scale for a pure diffusion process is  $\sqrt{h}$  (since  $\mathbb{E}|X(t+h) - X(t)|^2 \sim h$ ), we expect that  $|m_i(h) - x_0| \leq C\sqrt{h}$  and  $|S_i(h)| \leq C_M h$ . Therefore, by Corollary 2.1, the scheme will be of weak second-order if the following holds for all  $\phi \in C_b^\infty$ :

$$\sum_{i=1}^M w_i \left( \mathcal{L}_i \phi(x_0) + \frac{1}{2} \mathcal{L}_i^2 \phi(x_0) + \frac{1}{6} \mathcal{L}_i^3 \phi(x_0) + \frac{1}{24} \mathcal{L}_i^4 \phi(x_0) \right) - \left( h \mathcal{L} \phi(x_0) + \frac{1}{2} h^2 \mathcal{L}^2 \phi(x_0) \right) = R(x_0, h) h^3, \quad (3.12)$$

where  $R(h)$  is bounded and depends on the function  $\phi$ . We stop at  $\mathcal{L}_i^4$  because of the expectation  $|m_i(h) - x_0| \leq C\sqrt{h}$  so  $\mathcal{L}_i^k$  is of order at least  $O(h^{\frac{k}{2}})$ . Note that by (2.4),

$$\begin{aligned} \mathcal{L} \phi(x_0) &= b(x_0) \phi'(x_0) + \frac{1}{2} \Lambda(x_0) \phi''(x_0), \\ \frac{1}{2} \mathcal{L}^2 \phi(x_0) &= \frac{1}{2} \left( b(x_0) (b\phi') + \frac{1}{2} \Lambda \phi'' \right)' + \frac{1}{2} \Lambda(x_0) \left( b\phi' + \frac{1}{2} \Lambda \phi'' \right)''. \end{aligned} \quad (3.13)$$

Due to the  $\sqrt{h}$  scale in displacement, we take the following ansatz for  $m_i(h)$  and  $S_i(h)$ :

$$\begin{aligned} m_i(h) &= x_0 + m_{i0} h^{1/2} + m_{i1} h + m_{i2} h^{3/2} + m_{i3} h^2 + m_{i4} h^{5/2} + R_{i1}(h) h^3, \\ S_i(h) &= S_{i1} h + S_{i2} h^{3/2} + S_{i3} h^2 + S_{i4} h^{5/2} + R_{i2}(h) h^3 > 0. \end{aligned} \quad (3.14)$$

Substituting the ansatz (3.14) into (3.12), after a tedious but straightforward calculation, we are able to derive the following conditions:

$$\begin{aligned} \sum_i w_i m_{i1} &= b, \\ \frac{1}{2} \sum_i w_i S_{i1} + \frac{1}{2} \sum_i w_i m_{i0}^2 &= \frac{1}{2} \Lambda, \\ \sum_i w_i m_{i3} &= \frac{1}{2} b b' + \frac{1}{4} \Lambda b'', \\ \frac{1}{2} \sum_i w_i S_{i3} + \frac{1}{2} \sum_i w_i (2m_{i0} m_{i2} + m_{i1}^2) &= \frac{1}{2} b^2 + \frac{1}{4} b \Lambda' + \frac{1}{2} \Lambda b' + \frac{1}{8} \Lambda \Lambda'', \\ \frac{1}{2} \sum_i w_i m_{i1} S_{i1} + \frac{1}{2} \sum_i w_i m_{i0} S_{i2} + \frac{1}{2} \sum_i w_i m_{i0}^2 m_{i1} &= \frac{1}{2} b \Lambda + \frac{1}{4} \Lambda \Lambda', \\ \frac{1}{8} \sum_i w_i S_{i1}^2 + \frac{1}{4} \sum_i w_i m_{i0}^2 S_{i1} + \frac{1}{24} \sum_i w_i m_{i0}^4 &= \frac{1}{8} \Lambda^2, \end{aligned}$$

All the odd powers of  $h^{1/2}$  vanish in  $\sum_i w_i \mathcal{L}_i^m$  for any  $m = 1, 2, 3, 4$ . (3.15)

In the above equations, functions  $b, \Lambda$  and their spatial derivatives are all evaluated at point  $x_0$ .

In the following Section 3.2, we consider a possible approach to satisfy these constraints, by choosing  $M = 3$ .

**REMARK 3.2.** We have not yet derived a weak third-order Gaussian mixture scheme. The number of variables and the equations grow to the point where our current methods to solve them are unfeasible. However, we expect that a minimum of five Gaussians is needed to reach third order, which is suggested by the second and sixth equations of (3.15). These are constraints for  $\phi''$  in first order and  $\phi^{(4)}$  in second order respectively (and there will be another constraint for  $\phi^{(6)}$  in third order), which only involve the weights  $w_i$  and the leading order diffusion scaling terms  $m_{i0}$  and  $S_{i1}$ .



**3.2. An ODE approach.** In this section, we give a particular construction of  $m_i(h)$  and  $S_i(h)$  that satisfy (3.15) which determines our numerical scheme. Our approach is to construct ODEs for  $m_i$  and  $S_i$  with a certain initial condition and solve them at time  $h$ , which has the advantage of avoiding derivative evaluations of  $b$  and  $\Lambda$ .

To satisfy the last condition of (3.15), we consider a “symmetric” construction. It is convenient to relabel the Gaussians as  $i = 0, \pm 1$ , so that  $\mathcal{N}(m_0(h), S_0(h))$  is “centralized” and does not contribute to the odd powers of  $h^{\frac{1}{2}}$ . The centers of the other two Gaussians  $m_{\pm 1}(h)$  are placed at both sides of  $m_0(h)$  with  $O(\sqrt{h})$  distance apart, and the variance matrices  $S_1(h), S_{-1}(h)$  are constructed similarly with each other. These two Gaussians will contribute powers like  $h^{k/2}$ . Moreover, we impose  $w_1 = w_{-1}$ , so that the odd powers of  $h^{\frac{1}{2}}$  will cancel out with each other due to symmetry.

For initial conditions, we set

$$\begin{aligned} m_i(0) &= x_0 + z_i \sqrt{\gamma h \Lambda(x_0)}, S_i(0) = 0, \\ w_1 &= w_{-1}, \quad w_0 + 2w_1 = 1, \end{aligned} \tag{3.16}$$

where  $\gamma > 0$  is a parameter and

$$z_i = i, \quad i = 0, \pm 1. \tag{3.17}$$

This choice takes into consideration that the diffusion scale is  $\sqrt{h}$ , while transportation scale is  $h$ . For the choice of ODE flows, we take the following ansatz, where the functions  $g_i$  are to be determined later:

$$\dot{m}_i(t) = b(m_i(t)), \tag{3.18a}$$

$$\dot{S}_i(t) = g_i(m_i(t)). \tag{3.18b}$$

Our choice in (3.18a) is natural in the sense that we expect

$$\frac{d}{dt} \mathbb{E}X(t) = \mathbb{E}b(X(t)) \approx b(\mathbb{E}X(t)),$$

and the approximation is exact if  $b$  is a linear function. For symmetry, we require  $g_1 = g_{-1}$ . Clearly, the  $\sqrt{h}$  factor enters  $m_i(h)$  through the initial value and then the equation. Due to symmetries in both  $m_{\pm 1}$  and  $S_{\pm 1}$  and  $w_1 = w_{-1}$ , all the odd powers of  $h^{1/2}$  indeed cancel out in  $\sum_i w_i \mathcal{L}_i^m$ .

We now find the constraints on the functions  $g_i$  and the parameters so that (3.15) can be satisfied. To start with, we have by Taylor expansion that

$$m_i(h) = m_i(0) + b(m_i(0))h + \frac{1}{2}b'(m_i(0))b(m_i(0))h^2 + O(h^3). \tag{3.19}$$

Hence, substituting (3.16) into (3.19), and considering our ansatz (3.14), we obtain

$$m_{i0} = z_i \sqrt{\gamma \Lambda}, \quad m_{i1} = b(x_0), \quad m_{i2} = z_i b' \sqrt{\gamma \Lambda}, \quad m_{i3} = \frac{1}{2} b'' z_i^2 \gamma \Lambda + \frac{1}{2} b b'. \tag{3.20}$$

Similarly, we can find  $S_{ij}$ 's:

$$S_{i1} = g_i(x_0), \quad S_{i2} = z_i g'_i \sqrt{\gamma \Lambda}, \quad S_{i3} = \frac{1}{2} z_i^2 g''_i \gamma \Lambda + \frac{1}{2} g'_i b. \tag{3.21}$$

However, substituting (3.20) and (3.21) into (3.15) cannot uniquely determine the parameters  $w_i$ ,  $g_i$  and  $\gamma$ . We further impose  $S_{01} = S_{11}$  which makes our construction of the scheme easier for higher dimensions, which uniquely determines the solution:

$$\gamma = \frac{3}{2}, \quad w_1 = \frac{1}{6}, \quad w_0 = \frac{2}{3},$$

$$\begin{aligned}
 g_0(x_0) &= g_1(x_0) = \frac{1}{2}\Lambda(x_0), \\
 g'_0(x_0) &= g'_1(x_0) = \Lambda'(x_0), \\
 g''_1(x_0) &= \Lambda''(x_0).
 \end{aligned}
 \tag{3.22}$$

Clearly, choosing the following functions will suffice:

$$g_0(x) = g_1(x) = g(x) := \Lambda(x) - \frac{1}{2}\Lambda(x_0).
 \tag{3.23}$$

Unfortunately, this choice has one issue:  $g_0$  and  $g_1$  are not always nonnegative. Indeed, it is possible that  $S_i(h)$  given by (3.18b) could be negative. To solve this issue, we simply set  $S_i(h)$  to zero if that happens. Fortunately, since  $g(x) \approx \frac{1}{2}\Lambda(x_0)$  is positive whenever  $x$  is close to  $x_0$ ,  $S_i(h)$  can be guaranteed to be positive whenever  $h$  is sufficiently small, thus it can be shown that this error has a lower-order effect. Similar situation also arises in [2].

The details of the procedure outlined above are expressed more exactly in the following Algorithm 1 which gives the pseudocode to generate  $x_{n+1}$  from  $x_n$ .

---

**Algorithm 1** Gaussian mixture scheme for SDEs (ODE method in 1D)

---

- 1: Generate  $z$  such that  $P(z=0) = \frac{2}{3}$  and  $P(z=1) = P(z=-1) = \frac{1}{6}$ . Then, set

$$m(0) = x_n + z\sqrt{\frac{3}{2}\Lambda(x_n)h}.
 \tag{3.24}$$

- 2: Solve the ODEs

$$\begin{aligned}
 \dot{m} &= b(m), \\
 \dot{S} &= g(m(t)),
 \end{aligned}
 \tag{3.25}$$

using an ODE solver of at least second-order accuracy (for example, Runge-Kutta methods of order  $k \geq 2$ ) to obtain  $m(h)$ ,  $S(h)$ . Here  $g(x) = \Lambda(x) - \frac{1}{2}\Lambda(x_n)$ .

- 3: If  $S(h) \leq 0$ , then  $x_{n+1} = m(h)$ . If  $S(h) > 0$ , then

$$x_{n+1} = m(h) + \sqrt{S(h)}\xi,
 \tag{3.26}$$

where  $\xi$  is a standard 1D normal variable.

---

REMARK 3.3. One may truncate the function and consider

$$g_0(x) = g_1(x) = \psi(x; x_0)(\Lambda(x) - \Lambda(x_0)) + \frac{1}{2}\Lambda(x_0)$$

where  $\psi(x; x_0)$  is some truncation function that is 1 in a neighborhood of  $x_0$  so that  $g_0, g_1$  are positive definite for all  $x$ . This approach, however, is not very convenient and in practice the behavior is not very satisfactory.

We are now in position to present the following theorem, which tells that our scheme is indeed of weak second-order.

**THEOREM 3.1.** *Let  $d=1$ . Suppose Assumptions 2.1-2.2 hold., then Algorithm 1 is a weak second-order scheme for SDE (1.1).*

*Proof.* It is clear that there exists  $h_0 > 0$  such that for  $h < h_0$ ,

$$\sqrt{\frac{3}{2}\|\Lambda\|_\infty h} + \|b\|_\infty h < \frac{\sigma_0^2}{2\|\Lambda'\|_\infty}.$$

Consider that  $X(0) = X^0 = x_0$ . By construction,  $|m_i(t) - m_i(0)| \leq \|b\|_\infty h$  for all  $t \leq h$ , and we have

$$g(m_i(t)) \geq \frac{1}{2}\Lambda(x_0) - \|\Lambda'\|_\infty \left( |m_i(t) - m_i(0)| + |z_i| \sqrt{\frac{3}{2}\|\Lambda\|_\infty h} \right) > 0.$$

Hence,  $S_i(h) > 0$  for  $h < h_0$ . Moreover, any reasonable numerical approximation to  $S_i(h)$  will also be positive for sufficiently small  $h$ .

By (3.23), we can conclude that for  $h < h_0$ , (3.15) holds, and  $S_i(h) > 0$ . In other words, (3.12) holds and

$$\left| \mathbb{E}_{x_0}(\phi(X^1)) - (\phi(x_0) + \mathcal{L}\phi(x_0) + \frac{1}{2}\mathcal{L}^2\phi(x_0)) \right| \leq \rho(\|\phi\|_{C^6})h^3.$$

By Corollary 2.1, we find that our scheme constructed here is of weak second-order if (3.25) is solved exactly. Since for any numerical solver on (3.25) that is of at least second order, the error induced by solving (3.25) is  $O(h^3)$  or smaller, and therefore the above local estimate still holds. Our Algorithm 1 is thus of weak second order as well.  $\square$

REMARK 3.4. The above construction with ODE flow gives  $S_i(h)$  that can be possibly negative, though it is positive asymptotically as  $h \rightarrow 0$  and when it becomes negative, we can always fix by setting it to zero. One may desire to have a method that ensures  $S_i(h)$  to be positive. In Appendix D, we provide a direct way to construct  $S_i$ 's so that positivity can be guaranteed. However, this method involves evaluation of the derivatives of  $\Lambda$ , which is oftentimes undesired.

#### 4. Gaussian mixture for multi-dimensions

In this section, we generalize the Gaussian mixture method constructed in Section 3 to higher dimensions. We assume that we have the eigen-decomposition for  $\Lambda(x)$ :

$$\Lambda(x) = \sum_{i=1}^d \lambda_i(x) v_i(x) v_i^T(x), \tag{4.1}$$

where  $\lambda_i(x)$ 's are the eigenvalues of the matrix  $\Lambda(x)$ , and  $\{v_i\}$  forms an orthonormal basis of  $\mathbb{R}^d$ .

As discussed in Section 3, we only need to focus on how to generate  $X^1$  given  $X^0 = x_0$ . Again, we assume that  $X^1$  has the conditional probability measure of the form

$$\bar{\rho} = \sum_{p \in P} w_p \mathcal{N}(m_p(h), S_p(h)), \tag{4.2}$$

for Gaussian mixture approximations. Here we use  $P$  to denote the set of indices  $p$ .

To illustrate our choice of the number of Gaussians and their initial positions, suppose we have  $d$ -dimensional decoupled diffusion process (diffusion matrix is diagonal), then we approximate each dimension using our 1D technique in Section 3 and then get a global second-order approximation. In each dimension, we have three Gaussians,

which means we have a total of  $3^d$  Gaussians. If the diffusion matrix is no longer diagonal, we can still consider using  $3^d$  Gaussians. At the first glance, the complexity is large, but fortunately, it turns out that the complexity grows linearly with  $d$  instead of exponentially.

We now explain our construction. Let the index set  $P = \{-1, 0, 1\}^d$ , so that  $|P| = 3^d$ , and each index  $p \in P$  can be expressed as  $p = (z_p^1, \dots, z_p^d)$  where  $z_p^i \in \{0, \pm 1\}$ . Let us consider the Gaussians with initial centers  $y_p$ , given by

$$y_p = x_0 + \sum_{i=1}^d z_p^i \sqrt{\gamma \lambda_i} h v_i, \text{ where } \gamma = \frac{3}{2}. \tag{4.3}$$

These formulas and  $\gamma = \frac{3}{2}$  are obtained from the 1D construction in Section 3. The weight for the Gaussian with index  $p$  is

$$w_p = \prod_{i=1}^d w^{z_p^i}, \quad 1 \leq p \leq 3^d, \tag{4.4}$$

with the parameters given by

$$w^1 = w^{-1} = \frac{1}{6}, \quad w^0 = \frac{2}{3}. \tag{4.5}$$

REMARK 4.1. Another natural idea is to place the initial points at  $x_0, x_0 \pm \sqrt{\gamma \lambda_i} h v_i$  and there are  $2d + 1$  such points. After some attempts, we found that this strategy hardly works when  $d$  is large.

With these initial positions and weights, we can easily generalize our Gaussian mixture constructions for  $d = 1$  to arbitrary dimensions.

**4.1. The ODE approach for multi-dimensions.** Following the construction in the 1D case, we consider  $m_p(h)$  and  $S_p(h)$  for  $p \in P$  given by

$$\begin{aligned} \dot{m}_p(t) &= b(m_p(t)), \quad m_p(0) = y_p, \\ \dot{S}_p(t) &= G(m_p(t)), \quad S_p(0) = 0, \end{aligned} \tag{4.6}$$

where

$$G(x) = \Lambda(x) - \frac{1}{2} \Lambda(x_0). \tag{4.7}$$

Thanks to imposing  $S_{01} = S_{11}$  in (3.21) we are able to have a simple expression (4.7). The algorithm can then be summarized as the following Algorithm 2.

REMARK 4.2. Our algorithm requires a matrix factorization at every time step, which is the most computationally costly step. However, as  $\Lambda(X(t))$  does not change much between consecutive time steps, one could use the matrix of  $v_i$ 's as the preconditioner for next step's computation, which will significantly reduce the computational cost in high dimensions.

We now establish the main result for multi-dimensions:

THEOREM 4.1. *Suppose the Assumptions 2.1-2.2 are satisfied, then there exists  $h_0 > 0$  such that when  $h < h_0$ :*

- (i)  $S_p(h)$  is positive definite for all  $p \in P$  and for any initial position  $x_0$ .

(ii) For any test function  $\phi \in C_b^\infty$ , there exists a constant  $C$  depending on the  $C^6$  norms of  $\phi, b, \sigma$  only, such that

$$|\mathbb{E}_{x_0}(\phi(X^1)) - \mathbb{E}_{x_0}(\phi(X(t_1)))| \leq Ch^3, \quad h < h_0.$$

Consequently, the Gaussian mixture Algorithm 2 is a weak second-order scheme to (1.1).

To prove this theorem, we first present a useful lemma, the proof of which is deferred to Appendix C:

LEMMA 4.1. For a function  $\phi \in C_b^\infty$ , we have

$$\begin{aligned} \sum_{p \in P} w_p \phi(y_p) &= \phi(x_0) + \sum_{i=1}^d w^1 D_i^2 \phi(x_0) \gamma \lambda_i h + \frac{1}{2} \sum_{i \neq j} (w^1)^2 \gamma^2 D_i^2 D_j^2 \phi(x_0) \lambda_i \lambda_j h^2 \\ &\quad + \frac{1}{12} \sum_{i=1}^d (w^1 \gamma^2) D_i^4 \phi(x_0) \lambda_i^2 h^2 + R(h) h^3. \end{aligned}$$

Here we use shorthand notation  $D_i := D_{v_i}$ .

**Algorithm 2** Gaussian mixture methods for SDEs (ODE method in higher D)

1: Compute the matrix eigen-decomposition

$$\Lambda(x_n) = \sum_{i=1}^d \lambda_i v_i v_i^T. \tag{4.8}$$

2: Generate  $z^i, i = 1, 2, \dots, d$  so that  $P(z^i = 0) = \frac{2}{3}$  while  $P(z^i = 1) = P(z^i = -1) = \frac{1}{6}$ .

3: Let

$$m(0) = x_n + \sum_{i=1}^d z^i \sqrt{\frac{3}{2}} \lambda_i h v_i, \tag{4.9}$$

and find  $m(h)$  by solving

$$\dot{m}(t) = b(m(t))$$

using a method with at least second-order accuracy.

4: Find  $S(h)$  by solving

$$\dot{S}(t) = G(m(t)), \quad S(0) = 0.$$

5: If  $S(h)$  is not positive definite, then set all negative eigenvalues to zero (keeping the same eigenvectors) and obtain  $\tilde{S}(h)$ . Sample  $x_{n+1} \sim \mathcal{N}(m(h), \tilde{S}(h))$ . In other words,

$$x_{n+1} = m(h) + \sum_{i=1}^d \sqrt{\mu_i^+} \xi_i u_i$$

where  $S(h) = \sum_{i=1}^d \mu_i u_i u_i^T$  with  $\{u_i\}_{i=1}^d$  being orthonormal,  $\mu_i^+ = \max(\mu_i, 0)$  and  $\{\xi_i\}$  are i.i.d standard 1D normal variables.

REMARK 4.3. Here  $D_{v_i}\phi(x) := v_i(x_0) \cdot \nabla\phi(x)$ , so we have  $D_{v_i}^2\phi = v_i \cdot \nabla(v_i \cdot \nabla\phi(x)) = v_i \otimes v_i : \nabla^2\phi(x)$ . The function inside is  $v_i(x_0) \cdot \nabla\phi(x)$ . In other words, we allow  $\phi$  to change for  $x \neq x_0$  but  $v_i$  is frozen to be its value at  $x_0$ .

*Proof. (Proof of Theorem 4.1.)*

(i). To prove this claim, we find that for all  $p \in P$ ,

$$|m_p(t) - m_p(0)| \leq \|b\|_\infty h, \quad \forall t \in [0, h]. \tag{4.10}$$

Hence,

$$\begin{aligned} \min \lambda(G(m_p(t))) &\geq \frac{1}{2} \min \lambda(\Lambda(x_0)) - \|\Lambda(m_p(t)) - \Lambda(x_0)\|_2 \\ &\geq \frac{1}{2} \min \lambda(\Lambda(x_0)) - \sup_{x \in \mathbb{R}^d} \|\Lambda'\|_2 \left( |m_p(t) - m_p(0)| + \sqrt{\frac{3}{2} \max_{1 \leq i \leq d} \lambda_i h} \right). \end{aligned}$$

Recall that we use  $\lambda(M)$  to represent the set of eigenvalues of matrix  $M$ . If  $h$  is sufficiently small,  $\min \lambda(G(m_p(t)))$  is positive for all  $p \in P$  for  $t \leq h$ . By Equation (4.6),  $\min \lambda(S_p(h))$  is positive for all  $p$ .

(ii). Noticing that  $\partial_{ijkl}\phi$  is a symmetric tensor on any indices, we find (the Einstein summation convention is used)

$$\begin{aligned} \mathbb{E}_{x_0}(\phi(X^1)) &= \sum_{p \in P} w_p \left( \phi + \frac{1}{2} \partial_i \partial_j \phi S_{p,ij}(h) + \frac{1}{8} \partial_{ijkl} \phi S_{p,ij} S_{p,kl} \right) \Big|_{x=m_p(h)} + R(h)h^3 \\ &= \sum_{p \in P} w_p \left[ \phi + b_i \partial_i \phi h + \frac{1}{2} (b_i \phi_{ij} b_j + b_i (\partial_i b_j) \partial_j \phi) h^2 + \frac{1}{2} \partial_{ij} \phi G_{ij} h \right. \\ &\quad \left. + \frac{1}{4} \left( 2 \partial_{ijk} \phi b_k G_{ij} + \partial_{ij} \phi \partial_k G_{ij} b_k \right) h^2 + \frac{1}{8} \partial_{ijkl} \phi G_{ij} G_{kl} h^2 \right] \Big|_{x=m_p(0)} \\ &\quad + R(h)h^3. \end{aligned}$$

Using Lemma 4.1, we are able to compute the sums. For example, we find:

$$\begin{aligned} \sum_{p \in P} w_p \frac{1}{2} \partial_{ij} \phi G_{ij} h \Big|_{x=m_p(0)} &= \frac{1}{4} h \lambda_m D_m^2 \phi + \frac{h^2}{4} \sum_{m \neq n} w^1 D_m^2 D_n^2 \phi \gamma \lambda_m \lambda_n \\ &\quad + \frac{1}{2} h^2 w^1 \left( D_m^2 \Lambda_{ij} \partial_{ij} \phi + 2 D_m \Lambda_{ij} D_m \partial_{ij} \phi \right) \gamma \lambda_m \\ &\quad + \frac{1}{4} h^2 w^1 D_m^4 \phi \gamma \lambda_m^2 + R(h)h^3. \end{aligned}$$

Here, we used (4.7) and identities like

$$D_m^2 \partial_{ij} \phi \frac{1}{2} \Lambda_{ij} = \frac{1}{2} D_m^2 D_n^2 \phi \lambda_n.$$

Noting

$$\gamma = 3/2, \quad w^1 \gamma = 1/4,$$

we have after some computation:

$$\mathbb{E}_{x_0}(\phi(X^1)) = \phi(x_0) + Ah + Bh^2 + R(h)h^3, \tag{4.11}$$

where

$$A = b_i \partial_i \phi + \frac{1}{2} \lambda_m D_m^2 \phi,$$

and

$$\begin{aligned} B = & \frac{1}{8} \sum_{m \neq n} D_m^2 D_n^2 \phi \lambda_m \lambda_n + \frac{1}{8} D_m^4 \phi \lambda_m^2 + \frac{1}{4} D_m^2 (b_i \partial_i \phi) \lambda_m \\ & + \frac{1}{2} (b_i \partial_{ij} \phi b_j + b_i \partial_i b_j \partial_j \phi) + \frac{1}{8} (D_m^2 \Lambda_{ij} \partial_{ij} \phi + 2 D_m \Lambda_{ij} D_m \partial_{ij} \phi) \lambda_m \\ & + \frac{1}{4} db_k \partial_k D_m^2 \phi \lambda_m + \frac{1}{4} \partial_{ij} \phi b_k \partial_k \Lambda_{ij}. \end{aligned}$$

Again, by the eigen-decomposition  $\Lambda = \sum_{i=1}^d \lambda_i v_i v_i^T$ , we find

$$\mathcal{L}\phi(x_0) = b_i \partial_i \phi + \frac{1}{2} \Lambda_{ij} \partial_{ij} \phi|_{x=x_0} = A. \tag{4.12}$$

Similarly, we find

$$\begin{aligned} \frac{1}{2} \mathcal{L}^2 \phi(x_0) = & \frac{1}{2} (b_k \partial_k b_i \partial_i \phi + b_k b_i \partial_{ik} \phi + \frac{1}{2} b_k \partial_k \Lambda_{ij} \partial_{ij} \phi + \frac{1}{2} b_k \Lambda_{ij} \partial_{ijk} \phi) \\ & + \frac{1}{4} \Lambda_{kl} (\partial_k \partial_l b_i \partial_i \phi + 2 \partial_k b_i \partial_{il} \phi + b_i \partial_{ikl} \phi + \frac{1}{2} \partial_k \partial_l \Lambda_{ij} \partial_{ij} \phi \\ & + \partial_k \Lambda_{ij} \partial_{ijl} \phi + \frac{1}{2} \Lambda_{ij} \partial_{ijkl} \phi), \end{aligned}$$

which equals to  $B$ . Together with (i), Corollary 2.1 gives the claim. □

REMARK 4.4. For the multi-dimensional Algorithm 2, though we have exponentially many Gaussians, the complexity is just linear in  $d$ . In fact, one needs the number of random variables to grow at least linearly in  $d$  to get a weak second-order scheme for general SDEs [18, 19, 28].

**4.2. Efficiency of the Monte Carlo method.** For the multi-dimensional Algorithm 2, though we have exponentially many Gaussians, we see that the complexity is just linear in  $d$ , which means our algorithm has good computational efficiency. Since we only care about the distributions, we often use Monte Carlo methods [17, 27] to generate a large number of samples and use the empirical measure to approximate the probability measure. As we know, the error and efficiency of Monte Carlo methods depend on the variance. The variance of the Euler-Maruyama scheme (3.1) is  $\Lambda(x_n)h$ , where  $x_n$  is the value of the scheme at  $t_n$ . For the same reason, if we can show that the variance of Algorithm 2 after one step is proportional to  $h$ , then the Monte Carlo method based on our algorithm is as efficient as the Monte Carlo method based on the Euler-Maruyama method (3.1).

In this section, we compute the second moment

$$M_2 := \mathbb{E}(|X^{n+1} - x_n|^2 | X^n = x_n) \tag{4.13}$$

and show that it is indeed  $O(h)$  despite us having exponentially many Gaussians. For the notational convenience, we define the matrix norm

$$\|\Lambda\|_{\text{tr}} = \sup_{x \in \mathbb{R}^d} \text{tr}(|\Lambda(x)|), \tag{4.14}$$

where  $|\Lambda(x)| = \sum_{i=1}^d |\lambda_i(x)| v_i(x) v_i^T(x)$  if  $\Lambda(x)$  is given by (4.1).

PROPOSITION 4.1. *There exists  $h_0 > 0$  such that when  $h < h_0$*

$$M_2 \leq 3\|\Lambda\|_{\text{tr}}h,$$

for Algorithm 2.

*Proof.* By (4.2), direct computation shows that

$$\begin{aligned} M_2 &= \int_{\mathbb{R}} \|x - x_n\|^2 \bar{\rho}(x) dx = \sum_{p \in P} w_p (\text{tr}(S_p(h)) + \|m_p(h) - x_n\|^2) \\ &\leq \sum_{p \in P} w_p \text{tr}(S_p(h)) + 2h^2 \|b\|_{\infty}^2 + 2 \sum_{p \in P} w_p \left\| \sum_{i=1}^d z_p^i \sqrt{\frac{3}{2}} \lambda_i h v_i \right\|^2 \\ &= \sum_{p \in P} w_p \text{tr}(S_p(h)) + 2h^2 \|b\|_{\infty}^2 + \text{tr}(\Lambda(x_n))h. \end{aligned} \tag{4.15}$$

The first inequality in (4.15) follows from (4.9) and (4.10):

$$\begin{aligned} |m_p(h) - x_n|^2 &= |m_p(h) - m_p(0) + m_p(0) - x_n|^2 \leq 2(|m_p(h) - m_p(0)|^2 + |m_p(0) - x_n|^2) \\ &\leq 2h^2 \|b\|_{\infty}^2 + 2 \left\| \sum_{i=1}^d z_p^i \sqrt{\frac{3}{2}} \lambda_i h v_i \right\|^2. \end{aligned}$$

For the last equality, we have by the fact that  $\{v_i\}$ 's are orthonormal:

$$\sum_{p \in P} w_p \left\| \sum_{i=1}^d z_p^i \sqrt{\frac{3}{2}} \lambda_i h v_i \right\|^2 = \frac{3h}{2} \sum_{p \in P} \sum_{i=1}^d w_p |z_p^i|^2 \lambda_i,$$

and the last equality in (4.15) follows since  $\sum_{p \in P} w_p |z_p^i|^2 = 2w_1 = \frac{1}{3}$  (see (D.7)). Now, noticing  $\text{tr}(G(m(t))) \leq \frac{3}{2} \|\Lambda\|_{\text{tr}}$ , we obtain

$$\text{tr}(S_p(h)) \leq \frac{3}{2} h \|\Lambda\|_{\text{tr}}, \quad p \in P.$$

For Algorithm 2, when  $h$  is small enough, we have

$$\text{tr}(S_p(h)) \leq h \|\Lambda\|_{\text{tr}}.$$

Since  $\|b\|_{\infty}^2 h^2$  is in higher order, the claim follows. □

### 5. Numerical experiments

In this section, we apply the algorithm on SDE (1.1) in Itô sense with different choices of  $b$  and  $\sigma$ . Note that the Assumption 2.2  $\sigma, b \in C_b^m$  is only listed for convenience of theoretical analysis. For a diffusion process starting at  $x_0$ , within finite time  $T$ , the probability density is concentrated in a finite domain and the far away behaviors of  $b$  and  $\sigma$  are not important. Hence, in the simulation here, we may use unbounded  $b$  and  $\sigma$ . We also check how the algorithm behaves if there are some degenerate points of  $\Lambda$  (i.e.  $\Lambda$  is only positive semi-definite at these points).



**5.1. A 1D example with regular  $\sigma$ .** This example is designed to test the correctness of Algorithm 1. The dimension is  $d=1$  and  $\sigma^2$  is uniformly bounded from below. We will also plot the empirical distribution generated by our algorithm to compare with the one generated by Euler-Maruyama scheme (3.1).

The SDE we consider is as following:

$$dX(t) = \lambda X(t)dt + \sqrt{X(t)^2 + 4}dW. \tag{5.1}$$

The diffusion coefficient  $\sigma(x) = \sqrt{x^2 + 4}$  is bounded below uniformly so that there is no degenerate point.

To test the correctness of our algorithm, we use the test function  $\phi(x) = x^2$  and define the relative error as

$$E = \frac{1}{\mathbb{E}X^2(T)} \left| \frac{1}{N} \sum_{k=1}^N (X^{(k),[T/h]} )^2 - \mathbb{E}X^2(T) \right|, \tag{5.2}$$

where  $X^{(k)} = \{X^{(k),n}\}_{n \geq 0}$  is the sequence generated by the numerical algorithm in the  $k$ -th experiment. Hence,  $X^{(k)}$  is a sample path. The exact expectation  $\mathbb{E}_{x_0} X^2(T)$ , by Itô's formula [23, Chap. 4], is given by

$$\mathbb{E}_{x_0} X^2(T) = x_0^2 \exp((2\lambda + 1)T) + 4 \frac{\exp((2\lambda + 1)T) - 1}{2\lambda + 1}.$$

In Figure 5.1, we plot the results of the simulation for  $X(0) = 2, \lambda = -2$  and  $T = 2$ . Each error is computed using  $N = 10^8$  trajectories. The "error bars" are obtained by chopping all samples into 10 slices, with each slice containing  $10^7$  trajectories. We then compute the relative error (5.2) in each slice, denoted by  $E^{(m)}$  ( $1 \leq m \leq 10$ ). We find the standard deviation  $\sigma_E$  for the data  $\{E^{(m)}\}_{m=1}^{10}$ , and use  $[E - 1.65\sigma_E, E + 1.65\sigma_E]$  as our confidence interval. We find that our Gaussian mixture method gives weak second-order accuracy.

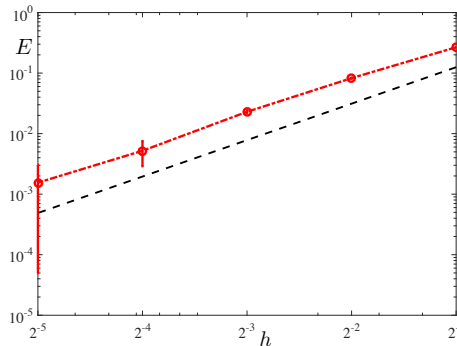


FIG. 5.1.  $X(0) = 2, \lambda = -2, T = 2$ . We plot the errors obtained by the Gaussian mixture method. The vertical short segments are the "error bars" and the black dashed line indicates  $E = h^2$ .

To confirm that the Gaussian mixture method gives the desired distribution, we now plot the empirical distribution in Figure 5.2 by histcounts. All the empirical densities are obtained by using  $N = 10^6$  points, and the initial condition  $X(0) = 2$ . We take the results obtained from Euler-Maruyama (E-M) scheme (3.1) with  $\Delta t = h^3$  as the reference density (green curves in Figure 5.2 (a) and (b)).

In Figure 5.2 (a), we plot the empirical densities obtained by Algorithm 1 (red) and Euler-Maruyama (black) after one step with step size  $\Delta t = h = 1/32$ . At time  $t = h$ , the reference density (green curve) has a peak at  $x_c \approx 1.79$  while its mean is located at the black dot ( $\bar{x} \approx 1.88$ ). We also calculated the empirical skewness  $\gamma_1 = \mathbb{E}\left(\frac{X(h) - \bar{x}}{\sigma}\right)^3 \approx 0.3695$  (here only  $\sigma$  denotes the variance of the reference density), and the kurtosis  $K = \mathbb{E}\left(\frac{X(h) - \bar{x}}{\sigma}\right)^4 \approx 3.3078$ , while the accurate skewness and kurtosis are 0.3718 and 3.3153 respectively. The skewness and kurtosis for a Gaussian (Euler-Maruyama method) are 0 and 3 respectively. For Algorithm 1, these two numbers are 0.3717 and 3.1888.

In Figure 5.2 (b), we plot the empirical densities obtained by Algorithm 1 (red) and Euler-Maruyama (black) at time  $t = 1$  with step size  $\Delta t = h = 1/32$ . We find that the densities given by our weak second-order algorithm almost coincides with the reference density, while the one given by E-M is worse.

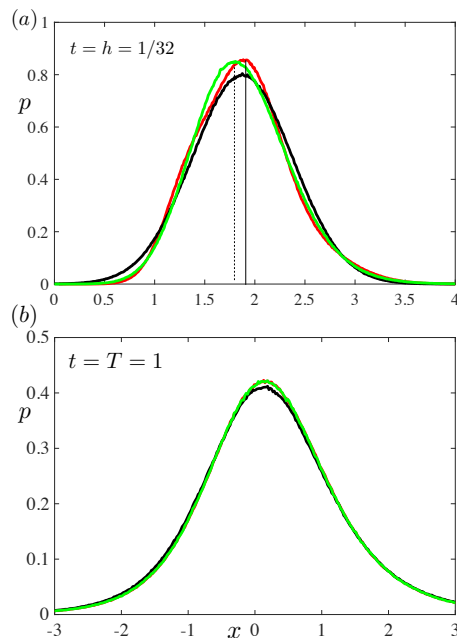


FIG. 5.2. The green lines are the ‘accurate densities’ obtained using E-M with step size  $\Delta t = h^3$ . Other curves are empirical distributions (black dashed line: E-M; red broken line: Gaussian mixture) obtained with step size  $\Delta t = h$ . (a). Empirical distributions after one step. The solid vertical line shows the mean of the green curve while the dashed line shows the peak. Empirical skewness is 0.3695 and kurtosis is 3.3078. (b). Empirical distributions are at  $t = 1$ .

To sum up, for this example (5.1), the Gaussian mixture method has weak second-order accuracy and is able to capture the correct distribution better.

**5.2. 1D geometric Brownian motion.** In this example, we consider the 1D geometric Brownian motion

$$dX(t) = \lambda X(t) dt + \sigma X(t) dW,$$

which has a degenerate diffusion coefficient

$$\sigma(x) = \sigma x.$$

Again, we test the weak accuracy with test function  $\phi(x) = x^2$  and define the weak error

$$E = \frac{1}{\mathbb{E}X^2(T)} \left| \frac{1}{N} \sum_{k=1}^N (X^{(k),[T/h]})^2 - \mathbb{E}X^2(T) \right|.$$

By Itô calculus, it is straightforward to find

$$\mathbb{E}X^2(T) = x_0^2 \exp((2\lambda + \sigma^2)T).$$

In Figure 5.3, we plot the weak error of simulations for  $\lambda = -0.8, \sigma = 0.85, x_0 = 5, T = 1$  with  $N = 2 \times 10^8$ . The error bars are computed by slicing the samples into 5 pieces of equal size, and the method is the same as in Section 5.1 (confidence interval is  $[E - 1.65\sigma_E, E + 1.65\sigma_E]$ ).

For the tested parameters our Gaussian mixture method still works and is of weak second-order. For this example, the error of Algorithm 1 scales like  $h^2$  only when  $h$  becomes small. This can be seen in the kink in Figure 5.3 where only the left-most two points seem to line up with the order  $h^2$  line. After further investigation, we find that for the first three  $h$  values ( $h = 0.25, 0.125, 0.0833$ ), there is roughly 1/6 chance that the computed  $S(h)$  from the ODE is negative. For smaller values of  $h$  ( $h = 0.0625, 0.05$ ),  $S(h)$  is always nonnegative for the samples we have. In light of this, we expect the second-order behavior for our approach to appear in the examples with  $h \lesssim 0.0625$ . When there is a reasonable chance that  $\sigma^2$  is degenerate, our approach seems to lose the second-order accuracy.

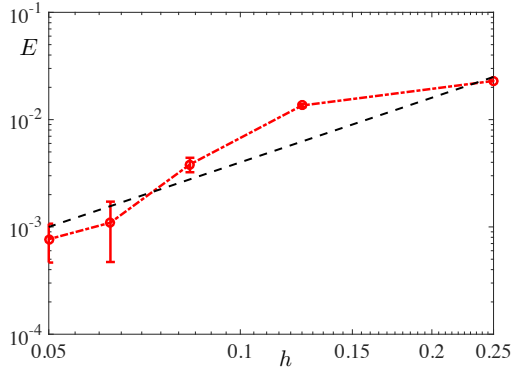


FIG. 5.3. Log-Log error plot for geometric Brownian motion, with  $X(0) = 5, \lambda = -0.8, \sigma = 0.85, T = 1$ . The red line with circles is the error obtained by the Gauss mixture method. The black dashed line is  $E = 0.4h^2$  and vertical bars represent the “error bars”.

**5.3. A 2D example.** In this example, we consider a 2D SDE, which is a modification of the first example in [2]:

$$\begin{pmatrix} dX_1(t) \\ dX_2(t) \end{pmatrix} = \begin{pmatrix} X_1(t) \\ -X_2(t) \end{pmatrix} + X_1(t) \begin{pmatrix} 0 \\ 1 \end{pmatrix} dW_1(t) + \sigma \begin{pmatrix} 1 \\ 1 \end{pmatrix} dW_2(t), \tag{5.3}$$

where  $W_1(t)$  and  $W_2(t)$  are independent standard Brownian motions, and  $\sigma$  is a positive constant. The purpose here is to show that our Gaussian mixture method for multi-dimensions (Algorithm 2) works for  $\Lambda(x)$  that has varying eigen-directions.

We consider the solution of (5.3) at  $T=1$  with initial condition  $X_1(0)=X_2(0)=1$  and  $\sigma=0.1$ . We will use the test function  $\phi(x)=x_2^2$  to check the weak accuracy. By Itô’s formula,

$$\mathbb{E}X_2^2(t) = e^{-2t} \left( \mathbb{E}X_2^2(0) - \frac{1}{4} \mathbb{E}X_1^2(0) - \frac{3\sigma^2}{8} \right) + \frac{\sigma^2}{4} + e^{2t} \left( \frac{1}{4} \mathbb{E}X_1^2(0) + \frac{\sigma^2}{8} \right).$$

As before, the relative error is computed as

$$E = \frac{1}{\mathbb{E}X_2^2(T)} \left| \frac{1}{N} \sum_{k=1}^N (X_2^{(k),[T/h]})^2 - \mathbb{E}X_2^2(T) \right|.$$

In Figure 5.4, we sketch the error plots with  $N=2 \times 10^8$  and also slice these samples into 10 equal pieces for the “error bar” calculation (confidence interval  $[E - 1.65\sigma_E, E + 1.65\sigma_E]$  and  $\sigma_E$  is the standard deviation for these 10 data). We find that our Gaussian mixture method gives weak second-order accuracy for this 2D example as well.

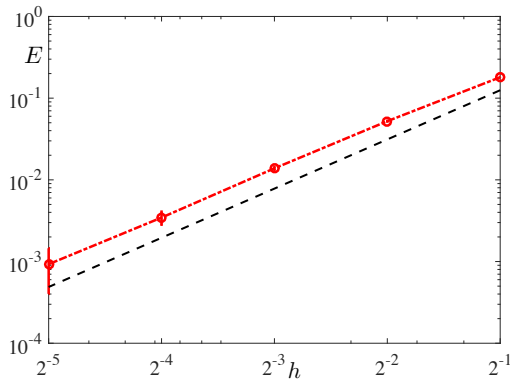


FIG. 5.4. Log-log error plot of the Gaussian mixture method for the 2D example with  $\sigma=0.1$  (red line). The black line is  $E=0.5h^2$  while the vertical segments are the “error bars”.

**5.4. A 6D example.** According to Algorithm 2, the proposed Gaussian mixture method depends explicitly on the dimension and one is surely curious with what will happen if the dimension gets higher. In this example, we look at a 6D problem and verify that our algorithm is still weak second-order.

The SDE we consider is given by:

$$d \begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \end{pmatrix} = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 & -1 \\ -1 & -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & -1 & 1 \\ 1 & 0 & 0 & 0 & -1 & -1 \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \end{pmatrix} dt + \sigma \begin{pmatrix} \sqrt{0.1+X_1^2} & -0.1 & 0 & 0 & 0 & -0.1 \\ -0.1 & \sqrt{0.2+X_2^2} & -0.1 & 0 & 0 & 0 \\ 0 & -0.1 & \sqrt{0.3+X_3^2} & -0.1 & 0 & 0 \\ 0 & 0 & -0.1 & \sqrt{0.4+X_4^2} & -0.1 & 0 \\ 0 & 0 & 0 & -0.1 & \sqrt{0.5+X_5^2} & -0.1 \\ -0.1 & 0 & 0 & 0 & -0.1 & \sqrt{0.6+X_6^2} \end{pmatrix} d \begin{pmatrix} W_1 \\ W_2 \\ W_3 \\ W_4 \\ W_5 \\ W_6 \end{pmatrix} \quad (5.4)$$

We take  $\sigma=0.7$  and check the solution at  $t=2$ . The initial condition we use is  $X_i(0)=1$  for all  $1 \leq i \leq 6$ . The test function we use is

$$\phi(x) = \sum_{i=1}^6 x_i^2. \tag{5.5}$$

By Itô’s formula

$$\mathbb{E}\phi(X(T)) = \left(\sum_{i=1}^6 X_i^2(0)\right)\exp((-2+\sigma^2)t) + \frac{2.22\sigma^2}{\sigma^2-2}(\exp((-2+\sigma^2)t)-1). \tag{5.6}$$

The relative error is again defined as

$$E = \frac{1}{\mathbb{E}\phi(X(T))} \left| \frac{1}{N}\phi(X^{(k),[T/h]}) - \mathbb{E}\phi(X(T)) \right|. \tag{5.7}$$

For the following log-log error plot (Figure 5.5), we choose  $h = \frac{1}{4k}$ ,  $1 \leq k \leq 5$ . The sample size is  $N = 2 \times 10^8$  for  $h \geq \frac{1}{16}$  and  $5 \times 10^8$  for  $h = \frac{1}{20}$ , chopped into 10 equal slices to produce the error bars with confidence interval  $[E - 1.65\sigma_E, E + 1.65\sigma_E]$  ( $\sigma_E$  is again the standard deviation of these 10 data). The plot demonstrates that the scheme works in high dimensions as well.

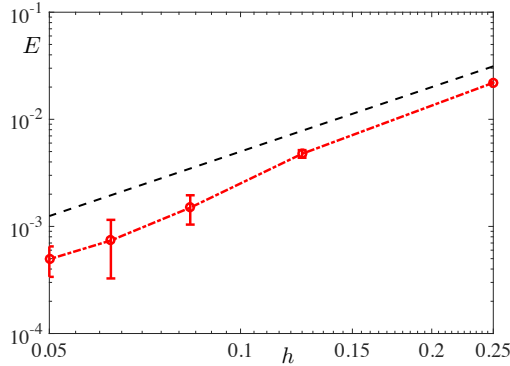


FIG. 5.5. Log-log error plot of Gaussian mixture method (red line). The black dashed line is  $E=0.5h^2$  and the vertical segments indicate the “error bars”.

**Acknowledgements.** The work of L. Li is partially sponsored by NSFC 11901389,11971314, and Shanghai Sailing Program 19YF1421300. The work of J. Lu and L. Wang is supported in part by National Science Foundation under grant DMS-1454939, while J. Mattingly is supported in part by National Science Foundation under grant DMS-1613337.

**Appendix A. Proof of Proposition 2.1.**

*Proof.* Let us fix  $\phi \in C_b^{2(r+1)}$  and define

$$u^n(x) = \mathbb{E}_x\phi(X^n),$$

$$u(x,t) = \mathbb{E}_x\phi(X(t)).$$

By the Markov property, we have

$$u^{n+1}(x) = \mathbb{E}_x(\mathbb{E}_x(\phi(X^{n+1})|X^1)) = \mathbb{E}_x(u^n(X^1)). \tag{A.1}$$

Similarly, we have

$$u(x, (n+1)h) = \mathbb{E}_x(u(X(t_1), nh)). \tag{A.2}$$

Note that  $u$  satisfies the backward Kolmogorov equation

$$u_t = \mathcal{L}u = b \cdot \nabla u + \frac{1}{2} \Lambda_{ij} \partial_{ij} u,$$

with initial condition

$$u(x, 0) = \phi(x).$$

By standard parabolic PDE theory, for  $b, \sigma \in C_b^{2(r+1)}$ , we have

$$\sup_{0 \leq t \leq T} \|u\|_{C^{2(r+1)}} \leq C(T). \tag{A.3}$$

By Equations (A.1) and (A.2), we have for all  $x \in \mathbb{R}^d$  that

$$\begin{aligned} |u^{n+1}(x) - u(x, (n+1)h)| &\leq |\mathbb{E}_x(u^n(X^1) - u(X^1, nh))| \\ &\quad + |\mathbb{E}_x(u(X^1, nh)) - \mathbb{E}_x(u(X(t_1), nh))|. \end{aligned}$$

Define

$$E^n = \sup_{x \in \mathbb{R}^d} |u^n(x) - u(x, nh)|,$$

by the assumption of Proposition 2.1 on local truncation error and Equation (A.3) we have

$$E^{n+1} \leq \mathbb{E}_x E^n + |\mathbb{E}_x(u(X^1, nh)) - \mathbb{E}_x(u(X(t_1), nh))| \leq E^n + Ch^{r+1},$$

where  $C = \sup_{0 \leq t \leq T} \rho(\|u(\cdot, t)\|_{C^{2(r+1)}})$ . This further implies that

$$\sup_{n: nh \leq T} E^n \leq C_1 h^r.$$

□

**Appendix B. Proof of Proposition 3.1.**

*Proof.* For the convenience of notations, we will drop the dependence on  $x_0$  so that  $m(h)$  indeed means  $m(h, x_0)$  and  $m_1$  means  $m_1(x_0)$  and so on. Denote  $\mathcal{L}_1 := (m(h) - x_0)\partial_x + \frac{1}{2}S(h)\partial_{xx}$ , and we have

$$\mathbb{E}_{x_0}(\phi(X^1)) = \phi(x_0) + \mathcal{L}_1\phi(x_0) + \frac{1}{2}\mathcal{L}_1^2\phi(x_0) + O(h^3).$$

It follows that

$$\mathbb{E}_{x_0}(\phi(X^1)) =: \phi(x_0) + Bh + Ch^2 + O(h^3). \tag{B.1}$$

Here,  $B$  and  $C$  are the coefficients of  $h$  and  $h^2$ :

$$\begin{aligned}
 B &= \phi'(x_0)m_1 + \frac{1}{2}\phi''(x_0)S_0, \\
 C &= \frac{1}{2}(\phi''(x_0)m_1^2 + \phi'(x_0)m_2) + \frac{1}{2}\left(\frac{1}{2}\phi''(x_0)S_1 + \phi'''(x_0)m_1S_0\right) + \frac{1}{8}\phi^{(4)}(x_0)S_0^2.
 \end{aligned}$$

To satisfy the condition (2.13), we need to have

$$B = \mathcal{L}\phi(x_0), \quad C = \frac{1}{2}\mathcal{L}^2\phi(x_0). \tag{B.2}$$

Recall that  $\mathcal{L} = b\partial_x + \frac{1}{2}\Lambda(x)\partial_x^2$ , so  $B = \mathcal{L}\phi(x_0)$  requires that for any sufficiently smooth  $\phi$ ,

$$b(x_0)\phi'(x_0) + \frac{1}{2}\Lambda(x_0)\phi''(x_0) = \phi'(x_0)m_1 + \frac{1}{2}\phi''(x_0)S_0,$$

which requires

$$m_1 = b(x_0), \quad S_0 = \Lambda(x_0). \tag{B.3}$$

On the other hand, the requirement  $C = \frac{1}{2}\mathcal{L}^2\phi(x_0)$  can be expanded as

$$\begin{aligned}
 &\frac{1}{2}\left(b(x_0)(b\phi' + \frac{1}{2}\Lambda\phi'')' + \frac{1}{2}\Lambda(x_0)(b\phi' + \frac{1}{2}\Lambda\phi'')''\right)\Big|_{x=x_0} \\
 &= \left(\frac{1}{2}(\phi''m_1^2 + \phi'm_2) + \frac{1}{2}\left(\frac{1}{2}\phi''S_1 + \phi'''m_1S_0\right) + \frac{1}{8}\phi^{(4)}S_0^2\right)\Big|_{x=x_0}.
 \end{aligned}$$

This is impossible in general. For example, the coefficient of  $\phi'''$  on right-hand side is  $\frac{1}{2}m_1S_0$ , or  $\frac{1}{2}b(x_0)\Lambda(x_0)$  but the one on left-hand side is  $\frac{1}{2}b(x_0)\Lambda(x_0) + \frac{1}{4}\Lambda(x_0)\Lambda'(x_0)$ . They can not balance unless the diffusion matrix  $\Lambda(x)$  is constant.  $\square$

**Appendix C. Proof of Lemma 4.1.**

*Proof.* In this proof, we will again use  $R$  to denote a generic function that can depend on the  $C^6$  norm of the test function but can be bounded uniformly in  $x_0$  and  $h$ . However, its concrete meaning can change from line to line.

Clearly, due to the symmetry, we only need to prove that for all  $\phi \in C_b^\infty$ ,

$$\begin{aligned}
 \sum_{p \in P} w_p \phi(x_p) &= \phi(x_0) + \sum_{i=1}^d w^1 D_i^2 \phi(x_0) \gamma \lambda_i h + \sum_{i < j} (w^1)^2 D_i^2 D_j^2 \phi(x_0) \gamma^2 \lambda_i \lambda_j h^2 \\
 &\quad + \frac{1}{12} \sum_{i=1}^d w^1 D_i^4 \phi(x_0) \gamma^2 \lambda_i^2 h^2 + R(x_0, h) h^3.
 \end{aligned} \tag{C.1}$$

Without loss of generality, we set  $x_0 = 0$ . With Equation (4.4), it is convenient to denote the left-hand side of (C.1) as

$$T_d(\phi) = \sum_{p \in P} \left(\prod_{i=1}^d w^{z_p^i}\right) \phi\left(\sum_{i=1}^d z_p^i \sqrt{\gamma \lambda_i} h v_i\right). \tag{C.2}$$

If  $d = 1$ , the claim follows from 1D Taylor expansion derived in Section 3. Assume that the claim is valid for all  $d = 1, 2, \dots, m$ ,  $m \geq 1$ , and we want to prove for  $d = m + 1$ . Define  $P_m$  to be the index set with  $d = m$ . We find by definition:

$$T_{m+1}(\phi) = w^1 \sum_{p \in P_m} \left(\prod_{i=1}^m w^{z_p^i}\right) \phi\left(\sum_{i=1}^m z_p^i \sqrt{\gamma \lambda_i} h v_i + \sqrt{\gamma \lambda_{m+1}} h\right)$$

$$\begin{aligned}
 &+ w^0 \sum_{p \in P_m} \left( \prod_{i=1}^m w^{z_p^i} \right) \phi \left( \sum_{i=1}^m z_p^i \sqrt{\gamma \lambda_i h} v_i \right) \\
 &+ w^{-1} \sum_{p \in P_m} \left( \prod_{i=1}^m w^{z_p^i} \right) \phi \left( \sum_{i=1}^m z_p^i \sqrt{\gamma \lambda_i h} v_i - \sqrt{\gamma \lambda_{m+1} h} \right).
 \end{aligned}$$

For each  $p$ , we do Taylor expansion of  $\phi$  about  $\sum_{i=1}^m z_p^i \sqrt{\gamma \lambda_i h} v_i$  and have

$$T_{m+1}(\phi) = T_m(\phi) + w^1 T_m(D_{m+1}^2 \phi) \gamma \lambda_{m+1} h + \frac{1}{12} w^1 T_m(D_{m+1}^4 \phi) \gamma^2 \lambda_{m+1}^2 h^2 + Rh^3. \tag{C.3}$$

By the induction hypothesis, we have

$$\begin{aligned}
 T_{m+1}(\phi) &= \phi(0) + \sum_{i=1}^m w^1 D_i^2 \phi \gamma \lambda_i h + \sum_{i < j} (w^1)^2 D_i^2 D_j^2 \gamma^2 \lambda_i \lambda_j h^2 + \frac{1}{12} \sum_{i=1}^m w^1 D_i^4 \phi \gamma^2 \lambda_i^2 h^2 \\
 &+ w^1 (D_{m+1}^2 \phi + \sum_{i=1}^m w^1 D_i^2 D_{m+1}^2 \phi \gamma \lambda_i h) \gamma \lambda_{m+1} h \\
 &+ \frac{1}{12} w^1 D_{m+1}^4 \phi \gamma^2 \lambda_{m+1}^2 h^2 + Rh^3.
 \end{aligned}$$

Arranging the terms on the right-hand side, we find the claim is also true for  $d = m + 1$ . □

**Appendix D. A variance construction approach.**

**D.1. The variance construction method for one dimension.** Motivated by (3.21) and (3.22), we can construct

$$\begin{aligned}
 S_i(h) &= \frac{h}{2} \Lambda(x_0 + z_i \sqrt{6\Lambda(x_0)h} + hb(x_0)), \quad i = \pm 1, \\
 S_0(h) &= \frac{1}{2} h \Lambda(x_0 + b(x_0)h) - \frac{3}{8} \Lambda(x_0) \Lambda''(x_0) h^2 + \frac{(3\Lambda(x_0) \Lambda''(x_0)/8)^2}{\Lambda(x_0 + b(x_0)h)/2} h^3.
 \end{aligned}$$

We can verify that the constraints are all satisfied. The third term added is to ensure that  $S_0$  is non-negative. Compared with the ODE flow method, the drawback of this method is that it involves higher order spatial derivatives, such as  $\Lambda''$ . In practice, one may approximate it by finite difference  $\frac{1}{h^2} (\Lambda(x_0 + h) - 2\Lambda(x_0) + \Lambda(x_0 - h))$ .

REMARK D.1. The third correction term can be thrown away if  $h$  is small enough. For example,

$$h < \frac{4 \inf_x |\Lambda(x)|}{3 \|\Lambda \Lambda''\|_\infty}.$$

This construction gives the following Algorithm 3 to generate  $x_{n+1}$  given  $X^n = x_n$ . One can verify that the requirements in (3.15) are satisfied, which gives the following theorem:

**THEOREM D.1.** *Let  $d = 1$ . Suppose Assumptions 2.1-2.2 hold, then Algorithm 3 is a weak second-order scheme for the 1D diffusion process (1.1).*

The proof is identical to that of Theorem 3.1 and is omitted here.



---

**Algorithm 3** Gaussian mixture scheme for SDEs (variance construction method in 1D)

---

1: Generate  $z$  such that  $P(z=0) = \frac{2}{3}$  and  $P(z=1) = P(z=-1) = \frac{1}{6}$ . Then,

$$m(0) = x_n + z\sqrt{\frac{3}{2}\Lambda h}$$

If  $z=0$ ,

$$S(h) = \frac{1}{2}h\Lambda(x_n + b(x_n)h) - \frac{3}{8}\Lambda(x_n)\Lambda''(x_n)h^2 + \frac{(3\Lambda(x_n)\Lambda''(x_n)/8)^2}{\Lambda(x_n + b(x_n)h)/2}h^3.$$

Otherwise,

$$S(h) = \frac{h}{2}\Lambda(x_n + z\sqrt{6\Lambda(x_n)h} + hb(x_n)).$$

2: Solve the ODE  $\dot{m} = b(m)$  with the initial value  $m(0)$  using a scheme of at least second order to obtain  $m(h)$ .

3: Sample

$$x_{n+1} = m(h) + \sqrt{S(h)}\xi, \tag{D.1}$$

where  $\xi$  is a standard 1D normal variable.

---

**D.2. The variance construction method for multi-dimensions.** As before, one may want to guarantee that  $S_p(h)$  is positive definite for  $p \in P$ . We now present a variance construction method for  $S_p(h)$  for multi-dimensions. Consider that  $m_p(h)$  and  $S_p(h)$  are given by

$$\begin{aligned} \dot{m}_p &= b(m_p(t)), \quad m_p(0) = y_p, \\ S_p(h) &= \frac{h}{2}\Lambda\left(x_0 + \sum_{i=1}^d z_p^i \sqrt{6\lambda_i} h v_i + hb\right) - \frac{3h^2}{8} \sum_{i=1}^d (1 - |z_p^i|) \lambda_i D_i^2 \Lambda + F_p(h)h^3, \end{aligned}$$

where  $\sum_{i=1}^d (1 - |z_p^i|) \lambda_i D_i^2 \Lambda$  can be approximated by finite difference. In particular, if we set  $\theta = \sum_{i=1}^d \sqrt{(1 - |z_p^i|) \lambda_i} v_i$ , then

$$\sum_{i=1}^d (1 - |z_p^i|) \lambda_i D_i^2 \Lambda \approx \frac{1}{h^2} \left( \Lambda(x_0 + h\theta) - 2\Lambda(x_0) + \Lambda(x_0 - h\theta) \right).$$

$F_p(h)h^3$  is added to ensure that  $S_p$  is positive semi-definite. Let the first two terms in  $S_p$  be  $hA_p$  and  $h^2B_p$ , where  $A_p$  is positive definite and thus invertible. Then, we have

$$F_p = \frac{1}{4}B_p A_p^{-1} B_p. \tag{D.2}$$

We propose Algorithm 4 to generate  $x_{n+1}$  given  $X^n = x_n$ .

REMARK D.2. Notice that we need to invert a matrix to get  $F(h)$ , which is not desirable when  $d$  is large. However, similar as the 1D case, if  $h$  is small enough,  $F(h)h^3$  can be thrown away and we can still guarantee the positive definiteness.

**Algorithm 4** Gaussian mixture scheme for SDEs (variance construction method in higher D)

1: Using some fast algorithm, decompose

$$\Lambda(x_n) = \sum_{i=1}^d \lambda_i(x_n) v_i(x_n) v_i^T(x_n).$$

2: Generate  $z^i, i=1, 2, \dots, d$  so that  $P(z^i = 0) = \frac{2}{3}$  while  $P(z^i = 1) = P(z^i = -1) = \frac{1}{6}$ .

3: Construct the matrix

$$S(h) = \frac{h}{2} \Lambda(x_n) + \sum_{i=1}^d z^i \sqrt{6\lambda_i h v_i} - \frac{3h^2}{8} \sum_{i=1}^d (1 - |z^i|) \lambda_i D_i^2 \Lambda + F(h) h^3 \tag{D.3}$$

where  $F$  is constructed according to (D.2).

4: Let

$$m(0) = x_n + \sum_{i=1}^d z^i \sqrt{\frac{3}{2} \lambda_i h v_i}.$$

and then  $m(h)$  is obtained by solving  $\dot{m} = b(m)$  using an ODE solver with at least second-order accuracy (e.g. RK2, RK4).

5: Sample  $x_{n+1} \sim \mathcal{N}(m(h), S(h))$ , or

$$x_{n+1} = m(h) + \sum_{i=1}^d \sqrt{\mu_i^+} \xi_i u_i.$$

where  $S(h) = \sum_{i=1}^d \mu_i u_i u_i^T$  with  $\{u_i\}_{i=1}^d$  being orthonormal, and  $\{\xi_i\}$  are i.i.d standard 1D normal variables.

**THEOREM D.2.** *Suppose Assumptions 2.1-2.2 hold, then Algorithm 4 is a second-order scheme for the multi-dimensional diffusion process (1.1).*

*Proof.* Again, the idea is to check the conditions in Corollary 2.1. Our strategy is not to verify the conditions directly, instead, we compare it to Algorithm 2, the one using an ODE approach.

Again, we only have to check  $X^1$  given  $X^0 = x_0$ . Let  $S_p^o$  be the covariance matrix obtained following Algorithm 2 at time  $h$  while  $S_p^s$  be the covariance matrix constructed in this section at time  $h$  for  $p \in P$ . Let  $\mathbb{E}_{x_0}^s$  denote the expectation under the process constructed here while  $\mathbb{E}_{x_0}^o$  be the expectation under the process in Algorithm 2.

Consider

$$E = \mathbb{E}_{x_0}^s(\phi(X^1)) - \mathbb{E}_{x_0}^o(\phi(X^1)) =: E_1 - E_2.$$

Since the two algorithms only give different covariance matrices, we have by Equation (4.2):

$$|E| \leq \left| \frac{1}{2} \sum_{p \in P} w_p \partial_i \partial_j \phi(m_p(h)) (S_{p,ij}^o - S_{p,ij}^s) \right|$$

$$+ \frac{1}{8} \sum_{p \in P} w_p \partial_{ijkl} \phi(m_p(h)) (S_{p,ij}^o S_{p,kl}^o - S_{p,ij}^s S_{p,kl}^s) \Big| + R(h)h^3. \tag{D.4}$$

We denote

$$\eta_p = (m_p(0) - x_0) / \sqrt{h} = \sum_{i=1}^d z_p^i \sqrt{\frac{3}{2}} \lambda_i v_i. \tag{D.5}$$

By Equation (4.6) and direction Taylor expansion on  $t$ , we find for  $p \in P$ ,

$$\begin{aligned} S_p^o &= G(m_p(0))h + \frac{1}{2}b(m_p(0)) \cdot \nabla G(m_p(0))h^2 + R(h)h^3 \\ &= \frac{1}{2}h\Lambda + \eta_p \cdot \nabla G h^{3/2} + \frac{1}{2}(\eta_p)_i (\eta_p)_j \partial_{ij} G h^2 + \frac{1}{2}b \cdot \nabla G h^2 + K_p^1(h)h^{5/2} + R(h)h^3, \end{aligned}$$

where  $K_p^1(h)$  is a bounded function.

We do expansion on  $S_p^s$  and have

$$\begin{aligned} S_p^s &= \frac{1}{2}h\Lambda + \eta_p \cdot \nabla G h^{3/2} + \frac{1}{2}b \cdot \nabla \Lambda h^2 + (\eta_p)_i (\eta_p)_j \partial_{ij} \Lambda h^2 - \frac{3h^2}{8} \sum_{i=1}^d (1 - |z_p^i|) \lambda_i D_i^2 \Lambda \\ &\quad + K_p^2(h)h^{5/2} + R(h)h^3, \end{aligned}$$

where  $K_p^2$  is some bounded function. This implies

$$S_{p,ij}^o - S_{p,ij}^s = \frac{3h^2}{8} \sum_{i=1}^d (1 - |z_p^i|) \lambda_i D_i^2 \Lambda - \frac{1}{2}(\eta_p)_i (\eta_p)_j \partial_{ij} \Lambda h^2 + K_p^3 h^{5/2} + R(h)h^3. \tag{D.6}$$

Hence, we can replace  $m_p(h)$  with  $y_p$  and throw away the terms involving  $\partial_{ijkl} \phi$  in (D.4) with introducing errors at most  $R(h)h^3$ :

$$|E| \leq \left| \frac{1}{2} \sum_{p \in P} w_p \partial_i \partial_j \phi(y_p) (S_{p,ij}^o - S_{p,ij}^s) \right| + R(h)h^3.$$

By (D.6) and (D.5), we find

$$\begin{aligned} &\sum_{p \in P} w_p \partial_i \partial_j \phi(y_p) (S_{p,ij}^o - S_{p,ij}^s) \\ &= \frac{h^2}{2} \partial_i \partial_j \phi(x_0) \sum_{p \in P} w_p \left( \frac{3}{8} \sum_{i=1}^d (1 - |z_p^i|) \lambda_i D_i^2 \Lambda - \frac{3}{4} \sum_{m,n} z_p^m z_p^n \sqrt{\lambda_m \lambda_n} D_m D_n \Lambda \right) + R(h)h^{5/2}. \end{aligned}$$

We note first that

$$\sum_{p \in P} w_p (1 - |z_p^i|) = w^0 = \frac{2}{3}, \quad \sum_{p \in P} w_p z_p^m z_p^n = 2w^1 \delta_{mn} = \frac{1}{3} \delta_{mn}. \tag{D.7}$$

We justify the second equality for an example. Let  $j \in \{\pm 1, 0\}$  be the index over the beams in one dimension, and  $z_j = j$ . Then, when  $m = n$ ,

$$\sum_{p \in P} w_p z_p^m z_p^n = \left( \sum_{j=-1}^1 |z_j|^{2w^j} \right) \prod_{i \neq m} \left( \sum_{j=-1}^1 w^j \right) = \sum_{j=-1}^1 |z_j|^2 w^j = 2w^1.$$

When  $m \neq n$ ,

$$\sum_{p \in P} w_p z_p^m z_p^n = \left( \sum_{j=-1}^1 z_j w^j \right)^2 \prod_{i \neq m, i \neq n} \left( \sum_{j=-1}^1 w^j \right) = \left( \sum_{j=-1}^1 z_j w^j \right)^2 = 0.$$

Using (D.7), we find that

$$\sum_{p \in P} w_p \partial_i \partial_j \phi(y_p) (S_{p,ij}^o - S_{p,ij}^s) = R(h) h^{5/2}.$$

Therefore

$$|E| \leq R(h) h^{5/2}.$$

However, we know that  $E_1$  does not contain  $h^{5/2}$  terms while neither does  $E_2$  because of the symmetry. Hence,  $|E| \leq R(h) h^3$ , which finishes the proof.  $\square$

#### REFERENCES

- [1] A. Abdulle, D. Cohen, G. Vilmart, and K.C. Zygalakis, *High weak order methods for stochastic differential equations based on modified equations*, SIAM J. Sci. Comput., **34(3):A1800–A1823, 2012. 1**
- [2] D.F. Anderson and J.C. Mattingly, *A weak trapezoidal method for a class of stochastic differential equations*, Commun. Math. Sci., **9(1):301–318, 2011. 1, 2.1, 2.1, 3.2, 5.3**
- [3] C. Archambeau, D. Cornford, M. Opper, and J. Shawe-Taylor, *Gaussian process approximations of stochastic differential equations*, in N.D. Lawrence, A. Schwaighofer, and J.Q. Candela (eds.), *Gaussian Processes in Practice*, Proc. Mach. Learn. Res., **1–16, 2007. 1**
- [4] F. Black and M. Scholes, *The pricing of options and corporate liabilities*, J. Political Econ., **81(3):637–654, 1973. 1**
- [5] W.T. Coffey and Y.P. Kalmykov, *The Langevin Equation: With Applications to Stochastic Problems in Physics, Chemistry and Electrical Engineering*, World Scientific, **27, 2012. 1**
- [6] W. E, T. Li, and E. Vanden-Eijnden, *Applied Stochastic Analysis*, Amer. Math. Soc., **199, 2019. 1, 1**
- [7] Y. Feng, L. Li, and J.-G. Liu, *Semigroups of stochastic gradient descent and online principal component analysis: properties and diffusion approximations*, Commun. Math. Sci., **16(3):777–789, 2018. 1**
- [8] E. Hille and R.S. Phillips, *Functional Analysis and Semi-groups*, Amer. Math. Soc., **31, 1996. 2.1, 2.1**
- [9] W. Hu, C.J. Li, L. Li, and J.-G. Liu, *On the diffusion approximation of nonconvex stochastic gradient descent*, Ann. Math. Sci. Appl., **4(1):3–32, 2019. 1**
- [10] P.E. Kloeden and E. Platen, *Numerical Solution of Stochastic Differential Equations*, Springer-Verlag Berlin Heidelberg New York, **1992. 1, 2.1, 3**
- [11] D. Lamberton and B. Lapeyre, *Introduction to Stochastic Calculus Applied to Finance*, CRC Press, **2007. 1**
- [12] L. Li, J.-G. Liu, and J. Lu, *Fractional stochastic differential equations satisfying fluctuation-dissipation theorem*, J. Stat. Phys., **169(2):316–339, 2017. 1**
- [13] Q. Li, C. Tai, and W. E, *Stochastic modified equations and adaptive stochastic gradient algorithms*, in Proceedings of the 34th International Conference on Machine Learning-Volume 70, **2101–2110, 2017. 1**
- [14] Y. Lu, A. Stuart, and H. Weber, *Gaussian approximations for probability measures on  $\mathbb{R}^d$* , SIAM/ASA J. Uncertain. Quantif., **5(1):1136–1165, 2017. 1**
- [15] Y. Lu, A. Stuart, and H. Weber, *Gaussian approximations for transition paths in Brownian dynamics*, SIAM J. Math. Anal., **49(4):3005–3047, 2017. 1**
- [16] J.C. Mattingly, A.M. Stuart, and D.J. Higham, *Ergodicity for SDEs and approximations: locally Lipschitz vector fields and degenerate noise*, Stochast. Process. Appl., **101(2):185–232, 2002. 2.1**
- [17] N. Metropolis and S. Ulam, *The Monte Carlo method*, J. Am. Stat. Assoc., **44(247):335–341, 1949. 4.2**

- [18] G.N. Milshtein, *A method of second-order accuracy integration of stochastic differential equations*, Theory Probab. Appl., **23(2):396–401**, 1979. [1](#), [4.4](#)
- [19] G.N. Milstein, *Weak approximation of solutions of systems of stochastic differential equations*, Theory Probab. Appl., **30(4):750–766**, 1986. [2.1](#), [2.1](#), [4.4](#)
- [20] G.N. Milstein and M.V. Tretyakov, *Stochastic Numerics for Mathematical Physics*, Springer Science & Business Media, **2013**. [1](#)
- [21] M. Ninomiya and S. Ninomiya, *A new higher-order weak approximation scheme for stochastic differential equations and the Runge–Kutta method*, Financ. Stoch., **13(3):415–443**, 2009. [1](#)
- [22] S. Ninomiya and N. Victoir, *Weak approximation of stochastic differential equations and application to derivative pricing*, Appl. Math. Financ., **15(2):107–121**, 2008. [1](#)
- [23] B. Øksendal, *Stochastic Differential Equations*, Springer, **2003**. [1](#), [2.1](#), [2.1](#), [5.1](#)
- [24] G.A. Pavliotis, *Stochastic Processes and Applications: Diffusion Processes, the Fokker-Planck and Langevin Equations*, Springer, **60**, 2014. [1](#)
- [25] A. Rössler, *Runge–Kutta methods for Itô stochastic differential equations with scalar noise*, BIT Numer. Math., **46(1):97–110**, 2006. [1](#)
- [26] A. Rößler, *Second order Runge–Kutta methods for Itô stochastic differential equations*, SIAM J. Numer. Anal., **47(3):1713–1738**, 2009. [1](#)
- [27] R.Y. Rubinstein and D.P. Kroese, *Simulation and the Monte Carlo Method*, John Wiley & Sons, **10**, 2016. [4.2](#)
- [28] D. Talay, *Efficient numerical schemes for the approximation of expectations of functionals of the solution of a SDE, and applications*, in H. Korezlioglu, G. Mazziotto and J. Szpirglas (eds.), Filtering and Control of Random Processes, Springer, **294–313**, 1984. [1](#), [4.4](#)
- [29] D. Talay and L. Tubaro, *Expansion of the global error for numerical schemes solving stochastic differential equations*, Stoch. Anal. Appl., **8(4):483–509**, 1990. [1](#)
- [30] N.G. Van Kampen, *Stochastic Processes in Physics and Chemistry*, Elsevier, **1**, 1992. [1](#)