第一页为封面页

参赛学生姓名：Thomas Sun

中学：武汉爱莎文华高级中学

省份：湖北

国家/地区：中华人民共和国

指导老师姓名：刘显明

指导老师单位：华中科技大学

论文题目：Density evolution in stochastic dynamical systems with memory: A universal algorithm

Density evolution in stochastic dynamical systems with memory: A universal algorithm

Thomas Sun

ISA Wuhan Wenhua School, Wuhan 430119, Hubei, China

October 31, 2024

Abstract: Stochastic dynamical systems with memory are usually modeled using stochastic functional differential equations. Quantifying the probability density evolution in these systems remains an open problem with strong practical applications. However, due to a lack of efficient methods for computing the probability density of stochastic functional differential equations in their general form, the application of these systems are severely restricted. We address this challenge by presenting a universal approach for computing the evolution of probability density in a broad class of stochastic dynamical systems with memory. To verify and illustrate the proposed approach, it is applied to compute both the transient and long time evolution of probability density for some typical climate models.

Keywords: probability density, numerical method, stochastic dynamical system, memory.

Contents

1 Introduction

Conventional stochastic differential equations (SDEs), which have the Markovian property, are used extensively to model stochastic dynamical systems without memory (i.e., the future states of the systems depend only on the current states). One of the central tasks of stochastic dynamics is to quantify the probabilistic behavior of the systems, which is often described by the evolution of probability density. There are well established tools such as Fokker-Planck equations [1, 2], to predict the time evolution of probability density for SDEs without memory.

In practice, many engineering and scientific systems are subject to both noise and memory. For example, it is suggested in [3], from the viewpoint of mathematical modeling, that appropriate models of climate systems should reflect the roles of both noise and memory, and thus these models are better modeled by stochastic differential equations with memory. However, the reported applications of stochastic differential equations with memory are much less compared to its counterpart without memory. The reason for this lies partly in the fact that approaches are rarely available to predict the probabilistic behavior for SDEs with memory. In fact, It is a well known open problem to quantify probability density in stochastic dynamical systems with memory. Some challenges regarding this open problem are mentioned in [4, 5].

Recently, a governing equation and its numerical method was presented in [6, 7] for the probability density of stochastic dynamical systems subject to discrete time delays of special form. However, the methods given in [6, 7] are only applicable to discrete time delays of a special form, and suffer from high computational loads due to the fact that the dimensions of the governing equation increases with time.

The main objective of this paper is to develop an universal algorithm to compute the probability density for stochastic systems modeled by a general class of stochastic functional equations.

Many stochastic dynamical systems with with discrete time delays can be modeled by the following stochastic delay differential equation (SDDE)

$$\begin{aligned} \mathrm{d}X(t) &= f(X(t), X(t-\tau_1), X(t-\tau_2)), \cdots, X(t-\tau_m)))\mathrm{d}t \\ &\quad + g(X(t), X(t-\tau_1), X(t-\tau_2)), \cdots, X(t-\tau_m)))\mathrm{d}B(t), \quad \text{for} \quad t > 0, \end{aligned} \tag{1}$$

with initial data

$$X(t) = \gamma(t), \quad \text{for} \quad -\max\{\tau_1, \tau_2, \cdots, \tau_m\} \leq t \leq 0, \tag{2}$$

where $\tau_1, \tau_2, \cdots, \tau_m$ are m discrete time delays, $X(t)$ is a $\mathbb{R}^d$-valued stochastic process, $B(t)$ is a $\mathbb{R}^n$-valued Brownian motion defined on some probability space $(\Omega, \mathcal{F}, P)$, $f : \mathbb{R}^{(m+1)\times d} \to \mathbb{R}^d$, $g : \mathbb{R}^{(m+1)\times d} \to \mathcal{M}^{d\times n}$ and $\gamma : [-\tau, 0] \to \mathbb{R}^d$. Here $\mathcal{M}^{d\times n}$ is the set of all d-by-n real matrices.

The SDDE defined by (1) and (2) is an important special case of the following stochastic functional differential equation (SFDE):

$$\mathrm{d}X(t) = f(X_t)\mathrm{d}t + g(X_t)\mathrm{d}B(t), \quad \text{for} \quad t > 0, \tag{3}$$

with initial data

$$X(t) = \gamma(t), \quad \text{for} \quad -\tau \leq t \leq 0. \tag{4}$$

Here, $X(t)$ is a $\mathbb{R}^d$-valued stochastic process, $B(t)$ is a $\mathbb{R}^n$-valued Brownian motion, f and g are functionals with $f : C([-\tau,0];\mathbb{R}^d] \to \mathbb{R}^d$ and $g : C([-\tau,0];\mathbb{R}^d] \to \mathcal{M}^{d\times n}$. Note that X_t, which is a $C([-\tau,0];\mathbb{R}^d)$-valued random variable, can be expressed more accurately as $X_t : [-\tau,0] \to \mathbb{R}^d, \ \ \theta \to X(t+\theta)$. Here, the main use of X_t is to represent the segment of solution path on the interval $[t-\tau,t]$.

Throughout the paper, unless we state otherwise, we assume that: (i) f and g are Lipschitz continuous; (ii) g satisfies the strong elliptic condition, i.e. there exists a constant $\epsilon > 0$ such that $gg^T \geq \epsilon I$. Here I is the identity matrix. Note that under assumptions (i) and (ii), both (1) and (3) have a unique, strong solution, and the probability densities for the solutions exists and is continuous, see [8, 9], among others, for a more detailed discussion. In this paper, we derive an algorithm to compute the probability density associated with (1) and (2), and (3) and (4), respectively, under assumptions (i) and (ii). Note that for our purposes, the above assumptions (i) and (ii) can be replaced by some more relaxed conditions, although it will not be discussed here.

The paper is is organized as follows: In Section 2, the numerical method to compute the probability density for SDDE (1) with initial condition (2) is derived. In Section 3, the method is generalized to compute the probability density for SDFE (3) with initial condition (4). In Section 4, convergence rate of the numerical method with respect to the time step is discussed. Finally, in Section 5, some numerical examples are presented to illustrate and verify the proposed algorithm.

2 The algorithm for SDDEs

In this section, we develop our algorithm to compute the probability density for SDDE (1) with initial conditions (2). For brevity, we only consider the special case of (1) with $d = 1$, $n = 1$, and $k = 2$. For general cases, the derivation is essentially the same, and will not be presented here.

With $d = 1$, $n = 1$, and $k = 2$, equations (1) and (2) become

$$\mathrm{d}X(t) = f(X(t), X(t-\tau_1), X(t-\tau_2))\mathrm{d}t + g(X(t), X(t-\tau_1), X(t-\tau_2))\mathrm{d}B(t) \quad \text{for} \quad t > 0 \tag{5}$$

and

$$X(t) = \gamma(t), \quad \text{for} \quad -\max\{\tau_1,\tau_2\} \leq t \leq 0, \tag{6}$$

respectively.

Consider $N+1$ discrete time points t_j $(j = 0,1,2,\cdots,N)$ on the time interval $[0,T]$ such that $t_j = j\Delta t$ $(j = 0,1,2,\cdots,N)$, where $\Delta t = \dfrac{T}{N}$ is the time step. By the Euler method, (5) and (6) can be discretized as

$$\hat{X}(t_{k+1}) = \hat{X}(t_k) + f(\hat{X}(t_k), \hat{X}(t_{k-m_1}), \hat{X}(t_{k-m_2}))\Delta t + g(\hat{X}(t_k), \hat{X}(t_{k-m_1}), \hat{X}(t_{k-m_2}))\Delta B_k, \tag{7}$$

with initial values

$$\hat{X}(t_{-j}) = \gamma(t_{-j}) \quad \text{for} \quad j = 0, 1, 2, \cdots, m, \tag{8}$$

where $m_1 = \lfloor \frac{\tau_1}{\Delta t} \rfloor$, $m_2 = \lfloor \frac{\tau_2}{\Delta t} \rfloor$, $m = \max\{m_1, m_2\}$, $\Delta B_k = B(t_{k+1}) - B(t_k)$, and $k = 0, 1, 2, \cdots, N$. Here, $\lfloor \cdot \rfloor$ represents the floor function.

Denote $\rho_{\mathcal{A}}(x,t)$ as the probability density of $X(t)$ given in (5) and (6), and $\hat{\rho}_{\hat{\mathcal{A}},\Delta t}(x_k, t_k)$ as the probability density of $\hat{X}(t_k)$ given in (7) and (8). Here the subscript $\mathcal{A}$ and $\hat{\mathcal{A}}$ are used to explicitly indicate the inital conditions (6) and (8) respectively. By the above definition, $\rho_{\mathcal{A}}(x,t)$ and $\hat{\rho}_{\hat{\mathcal{A}},\Delta t}(x_k, t_k)$ can be expressed in form of conditional densities, i.e.,

$$\rho_{\mathcal{A}}(x,t) = p(x, t \big| X(s) = \gamma(s) \quad \text{for} \quad -\tau \leq s \leq 0), \tag{9}$$

and

$$\hat{\rho}_{\hat{\mathcal{A}},\Delta t}(x_k, t_k) = \hat{p}(x_k, t_k \big| x_0, t_0; x_{-1}, t_{-1}; \cdots; x_{-m}, t_{-m}), \tag{10}$$

where $p(x, t \big| X(s) = \gamma(s) \quad \text{for} \quad -\tau \leq s \leq 0)$ represents the probability density of $X(t)$ at $X(t) = x$ under the condition that $X(s) = \gamma(s) \quad \text{for} \quad -\tau \leq s \leq 0$, $p(x_k, t_k \big| x_0, t_0; x_{-1}, t_{-1}; \cdots; x_{-m}, t_{-m})$ represents the probability density of $\hat{X}(t_k)$ at $\hat{X}(t_k) = x_k$ under the condition that $\hat{X}(t_j) = x_{-j}$. Note that throughout the paper, x_{-j} with $j = 0, 1, 2, \cdots, m$ is a constant equal to the initial value $\gamma(t_{-j})$.

Both the weak and strong convergence of $\hat{X}(t)$, given in (7), to the original solution $X(t)$, given in (5), has been studied extensively. See [10, 11, 12], among others, for more details. Under the condtion that the probability density exists and is continuous, either weak or strong convergence as mentioned above guarantees the convergence of the probability density, i.e.,

$$\forall y \in \mathbb{R}, \quad \left| \hat{\rho}_{\hat{\mathcal{A}},\Delta t}(y, t_k) - \rho_{\mathcal{A}}(y, t_k) \right| \to 0 \quad \text{as} \quad \Delta t \to 0. \tag{11}$$

The convergence given in (11) provides the theoretical foundation to approximate $\rho_{\mathcal{A}}(y, t_k)$, the probability density of $X(t)$, by using $\hat{\rho}_{\hat{\mathcal{A}},\Delta t}(y, t_k)$, the probability density of $\hat{X}(t)$, given that the time step Δt is sufficiently small. The main goal of the proposed algorithm is to approximate $\rho_{\mathcal{A}}(y, t_k)$ by designing an efficient way to compute $\hat{p}_{\hat{\mathcal{A}},\Delta t}(x_k, t_k)$.

It follows from (10) that

$$\begin{aligned}
&\hat{\rho}_{\hat{\mathcal{A}},\Delta t}(x_k, t_k) = \hat{p}(y, t_k | x_0, t_0; x_{-1}, t_{-1}; \cdots; x_{-m}, t_{-m}) \\
&= \overbrace{\int_{\mathbb{R}} \int_{\mathbb{R}} \cdots \int_{\mathbb{R}}}^{k-1} \hat{p}(x_k, t_k; x_{k-1}, t_{k-1}; \cdots; x_1, t_1 | x_0, t_0; x_{-1}, t_{-1}; \cdots; x_{-m}, t_{-m}) \prod_{i=1}^{k-1} \mathrm{d}x_i \\
&= \overbrace{\int_{\mathbb{R}} \int_{\mathbb{R}} \cdots \int_{\mathbb{R}}}^{k-1} \hat{p}(x_k, t_k | x_{k-1}, t_{k-1}; \cdots; x_{-m}, t_{-m}) \, \hat{p}(x_{k-1}, t_{k-1} | x_{k-2}, t_{k-2}; \cdots; x_{-m}, t_{-m}) \\
&\qquad \cdots \hat{p}(x_1, t_1 | x_0, t_0; \cdots; x_{-m}, t_{-m}) \prod_{i=1}^{k-1} \mathrm{d}x_i.
\end{aligned} \tag{12}$$

Equation (7) implies that $\hat{X}_{t_{k+1}}$ depends only on $\hat{X}_{t_k}$, $\hat{X}_{t_{k-m_1}}$ and $\hat{X}_{t_{k-m_2}}$, i.e.,

$$\hat{p}(x_k, t_k \big| x_{k-1}, t_{k-1}; x_{k-2}, t_{k-2}; \cdots; x_{-m}, t_{-m}) = \hat{p}(x_k, t_k \big| x_{k-1}, t_{k-1}; x_{k-m_1}, t_{k-m_1}; x_{k-m_2}, t_{k-m_2}). \tag{13}$$

It is straightforward to check, by using (7), that the probability density of $\hat{X}(t_{k+1})$ at $\hat{X}(t_{k+1}) = x_{k+1}$ given $\hat{X}(t_k) = x_k$, $\hat{X}(t_{k-m_1}) = x_{k-m_1}$ and $\hat{X}(t_{k-m_2}) = x_{k-m_2}$, can be expressed as

$$\begin{aligned} &\hat{p}(x_{k+1}, t_{k+1}|x_k, t_k; x_{k-m_1}, t_{k-m_1}; x_{k-m_2}, t_{k-m_2}) \\ =&\frac{1}{\sqrt{2\pi\Delta t} \cdot g(x_k, x_{k-m_1}, x_{k-m_2})} \exp\left(-\frac{(x_{k+1} - x_k - f(x_k, x_{k-m_1}, x_{k-m_2})\Delta t)^2}{2\Delta t \cdot g^2(x_k, x_{k-m_1}, x_{k-m_2})}\right). \end{aligned} \tag{14}$$

Substituting (13) into (12), we get

$$\begin{aligned} &\hat{\rho}_{\hat{\mathcal{A}},\Delta t}(x_k, t_k) \\ =&\overbrace{\int_{\mathbb{R}}\int_{\mathbb{R}}\cdots\int_{\mathbb{R}}}^{k-1} \hat{p}(x_k, t_k|x_{k-1}, t_{k-1}; x_{k-m_1-1}, t_{k-m_1-1}; x_{k-m_2-1}, t_{k-m_2-1})) \\ &\times \hat{p}(x_{k-1}, t_{k-1}|x_{k-2}, t_{k-2}; x_{k-m_1-2}, t_{k-m_1-2}; x_{k-m_2-2}, t_{k-m_2-2}) \\ &\times \cdots \times \hat{p}(x_1, t_1|x_0, t_0; x_{-m_1}, t_{-m_1}; x_{-m_2}, t_{-m_2}) \prod_{i=1}^{k-1} \mathrm{d}x_i \end{aligned} \tag{15}$$

Next we will proceed to compute the right hand side of (15) recursively. To this end, introduce a family of functions I_k indexed by k ($k = 1, 2, \ldots, N$) such that $I_k : \mathbb{R}^{m+1} \to \mathbb{R}$, $(z_1, z_2, \cdots, z_{m+1}) \to I_k(z_1, z_2, \cdots, z_{m+1})$ and

$$\begin{aligned} &I_k(x_k, x_{k-1}, \cdots, x_{k-m}) \\ =& \overbrace{\int_{\mathbb{R}}\int_{\mathbb{R}}\cdots\int_{\mathbb{R}}}^{\max\{k-m-1,0\}} \hat{p}(x_k, t_k|x_{k-1}, t_{k-1}; x_{k-1-m_1}, t_{k-1-m_1}; x_{k-1-m_2}, t_{k-1-m_2}) \\ &\times \hat{p}(x_{k-1}, t_{k-1}|x_{k-1-m_1}, t_{k-1-m_1}; x_{k-2-m_2}, t_{k-2-m_2}) \\ &\times \cdots \times \hat{p}(x_1, t_1|x_0, t_0; x_{-m_1}, t_{-m_1}; x_{-m_2}, t_{-m_2}) \prod_{i=1}^{k-m-1} \mathrm{d}x_i \end{aligned} \tag{16}$$

Note that for $k - m - 1 \leq 0$, the right hand side of (16) just represents the integrand itself (i.e., no integrating is performed).

Examining (16), we find that I_k can be expressed recursively as

$$I_1(x_1, x_0, x_{-1}, \ldots, x_{1-m}) = \hat{p}(x_1, t_1|x_0, t_0; x_{-m_1}, t_{-m_1}; x_{-m_2}, t_{-m_2}), \tag{17}$$

$$\begin{aligned} &I_k(x_k, x_{k-1}, \ldots, x_{k-m}) \\ &= \hat{p}(x_k, t_k|x_{k-1}, t_{k-1}; x_{k-m_1-1}, t_{k-m_1-1}; x_{k-m_2-1}, t_{k-m_2-1}) \\ &\times I_{k-1}(x_{k-1}, x_{k-2}, \ldots, x_{k-m-1}) \\ &\quad \text{for} \quad k = 2, 3, \ldots, m+1, \end{aligned} \tag{18}$$

and

$$
\begin{aligned}
&I_k(x_k, x_{k-1}, \cdots, x_{k-m}) \\
&= \int_{\mathbb{R}} \hat{p}(x_k, t_k | x_{k-1}, t_{k-1}; x_{k-m_1-1}, t_{k-m_1-1}; x_{k-m_2-1}, t_{k-m_2-1}) \\
&\quad \times I_{k-1}(x_{k-1}, x_{k-2}, \ldots, x_{(k-m-1)}) \, \mathrm{d}x_{k-m-1} \\
&\quad \text{for} \quad k \geq m+2.
\end{aligned}
\tag{19}
$$

Given I_k defined in (17), (18) and (19), (15) can be expressed in term of I_k as

$$
\hat{\rho}_{\hat{\mathcal{A}},\Delta t}(x_k, t_k) = \begin{cases}
I_1(x_1, x_0, x_{-1}, \ldots, x_{1-m}) \quad \text{for} \quad k = 1, \\
\overbrace{\int_{\mathbb{R}} \int_{\mathbb{R}} \cdots \int_{\mathbb{R}}}^{k-1} I_k(x_k, \ldots, x_1, x_0, \cdots, x_{-(m-k)}) \prod_{i=1}^{k-1} \mathrm{d}x_i \quad \text{for} \quad 2 \leq k \leq m+1, \\
\overbrace{\int_{\mathbb{R}} \int_{\mathbb{R}} \cdots \int_{\mathbb{R}}}^{m} I_k(x_k, x_{k-1}, \cdots, x_{k-m}) \prod_{i=k-m}^{k-1} \mathrm{d}x_i, \quad \text{for} \quad k \geq m+2.
\end{cases}
\tag{20}
$$

To evaluate $\hat{p}_{\hat{\mathcal{A}},\Delta t}$ numerically, we approximate integrals over $\mathbb{R}$ by integrals over the interval $[-D, D]$, where D is a sufficiently large constant. Then (19) and (20) become

$$
\begin{aligned}
&I_k(x_k, x_{k-1}, \cdots, x_{k-m}) \\
&= \int_{-D}^{D} \hat{p}(x_k, t_k | x_{k-1}, t_{k-1}; x_{k-1-m_1}, t_{k-1-m_1}; x_{k-1-m_2}, t_{k-1-m_2}) \\
&\quad \times I_{k-1}(x_{k-1}, x_{k-2}, \ldots, x_{k-m-1}) \, \mathrm{d}x_{k-m-1} \\
&\quad \text{for} \quad k \geq m+2.
\end{aligned}
\tag{21}
$$

and

$$
\hat{\rho}_{\hat{\mathcal{A}},\Delta t}(x_k, t_k) = \begin{cases}
I_1(x_1, x_0, x_{-1}, \ldots, x_{1-m}) \quad \text{for} \quad k = 1, \\
\overbrace{\int_{-D}^{D} \int_{-D}^{D} \cdots \int_{-D}^{D}}^{k-1} I_k(x_k, \ldots, x_1, x_0, \cdots, x_{-(m-k)}) \prod_{i=1}^{k-1} \mathrm{d}x_i \quad \text{for} \quad 2 \leq k \leq m+1, \\
\overbrace{\int_{-D}^{D} \int_{-D}^{D} \cdots \int_{-D}^{D}}^{m} I_k(x_k, x_{k-1}, \cdots, x_{k-m}) \prod_{i=k-m}^{k-1} \mathrm{d}x_i, \quad \text{for} \quad k \geq m+2,
\end{cases}
\tag{22}
$$

respectively.

Next, discretize the interval $[-D, D]$ with $M+1$ equally distanced grid points $z_0, z_1, \cdots, z_M$ such that $z_i = -D + ih$, here $i = 0, 1, 2, \cdots, M$ and $h = \frac{2D}{M}$. Then, by using the Euler integration scheme, (21) and (22) become

$$\begin{aligned}
&I_k(x_k, z_{i_{k-1}}, \ldots, z_{i_{k-m}}) \\
&= h \times \sum_{i_{k-m-1}=0}^{M} \hat{p}(x_k, t_k | z_{i_{k-1}}, t_{k-1}; z_{i_{k-m_1-1}}, t_{k-m_1-1}; z_{i_{k-m_2-1}}, t_{k-m_2-1}) \\
&\quad \times I_{k-1}(x_{k-1}, z_{i_{k-2}}, \ldots, z_{i_{k-m-1}}) \\
&\quad \text{for} \quad k \geq m+2.
\end{aligned} \tag{23}$$

and

$$\hat{\rho}_{\hat{\mathcal{A}},\Delta t}(x_k, t_k) = \begin{cases} I_1(x_1, x_0, x_{-1}, \ldots, x_{1-m}) \quad \text{for} \quad k = 1, \\ h^{k-1} \times \overbrace{\sum_{i_{k-1}=0}^{M} \sum_{i_{k-2}=0}^{M} \cdots \sum_{i_1=0}^{M}}^{k-1} I_k(x_k, \ldots, z_{i_1}, x_0 \ldots, x_{-(m-k)}) \quad \text{for} \quad 2 \leq k \leq m+1, \\ h^m \times \overbrace{\sum_{i_{k-1}=0}^{M} \sum_{i_{k-2}=0}^{M} \cdots \sum_{i_{k-m}=0}^{M}}^{m} I_k(x_k, z_{i_{k-1}}, \ldots, z_{i_{k-m}}) \quad \text{for} \quad k \geq m+2, \end{cases} \tag{24}$$

respectively.

Substituting (14) into (17), (18), and (23), respectively, we get

$$\begin{aligned}
&I_1(x_1, x_0, x_{-1}, \ldots, x_{1-m}) \\
&= \left(\frac{1}{\sqrt{2\pi\Delta t} \cdot g(x_0, x_{-m_1}, x_{-m_2})} \right) \exp \frac{(x_1 - x_0 - f(x_0, x_{-m_1}, x_{-m_2})\Delta t)^2}{2\Delta t \cdot g^2(x_0, x_{-m_1}, x_{-m_2})} \\
&\quad \text{for} \quad k = 1,
\end{aligned} \tag{25}$$

$$\begin{aligned}
&I_k(x_k, z_{i_{k-1}}, \ldots, z_{i_1}, x_0, \ldots, x_{-(m-k)}) \\
&= \left(\frac{1}{\sqrt{2\pi\Delta t} \cdot g(z_{i_{k-1}}, z^*_{k-m_1-1}, z^*_{k-m_2-1})} \right) \exp \frac{(x_k - z_{i_{k-1}} - f(z_{i_{k-1}}, z^*_{k-m_1-1}, z^*_{k-m_2-1})\Delta t)^2}{2\Delta t \cdot g^2(z_{i_{k-1}}, z^*_{k-m_1-1}, z^*_{k-m_2-1})} \\
&\quad \times I_{k-1}(x_{k-1}, z_{i_{k-2}}, \ldots, z_{i_1}, x_0, \ldots, x_{-(m-k-1)}) \\
&\quad \text{for} \quad 2 \leq k \leq m+1,
\end{aligned} \tag{26}$$

$$\begin{aligned}
&I_k(x_k, z_{i_{k-1}}, \ldots, z_{i_{k-m}}) \\
&= \left(\frac{2D}{M} \right) \times \left(\frac{1}{\sqrt{2\pi\Delta t} \cdot g(z_{i_{k-1}}, z_{i_{k-m_1-1}}, z_{i_{k-m_2-1}})} \right) \\
&\quad \times \sum_{i_{k-m-1}=0}^{M} \left\{ \exp \left(\frac{(x_k - z_{i_{k-1}} - f(z_{i_{k-1}}, z_{i_{k-m_1-1}}, z_{i_{k-m_2-1}})\Delta t)^2}{2\Delta t \cdot g^2(z_{i_{k-1}}, z_{i_{k-m_1-1}}, z_{i_{k-m_2-1}})} \right) I_{k-1}(x_{k-1}, z_{i_{k-2}}, \ldots, z_{i_{k-m-1}}) \right\} \\
&\quad \text{for} \quad k \geq m+2,
\end{aligned} \tag{27}$$

where z_s^* in (26) is defined as being equal to z_{i_s} for $s \geq 1$, and being equal to x_s for $s \leq 0$.

Combining (25), (26), (27) and (24), we get the proposed algorithm to compute $\hat{\rho}_{\hat{\mathbb{A}},\Delta t}(x_k, t_k)$. The steps to implement the algorithm are as below:

```
Algorithm: Compute the probability density associated with SDDE (1) with initial condition (2).
  Input: $T$, $D$, $N$, $M$, $f$ and $g$. Here, $N+1$ is the number of time nodes, $M+1$ is the number of
         space nodes, $f$ and $g$ are functions used in the coefficients of (1).
  Output: $\hat{\rho}_{\hat{\mathcal{A}},\Delta t}(x_k, t_k)$, which is the numerical solution to the probability density of $X(t)$
  Initialize time step $\Delta t = \frac{T}{N}$, and space step $h = \frac{2D}{M}$
  for $k = 0$ to $N$ do
      set $t_k = k\Delta t$
  for $k = 0$ to $M$ do
      compute $I_k$ by using (25), (26) and (27)
  Get $\hat{\rho}_{\hat{\mathcal{A}},\Delta t}(x_k, t_k)$ by using (24)
  Output $\hat{\rho}_{\hat{\mathcal{A}},\Delta t}(x_k, t_k)$
```

3 The algorithm for SFDEs

In this section we shall derive the algorithm to compute the probability density for SFDE (3) with $X(t)$ being a scalar process. For more general cases of (3) with $X(t)$ being a $\mathbb{R}^d$-valued process, the derivation is essentially the same and will be omitted here for brevity.

Equation (3), with $X(t)$ being a scalar process, becomes

$$\mathrm{d}X(t) = f(X_t)\mathrm{d}t + g(X_t)\mathrm{d}B(t), \quad \text{for} \quad t > 0, \tag{28}$$

with the initial conditions

$$X(t) = \gamma(t), \quad \text{for} \quad -\tau \leq t \leq 0, \tag{29}$$

where $B(t)$ is a scalar Brownian motion, $X(t)$ is a scalar process, X_t is a $C([-\tau,0],\mathbb{R})$-valued random varible, f and g are functionals with $f : C([-\tau,0];\mathbb{R}) \to \mathbb{R}$ and $g : C([-\tau,0];\mathbb{R}) \to \mathbb{R}$. Recall that X_t can be expressed more clearly as $X_t : [-\tau, 0] \to \mathbb{R}^d, \ \ \theta \to X(t+\theta)$.

Consider $N+1$ discrete time points t_j $(j = 0, 1, 2, \cdots, N)$ on the time interval $[0, T]$ such that $t_j = j\Delta t$ $(j = 0, 1, 2, \cdots, N)$, where $\Delta t = \dfrac{T}{N}$ is the time step. By the Euler method, (28) and (29) can be discretized as

$$\hat{X}(t_{k+1}) = \hat{X}(t_k) + f(\hat{X}_{t_k})\Delta t + g(\hat{X}_{t_k})\Delta B_k, \quad \text{for} \quad k > 0, \tag{30}$$

with initial conditions

$$\hat{X}(t_k) = \gamma(t_k), \quad \text{for} \quad -m \leq k \leq 0, \tag{31}$$

where, $m = \lfloor \frac{\tau}{\Delta t} \rfloor$, $\Delta B_k = B(t_{k+1}) - B(t_k)$, $k = 0, 1, 2, \cdots, N$. As used here, $\hat{X}_{t_k}$ is a $C([-\tau, 0]; \mathbb{R})$-valued random variable defined as the linear interpolation of $\hat{X}(t_{k-m}), \hat{X}(t_{k-m+1}), \cdots, \hat{X}(t_k)$, i.e.,

$$\begin{gathered} \hat{X}_{t_k}(\theta) = \frac{t_{i+1} - \theta}{\Delta t} \hat{X}(t_{k+i+1}) + \frac{\theta - t_i}{\Delta t} \hat{X}(t_{k+i}), \\ \text{for} \quad i\Delta t \leq \theta \leq (i+1)\Delta t, \ i = -m, -(m-1), \cdots, -1. \end{gathered} \tag{32}$$

It follows from (32) that the coefficients $f(\hat{X}_{t_k})$ and $f(\hat{X}_{t_k})$ in (30) can be regarded as functions of $\hat{X}(t_{k-m})$, $\hat{X}(t_{k-m+1})$, $\cdots$, $\hat{X}(t_k)$. Introduce functions $\tilde{f} : \mathbb{R}^{m+1} \to \mathbb{R}$ and $\tilde{g} : \mathbb{R}^{m+1} \to \mathbb{R}$ such that

$$f(\hat{X}_{t_k}) = \tilde{f}(\hat{X}(t_{k-m}), \hat{X}(t_{k-m+1}), \cdots, \hat{X}(t_k)) \tag{33}$$

and

$$g(\hat{X}_{t_k}) = \tilde{g}(\hat{X}(t_{k-m}), \hat{X}(t_{k-m+1}), \cdots, \hat{X}(t_k)), \tag{34}$$

then (30) can be rewritten as

$$\begin{aligned} \hat{X}(t_{k+1}) =& \hat{X}(t_k) + \tilde{f}(\hat{X}(t_{k-m}), \hat{X}(t_{k-m+1}), \cdots, \hat{X}(t_k))\Delta t \\ &+ \tilde{g}(\hat{X}(t_{k-m}), \hat{X}(t_{k-m+1}), \cdots, \hat{X}(t_k))\Delta B_k, \quad \text{for} \quad k > 0, \end{aligned} \tag{35}$$

with initial conditions

$$\hat{X}(t_k) = \gamma(t_{-k}), \quad \text{for} \quad -m \leq k \leq 0. \tag{36}$$

Denote $q_{\mathcal{A}}(x, t)$ as the probability density of $X(t)$ defined by (28) and (29), and $\hat{q}_{\hat{\mathcal{A}},\Delta t}(x_k, t_k)$ as the probability density of $\hat{X}(t_k)$ defined by (35) and (36). Here the subscripts $\mathcal{A}$ and $\hat{\mathcal{A}}$ are used to explicitly indicate the initial conditions (29) and (36), respectively. Note that $\rho_{\mathcal{A}}(x, t)$ and $\hat{\rho}_{\hat{\mathcal{A}},\Delta t}(x_k, t_k)$ can be expressed as

$$q_{\mathcal{A}}(x, t) = q(x, t \big| X(s) = \gamma(s) \quad \text{for} \quad -\tau \leq s \leq 0), \tag{37}$$

and

$$\hat{q}_{\hat{\mathcal{A}},\Delta t}(x, t) = \hat{q}(x_k, t_k \big| x_0, t_0; x_{-1}, t_{-1}; \cdots; x_{-m}, t_{-m}), \tag{38}$$

where $q(x, t \big| X(s) = \gamma(s) \quad \text{for} \quad -\tau \leq s \leq 0)$ represents the probability density of $X(t)$ at $X(t) = x$ under the condition that $X(s) = \gamma(s) \quad \text{for} \quad -\tau \leq s \leq 0$, and $\hat{q}(x_k, t_k \big| x_0, t_0; x_{-1}, t_{-1}; \cdots; x_{-m}, t_{-m})$ represents the probability density of $\hat{X}(t_k)$ at $\hat{X}(t_k) = x_k$ under the condition that $\hat{X}(t_j) = x_{-j}$. Similar to Section 3, x_{-j}, with $j = 0, 1, 2, \cdots, m$, is a constant equal to the initial value $\gamma(t_{-j})$.

The strong convergence of $\hat{X}(t_k)$, as given by (35) and (36), to $X(t_k)$, as given by (28) and (29), is proven in [12]. Assuming the probability density of $X(t_k)$ in (28) exists and is continuous, the strong convergence mentioned above guarantees the convergence of probability density, i.e.,

$$\forall y \in \mathbb{R}, \quad \big|\hat{q}_{\hat{\mathcal{A}},\Delta t}(y, t_k) - q_{\mathcal{A}}(y, t_k)\big| \to 0 \quad \text{as} \quad \Delta t \to 0. \tag{39}$$

Hence we see that $\hat{q}_{\hat{\mathcal{A}},\Delta t}(y,t_k)$ can be used to approximate $q_{\mathcal{A}}(y,t_k)$ when Δt is sufficiently small, and it only remains to devise a method to compute $\hat{q}_{\hat{\mathcal{A}},\Delta t}(x_k,t_k)$ efficiently. Notice that equations (35) to (39) in this section correspond to equations (7) to (11) in Section 3. Following the same procedure as in Section 3, where we derived the algorithm given by (25), (26), (27) and (24) starting from equations (7), (8) and (11), we can start from equations (35), (36), (39) to derive the following algorithm for computing $\hat{q}_{\hat{\mathcal{A}},\Delta t}(x_k,t_k)$, the probability density of SDE (35):

$$\begin{aligned}
&I_1(z_{i_1}, x_0, x_{-1}, \ldots, x_{1-m})\\
&=\left(\frac{1}{\sqrt{2\pi\Delta t}\cdot\tilde{g}(x_0,x_{-1},\ldots,x_{-m})}\right)\exp\frac{(z_{i_1}-x_0-\tilde{f}(x_0,x_{-1},\ldots,x_{-m})\Delta t)^2}{2\Delta t\cdot\tilde{g}^2(z_{i_1},x_0,x_{-1},\ldots,x_{-m})}\\
&\text{for}\quad k=1,
\end{aligned}\tag{40}$$

$$\begin{aligned}
&I_k(z_{i_k}, z_{i_{k-1}}, \ldots, z_{i_1}, x_0, \ldots, x_{-(m-k)})\\
&=\left(\frac{1}{\sqrt{2\pi\Delta t}\cdot\tilde{g}(z_{i_{k-1}},\ldots,z_{i_1},x_0,\ldots,x_{-(m-k)})}\right)\exp\frac{(z_{i_k}-z_{i_{k-1}}-\tilde{f}(z_{i_{k-1}},\ldots,z_{i_1},x_0,\ldots,x_{-(m-k)})\Delta t)^2}{2\Delta t\cdot\tilde{g}^2(z_{i_{k-1}},\ldots,z_{i_1},x_0,\ldots,x_{-(m-k)})}\\
&\quad\times I_{k-1}(z_{i_{k-1}}, z_{i_{k-2}}, \ldots, z_{i_1}, x_0, \ldots, x_{-(m-1-k)})\\
&\text{for}\quad 2\le k\le m+1,
\end{aligned}\tag{41}$$

$$\begin{aligned}
&I_k(z_{i_k}, z_{i_{k-1}}, \ldots, z_{i_{k-m}})\\
&=\left(\frac{2D}{M}\right)\times\left(\frac{1}{\sqrt{2\pi\Delta t}\cdot\tilde{g}(z_{i_{k-1}},z_{i_{k-2}},\cdots,z_{i_{k-m-1}})}\right)\\
&\quad\times\sum_{i_{k-1}=0}^{M}\left\{\exp\left(\frac{(z_{i_k}-z_{i_{k-1}}-\tilde{f}(z_{i_{k-1}},z_{i_{k-2}},\cdots,z_{i_{k-m-1}})\Delta t)^2}{2\Delta t\cdot\tilde{g}^2(z_{i_{k-1}},z_{i_{k-2}},\cdots,z_{i_{k-m-1}})}\right)I_{k-1}(z_{i_{k-1}},z_{i_{k-2}},\cdots,z_{i_{k-1-m}})\right\}\\
&\text{for}\quad k\ge m+2.
\end{aligned}\tag{42}$$

$$\hat{q}_{\hat{\mathcal{A}},\Delta t}(x_k,t_k)=\begin{cases}
I_1(z_{i_1},x_0,x_{-1},\ldots,x_{1-m}) \quad \text{for} \quad k=1,\\
h^{k-1}\times\overbrace{\sum_{i_{k-1}=0}^{M}\sum_{i_{k-2}=0}^{M}\cdots\sum_{i_1=0}^{M}}^{k-1}I_k(z_{i_k},\ldots,z_{i_1},x_0\ldots,x_{-(m-k)}) \quad \text{for} \quad 2\le k\le m+1,\\
h^{m}\times\overbrace{\sum_{i_{k-1}=0}^{M}\sum_{i_{k-2}=0}^{M}\cdots\sum_{i_{k-m}=0}^{M}}^{m}I_k(z_{i_k},z_{i_{k-1}},\ldots,z_{i_{k-m}}) \quad \text{for} \quad k\ge m+2.
\end{cases}\tag{43}$$

Comparing (25), (26), (27) and (24), which express the algorithm for SDDE (1) given in Section 2, with (40), (41), (42) and (43), which express the algorithm for SFDE (3), we observe that they are structurally identical. The sole distinction lies in the notation: f and g are used for the former while $\tilde{f}$ and $\tilde{g}$ are used for the latter. Consequently, the steps to implement the algorithm for SFDEs are the same as the steps to implement it for SDDEs, which were outlined at the end of Section 2.

4 Remarks on the convergence rate

The convergence for the algorithms proposed in Sections 3 and 4 is shown by (11) and (39), respectively. However, the convergence rate of the algorithm is not clear so far.

Convergence rate of the weak convergence and strong convergence for the Euler method has been studied extensively. In contrast, research results on the convergence rate for probability density are much rarer. It is shown in [13] that, under some Hormander type condition, the convergence rate of probability density for stochastic differential equations driven by Brownian motion and without memory is of the first order, but the convergence rate of probability density for SDEs with memory has not been proven so far.

Our numerical simulations suggest that the convergence rate of probability density for SFDEs with memory is of the first order, provided that f and g are smooth in addition to assumptions (i) and (ii) as mentioned in the introduction, i.e.,

$$\forall y \in \mathbb{R}, \quad \left|\hat{\rho}_{\hat{\mathcal{A}},\Delta t}(y,t_k) - \rho_{\mathcal{A}}(y,t_k)\right| \leq C\Delta t \quad \text{as} \quad \Delta t \to 0, \tag{44}$$

$$\forall y \in \mathbb{R}, \quad \left|\hat{q}_{\hat{\mathcal{A}},\Delta t}(y,t_k) - q_{\mathcal{A}}(y,t_k)\right| \leq C\Delta t \quad \text{as} \quad \Delta t \to 0. \tag{45}$$

However, a rigorous proof of (44) and (45) is a nontrivial task and will be left for our future work.

5 Numerical Examples

Three examples are provided in this section to illustrate and verify the proposed algorithm. Example 1 is a toy example, where the analytical solution is available to test the proposed algorithm. In Example 2, the probability density of an SFDE with continuous time delay is considered. In Example 3, the proposed algorithm is applied to an SDDE—the stochastic EINO model, revealing its density evolution into the steady state.

Example 1 Consider the following SDDE,

$$\begin{cases} \mathrm{d}X(t) = X(t-\tau)\mathrm{d}t + \mathrm{d}B(t), \\ X(t) = 0, \quad -\tau \leq t \leq 0, \end{cases} \tag{46}$$

where $X(t)$ is a $\mathbb{R}$-valued process, and $B(t)$ is the standard scalar Brownian motion. Note that (46) is a special form of SDDE (1) with $\tau_1 = \tau_2 = \cdots = \tau_m$ and a diffusion coefficient of 1.

The analytical solution is available for SDDE (46) for $t \in (0, 2\tau]$. In fact, it is straightforward to check that the solution to (46) is

$$X(t) = \begin{cases} B(t), & \text{for} \quad t \in (0, \tau], \\ (t-\tau)B(t-\tau) + B(t) - \int_0^{t-\tau} s\mathrm{d}B(s), & \text{for} \quad t \in (\tau, 2\tau], \end{cases} \tag{47}$$

and the probability density for the solution $X(t)$ is

$$\mathcal{P}_{\mathcal{A}}(x,t) = \begin{cases} \dfrac{1}{\sqrt{2\pi t}} \exp\left(-\dfrac{x^2}{2t}\right). & \text{for} \quad t \in (0, \tau], \\ \dfrac{\sqrt{3}}{\sqrt{2\pi\left((t-\tau)^3 + 3\tau - 1\right)}} \exp\left(-\dfrac{3x^2}{2((t-\tau)^3 + 3\tau - 1)}\right), & \text{for} \quad t \in (\tau, 2\tau]. \end{cases} \tag{48}$$

Table 1: Convergence order of the proposed algorithm.

Δt	$\text{Err}(\Delta t)$	$r = \log_2\left(\frac{\text{Err}(2\Delta t)}{\text{Err}(\Delta t)}\right)$
1/5	0.0109	*
1/10	0.0054	1.01
1/20	0.0027	1.00

Now we solve equation (46) using the numerical method presented in Section 2. The analytical solution and numerical solution to the probability density at $t = 0.4$ are compared in Figure 1. In the numerical computation, $\tau = 0.2$, the time step is $\frac{1}{20}$, the space step is $\frac{1}{10}$, and the space domain is $[-3.5, 3.5]$. It can be seen from Figure 1 that the numerical solution agrees very well with the analytical solution.

The errors and convergence orders for the numerical solutions with different time steps are listed in Table 1, where the errors (the 2nd column) are defined as the square root of the mean square error at all space grid points, and convergence order (the 3rd column) for time step Δt is defined by $r = \log_2\left(\frac{\text{Err}(\Delta t)}{\text{Err}(2\Delta t)}\right)$. The space step $\Delta x = \frac{1}{10}$ is fixed while the time step Δt is chosen as $\frac{1}{5}$, $\frac{1}{10}$ and $\frac{1}{20}$, respectively. Table 1 shows that the proposed algorithm has a first order convergence with respect to the time step.

Example 2 Consider the following SFDE,

$$\begin{cases} \mathrm{d}X(t) = -\left(\int_0^\tau X(t-s)\,\mathrm{d}s\right)\mathrm{d}t + \left(\int_0^\tau (1 + 0.5\cos(X(t-s)))\,\mathrm{d}s\right)\mathrm{d}B(t), \\ X(t) = -0.1, \quad -\tau \le t \le 0. \end{cases} \tag{49}$$

Take $\tau = 0.2$. The probability density for (49) is computed using the algorithm proposed in Section 3. In the numerical computation, the time step is $1/20$, the space step is $1/10$, and the space domain is $[-1, 1]$. The probability density at $t = 0.5$ obtained by the proposed algorithm is compared with that obtained by Monte Carlo simulations, as shown in Figure 2. It can be seen from the Figure that the results of the proposed method are consistent with that of Monte Carlo simulations.

Example 3 Consider the following simplified model for EI-Nino Southern Oscillation (ENSO), [14, 15],

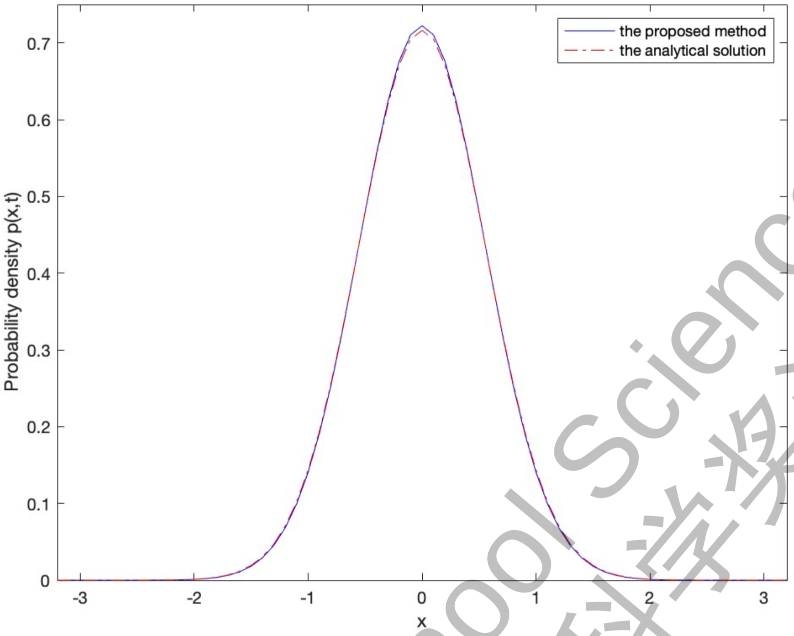

Figure 1: Probability density for SDDE (**??**) at $t = 0.5$.

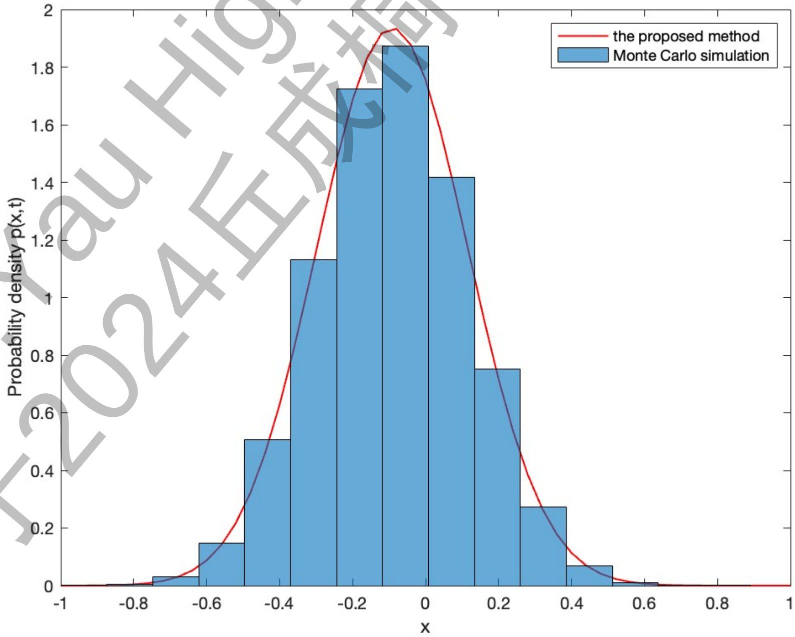

Figure 2: Probability density for SDDE (49) at $t = 0.5$.

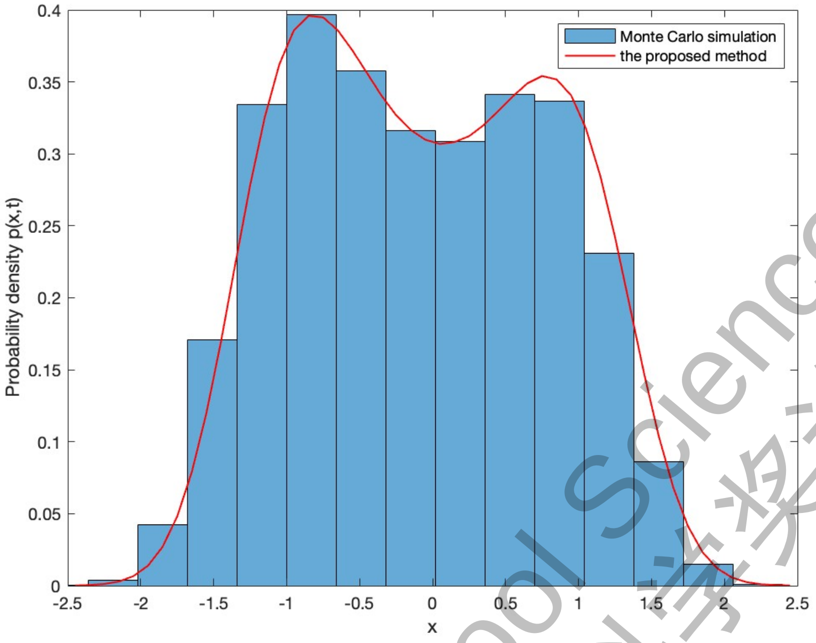

Figure 3: Probability density of SDDE (50) at $t = 2$

with additive noise,

$$\begin{cases} \mathrm{d}X(t) = \big(X(t) - X^3(t) - \alpha_1 X(t-\tau_1) - \alpha_2 X(t-\tau_2)\big)\mathrm{d}t + \mathrm{d}B(t), \quad \text{for} \quad t > 0, \\ X(t) = -0.1, \quad -\tau \leq t \leq 0. \end{cases} \tag{50}$$

where τ_1 and τ_2 are the nondimensional delays. Here, we take $\tau_1 = 0.2$, $\tau_2 = 0.4$, and $\alpha_1 = \alpha_2 = 0.1$. The deterministic model corresponding to (50) has two stable equilibriums at $x = (1 - \alpha_1 - \alpha_2)^{1/2}$ and $x = -(1 - \alpha_1 - \alpha_2)^{1/2}$, and one unstable equilibrium at $x = 0$.

The probability density for (50) is solved for by the proposed algorithm in Section 2. In the computation, both the time step and the space step is 1/10. Time evolution of the probability density is shown in Figure 3. It can be seen from the figure that the probability density gradually becomes steady after some period of drastic transition. Note that Figure 3 does not show the values of $p(x,t)$ for $t \leq 0.5$ since it is already known that the density function is approximately the delta function when t is small, and we are mainly interested in the evolution of the probability density for values of t not close to 0. Figure 4 shows that the probability density at $t = 2$ by the proposed algorithm is consistent with that by Monte Carlo simulations.

6 Conclusion

Quantifying the density evolution in a stochastic system is a central task for stochastic dynamics. Stochastic functional differential equations are appropriate models for stochastic dynamical systems with memory.

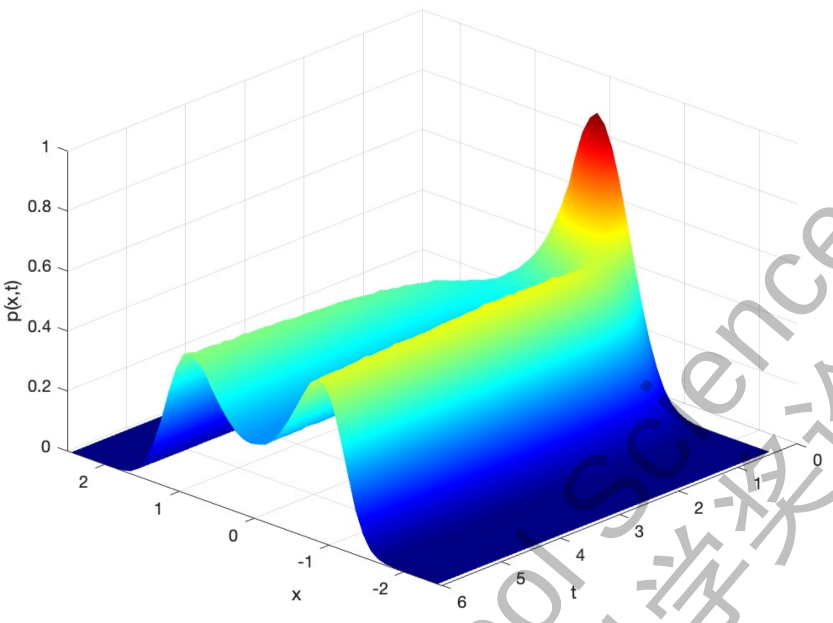

Figure 4: Evolution of probability density for SDDE (50)

However, the application of SFDEs is severely restricted due to the fact that there is no efficient algorithm to compute its density evolution. A universal algorithm is presented in this paper to compute the probability density for SDDE (1) and SFDE (3). This algorithm provides an efficient method for predicting transitional and long-term probabilistic behavior in a general class of stochastic systems with memory, thus demonstrating significant potential for widespread applications. We intend to explore these applications further in our future work.

References

[1] Y. K. Lin and G. Q. Cai. *Probabilistic Structural Dynamics: Advanced Theory and Applications*. Springer, 2005.

[2] C. Gardiner. *Handbook of Stochastic Methods*. Springer; 3rd edition, 2004.

[3] G. Gottwald, Crommelin D., and C. Franzke. *Stochastic Climate Theory. In Christian L E Franzke, Terence J OKane (Eds.), Nonlinear and Stochastic Climate Dynamics, (pp. 209-240)*. Cambridge University Press, 2017.

[4] Jérôme Losson, Michael Mackey, Richard Taylor, and Marta Tyran-Kamińska. *Density Evolution Under Delayed Dynamics: An Open Problem*. Springer, 2020.

[5] Michael C. Mackey and Marta Tyran-Kamińska. How can we describe density evolution under delayed dynamics? *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 31(4):043114, 04 2021.

[6] Y. Y. Zheng and X. Sun. Govering equations for probability densities of stochastic differential equations with discrete time delays. *Discrete and continuous dynamicla systems-series B*, 22(9):3615–3628, 2017.

[7] Xu Sun and Fang Yang. Time evolution of probability density in stochastic dynamical systems with time delays: The governing equation and its numerical solution. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 32(12):123124, 2022.

[8] Alberto Ohashi, Francesco Russo, and Evelina Shamarova. Smoothness of densities for path-dependent sdes under hörmander's condition. *Journal of Functional Analysis*, 281(11):109225, 2021.

[9] R. Chhaibi and I. Ekren. A Hörmander condition for delayed stochastic differential equations. *Annales Henri Lebesgue*, 3:1023–1048, 2020.

[10] Evelyn Buckwar, Rachel Kuske, Salah-Eldin A. Mohammed, and Tony Shardlow. Weak convergence of the euler scheme for stochastic differential delay equations. *Lms Journal of Computation and Mathematics*, 11:60–99, 2008.

[11] Christopher T. H. Baker and Evelyn Buckwar. Numerical analysis of explicit one-step methods for stochastic delay differential equations. *LMS Journal of Computation and Mathematics*, 3:315–335, 2000.

[12] Xuerong Mao. Numerical solutions of stochastic functional differential equations. *LMS Journal of Computation and Mathematics*, 6:141–161, 2003.

[13] V. Bally and D. Talay. The law of the euler scheme for stochastic differential equations II. convergence rate of the density. *Monte Carlo Methods and Applications*, 2(2):93–128, 1996.

[14] M. Ghil, I. Zaliapin, and S. Thompson. A delay differential model of ENSO variability: parametric instability and the distribution of extremes. *Nonlinear Processes in Geophysics*, 15(3):471–433, 2008.

[15] M. J. Suarez. A delayed action oscillator for ENSO. *Journal of Atmospheric Sciences*, 45(21):3283–3287, 1988.

Acknowledgements

I would like to give sincere thanks to my advisor, Professor Xianming Liu of Huazhong university of science and technology, who was generous enough to volunteer his time to give me valuable guidance and helpful advice. I first reached out to Professor Liu some time ago, after studying ODEs and Numerical analysis, in search of research opportunities. Professor Liu recommended that I also study some SDEs, and graciously allowed me to participate in some very interesting seminars. Coincidently, Professor Liu was interested in the open problem of quantifying the evolution of density in dynamical systems with memory, which led to him proposing it as a research topic. I am also indebted to Professor Liu for recommending me relevant reading references, proofreading my countless drafts, as well as encouraging me throughout the entire journey. In addition, I'd like to thank my high school, for allowing me enough free time to pursue my interests, my family, for always believing in me, and the any number of people whose prior research this project is built upon.

As mentioned above, the research topic of computing the evolution of density in dynamical systems with memory was proposed by my advisor. The essence of the algorithm proposed in the paper was formulated by me after a long period of pondering and experimenting. The main challenge I faced while formulating the algorithm was coming up with the approach to compute the probability density using the multiplication theorem with a recursive structure. At first, I had thought about trying to do something with Fokker-Plank equations, since it's the main way probability density for dynamical systems without memory are computed, but in the end, I couldn't follow up with that approach. It was only after many days of thinking did it suddenly occur to me that I could use the multiplication theorem, and it took me a few more days to realize I could use recursion to simplify the computations. From then, I started to make progress steadily. There were already proven results about the existence and continuity of the probability density, as well as convergence of the probability density, so I didn't need to do too much research in that part. I developed my algorithm first for SDDEs, and then generalized it to SFDEs, essentially fixing the big structure of my algorithm. During this process, I am grateful to my advisor for encouraging me and giving me feedback that led to the improvement of my algorithm.

After the big structure of the algorithm was fixed, my advisor recommended a few examples for me to test it out on, which I proceeded to do. I did all the programming involved in the numerical computations and gathered the results. By then, I had enough material to condense into an actual research paper, so I started typesetting everything into LaTeX, which is actually easier said than done. During this stage, my advisor was patient enough to proof-read my drafts over and over again, providing invaluable feedback and

suggestions that made this paper what it is now. Personally, the main challenge for me in this part was getting used to writing in the academic style. I had to make sure my language was formal, clear and organized. They say practice makes perfect, and I can feel that I'm already much better at it. In fact, the formal manuscript was revised so much that it almost looks nothing like my first draft.