SCORE-BASED PARAMETER ESTIMATION FOR A CLASS OF CONTINUOUS-TIME STATE SPACE MODELS

ALEXANDROS BESKOS*, DAN CRISAN†, AJAY JASRA‡, NIKOLAS KANTAS†, AND HAMZA RUZAYQAT‡

Abstract. We consider the problem of parameter estimation for a class of continuous-time state space models (SSMs). In particular, we explore the case of a partially observed diffusion, with data also arriving according to a diffusion process. Based upon a standard identity of the score function, we consider two particle filter based methodologies to estimate the score function. Both methods rely on an online estimation algorithm for the score function, as described, e.g., in [13], of $\mathcal{O}(N^2)$ cost, with $N \in \mathbb{N}$ the number of particles. The first approach employs a simple Euler discretization and standard particle smoothers and is of cost $\mathcal{O}(N^2 + N\Delta_l^{-1})$ per unit time, where $\Delta_l = 2^{-l}$, $l \in \mathbb{N}_0$, is the time-discretization step. The second approach is new and based upon a novel diffusion bridge construction. It yields a new backward type Feynman-Kac formula in continuous-time for the score function and is presented along with a particle method for its approximation. Considering a time-discretization, the cost is $\mathcal{O}(N^2\Delta_l^{-1})$ per unit time. To improve computational costs, we then consider multilevel methodologies for the score function. We illustrate our parameter estimation method via stochastic gradient approaches in several numerical examples.

Key words. Score Function, Parameter Estimation, Particle Filter, Diffusion Bridges.

AMS subject classifications. 65C05, 65C35, 60G35, 60J60, 60J65, 60H10, 60H35, 91G60

1. Introduction.

We consider the problem of parameter estimation for continuous-time SSMs. These are models comprising stochastic differential equations (SDEs) describing a hidden dynamic state and their observations. Such models are ubiquitous in a large number of practical applications in science, engineering, finance, and economics, see [11] for an overview. Inference in SSMs, also known as hidden Markov models, hinges upon computing conditional probability distributions of the dynamic hidden state given the acquired observations and unknown static parameters. This is referred to as the stochastic filtering problem, which is in general intractable, but reliable numerical approximations are routinely available [1, 11]. The problem of inferring the unknown parameters is more challenging. In this paper we focus on maximum likelihood inference and gradient methods that are performed in an online manner. The offline or iterative case given a fixed batch of observations can also be treated using our proposed methods. The approach considered in this article is to make use of the gradient of the log-likelihood, commonly referred to as the *score function*, within a stochastic gradient algorithm (see e.g. [13, 14]). Intrinsically, there are several challenges arising with such an approach. Firstly, when one adopts a continuous-time model and assumes access to arbitrarily high frequency observations, one does not observe in practice truly in continuous-time, therefore some sort of time-discretization is required. Secondly, in both discrete-time and continuous-time formulations, there are very few cases when the score function is analytically available. Both these issues imply that numerical approximations are required.

We consider two different approaches for the numerical approximation of the score function. The first is to simply time-discretize a representation of the score function and then apply discrete-time numerical approximation schemes [13, 14, 23]. The second, is to develop a numerical approx-

*Department of Statistical Science, University College London, London, WC1E 6BT, UK. (a.beskos@ucl.ac.uk).

†Department of Mathematics, Imperial College London, London, SW7 2AZ, UK. (d.crisan@imperial.ac.uk, n.kantas@imperial.ac.uk).

‡Computer, Electrical and Mathematical Sciences and Engineering Division, King Abdullah University of Science and Technology, Thuwal, 23955, KSA. (ajay.jasra@kaust.edu.sa, hamza.ruzayqat@kaust.edu.sa).

imation scheme directly on continuous-time path-space to estimate the score function and then (necessarily) discretize the algorithm in time. The order of designing the estimation method and time-discretization can be rather important. Often the second approach is preferable in terms of both performance and robustness as the discretization mesh vanishes, see e.g. [25, 24]. We then use the score estimate for implementing recursive maximum likelihood, where the parameters are updated at unit time intervals ([21, 27]). The particular choice of time interval length is without loss of generality, and allows the score to accumulate sufficient information from the observations before updating the parameters.

In the first approach, we consider a well-known expression for the score function, for instance as given in [8]. Given this formula, one can produce an Euler discretized version of the score and work in discrete-time but with high frequency. The score is an expectation of an additive functional of the hidden state path conditioned on the available observations, which is commonly referred to as the smoothing distribution. In this context, many well-known particle smoothing schemes can now be adopted, such as the ones described in [13, 23]. These latter approaches are simulation-based schemes whose convergence is based upon the number of samples $N \in \mathbb{N}$. We prove some technical results for the discretized problem, which together with the work in [13] allow us conjecture that to obtain a mean square error (MSE) of $\mathcal{O}(\epsilon^2)$, for given $\epsilon > 0$, we require a computational cost of $\mathcal{O}(\epsilon^{-4})$ per unit time. The latter derives from an algorithmic cost of $\mathcal{O}(N^2 + N\Delta_l^{-1})$ per unit time, where $\Delta_l = 2^{-l}$ is the time-discretization. As we explain later in the article, this corresponds to a best case scenario, due to the intrinsic nature of the algorithm. In particular, we start with a continuous-time formula and time-discretize it, but the deduced numerical algorithm can be problematic in terms of computational complexity as $l \to \infty$. Whilst in some scenarios one does not observe any issues, examples can be found where the variance of the method can explode as l grows ([36]), thus putting into question the validity of the conjecture on the cost to achieve an MSE of $\mathcal{O}(\epsilon^2)$.

This motivates the introduction of our second approach, where we build upon a change of measure technique proposed in [29]. This latter approach has so far been used in very different contexts than the present paper, namely related to discretely observed diffusions for Bayesian inference and Markov chain Monte Carlo (MCMC) [34, 35] or smoothing for potentially non-linear observation functions [22]. Our approach is a data augmentation scheme, whereby, at unit time intervals, the end points for the hidden state are sampled and the path is connected using diffusion bridges. Then, starting again from the formula for the score function in [8], we will use this change of measure associated to a diffusion bridge and its driving Brownian noise; see also [36] where a related approach is used for a different class of problems. Based upon this change of measure we develop a new backward type Feynman-Kac formula in continuous-time. This new formula facilitates an adaptation of the method in [13] in true continuous-time, albeit one cannot apply it in practice. We time-discretize the algorithm and conjecture that to obtain a MSE of $\mathcal{O}(\epsilon^2)$, for given $\epsilon > 0$, we need a cost of $\mathcal{O}(\epsilon^{-6})$ per unit time, which derives from an algorithmic cost of $\mathcal{O}(N^2\Delta_l^{-1})$ per unit time. We note however, that this computational complexity will not explode with increasing l as may be the case in the first approach. To improve the cost required for a given MSE, we develop a novel multilevel Monte Carlo extension that can, in some cases, achieve a MSE of $\mathcal{O}(\epsilon^2)$ at a cost per unit time of $\mathcal{O}(\epsilon^{-4})$. We remark that although our MSE-cost statements are based upon conjectures, they are verified numerically. Direct proofs of these require substantial technical results that will be the topic of future work.

1.1. Contributions and Organization. We conclude this introduction by emphasizing that our contributions are aimed to deal with both continuous-time observations and hidden states.

As mentioned earlier this poses very particular challenges relative to earlier works that deal with filtering and smoothing when discrete-time observations/models are used as in [7, 15, 16, 22, 35, 28]. None of these works look at continuous-time observations. Similarly online likelihood estimation of the parameters using the score function is considered in [15, 16] only for the case of discrete-time observations of hidden diffusions.

The contributions of this paper can be summarized as follows:

- We investigate the efficiency and accuracy of two fundamentally different numerical approximations of the score function on its own and when used for the purpose of recursive maximum likelihood. Both methods rely on fairly standard tools such as changes of measure, particle smoothing and Euler time-discretization.
- We provide a detailed discussion on the computational complexity of each method. We illustrate the performance and computational cost for several models in numerical examples that consider estimation of the score function and parameter estimation.
- The second approach is a novel method that operates directly on the path-space. The approach improves performance and is robust to arbitrarily small time-discretization at the expense of additional computational cost. The latter is reduced using a new Multilevel Particle Filter; see [17, 20] for some existing approaches.

This article is structured as follows. In Section 2 the basic problem is formulated in continuous-time. In Section 3 we consider our first method for online score estimation and explain the various features associated to it. In Section 4 our second method for online score estimation is developed. In Section 5 our numerical results are presented.

1.2. Notation. Let $(\mathsf{X},\mathcal{X})$ be a measurable space. We write $\mathcal{B}_b(\mathsf{X})$ for the set of bounded measurable functions, $\varphi:\mathsf{X}\rightarrow\mathbb{R}^d$, $d\in\mathbb{N}$, and $\mathcal{C}(\mathsf{X})$ for the continuous ones. Let $\varphi:\mathbb{R}^d\rightarrow\mathbb{R}$; $\text{Lip}_{\|\cdot\|_2}(\mathbb{R}^d)$ denotes the collection of real-valued functions that are Lipschitz w.r.t. the Euclidean distance $\|\cdot\|_2$. That is, $\varphi\in\text{Lip}_{\|\cdot\|_2}(\mathbb{R}^d)$ if there exists a $C<+\infty$ such that for any $(x,y)\in\mathbb{R}^{2d}$,

$$|\varphi(x)-\varphi(y)|\leq C\|x-y\|_2.$$

$\mathcal{N}_s(\mu,\Sigma)$ denotes an s-dimensional Gaussian law of mean μ and covariance Σ; if $s=1$ we omit subscript s. For a vector/matrix X, X^* denotes the transpose of X. For $A\in\mathcal{X}$, $\delta_A(du)$ denotes the Dirac measure of A, and if $A=\{x\}$ with $x\in\mathsf{X}$, we write $\delta_x(du)$. For a vector-valued function in d-dimensions $\varphi(x)$ (resp. d-dimensional vector x) we write the i^{th}-component, $1\leq i\leq d$, as $\varphi^{(i)}(x)$ (resp. x^i). For a $d\times q$ matrix x we write the $(i,j)^{\text{th}}$-entry as $x^{(ij)}$. $\mathbb{N}=\{1,2,\dots\}$ and $\mathbb{N}_0=\mathbb{N}\cup\{0\}$.

2. Problem Formulation.

2.1. Preliminaries. We consider the parameter space $\theta\in\Theta\subset\mathbb{R}^{d_\theta}$, with Θ being compact, $d_\theta\in\mathbb{N}$. The stochastic processes $\{Y_t\}_{t\geq 0}$, $\{X_t\}_{t\geq 0}$ of interest are defined upon the probability triple $(\Omega,\mathcal{F},\mathbb{P}_\theta)$, with $Y_t\in\mathbb{R}^{d_y}$, $X_t\in\mathbb{R}^{d_x}$, $d_y,d_x\in\mathbb{N}$, initial conditions $X_0=x_*\in\mathbb{R}^{d_x}$, $Y_0=y_*\in\mathbb{R}^{d_y}$, and are determined as the solution of the system of SDEs:

$$dY_t=h_\theta(X_t)dt+dB_t; \tag{2.1}$$

$$dX_t=b_\theta(X_t)dt+\sigma(X_t)dW_t. \tag{2.2}$$

Here, for each $\theta\in\Theta$, $h_\theta:\mathbb{R}^{d_x}\rightarrow\mathbb{R}^{d_y}$, $b_\theta:\mathbb{R}^{d_x}\rightarrow\mathbb{R}^{d_x}$, $\sigma:\mathbb{R}^{d_x}\rightarrow\mathbb{R}^{d_x\times d_x}$, with σ being of full rank, and $\{B_t\}_{t\geq 0},\{W_t\}_{t\geq 0}$ are independent standard Brownian motions of dimension d_y, d_x respectively.

To minimize technical difficulties, the following assumptions are made throughout the paper:

(D1) (i) σ is continuous, bounded; $a(x) := \sigma(x)\sigma(x)^*$ is uniformly elliptic;

(ii) for each θ, h_θ and b_θ are bounded, measurable; $h_\theta^{(i)} \in \text{Lip}_{\|\cdot\|_2}(\mathbb{R}^{d_x})$, $1 \le i \le d_y$;

(iii) the gradients $\nabla_\theta h_\theta : \mathbb{R}^{d_x} \to \mathbb{R}^{d_y \times d_\theta}$ and $\nabla_\theta b_\theta : \mathbb{R}^{d_x} \to \mathbb{R}^{d_x \times d_\theta}$ exist, and are continuous, bounded, measurable; $\nabla_\theta h_\theta^{(ij)} \in \text{Lip}_{\|\cdot\|_2}(\mathbb{R}^{d_x})$, $1 \le i \le d_y$, $1 \le j \le d_\theta$;

(iv) let $\phi_\theta(x) = (\nabla_\theta b_\theta(x))^* \, a(x)^{-1}\sigma(x)$; for any θ, $\phi_\theta^{(ij)} \in \text{Lip}_{\|\cdot\|_2}(\mathbb{R}^{d_x})$, $1 \le i \le d_\theta$, $1 \le j \le d_x$.

We introduce the probability measure $\overline{\mathbb{P}}_\theta$, defined via the Radon-Nikodym derivative:

$$Z_{t,\theta} := \frac{d\mathbb{P}_\theta}{d\overline{\mathbb{P}}_\theta}\Big|_{\mathcal{F}_t} = \exp\Big\{ \int_0^t h_\theta(X_s)^* dY_s - \tfrac{1}{2}\int_0^t h_\theta(X_s)^* h_\theta(X_s) ds \Big\}, \tag{2.3}$$

with $\mathcal{F}_t = \sigma(\{X_s, Y_s\}_{0\le s \le t})$. Henceforth, $\overline{\mathbb{E}}_\theta$ denotes expectation w.r.t. $\overline{\mathbb{P}}_\theta$, so that under $\overline{\mathbb{P}}_\theta$, the process $\{X_t\}_{t\ge 0}$ follows the dynamics in (2.2), whereas $\{Y_t\}_{t\ge 0}$ is a Brownian motion independent of $\{X_t\}_{t\ge 0}$. We define, for $\varphi \in \mathcal{B}_b(\mathbb{R}^{d_x})$:

$$\gamma_{t,\theta}(\varphi) := \overline{\mathbb{E}}_\theta \left[\varphi(X_t) Z_{t,\theta} \,\middle|\, \mathcal{Y}_t \right],$$

where $\mathcal{Y}_t$ is the filtration generated by the process $\{Y_s\}_{0\le s \le t}$. Our objective is to produce estimates of the gradient of the score function $\nabla_\theta \log(\gamma_{T,\theta}(1))$.

Remark 2.1. *To connect the changes of measures with standard likelihood derivations, notice that – via Girsanov's theorem – $Z_{t,\theta}$ is the density (w.r.t. to a Wiener measure) of the distribution of $\{Y_s\}_{0\le s\le t}$ conditionally on $\{X_s\}_{0\le s\le t}$. Then, $\gamma_{T,\theta}(1)$ integrates out $\{X_t\}_{0\le t\le T}$, thus corresponds to the marginal density – i.e., the likelihood – of the observations $\{Y_t\}_{0\le t\le T}$.*

In our setting, the score function writes as (see e.g. [8]):

$$\nabla_\theta \log(\gamma_{T,\theta}(1)) = \frac{\overline{\mathbb{E}}_\theta \left[\lambda_{T,\theta} Z_{T,\theta} \,|\, \mathcal{Y}_T \right]}{\overline{\mathbb{E}}_\theta \left[Z_{T,\theta} \,|\, \mathcal{Y}_T \right]}, \tag{2.4}$$

where we have defined:

$$\lambda_{T,\theta} := \int_0^T (\nabla_\theta b_\theta(X_t))^* a(X_t)^{-1}\sigma(X_t) dW_t + \int_0^T (\nabla_\theta h_\theta(X_t))^* dY_t - \int_0^T (\nabla_\theta h_\theta(X_t))^* h_\theta(X_t) dt.$$

For completeness, a derivation of (2.4) can be found in [5, Appendix] and the supplementary material of this article. We remark that one can derive a formula for the score function when σ depends upon θ, which is given in Section 4. We will assume throughout that $T \in \mathbb{N}$. Note also that an application of Bayes' rule gives that, almost surely:

$$\frac{\overline{\mathbb{E}}_\theta \left[\lambda_{T,\theta} Z_{T,\theta} \,|\, \mathcal{Y}_T \right]}{\overline{\mathbb{E}}_\theta \left[Z_{T,\theta} \,|\, \mathcal{Y}_T \right]} = \mathbb{E}_\theta \left[\lambda_{T,\theta} \,|\, \mathcal{Y}_T \right], \tag{2.5}$$

where $\mathbb{E}_\theta$ denotes expectation w.r.t. $\mathbb{P}_\theta$.

2.2. Parameter Estimation. In the offline case suppose one has obtained data $\{Y_t\}_{0\leq t\leq T}$. Then it is possible to perform standard gradient descent using (2.5) and updating θ iteratively:

$$\theta^{m+1} = \theta^m + \alpha_m \, \mathbb{E}_{\theta^m}\left[\, \lambda_{T,\theta^m} \,|\, \mathcal{Y}_T \,\right], \tag{2.6}$$

where $\alpha_m \in \mathbb{R}^+$, $m \in \mathbb{N}_0$, are decreasing step-sizes. Instead here we will mainly focus on an online gradient estimation procedure. To obtain this one can aim to maximise the following limiting *average log-likelihood*,

$$\mathcal{L}(\theta) = \lim_{t\to\infty} \frac{1}{t}\int_0^t \log \gamma_{s,\theta}(1) ds.$$

Let the filter be denoted as $\pi_{t,\theta}(\varphi) = \mathbb{E}_\theta[\varphi(X_t)|\mathcal{Y}_s]$ and using standard arguments (e.g. Lemma 3.29 p. 67 [1]) one can re-write $\gamma_{t,\theta}(1)$ as

$$\log \gamma_{t,\theta}(1) = \int_0^t \pi_{s,\theta}(h)^T dY_s - \frac{1}{2}\int_0^t \pi_{s,\theta}(h)^T \pi_{s,\theta}(h) ds.$$

Under appropriate stability and regularity conditions for both Y_t and $\pi_{t,\theta}$ (see [31] for more details), then both $\mathcal{L}(\theta)$ and $\nabla_\theta \mathcal{L}(\theta)$ are ergodic averages. This means one can implement stochastic gradient ascent using either $\nabla_\theta \log \gamma_{t,\theta}(1)$ or $\nabla_\theta \log \gamma_{t_n,\theta}(1) - \nabla_\theta \log \gamma_{t_{n-1},\theta}(1)$ for any $t_{n-1} < t_n$ as estimates of $\mathcal{L}(\theta)$. Given an initial $\theta_0 \in \Theta$, as we obtain the observation path continuously in time, we will update θ at times $T \in \mathbb{N}$ using the following recursion:

$$\theta_T = \theta_{T-1} + \alpha_T \Big(\nabla_\theta \log(\gamma_{T,\theta_{T-1}}(1)) - \nabla_\theta \log(\gamma_{T-1,\theta_{T-1}}(1)) \Big) \tag{2.7}$$

where, for $T \in \mathbb{N}$, $\alpha_T \in \mathbb{R}^+$ is a collection of step-sizes that satisfy $\sum_{T\in\mathbb{N}} \alpha_T = \infty$, $\sum_{T\in\mathbb{N}} \alpha_T^2 < \infty$ to ensure convergence of the estimation as $T \to \infty$; see [3, 21] for details. This scheme can provide an online estimate for the parameter vector as data arrive. Steps are performed at $\mathcal{O}(1)$ times to ensure that enough information has accumulated to update the parameter. The adoption of unit times is made only for notational convenience. As both recursions (2.6) and (2.7) cannot be computed exactly, we focus upon methodologies that approximate the score function $\nabla_\theta \log(\gamma_{T,\theta}(1))$.

3. Direct Feynman-Kac Formulation.

3.1. Discretized Model. In practice, we will have to work with a discretization of the model in (2.1)-(2.2). We assume access to path of the data $\{Y_t\}_{0\leq t\leq T}$ which is available up-to an (almost) arbitrarily fine level of time discretization. This would be a very finely discretized path, as accessing the actual continuous path of observation is not possible; this point is discussed later on. One could focus on a time-discretization of either side of (2.5), however, as is conventional in the literature (e.g. [1, 20]) we focus on the left hand side.

Let $l \in \mathbb{N}_0$ and consider an Euler-Maruyama time-discretization with step-size $\Delta_l = 2^{-l}$. That is, for $k \in \{1, 2, \ldots, T/\Delta_l\}$:

$$\widetilde{X}_{k\Delta_l} = \widetilde{X}_{(k-1)\Delta_l} + b_\theta(\widetilde{X}_{(k-1)\Delta_l})\Delta_l + \sigma(\widetilde{X}_{(k-1)\Delta_l})(W_{k\Delta_l} - W_{(k-1)\Delta_l}), \quad \widetilde{X}_0 = x_*. \tag{3.1}$$

Note that the Brownian motion in (3.1) is the same as in (2.2) under both $\mathbb{P}_\theta$ and $\overline{\mathbb{P}}_\theta$. We set:

$$\begin{aligned}\lambda^l_{T,\theta}(x_0,x_{\Delta_l},\ldots,x_T) := \sum_{k=0}^{T/\Delta_l - 1} \Big\{ &(\nabla_\theta b_\theta(x_{k\Delta_l}))^* a(x_{k\Delta_l})^{-1}\sigma(x_{k\Delta_l})(W_{(k+1)\Delta_l} - W_{k\Delta_l}) \\ &+ (\nabla_\theta h_\theta(x_{k\Delta_l}))^*(Y_{(k+1)\Delta_l} - Y_{k\Delta_l}) - (\nabla_\theta h_\theta(x_{k\Delta_l}))^* h_\theta(x_{k\Delta_l})\Delta_l \Big\}.\end{aligned} \tag{3.2}$$

We remark that $\lambda^l_{T,\theta}$ is a function also of the observations, but this dependence is suppressed from the notation. For $k \in \{0, 1, \dots, T/\Delta_l - 1\}$, we define:

$$g^l_{k,\theta}(x_{k\Delta_l}) := \exp\Big\{h_\theta(x_{k\Delta_l})^*(y_{(k+1)\Delta_l} - y_{k\Delta_l}) - \tfrac{\Delta_l}{2} h_\theta(x_{k\Delta_l})^* h_\theta(x_{k\Delta_l})\Big\}.$$

Note that:

$$\begin{aligned} Z^l_{T,\theta}(x_0, x_{\Delta_l}, \dots, x_T) &:= \prod_{k=0}^{T/\Delta_l - 1} g^l_{k,\theta}(x_{k\Delta_l}) \\ &= \exp\Big\{ \sum_{k=0}^{T/\Delta_l - 1} \big[\, h_\theta(x_{k\Delta_l})^*(y_{(k+1)\Delta_l} - y_{k\Delta_l}) - \tfrac{\Delta_l}{2} h_\theta(x_{k\Delta_l})^* h_\theta(x_{k\Delta_l}) \,\big] \Big\} \end{aligned}$$

is a time-discretization of $Z_{T,\theta}$. We thus obtain the discretized approximation of the score function $\nabla_\theta \log(\gamma_{T,\theta}(1))$:

$$\nabla_\theta \log(\gamma^l_{T,\theta}(1)) := \frac{\overline{\mathbb{E}}_\theta\,[\, \lambda^l_{T,\theta}(\widetilde{X}_0, \widetilde{X}_{\Delta_l}, \dots, \widetilde{X}_T)\, Z^l_{T,\theta}(\widetilde{X}_0, \widetilde{X}_{\Delta_l}, \dots, \widetilde{X}_T) \,|\, \mathcal{Y}_T \,]}{\overline{\mathbb{E}}_\theta\,[\, Z^l_{T,\theta}(\widetilde{X}_0, \widetilde{X}_{\Delta_l}, \dots, \widetilde{X}_T) \,|\, \mathcal{Y}_T \,]}. \tag{3.3}$$

We have the following result which establishes the convergence of our Euler approximation. Below $\|\cdot\|_2$ is the L_2-norm for vectors. The proof is given in the supplementary material of the article.

Theorem 3.1. *Assume (D1). Then for any $(r, T) \in [1, \infty) \times \mathbb{N}$ there exists a $C < +\infty$ such that for any $l \in \mathbb{N}_0$*

$$\mathbb{E}_\theta \Big[\big\| \nabla_\theta \log(\gamma_{T,\theta}(1)) - \nabla_\theta \log(\gamma^l_{T,\theta}(1)) \big\|_2^r \Big]^{1/r} \leq C \Delta_l^{1/2}.$$

The result is fairly standard, but we note it is not a simple application of results on SDEs in filtering (e.g. [26, 32]) and this is reflected in the proof. The rate of convergence of the approximation will be very relevant for some of our subsequent discussions.

3.2. Backward Feynman-Kac Model and Particle Smoothing. From herein the $\widetilde{X}$ notation is dropped for simplicity. Consider the time interval $[k, k+1]$ and the k-th update of (2.7). We define the discrete-time approximation (at level l) as:

$$u_{k,l} = (x_{k+\Delta_l}, \dots, x_{k+1}) \in E_l := (\mathbb{R}^{d_x})^{\Delta_l^{-1}}, \quad k \in \mathbb{N}_0.$$

Recall (2.3). A discrete-time approximation of $p_\theta(\{Y_t\}_{k \leq t \leq k+1} | \{X_t\}_{0 \leq t \leq T})$ is:

$$G^l_{k,\theta}(u_{k-1,l}, u_{k,l}) := \prod_{p=0}^{\Delta_l^{-1}-1} g^l_{k+p,\theta}(x_{k+p\Delta_l}),$$

where we set $u_{-1,l} = x_*$, for each $l \in \mathbb{N}_0$. We denote by m^l_θ the Euler transition density induced by time-discretisation (3.1), and then write the initial distribution and Markov transition kernel for the discrete-time process with $k \in \mathbb{N}$ as follows:

$$\begin{aligned} \eta^l_{0,\theta}(du_{0,l}) &= \prod_{p=1}^{\Delta_l^{-1}} m^l_\theta(x_{(p-1)\Delta_l}, x_{p\Delta_l}) dx_{p\Delta_l}; \\ M^l_\theta(u_{k-1,l}, du_{k,l}) &= \prod_{p=1}^{\Delta_l^{-1}} m^l_\theta(x_{k+(p-1)\Delta_l}, x_{k+p\Delta_l}) dx_{k+p\Delta_l}. \end{aligned}$$

Remark 3.1. *The definition of $G^l_{k,\theta}(u_{k-1,l}, u_{k,l})$, $M^l_\theta(u_{k-1,l}, du_{k,l})$ implies: i) $G^l_{k,\theta}(u_{k-1,l}, u_{k,l})$ involves $u_{k-1,l}$ only via its very last element, x_k; ii) the dynamics of $u_{k,l}$ conditionally on $u_{k-1,l}$ depend only on the very last element, x_k, of $u_{k-1,l}$.*

We can now state the discrete-time filtering distribution for $k \in \mathbb{N}_0$:

$$\pi^l_{k,\theta}\big(d(u_{0,l}, \dots, u_{k,l})\big) := \frac{\big(\prod_{p=0}^k G^l_{p,\theta}(u_{p-1,l}, u_{p,l})\big)\, \eta^l_{0,\theta}(du_{0,l}) \prod_{p=1}^k M^l_\theta(u_{p-1,l}, du_{p,l})}{\int_{E_l^{k+1}} \big(\prod_{p=0}^k G^l_{p,\theta}(u_{p-1,l}, u_{p,l})\big)\, \eta^l_{0,\theta}(du_{0,l}) \prod_{p=1}^k M^l_\theta(u_{p-1,l}, du_{p,l})}. \tag{3.4}$$

That is, $\pi^l_{k,\theta}\big(d(u_{0,l}, \dots, u_{k,l})\big)$ is a discrete-time approximation of the filtering distribution:

$$\pi_{k,\theta}\big(d(\{X_t\}_{0\le t\le k})\big) := \mathbb{P}_\theta(d\{X_t\}_{0\le t\le k} | \{Y_t\}_{0\le t\le k}).$$

Expression (3.4) corresponds to a standard Feynman-Kac model (see e.g. [12]), thus one can approximate the involved filtering distributions via the corresponding Monte Carlo methodology.

We develop a Monte Carlo method for the approximation of the discretised score function in (3.3). This is accomplished by presenting a backward formula for (3.3). We define for any $p \in \mathbb{N}_0$:

$$\begin{aligned} f^l_\theta(x_{p\Delta_l}, x_{(p+1)\Delta_l}) :=& (\nabla_\theta b_\theta(x_{p\Delta_l}))^* a(x_{p\Delta_l})^{-1} \sigma(x_{p\Delta_l})(W_{(p+1)\Delta_l} - W_{p\Delta_l}) \\ &+ (\nabla_\theta h_\theta(x_{p\Delta_l}))^* (Y_{(p+1)\Delta_l} - Y_{p\Delta_l}) - (\nabla_\theta h_\theta(x_{p\Delta_l}))^* h_\theta(x_{p\Delta_l}) \Delta_l. \end{aligned}$$

and let:

$$\Lambda^l_{k,\theta}(u_{k-1,l}, u_{k,l}) := \sum_{p=0}^{\Delta_l^{-1}-1} f^l_\theta(x_{k+p\Delta_l}, x_{k+(p+1)\Delta_l});$$

$$F^l_{T,\theta}(u_{0,l}, \dots, u_{T-1,l}) := \sum_{k=0}^{T-1} \Lambda^l_{k,\theta}(u_{k-1,l}, u_{k,l}) \ \Big(\equiv \lambda^l_{T,\theta}(x_0, x_{\Delta_l}, \dots, x_T)\Big), \tag{3.5}$$

for $\lambda^l_{T,\theta}(x_0, x_{\Delta_l}, \dots, x_T)$ as defined in (3.2) and used in the score function approximation (3.3). Thus:

$$\nabla_\theta \log(\gamma^l_{T,\theta}(1)) = \int_{E_l^T} F^l_{T,\theta}(u_{0,l}, \dots, u_{T-1,l})\, \mathbb{Q}^l_{T-1,\theta}\big(d(u_{0,l}, \dots, u_{T-1,l})\big), \tag{3.6}$$

where $\mathbb{Q}^l_{T-1,\theta}\big(d(u_{0,l}, \dots, u_{T-1,l})\big)$ is a time-discretisation of the smoothing law:

$$\mathbb{Q}_{T-1,\theta}(d\{X_t\}_{0\le t\le T}) := \mathbb{P}_\theta(d\{X_t\}_{0\le t\le T} | \{Y_t\}_{0\le t\le T}).$$

Now, by the time-reversal formula for hidden Markov models (see e.g. [13, 14]) one has:

$$\mathbb{Q}^l_{T-1,\theta}\big(d(u_{0,l}, \dots, u_{T-1,l})\big) := \pi^l_{T-1,\theta}(du_{T-1,l}) \prod_{k=1}^{T-1} B^l_{k,\theta,\pi^l_{k-1,\theta}}(u_{k,l}, du_{k-1,l}),$$

for the backward Markov kernel:

$$B^l_{k,\theta,\pi^l_{k-1,\theta}}(u_{k,l}, du_{k-1,l}) := \frac{\pi^l_{k-1,\theta}(du_{k-1,l})\, G^l_{k,\theta}(u_{k-1,l}, u_{k,l}) m^l_\theta(u_{k-1,l}, u_{k,l})}{\pi^l_{k-1,\theta}(G^l_{k,\theta}(\cdot, u_{k,l}) m^l_\theta(\cdot, u_{k,l}))}, \tag{3.7}$$

under the standard notation:

$$\pi^l_{k-1,\theta}(G^l_{k,\theta}(\cdot,u_{k,l})m^l_\theta(\cdot,u_{k,l})) = \int_{E_l}\pi^l_{k-1,\theta}(du_{k-1,l})G^l_{k,\theta}(u_{k-1,l},u_{k,l})m^l_\theta(u_{k-1,l},u_{k,l}).$$

REMARK 3.2. *The model structure gives important cancellations in (3.7), so that:*

$$B^l_{k,\theta,\pi^l_{k-1,\theta}}(u_{k,l},du_{k-1,l}) \equiv \frac{\pi^l_{k-1,\theta}(du_{k-1,l})g^l_{k,\theta}(x_k)m^l_\theta(x_k,x_{k+\Delta_l})}{\int_{E_l}\pi^l_{k-1,\theta}(du_{k-1,l})g^l_{k,\theta}(x_k)m^l_\theta(x_k,x_{k+\Delta_l})}.$$

The objective now is to approximate the right hand side of (3.6) using particle approximations. Our online particle approximation of the gradient of the log-likelihood in (3.6), for a given $l\in\mathbb{N}_0$ is presented in Algorithm 3.1. Our estimates are given in (3.8) and (3.10) in Algorithm 3.1. The approach is the method introduced in [13, 14].

Algorithm 3.1 Online Score Function Estimation for a given $l\in\mathbb{N}_0$.

1. For $i\in\{1,\dots,N\}$, sample $u^i_{0,l}$ i.i.d. from $\eta^l_{0,\theta}(\cdot)$. The estimate of $\nabla_\theta\log(\gamma^l_{1,\theta}(1))$ is:

$$\nabla_\theta\widehat{\log(\gamma^l_{1,\theta}(1))} := \frac{\sum_{i=1}^N G^l_{0,\theta}(x_*,u^i_{0,l})\Lambda^l_{0,\theta}(x_*,u^i_{0,l})}{\sum_{i=1}^N G^l_{0,\theta}(x_*,u^i_{0,l})}. \tag{3.8}$$

   Set $k=1$, and for $i\in\{1,\dots,N\}$, $\breve{u}^i_{-1,l}=x_*$.
2. For $i\in\{1,\dots,N\}$ sample $\breve{u}^i_{k-1,l}$ from:

$$\sum_{i=1}^N \frac{G^l_{k-1,\theta}(\breve{u}^i_{k-2,l},u^i_{k-1,l})}{\sum_{j=1}^N G^l_{k-1,\theta}(\breve{u}^j_{k-2,l},u^j_{k-1,l})}\delta_{\{u^i_{k-1,l}\}}(\cdot).$$

   If $k=1$, for $i\in\{1,\dots,N\}$, set $F^{l,N}_{k-1,\theta}(\breve{u}^i_{0,l})=\Lambda^l_{0,\theta}(x_*,\breve{u}^i_{0,l})$.
3. For $i\in\{1,\dots,N\}$, sample $u^i_{k,l}$ from $M^l_\theta(\breve{u}^i_{k-1,l},\cdot)$. For $i\in\{1,\dots,N\}$, compute:

$$F^{l,N}_{k,\theta}(u^i_{k,l}) = \frac{\sum_{j=1}^N g^l_{k,\theta}(\breve{x}^j_k)m^l_\theta(\breve{x}^j_k,x^i_{k+\Delta_l})\{F^{l,N}_{k-1,\theta}(\breve{u}^j_{k-1,l})+\Lambda^l_{k,\theta}(\breve{u}^j_{k-1,l},u^i_{k,l})\}}{\sum_{j=1}^N g^l_{k,\theta}(\breve{x}^j_k)m^l_\theta(\breve{x}^j_k,x^i_{k+\Delta_l})}. \tag{3.9}$$

   The estimate of $\nabla_\theta\log(\gamma^l_{k+1,\theta}(1))$ is:

$$\nabla_\theta\widehat{\log(\gamma^l_{k+1,\theta}(1))} := \frac{\sum_{i=1}^N G^l_{k,\theta}(\breve{u}^i_{k-1,l},u^i_{k,l})F^{l,N}_{k,\theta}(u^i_{k,l})}{\sum_{i=1}^N G^l_{k,\theta}(\breve{u}^i_{k-1,l},u^i_{k,l})}. \tag{3.10}$$

   Set $k=k+1$ and return to the start of 2..

3.3. Discussion of Algorithm 3.1. There are several remarks worth making, before proceeding. Firstly, the cost of this algorithm per unit time is $\mathcal{O}(N\Delta_l^{-1}+N^2)$. In detail, the cost of the particle filter is $\mathcal{O}(N\Delta_l^{-1})$. The cost of calculating $F^{l,N}_{k,\theta}(u^i_{k,l})$, $1\le i\le N$, in (3.9), is $\mathcal{O}(N^2)$;

Δ_l is not involved here due to cancellations – see Remark 3.2. The cost of (3.10) is $\mathcal{O}(N)$, given the particle filter has already been executed. There are several implications of this remark. Based upon the results in [13] and Theorem 3.1, in a sequel work we prove, under appropriate assumptions, we will have the following MSE, for $(k, N) \in \mathbb{N}^2$:

$$\mathbb{E}_\theta\Big[\Big\|\nabla_\theta \widehat{\log(\gamma_{k,\theta}^l}(1)) - \nabla_\theta \log(\gamma_{k,\theta}(1))\Big\|_2^2\Big] \leq C\Big(\frac{1}{N} + \Delta_l\Big), \tag{3.11}$$

where C is a constant that does not depend on N or l. To achieve an MSE of $\mathcal{O}(\epsilon^2)$ for some $\epsilon > 0$ given, one sets l so that $\Delta_l = \epsilon^2$ (i.e. $l = \mathcal{O}(|\log(\epsilon)|)$) and $N = \mathcal{O}(\epsilon^{-2})$. The cost per unit time of doing this is then $\mathcal{O}(\epsilon^{-4})$. We note that to choose l as specified, one has to have access to an appropriately finely observed data set and this is assumed throughout. Typically, one could use a multilevel Monte Carlo method, as in [20], to reduce the cost to achieve an MSE of $\mathcal{O}(\epsilon^2)$. However, in this case as the $\mathcal{O}(N^2)$ cost dominates and does not depend on l, one can easily check that such a variance reduction method *will not improve* our particle method. To understand this, one can prove a bound on the MSE, for instance of the type conjectured later in this article (4.19), and then try to minimize the cost, by selecting the appropriate number of samples on each level to obtain a given MSE. This latter problem leads to a constrained minimization problem that can be solved using Lagrange multipliers (as in e.g. [9]), but one can show that this yields that the order of the cost to achieve an MSE of $\mathcal{O}(\epsilon^2)$ is still $\mathcal{O}(\epsilon^{-4})$.

Secondly, it is important to note that the method in [23] can reduce the cost of online score estimation to $\mathcal{O}(N\Delta_l^{-1})$ per unit time. However, in order to do so, one requires that $m_\theta^l(x, x')$ is uniformly lower-bounded in (x, x'), which does not typically occur for Euler-discretized diffusion densities. As a result, we only use the approach shown in Algorithm 3.1. Note that [16], in a different but related context, considers using unbiased and non-negative estimates of the transition density in the approach in [23], but such estimates are not always available.

Thirdly and rather importantly, there is a potential issue related to the construction of the algorithm. We have started with a continuous-time formula, discretized it and applied what are essentially discrete-time methods for smoothing of additive functionals. A serious caveat is that the algorithm is not well-defined as $l \to \infty$, which is what we mean by saying it has *no (Wiener) path-space formulation*. The source of the issue is related to using approximations of the transition density $m_\theta^l(\check{x}_k^j, x_{k+\Delta_l}^i)$ in (3.9), which can degenerate when l is high. This will result in increasing Monte Carlo variance and computational cost and may mean that C in (3.11) explodes exponentially in l. We refer the reader to [36, Figure 1.1] for a numerical example. This issue has manifested itself in MCMC schemes for inferring fully observed SDEs (see e.g. [24]), but in the context of particle smoothing and Algorithm 3.1 there are additional considerations. The resampling operation introduces *discontinuities*. Often such terminology refers to the lack of continuity of the transition density:

$$\theta \to \nabla_\theta \widehat{\log(\gamma_{k,\theta}^l}(1)),$$

but here we are interested in the behavior of $m_\theta^l(\check{x}_k^j, x_{k+\Delta_l}^i)$ when we combine points $\check{x}_k^j$ and $x_{k+\Delta_l}^i$ that are intrinsically not obtained in a continuous way as Δ_l diminishes. This issue has not received attention in earlier numerical studies, but remains a concern. As a result, we now consider defining an algorithm that is robust to the size of the time-discretization mesh and hence has a path-space formulation.

4. Path-Space Feynman-Kac Formulation.

4.1. Data Augmentation using Bridges.

We begin this section with a review of the method in [29, 34]. For simplicity we consider the case $t \in [0,1]$ and let $\mathbf{X} := \{X_t\}_{t\in[0,1]}$, and $\mathbf{W} := \{W_t\}_{t\in[0,1]}$. Let also $p_\theta(x,t;x',1)$ denote the unknown transition density from time t to 1 associated to (2.2) and let also $p_\theta(x,x') := p_\theta(x,0;x',1)$. Suppose one could sample from p_θ to obtain $(x,x') \in \mathbb{R}^{2d_x}$. Then one can interpolate these points by using a bridge process which has a drift given by $b_\theta(x) + a(x)\nabla_x \log p_\theta(x,t;x',1)$, as we will explain below. Let $\overline{\mathbb{P}}_{\theta,x,x'}$ denote the law of the solution of the SDE (2.2), on $[0,1]$, started at $x_* = x$ and conditioned to hit x' at time 1.

As p_θ is intractable in general, we consider a user-specified auxiliary process $\{\tilde{X}_t\}_{t\in[0,1]}$ following:

$$d\tilde{X}_t = \tilde{b}_\theta(t,\tilde{X}_t)dt + \tilde{\sigma}(t,\tilde{X}_t)dW_t, \quad t\in[0,1], \quad \tilde{X}_0 = x, \tag{4.1}$$

where for each parameter value $\theta\in\Theta$, $\tilde{b}_\theta : [0,1]\times\mathbb{R}^{d_x} \to \mathbb{R}^{d_x}$ and $\tilde{\sigma} : \mathbb{R}^{d_x} \to \mathbb{R}^{d_x\times d_x}$ is such that $\tilde{a}(1,x') := \tilde{\sigma}(1,x')\tilde{\sigma}(1,x')^* \equiv a(x')$. Most importantly, (4.1) is chosen so that its transition density is available. To avoid confusion – as the specification of process (4.1) can involve parameter θ and a given ending position x' – we note that the transition density of (4.1) from time t to time 1 corresponds to a mapping $z \to \tilde{p}_{\theta,x'}(x,t;z,1)$. We also use the notation $\tilde{p}_{\theta,x'}(x,z) := \tilde{p}_{\theta,x'}(x,0;z,1)$. One possible choice is to use an Ornstein-Uhlenbeck process (e.g. obtained using linearizations or variational inference with (2.2) [29, Section 1.3]); see also [29, Section 2.2] for technical conditions on $\tilde{b}_\theta, \tilde{a}, \tilde{p}_{\theta,x'}$. The main purpose of $\{\tilde{X}_t\}_{t\in[0,1]}$ is to construct another process $\{X^\circ_t\}_{t\in[0,1]}$ conditioned to hit a given x' at $t=1$. The latter will form an importance proposal for $\{X_t\}_{t\in[0,1]}$. Let:

$$dX^\circ_t = b^\circ_\theta(t,X^\circ_t;x')dt + \sigma(X^\circ_t)dW_t, \quad t\in[0,1], \quad X^\circ_0 = x, \tag{4.2}$$

where:

$$b^\circ_\theta(t,x;x') = b_\theta(x) + a(x)\nabla_x \log \tilde{p}_{\theta,x'}(x,t;x',1),$$

and denote by $\mathbb{P}^\circ_{\theta,x,x'}$ the probability law of the solution of (4.2). The SDE in (4.2) gives rise to a function:

$$\mathbf{W} \to C_\theta(x,\mathbf{W},x'), \tag{4.3}$$

mapping the driving Wiener noise $\mathbf{W}$ to the solution of (4.2), so we have effectively reparameterized the problem from $\mathbf{X}$ to $(\mathbf{W},x')$.

Now, following [29], the two measures $\overline{\mathbb{P}}_{\theta,x,x'}$ and $\mathbb{P}^\circ_{\theta,x,x'}$ are absolutely continuous w.r.t. each other, with Radon-Nikodym derivative:

$$\frac{d\overline{\mathbb{P}}_{\theta,x,x'}}{d\mathbb{P}^\circ_{\theta,x,x'}}(\mathbf{X}) = \exp\left\{\int_0^1 L_\theta(t,X_t)dt\right\} \times \frac{\tilde{p}_{\theta,x'}(x,x')}{p_\theta(x,x')}, \tag{4.4}$$

where:

$$\begin{aligned}&L_\theta(t,x) := \big(b_\theta(x) - \tilde{b}_\theta(t,x)\big)^* \nabla_x \log \tilde{p}_{\theta,x'}(x,t;x',1)\\ &-\tfrac{1}{2}\mathrm{Tr}\Big\{\big[a(x)-\tilde{a}(t,x)\big]\big[-\nabla_x^2 \log \tilde{p}_{\theta,x'}(x,t;x',1) - \nabla_x \log \tilde{p}_{\theta,x'}(x,t;x',1)\nabla_x \log \tilde{p}_{\theta,x'}(x,t;x',1)^*\big]\Big\}\end{aligned}$$

with $\mathrm{Tr}(\cdot)$ denoting the trace of a squared matrix. Note that, in the case when $\sigma = \sigma(x)$ is not a constant function, then, typically, $x' \to \tilde{p}_{\theta,x'}(x,x')$ will not integrate to 1 and will give rise to a non-trivial distribution to sample from. As the complete algorithm will require being able to sample from the transition density, we rewrite:

$$\frac{d\overline{\mathbb{P}}_{\theta,x,x'}}{d\widetilde{\mathbb{P}}_{\theta,x,x'}}(\mathbf{X}) = \exp\Big\{\int_0^1 L_\theta(t,X_t)dt\Big\} \times \frac{\tilde{p}_{\theta,x'}(x,x')}{p_\theta(x,x')\hat{p}_\theta(x,x')} \times \hat{p}_\theta(x,x'), \tag{4.5}$$

where an arbitrary, tractable and easy to sample density $\hat{p}_\theta(x,x')$ is used to sample x'.

4.2. Estimation of Score in Continuous-Time. We return to the expression of the score function in (2.4) and use the alternative change of measure described above. Consider the processes:

$$\mathbf{X}_k := \{X_t\}_{t\in[k,k+1]}, \quad \mathbf{Y}_k := \{Y_t\}_{t\in[k,k+1]}, \quad k \in \mathbb{N}_0.$$

We introduce the following notation:

$$\Psi_\theta(\mathbf{X}_k) = \int_k^{k+1} L_\theta(t,X_t)dt;$$

$$J_{k,\theta}(\mathbf{X}_k,\mathbf{Y}_k) = \int_k^{k+1} h_\theta(X_t)^* dY_t - \tfrac{1}{2}\int_k^{k+1} h_\theta(X_t)^* h_\theta(X_t)dt;$$

$$\Lambda_{k,\theta}(\mathbf{X}_k,\mathbf{Y}_k) = \int_k^{k+1} (\nabla_\theta b_\theta(X_t))^* a(X_t)^{-1}(dX_t - b_\theta(X_t)dt) + \int_k^{k+1} (\nabla_\theta h_\theta(X_t))^* dY_t - \int_k^{k+1} (\nabla_\theta h_\theta(X_t))^* h_\theta(X_t)dt;$$

$$\Phi_{k,\theta}(\mathbf{X}_k,\mathbf{Y}_k) = J_{k,\theta}(\mathbf{X}_k,\mathbf{Y}_k) + \Psi_\theta(\mathbf{X}_k) + \log\frac{\tilde{p}_{\theta,x_{k+1}}(x_k,x_{k+1})}{\hat{p}_\theta(x_k,x_{k+1})}.$$

Note all the integrands can be computed point wise.

Remark 4.1. *The Wiener process in (4.3) is defined on the time interval $[0,1]$, thus so is the transform $C_\theta(x,\mathbf{W},x')$. In the derivations below, one needs to calculate $J_{k,\theta}(C_\theta(x_k,\mathbf{W}_k,x_{k+1}),\mathbf{Y}_k)$ and $\Lambda_{k,\theta}(C_\theta(x_k,\mathbf{W}_k,x_{k+1}),\mathbf{Y}_k)$, for $\mathbf{W}_k$'s that correspond to samples from the Wiener measure on $[0,1]$. With some abuse of notation, it is to be understood that the path $C_\theta(x_k,\mathbf{W}_k,x_{k+1})$ is 'shifted' from $[0,1]$ to $[k,k+1]$, so all quantities below agree with the notation introduced above. Also, the calculation of $\Psi_\theta(C_\theta(x_k,\mathbf{W}_k,x_{k+1}))$ will be required, but this should create no confusion.*

Under these definitions, for any $T\in\mathbb{N}$ the score function in (2.4) can be rewritten as:

$$\nabla_\theta \log(\gamma_{T,\theta}(1)) \equiv \frac{\overline{\mathbb{E}}_\theta\Big[\,\big(\sum_{k=0}^{T-1}\Lambda_{k,\theta}(\mathbf{X}_k,\mathbf{Y}_k)\big)\exp\big\{\sum_{k=0}^{T-1}J_{k,\theta}(\mathbf{X}_k,\mathbf{Y}_k)\big\}\,\big|\,\mathcal{Y}_T\Big]}{\overline{\mathbb{E}}_\theta\Big[\,\exp\big\{\sum_{k=0}^{T-1}J_{k,\theta}(\mathbf{X}_k,\mathbf{Y}_k)\big\}\,\big|\,\mathcal{Y}_T\Big]}.$$

Making use of the transform (4.3) and the density expression in (4.4), we can equivalently write:

$$\nabla_\theta \log(\gamma_{T,\theta}(1)) = \tag{4.6}$$

$$= \frac{\widetilde{\mathbb{E}}_\theta\Big[\,\big(\sum_{k=0}^{T-1}\Lambda_{k,\theta}(C_\theta(x_k,\mathbf{W}_k,x_{k+1}),\mathbf{Y}_k)\big)\exp\big\{\sum_{k=0}^{T-1}\Phi_{k,\theta}(C_\theta(x_k,\mathbf{W}_k,x_{k+1}),\mathbf{Y}_k)\big\}\,\big|\,\mathcal{Y}_T\Big]}{\widetilde{\mathbb{E}}_\theta\Big[\,\exp\big\{\sum_{k=0}^{T-1}\Phi_{k,\theta}(C_\theta(x_k,\mathbf{W}_k,x_{k+1}),\mathbf{Y}_k)\big\}\,\big|\,\mathcal{Y}_T\Big]},$$

where, the expectation $\widetilde{\mathbb{E}}_\theta\,[\,\cdot\,|\,\mathcal{Y}_T\,]$ is considered under the probability measure:

$$\widetilde{\mathbb{P}}_\theta\left(\,d(\mathbf{W}_0, x_1, \dots, \mathbf{W}_{T-1}, x_T)\,\right) := \bigotimes_{k=0}^{T-1}\big[\,\mathbb{W}(d\mathbf{W}_k)\otimes \hat{p}_\theta(x_k, x_{k+1})dx_{k+1}\,\big], \tag{4.7}$$

independently of $\mathcal{Y}_T$; here, $\mathbb{W}$ is the standard Wiener measure on $[0,1]$ and $x_0 = x_*$.

REMARK 4.2. *The approach that has been adopted here can also be used if σ depends upon θ. Assuming the formula is well-defined, one would have a score function with an expression of the type:*

$$\frac{\widetilde{\mathbb{E}}_\theta\Big[\,\big(\sum_{k=0}^{T-1}\Xi_{k,\theta}(C_\theta(x_k,\mathbf{W}_k,x_{k+1})\big)\exp\big\{\sum_{k=0}^{T-1}\Phi_{k,\theta}(C_\theta(x_k,\mathbf{W}_k,x_{k+1}),\mathbf{Y}_k)\big\}\,\big|\,\mathcal{Y}_T\,\Big]}{\widetilde{\mathbb{E}}_\theta\Big[\,\exp\big\{\sum_{k=0}^{T-1}\Phi_{k,\theta}(C_\theta(x_k,\mathbf{W}_k,x_{k+1}),\mathbf{Y}_k)\big\}\,\big|\,\mathcal{Y}_T\,\Big]},$$

where:

$$\Xi_{k,\theta}(C_\theta(x_k,\mathbf{W}_k,x_{k+1}) = \{\nabla_\theta\Phi_{k,\theta}(C_\theta(x_k,\mathbf{W}_k,x_{k+1}),\mathbf{Y}_k)\} + \nabla_\theta\log\{\hat{p}_\theta(x_k,x_{k+1})\},$$

and with an appropriate modification of the approach to allow σ to depend on θ. To keep consistency with the ideas in Section 3 we do not consider the formula from herein, but remark that extension of the forthcoming methodology to this case is straightforward.

REMARK 4.3. *Recent advances in [6] have extended the construct of the auxiliary bridge process – developed via (4.1), (4.2) herein – to the setting of* hypoelliptic *SDEs. Though we do not pursue this direction here for the purpose of easing the exposition, we remark that, given these new developments, one can now, in principle, obtain score function estimates – thus, also carry out parameter inference – in the hypoelliptic regime along the same steps we follow in the current work.*

The expression in (4.6), together with the (trivially) Markovian dynamics for the process in (4.7) allow one to construct a backward Feynman-Kac type formula as in (3.6). To better connect the approach here and that in Section 3, define $u_k = (\mathbf{W}_k, x_{k+1})$ for $k\in\mathbb{N}_0$, $u_{-1} = x_*$ and:

$$\begin{aligned}\Lambda^C_{k,\theta}(u_{k-1},u_k) &= \Lambda_{k,\theta}(C_\theta(x_k,\mathbf{W}_k,x_{k+1}),\mathbf{Y}_k);\\ \Phi^C_{k,\theta}(u_{k-1},u_k) &= \Phi_{k,\theta}(C_\theta(x_k,\mathbf{W}_k,x_{k+1}),\mathbf{Y}_k);\\ F_{T,\theta}(u_0,\dots,u_{T-1}) &= \sum_{k=0}^{T-1}\Lambda^C_{k,\theta}(u_{k-1},u_k).\end{aligned}$$

Superscript C is motivated by the well-posedness of the formula in *continuous-time* path-space. Set:

$$\pi_{k,\theta}\big(\,d(u_0,\dots,u_k)\,\big) := \frac{\Big(\prod_{p=0}^{k}\exp\{\Phi^C_{p,\theta}(u_{p-1},u_p)\}\Big)\widetilde{\mathbb{P}}_\theta\left(\,d(u_0,\dots,u_k)\,\right)}{\int_{E^{k+1}}\Big(\prod_{p=0}^{k}\exp\{\Phi^C_{p,\theta}(u_{p-1},u_p)\}\Big)\widetilde{\mathbb{P}}_\theta\left(\,d(u_0,\dots,u_k)\,\right)},$$

with $E = C([0,1],\mathbb{R}^{d_x})$. Then, we have the representation:

$$\nabla_\theta\log(\gamma_{T,\theta}(1)) = \int_{E^T} F_{T,\theta}(u_0,\dots,u_{T-1})\,\mathbb{Q}_{T-1,\theta}\big(\,d(u_0,\dots,u_{T-1})\,\big), \tag{4.8}$$

where:

$$\mathbb{Q}_{T-1,\theta}\big(d(u_0,\dots,u_{T-1})\big) = \pi_{T-1,\theta}(du_{T-1})\prod_{k=1}^{T-1} B_{k,\theta,\pi_{k-1,\theta}}(u_k, du_{k-1}),$$

and:

$$B_{k,\theta,\pi_{k-1,\theta}}(u_k, du_{k-1}) := \frac{\pi_{k-1,\theta}(du_{k-1})\exp\{\Phi^C_{k,\theta}(u_{k-1},u_k)\}\hat{p}_\theta(x_k,x_{k+1})}{\pi_{k-1,\theta}(\exp\{\Phi^C_{k,\theta}(\cdot,u_k)\}\hat{p}_\theta(\cdot,x_{k+1}))}.$$

We remark that formula (4.8) is a type of backward Feynman-Kac formula in continuous-time, which to our knowledge is new.

As in the case of Algorithm 3.1, an effective Monte Carlo approximation of such a smoothing expectation (4.8) is given in Algorithm 4.1. The estimates of the score function are given in equations (4.9)-(4.10) in Algorithm 4.1.

Algorithm 4.1 Online Score Function Estimation on Path-Space

1. For $i \in \{1,\dots,N\}$, sample u_0^i i.i.d. from $\mathbb{W}(\cdot)\otimes\hat{p}_\theta(x_*,\cdot)$. The estimate of $\nabla_\theta\log(\gamma_{1,\theta}(1))$ is:

$$\nabla_\theta \widehat{\log(\gamma_{1,\theta}}(1)) := \frac{\sum_{i=1}^N \exp\{\Phi^C_{0,\theta}(x_*,u_0^i)\}\Lambda^C_{0,\theta}(x_*,u_0^i)}{\sum_{i=1}^N \exp\{\Phi^C_{0,\theta}(x_*,u_0^i)\}}. \tag{4.9}$$

   Set $k=1$ and for $i\in\{1,\dots,N\}$, $\check{u}^i_{-1} = x_*$.

2. For $i\in\{1,\dots,N\}$ sample $\check{u}^i_{k-1}$ from:

$$\sum_{i=1}^N \frac{\exp\{\Phi^C_{k-1,\theta}(\check{u}^i_{k-2},u^i_{k-1})\}}{\sum_{j=1}^N \exp\{\Phi^C_{k-1,\theta}(\check{u}^j_{k-2},u^j_{k-1})\}}\delta_{\{u^i_{k-1}\}}(\cdot).$$

   If $k=1$, for $i\in\{1,\dots,N\}$, set $\widetilde{F}^N_{k-1,\theta}(\check{u}^i_0) = \Lambda^C_{0,\theta}(x_*,\check{u}^i_0)$.

3. For $i\in\{1,\dots,N\}$, sample u^i_k from $\mathbb{W}(\cdot)\otimes\hat{p}^l_\theta(\check{x}^i_k,\cdot)$. For $i\in\{1,\dots,N\}$, compute:

$$\widetilde{F}^N_{k,\theta}(u^i_k) = \frac{\sum_{j=1}^N \exp\{\Phi^C_{k,\theta}(\check{u}^j_{k-1},u^i_k)\}\hat{p}_\theta(\check{x}^j_k,x^i_{k+1})\{\widetilde{F}^N_{k-1,\theta}(\check{u}^j_{k-1}) + \Lambda^C_{k,\theta}(\check{u}^j_{k-1},u^i_k)\}}{\sum_{j=1}^N \exp\{\Phi^C_{k,\theta}(\check{u}^j_{k-1},u^i_k)\}\hat{p}_\theta(\check{x}^j_k,x^i_{k+1})}.$$

   The estimate of $\nabla_\theta\log(\gamma_{k+1,\theta}(1))$ is:

$$\nabla_\theta \widehat{\log(\gamma_{k+1,\theta}}(1)) := \frac{\sum_{i=1}^N \exp\{\Phi^C_{k,\theta}(\check{u}^i_{k-1},u^i_k)\}\widetilde{F}^N_{k,\theta}(u^i_k)}{\sum_{i=1}^N \exp\{\Phi^C_{k,\theta}(\check{u}^i_{k-1},u^i_k)\}}. \tag{4.10}$$

   Set $k=k+1$ and return to the start of 2.

4.3. Time-Discretization. Whilst conceptually important, path-space valued Algorithm 4.1 can seldom be implemented directly in practice; unbiased methods e.g. [4] may be possible, but would be cumbersome. We develop a time-discretization procedure, in a similar manner to that considered in Section 3.2.

We will discretize on the uniform grid with increment $\Delta_l = 2^{-l}$. Let $k \in \mathbb{N}_0$ and define:

$$u_{k,l} = (z_{k+\Delta_l}, z_{k+2\Delta_l}, \dots, z_{k+1-\Delta_l}, x_{k+1}) \in (\mathbb{R}^{d_x})^{\Delta_l^{-1}} = E_l,$$

where $z_{k+\Delta_l}, z_{k+2\Delta_l}, \dots, z_{k+1-\Delta_l}$ will represent increments of Brownian motion and $u_{-1,l} = x_*$. Define the Markov kernel on E_l, for $k \in \mathbb{N}$:

$$\widetilde{M}_\theta^l(u_{k-1,l}, du_{k,l}) = \Big(\prod_{s=1}^{\Delta_l^{-1}-1} \phi_l(z_{k+s\Delta_l})dz_{k+s\Delta_l}\Big)\hat{p}_\theta(x_k, x_{k+1})dx_{k+1},$$

where $\phi_l(z_{k+s\Delta_l})$ is the density associated to the $\mathcal{N}_{d_x}(0, \Delta_l I_{d_x})$ distribution. We denote the density of $\widetilde{M}_\theta^l$ as $\widetilde{Q}_\theta^l$. Set:

$$\widetilde{\eta}_{0,\theta}^l(du_{0,l}) = \Big(\prod_{s=1}^{\Delta_l^{-1}-1} \phi_l(z_{s\Delta_l})dz_{s\Delta_l}\Big)\hat{p}_\theta(x_*, x_1)dx_1.$$

Now set, for $(k, s) \in \{0, 1, \dots, T-1\} \times \{0, 1, \dots, \Delta_l^{-1} - 2\}$:

$$X_{(s+1)\Delta_l+k} = X_{s\Delta_l+k} + b_\theta^\circ(s\Delta_l, X_{s\Delta_l+k}; x_{k+1})\Delta_l + \sigma(X_{s\Delta_l+k})Z_{(s+1)\Delta_l+k}. \tag{4.11}$$

Define for $k \in \mathbb{N}_0$:

$$\begin{aligned}
\Psi_\theta^l(u_{k-1,l}, u_{k,l}) &= \sum_{p=0}^{\Delta_l^{-1}-1} L_\theta(p\Delta_l, x_{k+p\Delta_l})\Delta_l;\\
J_{k,\theta}^l(u_{k-1,l}, u_{k,l}) &= \sum_{p=0}^{\Delta_l^{-1}-1} h_\theta(x_{k+p\Delta_l})^*(Y_{k+(s+1)\Delta_l} - Y_{k+s\Delta_l}) - \frac{1}{2}\sum_{p=0}^{\Delta_l^{-1}-1} h_\theta(x_{k+p\Delta_l})^* h_\theta(x_{k+p\Delta_l})\Delta_l;\\
\Phi_{k,\theta}^l(u_{k-1,l}, u_{k,l}) &= J_{k,\theta}^l(u_{k-1,l}, u_{k,l}) + \Psi_\theta^l(u_{k-1,l}, u_{k,l}) + \log\frac{\tilde{p}_{\theta,x_{k+1}}(x_k, x_{k+1})}{\hat{p}_\theta(x_k, x_{k+1})};\\
\widetilde{G}_{k,\theta}^l(u_{k-1,l}, u_{k,l}) &= \exp\{\Phi_{k,\theta}^l(u_{k-1,l}, u_{k,l})\}.
\end{aligned}$$

Now, for $k \in \mathbb{N}_0$:

$$\begin{aligned}
\widetilde{\Lambda}_{k,\theta}^l(u_{k-1,l}, u_{k,l}) = &\sum_{p=0}^{\Delta_l^{-1}-1} (\nabla_\theta b_\theta(x_{k+p\Delta_l}))^* a(x_{k+p\Delta_l})^{-1}(x_{k+(p+1)\Delta_l} - x_{k+p\Delta_l} - b_\theta(x_{k+p\Delta_l})\Delta_l)\\
&+ \sum_{p=0}^{\Delta_l^{-1}-1} (\nabla_\theta h_\theta(x_{k+p\Delta_l}))^*(Y_{k+(p+1)\Delta_l} - Y_{k+p\Delta_l}) - \sum_{p=0}^{\Delta_l^{-1}-1} (\nabla_\theta h_\theta(x_{k+p\Delta_l}))^* h_\theta(x_{k+p\Delta_l})\Delta_l.
\end{aligned}$$

Set:

$$\widetilde{F}_{T,\theta}^l(u_{0,l}, \dots, u_{T-1,l}) = \sum_{k=0}^{T-1} \widetilde{\Lambda}_{k,\theta}^l(u_{k-1,l}, u_{k,l}).$$

Writing expectations w.r.t. $\widetilde{\eta}^l_\theta(du_{0,l})\prod_{k=1}^{T-1}\widetilde{M}^l_\theta(u_{k-1,l},du_{k,l})$ as $\widetilde{\mathbb{E}}^l_\theta[\,\cdot\,|\mathcal{Y}_T]$, our discretized approximation of $\nabla_\theta\log(\gamma_{T,\theta}(1))$ is:

$$\nabla_\theta\log(\widetilde{\gamma}^l_{T,\theta}(1)) := \frac{\widetilde{\mathbb{E}}^l_\theta\left[\,\widetilde{F}^l_{T,\theta}(U_{0,l},\dots,U_{T-1,l})\prod_{k=0}^{T-1}\widetilde{G}^l_{k,\theta}(U_{k-1,l},U_{k,l})\,\middle|\,\mathcal{Y}_T\,\right]}{\widetilde{\mathbb{E}}^l_\theta\left[\,\prod_{k=0}^{T-1}\widetilde{G}^l_{k,\theta}(U_{k-1,l},U_{k,l})\,\middle|\,\mathcal{Y}_T\,\right]}.$$

We note that, whilst terms $\nabla_\theta\log(\widetilde{\gamma}^l_{T,\theta}(1))$, $\nabla_\theta\log(\gamma^l_{T,\theta}(1))$ should both converge to $\nabla_\theta\log(\gamma_{T,\theta}(1))$, as $l\to\infty$, they will in general be different for any fixed l.

One can also easily develop a discretized time reversal formula such as (3.6) which will converge precisely to (4.8) as $l\to\infty$. Define, for $k\in\mathbb{N}_0$:

$$\widetilde{\pi}^l_{k,\theta}\big(d(u_{0,l},\dots,u_{k,l})\big) := \frac{\big(\prod_{p=0}^k\widetilde{G}^l_{p,\theta}(u_{p-1,l},u_{p,l})\big)\,\widetilde{\eta}^l_{0,\theta}(du_{0,l})\prod_{p=1}^k\widetilde{M}^l_\theta(u_{p-1,l},du_{p,l})}{\int_{E_l^{k+1}}\Big(\prod_{p=0}^k\widetilde{G}^l_{p,\theta}(u_{p-1,l},u_{p,l})\Big)\widetilde{\eta}^l_{0,\theta}(du_{0,l})\prod_{p=1}^k\widetilde{M}^l_\theta(u_{p-1,l},du_{p,l})}.$$

Then, we have that:

$$\nabla_\theta\log(\widetilde{\gamma}^l_{T,\theta}(1)) = \int_{E_l^T}\widetilde{F}^l_{T,\theta}(u_{0,l},\dots,u_{T-1,l})\,\widetilde{\mathbb{Q}}^l_{T-1,\theta}\big(d(u_{0,l},\dots,u_{T-1,l})\big),\tag{4.12}$$

where we have defined:

$$\widetilde{\mathbb{Q}}^l_{T-1,\theta}\big(d(u_{0,l},\dots,u_{T-1,l})\big) := \widetilde{\pi}^l_{T-1,\theta}(du_{T-1,l})\prod_{k=1}^{T-1}\widetilde{B}^l_{k,\theta,\widetilde{\pi}^l_{k-1,\theta}}(u_{k,l},du_{k-1,l})$$

and:

$$\widetilde{B}^l_{k,\theta,\widetilde{\pi}^l_{k-1,\theta}}(u_{k,l},du_{k-1,l}) := \frac{\widetilde{\pi}^l_{k-1,\theta}(du_{k-1,l})\widetilde{G}^l_{k,\theta}(u_{k-1,l},u_{k,l})\widetilde{Q}^l_\theta(u_{k-1,l},u_{k,l})}{\widetilde{\pi}^l_{k-1,\theta}(\widetilde{G}^l_{k,\theta}(\cdot,u_{k,l})\widetilde{Q}^l_\theta(\cdot,u_{k,l}))}.$$

We remark that, due to the structure of the model:

$$\widetilde{B}^l_{k,\theta,\widetilde{\pi}^l_{k-1,\theta}}(u_{k,l},du_{k-1,l}) = \frac{\widetilde{\pi}^l_{k-1,\theta}(du_{k-1,l})\widetilde{G}^l_{k,\theta}(u_{k-1,l},u_{k,l})\hat{p}_\theta(x_k,x_{k+1})}{\int_{E_l}\widetilde{\pi}^l_{k-1,\theta}(du_{k-1,l})\widetilde{G}^l_{k,\theta}(u_{k-1,l},u_{k,l})\hat{p}_\theta(x_k,x_{k+1})}.$$

REMARK 4.4. *It is important to note that – in contrast to Remark 3.2 – there is no cancellation of terms of $\widetilde{G}^l_{k,\theta}(u_{k-1,l},u_{k,l})$ in the numerator and denominator of this backward kernel. This is precisely due to recursion (4.11) which leads to a path-dependence of the future coordinates of the discretized bridge on the terminal position x_{k+1}.*

4.4. Particle Approximation. Our online particle approximation of the gradient of the log-likelihood in (4.12), for a given $l\in\mathbb{N}_0$ is presented in Algorithm 4.2. Our estimates are given in (4.14) and (4.16) in Algorithm 4.2.

Algorithm 4.2 is simply the time-discretization of the procedure presented in Algorithm 4.1. A number of remarks are again of interest. Firstly, the cost of the algorithm per unit time is now $\mathcal{O}(N^2\Delta_l^{-1})$. The increase in computational cost over Algorithm 3.1 is the fact that when computing $\widetilde{\Lambda}^l_{k,\theta}(\breve{u}^j_{k-1,l},u^i_{k,l})$ in (4.15), one must solve the recursion (4.11) for each $(i,j)\in\{1,\dots,N\}^2$, which has a cost $\mathcal{O}(\Delta_l^{-1})$ and it is this cost that dominates. Secondly, following the discussion in Section 3.3, we have proved in a companion work that, under appropriate assumptions, the MSE for $(k,N)\in\mathbb{N}^2$:

$$\mathbb{E}_\theta\Big[\Big\|\nabla_\theta \widehat{\log(\widetilde{\gamma}^l_{k,\theta}}(1)) - \nabla_\theta \log(\gamma_{k,\theta}(1))\Big\|_2^2\Big] \leq C\Big(\frac{1}{N}+\Delta_l\Big), \tag{4.13}$$

for constant C that does not depend on N, l. To achieve an MSE of $\mathcal{O}(\epsilon^2)$ for some $\epsilon>0$ given, one sets $l=\mathcal{O}(|\log(\epsilon)|)$ and $N=\mathcal{O}(\epsilon^{-2})$. The cost per unit time of doing this, is then $\mathcal{O}(\epsilon^{-6})$. This is significantly worse than the approach in Algorithm 3.1, but we remark that when discussing the cost of Algorithm 3.1, in the bound (3.11), we have assumed that the constant C does not depend upon l. However, in a sequel work we will show that under assumptions that this afore-mentioned C explodes exponentially in l. Conversely, C in (4.13) can be proved to be independent of l, precisely due to the path-space development we have adopted. We remark, however, that one can use an MLMC method to reduce this cost of $\mathcal{O}(\epsilon^{-6})$ per unit time of Algorithm 4.2 and this algorithm is presented in the next section.

```
Algorithm 4.2 Modified Online Score Function Estimation for a given $l \in \mathbb{N}_0$.
  1. For $i\in\{1,\dots,N\}$, sample $u^i_{0,l}$ i.i.d. from $\widetilde{\eta}^l_{0,\theta}(\cdot)$. The estimate of $\nabla_\theta\log(\widetilde{\gamma}^l_{1,\theta}(1))$ is:
     (4.14)  $\nabla_\theta \widehat{\log(\widetilde{\gamma}^l_{1,\theta}}(1)) := \frac{\sum_{i=1}^N \widetilde{G}^l_{0,\theta}(x_*,u^i_{0,l})\widetilde{\Lambda}^l_{0,\theta}(x_*,u^i_{0,l})}{\sum_{i=1}^N \widetilde{G}^l_{0,\theta}(x_*,u^i_{0,l})}.$
     Set $k=1$ and for $i\in\{1,\dots,N\}$, $\breve{u}^i_{-1,l}=x_*$.
  2. For $i\in\{1,\dots,N\}$ sample $\breve{u}^i_{k-1,l}$ from:
     $\sum_{i=1}^N \frac{\widetilde{G}^l_{k-1,\theta}(\breve{u}^i_{k-2,l},u^i_{k-1,l})}{\sum_{j=1}^N \widetilde{G}^l_{k-1,\theta}(\breve{u}^j_{k-2,l},u^j_{k-1,l})}\delta_{\{u^i_{k-1,l}\}}(\cdot).$
     If $k=1$, for $i\in\{1,\dots,N\}$, set $\widetilde{F}^{l,N}_{k-1,\theta}(\breve{u}^i_{0,l})=\widetilde{\Lambda}^l_{0,\theta}(x_*,\breve{u}^i_{0,l})$.
  3. For $i\in\{1,\dots,N\}$, sample $u^i_{k,l}$ from $\widetilde{M}^l_\theta(\breve{u}^i_{k-1,l},\cdot)$. For $i\in\{1,\dots,N\}$, compute:
     (4.15)
     $\widetilde{F}^{l,N}_{k,\theta}(u^i_{k,l}) = \frac{\sum_{j=1}^N \widetilde{G}^l_{k,\theta}(\breve{u}^j_{k-1,l},u^i_{k,l})\hat{p}_\theta(\breve{x}^j_k,x^i_{k+1})\{\widetilde{F}^{l,N}_{k-1,\theta}(\breve{u}^j_{k-1,l})+\widetilde{\Lambda}^l_{k,\theta}(\breve{u}^j_{k-1,l},u^i_{k,l})\}}{\sum_{j=1}^N \widetilde{G}^l_{k,\theta}(\breve{u}^j_{k-1,l},u^i_{k,l})\hat{p}_\theta(\breve{x}^j_k,x^i_{k+1})}.$
     The estimate of $\nabla_\theta\log(\widetilde{\gamma}^l_{k+1,\theta}(1))$ is:
     (4.16)  $\nabla_\theta \widehat{\log(\widetilde{\gamma}^l_{k+1,\theta}}(1)) = \frac{\sum_{i=1}^N \widetilde{G}^l_{k,\theta}(\breve{u}^i_{k-1,l},u^i_{k,l})\widetilde{F}^{l,N}_{k,\theta}(u^i_{k,l})}{\sum_{i=1}^N \widetilde{G}^l_{k,\theta}(\breve{u}^i_{k-1,l},u^i_{k,l})}.$
     Set $k=k+1$ and return to the start of 2..
```

4.5. Multilevel Particle Filter. We now present a new multilevel particle filter along with online estimation of the score-function. We fix $l \in \mathbb{N}$ for now and for $(k, s) \in \mathbb{N}_0 \times \{l, l-1\}$ define:

$$u_{k,s} = (z_{k+\Delta_l,s}, z_{k+2\Delta_l,s}, \dots, z_{k+1-\Delta_l,s}, x_{k+1,s}) \in (\mathbb{R}^{d_x})^{\Delta_s^{-1}} = E_s.$$

We give the approach in Algorithm 4.3. Before explaining how one can use Algorithm 4.3 to provide online estimates of the score function, several remarks are required to continue. The first is related to the couplings mentioned in Algorithm 4.3 point 2. and point 3. bullet 1. The coupling in point 2, requires a way to resample the indices of the particles so that they have the correct marginals. This topic has been investigated considerably in the literature, see e.g. [19, 30], and techniques that have been adopted include sampling maximal coupling, e.g. [17], or using the $\mathbb{L}_2$-Wasserstein optimal coupling [2]; in general the latter is found to be better in terms of variance reduction, but can only be implemented when $d_x = 1$. We rely upon the maximal coupling in this paper, which has a cost of $\mathcal{O}(N)$ per unit time. For point 3. bullet 1, one again has a considerable degree of flexibility. In this article we sample the maximal coupling which can be achieved at a cost which is at most $\mathcal{O}(N)$ cost per-unit time using the algorithm of [33]. The second main remark of interest is that the basic filter that is sampled in Algorithm 4.3 is an entirely new coupled particle filter for diffusions (i.e. different to [17, 20]). The utility of the approach relative to [20] is of great interest, in the context of filtering.

Set $(l_*, L) \in \mathbb{N}^2$ with $l_* < L$. The idea is to run Algorithm 4.3, independently, for $l \in \{l_*, \dots, L\}$ each with N_l particles and, independently, Algorithm 4.2 for $l = l_* - 1$ with N_{l_*-1} particles. We then consider the estimate, for $k \in \mathbb{N}$

$$\nabla_\theta \widehat{\log(\widetilde{\gamma}^L_{k,\theta}}(1))_{ML} := \sum_{l=l_*}^{L} \Big\{ \nabla_\theta \widehat{\log(\widetilde{\gamma}^l_{k,\theta}}(1)) - \nabla_\theta \widehat{\log(\widetilde{\gamma}^{l-1}_{k,\theta}}(1)) \Big\} + \nabla_\theta \widehat{\log(\widetilde{\gamma}^{l_*-1}_{k,\theta}}(1)),$$

where the summands on the right hand side are defined in (4.17) and (4.18) and the last term on the right hand side is as either (4.14) or (4.16) (depending on k). Now, we show in an on-going companion work, under appropriate assumptions, one has the following result for $(k, N_{l_*-1:L}) \in \mathbb{N}^{L-l_*+2}$:

$$\mathbb{E}_\theta\Big[\Big\| \nabla_\theta \widehat{\log(\widetilde{\gamma}^L_{k,\theta}}(1))_{ML} - \nabla_\theta \log(\gamma_{k,\theta}(1)) \Big\|_2^2 \Big] \leq C\Big(\sum_{l=l_*-1}^{L} \frac{\Delta_l^\beta}{N_l} + \Delta_L \Big), \tag{4.19}$$

for constant C that does not depend on N, l; also, $\beta = 1$ if σ is a constant function, else $\beta = 1/2$. Choose: i) L so that $\Delta_L = \mathcal{O}(\epsilon^2)$, for $\epsilon > 0$ given; ii) if $\beta = 1$, $N_l = \mathcal{O}(\epsilon^{-2}\Delta_l^{1/2+\rho})$ for some $0 < \rho < 1/2$. These selections yield an MSE of $\mathcal{O}(\epsilon^2)$ for a cost of $\mathcal{O}(\epsilon^{-4})$. If $\beta = 1/2$, one can set $N_l = \mathcal{O}(\epsilon^{-2}\Delta_l^{1/2+\rho}\Delta_L^{-\rho})$ for some $\rho > 0$. This will yield an MSE of $\mathcal{O}(\epsilon^2)$ for a cost of $\mathcal{O}(\epsilon^{-4(1+\rho)})$. Such results are at least as good as the method in Section 3.3, assuming that latter approach does not collapse with l.

We remark that it is possible to produce an almost-surely unbiased estimator of the score function, when θ is the true parameter, using a combination of the multilevel method that has been developed here and the approach in [18]. This is left for future work.

5. Numerical Results. In this section, we consider four models to investigate the various properties of our algorithms. The score function is estimated using both Algorithms 3.1 and 4.2 for a fixed θ. We will show, as expected, that they are equivalent for a large number of particles N and

Algorithm 4.3 Coupled Online Score Function Estimation for a given $l \in \mathbb{N}$.

```
1.  • For $i \in \{1,\dots,N\}$, sample $u_{0,l}^i$ i.i.d. from $\widetilde{\eta}_{0,\theta}^l(\cdot)$.
    • For $(i,p) \in \{1,\dots,N\} \times \{1,\dots,\Delta_{l-1}^{-1}-1\}$, set $z_{p\Delta_{l-1},l-1}^i = z_{p\Delta_{l-1},l}^i + z_{p\Delta_{l-1}-\Delta_l,l}^i$ and $x_{1,l-1}^i = x_{1,l}^i$.
    The estimate of $\nabla_\theta \log(\widetilde{\gamma}_{1,\theta}^l(1)) - \nabla_\theta \log(\widetilde{\gamma}_{1,\theta}^{l-1}(1))$ is:
    $$\nabla_\theta \widehat{\log(\widetilde{\gamma}_{1,\theta}^l}(1)) - \nabla_\theta \widehat{\log(\widetilde{\gamma}_{1,\theta}^{l-1}}(1)) :=$$
    $$\frac{\sum_{i=1}^N \widetilde{G}_{0,\theta}^l(x_*, u_{0,l}^i)\widetilde{\Lambda}_{0,\theta}^l(x_*, u_{0,l}^i)}{\sum_{i=1}^N \widetilde{G}_{0,\theta}^l(x_*, u_{0,l}^i)} - \frac{\sum_{i=1}^N \widetilde{G}_{0,\theta}^{l-1}(x_*, u_{0,l-1}^i)\widetilde{\Lambda}_{0,\theta}^{l-1}(x_*, u_{0,l-1}^i)}{\sum_{i=1}^N \widetilde{G}_{0,\theta}^{l-1}(x_*, u_{0,l-1}^i)}. \tag{4.17}$$
    Set $k=1$ and for $i \in \{1,\dots,N\}$, $\check{u}_{-1,l}^i = \check{u}_{-1,l-1}^i = x_*$.
2.  For $i \in \{1,\dots,N\}$, sample $(\alpha_l(i), \alpha_{l-1}(i)) \in \{1,\dots,N\}^2$ from a coupling of:
    $$\sum_{i=1}^N \frac{\widetilde{G}_{k-1,\theta}^l(\check{u}_{k-2,l}^{\alpha_l(i)}, u_{k-1,l}^{\alpha_l(i)})}{\sum_{j=1}^N \widetilde{G}_{k-1,\theta}^l(\check{u}_{k-2,l}^j, u_{k-1,l}^j)} \quad \text{and} \quad \sum_{i=1}^N \frac{\widetilde{G}_{k-1,\theta}^{l-1}(\check{u}_{k-2,l-1}^{\alpha_{l-1}(i)}, u_{k-1,l}^{\alpha_{l-1}(i)})}{\sum_{j=1}^N \widetilde{G}_{k-1,\theta}^{l-1}(\check{u}_{k-2,l-1}^j, u_{k-1,l-1}^j)},$$
    and set $(\check{u}_{k-1,l}^i, \check{u}_{k-1,l-1}^i) = (u_{k-1,l}^{\alpha_l(i)}, u_{k-1,l-1}^{\alpha_{l-1}(i)})$. If $k=1$, for $i \in \{1,\dots,N\}$, $s \in \{l, l-1\}$, set $\widetilde{F}_{k-1,\theta}^{s,N}(\check{u}_{0,s}^i) = \widetilde{\Lambda}_{0,\theta}^s(x_*, \check{u}_{0,s}^i)$.
3.  • For $i \in \{1,\dots,N\}$, sample $(x_{k+1,l}^i, x_{k+1,l-1}^i)$ from a coupling of $\hat{p}_\theta(\check{x}_{k,l}^i, \cdot)$ and $\hat{p}_\theta(\check{x}_{k,l-1}^i, \cdot)$.
    • For $i \in \{1,\dots,N\}$ sample $z_{k+\Delta_l,l}^i, \dots, z_{k+1-\Delta_l,l}^i$ i.i.d. from $\prod_{s=1}^{\Delta_l^{-1}-1} \phi_l(\,\cdot\,)$.
    • For $(i,p) \in \{1,\dots,N\} \times \{1,\dots,\Delta_{l-1}^{-1}-1\}$ set $z_{k+p\Delta_{l-1},l-1}^i = z_{k+p\Delta_{l-1},l}^i + z_{k+p\Delta_{l-1}-\Delta_l,l}^i$.
    • For $(i,s) \in \{1,\dots,N\} \times \{l.l-1\}$, compute:
    $$\widetilde{F}_{k,\theta}^{s,N}(u_{k,l}^i) = \frac{\sum_{j=1}^N \widetilde{G}_{k,\theta}^s(\check{u}_{k-1,s}^j, u_{k,s}^i)\hat{p}_\theta(\check{x}_{k,s}^j, x_{k+1,s}^i)\{\widetilde{F}_{k-1,\theta}^{s,N}(\check{u}_{k-1,s}^j) + \widetilde{\Lambda}_{k,\theta}^s(\check{u}_{k-1,s}^j, u_{k,s}^i)\}}{\sum_{j=1}^N \widetilde{G}_{k,\theta}^s(\check{u}_{k-1,s}^j, u_{k,s}^i)\hat{p}_\theta(\check{x}_{k,s}^j, x_{k+1,s}^i)}.$$
    The estimate of $\nabla_\theta \log(\widetilde{\gamma}_{k+1,\theta}^l(1)) - \nabla_\theta \log(\widetilde{\gamma}_{k+1,\theta}^{l-1}(1))$ is:
    $$\nabla_\theta \widehat{\log(\widetilde{\gamma}_{k+1,\theta}^l}(1)) - \nabla_\theta \widehat{\log(\widetilde{\gamma}_{k+1,\theta}^{l-1}}(1)) :=$$
    $$\frac{\sum_{i=1}^N \widetilde{G}_{k,\theta}^l(\check{u}_{k-1,l}^i, u_{k,l}^i)\widetilde{F}_{k,\theta}^{l,N}(u_{k,l}^i)}{\sum_{i=1}^N \widetilde{G}_{k,\theta}^l(\check{u}_{k-1,l}^i, u_{k,l}^i)} - \frac{\sum_{i=1}^N \widetilde{G}_{k,\theta}^{l-1}(\check{u}_{k-1,l-1}^i, u_{k,l-1}^i)\widetilde{F}_{k,\theta}^{l-1,N}(u_{k,l-1}^i)}{\sum_{i=1}^N \widetilde{G}_{k,\theta}^{l-1}(\check{u}_{k-1,l-1}^i, u_{k,l-1}^i)}. \tag{4.18}$$
    Set $k = k+1$ and return to the start of 2..
```

a high level of descritization l. We then compare the cost of Algorithm 4.2 and its multilevel version Algorithm 4.3. As an application of our methods, we use Algorithms 3.1 and 4.3 for parameter estimation via stochastic gradient. The code is written in MATLAB and it can be downloaded from https://github.com/ruzayqat/score_based_par_est.

We remark that we will not use Algorithm 4.2 for parameter estimation because it is 'slow' compared to the algorithms as illustrated in the previous section and in Figure 1. In Figure 1 we consider the Model 1, as described in the next section, with $T = 40$, $l = 10$, $\theta = (-0.4, -0.5)$, $\kappa = 2$ and $x_* = 0.2$. Figure 1 provides a comparison between the cost of Algorithms 3.1 and 4.2 , which is the average machine time measured in seconds needed per each simulation, versus the number of particles N. As predicted by our theoretical conjectures, we see that the cost of Algorithm 3.1 is significantly lower than that of Algorithm 4.2.

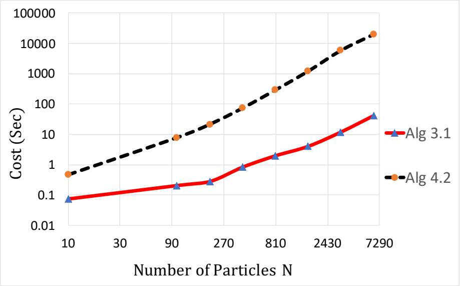

Figure 1: Comparison between the cost of Algorithms 3.1 and 4.2 per each simulation versus the number of particles N. We run both algorithms on Model 1.

5.1. Models.

In the following, parameters (κ, σ) are fixed.

Model 1: Let $d_x = d_y = 1$, $d_\theta = 2$ and consider the following linear SDEs:

$$\begin{aligned} dX_t &= \theta_1 X_t dt + \sigma dW_t; \\ dY_t &= \theta_2(\kappa - X_t)dt + dB_t. \end{aligned}$$

Model 2: Let $d_x = d_y = 1$, $d_\theta = 3$ and consider a nonlinear diffusion process along with a linear diffusion process of observations:

$$\begin{aligned} dX_t &= \big(\tfrac{\theta_1}{X_t} + \theta_2 X_t\big)dt + \sigma dW_t; \\ dY_t &= \theta_3(\kappa - X_t)dt + dB_t. \end{aligned}$$

Model 3: Let $d_x = d_y = 1$, $d_\theta = 3$ and consider a nonlinear signal along with a nonlinear diffusion process of observations. The first SDE is a Cox-Ingersoll-Ross process after an 1-1 transform. Thus:

$$\begin{aligned} dX_t &= \frac{1}{2}\big(\tfrac{\theta_1\theta_2 - \sigma^2}{X_t} - \theta_2 X_t\big)dt + \sigma dW_t; \\ dY_t &= \theta_3(\kappa - X_t^2)dt + dB_t. \end{aligned}$$

This model has a solution if and only if $\theta_1\theta_2 > 2\sigma^2$.

Model 4: Let $d_x = d_y = 1$, $d_\theta = 3$ and consider a type of Black-Scholes model with a stochastic volatility:

$$\begin{aligned} dX_t &= \theta_1(\theta_2 - X_t)dt + \sigma(X_t)dW_t; \\ dY_t &= (\theta_3 - \frac{1}{2}X_t^2)dt + dB_t. \end{aligned}$$

where $\sigma(X_t) = \beta/\sqrt{X_t^2+1}$ and β is fixed. In the hidden process, θ_1 and θ_2 are the speed and level of mean reversion and θ_3 is a mean type level for the observation process, We will apply our methodology (see Figure 8 later on) on the log mid-price of Tesla Inc. stock in 2018. The dataset shown in Figure 2 represents the log mid-price at every second during a trading day for a total of 250 trading days.

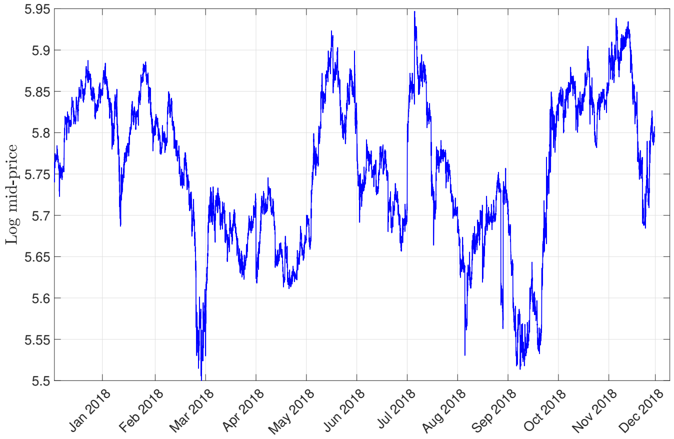

Figure 2: The log of mid-price for every second of Tesla stock in 2018. The dataset contains 5.85×10^6 data points.

5.2. Simulation Results. In all our results data are generated from the model under the finest discretization considered except in model 4, where we use a real data. In Algorithm 4.2, we consider the auxiliary linear process following:

$$d\tilde{X}_t = \sigma dW_t$$

in models 1-3 and in model 4 it follows:

$$d\tilde{X}_t = \sigma(x')dW_t$$

In models 1-3, $\tilde{p}_{\theta,x'}(x,t;x',1) = \mathcal{N}(x';x,(1-t)\sigma^2)$, hence $\tilde{p}_{\theta,x'}(x,x') = \mathcal{N}(x';x,\sigma^2)$, which is easy to sample x' from, and therefore, $\hat{p}_\theta(x,x') = \tilde{p}_{\theta,x'}(x,x')$. But in model 4, $\tilde{p}_{\theta,x'}(x,x') = \mathcal{N}(x';x,\sigma^2(x'))$ which is not easy to sample x' from. Therefore, we take $\hat{p}_\theta(x,x') = \mathcal{N}(x';x,\sigma^2(x))$.

5.2.1. Estimation of the Score Function. For each model, we fix parameter θ and estimate the score function using Algorithms 3.1 and 4.2. In Algorithm 3.1, $N \in \{3000, 7000, 4000, 5000\}$ in the 1st, 2nd, 3rd & 4th models, respectively. In Algorithm 4.2, $N \in \{1000, 2000, 1000, 1500\}$ in the 1st, 2nd, 3rd & 4th models, respectively. In both algorithms, we set the discretization level to $l = 10$. In Models 1, 2 and 3, we set $\kappa = 2,\ 2.2,\ 1.5$, $x_* = 0.2,\ 1,\ 2$ and $\sigma = 0.3,\ 0.25,\ 0.25$, respectively. While in model 4, we set $x_* = 1.3$ and $\beta = 2$; $T = 50$ for all 4 models ($T = 50$ in model 4 corresponds to 14.22 hours of trading).

Figure 3 summarizes the results of 56 replications of estimates of the score function for each model and for each unit time point. These simulations are implemented in parallel using 8 CPUs. The figure illustrates that both algorithms are equivalent for large N and l as one would expect.

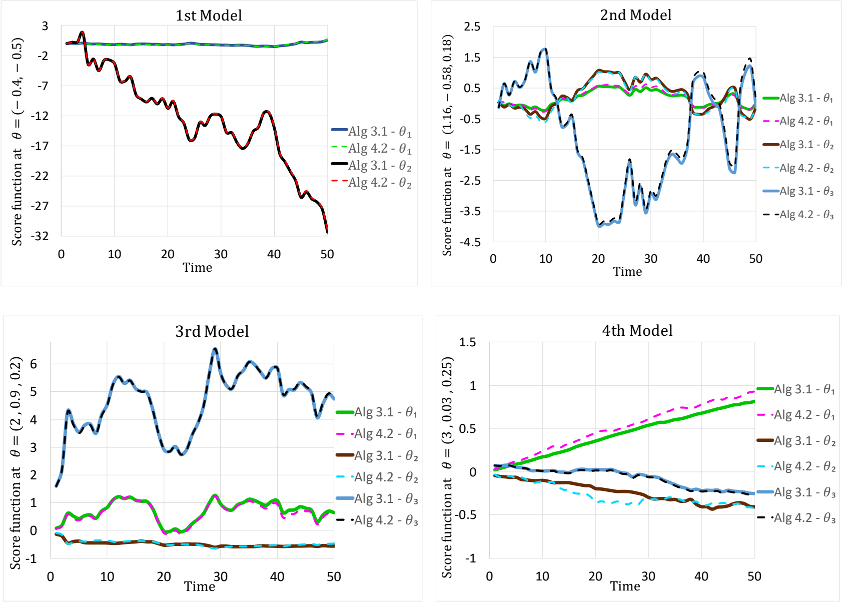

Figure 3: Trajectories from the execution of Algorithms 3.1 and 4.2 for the estimation of the score function in Models 1-4.

5.2.2. Cost Comparison. We now consider comparing the costs of Algorithms 4.2 and 4.3. We take l_* to be 7 in the models 1, 2 & 3 and 8 in the 4th model. The parameters of the model are as in the previous section. The ground truth is computed at level 11 with $N = 2000$ using Algorithm 4.2. We run 56 simulations of both algorithms for $L \in \{l_* - 1, \cdots, 10\}$. For each L, the number of particles are carefully chosen to give similar MSE values from both algorithms. Particularly, the number of particles in Algorithm 4.2 is $N_L = \lfloor C_1 2^L \rfloor$ and for each level l in Algorithm 4.2. In Figure 4 the number of particles is $N_l = \left\lfloor C_2 2^L (L - l_* + 2)\Delta_l^{1/2+\rho} \right\rfloor$ (in models 1 to 3) and $N_l = \left\lfloor C_2 2^L (L - l_* + 2)\Delta_l^{1/2+\rho}\Delta_L^{-\rho} \right\rfloor$ (in model 4), where C_1 and C_2 are constants. In we can observe the cost against MSE curve, that appear to follow our conjectures over algorithmic costs earlier in the article.

5.2.3. Parameter Estimation. We use Algorithms 3.1, 4.3 to estimate the parameters in each model. In Algorithm 3.1, the level of discretization, l, is 10 for models 1-3 and 9 for model 4, and the number of particles, N, is 2,000 for models 1-3 and 2500 for model 4. In Algorithm 4.3, we use $l_* = 7$, $L = 10$ and the number of particles on each level $l \in \{l_* - 1, \cdots, L\}$ is $N_l = 2^L(L - l_* + 2)\Delta_l^{1/2+\rho}$, where $\rho \in \{0.14, 0.09, 0.11\}$ in Models 1, 2 and 3, respectively. In model 4, $l_* = 8$, $L = 9$ and the number of particles on each level $l \in \{l_* - 1, \cdots, L\}$ is $N_l = 1.4 \times 2^L(L - l_* + 2)\Delta_l^{1/2+\rho}\Delta_L^{-\rho}$ where $\rho = 0.1$.

Figure 5 considers Model 1. We fix $x_* = 0.2$, $\sigma = 0.3$, $\kappa = 2$, $T = 20,000$. The parameter values used to generate the data are $(\theta_1^\star, \theta_2^\star) = (-0.7, -0.5)$. For the stochastic gradient algorithm,

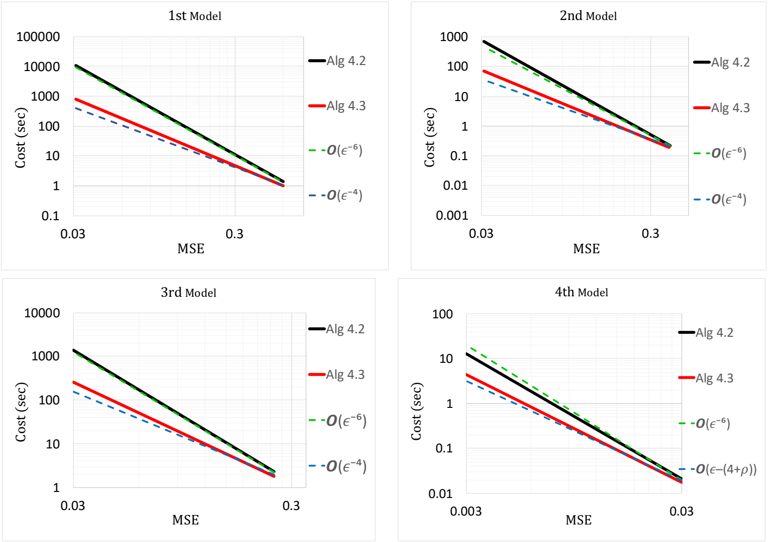

Figure 4: Cost per each simulation versus MSE on a log-log scale for Algorithms 4.2 and 4.3. The dashed lines are for reference.

we used an initial value $(-0.05, -1.5)$ and step-size $\alpha_k = k^{-0.85}$. Figure 6 considers Model 2. We fix $x_* = 1.8$, $\sigma = 0.25$, $\kappa = 2.2$, $T = 20,000$. The parameter values used to generate the data are $(\theta_1^\star, \theta_2^\star, \theta_3^\star) = (1.3, -0.5, 0.18)$. For the stochastic gradient algorithm, we used initial value $(0.8, -1, 0.8)$ and step-size $\alpha_k = k^{-0.95}$. Figure 7 considers Model 3. We fix $x_* = 1.5$, $\sigma = 0.25$, $\kappa = 2$, $T = 20,000$. The parameter values used to generate the data are $(\theta_1^\star, \theta_2^\star, \theta_3^\star) = (2, 1, 0.45)$. For the stochastic gradient algorithm, we used an initial value $(1.24, 0.6, 1.11)$ and step size $\alpha_k = k^{-0.9}$. Figure 8 considers Model 4 applied to the data in Figure 2. We fix $x_* = 1.3$, $\beta = 2$, $T = 11425$ (there is a rescaling of the time parameter). For the stochastic gradient algorithm, we used an initial value $(2.4, 0.5, 0.4)$ and step size $\alpha_k = k^{-0.82}$. In all cases considered (Figures 5-8) our selected settings allow for an accurate estimation of the parameter values over long time periods.

Acknowledgements. AJ & HR were supported by KAUST baseline funding. AB acknowledges support from a Leverhulme Trust Prize. DC was partially supported by EU Synergy project STUOD - DLV-856408. NK acknowledges funding by a JP Morgan A.I. Faculty award. We thank two referees and an associate editor for their comments which have greatly enhanced the article.

REFERENCES

[1] Bain, A. & Crisan, D. (2009). *Fundamentals of Stochastic Filtering.* Springer: New York.

[2] Ballesio, M., Jasra, A., von Schwerin, E. & Tempone, R. (2020). A Wasserstein coupled particle filter for multilevel estimation. arXiv:2004.03981.

[3] Benveniste, A., Métivier, M. & Priouret, P. (1990). *Adaptive Algorithms and Stochastic Approximation.* New York: Springer-Verlag.

[4] Beskos, A., Papaspiliopoulos, O., Roberts, G., Fearnhead, P. (2006). Exact and computationally efficient likelihood-based estimation for discretely observed diffusion processes (with discussion). *J. R. Statist. Soc.*

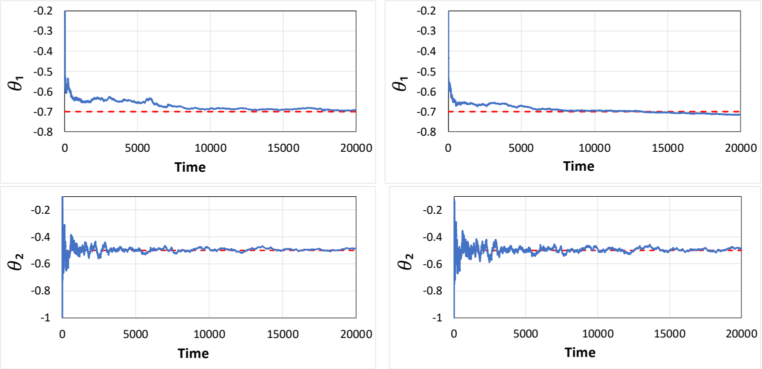

Figure 5: Trajectories from the execution of Algorithm 3.1 (left panel) and Algorithm 4.3 (right panel) for the estimation of (θ_1, θ_2) from Model 1. The horizontal dashed lines in the plots show the true parameter values $(\theta_1^\star, \theta_2^\star) = (-0.7, -0.5)$.

Ser. B, **68**, 333-382.

[5] Beskos, A., Crisan, D., Jasra, A., Kantas, N. & Ruzayqat, H. (2020). Score-Based Parameter Estimation for a Class of Continuous-Time State Space Models. arXiv:2008.07803v1

[6] Bierkens, J., Van Der Meulen, F., Schauer, M. (2020). Simulation of elliptic and hypo-elliptic conditional diffusions, *Advances in Applied Probability*, **52**, 173-212.

[7] Botha, I., Kohn, R., & Drovandi, C. (2020). Particle methods for stochastic differential equation mixed effects models. *Bayes. Anal.* (to appear).

[8] Campillo, F. & Le Gland, F. (1989). Maximum likelihood estimation for partially observed diffusions: Direct Maximization vs The EM algorithm. *Stoch. Proc. Appl.*, **33**, 245–274.

[9] Cliffe, K. A., Giles, M. B., Scheichl, R., & Teckentrup, A. L. (2011). Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients. *Comput. Vis. Sci.*, **14**, 3–15.

[10] Crisan, D. (2011). Discretizing the continuous-time filtering problem: Order of Convergence. In [11], 572–597.

[11] Crisan, D., & Rozovskii, B. (2011). *The Oxford Handbook of Nonlinear Filtering.* Oxford University Press.

[12] Del Moral, P. (2004). *Feynman-Kac Formulae.* Springer.

[13] Del Moral, P., Doucet, A., & Singh S. S. (2010). A backward particle interpretation of Feynman-Kac formuale. *M2AN*, **44**, 947–975.

[14] Del Moral, P., Doucet, A., & Singh S. S. (2010). Forward smoothing using sequential Monte Carlo, arXiv:1012.5390

[15] Etienne, M. P., Gloaguen, P., Corff, S. L., & Olsson, J. (2020). Backward importance sampling for partially observed diffusion processes. arXiv:2002.05438.

[16] Gloaguen, P., Etienne, M. P. & Le Corff, S. (2018). Online sequential Monte Carlo smoother for partially observed diffusion processes. *EURASIP J. Adv. Sig. Proc*, article 9.

[17] Jasra, A., Kamatani, K., Law K. J. H. & Zhou, Y. (2017). Multilevel particle filters. *SIAM J. Numer. Anal.*, **55**, 3068-3096.

[18] Jasra, A., Law, K. J. H., & Yu, F. (2020). Unbiased filtering of a class of partially observed diffusions. arXiv:2002.03747.

[19] Jasra, A., & Yu, F. (2020). Central limit theorems for coupled particle filters. *Adv. Appl. Probab.* **52**, 942–1001.

[20] Jasra, A., Yu, F. & Heng, J. (2020). Multilevel particle filters for the nonlinear filtering problem in continuous time. *Stat. Comp.* **30**, 1381–1402.

[21] Le Gland, F. and Mevel, M. (1997). Recursive identification in hidden Markov models. *Proc. 36th IEEE Conf. Dec. Contr.*, 3468-3473.

[22] Mider, M., Schauer, M. & van der Meulen, F. (2020). Continuous-discrete smoothing of diffusions. arXiv:1712.03807.

[23] Olsson, J. & Westerborn, J. (2017). Efficient particle-based online smoothing in general hidden Markov

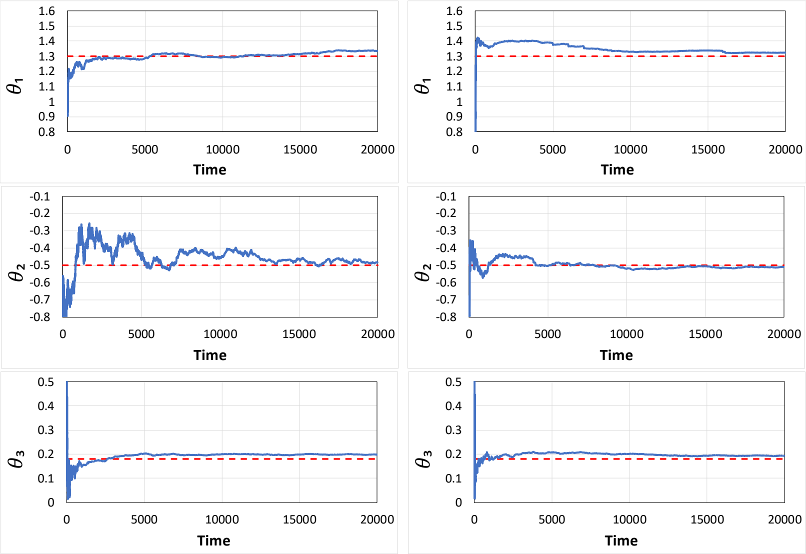

Figure 6: Trajectories from the execution of Algorithm 3.1 (left panel) and Algorithm 4.3 (right panel) for the estimation of $(\theta_1, \theta_2, \theta_3)$ of the second model. We used an initial value $(0.8, -1, 0.8)$. The horizontal dashed lines in the plots show the true parameter values $(\theta_1^\star, \theta_2^\star, \theta_3^\star) = (1.3, -0.5, 0.18)$.

models: The PaRIS algorithm. *Bernoulli*, **23**, 1951-1996.

[24] Papaspiliopoulos, O., Roberts, G. O., & Stramer, O. (2013). Data augmentation for diffusions. *J. Comp. Graph. Stat.*, **22**, 665-688.

[25] Papaspiliopoulos, O. & Roberts, G. (2012). Importance sampling techniques for estimation of diffusion models. *Stat. Meth. Stoch. Diff. Eq.*, **124**, 311-340.

[26] Picard, J. (1984). Approximation of nonlinear filtering problems and order of convergence. In *Filtering and control of random processes*, 219-236, Springer, Berlin, Heidelberg.

[27] Poyiadjis, G., Doucet, A., & Singh, S. S. (2011). Particle approximations of the score and observed information matrix in state space models with application to parameter estimation. *Biometrika*, **98**, 65-80.

[28] Särkkä, S., & Sottinen, T. (2008). Application of Girsanov theorem to particle filtering of discretely observed continuous-time non-linear systems. *Bayes. Anal.*, **3**, 555-584.

[29] Schauer, M., van der Meulen, F. & van Zanten, H. (2017). Guided proposals for simulating multi-dimensional diffusion bridges. *Bernoulli*, **23**, 2917–2950.

[30] Sen, D., Thiery, A., Jasra, A. (2018). On coupling particle filters. *Statist. Comp.*, **28**, 461-475.

[31] Surace, S. C., & Pfister, J. P. (2018). Online Maximum-Likelihood Estimation of the Parameters of Partially Observed Diffusion Processes. *IEEE Transactions on Automatic Control*, 64(7), 2814-2829.

[32] Talay, D. (1984). Efficient numerical schemes for the approximation of expectations of functionals of the solution of a SDE, and applications. In *Filtering and control of random processes*, 294-313, Springer, Berlin, Heidelberg.

[33] Thorisson, H. (2000). *Coupling, stationarity, and regeneration.* Springer:New York.

[34] van der Meulen, F., & Schauer, M. (2017). Bayesian estimation of discretely observed multi-dimensional diffusion processes using guided proposals. *Elec. J. Stat.*, **11**, 2358-2396.

[35] Whitaker, G. A., Golightly, A., Boys, R. J., & Sherlock, C. (2017). Improved bridge constructs for stochastic differential equations. *Stat. Comp.*, **27**, 885-900.

[36] Yonekura, S. & Beskos, A. (2020). Online smoothing for diffusion processes observed with noise. arXiv: 200312247.

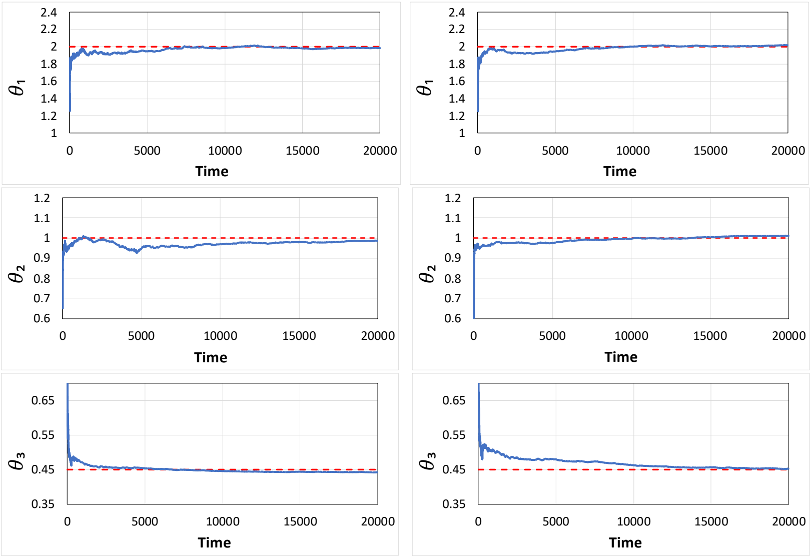

Figure 7: Trajectories from the execution of Algorithm 3.1 (left panel) and Algorithm 4.3 (right panel) for the estimation of $(\theta_1, \theta_2, \theta_3)$ from Model 3. The horizontal dashed lines in the plots show the true parameter values $(\theta_1^\star, \theta_2^\star, \theta_3^\star) = (2, 1, 0.45)$.

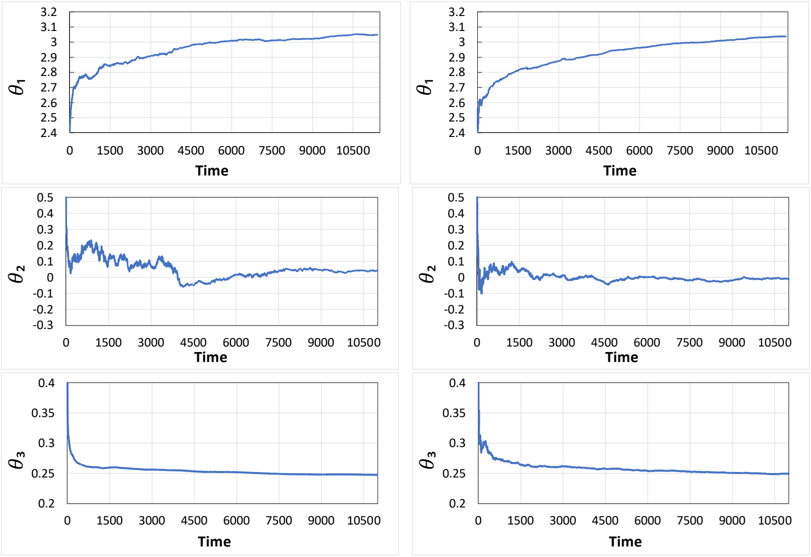

Figure 8: Trajectories from the execution of Algorithm 3.1 (left panel) and Algorithm 4.3 (right panel) for the estimation of $(\theta_1, \theta_2, \theta_3)$ from Model 4.