Statistica Sinica Preprint No: SS-2022-0210	
Title	De-biasing Particle Filtering for a Continuous Time Hidden Markov Model With a Cox Process Observation Model
Manuscript ID	SS-2022-0210
URL	http://www.stat.sinica.edu.tw/statistica/
DOI	10.5705/ss.202022.0210
Complete List of Authors	Ruiyang Jin, Sumeetpal S. Singh and Nicolas Chopin
Corresponding Authors	Ruiyang Jin
E-mails	rj420@cam.ac.uk

Statistica Sinica

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0210

Statistica Sinica

De-biasing particle filtering for a continuous time hidden Markov model with a Cox process observation model

Ruiyang Jin[1], Sumeetpal S. Singh[2], Nicolas Chopin[3]

[1]*Department of Engineering, University of Cambridge*

[2]*NIASRA, School of Mathematics and Applied Statistics, University of Wollongong*

[3]*ENSAE, Institut Polytechnique de Paris*

Abstract: We develop a (nearly) unbiased particle filtering algorithm for a specific class of continuous-time state-space models in which (a) the latent process X_t is a linear Gaussian diffusion, and (b) the observations arise from a Poisson process with intensity $\lambda(X_t)$. The likelihood and the posterior probability density function of the latent process include an intractable path integral. Our algorithm relies on using Poisson estimates to approximate this integral in an unbiased manner. We show how to tune these Poisson estimates to ensure that, with large probability, all but a few of the estimates generated by the algorithm are positive. Then setting the negative estimates to zero leads to a much smaller bias than that obtained using discretization. We quantify the probability of negative estimates for certain special cases, and show that our particle filter is effectively unbiased. We apply our method to a challenging 3D single molecule tracking example using a Born–Wolf observation model.

Key words and phrases: Sequential Monte Carlo; Continuous-time; Cox process; Diffusions; Hidden Markov model; Particle filter; Path integral; Poisson estimate

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0210

1. Introduction

1.1 Background

Diffusion processes have been used extensively to model continuous-time phenomena in a range of scientific areas, including finance (Merton, 1975), biochemistry (McAdams and Arkin, 1997; Gillespie, 1977), and physics (Obukhov, 1959). These processes are usually applied to model both an observed process and an unobserved signal/state process in a hierarchical model.

This study develops novel methods for the optimal filtering of multivariate diffusion processes observed at irregular time instances, which follow a Cox process with intensity that is a (nonnegative) function of the state process. The complete data likelihood of such a model includes a path integral of the state trajectory (in the intensity function), which is intractable. This precludes using standard particle filters. Another common problem in continuous-time filtering for diffusion processes is the unavailability of transition densities (Fearnhead et al., 2010). In our problem though, the hidden state is described by a linear stoachstic differential equation (SDE), and thus the state transition density is available, although the likelihood remains intractable. Nicolau (2002) proposes a Riemann sum approximation method for approximating intractable path integrals (with respect to time). This approach uses a set number of intermediate points, but results in biased estimates. Nevertheless, Durham and Gallant (2002) use it to develop a

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0210

transition density estimator in a filtering context.

A Poisson estimator is often used to remove the time-discretization error in the numerical approximation of the path integral. The path integral estimate is computed using an (infinite) series expansion expressed as a random finite series, where the random truncation is given by a Poisson random variable. The first Poisson estimator was introduced in the field of statistical physics by Wagner (1988). It was developed further in the computational statistics literature, for example, for unbiased estimations of diffusion transition densities by Beskos et al. (2006), and for sequential importance sampling by Fearnhead et al. (2008); see Papaspiliopoulos (2011) for other developments. One drawback of using a Poisson estimator is that it may return negative values, which can result in an overall negative likelihood estimate, thus prohibiting us from using the likelihood estimate for model calibration in a particle Markov-chain Monte Carlo (MCMC) method (Andrieu et al., 2010). A naive way to ensure positive estimates is to truncate all negative estimates to zero, which may introduce bias into the estimate. Fearnhead et al. (2010) use Wald's identity (for martingales) to generate an unbiased estimate of the path integral that is guaranteed to be positive. However, this method does not seem to yield an unbiased estimate of the likelihood itself (see Section 4.2.3 for an elaboration on this point), and has a bias that appears difficult to quantify.

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0210

1.2 Contributions

We use the standard Poisson estimate, and retain only the positive part of the returned estimate. (In Section 4.2.3, we discuss the retaining the absolute value, which enables us to completely de-bias the estimate.) We quantify the probability of encountering a negative weight (in certain idealized scenarios), and show that this probability decreases exponentially with the inverse of the time interval size over which the estimate is computed. (For some typical experimental settings in our numerical work, the probability is exceptionally small, of the order 10^{-50}.) This exponential decrease in the probability of a negative estimate has several advantages. The first is a rapidly diminishing mean squared error (MSE), for the likelihood estimate within the available CPU time. Second, the probability of a complete run of an N-particle approximation for T/Δ time steps encountering a negative estimate (thus, needing truncation) is extremely small, and is straightforward to control using our proposed (heuristic) tuning procedure. (Here, $[0, T]$ is the time interval for smoothing and Δ is the interval over which the path integral is estimated; thus, we need to estimate T/Δ path integrals for each particle.) To control for a negative weight event, the extra simulation cost per particle per time step is $\mathcal{O}(\Delta)$, and thus the total extra cost is $N \times (T/\Delta) \times \Delta$, which does not increase significantly as Δ tends to zero.

As our second contribution, we apply our methodology to a challenging model

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0210

calibration problem arising from single-molecule fluoresencence microscopy, which is a very popular live-cell imaging technology. We combine our likelihood estimate with the particle marginal Metropolis–Hastings (PMMH) algorithm (Andrieu et al., 2010) to estimate the model parameters for data from observing a diffusing molecule in three dimensions, using a Cox process and a Born–Wolf (BW) observation model. We show that our particle filter significantly outperforms the conventional time discretization-based approach for the intractable path integral, as implemented in d'Avigneau et al. (2022). Our method is shown to have negligible bias owing to our tuning heuristic that controls the occurrence of a negative Poisson path integral estimate (and thus the truncation-induced bias).

The remainder of the paper is organized as follows. Section 2 presents the problem formulation, and Sections 3 and 4 present the particle filtering methodology in continuous time. In Section 3, the particle filter uses a simple time discretization of the path integral, and in Section 4, we present a more sophisticated particle filter that uses a Poisson estimator of the path integral. Our proposed algorithm and accompanying theoretical results on its performance are also presented in Section 4. Experiments, including likelihood estimation, state estimation (smoothing), and parameter estimation, are presented in Section 5. Proofs and additional algorithms can be found in the Supplementary Material.

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0210

2. Problem Formulation

2.1 Notation

The latent continuous-time Markov process $\{X_t\}_{t\geq 0}$ takes values in $\mathcal{X} \subset \mathbb{R}^n$. It has a time-inhomogeneous Markov transition density, $X_{t_k}|(X_{t_{k-1}} = x_{k-1}) \sim f^{\theta}_{t_{k-1},t_k}(x_k|x_{k-1})$, and initial density ν^{θ}. The superscript θ is the parameter of the model and is defined below. By $X \sim \mathcal{N}(\mu, \Sigma)$, we mean that X has the distribution of a Gaussian random vector with mean μ and covariance Σ, whereas $\mathrm{N}(x; \mu, \Sigma)$ is the evaluation of this Gaussian density at x. We use the standard notation $i : j$ to denote the sequence $\{i, i+1, \ldots, j-1, j\}$, and $\lceil x \rceil$ to denote the smallest integer greater than or equal to $x \in \mathbb{R}$. The $\mathcal{Y} \subset \mathbb{R}^m$-valued stochastic process $\{Y_k\}_{k\in\mathbb{Z}_+}$ corresponds to the observed process with observation density $g^{\theta}(y_k|x_k)$. A realization of a Poisson point process on the positive real line is a sequence of increasing time points $0 < t_1 < t_2 < \ldots$ generated according to a nonnegative intensity function $t \mapsto \lambda_t$.

2.2 Hidden Markov Model Formulation

Let $\{(t_i, y_{t_i})\}_{i\in\{1:n_p\}}$ be an observed sequence of nonnegative increasing arrival times $0 < t_i < T$ and arrival locations y_{t_i} of a marked Poisson point process on the real line, recorded in the time interval $[0, T]$. The arrival times are generated by a Poisson point process on $[0, \infty)$ with stochastic intensity function $\lambda(X_t)$, which

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0210

is determined by the latent continuous-time Markov process $\{X_t\}_{t\geq 0} \subset \mathcal{X}$ and a nonnegative real-valued function $\lambda : \mathcal{X} \to \mathbb{R}$. The locations $y_{t_i} \in \mathcal{Y}$ are marks of the point process, and are generated according to the conditional (on $X_{t_i} = x$) probability density function

$$Y_{t_i}|(X_{t_i} = x) \sim g^{\theta}(y|x)dy, \qquad i \in \{1 : n_p\}\,.$$

The *exact* likelihood (Snyder and Miller, 2012, Chapter 7.3.1) is

$$\mathcal{L} = \mathbb{E}\left\{\left(\prod_{i=1}^{n_p} \lambda\left(X_{t_i}\right) g^{\theta}\left(y_{t_i}|X_{t_i}\right)\right\} \times \exp\left(-\int_0^T \lambda\left(X_s\right) ds\right)\right\}, \tag{2.1}$$

where the expected value is computed with respect to the law of $\{X_t\}_{0\leq t\leq T}$.

3. Particle Filtering

We adopt a discretization of the positive real axis, which is divided into segments of maximum length Δ, defined sequentially as follows:

$$\begin{aligned} t_0^{\Delta} &= 0, \\ t_k^{\Delta} &= t_{k-1}^{\Delta} + \min\left\{\Delta, T - t_{k-1}^{\Delta}, \min_{t_i > t_{k-1}^{\Delta}} t_i - t_{k-1}^{\Delta}\right\}, \quad k > 1, \end{aligned} \tag{3.1}$$

where t_i is the (observed) arrival time. Thus, (3.1) defines an increasing sequence of time points $t_0^{\Delta} = 0 < t_1^{\Delta} < \ldots < t_{m-1}^{\Delta} < t_m^{\Delta} = T$, spaced Δ apart, unless the spacing is narrowed to coincide with the arrival of observation y_{t_i} at time t_i, and ensures $\{t_1, \ldots, t_{n_p}\} \subset \{t_1^{\Delta}, \ldots, t_{m-1}^{\Delta}\}$. The *exact* likelihood (2.1) may be

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0210

re-expressed using the time points t_i^Δ as

$$\mathcal{L} = \mathbb{E}\left\{\left(\prod_{i=1}^{n_p}\lambda(X_{t_i})g^\theta(y_{t_i}|X_{t_i})\right)\times\left(\prod_{j=1}^{m}\exp\left(-\int_{t_{j-1}^\Delta}^{t_j^\Delta}\lambda(X_s)\mathrm{d}s\right)\right)\right\}. \tag{3.2}$$

Using the exact likelihood (using an approach such as that in Algorithm 1) for an unbiased estimation using particle filtering is not straightforward, because of the path integrals of $\lambda(X_s)$. A simple approach is to replace the path integral over $[0,T]$ with the following Reimann approximation:

$$\mathcal{L}_\Delta = \mathbb{E}\left\{\left(\prod_{i=1}^{n_p}\lambda(X_{t_i})g^\theta(y_{t_i}|X_{t_i})\right)\times\prod_{j=1}^{m}\exp\left(-\lambda(X_{t_{j-1}^\Delta})(t_j^\Delta - t_{j-1}^\Delta)\right)\right\}. \tag{3.3}$$

(The subscript Δ denotes the dependence on the time discretization, and emphasizes that $\mathcal{L}_\Delta \neq \mathcal{L}$.) The posterior density function of $(X_0, X_1, \ldots, X_m) = (X_{t_0^\Delta}, X_{t_1^\Delta}, \ldots, X_{t_m^\Delta})$ for this time-discretized model is defined in terms of integrals of real-valued test functions h:

$$\begin{aligned}&\int p_\Delta^\theta(x_0,\ldots,x_m)h(x_{0:m})dx_{0:m}\\ &\propto \mathbb{E}\left\{h\left(X_{t_0^\Delta},\ldots,X_{t_m^\Delta}\right)\times\left(\prod_{i=1}^{n_p}\lambda(X_{t_i})g^\theta(y_{t_i}|X_{t_i})\right)\times\prod_{j=1}^{m}\exp\left(-\lambda(X_{t_{j-1}^\Delta})(t_j^\Delta - t_{j-1}^\Delta)\right)\right\}.\end{aligned} \tag{3.4}$$

We can estimate this posterior density function and its likelihood using a conventional particle filter' as described in Algorithm 1 (d'Avigneau et al., 2022).

The estimate $\hat{\mathcal{L}}_\Delta$ returned by Algorithm 1 is an unbiased estimate of the time-discretized likelihood $\mathcal{L}_\Delta$. In the next section, we develop a particle method that approximates the exact (not-discretized) likelihood, and in the numerical section

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0210

Algorithm 1: Bootstrap particle filter

for $i \in \{1:N\}$ **do**
  Sample $X_0^{(i)} \sim \nu^\theta(\cdot)$.
  Set $W_0^{(i)} = \exp\left(-X_0^{(i)}(t_1^\Delta - t_0^\Delta)\right)$.
  Resample $\{X_0^{(i)}, W_0^{(i)}\}$ to obtain $\{\tilde{X}_0^{(i)}, \frac{1}{N}\}$.
end
for $k \in \{1:m-1\}$ **do**
  for $i \in \{1:N\}$ **do**
    Sample $X_k^{(i)} \sim f^\theta_{t_{k-1}^\Delta, t_k^\Delta}(\cdot|\tilde{X}_{k-1}^{(i)})$ and set $X_{0:k}^{(i)} = (\tilde{X}_{0:k-1}^{(i)}, X_k^{(i)})$.
    Set $W_k^{(i)} =$
    $\exp\left(-X_k^{(i)}(t_{k+1}^\Delta - t_k^\Delta)\right) \times \prod_{j=1}^{n_p}\left(\lambda(X_k^{(i)})g^\theta(y_{t_j}|X_k^{(i)})\right)^{\mathbb{I}[t_k^\Delta \le t_j < t_{k+1}^\Delta]}$.
    `% Find all` y_{t_j} `with` $t_j \in [t_k^\Delta, t_{k+1}^\Delta)$.
    Resample $\{X_{0:k}^{(i)}, W_k^{(i)}\}$ to obtain $\{\tilde{X}_{0:k}^{(i)}, \frac{1}{N}\}$.
  end
end
Compute the (unbiased) estimate of the likelihood in (3.3):

$$\hat{\mathcal{L}}_\Delta = \prod_{k=0}^{m-1}\left\{\frac{1}{N}\sum_{i=1}^{N} W_k^{(i)}\right\}. \tag{3.5}$$

(Section 5), we compare its estimation accuracy with that of Algorithm 1 applied to model (3.4).

4. Particle Filtering to Mitigate the Model Discretization Error

We propose a simple method to nearly unbiasedly estimate the true likelihood $\mathcal{L}$. We discretize the path integrals into smaller Δ length time integrals, $\exp\left(-\int_t^{t+\Delta}\lambda(X_s)ds\right)$, which are amenable to a simple unbiased estimation, and have a probability of being positive that approaches one rapidly as Δ tends to zero. We truncate a negative

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0210

estimate to zero. Combined with the rarity of such events, it is simple to quantify the bias, which is also shown to decrease rapidly as Δ tends to zero. This estimate can be used within particle filtering and a particle MCMC; such methods are known as particle filtering with "random weights", as in Rousset and Doucet (2006), Fearnhead et al. (2008), and Fearnhead et al. (2010).

Specifically, we construct real-valued random variables $E_1, \ldots, E_m$, which are conditionally independent given $X_{t_0^\Delta}, \ldots, X_{t_m^\Delta}$ (in the manner made precise in (4.1)), and each unbiasedly estimates the corresponding term $\exp(-\int_{t_{i-1}^\Delta}^{t_i^\Delta} \lambda(X_s)ds)$ in the manner of (4.2):

$$p(e_1, \ldots, e_m | x_{t_0^\Delta}, \ldots, x_{t_m^\Delta}) = \prod_{i=1}^{m} p_{t_{i-1}^\Delta, t_i^\Delta}(e_i | x_{t_{i-1}^\Delta}, x_{t_i^\Delta}) \tag{4.1}$$

$$\int_{-\infty}^{\infty} e_i p_{t_{i-1}^\Delta, t_i^\Delta}(e_i | x_{t_{i-1}^\Delta}, x_{t_i^\Delta}) de_i = \mathbb{E}\Big\{ \exp\big(-\int_{t_{i-1}^\Delta}^{t_i^\Delta} \lambda(X_s)ds\big) \Big| X_{t_{i-1}^\Delta} = x_{t_{i-1}^\Delta}, X_{t_i^\Delta} = x_{t_i^\Delta} \Big\}. \tag{4.2}$$

With these random variables $E_1, \ldots, E_m$, we retain the unbiasedness of the estimate of the numerator and denominator (the likelihood),

$$\int p_T(x_{0:m}) h(x_{0:m}) dx_{0:m} \propto \mathbb{E}\Big\{ h(X_{t_0^\Delta}, X_{t_1^\Delta}, \ldots, X_{t_m^\Delta}) \times \Big(\prod_{i=1}^{n_p} \lambda(X_{t_i}) g^\theta(y_{t_i} | X_{t_i}) \Big) \times \prod_{j=1}^{m} E_j \Big\},$$

which follows from a conditioning expectation argument. For $k \in \{1 : m\}$, let

$$\begin{aligned} &\int p_{t_k^\Delta}(x_0, \ldots, x_k) h_k(x_{0:k}) dx_{0:k} \\ &\propto \mathbb{E}\Big\{ h_k(X_{t_0^\Delta}, X_{t_1^\Delta}, \ldots, X_{t_k^\Delta}) \times \Big(\prod_{i=1}^{n_p} [\lambda(X_{t_i}) g^\theta(y_{t_i} | X_{t_i})]^{\mathbb{I}[t_i \le t_k^\Delta]} \Big) \times \prod_{j=1}^{k} E_j \Big\} \end{aligned} \tag{4.3}$$

where, $t_m = T$. Once we have defined (4.1), it is straightforward to construct a

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0210

particle approximation of the conditional probability density functions (4.3). These posterior densities, unlike (3.4), do not have a time discretization bias. Our particle filtering algorithm, detailed in Algorithm 3, also returns an estimate of the exact likelihood (3.2). The next subsection explains how to construct these variables E_i using the Poisson estimate approach. The following subsections will explain how to ensure that the probability of $E_i < 0$ is negligible.

4.1 The Poisson Estimator

We first consider a fixed trajectory $\{X_s\}_{0<s\leq t_1^\Delta}$. Then,

$$\begin{aligned}
\exp\left(-\int_0^{t_1^\Delta}\lambda(X_s)\mathrm{d}s\right) &= \exp(c)\exp(I-c)\\
&= \exp(c)\sum_{k=0}^{\infty}\frac{(I-c)^k}{k!}\\
&= \exp(c+\eta)\sum_{k=0}^{\infty}\exp(-\eta)\frac{\eta^k}{k!}\left(\frac{I-c}{\eta}\right)^k\\
&= \exp(c+\eta)\sum_{k=0}^{\infty}\exp(-\eta)\frac{\eta^k}{k!}\prod_{i=1}^{k}\mathbb{E}_{\tau_i}\left(\frac{-t_1^\Delta\lambda(X_{\tau_i})-c}{\eta}\right)\\
&= \exp(c+\eta)\mathbb{E}_\kappa\left[\prod_{i=1}^{\kappa}\mathbb{E}_{\tau_i}\left(\frac{-t_1^\Delta\lambda(X_{\tau_i})-c}{\eta}\right)\right],
\end{aligned}$$

where $-t_1^\Delta\lambda(X_{\tau_i})$ are the unbiased estimates of $I = -\int_0^{t_1^\Delta}\lambda(X_s)ds$. The above derivation follows the approach outlined in Papaspiliopoulos (2011).

The inclusion of the constant c optimises the resulting estimator. The inclusion of the $\mathcal{P}\mathrm{o}(\eta)$ distribution allows an unbiased estimate to be based on a truncated sum. Finally, $\mathbb{E}_\kappa(\cdot)$ and $\mathbb{E}_{\tau_i}(\cdot)$ denote the expectations with respect to the indepen-

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0210

dent random variables $\kappa \sim \mathcal{P}o(\eta)$ and $\tau_i \sim \mathcal{U}(0, t_1^\Delta)$, respectively, and $\{X_s\}_{0<s\leq t_1^\Delta}$ is treated as a fixed trajectory. The final line yields the resulting unbiased estimator

$$E_1 = \exp(c+\eta)\Big[\mathbb{I}_{\{\kappa=0\}} + \mathbb{I}_{\{\kappa>0\}}\Big(\prod_{i=1}^{\kappa}\frac{-t_1^\Delta\lambda(X_{\tau_i})-c}{\eta}\Big)\Big] \tag{4.4}$$

as the sample from $E_1 \sim p(e_1|x_0, x_{t_1^\Delta})$.

Papaspiliopoulos (2011) discusses how to choose c and η in order to minimize the variance of the estimate showing that $c^\star = I - \eta$ is the value of c that minimizes the variance (for a fixed η). Our approach is slightly different, in that we aim to control the probability of the estimate being negative. For that purpose, we set $c = -t_1^\Delta\lambda(X_0) - \eta$ (which can also be seen as a tractable approximation of $c^\star$). This yields

$$E_1 = \exp\left\{-t_1^\Delta\lambda\left(X_0\right)\right\)\left[\mathbb{I}_{\{\kappa=0\}} + \mathbb{I}_{\{\kappa>0\}}\left(\prod_{i=1}^{\kappa}\left[1+\frac{t_1^\Delta}{\eta}\left(\lambda\left(X_0\right)-\lambda\left(X_{\tau_i}\right)\right)\right]\right)\right]. \tag{4.5}$$

We discuss how to control the probability of a negative estimate using the parameters (η, Δ) in the next sub-section.

The Poisson estimator for any time interval $t_{i-1}^\Delta \leq t \leq t_i^\Delta$ is detailed in Algorithm 2. Note that we assume we can exactly sample X_{τ_j} from $p(x_\tau|x_{\tau_{j-1}})$, for $j \in \{1:\kappa\}$. This is possible for linear Gaussian diffusions, as discussed in the introduction; see **S1** of the Supplementary Material for details. The particle filter with the Poisson estimator is described in Algorithm 3. Step 8 of this algorithm makes

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0210

Algorithm 2: PE$(\eta, t^{\Delta}_{i-1}, t^{\Delta}_{i}, X_{t^{\Delta}_{i-1}})$

Input: $\eta, t^{\Delta}_{i-1}, t^{\Delta}_{i}, X_{t^{\Delta}_{i-1}}$

1 Generate $\kappa \sim \mathcal{P}o(\eta)$.

2 Generate $\tau_1, \tau_2, \ldots, \tau_\kappa \sim \mathcal{U}(t^{\Delta}_{i-1}, t^{\Delta}_{i})$, sort them in ascending order and relabel them so that $\tau_1 < \tau_2 < \ldots, < \tau_\kappa$.

3 Sequentially sample X_{τ_j} from $p(x_\tau | x_{\tau_{j-1}})$ for $j \in \{1 : \kappa\}$ where $\tau_0 = t^{\Delta}_{i-1}$. Sample $X_{t^{\Delta}_{i}}$ from $p(x_{t^{\Delta}_{i}} | x_{\tau_\kappa})$.

4 Compute and return the estimate:

$$E = \exp\left(-(t^{\Delta}_{i} - t^{\Delta}_{i-1}) \times \lambda\left(X_{t^{\Delta}_{i-1}}\right)\right)\left[\mathbb{I}_{\{\kappa=0\}} + \mathbb{I}_{\{\kappa>0\}}\left(\prod_{j=1}^{\kappa}\left[1 + \frac{t^{\Delta}_{i} - t^{\Delta}_{i-1}}{\eta}\left(\lambda(X_{t^{\Delta}_{i-1}}) - \lambda(X_{\tau_j})\right)\right]\right)\right].$$

Output: $(E, X_{t^{\Delta}_{i}})$ `% The sample from` $p(e, x_{t^{\Delta}_{i}} | X_{t^{\Delta}_{i-1}})$

a call to Algorithm 2 to obtain the desired samples $E^{(i)}_k$ from $p(e_k | X^{(i)}_{t^{\Delta}_{k-1}}, X^{(i)}_{t^{\Delta}_{k}})$.

4.2 Negative Poisson Estimate Control

Although the Poisson estimator can return negative values, the following lemma shows that we can control the probability of this happening by adjusting (η, Δ) which decays exponentially fast in Δ.

Lemma 1. *Let $\{X_s\}_{0 \le s \le \Delta}$ be a one-dimensional Brownian motion that starts at $X_0 = x_0$. Consider the estimate (4.5) (with $t^{\Delta}_1 = \Delta$) of the path integral $\mathbb{E}\left\{\exp\left(-\int_0^{\Delta} \lambda(X_s) ds\right) \middle| X_\Delta = x_\Delta\right\}$. Let $\lambda(\cdot)$ be a nonnegative l-Lipschitz func-*

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0210

Algorithm 3: Bootstrap particle filter in continuous time

Find Δ (4.10) and define time steps (3.1).
for $i \in \{1:N\}$ **do**
  Sample $\tilde{X}_{t_0^\Delta}^{(i)} \sim \nu^\theta(\cdot)$ and set $W_0^{(i)} = \frac{1}{N}$.
Estimate $\hat{l}_0$. `% See Section 4.2.1.`
for $k \in \{1:m\}$ **do**
  for $i \in \{1:N\}$ **do**
    Set $\eta_k = \left(t_k^\Delta - t_{k-1}^\Delta\right) \hat{l}_{k-1}$.
    Sample $(E_k^{(i)}, X_{t_k^\Delta}^{(i)}) \leftarrow \mathrm{PE}(\eta_k, t_{k-1}^\Delta, t_k^\Delta, \tilde{X}_{t_{k-1}^\Delta}^{(i)})$ and set
    $\left(X_{t_0^\Delta}^{(i)}, \ldots, X_{t_k^\Delta}^{(i)}\right) = \left(\tilde{X}_{t_0^\Delta}^{(i)}, \ldots, \tilde{X}_{t_{k-1}^\Delta}^{(i)}, X_{t_k^\Delta}^{(i)}\right)$.
    Update $\hat{l}_k$ using (4.11).
    Set $W_k^{(i)} = \max\{E_k^{(i)}, 0\} \times \prod_{j=1}^{n_p} \left(\lambda(\tilde{X}_{t_{k-1}^\Delta}^{(i)}) g^\theta(y_{t_j} | \tilde{X}_{t_{k-1}^\Delta}^{(i)})\right)^{\mathbb{I}[t_j = t_{k-1}^\Delta]}$.
    `% Incorporating` y_{t_j} `with` $t_j = t_{k-1}^\Delta$.
    Resample $\left\{(X_{t_0^\Delta}^{(i)}, \ldots, X_{t_k^\Delta}^{(i)}), W_k^{(i)}\right\}$ to obtain $\left\{(\tilde{X}_{t_0^\Delta}^{(i)}, \ldots, \tilde{X}_{t_k^\Delta}^{(i)}), \frac{1}{N}\right\}$.
Compute the likelihood estimate:

$$\hat{\mathcal{L}} = \prod_{k=1}^{m} \left\{ \frac{1}{N} \sum_{i=1}^{N} W_k^{(i)} \right\}. \tag{4.6}$$

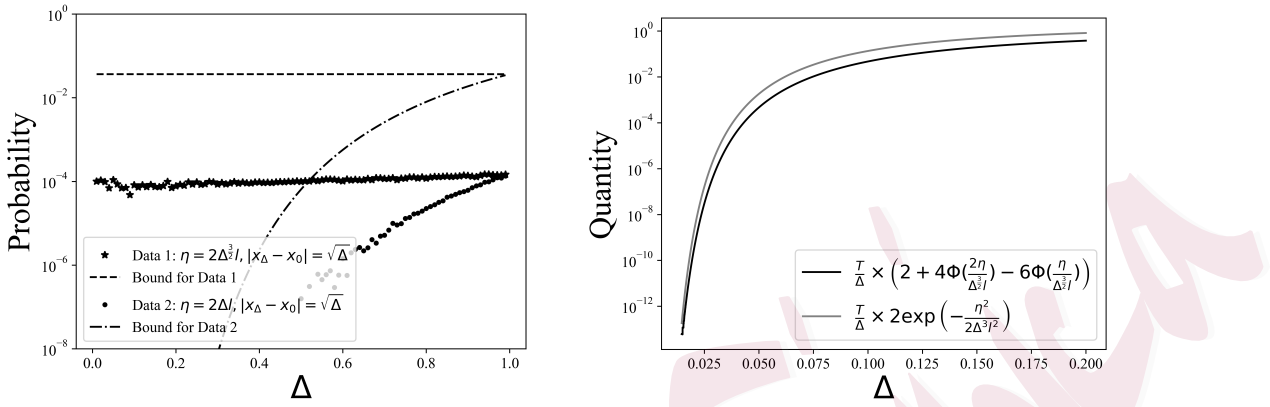
tion. Then the following bound holds when $\eta > \Delta l \, |x_\Delta - x_0|$*:*

$$\Pr\left(E_1 < 0 | \kappa > 0, X_\Delta = x_\Delta\right) < 2 \exp\left\{ - \frac{\frac{2\eta}{\Delta l}\left(\frac{\eta}{\Delta l} - |x_\Delta - x_0|\right)}{\Delta} \right\}. \tag{4.7}$$

Proof. See **S2** of the Supplementary Material. □

Note that the estimate is trivially positive when $\kappa = 0$, and hence the bound is given conditionally on $\kappa > 0$. Figure 1a shows how $\Pr(E_1 < 0 | \kappa > 0)$ and its corresponding bound evolve as Δ changes for different choices of η. Each data

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0210



(a) (b)

Figure 1: Plot of (a) (4.7) for various choices of η, and (b) (4.8) multiplied by T/Δ for the design choice of $\eta = \Delta l$ when $T = 1$ with fitted relationship vs. Δ.

point is a Monte Carlo estimate of the conditional probability (conditioned on $\kappa > 0$) that the random variable (4.4), with $t_1^\Delta = \Delta$, is negative. The Monte Carlo estimate of the conditional probability is computed for various choices of η and $|x_\Delta - x_0|$ using 10^8 experiments each. (4.7) suggests that choosing $\eta = c\Delta^{\frac{3}{2}}l$ with $|x_\Delta - x_0| = d\Delta^{\frac{1}{2}}$ (for some positive constants c and d) results in a constant bound. This is reflected in the straight line behavior of Data 1 in Figure 1a. For contrast, we also show the bound on the conditional probabilities and compute the bound when averaging over X_Δ. Combining the bound (4.7) with the expansion $\mathbb{I}_{[E_1<0]} \leq \mathbb{I}_{[E_1<0]}\mathbb{I}_{[\eta>\Delta l|X_\Delta - x_0|]} + \mathbb{I}_{[\eta\leq\Delta l|X_\Delta - x_0|]}$, we compute the unqualified bound for $\Pr(E_1 < 0|\kappa > 0)$ as

$$\Pr(E_1 < 0|\kappa > 0) \leq 2 + 4\Phi\left(\frac{2\eta}{\Delta^{\frac{3}{2}}l}\right) - 6\Phi\left(\frac{\eta}{\Delta^{\frac{3}{2}}l}\right), \tag{4.8}$$

where Φ is the cumulative distribution function of a standard normal distribu-

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0210

tion. (The proof is provided in **S3** of the Supplementary Material.) In Section 4.2.1, we advocate a design choice of $\eta = \Delta l$ (with the Lipschitz constant estimated in a causal manner with the population of particles) to ensure the simulation cost decreases proportionally with the time discretization Δ. An estimate of $\mathbb{E}\left\{\exp\left(-\int_0^T \lambda(X_t)dt\right)\right\}$ or $\mathbb{E}\left\{\exp\left(-\int_0^\Delta \lambda(X_t)dt\right)\cdots\exp\left(-\int_{\lfloor\frac{T}{\Delta}\rfloor\Delta}^T \lambda(X_t)dt\right)\right\}$ includes the product of T/Δ (conditionally independent) estimates for the individual intervals. Using (4.8), this estimate is negative with a probability no greater than $\Pr(E_1 < 0|\kappa > 0) \times T/\Delta$. Figure 1b illustrates how the bound in (4.8), when multiplied by T/Δ, decays with the choice $\eta = \Delta l$.

4.2.1 Design choice for (η, Δ)

We employ reasoning similar to the above to bound the probability of Algorithm 3 encountering a negative Poisson estimate. For a step-size Δ, an N-particle implementation has $\lceil T/\Delta \rceil$ forward steps, and the event of encountering at least one negative Poisson estimate is $\left\{\bigcup_{n=1}^N \bigcup_{i=1}^{\lceil T/\Delta\rceil}\{E_i^{(n)} < 0\}\right\}$. Using Lemma 1, its probability may be bounded above by the union bound

$$\Pr\left(\bigcup_{n=1}^{N}\bigcup_{i=1}^{\lceil T/\Delta\rceil}\{E_i^{(n)} < 0\}\right) \leq \sum_{n=1}^{N}\sum_{i=1}^{\lceil T/\Delta\rceil}\Pr\left(E_i^{(n)} < 0\right) < \left\lceil\frac{NT}{\Delta}\right\rceil \times 2\exp\left\{-\frac{\frac{2\eta}{\Delta l}(\frac{\eta}{\Delta l} - d\sqrt{\Delta})}{\Delta}\right\},$$

where we assume that $|x_\Delta - x_{(i-1)\Delta}| \leq d\sqrt{\Delta}$, for all $i \in \{1 : \lceil T/\Delta\rceil\}$, for some constant $d > 0$ and $\eta/(\Delta l) > d\sqrt{\Delta}$. (A similar heuristic can also be found using (4.8).) We can choose η and Δ (within the constraints $\eta \geq \Delta^{\frac{3}{2}} l$ and $\eta/(\Delta l) > d\sqrt{\Delta}$)

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0210

to ensure that the probability of encountering a negative estimate is at most ϵ, as follows:

$$\left\lceil \frac{NT}{\Delta} \right\rceil \times 2\exp\left\{ - \frac{\frac{2\eta}{\Delta l}(\frac{\eta}{\Delta l} - d\sqrt{\Delta})}{\Delta} \right\} \leq \epsilon. \tag{4.9}$$

For example, using $\eta = \Delta l$, the bound will fall below ϵ once Δ is sufficiently small, say $\Delta = \bar{\Delta}$, and will continue to hold as Δ decreases further because the left-hand side decreases as Δ decreases. A similar heuristic can also be found using (4.8). In summary, set $\eta = \Delta l$ and

$$\Delta = \sup\{\Delta > 0 : (4.9) \text{ and } \lceil NT/\Delta \rceil \times (4.8) \leq \epsilon\}. \tag{4.10}$$

One can apply numerical methods such as Newton's method to solve (4.9). In addition, ϵ can be exceptionally small, for example, $\Delta = 0.01$, $NT = 10^4, d = 3$, and $\eta = \Delta l$ yields $\epsilon \approx 10^{-55}$.

The design choice $\eta_k = \Delta \hat{l}_{k-1}$ can be computed sequentially in Algorithm 3. Here, $\hat{l}_{k-1}$ is the empirical Lipschitz constant, updated sequentially, as follows:

$$\hat{l}_k := \max\left\{ \max_{i \in \{1:N\}} \frac{|\lambda(X^{(i)}_{t^\Delta_k}) - \lambda(X^{(i)}_{t^\Delta_{k-1}})|}{|X^{(i)}_{t^\Delta_k} - X^{(i)}_{t^\Delta_{k-1}}|}, \hat{l}_{k-1} \right\}, \tag{4.11}$$

where the initial estimate $\hat{l}_0$ is chosen as the maximum ratio estimate, as in (4.11), but computed using the particle set at time t^Δ_0 only, and the maximum is found over $i \neq j \in \{1 : N\}$. These design choices for $\hat{l}_k$ and η_k for Algorithm 3 are used in all the numerical experiments presented in Section 5.

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0210

4.2.2 Truncation Bias

In Algorithm 3, we truncate the negative Poisson estimates to zero, which induces a bias. Hence, we wish to study the bias of this truncated estimate for time discretization, $0 < \Delta < \cdots < m\Delta = T$, when Δ approaches zero, that is,

$$\mathbb{E}\left\{\exp\left(-\int_0^T \lambda(X_s)ds\right)\right\} - \mathbb{E}\left\{E_1^+ \cdots E_m^+\right\},$$

where $E_i^+ = E_i \mathbb{I}_{A_i^c}$ is the truncated Poisson estimate, and A_i denotes the event $E_i < 0$. To do so, we bound the omitted term $\mathbb{I}_A \prod_{i=1}^m E_i$, where $A = A_1 \cup \ldots \cup A_m$, using the following lemma.

Lemma 2. *Let $\{X_s\}_{0\le s\le\Delta}$ be a one-dimensional Brownian motion that starts at $X_0 = x_0$. Let $\lambda(\cdot)$ be a nonnegative l-Lipschitz function, and consider the estimate of the path integral*

$$\mathbb{E}\left\{\exp\left(-\int_0^{\Delta} \lambda(X_t)dt\right)\cdots\exp\left(-\int_{(m-1)\Delta}^{m\Delta} \lambda(X_t)dt\right)\right\} = \mathbb{E}\left\{E_1 \cdots E_m\right\},$$

where $(E_{i+1}, X_{(i+1)\Delta}) \leftarrow \mathrm{PE}(\Delta l, i\Delta, (i+1)\Delta, X_{i\Delta})$ (see Algorithm 2), for $i = 0, \ldots, m-1$. Then, the following bound holds:

$$\left|\mathbb{E}\left\{\mathbb{I}_A \prod_{i=1}^m E_i\right\}\right| \le \exp(\frac{Tl}{2}) \times \left(\frac{1+4\Delta^2 l}{1-4\Delta^2 l}\right)^{\frac{m}{2}} \times m^{\frac{1}{2}} \left[2\exp\left(-\frac{1}{2\Delta}\right)\right]^{\frac{1}{2}}.$$

Proof. See **S4** of the Supplementary Material. □

For $m = T/\Delta$, the second (ratio) term in the product recedes quickly to one as Δ approaches zero, which implies the final term dominates the bias. For $m = T/\Delta$, the final term also tends to zero. Based on this result, as an indicative trend, the

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0210

square of the relative bias (which contributes additively in the relative MSE (rMSE) calculation) of Algorithm 3 is of the order

$$\frac{\left(\mathcal{L} - \mathbb{E}(\hat{\mathcal{L}})\right)^2}{\mathcal{L}^2} \leq \text{const}(T) \times \frac{1}{\Delta} \exp\left(-\frac{1}{2\Delta}\right),$$

where $\hat{\mathcal{L}}$ is given in (4.6). This result is commented on further in Section 5.1.

4.2.3 Further comments

The following idea, based on *Wald's identity* for sampling, is used by Fearnhead et al. (2010) to deal with negative weights in particle filtering. We describe it here in the context of a single step within particle filtering, and discuss its implications for estimating the likelihood. Consider $X_0 \sim \nu^\theta$, and let $G^\theta(x_0)$ be a nonnegative function, also assumed to be θ-dependent. The aim is to estimate the likelihood $L(\theta) = \mathbb{E}^\theta\left(G^\theta(X_0)\right)$. Assume there exists an unbiased estimate of $G^\theta(x_0)$ for any (θ, x_0), defined as follows. Let $p^\theta(e|x_0)$ be a conditional probability density function on the real line with mean $\int_{-\infty}^{\infty} e p^\theta(e|x_0)\mathrm{d}e = G^\theta(x_0)$. Given X_0, let $E^{(i)}$, for $i = 1, 2, \ldots$, be independent samples from $p^\theta(e|X_0)$, and let $K = \inf\left\{k > 0 : E^{(1)} + \ldots + E^{(k)} > 0\right\}$. Then, $\hat{L} = \sum_{i=1}^{K} E^{(i)}$ has mean

$$\mathbb{E}^\theta(\hat{L}) = \mathbb{E}^\theta\left(G^\theta(X_0)\mathbb{E}^\theta(K|X_0)\right) \neq L(\theta) \times \text{constant},$$

where the product $G^\theta(X_0)\mathbb{E}^\theta(K|X_0)$ is Wald's identity, $\mathbb{E}^\theta(K|X_0)$ is the mean of the number of independent draws needed to ensure positivity, and the constant on the right is θ-independent; we need the θ-independent constant to use the

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0210

method for model calibration. This approach of sampling until the estimate is positive is proposed in Fearnhead et al. (2010) to address the event that a negative estimate is returned by $p^\theta(e|X_0)$. The constant $\mathbb{E}^\theta(K|X_0)$ seems to play no role in a particle filtering algorithm, because the weights are normalized before being used as an input to the resampling step. However, $\mathbb{E}^\theta(K|X_0)$, which is clearly X_0-dependent, can be θ-dependent as well, for example, as it would for $G^\theta(x_0) = \mathbb{E}^\theta_{x_0}\{\exp(-\int_0^\Delta \lambda(X_s)\mathrm{d}s)\}$ and its estimate $E^{(i)}$ returned by Algorithm 2 (for $\mathrm{PE}(\eta, 0, \Delta, x_0)$), because K depends on the law of $\{X_t\}_t$. Furthermore, the function $G^\theta(x_0)$ can be θ-dependent. Because there is no easy way to compute or remove the factor $\mathbb{E}^\theta(K|X_0)$, this precludes its use in, for example, a PMMH sampler, which requires a (positive) unbiased estimator of $L(\theta)$ to generate MCMC samples from the posterior density of the model parameters θ. We provide several experiments in **S5** of the Supplementary Material to show that the idea of using Wald's identity for sampling returns biased estimates.

Finally, note that it is possible to adapt our approach slightly to return (perfectly) unbiased estimates. Recall that a particle filter such as Algorithm 3 may return an unbiased estimate of not only the normalizing constant, but also, more generally, of any unnormalized path expectation (Del Moral, 2004); that is, the quantity

$$\hat{\mathcal{L}} \times \frac{\sum_{i=1}^N W_m^{(i)} \varphi(X_{t_0^\Delta}^{(i)}, \ldots, X_{t_m^\Delta}^{(i)})}{\sum_{i=1}^N W_m^{(i)}} \tag{4.12}$$

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0210

is an unbiased estimate of

$$\mathbb{E}\left\{\left(\prod_{i=1}^{n_p}\lambda(X_{t_i})g^{\theta}(y_{t_i}|X_{t_i})\right)\Psi(X_{t_0^{\Delta}},\ldots,X_{t_m^{\Delta}})\times\varphi(X_{t_0^{\Delta}},\ldots,X_{t_m^{\Delta}})\right\},$$

where $\Psi(\cdot)$ is the expectation of a product of Poisson estimates of the form $\prod_{i=1}^{m}\max(0,E_i)$. This quantity is equal to $\exp\left(-\int_0^T\lambda(X_s)ds\right)$ if we replace each truncated estimate $\max(0,E_i)$ with the estimate E_i itself.

We may use this to estimate the marginal likelihood of an alternative model in an unbiased way, based on a different likelihood for the data (given the states). In particular, consider a variant of Algorithm 3 in which $\max\{E_k^{(i)},0\}$ is replaced with $|E_k^{(i)}|$ in line 11. (Adapt the definition of Ψ accordingly.) The weights remain nonnegative, and the output remains biased (for estimating the true likelihood $\mathcal{L}$). In (4.12), replace $\varphi(\cdot)$ with $(-1)^n$, where n is the number of negative Poisson estimates E_k that have occurred while constructing the considered trajectory (the argument of $\varphi(\cdot)$). It is easy to see that this is an unbiased estimate of the true likelihood $\mathcal{L}$. (Formally, φ is then a function of both the state trajectory and the E_k variables generated while constructing that trajectory).

In our numerical experiments, we set the tuning parameters to make the number of occurrences of negative weights almost zero, so we find no practical benefit in removing the bias (entirely). However, this approach may be useful for more complicated scenarios.

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0210

5. Numerical Experiments

In this section, we present numerical examples to compare Algorithm 1 and Algorithm 3 in terms of their likelihood estimation, smoothing, and model calibration using a particle MCMC.

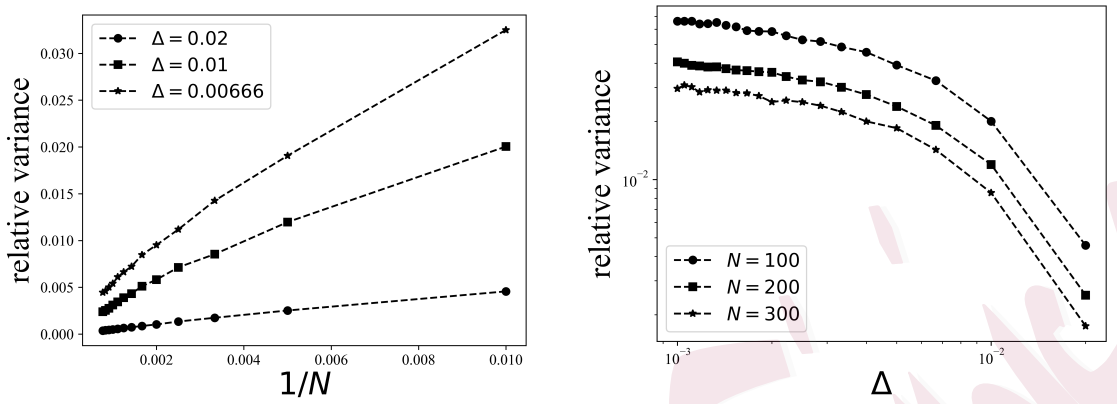
5.1 One-dimensional example with exact calculation

We first consider a simple example in which the state X_t is a one-dimensional Brownian motion and X_t is observed in zero-mean unit-variance Gaussian noise. The intensity function of the Cox process is $\lambda(x) = x + 10$. The state starts at $x_0 = 0$ at time $t = 0$, and the record of observations stops at time $T = 2$.

The integration that defines this likelihood can be computed exactly, and thus can serve as a ground truth; see **S6** of the Supplementary Material. We assume we observe $n_p = 2$ data-points to make it possible to perform a large number of runs; see **S8** of the Supplementary Material for extra results with $n_p > 2$. For the analysis below, we use the relative MSE (rMSE) as the metric to measure the quality of the likelihood estimates.

The numerical results displayed in Figure 2a show that the estimate of $\mathcal{L}_\Delta$ given by Algorithm 1, for any Δ, has a relative variance that is inversely proportional to the number of particles N used in the particle filter, where the relative variance is defined as $\mathbb{E}\{(\hat{\mathcal{L}}_\Delta)^2\}/\mathcal{L}_\Delta^2 - 1$. (Note though that the slope varies very slightly

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0210



(a) (b)

Figure 2: Plot of relative variance, defined as $\mathbb{E}\{(\hat{\mathcal{L}}_\Delta)^2\}/\mathcal{L}_\Delta^2 - 1$, for $\hat{\mathcal{L}}_\Delta$ given by Algortihm 1 in (a) versus $1/N$ and in (b) as Δ varies on a log scale.

with $1/N$.) In Figure 2b, as expected, the relative variance for a fixed N stabilizes as Δ decreases. This is because a time-discretized particle system with systematic resampling converges to a continuous time limit as Δ approaches zero, as shown in Chopin et al. (2022). For any sufficiently smooth function $\lambda(\cdot)$, the weak error of the Euler scheme (i.e., the relative bias $(\mathcal{L}_\Delta/\mathcal{L}) - 1$ in our case) is at most of order Δ (Kloeden and Platen, 2011, Chapter 17). Overall, this implies the following empirical relationship for all values of n_p when Δ is small:

$$\text{rMSE} = \frac{1}{\mathcal{L}^2}\mathbb{E}\left\{\left(\hat{\mathcal{L}}_\Delta - \mathcal{L}\right)^2\right\} = \frac{c_1}{N} + c_2\Delta^2 = \frac{c_1}{\mathcal{C}\Delta} + c_2\Delta^2, \tag{5.1}$$

where $\mathcal{C}$ denotes the CPU time spent running the particle filter (Algorithm 1) to completion. In the last equality, we use the relationship that $\mathcal{C}$ increases linearly with NT/Δ, which corresponds to T/Δ propagation steps for N particles. (Figure 3 confirms (5.1).) For a fixed CPU time $\mathcal{C}$, the value of Δ that minimizes the rMSE

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0210

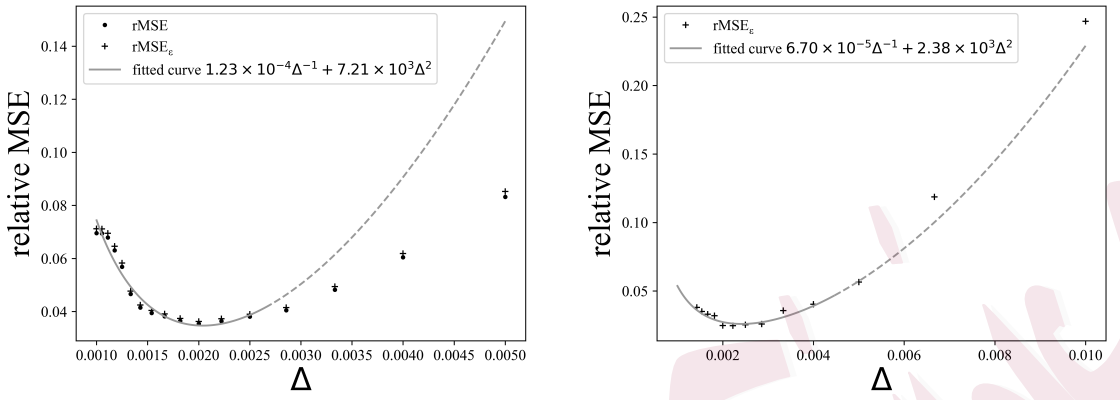
is $\Delta^* = \left(\frac{c_1}{c_2\mathcal{C}}\right)^{\frac{1}{3}}$. Substituting this Δ^* into (5.1) gives the best rMSE value for each $\mathcal{C}$, of order $\mathcal{O}(\mathcal{C}^{-\frac{2}{3}})$ as shown in Figure 4. Similarly, we can apply the same idea to determine Δ that minimizes the rMSE for Algorithm 3, as follows:

$$\text{rMSE} \leq \frac{c_1}{N} + \frac{c_2}{\Delta}\exp\left(-\frac{1}{2\Delta}\right) = \frac{c_1}{\mathcal{C}\Delta} + \frac{c_2}{\Delta}\exp\left(-\frac{1}{2\Delta}\right). \tag{5.2}$$

Because the minimization problem cannot be solved exactly, we use a surrogate for Δ^*, in its vicinity, by minimizing $f(\Delta) = \frac{c_1}{\mathcal{C}\Delta} + c_2\exp\left(-\frac{1}{2\Delta}\right)$. Note that $\text{rMSE}(\Delta) > f(\Delta)$, for $0 < \Delta < 1$. Minimizing this equation gives $\Delta^* = \left(2\log\left(\frac{c_2\mathcal{C}}{2c_1}\right)\right)^{-1}$. Hence, Δ^*, not being the true minimizer of (5.2), is a more conservative solution. Substituting this Δ^* into (5.2) gives an indication of the best rMSE value for each $\mathcal{C}$, of order $\mathcal{O}\left(\mathcal{C}^{-1}\log(\mathcal{C})\right)$. In practice, we do not recommend this optimization. Instead, choose (Δ, η), as discussed in Section 4.2.1, and then stick to this choice, even if more CPU time $\mathcal{C}$ becomes available.

We define rMSE_ϵ as in (5.1), with $\mathcal{L}$ replaced with $\mathcal{L}_{\text{MC}} = \mathcal{L} + \epsilon$. Recall that we denote by $\mathcal{L}_{\text{MC}}$ the Monte Carlo estimate returned by the modified Algorithm 3, which uses the true path integral given by **S6.1** of the Supplementary Material, rather than the Poisson estimate. We ensure that the Monte Carlo error ϵ is sufficiently small that our conclusions when comparing the accuracy of Algorithms 1 and 3 are not rendered inaccurate for the case $n_p > 2$. We use $n_p = 2$ to choose a value of ϵ that ensures that the best Δ found using rMSE_ϵ is close enough to the desired (best) Δ for the rMSE.

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0210



(a) (b)

Figure 3: Plot of rMSE versus Δ for a fixed 1.5s CPU time for likelihood estimates computed by Algorithm 1 (using (3.5)) for (a) $n_p = 2$ and (b) $n_p = 13$. Overlaid is the fitted relationship in (5.1) to a range of Δ-values around the minimum of rMSE, illustrated by the solid segment of the line. The dashed segment covers data points not used in fitting. The coefficients of the fitted curve to rMSE$_\epsilon$ are only 2.4% different to that of the rMSE data points; hence, it would be indistinguishable graphically.

Continuing with $n_p = 2$, Figure 3a reports the rMSE and rMSE$_\epsilon$ of Algorithm 1 for a fixed CPU budget and different Δ-values, with the expected relationship in (5.1) fitted to a range of Δ-values around the minimum. Here, rMSE$_\epsilon$ uses $\mathcal{L}_{\text{MC}}$, which is the average estimate of $\mathcal{L}$ given by 10^6 runs of the modified Algorithm 3, with each run using $N = 10^6$ particles. We calculate the relative error between Δ_ϵ^* and Δ^*, and between rMSE$_\epsilon(\Delta_\epsilon^*)$ and rMSE(Δ^*), using their fitted values for c_1 and c_2,

$$\left|\frac{\Delta_\epsilon^* - \Delta^*}{\Delta^*}\right| = 1.1 \times 10^{-9}, \quad \left|\frac{\text{rMSE}_\epsilon(\Delta_\epsilon^*) - \text{rMSE}(\Delta^*)}{\text{rMSE}(\Delta^*)}\right| = 0.024.$$

This shows that 10^6-averaged runs of the modified Algorithm 3 with $N = 10^6$

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0210

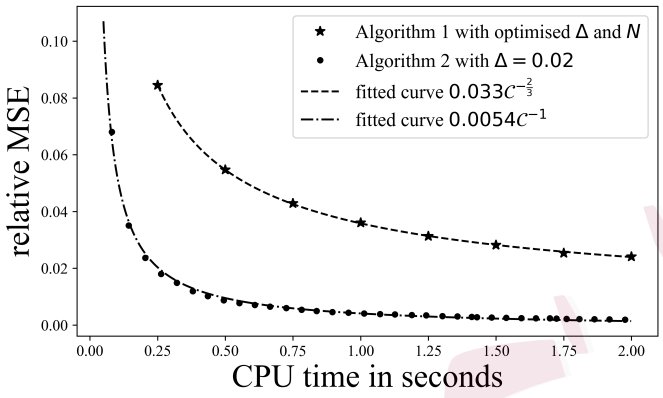


Figure 4: Comparison between the likelihood estimates computed by Algorithm 1 (using (3.5)) and Algorithm 3 (using (4.6)). The true likelihood is approximated using Algorithm 3 with $N = 10^4$ particles and $\Delta = 0.02$. Note that Algorithm 1 uses optimized Δ and N to obtain the best MSE for a given CPU budget. Algorithm 3 uses fixed $\Delta = 0.02$ and the design choice for η described in Section 4.2.1.

particles is more than sufficient to produce an accurate estimate $\mathcal{L}_{\text{MC}}$ as a substitute for $\mathcal{L}$. We use the same number of Monte Carlo repetitions and N for values of $n_p > 2$ up to $n_p = 13$, which are reported in Figure 3b. Figures 3a and 3b validate the expression for the rMSE (5.1) in the locality of the minimum Δ. We continue to use $\mathcal{L}_{\text{MC}}$ to compare Algorithms 1 and 3. We use $\mathcal{L}_{\text{MC}}$ to find the smallest rMSE that Algorithm 1 can achieve for a given CPU budget, and use $\mathcal{L}_{\text{MC}}$ to compute the rMSE of Algorithm 3 for the same CPU budget. For Algorithm 1, for each value of $\mathcal{C}$, we repeat the procedure illustrated in Figure 3 to find Δ that yields the smallest rMSE_ϵ; this Δ is the minimizer of the fitted line, as illustrated in Figure 3b. For Algorithm 3, we spend the budget on increasing the number of particles N, while using a fixed value of $\Delta = 0.02$. The results of this comparison are shown in Figure

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0210

4. It appears that Algorithm 3 achieves the best decay rate of the rMSE within a CPU budget, which is the inverse relationship, whereas Algorithm 1 achieves only a rate of $\mathcal{C}^{-2/3}$.

5.2 Three-Dimensional Single-Molecule Model

In this section, we apply our method to track a moving biological molecule in a live cell, in three dimensions, arising from single molecule fluorescence microscopy. An illustration of how the data are generated is given in Figure 5a. Single molecule fluorescence microscopy is a live cell imaging technique in which molecules of interest are tagged with a fluorophore, which are then excited using light at a particular frequency. These molecules fluoresce under excitation and emit light at different frequencies, which is then captured by a CCD camera after optical magnification. The recorded images are used to uncover their motion. In particular, the moving molecule follows a diffusion model, and its observations are the (random) arrival times and locations of individual photons. The photon emission process is modeled as a Poisson process (Ober et al., 2020), where its photon rate, denoted as $\lambda(t)$, is the rate at which photons are emitted by the object at time t. d'Avigneau et al. (2022) and Vahid et al. (2020) assume a static molecule on the optical axis, where they apply particle filtering to jointly calibrate the model and localize the single molecule. In contrast, we follow the approach of Szalai et al. (2021) to incorporate movement in all three coordinates, where the photon arrival times are governed by

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0210

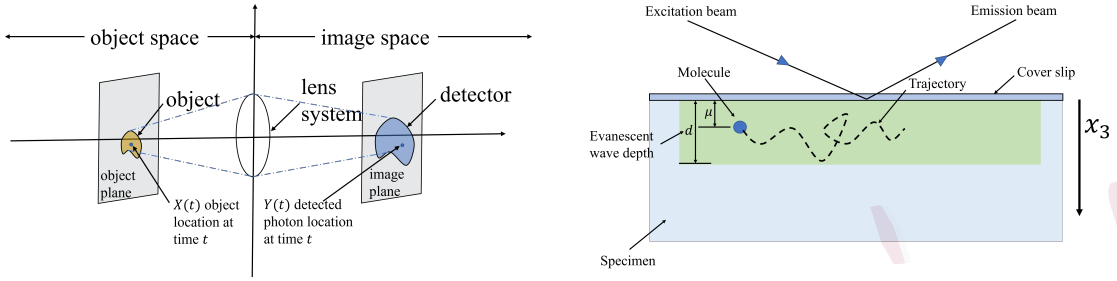
the depth of the molecule (see Figure 5b). The excitation of the molecule varies inversely with the molecule's depth, owing to the attenuation of the excitation light. Szalai et al. (2021) show that the photon rate $\lambda(\cdot)$ decays exponentially along the x_3-axis, $\lambda(x_3) = \lambda_0 \exp(-\frac{x_3}{d})$, where λ_0 denotes the rate of photons emitted by a fluorophore at $x_3 = 0$. Photon arrival locations are imprecise (noise corrupted) observations of the molecule's location in the other two dimensions, as governed by diffraction theory. The relevant photon location model is the three-dimensional BW model for the point spread function, which describes how a point light source appears in an image as it moves in and out of focus (Ober et al., 2020). Hence, the molecule's depth affects the photon arrival rate and arrival locations, the former through a state (depth)-dependent photon detection rate $\lambda(X_t)$, and the latter through the three-dimensional BW model.

We let $(X_t)_{0\le t\le T} := (X_{1,t}, X_{2,t}, X_{3,t})^\top_{0\le t\le T}$ denote the three-dimensional location of the molecule at time t. The three components of the molecule state are its $(x_1, x_2, x_3)^\top$ location, and are assumed to follow the *Ornstein–Uhlenbeck* (OU) model,

$$dX_{i,t} = -\phi_i(X_{i,t} - \mu_i)dt + dW_{i,t}, \qquad \text{for } i = 1, 2, 3,$$

where $\phi_i > 0$ and $(W_{i,t})_{0\le t\le T}$, for $i = 1, 2, 3$, are independent Brownian motions. We assume that the initial distribution that generates X_0 is $\mathcal{N}(\mu, \Sigma_0)$, where the

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0210



(a) (b)

Figure 5: Illustration of (a) how the image is acquired by a microscope, and (b) the total internal reflection fluorescence. An excitation beam with wavelength λ_v traveling from a high refractive index (n_1) medium into a lower refractive index (n_2) medium is totally internal reflected at a planar interface. The reflection generates a thin layer of light in the lower refractive index medium, which has an intensity that decays exponentially along the x_3-axis with a characteristic constant d. While the molecule moves in the field, it is illuminated and thus fluoresces. Parameter μ along the x_3-direction is the mean of $X_{3,t}$, which the molecule diffuses about.

covariance matrix $\Sigma_0 = p_0 \times \mathbb{I}_{3\times 3}$. The transition density $f_\delta^\theta(x'|x)$ of the process is

$$X_{i,t+\delta}|\big(X_{i,t} = x_i\big) \sim \mathcal{N}\big(\mu_i + e^{-\delta\phi_i}(x_i - \mu_i), \frac{1}{2\phi_i}(1 - e^{-2\delta\phi_i})\big), \quad i = 1, 2, 3. \tag{5.3}$$

For an object located at $(x_1, x_2, x_3)^{\mathrm{T}} \in \mathbb{R}^3$ in the object space (prior to magnification), the location (on the detector) at which a photon is detected is specified probabilistically using a two-dimensional probability density function as

$$g^\theta(y|x) := \frac{1}{|M|} q_{x_3}\big(M^{-1}y - (x_1, x_2)^{\intercal}\big), \qquad y \in \mathbb{R}^2, \tag{5.4}$$

where $M \in \mathbb{R}^{2\times 2}$ is an invertible lateral magnification matrix, and the image function $q_{x_3} : \mathbb{R}^2 \to \mathbb{R}$ describes the image of an object in the detector space when that object is located at $(0, 0, x_3)$ in the object space, where $x_3 \in \mathbb{R}$ is the location of the object on the optical axis. This three-dimensional BW model is the resulting

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0210

image function, derived from diffraction theory, for a point source that can also be out of focus (Born and Wolf, 2013). For $(x_1, x_2) \in \mathbb{R}^2$,

$$q_{x_3}(x_1, x_2) = \frac{4\pi n_\alpha^2}{\lambda_e^2} \left| \int_0^1 J_0\left(\frac{2\pi n_\alpha}{\lambda_e}\sqrt{x_1^2 + x_2^2}\rho\right) \exp\left(\frac{j\pi n_\alpha^2 x_3}{n_0 \lambda_e}\rho^2\right)\rho d\rho \right|^2, \tag{5.5}$$

where n_0 is the refractive index of the objective lens immersion medium, and n_α is the numerical aperture of the objective lens. In addition, λ_e is the emission wavelength of the molecule, and $J_0(\cdot)$ and $J_1(\cdot)$ represent the zero-th-order and the first-order Bessel function, respectively, of the first kind. The probability density functions of the BW model at different defocus levels are plotted in **S9** of the Supplementary Material. Note that a large defocus tends to produce images of poor quality, making it more difficult to estimate the molecule's position.

Under this setup, instead of considering the molecular movement as a reflected diffusion process, we adopt a simpler approach by assuming a standard OU process, which is suitable if the molecule does not hit a boundary (i.e., either cover slip $x_3 = 0$ or its maximum depth d) over its observation period, for example, if the observation period is short and/or the molecule is diffusing about a mean depth μ in the middle of a cell with large ϕ_3 (i.e., stronger attraction to μ); see Figure 5b.

In Figure 6a, we plot the true trajectory of the molecule for the numerical studies, which is generated by the SDE in (5.3) with the parameters $\{\phi = (\phi_1, \phi_2, \phi_3)^\top = (1, 1, 4)^\top, \mu = (\mu_1, \mu_2, \mu_3)^\top = (0, 0, 2)^\top, p_0 = 1/(2\phi)\}$ for the time interval [0, 5.0]. We used a thinning algorithm (detailed in **S10** in the Supplemen-

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0210

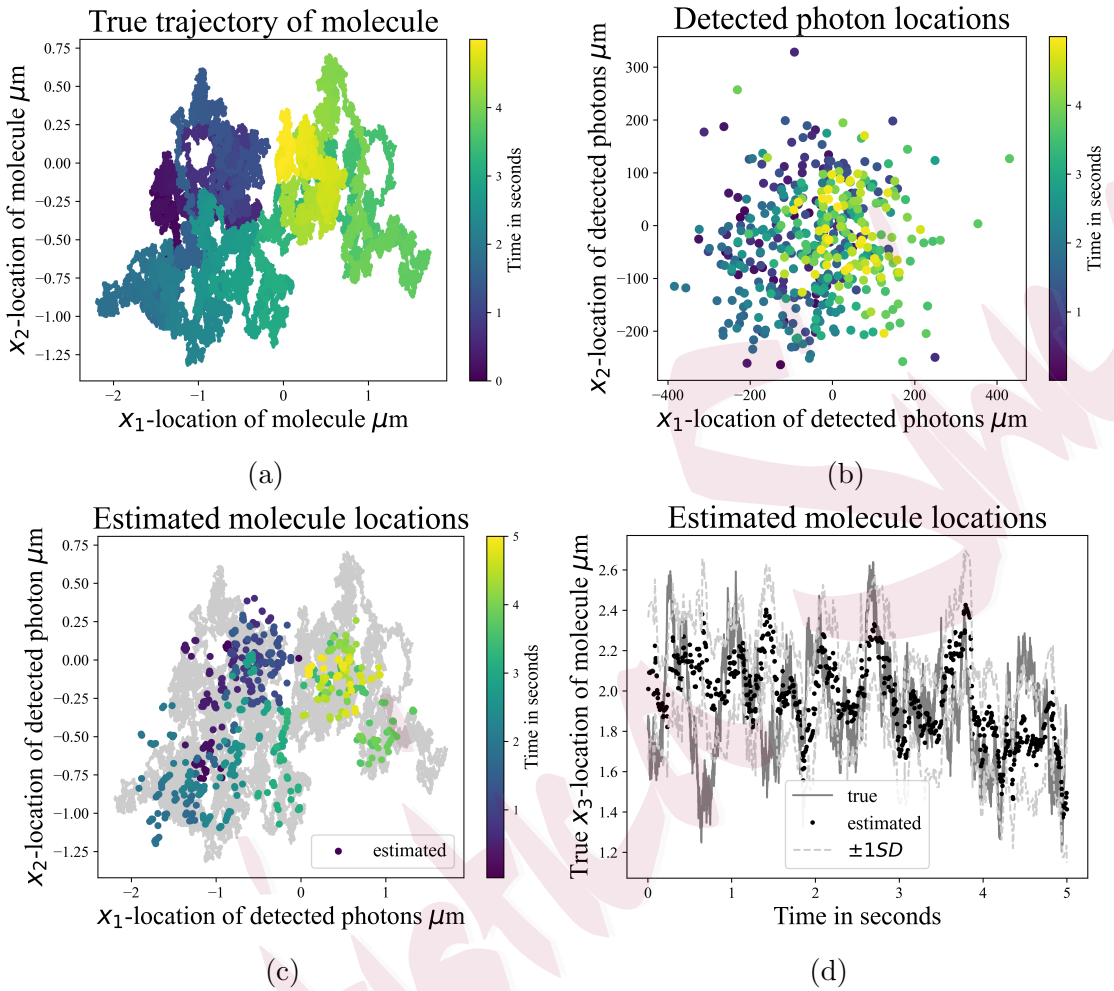


Figure 6: (a) True trajectory of a molecule; (b) observed photon locations; (c) estimated (x_1, x_2) molecule locations; and (d) true x_3 molecule locations and estimated location.

tary Material) to generate the observation times by using the intensity function (see (5.2)) with parameters $\{\lambda_0 = 100, d = 20\mu m\}$. Given these observation times, we generate the observed photon locations using the photon distribution profile given by (5.4) and (5.5). The associated parameters are $\{M = m\mathbb{I}_{2\times 2}, m = 100, n_\alpha = 1.4, \lambda_e = 0.52\mu\text{m}, n_0 = 1.515, \sigma_a^2 = 49 \times 10^{-4}\mu\text{m}^2\}$, and the corresponding data set

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0210

of photon locations is shown in Figure 6b. The colors in Figure 6a indicate time, becoming lighter as more time passes. Figure 6c shows the mean of the estimated (X_1, X_2) locations of the molecule found using Algorithm 3, which does track the true trajectory. Figure 6d shows the true X_3 position of the molecule, the mean of the estimated X_3 positions (also obtained using Algorithm 3), and the standard deviations. During periods with no observations, the estimated X_3-value is larger, as expected, because this corresponds to a smaller photon arrival intensity function. Figure 6d also shows that large X_3-values degrade the estimation quality (which is more clearly seen for the X_3-values), owing the BW observation model for an out-of-focus molecule; see (5.5). Additional results of this phenomenon are reported in **S11** of the Supplementary Material.

Figure 7 shows a comparison between the estimation quality of Algorithms 1 (with $\Delta = 0.03, 0.003$ and 0.0003) and 3 (with $\Delta = 0.1$) for this single-molecule example. For both methods, the CPU time increases with the number of particles used in the algorithms. The superiority of Algorithm 3 is apparent, as measured using rMSE of the likelihood estimate. The best Δ for Algorithm 1 is not necessarily the smallest one for a fixed CPU budget. This also has practical consequences. High-frequency data have potentially more time intervals between observation arrivals that are much smaller than Δ. This leads to a small bias for Algorithm 1, although at a higher computational cost. Further reducing the bias, the rMSE is dominated by the variance if the CPU budget permits only a smaller number of

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0210

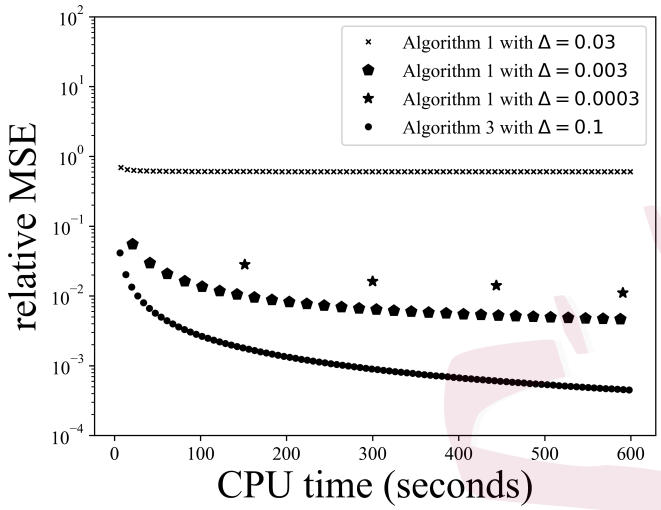


Figure 7: Plot of rMSE_ϵ of likelihood estimates versus CPU time for Algorithm 3 with $\Delta = 0.1$ and Algorithm 1 with $\Delta = 0.03, 0.003$ and 0.0003. rMSE of likelihood estimates based on the average estimate of $\mathcal{L}$ given by 10^6 runs of Algorithm 3 with each run using $N = 10^6$ particles, and the design choice for l and η given in Section 4.2.1. Here 600 CPU seconds corresponds to 2×10^5 particles for Algorithm 3.

particles.

5.2.1 Model calibration using a particle MCMC

Estimating the parameters of the molecular dynamics is also important in single molecule studies. For example, d'Avigneau et al. (2022) calibrate the model using a maximum likelihood estimation after discretizing the path integral. In contrast, we use the PMMH algorithm Andrieu et al. (2010) to sample from the posterior density $p(\theta, x_{0:m}|y_{0:m})$, where $\theta = (\phi, \mu)$. Data are simulated from the model $\{\phi = (1,1,4)^\top, \mu = (0,0,2)^\top, p_0 = (0.5, 0.5, 0.125)^\top\}$, with the intensity function parametrized by $\{\lambda_0 = 25, d = 20\}$. The parameters of the BW model remain the

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0210

same as before. A precise estimation of ϕ_3 requires a longer time series, because it is weakly identified; we use 350 observations collected in the time interval $[0, 15s]$. The following independent priors are used: $\phi_3 \sim \mathcal{U}(0, 10)$ and $\mu_3 \sim \mathcal{U}(0, 10)$. We use a normal random-walk Metropolis–Hastings proposal with initial covariance $0.1 \times \mathbb{I}_{2\times 2}$ to update the parameters jointly. We use the continuous covariance adaptation scheme of Haario et al. (2001) in the PMMH algorithm. We chose the following three experimental settings:

• Experiment 1 (low CPU budget): $\mathcal{C} = 1.5s$, allowing a coarse time discretization that coincides with the times of arrivals of the data for both Algorithms 1 and 3. This forces a large Δ in $\mathcal{L}_\Delta$ for Algorithm 1. We adjust N so that the CPU budget is the same for both algorithms.

• Experiment 2 (larger CPU budget): $\mathcal{C} = 2.5s$ permits a finer time discretization than the observation arrival times. The best Δ and N (within the CPU budget) are chosen for Algorithm 1 using the procedure outlined in Section 5.1. Using a larger CPU budget allows a smaller Δ than that in Experiment 1 in $\mathcal{L}_\Delta$. For Algorithm 3, we use $\Delta = 0.01$ and its cost-adjusted N.

• Experiment 3 (effective sample size-based comparison): $\mathcal{C} = 2.0s$; here, we find the effective sample size (ESS) for the PMMH using Algorithm 3 with $\Delta = 0.01$, and then choose the best Δ and N for Algorithm 1, while ensuring its ESS matches that of Algorithm 3. The ESS, which measures the number of "independent samples", is $M/\left(-1 + 2\sum_{t=0}^{K}(\rho_{2t} + \rho_{2t+1})\right)$, where ρ_t is the estimated

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0210

autocorrelation at lag n, and K is the last integer for which the sum in the sum bracket is still positive. The general trend is that the ESS of the PMMH with Algorithm 1 increases with Δ (i.e., larger N for the fixed CPU budget), although the estimation is more biased for Algorithm 1.

We ran the algorithms 10^5 times with 10^4 burn-in iterations. Figure 8 displays the estimates of the marginal posterior densities for μ_3 and ϕ_3 for all three experiments. Experiment 1 shows that Algorithm 3 effectively removes the bias in the estimation of the parameters, whereas the performance of Algorithm 1 is compromised by the limited CPU budget. Experiment 2 shows that, given a sufficient budget, Algorithm 1 achieves almost the same performance as that of Algorithm 3. A comparison between experiments 1 and 2 shows that the unbiased posterior produced by Algorithm 3 is unaffected by the CPU budget. Experiment 3 is carried out under the setting that both algorithms produce equally "efficient" MCMC samples for a fixed CPU budget. The results show that Algorithm 1 still yields some bias in the estimation.

6. Conclusion

We have examined smoothing and model calibration for a partially observed diffusion with a Cox process observation model. We estimate the intractable likelihood using the positive part of a Poisson estimate, for the path integrals within, embedded within particle filtering. The probability of encountering a negative Poisson

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0210

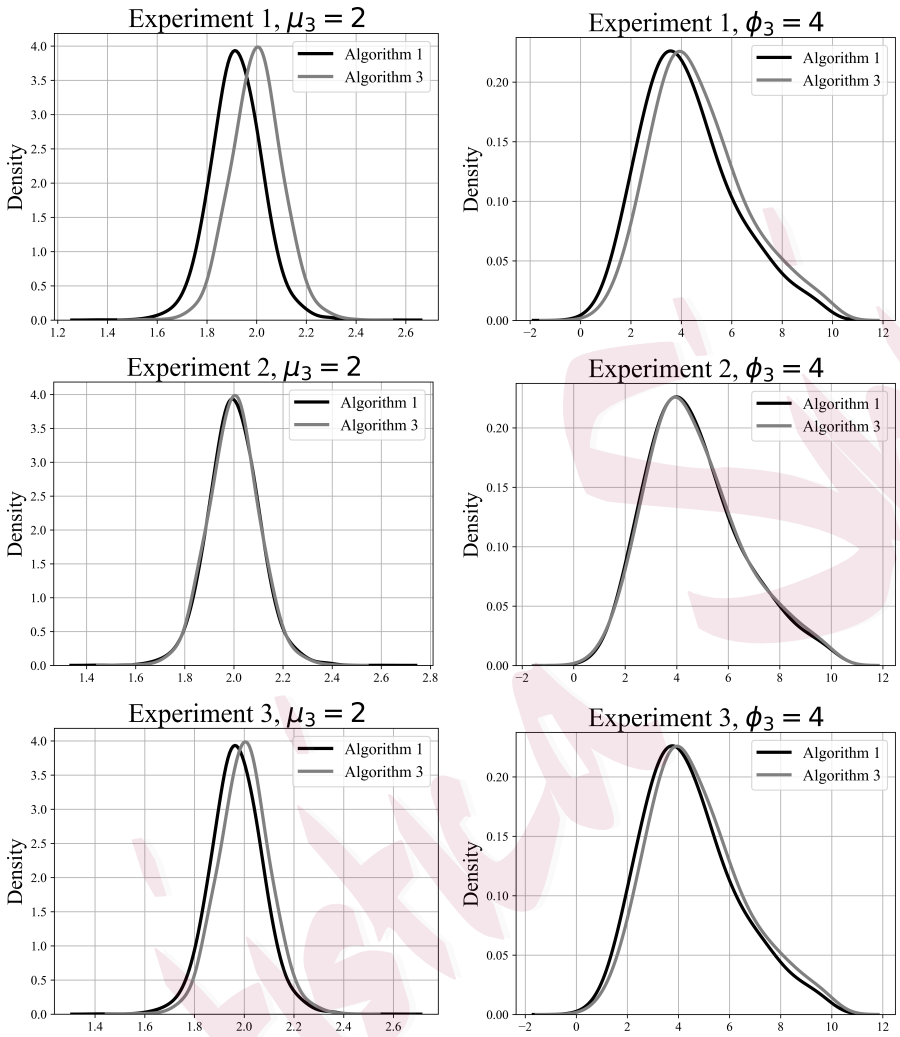


Figure 8: PMMH experiments: posterior densities $p(\mu_3|y_{t_1:t_{np}})$ and $p(\phi_3|y_{t_1:t_{np}})$.

estimate in one complete particle filtering pass through the data is strongly controlled by adjusting $\eta = \mathcal{O}(\Delta l)$. As such, because of the rarity of the occurrence of a negative estimate, which triggers the particle weight truncation, we effectively remove the time discretization error that biases conventional particle implementations, such as that of d'Avigneau et al. (2022).

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0210

Our numerical results show that the proposed particle method (Algorithm 3) outperforms the conventional (discretization-based) particle filter in terms of the rMSE, where ours decays with order $\mathcal{O}(\mathcal{C}^{-1})$ compared to $\mathcal{O}(\mathcal{C}^{-\frac{2}{3}})$, where $\mathcal{C}$ is the computational budget. We then applied our particle filter to a challenging three-dimensional single molecule microscopy example to both estimate the trajectory of the moving molecule and calibrate the model. We clearly show the bias in the posterior distribution for the model parameters computed using a conventional implementation, such as that of d'Avigneau et al. (2022), whereas using the proposed method, the bias is not discernible. Although the bias in the conventional method can be reduced by employing a smaller Δ, the time discretization interval, this not only requires significant additional CPU time, but also prohibits the application of backward sampling steps in particle filtering. This is a possible direction for future work, that is, defining a forward filtering backward sampling implementation of our method. In the context of diffusions, this is a challenging problem; see Yonekura and Beskos (2022) for a recent study.

7. Supplementary Material

Section S1 contains the derivation of bridge density for linear Gaussian diffusion. Section S2 provides the proof of Lemma 1. Section S3 presents the derivation of expectation of the unqualified probability bound. Section S4 presents the proof of Lemma 2. Section S5 provides the experiments for the use of Wald's identity. Sec-

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0210

tion S6 shows the exact computation of the likelihood function. Section S7 includes no observation case and two observations case. Sections S8 provides the empirical relationship between relative variance and Δ. Section S9 provides the BW model and its point spread function. Section S10 describes the thinning algorithm for creating single-molecule data. Section 11 provides the results of additional single-molecule experiments.

Acknowledgments

We would like to thank Dr. Alix Marie d'Avigneau for sharing with us the python code used to generate the single-molecule data.

References

Andrieu, C., A. Doucet, and R. Holenstein (2010). Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B 72*(3), 269–342.

Beskos, A., O. Papaspiliopoulos, G. O. Roberts, and P. Fearnhead (2006). Exact and computationally efficient likelihood-based estimation for discretely observed diffusion processes. *Journal of the Royal Statistical Society: Series B 68*(3), 333–382.

Born, M. and E. Wolf (2013). *Principles of optics: electromagnetic theory of propagation, interference and diffraction of light*. Elsevier.

Chopin, N., S. S. Singh, T. Soto, and M. Vihola (2022). On resampling schemes for particle filters with weakly informative observations. *The Annals of Statistics 50*(6), 3197–3222.

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0210

REFERENCES

d'Avigneau, A. M., S. S. Singh, and R. J. Ober (2022). Limits of accuracy for parameter estimation and localization in single-molecule microscopy via sequential Monte Carlo methods. *SIAM Journal on Imaging Sciences 15*(1), 139–171.

Del Moral, P. (2004). *Feynman-Kac formulae: genealogical and interacting particle systems with applications*, Volume 88. Springer.

Durham, G. B. and A. R. Gallant (2002). Numerical techniques for maximum likelihood estimation of continuous-time diffusion processes. *Journal of Business & Economic Statistics 20*(3), 297–338.

Fearnhead, P., O. Papaspiliopoulos, and G. O. Roberts (2008). Particle filters for partially observed diffusions. *Journal of the Royal Statistical Society: Series B 70*(4), 755–777.

Fearnhead, P., O. Papaspiliopoulos, G. O. Roberts, and A. Stuart (2010). Random-weight particle filtering of continuous time processes. *Journal of the Royal Statistical Society: Series B 72*(4), 497–512.

Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *The journal of physical chemistry 81*(25), 2340–2361.

Haario, H., E. Saksman, and J. Tamminen (2001). An adaptive Metropolis algorithm. *Bernoulli*, 223–242.

Kloeden, P. and E. Platen (2011). *Numerical Solution of Stochastic Differential Equations*. Stochastic Modelling and Applied Probability. Springer Berlin Heidelberg.

McAdams, H. H. and A. Arkin (1997). Stochastic mechanisms in gene expression. *Proceedings of the National Academy of Sciences 94*(3), 814–819.

Merton, R. C. (1975). Optimum consumption and portfolio rules in a continuous-time model. In *Stochastic*

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0210

optimization models in finance, pp. 621–661. Elsevier.

Nicolau, J. (2002). A new technique for simulating the likelihood of stochastic differential equations. *The Econometrics Journal 5*(1), 91–103.

Ober, R. J., E. S. Ward, and J. Chao (2020). *Quantitative Bioimaging: An Introduction to Biology, Instrumentation, Experiments, and Data Analysis for Scientists and Engineers*. CRC Press.

Obukhov, A. M. (1959). Description of turbulence in terms of lagrangian variables. *Advances in Geophysics 6*, 113–116.

Papaspiliopoulos, O. (2011). *Monte Carlo probabilistic inference for diffusion processes: a methodological framework*, pp. 82–103. Cambridge University Press.

Rousset, M. and A. Doucet (2006). Discussion on 'exact and computationally efficient likelihood-based inference for discretely observed diffusion processes'. *Journal of the Royal Statistical Society: Series B 68*(3), 375–376.

Snyder, D. L. and M. I. Miller (2012). *Random point processes in time and space*. Springer Science & Business Media.

Szalai, A. M., B. Siarry, et al. (2021). Three-dimensional total-internal reflection fluorescence nanoscopy with nanometric axial resolution by photometric localization of single molecules. *Nature communications 12*(1), 1–13.

Vahid, M. R., B. Hanzon, and R. J. Ober (2020). Fisher information matrix for single molecules with stochastic trajectories. *SIAM Journal on Imaging Sciences 13*(1), 234–264.

Wagner, W. (1988). Unbiased multi-step estimators for the Monte Carlo evaluation of certain functional

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0210

integrals. *Journal of computational physics* *79*(2), 336–352.

Yonekura, S. and A. Beskos (2022). Online smoothing for diffusion processes observed with noise. *Journal of Computational and Graphical Statistics* *31*(4), 1344–1360.

Department of Engineering, University of Cambridge

E-mail: rj420@cam.ac.uk

NIASRA, School of Mathematics and Applied Statistics, University of Wollongong

E-mail: sumeetpals@uow.edu.au

ENSAE, Institut Polytechnique de Paris

E-mail: nicolas.chopin@ensae.fr

Statistica Sinica