Online Smoothing for Diffusion Processes Observed with Noise

Shouto Yonekura*

Department of Statistical Science, University College London,

Alan Turing Institute,

and

Alexandros Beskos†

Department of Statistical Science, University College London,

Alan Turing Institute,

August 12, 2021

Abstract

We introduce a methodology for online estimation of smoothing expectations for a class of additive functionals, in the context of a rich family of diffusion processes (that may include jumps) – observed at discrete-time instances. We overcome the unavailability of the transition density of the underlying SDE by working on the augmented pathspace. The new method can be applied, for instance, to carry out online parameter inference for the designated class of models. Algorithms defined on the infinite-dimensional pathspace have been developed the last years mainly in the context of MCMC techniques. There, the main benefit is the achievement of mesh-free mixing times for the practical time-discretised algorithm used on a PC. Our own methodology sets up the framework for infinite-dimensional online filtering – an important positive practical consequence is the construct of estimates with variance that does not increase with decreasing mesh-size. Besides regularity conditions, our method is, in principle, applicable under the weak assumption – relatively to restrictive conditions often required in the MCMC or filtering literature of methods defined on pathspace – that the SDE covariance matrix is invertible.

Keywords: Jump Diffusion; Data Augmentation; Forward-Only Smoothing; Sequential Monte Carlo; Online Parameter Estimation.

1 Introduction

Research in Hidden Markov Models (HMMs) has – thus far – provided effective online algorithms for the estimation of expectations of the smoothing distribution for the case of a class of additive functionals of the underlying signal. Such methods necessitate knowledge of the transition density of the Markovian part of the model between observation times. We carry out a related exploration for the (common in applications) case when the signal corresponds to a diffusion process, thus we are faced with the challenge that such transition densities are typically unavailable. Standard data augmentation schemes that work with the multivariate density of a large enough number of imputed points of the continuous-time signal will lead to ineffective algorithms. The latter will have the abnormal characteristic that – for given Monte-Carlo iterates – the variability of the produced estimates will increase rapidly as the resolution of the imputation becomes finer. One of the ideas underpinning the work in this paper is that development of effective algorithms instead requires respecting the structural properties of the diffusion process, thus we build up imputation schemes on the infinite-dimensional diffusion pathspace itself. As a consequence, the time-discretised algorithm used in practice on a PC will be stable under mesh-refinement.

We consider continuous-time jump-diffusion models observed at discrete-time instances. The d_x-dimensional process, $X = \{X_t; t \geq 0\}$, $d_x \geq 1$, is defined via the following time-homogeneous stochastic differential equation (SDE), with $X_{t-} := \lim_{s \uparrow t} X_t$,

$$dX_t = b(X_{t-})dt + \sigma(X_{t-})dW_t + dJ_t, \quad X_0 = x_0 \in \mathbb{R}^{d_x}, \;\; t \geq 0. \tag{1.1}$$

Solution X is driven by the d_w-dimensional Brownian motion, $W = \{W_t; t \geq 0\}$, $d_w \geq 1$, and the compound Poisson process, $J = \{J_t; t \geq 0\}$. The SDE involves a drift function $b = b_\theta : \mathbb{R}^{d_x} \mapsto \mathbb{R}^{d_x}$ and coefficient matrix $\sigma = \sigma_\theta : \mathbb{R}^{d_x} \mapsto \mathbb{R}^{d_x \times d_w}$, for parameter $\theta \in \mathbb{R}^p$, $p \geq 1$. Let $\{N_t; t \geq 0\}$ be a Poisson process with intensity function $\lambda_\theta(\cdot)$, and $\{\xi_k\}_{k \geq 1}$ i.i.d. sequence of random variables with Lebesgue density $h_\theta(\cdot)$; the càdlàg process J is determined as $J_t = J_{t,\theta} = \sum_{i=1}^{N_t} \xi_i$. We work under standard assumptions (e.g. linear growth, Lipschitz continuity for b, σ) that guarantee a unique global solution of (1.1), in a

2

weak or strong sense, see e.g. Øksendal and Sulem (2007).

SDE (1.1) is observed with noise at discrete-time instances $0 = t_0 < t_1 < t_2 < \cdots < t_n$, $n \geq 1$. Without loss of generality, we assume equidistant observation times, with $\Delta := t_1 - t_0$. We consider data $Y_{t_0}, \ldots, Y_{t_n}$, and for simplicity we set,

$$x_i := X_{t_i}, \quad y_i = Y_{t_i}, \quad 0 \leq i \leq n.$$

Let $\mathcal{F}_i$ be a filtration generated by X_s for $s \in [t_{i-1}, t_i], 0 < i \leq n$. We assume,

$$\big[Y_{t_i} \,\big|\, \{Y_{t_j}; j < i\}, \{X_s; s \in [0, t_i]\} \big] \sim g_\theta\big(dY_{t_i} \,\big|\, Y_{t_{i-1}}, \mathcal{F}_i\big), \quad 0 \leq i \leq n, \qquad (1.2)$$

for conditional distribution $g_\theta(\cdot | Y_{t_{i-1}}, \mathcal{F}_i)$ on $\mathbb{R}^{d_y}$, $d_y \geq 1$, under convention $Y_{t_{-1}} = y_{-1} = \emptyset$. We write,

$$[\, x_i \,|\, x_{i-1} \,] \sim f_\theta(dx_i | x_{i-1}), \qquad (1.3)$$

where $f_\theta(dx_i | x_{i-1})$ is the transition distribution of the driving SDE process (1.1). We consider the density functions of $g_\theta(dy_i | y_{i-1}, \mathcal{F}_i)$ and $f_\theta(dx_i | x_{i-1})$, and – with some abuse of notation – we write $g_\theta(dy_i | y_{i-1}, \mathcal{F}_i) = g_\theta(y_i | y_{i-1}, \mathcal{F}_i) dy_i$, $f_\theta(dx_i | x_{i-1}) = f_\theta(x_i | x_{i-1}) dx_i$, where dy_i, dx_i denote Lebesgue measures. Our work develops under the following regime.

Assumption 1. *The transition density $f_\theta(x'|x)$ is intractable; the density $g_\theta(y'|y, \mathcal{F})$ is analytically available – for appropriate x', x, y', y, $\mathcal{F}$ consistent with preceding definitions.*

The intractability of the transition density $f_\theta(\cdot | \cdot)$ will pose challenges for the main problems this paper aims to address.

Models defined via (1.1)-(1.3) are extensively used, e.g., in finance and econometrics, for instance for capturing the market microstructure noise, see Aït-Sahalia et al. (2005); Aït-Sahalia and Yu (2008); Hansen and Lunde (2006). The above setting belongs to the general class of HMMs, with a signal defined in continuous-time. See Cappé et al. (2005); Douc et al. (2014) for a general treatment of HMMs fully specified in discrete-time. A number of methods have been suggested in the literature for approximating the unavailable transition

density – mainly in the case of processes without jumps – including: asymptotic expansion techniques (Aït-Sahalia et al., 2005; Aït-Sahalia, 2002, 2008; Kessler, 1997); martingale estimating functions (Kessler and Sørensen, 1999); generalized method of moments (Hansen and Scheinkman, 1993); Monte-Carlo approaches (Wagner, 1989; Durham and Gallant, 2002; Beskos et al., 2006). See, e.g., Kessler et al. (2012) for a detailed review.

For a given sequence $\{a_m\}_m$, we use the notation $a_{i:j} := (a_i, \ldots, a_j)$, for integers $i \leq j$. Let $p_\theta(y_{0:n})$ denote the joint density of $y_{0:n}$. Throughout the paper, $p_\theta(\cdot)$ is used generically to represent probability distributions or densities of random variables appearing as its arguments.

Consider the maximum likelihood estimator (MLE),

$$\hat{\theta}_n := \arg\max_{\theta \in \Theta} \log p_\theta(y_{0:n}).$$

Except for limited cases, one cannot obtain the MLE analytically for HMMs (even for discrete-time signal) due to the intractability of $p_\theta(y_{0:n})$.

We have set up the modelling context for this work. The main contributions of the paper in this setting – several of which relate with overcoming the intractability of the transition density of the SDE, and developing a methodology that is well-posed on the infinite-dimensional pathspace – will be as follows:

(i) We present an *online* algorithm that delivers Monte-Carlo estimators of smoothing expectations,

$$\mathcal{S}_{\theta,n} = \int S_\theta(\mathbf{x}_{0:n}) p_\theta(d\mathbf{x}_{0:n}|y_{0:n}), \quad n \geq 1, \tag{1.4}$$

for the class of additive functionals $S_\theta(\cdot)$ of the structure,

$$S_\theta(\mathbf{x}_{0:n}) = \sum_{k=0}^{n} s_{\theta,k}(x_{k-1}, \mathbf{x}_k), \tag{1.5}$$

under the conventions $x_{-1} = \emptyset$, $\mathbf{x}_0 = x_0$. The bold type notation $\mathbf{x}_k$, $k \geq 0$, is reserved for carefully defined variables involving elements of the infinite-dimensional pathspace. The specific construction of $\mathbf{x}_k$ will depend on the model at hand,

and will be explained in the main part of the paper. We note that the class of additive functionals considered supposes additivity over time, and therefore includes the general case of evaluating the score function. The online solution of the smoothing problem is often used as the means to solve some concrete inferential problems - see, e.g., (ii) below.

(ii) We take advantage of the new approach to show numerical applications, with emphasis on carrying out *online* parameter inference for the designated class of models via a gradient-ascent approach (in a Robbins-Monro stochastic gradient framework). A critical aspect of this particular online algorithm (partly likelihood based, when concerned with parameter estimation; partly Bayesian, with regards to identification of filtering/smoothing expectations) is that it delivers estimates of the evolving score function, of the model parameters, together with particle representations of the filtering distributions, through a *single passage* of the data. This is a unique favourable algorithmic characteristic, when constrasted with alternative algorithms with similar objectives, such as, e.g., Particle MCMC (Andrieu et al., 2010), or SMC^2 (Chopin et al., 2013).

(iii) In this work, we will *not* characterise analytically the size of the time-discretisation bias relevant to the SDE models at hand, and are content that: (I) the bias can be decreased by increasing the resolution of the numerical scheme (typically an Eyler-Maruyama one, or some other Taylor scheme, see e.g. Kloeden and Platen (2013)); (II) critically, the Monte-Carlo algorithms are developed in a manner that the variance does not increase (in the limit, up to infinity) when increasing the resolution of the time-discretisation method; to achieve such an effect, the algorithms are (purposely) defined on the infinite-dimensional pathspace, and SDE paths are only discretised when implementing the algorithm on a PC (to allow, necessarily, for finite computations).

(iv) Our method draws inspiration from earlier works, in the context of online filtering for discrete-time HMMs and infinite-dimensional pathspace MCMC methods. The

complete construct is novel; one consequence of this is that it is applicable, in principle, for a wide class of SDEs, under the following weak assumption (relatively to restrictive conditions often imposed in the literature of infinite-dimensional MCMC methods).

Assumption 2. *The diffusion covariance matrix function,*

$$\Sigma_\theta(v) := \sigma_\theta(v)\sigma_\theta(v)^\top \in \mathbb{R}^{d_x \times d_x},$$

is invertible, for all relevant v, θ.

Thus, the methodology does not apply as defined here only for the class of hypoelliptic SDEs.

An elegant solution to the online smoothing problem posed above in (i), for the case of a standard HMM with discrete-time signal of known transition density $f_\theta(x'|x)$, is given in Del Moral et al. (2010); Poyiadjis et al. (2011). Our own work overcomes the unavailability of the transition density in the continuous-time scenario by following the above literature but augmenting the hidden state with the complete continuous-time SDE path. Related augmentation approaches in this setting – though for different inferential objectives – have appeared in Fearnhead et al. (2008); Ströjby and Olsson (2009); Gloaguen et al. (2018), where the auxiliary variables are derived via the Poisson estimator of transition densities for SDEs (under strict conditions on the class of SDEs; no jumps), introduced in Beskos et al. (2006), and in Särkkä and Sottinen (2008) where the augmentation involves indeed the continuous-time path (the objective therein is to solve the filtering problem and the method is applicable for SDEs with additive Wiener noise; no jump processes are considered).

A Motivating Example: Fig. 1.1 (Top-Panel) shows estimates of the score function, evaluated at the true parameter value $\theta = \theta^\dagger$, for parameter θ_3 of the Ornstein–Uhlenbeck (O–U) process, $dX_t = \theta_1(\theta_2 - X_t)dt + \theta_3 dW_t$, $X_0 = 0.0$, for $n = 10$ observations $y_i = x_i + \epsilon_i$, $\epsilon_i \overset{i.i.d}{\sim} \mathcal{N}(0, 0.1^2)$, $1 \le i \le n$. Data were simulated from $\theta^\dagger = (0.5, 0.0, 0.4)$ with an Euler-Maruyama scheme of $M^\dagger = 1,000$ grid points per unit of time. Fig. 1.1 (Top-Panel) illustrates the 'abnormal' effect of a standard data-augmentation scheme, where for $N = $

100 particles, the Monte-Carlo method (see later sections for details) produces estimates of increasing variability as algorithmic resolution increases with $M = 10, 50, 100, 200$ – i.e., as it approaches the 'true' resolution used for the data generation. In contrast, Fig. 1.1 (Bottom Panel) shows results for the same score function as estimated by our proposed method in this paper. Our method is well-defined on the infinite-dimensional pathspace, and, consequently, it manages to robustly estimate the score function under mesh-refinement.
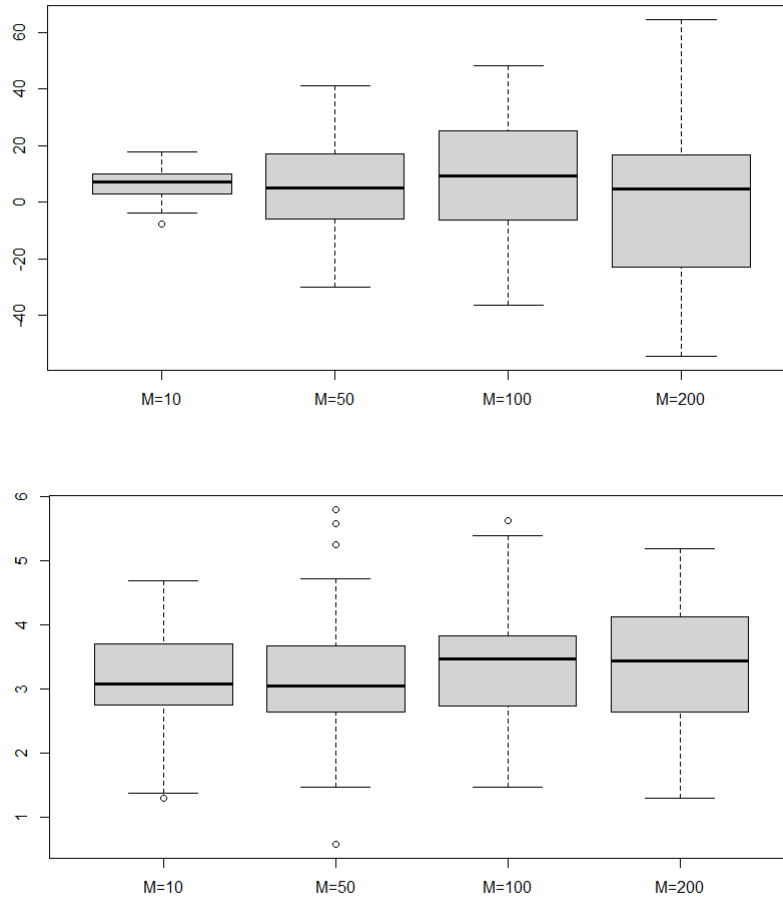

Figure 1.1: Boxplots of estimated score functions of θ_3 for the O–U process, for different resolution sizes M, over $R = 50$ experiment replications. $N = 100$ particles were used in all cases, for the same $n = 10$ data-points. *Top-Panel*: Results are produced via a Monte-Carlo algorithm that uses a naive 'finite-dimensional' data augmentation approach. *Bottom-Panel*: Results are produced via a Monte-Carlo algorithm that uses the 'infinite-dimensional' data augmentation approach introduced in this paper.

The rest of the paper is organised as follows. Section 2 reviews the Forward-Only algorithm for the online approximation of expectations of a class of additive functionals described in Del Moral et al. (2010). Section 3 sets up the framework for the treatment of pathspace-valued SDEs, first for the conceptually simpler case of SDEs without jumps, and then proceeding with incorporating jumps. Section 4 provides the complete online approximation algorithm, constructed on the infinite-dimensional pathspace. Section 5 discusses the adaptation of the developed methodology for the purposes of online inference for unknown parameters of the given SDE model. Section 6 shows numerical applications of the developed methodology. Section 7 contains conclusions and directions for future research.

2   Forward-Only Smoothing

A bootstrap filter (Gordon et al., 1993) is applicable in the continuous-time setting, as it only requires forward sampling of the underlying signal $\{X_t\}_{t\geq 0}$; this is trivially possible – under numerous approaches – and is typically associated with the introduction of some time-discretisation bias (Kloeden and Platen, 2013). However, the transition density is still required for the smoothing problem we have posed in the Introduction. In this section, we assume a standard discrete-time HMM, with initial distribution $p_\theta(dx_0)$, transition density $f_\theta(x'|x)$, and likelihood $g_\theta(y|x)$, for appropriate $x, x' \in \mathbb{R}^{d_x}$, $y \in \mathbb{R}^{d_y}$, and review the online algorithm developed in Del Moral et al. (2010) for this setting. Implementation of the bootstrap filter provides an approximation of the smoothing distribution $p_\theta(dx_{0:n}|y_{0:n})$ by following the geneology of the particles. This method is studied, e.g., in Cappe (2009); Dahlhaus and Neddermeyer (2010). Let $\{x_{0:n}^{(i)}, W_n^{(i)}\}_{i=1}^N$, $N \geq 1$, be a particle approximation of the smoothing distribution $p_\theta(dx_{0:n}|y_{0:n})$, in the sense that we have the estimate,

$$\widehat{p_\theta}(dx_{0:n}|y_{0:n}) = \sum_{i=1}^{N} W_n^{(i)} \delta_{x_{0:n}^{(i)}}(dx_{0:n}), \quad \sum_{i=1}^{N} W_n^{(i)} = 1, \tag{2.1}$$

with $\delta_{x_{0:n}^{(i)}}(dx_{0:n})$ the Dirac measure with an atom at $x_{0:n}^{(i)}$. Then, replacing $p_\theta(x_{0:n}|y_{0:n})$ with its estimate in (2.1) provides consistent estimators of expectations of the HMM

smoothing distributions. Though the method is online and the computational cost per time step is $\mathcal{O}(N)$, it typically suffers from the well-documented *path-degeneracy* problem – as illustrated via theoretical results or numerically (Del Moral et al., 2010; Kantas et al., 2015). That is, as n increases, the particles representing $p_\theta(dx_{0:n}|y_{0:n})$ obtained by the above method will eventually all share the same ancestral particle due to the resampling steps, and the approximation collapses for big enough n. This is well-understood not to be a solution to the approximation of the smoothing distribution for practical applications.

An approach which overcomes path-degeneracy is the Forward Filtering Backward Smoothing (FFBS) algorithm of Doucet et al. (2000). We briefly review the method here, following closely the notation and development in Del Moral et al. (2010). In the forward direction, assume that a filtering algorithm (e.g. bootstrap) has provided a particle approximation of the filtering distribution $p_\theta(dx_{k-1}|y_{0:k-1})$ – assuming a relevant k –,

$$\widehat{p}_\theta(dx_{k-1}|y_{0:k-1}) = \sum_{i=1}^{N} W_{k-1}^{(i)} \delta_{x_{k-1}^{(i)}}(dx_{k-1}), \tag{2.2}$$

for weighted particles $\{x_{k-1}^{(i)}, W_{k-1}^{(i)}\}_{i=1}^{N}$. In the backward direction, assume that one is given the particle approximation of the marginal smoothing distribution $p_\theta(dx_k|y_{0:n})$,

$$\widehat{p}_\theta(dx_k|y_{0:n}) = \sum_{i=1}^{N} W_{k|n}^{(i)} \delta_{x_k^{(i)}}(dx_k). \tag{2.3}$$

One has that (Kitagawa, 1987),

$$p_\theta(dx_{k-1:k}|y_{0:n}) = p_\theta(dx_k|y_{0:n}) \otimes p_\theta(dx_{k-1}|x_k, y_{0:k-1})$$

$$= p_\theta(dx_k|y_{0:n}) \otimes p_\theta(dx_{k-1}|y_{0:k-1}) \frac{f_\theta(x_k|x_{k-1})}{\int f_\theta(x_k|x_{k-1})p_\theta(x_{k-1}|y_{0:k-1})dx_{k-1}}. \tag{2.4}$$

Using (2.2)-(2.3), and based on equation (2.4), we obtain the approximation,

$$\widehat{p}_\theta(dx_{k-1:k}|y_{0:n}) = \sum_{j=1}^{N} W_{k|n}^{(j)} \sum_{i=1}^{N} \frac{f_\theta(x_k^{(j)}|x_{k-1}^{(i)})W_{k-1}^{(i)}}{\sum_{l=1}^{N} f_\theta(x_k^{(j)}|x_{k-1}^{(l)})W_{k-1}^{(l)}} \delta_{(x_{k-1}^{(i)}, x_k^{(j)})}(dx_{k-1:k}). \tag{2.5}$$

Recalling the expectation of additive functionals in (1.4)-(1.5) – where now, in the discrete-time setting, we can ignore the bold $\mathbf{x}_k$ elements, and simply use x_k instead – the above calculations give rise to the following estimator of the target quantity $\mathcal{S}_{\theta,n}$ in (1.5),

$$\widehat{\mathcal{S}}_{\theta,n} = \sum_{k=0}^{n} \int s_{\theta,k}(x_{k-1}, x_k)\, \widehat{p}_\theta(dx_{k-1:k}|y_{0:n}).$$

To be able to apply the above method, the marginal smoothing approximation in (2.3) is obtained via a backward recursive approach. In particular, starting from $k = n$ (where the approximation is provided by the standard forward particle filter), one proceeds as follows. Given k, the quantity for $k-1$ is directly obtained by integrating out x_k in (2.5), thus we have,

$$\widehat{p}_\theta(dx_{k-1}|y_{0:n}) = \sum_{i=1}^{N} W_{k-1|n}^{(i)} \delta_{x_{k-1}^{(i)}}(dx_{k-1}),$$

for the normalised weights,

$$W_{k-1|n}^{(i)} \propto \sum_{j=1}^{N} W_{k|n}^{(j)} \frac{f_\theta(x_k^{(j)}|x_{k-1}^{(i)}) W_{k-1}^{(i)}}{\sum_{l=1}^{N} f_\theta(x_k^{(j)}|x_{k-1}^{(l)}) W_{k-1}^{(l)}}.$$

Notice that – in this version of FFBS – the same particles $\{x_k^{(i)}\}_{i=1}^{N}$ are used in both directions (the ones before resampling at the forward filter), but with different weights.

An important development made in Del Moral et al. (2010) is transforming the above *offline* algorithm into an *online* one. This is achieved by consideration of the sequence of instrumental functionals,

$$T_{\theta,0}(x_0) = s_{\theta,0}(x_0); \qquad T_{\theta,n}(x_n) := \int S_{\theta,n}(x_{0:n}) p_\theta(dx_{0:n-1}|y_{0:n-1}, x_n), \quad n \geq 1.$$

Notice that, first,

$$\mathcal{S}_{\theta,n} = \int T_{\theta,n}(x_n) p_\theta(dx_n|y_{0:n}).$$

We also have that,

$$T_{\theta,n}(x_n) = \int \big[T_{\theta,n-1}(x_{n-1}) + s_{\theta,n}(x_{n-1}, x_n)\big] p_\theta(dx_{n-1}|y_{0:n-1}, x_n)$$
$$\equiv \frac{\int \big[T_{\theta,n-1}(x_{n-1}) + s_{\theta,n}(x_{n-1}, x_n)\big] f_\theta(x_n|x_{n-1}) p_\theta(dx_{n-1}|y_{0:n-1})}{\int f_\theta(x_n|x_{n-1}) p_\theta(dx_{n-1}|y_{0:n-1})}, \qquad (2.6)$$

– see Proposition 2.1 of Del Moral et al. (2010) for a (simple) proof. This recursion provides an online – forward-only – advancement of FFBS for estimating the smoothing expectation of additive functionals. The complete method is summarised in Algorithm 1: one key ingredient is that, during the recursion, values of the functional $T_{\theta,n}(x_n)$ are only required at the discrete positions $x_n^{(i)}$ determined by the forward particle filter.

In the SDE context, under Assumption 1, the transition density $f_\theta(\cdot|\cdot)$ is considered intractable, thus Algorithm 1 – apart from serving as a review of the method in Del Moral et al. (2010) – does not appear to be practical in the continuous-time case.

3   Data Augmentation on Diffusion Pathspace

To overcome the intractability of the transition density $f_\theta(\cdot|\cdot)$ of the SDE, we will work with an algorithm that is defined in continuous-time and makes use of the complete SDE path-particles in its development. The new method has connections with earlier works in the literature. Särkkä and Sottinen (2008) focus on the filtering problem for a class of models related to (1.1)-(1.3), and come up with an approach that requires the complete SDE path, for a limited class of diffusions with additive noise and no jumps. Fearnhead et al. (2008) also deal with the filtering problem, and – equipped with an unbiased estimator of the unknown transition density – recast the problem as one of filtering over an augmented space that incorporates the randomness for the unbiased estimate. The method is accompanied by strict conditions on the drift and diffusion coefficient (one should be able to transform the SDE – no jumps – into one of unit diffusion coefficient; the drift of the SDE must have a gradient form). Our contribution requires, in principle, solely the diffusion coefficient invertibility Assumption 2; arguably, the weakened condition we require stems from the

---

Algorithm 1 Online Forward-Only Smoothing (Del Moral et al., 2010)

(i) Initialise particles $\{x_0^{(i)}, W_0^{(i)}\}_{i=1}^N$, with $x_0^{(i)} \stackrel{iid}{\sim} p_\theta(dx_0)$, $W_0^{(i)} \propto g_\theta(y_0|x_0^{(i)})$, and functionals $\widehat{T}_{\theta,0}(x_0^{(i)}) = s_{\theta,0}(x_0^{(i)})$, for $1 \leq i \leq N$.

(ii) Assume that at time $n-1$, one has a particle approximation $\{x_{n-1}^{(i)}, W_{n-1}^{(i)}\}_{i=1}^N$ of the filtering law $p_\theta(dx_{n-1}|y_{0:n-1})$ and estimators $\widehat{T}_{\theta,n-1}(x_{n-1}^{(i)})$ of $T_{\theta,n-1}(x_{n-1}^{(i)})$, for $1 \leq i \leq N$.

(iii) At time n, sample $x_n^{(i)}$, for $1 \leq i \leq N$, from the mixture (Gordon et al., 1993),

$$x_n^{(i)} \sim \widehat{p}_\theta(x_n|y_{0:n}) = \sum_{j=1}^N W_{n-1}^{(j)} f_\theta(x_n|x_{n-1}^{(j)}),$$

and assign particle weights $W_n^{(i)} \propto g_\theta(y_n|x_n^{(i)})$, $1 \leq i \leq N$.

(iv) Then set, for $1 \leq i \leq N$,

$$\widehat{T}_{\theta,n}(x_n^{(i)}) = \frac{\sum_{j=1}^N W_{n-1}^{(j)} f_\theta(x_n^{(i)}|x_{n-1}^{(j)})}{\sum_{l=1}^N W_{n-1}^{(l)} f_\theta(x_n^{(i)}|x_{n-1}^{(l)})} \left[\widehat{T}_{\theta,n-1}(x_{n-1}^{(j)}) + s_{\theta,n}(x_{n-1}^{(j)}, x_n^{(i)}) \right].$$

(v) Obtain an estimate of $\mathcal{S}_{\theta,n}$ as,

$$\widehat{\mathcal{S}}_{\theta,n} = \sum_{i=1}^N W_n^{(i)} \widehat{T}_{\theta,n}(x_n^{(i)}).$$

---

fact that our approach appears as the relatively most 'natural' extension (compared to alternatives) of the standard discrete-time algorithm of Del Moral et al. (2010).

The latter discrete-time method requires the density $f_\theta(x'|x) = f_\theta(dx'|x)/dx'$. In continuous-time, we obtain an analytically available Radon-Nikodym derivative of $p_\theta(d\mathbf{x}'|x)$, for a properly defined variate $\mathbf{x}'$ that involves information about the continuous-time path for moving from x to x' within time Δ. We will give the complete algorithm in Section 4. In this section, we prepare the ground via carefully determining $\mathbf{x}'$ given x, and calculating the relevant densities to be later plugged in into our method.

3.1 SDEs with Continuous Paths

We work first with the process with continuous sample-paths, i.e. of dynamics,

$$dX_t = b_\theta(X_t)dt + \sigma_\theta(X_t)dW_t. \tag{3.1}$$

We adopt an approach motivated by techniques used for MCMC algorithms (Chib et al., 2004; Golightly and Wilkinson, 2008; Roberts and Stramer, 2001). Assume we are given starting point $x \in \mathbb{R}^{d_x}$, ending point $x' \in \mathbb{R}^{d_x}$, and the complete continuous-time path for the signal process in (3.1) on $[0, T]$, for some $T \geq 0$. That is, we now work with the path process,

$$\left[\{X_t; t \in [0, T]\} \,\big|\, X_0 = x, X_T = x' \right]. \tag{3.2}$$

Let $\mathbb{P}_{\theta, x, x'}$ denote the probability distribution of the pathspace-valued variable in (3.2). We consider the auxiliary bridge process $\tilde{X} = \{\tilde{X}_t; t \in [0, T]\}$ defined as,

$$d\tilde{X}_t = \left\{ b_\theta(\tilde{X}_t) + \frac{x' - \tilde{X}_t}{T - t} \right\} dt + \sigma_\theta(\tilde{X}_t)dW_t, \quad \tilde{X}_0 = x, \quad t \in [0, T], \tag{3.3}$$

with corresponding probability distribution $\mathbb{Q}_{\theta, x, x'}$. Critically, a path of $\tilde{X}$ starts at point x and finishes at x', w.p. 1. Under regularity conditions, Delyon and Hu (2006) prove that probability measures $\mathbb{P}_{\theta, x, x'}$, $\mathbb{Q}_{\theta, x, y}$ are absolutely continuous with respect to each other. We treat the auxiliary SDE (3.3) as a transform from the driving noise to the solution, whence a sample path, X, of the process $\tilde{X} = \{\tilde{X}_t; t \in [0, T]\}$, is produced by a mapping – determined by (3.3) – of a corresponding sample path, say Z, of the Wiener process. That is, we have set up a map, and – under Assumption 2 – its inverse,

$$Z \mapsto X =: F_\theta(Z; x, x'), \quad Z = F_\theta^{-1}(X; x, x'). \tag{3.4}$$

More analytically, F_θ^{-1} is given via the transform,

$$dZ_t = \sigma_\theta(X_t)^{-1}\left\{dX_t - b_\theta(X_t)dt - \frac{x' - X_t}{T - t}dt\right\}.$$

In this case we define,

$$\mathbf{x}' := (x', Z),$$

and the probability measure of interest is,

$$p_\theta(d\mathbf{x}'|x) := f_\theta(dx'|x) \otimes p_\theta(dZ|x', x). \tag{3.5}$$

Let $\mathbb{W}$ be the standard Wiener probability measure on $[0, T]$. Due to the 1–1 transform, we have that,

$$\frac{p_\theta(dZ|x', x)}{\mathbb{W}(dZ)} \equiv \frac{d\mathbb{P}_{\theta,x,x'}}{d\mathbb{Q}_{\theta,x,x'}}(F_\theta(Z; x, x')),$$

so it remains to obtain the density $d\mathbb{P}_{\theta,x,x'}/d\mathbb{Q}_{\theta,x,x'}$.

Such a Radom-Nikodym derivative has been object of interest in many works. Delyon and Hu (2006) provided detailed conditions and a proof, but (seemingly) omit an expression for the normalising constant which is important in our case, as it involves the parameter θ – in our applications later in the paper, we aim to infer about θ. Papaspiliopoulos et al. (2013); Papaspiliopoulos and Roberts (2012) provide an expression based on a conditioning argument for the projection of the probability measures on $[0, t)$, $t < T$, and passage to the limit $t \uparrow T$. The derivations in Delyon and Hu (2006) are extremely rigorous, so we will make use of the expressions in that paper. Following carefully the proofs of some of their main results (Theorem 5, together with Lemmas 7, 8) one can indeed retrieve the constant in the deduced density. In particular, Delyon and Hu (2006) impose the following conditions.

Assumption 3. *(i) SDE (3.1) admits a strong solution;*

(ii) $v \mapsto \sigma_\theta(v)$ is in $\mathcal{C}_b^2$ (i.e., twice continuously differentiable and bounded, with bounded first and second derivatives), and it is invertible, with bounded inverse;

(iii) $v \mapsto b_\theta(v)$ is locally Lipschitz, locally bounded.

Under Assumption 3, Delyon and Hu (2006) prove that,

$$\frac{d\mathbb{P}_{\theta,x,x'}}{d\mathbb{Q}_{\theta,x,x'}}(X) = \frac{|\Sigma_\theta(x')|^{1/2}}{|\Sigma_\theta(x)|^{1/2}} \times \frac{\mathcal{N}(x';x,T\Sigma_\theta(x))}{f_\theta(x'|x)} \times \varphi_\theta(X;x,x'), \tag{3.6}$$

where $\mathcal{N}(v;\mu,V)$ is the density function of the Gaussian law on $\mathbb{R}^{d_x}$ with mean μ, variance V, evaluated at v, $|\cdot|$ is matrix determinant, and $\varphi_\theta(X;x,x')$ is such that,

$$\log\varphi_\theta(X;x,x') = \int_0^T \left\langle b_\theta(X_t),\Sigma_\theta^{-1}(X_t)dX_t\right\rangle - \frac{1}{2}\int_0^T \left\langle b_\theta(X_t),\Sigma_\theta^{-1}(X_t)b_\theta(X_t)dt\right\rangle$$
$$- \frac{1}{2}\int_0^T \frac{\left\langle (x'-X_t),d\Sigma_\theta^{-1}(X_t)(x'-X_t)\right\rangle}{T-t} - \frac{1}{2}\int_0^T \frac{\sum_{i,j=1}^{d_x} d\left[\Sigma_{\theta,ij}^{-1},(x'_i-X_{t,i})(x'_j-X_{t,j})\right]}{T-t}.$$

Here, $[\cdot,\cdot]$ denotes the quadratic variation process for semi-martingales; also, $\langle\cdot,\cdot\rangle$ is the standard inner-product on $\mathbb{R}^{d_x}$. We note that transforms different from (3.4) have been proposed in the literature (Dellaportas et al., 2006; Kalogeropoulos et al., 2010) to achieve the same effect of obtaining an 1–1 mapping of the latent path that has a density with respect to a measure that does not depend on x, x' or θ. However, such methods are mostly applicable for scalar diffusions (Aït-Sahalia, 2008). Auxiliary variables involving a random, finite selection of points of the latent path, based on the (generalised) Poisson estimator of Fearnhead et al. (2008) are similarly restrictive. In contrast to other attempts, our methodology may be applied for a much more general class of SDEs, as determined by Assumption 2 – and further regularity conditions, e.g. as in Assumption 3. Thus, continuing from (3.5) we have obtained that,

$$\frac{p_\theta(d\mathbf{x}'|x)}{(\mathrm{Leb}^{\otimes d_x}\otimes\mathbb{W})(d\mathbf{x}')} = \varphi_\theta\big(F_\theta(Z;x,x');x,x'\big)\times\mathcal{N}(x';x,T\Sigma_\theta(x))\times\frac{|\Sigma_\theta(x')|^{1/2}}{|\Sigma_\theta(x)|^{1/2}}$$

$$=: p_\theta(\mathbf{x}'|x)\equiv p_\theta(\mathbf{x}'|x;T). \tag{3.7}$$

We have added the extra argument involving the length of path, T, in the last expression, as it will be of use in the next section.

Remark 1. *A critical point here is that the above density is* analytically tractable, *thus by working on pathspace we have overcome the unavailability of the transition density $f_\theta(x'|x)$.*

3.2 SDEs with Jumps

We extend the above developments to the more general case of the d_x-dimensional jump diffusion model given in (1.1), which we re-write here for convenience,

$$dX_t = b_\theta(X_{t-})dt + \sigma_\theta(X_{t-})dW_t + dJ_t, \quad X_0 = x \in \mathbb{R}^{d_x}, \ t \in [0, T]. \qquad (3.8)$$

Recall that $J_\theta = J = \{J_t\}$ denotes a compound Poisson process with jump intensity $\lambda_\theta(\cdot)$ and jump-size density $h_\theta(\cdot)$. Let $\mathbb{F}_{\theta,x}(\cdot)$ denote the law of the unconditional process (3.8) and $\mathbb{L}_\theta$ the law of the involved compound Poisson process. We write $J = ((\tau_1, b_1), \ldots, (\tau_\kappa, b_\kappa))$ to denote the jump process, where $\{\tau_i\}$ are the times of events, $\{b_i\}$ the jump sizes and $\kappa \geq 0$ the total number of events. In addition, we consider the reference measure $\mathbb{L}$, corresponding to unit rate Poisson process measure on $[0, T]$ multiplied with $\otimes_{i=1}^{\kappa+1} \mathrm{Leb}^{\otimes d_x}$.

Construct One

We consider the random variate,

$$\mathbf{x}' = \left(J, \{x_{\tau_i-}\}_{i=1}^{\kappa+1}, \{Z(i)\}_{i=1}^{\kappa+1} \right),$$

under the conventions $x_{\tau_0-} \equiv x$, $x_{\tau_{\kappa+1}-} \equiv x'$, where we have defined,

$$Z(i) = F_\theta^{-1}(X(i)\,;\, x_{\tau_{i-1}}, x_{\tau_i-}), \quad X(i) := \{X_t; t \in [x_{\tau_{i-1}}, x_{\tau_i})\}, \quad 1 \leq i \leq \kappa+1.$$

We have that,

$$p_\theta(d\mathbf{x}'|x) := \mathbb{L}_\theta(dJ) \otimes \left[\otimes_{i=1}^{\kappa+1} \left\{ f_\theta(dx_{\tau_i-}|x_{\tau_{i-1}}) \otimes p_\theta(dZ(i)|x_{\tau_{i-1}}, x_{\tau_i-}) \right\} \right].$$

Using the results about SDEs without jumps in Section 3.1, and in particular expression (3.7), upon defining,

$$\mathbf{x}'(i) := (x_{\tau_i-}, Z(i)),$$

we have that – under an apparent adaptation of Assumption 3,

$$\frac{f_\theta(dx_{\tau_i-}|x_{\tau_{i-1}}) \otimes p_\theta(dZ(i)|x_{\tau_{i-1}}, x_{\tau_i-})}{\mathrm{Leb}^{\otimes d_x}(dx_{\tau_i-}) \otimes \mathbb{W}(dZ(i)))} = p_\theta(\mathbf{x}'(i)|x_{\tau_i-}; \tau_i - \tau_{i-1}),$$

with the latter density $p_\theta(\mathbf{x}'(i)|x_{\tau_i-})$ determined as in (3.7), given clear adjustments. Thus, the density of $p_\theta(d\mathbf{x}'|x)$ with respect to the reference measure,

$$\mu(d\mathbf{x}') := \mathbb{L}(dJ) \otimes \left[\otimes_{i=1}^{\kappa+1} \left\{ \mathrm{Leb}^{\otimes d_x}(dx_{\tau_i-}) \otimes \mathbb{W}(dZ(i)) \right\} \right],$$

is equal to,

$$\frac{p_\theta(d\mathbf{x}'|x)}{\mu(d\mathbf{x}')} = \frac{e^{-\int_0^T \lambda_\theta(t)dt}}{e^{-T}} \cdot \prod_{i=1}^{\kappa} \left\{ \lambda_\theta(\tau_i)) h_\theta(b_i) \right\} \times \prod_{i=1}^{\kappa+1} p_\theta(\mathbf{x}'(i)|x_{\tau_i-}; \tau_i - \tau_{i-1}).$$

Construct Two

We adopt an idea used – for a different problem – in Gonçalves and Roberts (2014). Given $x, x' \in \mathbb{R}^{d_x}$, we define an auxiliary process $\tilde{X}_t$ as follows,

$$d\tilde{X}_t = \left\{ b_\theta(\tilde{X}_t) + \frac{x' - J_T - \tilde{X}_t + J_t}{T - t} \right\} dt + \sigma_\theta(\tilde{X}_t) dW_t + dJ_t, \quad X_0 = x, \qquad (3.9)$$

so that $\tilde{X}_T = x'$, w.p. 1. As with (3.4), we view (3.9) as a transform, projecting a path, Z of the Wiener process and the compound process, J, onto a path, X, of the jump process. That is, we consider the 1–1 maps,

$$(J, Z) \mapsto X =: G_\theta(J, Z; x, x'), \quad (J, Z) = G_\theta^{-1}(X; x, x').$$

Notice that for the inverse transform, the J-part is obtained immediately, whereas for the Z-part one uses the expression – well-defined due to Assumption 2 –,

$$dZ_t = \sigma_\theta(X_t)^{-1} \left\{ dX_t - dJ_t - b_\theta(X_t) dt - \frac{x' - J_T - X_t + J_t}{T - t} dt \right\}.$$

We denote by $\overline{\mathbb{P}}_{\theta,x,x'}$ the law of the original process in (3.8) conditionally on hitting x' at time T. Also, we denote the distribution on pathspace induced by (3.9) as $\overline{\mathbb{Q}}_{\theta,x,x'}$. Consider the variate,

$$\mathbf{x'} = (x', J, Z),$$

so that,

$$p_\theta(d\mathbf{x'}|x) := f_\theta(dx'|x) \otimes p_\theta\big(d(J,Z)|x',x\big).$$

Due to the employed 1–1 transforms, we have that,

$$\frac{p_\theta(d\mathbf{x'}|x)}{\big(\mathrm{Leb}^{\otimes d_x} \otimes \mathbb{L}_\theta \otimes \mathbb{W}\big)(d\mathbf{x'})} = f_\theta(x'|x) \times \frac{d\overline{\mathbb{P}}_{\theta,x,x'}}{d\overline{\mathbb{Q}}_{\theta,x,x'}}\big(G_\theta(J,Z;x,x')\big).$$

Thus, using the parameter-free reference measure,

$$\mu(d\mathbf{x'}) := \mathrm{Leb}^{\otimes d_x} \otimes \mathbb{L} \otimes \mathbb{W},$$

one obtains that,

$$\frac{p_\theta(d\mathbf{x'}|x)}{\mu(d\mathbf{x'})} = f_\theta(x'|x) \times \frac{e^{-\int_0^T \lambda_\theta(t)dt}}{e^{-T}} \cdot \prod_{i=1}^{\kappa} \big\{\lambda_\theta(\tau_i))h_\theta(b_i)\big\}$$

$$\times \frac{d\overline{\mathbb{P}}_{\theta,x,x'}}{d\overline{\mathbb{Q}}_{\theta,x,x'}}\big(F_\theta(J,Z;x,x')\big). \tag{3.10}$$

Remark 2. *Delyon and Hu (2006) obtained the Radon-Nikodym derivative expression in (3.6) after a great amount of rigorous analysis. A similar development for the case of conditioned jump diffusions does not follow from their work, and can only be subject of dedicated research at the scale of a separate paper. This is beyond the scope of our work. In practice, one can proceed as follows. For grid size $M \geq 1$, and $\delta = T/M$, let $\overline{\mathbb{P}}^M_{\theta,x,x'}(X_\delta, \ldots, X_{(M-1)\delta} \mid X_0 = x, X_{M\delta} = x')$ denote the time-discretised Lebesgue density of the $(M-1)$-positions of the conditioned diffusion with law $\overline{\mathbb{P}}_{\theta,x,x'}$. Once (3.10) is obtained,*

a time-discretisation approach will give,

$$\overline{\mathbb{P}}^M_{\theta,x,x'}(X_\delta, \ldots, X_{(M-1)\delta} \mid X_0 = x, X_{M\delta} = x')$$
$$= \frac{\overline{\mathbb{P}}^M_{\theta,x,x'}(X_\delta, \ldots, X_{(M-1)\delta}, X_{M\delta} = x' \mid X_0 = x)}{\overline{\mathbb{P}}^M_{\theta,x,x'}(X_{M\delta} = x' \mid X_0 = x)}.$$

In this time-discretised setting, $f_\theta(x'|x)$ in (3.6) will be replaced by $\overline{\mathbb{P}}^M_{\theta,x,x'}(X_{M\delta} = x' | X_0 = x)$. Thus, the intractable transition density over the complete time period will cancel out, and one is left with an explicit expression to use on a PC. Compared to the method in Section 3.1, and the Construct One in the current section, we do not have explicit theoretical evidence of a density on the pathspace. Yet, all numerical experiments we tried showed that the deduced algorithm was stable under mesh-refinement. We thus adopt the approach (or, conjecture) that the density in (3.10) exists – under assumptions –, and can be obtained pending future research.

4 Forward-Only Smoothing for SDEs

4.1 Pathspace Algorithm

We are ready to develop a forward-only particle smoothing method, under the scenario in (1.4)-(1.5) on the pathspace setting. We will work with the pairs of random elements

$$(x_{k-1}, \mathbf{x}_k), \quad 1 \le k \le n,$$

with $\mathbf{x}_k$ as defined in Section 3, i.e. containing pathspace elements, and given by an 1–1 transform of $\{X_s; s \in [t_{k-1}, t_k]\}$, such that we can obtain a density for $p_\theta(\mathbf{x}_k|x_{k-1})$ with respect to a reference measure that does not involve θ. Recall that $p_\theta(d\mathbf{x}_k|x_{k-1})$ denotes the probability law for the augmented variable $\mathbf{x}_k$ given x_{k-1}. We also write the corresponding

density as,

$$p_\theta(\mathbf{x}_k | x_{k-1}) := \frac{p_\theta(d\mathbf{x}_k | x_{k-1})}{\mu(d\mathbf{x}_k)}.$$

The quantity of interest is now,

$$\mathcal{S}_{\theta,n} = \int S_\theta(\mathbf{x}_{0:n}) \, p_\theta(d\mathbf{x}_{0:n} | y_{0:n}), \quad n \geq 1,$$

for the class of additive functionals $S(\cdot)$ of the structure,

$$S_\theta(\mathbf{x}_{0:n}) = \sum_{k=0}^{n} s_{\theta,k}(x_{k-1}, \mathbf{x}_k),$$

under the convention that $x_{-1} = \emptyset$. Notice that we now allow $s_k(\cdot, \cdot)$ to be a function of x_{k-1} and $\mathbf{x_k}$; thus, $s_k(\cdot, \cdot)$ can potentially correspond to integrals, or other pathspace functionals. We will work with a transition density on the enlarged space of $\mathbf{x}_k$.

Similarly to the discrete-time case in Section 2, we define the functional,

$$T_{\theta,n}(\mathbf{x}_n) := \int S_\theta(\mathbf{x}_{0:n}) \, p_\theta(d\mathbf{x}_{0:n-1} | y_{0:n-1}, \mathbf{x}_n).$$

Proposition 1. *We have that,*

$$\mathcal{S}_{\theta,n} = \int T_{\theta,n}(\mathbf{x}_n) \, p_\theta(d\mathbf{x}_n | y_{0:n}).$$

Proof. See Appendix A $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\square$

Critically, as in (2.6), we obtain the following recursion.

Proposition 2. *For any $n \geq 1$, we have that,*

$$
\begin{aligned}
T_{\theta,n}(\mathbf{x}_n) &= \int \left[T_{\theta,n-1}(\mathbf{x}_{n-1}) + s_{\theta,n}(x_{n-1}, \mathbf{x}_n) \right] p_\theta(d\mathbf{x}_{n-1} | y_{0:n-1}, \mathbf{x}_n) \\
&\equiv \frac{\int \left[T_{\theta,n-1}(\mathbf{x}_{n-1}) + s_{\theta,n}(x_{n-1}, \mathbf{x}_n) \right] p_\theta(\mathbf{x}_n | x_{n-1}) \, p_\theta(d\mathbf{x}_{n-1} | y_{0:n-1})}{\int p_\theta(\mathbf{x}_n | x_{n-1}) \, p_\theta(dx_{n-1} | y_{0:n-1})}.
\end{aligned}
$$

Proof. See Appendix B. □

Proposition 2 gives rise to a Monte-Carlo methodology for a forward-only, online approximation of the smoothing expectation of interest. This is given in Algorithm 2.

Algorithm 2 Online Forward-Only Particle Smoothing on Pathspace
---

(i) Initialise the particles $\{x_0^{(i)}, W_0^{(i)}\}_{i=1}^N$, with $x_0^{(i)} \overset{i.i.d.}{\sim} p_\theta(dx_0)$, $W_0^{(i)} \propto g_\theta(y_0|x_0^{(i)})$, and functionals $\widehat{T}_{\theta,0}(x_0^{(i)}) = s_{\theta,0}(\mathbf{x}_0^{(i)})$, where $\mathbf{x}_0^{(i)} \equiv x_0^{(i)}$, $1 \le i \le N$.

(ii) Assume that at time $n-1$, one has a particle approximation $\{\mathbf{x}_{n-1}^{(i)}, W_{n-1}^{(i)}\}_{i=1}^N$ of the filtering law $p_\theta(d\mathbf{x}_{n-1}|y_{0:n-1})$ and estimators $\widehat{T}_{\theta,n-1}(\mathbf{x}_{n-1}^{(i)})$ of $T_{\theta,n-1}(\mathbf{x}_{n-1}^{(i)})$, $1 \le i \le N$.

(iii) At time n, sample $\mathbf{x}_n^{(i)}$, for $1 \le i \le N$, from

$$\mathbf{x}_n^{(i)} \sim \widehat{p}_\theta(d\mathbf{x}_n|y_{0:n-1}) = \sum_{j=1}^N W_{n-1}^{(j)} p_\theta(d\mathbf{x}_n|x_{n-1}^{(j)}),$$

and assign particle weights $W_n^{(i)} \propto g_\theta(y_n|y_{n-1}, \mathcal{F}_n^{(i)})$, $1 \le i \le N$.

(iv) Then set, for $1 \le i \le N$,

$$\widehat{T}_{\theta,n}(\mathbf{x}_n^{(i)}) = \frac{\sum_{j=1}^N W_{n-1}^{(j)} p_\theta(\mathbf{x}_n^{(i)}|x_{n-1}^{(j)})}{\sum_{l=1}^N W_{n-1}^{(l)} p_\theta(\mathbf{x}_n^{(i)}|x_{n-1}^{(l)})} \left[\widehat{T}_{\theta,n-1}(\mathbf{x}_{n-1}^{(j)}) + s_{\theta,n}(x_{n-1}^{(j)}, \mathbf{x}_n^{(i)}) \right].$$

(v) Obtain an estimate of $\mathcal{S}_{\theta,n}$ as,

$$\widehat{\mathcal{S}}_{\theta,n} = \sum_{i=1}^N W_n^{(i)} \widehat{T}_{\theta,n}(\mathbf{x}_n^{(i)}).$$

Remark 3. *Algorithm 2 uses a simple bootstrap filter with multinomial resampling applied at each step. The variability of the Monte-Carlo estimates can be further reduced by incorporating: more effective resampling (e.g., systematic resampling (Carpenter et al., 1999), stratified resampling (Kitagawa, 1996)); dynamic resampling via use of Effective Sample Size; non-blind proposals in the propagation of the particles.*

4.2 Pathspace versus Finite-Dimensional Construct

One can attempt to define an algorithm without reference to the underlying pathspace. That is, in the case of no jumps (for simplicity) an alternative approach can involve working with a regular grid on the period $[0, T]$, say $\{s_j = j\delta\}_{j=0}^{M}$, with $\delta = T/M$ for chosen size $M \geq 1$. Then, defining $\mathbf{x}' = (x_\delta, x_{2\delta}, \ldots, x_{M\delta})$, and using, e.g., an Euler-Maruyama time-discretisation scheme to obtain the joint density of such an $\mathbf{x}'$ given $x = x_0$, a Radon-Nikodym derivative, $p_\theta^M(\mathbf{x}'|x)$ on $\mathbb{R}^{M \times d_x}$, can be obtained with respect to the Lebesgue reference measure $\mathrm{Leb}^{\otimes(d_x \times M)}$, as a product of M conditionally Gaussian densities. As shown e.g. in the motivating example in the Introduction, such an approach would lead to estimates with variability that increases rapidly with M, for fixed Monte-Carlo particles N. A central argument in this work is that one should develop the algorithm in a manner that respect the probabilistic properties of the SDE pathspace, before applying (necessarily) a time-discretisation for implementation on a PC. This procedure is not followed for purposes of mathematical rigour, but it has practical effects on algorithmic performance.

4.3 Consistency

For completeness, we provide an asymptotic result for Algorithm 2 following standard results from the literature. We consider the following assumptions.

Assumption 4. *Let $\mathbf{X}$ and X denote the state spaces of $\mathbf{x}$ and x respectively.*

(i) For any relevant y', y, $\mathcal{F}$, x', we have that $g_\theta(y'|y, \mathcal{F}) \equiv g_\theta(y'|x')$, where the latter is a positive function such that, for any y, $\sup_{x \in \mathsf{X}} g_\theta(y|x) < \infty$.

(ii) $\sup_{(x, \mathbf{x}') \in X \times \mathbf{X}} p_\theta(\mathbf{x}'|x) < \infty$.

Proposition 3. *(i) Under Assumption 4, for any $n \geq 0$, there exist constants $b_n, c_n > 0$, such that for any $\epsilon > 0$,*

$$\mathrm{Prob}\big[\,|\mathcal{S}_{\theta,n} - \hat{\mathcal{S}}_{\theta,n}| > \epsilon\,\big] \leq b_n e^{-c_n N \epsilon^2}.$$

(ii) For any $n \geq 0$, $\hat{\mathcal{S}}_{\theta,n} \to \mathcal{S}_{\theta,n}$ w.p.1, as $N \to \infty$.

Proof. See Appendix C. □

5   Online Parameter/State Estimation for SDEs

In this section, we derive an online gradient-ascent for partially observed SDEs. Poyiadjis et al. (2011) use the score function estimation methodology to propose an online gradient-ascent algorithm for obtaining an MLE-type parameter estimate, following ideas in LeGland and Mevel (1997). In more detail, the method is based on the Robbins-Monro-type of recursion,

$$\theta_{n+1} = \theta_n + \gamma_{n+1}\nabla \log p_{\theta_{0:n}}(y_n|y_{0:n-1})$$
$$= \theta_n + \gamma_{n+1}\big\{\nabla \log p_{\theta_{0:n}}(y_{0:n}) - \nabla \log p_{\theta_{0:n-1}}(y_{0:n-1})\big\} \tag{5.1}$$

where $\{\gamma_k\}_k$ is a positive decreasing sequence with,

$$\sum_{k=1}^{\infty} \gamma_k = \infty, \quad \sum_{k=1}^{\infty} \gamma_k^2 < \infty.$$

The meaning of quantity $\nabla \log p_{\theta_{0:n}}(y_{0:n})$ is that – given a recursive method (in n) for the estimation of $\theta \mapsto \nabla \log p_\theta(y_{0:n})$ as we describe below and based on the methodology of Algorithm 2 – one uses θ_{n-1} when incorporating y_{n-1}, then θ_n for y_n, and similarly for $k > n$. See LeGland and Mevel (1997); Tadic and Doucet (2018) for analytical studies of the convergence properties of the deduced algorithm, where under strong conditions the recursion is shown to converge to the 'true' parameter value, say $\theta^\dagger$, as $n \to \infty$.

Observe that, from Fisher's identity (see e.g. Poyiadjis et al. (2011)) we have that,

$$\nabla \log p_\theta(y_{0:n}) = \int \nabla \log p_\theta(\mathbf{x}_{0:n}, y_{0:n})p_\theta(d\mathbf{x}_{0:n}|y_{0:n}).$$

Thus, in the context of Algorithm 2, estimation of the score function corresponds to the

choice,

$$S_\theta(\mathbf{x}_{0:n}) = \nabla \log p_\theta(\mathbf{x}_{0:n}, y_{0:n}) \equiv \sum_{k=0}^{n} \nabla \log p_\theta(\mathbf{x}_k, y_k | x_{k-1}).$$

and,

$$s_{\theta,k}(x_{k-1}, \mathbf{x}_k) \equiv \nabla \log p_\theta(\mathbf{x}_k, y_k | x_{k-1}). \tag{5.2}$$

Combination of the Robins-Morno recursion (5.1) with the one in Algorithm 2, delivers Algorithm 3, which we have presented here in some detail for clarity.

Remark 4. *When the joint density of* $(\mathbf{x}_{0:n}, y_{0:n})$ *is in the exponential family, an online EM algorithm can also be developed; see Del Moral et al. (2010) for the discrete-time case.*

6   Numerical Applications

All numerical experiments shown in this section were obtained on a standard PC, with code written in `Julia`. Unless otherwise stated, executed algorithms discretised the pathspace using the Euler–Maruyama scheme with $M = 10$ imputed points between consecutive observations, the latter assumed to be separated by 1 time unit. Algorithmic costs scale linearly with M.

To specify the schedule for the scaling parameters $\{\gamma_k\}$ in (5.1), we use the standard adaptive method termed `ADAM`, by Kingma and Ba (2014). Assume that after n steps, we have $c_n := -(\nabla \log p_{\hat\theta_{0:n}}(y_{0:n}) - \nabla \log p_{\hat\theta_{0:n-1}}(y_{0:n-1}))$. `ADAM` applies the following iterative procedure,

$$m_n = m_{n-1}\beta_1 + (1 - \beta_1)c_n, \quad v_n = v_{n-1}\beta_2 + (1 - \beta_2)c_n^2,$$
$$\hat{m}_n = m_n/(1 - \beta_1^n), \quad \hat{v}_n = v_n/(1 - \beta_2^n),$$
$$\hat\theta_{n+1} = \hat\theta_n - \alpha\hat{m}_n/(\sqrt{\hat{v}_n} + \epsilon),$$

where $(\beta_1, \beta_2, \alpha, \epsilon)$ are tuning parameters. Convergence properties of `ADAM` have been widely

---

Algorithm 3 Online Gradient-Ascent for SDEs via Forward-Only Smoothing

---

(i) Assume that at time $n \geq 0$, one has current parameter estimate $\hat{\theta}_n$, particle approximation $\{\mathbf{x}_n^{(i)}, W_n^{(i)}\}_{i=1}^N$ of the 'filter' $p_{\hat{\theta}_{0:n}}(d\mathbf{x}_n|y_{0:n})$ and estimators $\widehat{T}_{\hat{\theta}_{0:n},n}(\mathbf{x}_n^{(i)})$ of $T_{\hat{\theta}_{0:n},n}(\mathbf{x}_n^{(i)})$, for $1 \leq i \leq N$.

(ii) Apply the iteration,

$$\hat{\theta}_{n+1} = \hat{\theta}_n + \gamma_{n+1}\big\{\nabla \log p_{\hat{\theta}_{0:n}}(y_{0:n}) - \nabla \log p_{\hat{\theta}_{0:n-1}}(y_{0:n-1})\big\}$$

(iii) At time $n + 1$, sample $\mathbf{x}_{n+1}^{(i)}$, for $1 \leq i \leq N$, from the mixture,

$$\mathbf{x}_{n+1}^{(i)} \sim \widehat{p}_{\hat{\theta}_{n+1}}(d\mathbf{x}_{n+1}|y_{0:n}) = \sum_{j=1}^N W_n^{(j)} \, p_{\hat{\theta}_{n+1}}(\mathbf{x}_{n+1}|x_n^{(j)}),$$

and assign particle weights $W_{n+1}^{(i)} \propto g_{\hat{\theta}_{n+1}}(y_{n+1}|y_n, \mathcal{F}_{n+1}^{(i)})$, $1 \leq i \leq N$.

(iii) Then set, for $1 \leq i \leq N$,

$$\widehat{T}_{\hat{\theta}_{0:n+1},n+1}(\mathbf{x}_{n+1}^{(i)}) = \frac{\sum_{j=1}^N W_n^{(j)} \, p_\theta(\mathbf{x}_{n+1}^{(i)}|x_n^{(j)})}{\sum_{l=1}^N W_n^{(l)} \, p_\theta(\mathbf{x}_{n+1}^{(i)}|x_n^{(l)})} \big[\widehat{T}_{\hat{\theta}_{0:n},n}(\mathbf{x}_n^{(j)}) + s_{\theta,n}(x_n^{(j)}, \mathbf{x}_{n+1}^{(i)})\big],$$

where, on the right-hand-side we use the parameter,

$$\theta = \hat{\theta}_{n+1}.$$

(iv) Obtain the estimate,

$$\widehat{S}_{\hat{\theta}_{n+1},n+1} = \sum_{i=1}^N W_{n+1}^{(i)} \, \widehat{T}_{\hat{\theta}_{0:n+1},n+1}(\mathbf{x}_{n+1}^{(i)}).$$

---

studied (Goodfellow et al., 2016; Kingma and Ba, 2014; Reddi et al., 2019). Following Kingma and Ba (2014), in all cases below we set $(\beta_1, \beta_2, \alpha, \epsilon) = (0.9, 0.999, 0.001, 10^{-8})$. ADAM is nowadays a standard and very effective addition to the type of recursive inference algorithms we are considering here, even more so as for increasing dimension of unknown parameters. See the above references for more motivation and details.

6.1   Ornstein-Uhlenbeck SDE

We consider the model,

$$dX_t = \theta_1(\theta_2 - X_t) + \theta_3 dW_t + dJ_t, \quad X_0 = 0,$$

$$y_i = x_i + \epsilon_i, \quad i \geq 1, \quad \epsilon_i \overset{i.i.d.}{\sim} \mathcal{N}(0, 0.1^2), \quad \Delta = 1, \tag{6.1}$$

where $J_t = \sum_{i=1}^{N_t} \xi_i$, N_t is a Poisson process of intensity $\lambda \geq 0$, and $\xi_i \overset{i.i.d.}{\sim} \mathcal{U}(-\zeta, \zeta)$, $\zeta > 0$.

O–U: Experiment 1. Here, we choose $\lambda = 0$, so the score function is analytically available; we also fix $\theta_2 = 0$. We simulated datasets of size $n = 2,500, 5,000, 7,500, 10,000$ under true remaining parameter values $(\theta_1^\dagger, \theta_3^\dagger) = (0.4, 0.5)$. We executed Algorithm 2 to approximate the score function with $N = 50, 100, 150$ particles, for all above datasets. Fig. 6.1 summarises results of $R = 50$ replicates of estimates of the bivariate score function, with the dashed lines showing the true values.

O–U: Experiment 2. We now present two scenarios: one still without jumps (i.e., $\lambda = 0$); and one with jumps, such that we fix $\lambda = 0.5$, $\zeta = 0.5$. In both scenarios, the parameters to be estimated are $\theta = (\theta_1, \theta_2, \theta_3)$. Beyond the mentioned fixed values for λ, ζ, in both scenarios we generated $n = 20,000$ data points using the true parameters values $\theta^\dagger = (\theta_1^\dagger, \theta_2^\dagger, \theta_3^\dagger) = (0.2, 0.0, 0.2)$. We applied Algorithm 3 with $N = 100$ particles, and initial parameter values $\hat{\theta}_0 = (1.0, 1.0, 1.0)$ for both scenarios. We used Construct 1 for the scenario with jumps. Fig. 6.2 summarises the results.

O–U: Experiment 3. To compare the efficiency of Constructs 1 and 2 for the case of jump processes, as developed in Section 3.2, we applied Algorithm 2 – for each case – to approximate the score function (at the true values for $(\theta_1, \theta_2, \theta_3)$), with $N = 50, 100, 150, 200$ particles, with $n = 10$ data generated from O–U process (6.1) with fixed $\lambda = 0.5$, $\zeta = 0.5$, and true remaining parameters $(\theta_1^\dagger, \theta_2^\dagger, \theta_3^\dagger) = (0.3, 0.0, 0.2)$. Fig. 6.3 shows boxplots summarising the results from $R = 50$ replications of the obtained estimates. We note that performance of Construct 2 is, in this case, much better than that of Construct 1.
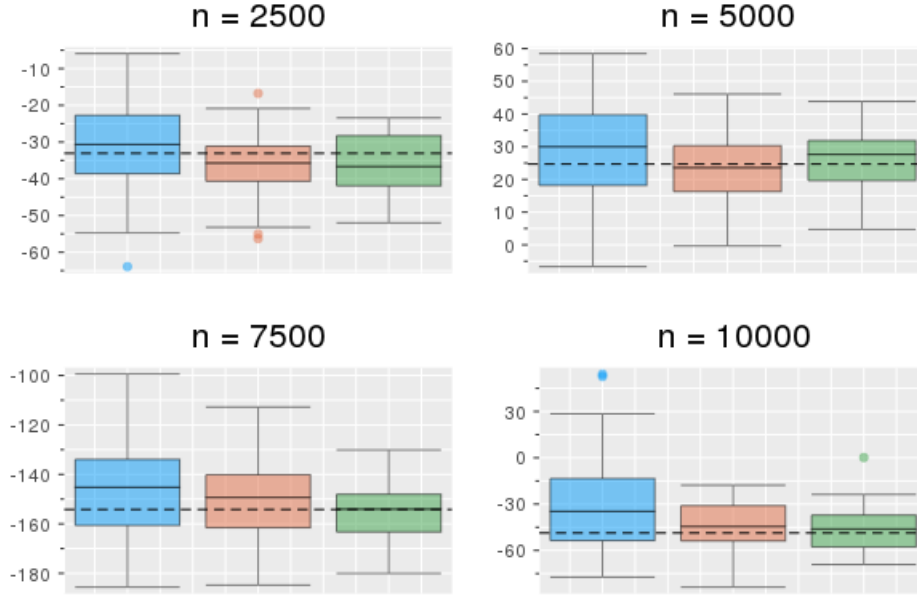
Figure 6.1: *O–U: Experiment 1:* Boxplots of estimated score functions of θ_1 at $\theta_1 = 0.4$, over $R = 50$ experiment replications, for the O–U model (6.1), without jumps ($\lambda = 0$), fixed $\theta_2 = 0$, and free parameters (θ_1, θ_3), with true values $(\theta_1^{\dagger}, \theta_3^{\dagger}) = (0.4, 0.5)$ used for data generation. The left, middle and right boxplots in each panel correspond to values $N = 50, 100, 150$, respectively. The horizontal dashed lines show the correct values $(-33.1, 24.7, -154.2, -48.7)$ for datasizes $n = 2,500, 5,000, 7,500, 10,000$, respectively.

6.2 Periodic Drift SDE

We consider the (highly) nonlinear model,

$$dX_t = \sin\left(X_t - \theta_1\right) dt + \theta_2 dW_t, \quad X_0 = 0,$$

$$y_i = x_i + \epsilon_i, \quad i \geq 1, \quad \epsilon_i \overset{i.i.d.}{\sim} \mathcal{N}(0, 0.1^2), \quad \Delta = 1, \tag{6.2}$$

for $\theta_1 \in [0, 2\pi)$, $\theta_2 > 0$. We used true values $(\theta_1^{\dagger}, \theta_2^{\dagger}) = (\pi/4, 0.9)$, generated $n = 10^4$ data, and applied Algorithm 3 with $N = 100$ particles and initial values $(\hat{\theta}_{1,0}, \hat{\theta}_{2,0}) = (0.1, 2)$. To illustrate the mesh-free performance of Algorithm 3, we choose two values, $M = 10$ and $M = 100$ and compare the obtained numerical results. Figure 6.4 shows the data (Left Panel) and the trajectory of the estimated parameter values (Right Panel). Note that the SDE in (6.2) is making infrequent jumps between a number of modes. Indeed, in this case, the results are stable when increasing M from 10 to 100.
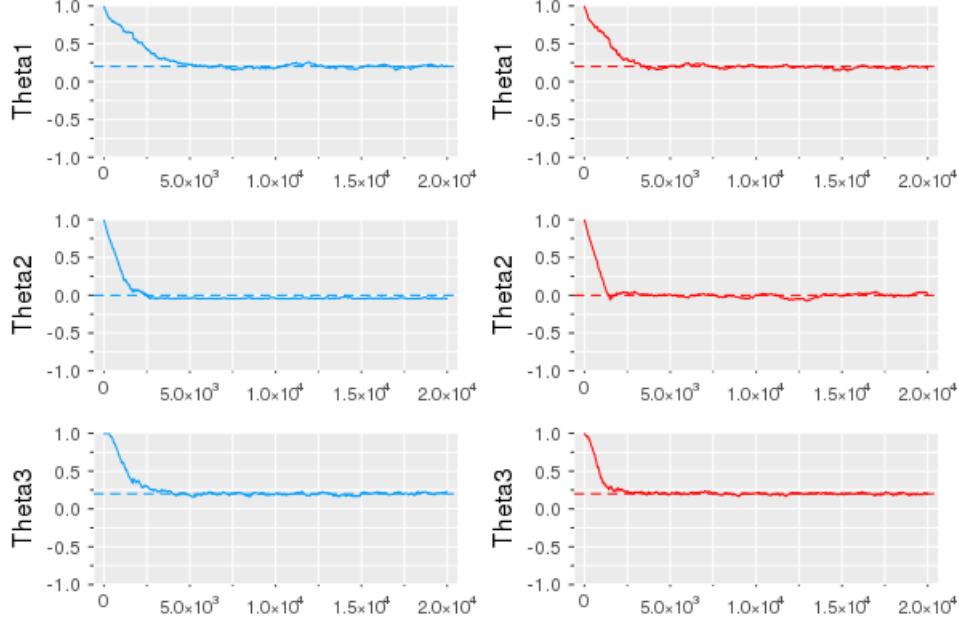
27

Figure 6.2: *O–U: Experiment 2:* Trajectories from execution of Algorithm 3 for the estimation of parameters $(\theta_1, \theta_2, \theta_3)$ of the O–U model (6.1). We used $N = 100$ particles and initial value $\hat{\theta}_0 = (1.0, 1.0, 1.0)$. *Left Panel:* results for O–U model with jumps ($\lambda = 0.5$, $\zeta = 0.5$). *Right Panel:* Results for O–U model without jumps ($\lambda = 0$). The horizontal dashed lines in the plots show the true parameter values $(\theta_1^\dagger, \theta_2^\dagger, \theta_3^\dagger) = (0.2, 0.0, 0.2)$ used in both scenarios.

6.3   Heston Model

Consider the Heston model (Heston, 1993), for price S_t and volatility X_t, modelled as,

$$dS_t = \theta_4 S_t dt + S_t X_t^{1/2} dB_t,$$

$$dX_t = \theta_1(\theta_2 - X_t)dt + \theta_3 X_t^{1/2} dW_t, \quad X_0 = \theta_2,$$

where B_t, W_t are independent standard Brownian motions. Define the log-price $U_t = \log(S_t)$, so that application of Itô's lemma gives,

$$dU_t = (\theta_4 - \tfrac{X_t}{2})dt + X_t^{1/2} dB_t,$$

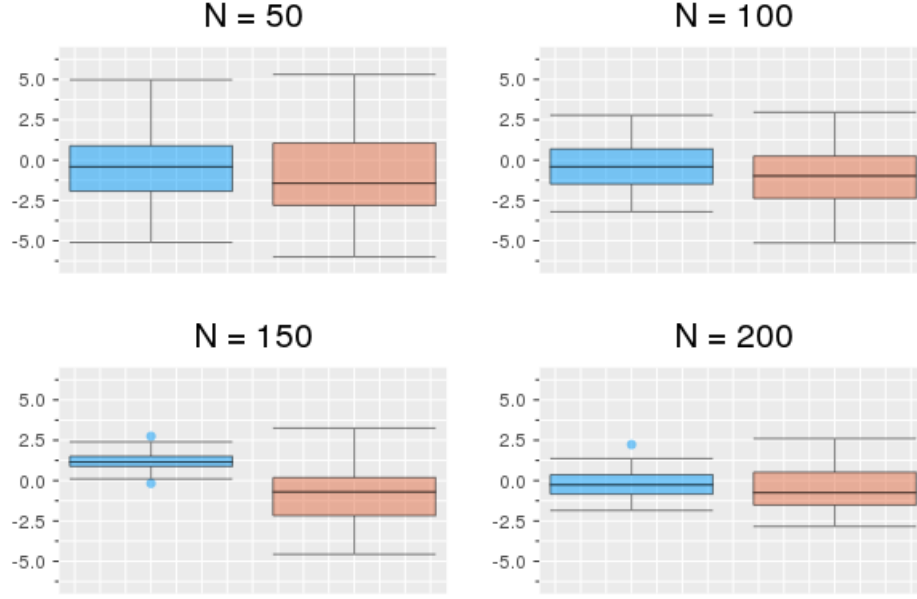$$dX_t = \theta_1(\theta_2 - X_t)dt + \theta_3 X_t^{1/2} dW_t, \quad X_0 = \theta_2. \tag{6.3}$$

Figure 6.3: *O–U: Experiment 3:* Boxplots of estimated score function of θ_1, at $\theta_1 = 0.3$, over $R = 50$ experiment replications, for O–U model (6.1), with fixed $\lambda = 0.5$, $\zeta = 0.5$, and true parameters $(\theta_1^\dagger, \theta_2^\dagger, \theta_3^\dagger) = (0.3, 0.0, 0.2)$, used to generate $n = 10$ observations. Boxplots on the right (left) of each panel correspond to Construct 1 (Construct 2).

We make the standard assumption $2\theta_1\theta_2 > \theta_3^2$, so that the CIR process $X = \{X_t\}$ will not hit 0; also, we have $\theta_1, \theta_2, \theta_3, \theta_4 > 0$. Process U_t is observed at discrete times, so that $y_i = U_{t_i}$, $i \geq 1$. Thus, we have,

$$\left[y_i \,\middle|\, y_{i-1}, \{X_s; s \in [t_{i-1}, t_i]\} \right] \sim \mathcal{N}(y_i; \mu_i, \Sigma_i),$$

where we have set,

$$\mu_i = y_{i-1} + \int_{t_{i-1}}^{t_i} (\theta_4 - \tfrac{X_s}{2})ds, \quad \Sigma_i = \int_{t_{i-1}}^{t_i} X_s ds.$$

We chose true parameter value $\theta^\dagger = (0.1, 1.0, 0.2, 0.45)$, and generated $n = 10^4$ observations, with $\Delta = 1$. We applied Algorithm 3 with $\hat{\theta}_0 = (0.005, 0.1, 0.4, 0.3)$ and $N = 100$ particles. The trajectories of the estimated parameters are given in Figure 6.5.
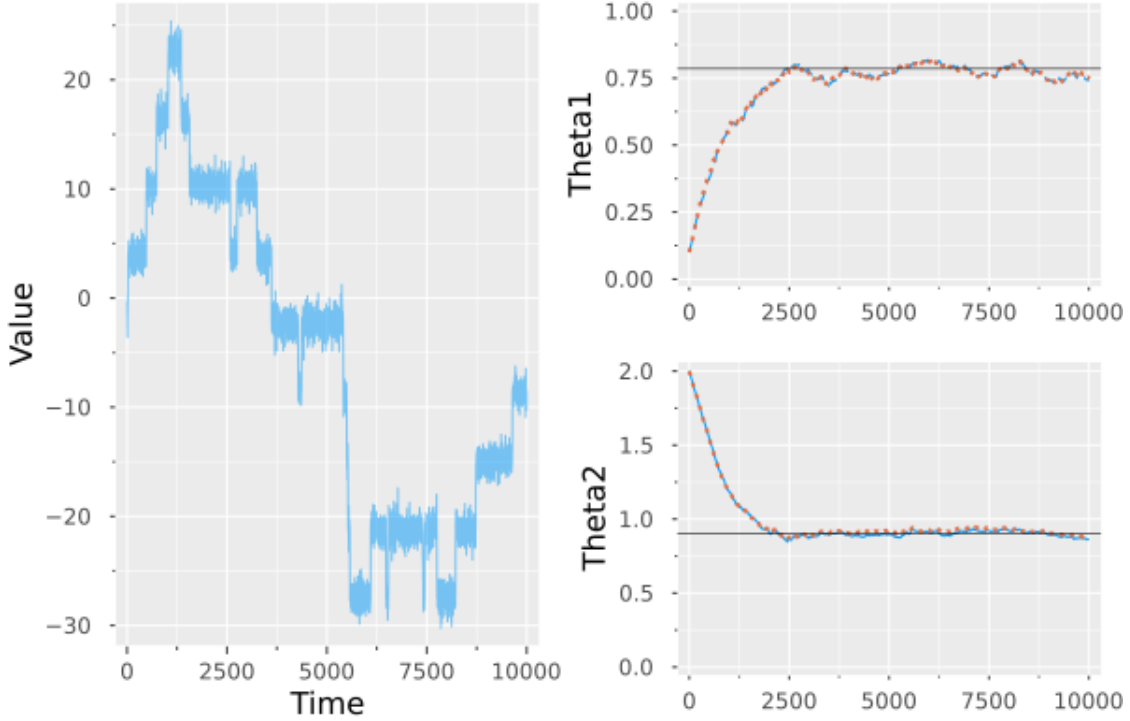
Figure 6.4: *Left Panel:* Data generated from model (6.2) for true values $\theta^\dagger = (\pi/4, 0.9)$. *Right Panel:* Trajectories obtained from Algorithm 3, with $N = 100$ and initial values $(\hat{\theta}_{1,0}, \hat{\theta}_{2,0}) = (0.1, 2)$. Solid lines show the results for $M = 10$ and dotted lines for $M = 100$.

6.4 Sequential Model Selection – Real Data

We use our methodology to carry out sequential model selection, motivated by applications in Eraker et al. (2003); Johannes et al. (2009). Recall that BIC (Schwarz, 1978) for model $\mathcal{M}$, with parameter vector θ, and data $y_{1:n}$ is given by,

$$BIC(\mathcal{M}) := -2\ell_{\hat{\theta}_{MLE}}(y_{1:n}) + \dim(\theta)\log n,$$

where $\hat{\theta}_{MLE}$ denotes the MLE, and $\ell(\cdot)$ the log-likelihood. BIC and the (closely related) Bayes Factor are known to have good asymptotic properties, e.g. they are consistent Model Selection Criteria, under the assumption of model-correctness, in the context of nested models, for specific classes of models (see e.g. Chib and Kuffner (2016); Nishii (1988); Sin and White (1996), and Yonekura et al. (2018) for the case of discrete-time nested HMMs). One can plausibly conjecture such criteria will also perform well for the type of continuous-time HMMs we consider here. See also Eguchi and Masuda (2018) for a rigorous analysis
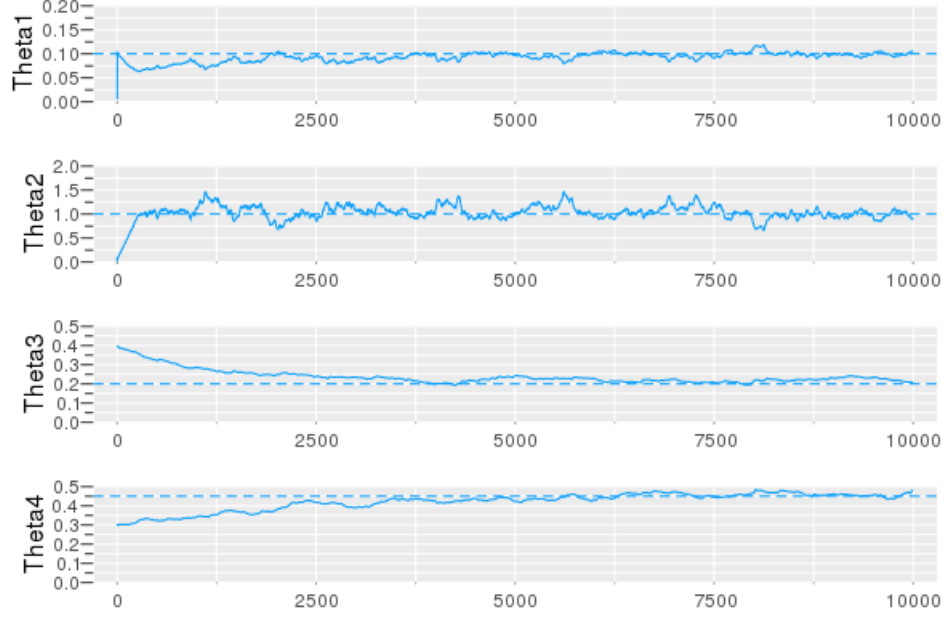
30

Figure 6.5: Parameter estimate trajectories produced by Algorithm 3, for data generated from Heston model (6.3). The algorithm used initial values $\hat{\theta}_0 = (0.005, 0.1, 0.4, 0.3)$, and $N = 100$ particles. The horizontal dashed lines indicate the true parameter values $\theta^\dagger = (0.1, 1.0, 0.2, 0.45)$.

of BIC for diffusion-type models. Given models $\mathcal{M}_a, \mathcal{M}_b$, with $\mathcal{M}_a \subset \mathcal{M}_b$, we use the difference,

$$BIC(\mathcal{M}_{ab}) := -2(\ell_{\hat{\theta}_{MLE,a}}(y_{1:n}) + \ell_{\hat{\theta}_{MLE,b}}(y_{1:n})) + (\dim(\theta_a) - \dim(\theta_b)) \log n,$$

with involved terms defined as obviously, to choose between $\mathcal{M}_a$ and $\mathcal{M}_b$.

We note that Eraker et al. (2003) use Bayes Factor for model selection in the context of SDE processes; that work uses MCMC to estimate parameters, and is not sequential (or online). We stress that our methodology allows for carrying out inference for parameters and the signal part of the model, and simultaneously allowing for model selection, all in an online fashion. Also, it is worth noting that Johannes et al. (2009) use the sequential likelihood ratio for model comparison, but such quantity can overshoot, i.e. the likelihood ratio will tend to choose a large model. Also, that work uses fixed calibrated parameters, so it does not relate to online parameter inference.

Remark 5. *We will use the running estimate of the parameter vector as a proxy for the*

MLE given the data already taken under consideration. Similarly, we use the weights of the particle filter to obtain a running proxy of the log-likelihood evaluated at the MLE, thus overall an online approximation of BIC. Such an approach, even not fully justified theoretically, can provide reasonable practical insights when performing model comparison in an online manner – particularly so, given when there is no alternative, to the best of our knowledge, for such an objective.

We consider the following family of nested SDE models,

$$dX_t = b_\theta^{(i)}(X_t) + \theta_4\sqrt{X_t}dW_t,$$

where,

$$\mathcal{M}_1: \quad b_\theta^{(1)}(X_t) = \theta_0 + \theta_1 X_t,$$

$$\mathcal{M}_2: \quad b_\theta^{(2)}(X_t) = \theta_0 + \theta_1 X_t + \theta_2^2 X_t^2,$$

$$\mathcal{M}_3: \quad b_\theta^{(3)}(X_t) = \theta_0 + \theta_1 X_t + \theta_2^2 X_t^2 + \frac{\theta_3}{X_t}.$$

Such models have been used for short-term interest rates; see Jones (2003); Durham (2003); Aït-Sahalia (1996); Bali and Wu (2006) and references therein for more details. Motivated by Dellaportas et al. (2006); Stanton (1997), we applied our methodology to daily 3-month Treasury Bill rates, from 2nd of January 1970, to 29th of December 2000, providing $n = 7,739$ observations. The data can be obtained from the Federal Reserve Bank of St. Louis, at webpage `https://fred.stlouisfed.org/series/TB3MS`. The dataset is shown at Fig. 6.6.

Fig. 6.7 shows the BIC differences, obtained online, for each of the three pairs of models. The results suggest that $\mathcal{M}_2$ does not provide a good fit to the data – relatively to $\mathcal{M}_1$, $\mathcal{M}_3$ – for almost all period under consideration. In terms of $\mathcal{M}_1$ against $\mathcal{M}_3$, there is evidence that once the data from around 1993 onwards are taken under consideration, $\mathcal{M}_3$ is preferred to $\mathcal{M}_1$. In general, one can claim that models with non-linear drift should be taken under consideration for fitting daily 3-month Treasury Bill rates, without strong evidence in favour or against linearity. This non-definite conclusion is in some agreement
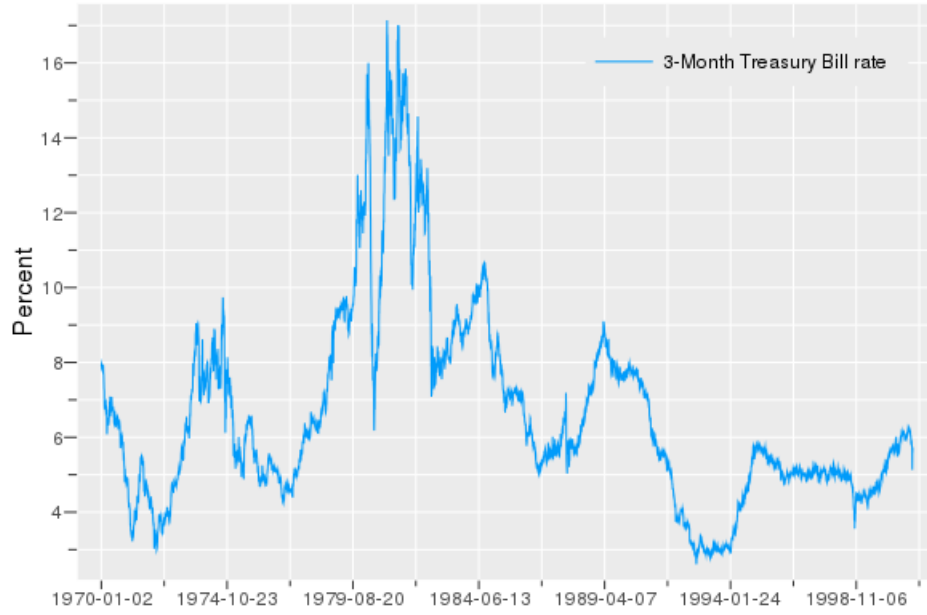
Figure 6.6: Daily values of 3-Month Treasury Bill rates from 2 Jan 1970 to 29 Dec 2000.

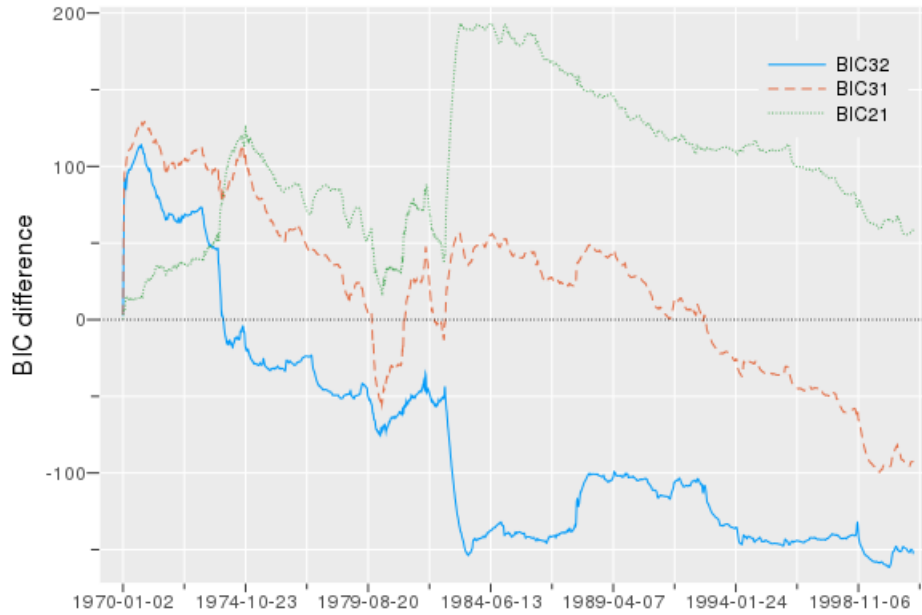with the empirical studies in Chapman and Pearson (2000); Durham (2003); Dellaportas et al. (2006).

Figure 6.7: Online estimation of BIC differences. The solid line stands for $BIC(\mathcal{M}_{32})$, the dashed line for $BIC(\mathcal{M}_{31})$, and the dotted line for $BIC(\mathcal{M}_{21})$.

7 Conclusions and Future Directions

We have introduced an online particle smoothing methodology for discretely observed (jump) diffusions with intractable transition densities. Our approach overcomes such intractability by formulating the problem on pathspace, thus delivering an algorithm that – besides regulatory conditions – requires only the weak invertibility Assumption 2. Thus, we have covered a rich family of SDE models, when related literature imposes quite strong restrictions. Combining our online smoothing algorithm with a Robbins-Monro-type approach of Recursive Maximum-Likelihood, we set up an online stochastic gradient-ascent for the likelihood function of the SDEs under consideration. The algorithm provides a wealth of interesting output, that can provide a lot of useful insights in statistical applications. The numerical examples show a lot of promise for the performance of the methodology. Our framework opens up a number of routes for insights and future research, including the ones described below.

(i) In the case of SDEs of jumps, we have focused on jump dynamics driven by compound Poisson processes. There is great scope for generalisation here, and one can extend the algorithm to different cases of jump processes, also characterised by more complex dependencies between the jumps and the paths of the solution of the SDE, $X = \{X_t\}$. Extensions to time-inhomogeneous cases with $b(v) = b(t, v)$, $\sigma(v) = \sigma(t, v)$, are immediate; we have chosen the time-homogeneous models only for purposes of presentation simplicity.

(ii) Since the seminal work of Delyon and Hu (2006), more 'tuned' auxiliary bridge processes have appeared in the literature, see e.g. the works of Schauer et al. (2017); van der Meulen and Schauer (2017). Indicatively, the work in Schauer et al. (2017) considers bridges of the form (in one of the many options they consider),

$$d\tilde{X}_t = \left\{ b(\tilde{X}_t) + \Sigma(\tilde{X}_t)\Sigma^{-1}(x')\frac{x' - \tilde{X}_t}{T - t} \right\}dt + \sigma(\tilde{X}_t)dW_t. \tag{7.1}$$

Auxiliary bridge processes that are closer in dynamics to the diffusion bridges of the given signal are expected to reduce the variability of Monte-Carlo algorithm,

thus progress along the above direction can be immediately incorporated in our methodology and improve its performance. For instance, as noted in Schauer et al. (2017), use of (7.1) will give a Radon-Nikodym derivative where stochastic integrals cancel out. Such a setting is known to considerably reduce the variability of Monte-Carlo methods, see e.g. the numerical examples in Durham and Gallant (2002) and the discussion in (Papaspiliopoulos and Roberts, 2012, Section 4).

(iii) The exact specification of the recursion used for the online estimation of unknown parameters is in itself an problem of intensive research in the field of stochastic optimisation. One would ideally aim for the recursion procedure to provide parameter estimates which are as close to the unknown parameter as the data (considered thus far) permit. In our case, we have used a fairly 'vanilla' recursion, maybe with the exception of the `Adam` variation. E.g., recent works in the Machine Learning community have pointed at the use of 'velocity' components in the recursion to speed up convergence, see, e.g., Sutskever et al. (2013); Yuan et al. (2016).

(iv) We mentioned in the main text several modifications that can improve algorithmic performance: dynamic resampling, stratified resampling, non-blind proposals in the filtering steps, choice of auxiliary processes. Parallelisation and use of HPC are obvious additions in this list.

(v) We stress that the algorithm involves a filtering step, and a step that approximates the values of the instrumental function. These two procedures should be thought of separately. One reason for including two approaches in the case of jump diffusions (Constructs 1 and 2) is indeed to highlight this point. The two Constructs are identical in terms of the filtering part. Construct 1 incorporates in $\mathbf{x}'$ the location of the path at all times of jumps; thus, when the algorithm 'mixes' all pairs of $\{x_{k-1}^{(j)}\}$, $\{\mathbf{x}_k^{(i)}\}$, at the update of the instrumental function (see Step (iv) of Algorithm 2), many of such pairs can be incompatible (simply consider any such pair with $i \neq j$, when $\{\mathbf{x}_k^{(i)}\}$ has in fact been generated conditionally on $\{x_{k-1}^{(i)}\}$). Such an effect is even stronger in the case of the standard algorithm applied in the motivating example in the

Introduction, and partially explains the inefficiency of that algorithm. In contrast, in Construct 2, $\mathbf{x}'$ contains less information about the underlying paths, thereby improving the compatibility of pairs selected from particle populations $\{x_{k-1}^{(j)}\}$, $\{\mathbf{x}_k^{(i)}\}$, thus – not surprisingly – Construct 2 seems more effective than Construct 1.

(vi) The experiments included in Section 6 of the paper aim to highlight the *qualitative* aspects of the introduced approach. The key property of robustness to mesh-refinement underlies all numerical applications, and is explicitly illustrated in the experiment in Section 6.2, and within Figure 6.4. In all cases, the code was written in `Julia`, was not parallelised or optimised and was run on a PC, with executions times that were quite small due to the online nature of the algorithm (the most expensive experiment took about 30 minutes). We have avoided a comparison with potential alternative approaches as we believe this would obscure the main messages we have tried to convey within the numerical results section. Such a detailed numerical study can be the subject of future work.

Acknowledgments

References

Aït-Sahalia, Y. (1996). Testing continuous-time models of the spot interest rate. *The Review of Financial Studies*, 9(2):385–426. 32

Aït-Sahalia, Y. (2002). Maximum likelihood estimation of discretely sampled diffusions: a closed-form approximation approach. *Econometrica*, 70(1):223–262. 4

Aït-Sahalia, Y. (2008). Closed-form likelihood expansions for multivariate diffusions. *The Annals of Statistics*, 36(2):906–937. 4, 15

Aït-Sahalia, Y., Mykland, P. A., and Zhang, L. (2005). How often to sample a continuous-time process in the presence of market microstructure noise. *The Review of Financial Studies*, 18(2):351–416. 3, 4

Aït-Sahalia, Y. and Yu, J. (2008). High frequency market microstructure noise estimates and liquidity measures. Technical report, National Bureau of Economic Research. 3

Andrieu, C., Doucet, A., and Holenstein, R. (2010). Particle Markov Chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342. 5

Azuma, K. (1967). Weighted sums of certain dependent random variables. *Tohoku Mathematical Journal, Second Series*, 19(3):357–367. 44

Bali, T. G. and Wu, L. (2006). A comprehensive analysis of the short-term interest-rate dynamics. *Journal of Banking & Finance*, 30(4):1269–1290. 32

Beskos, A., Papaspiliopoulos, O., Roberts, G. O., and Fearnhead, P. (2006). Exact and computationally efficient likelihood-based estimation for discretely observed diffusion processes (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):333–382. 4, 6

Cappe, O. (2009). Online sequential Monte Carlo EM algorithm. In *2009 IEEE/SP 15th Workshop on Statistical Signal Processing*. 8

Cappé, O., Moulines, E., and Ryden, T. (2005). *Inference in Hidden Markov Models*. Springer Science & Business Media. 3

Carpenter, J., Clifford, P., and Fearnhead, P. (1999). Improved particle filter for nonlinear problems. *IEE Proceedings-Radar, Sonar and Navigation*, 146(1):2–7. 21

Chapman, D. A. and Pearson, N. D. (2000). Is the short rate drift actually nonlinear? *The Journal of Finance*, 55(1):355–388. 33

Chib, S. and Kuffner, T. A. (2016). Bayes factor consistency. *arXiv preprint, arXiv:1607.00292*. 30

Chib, S., Pitt, M. K., and Shephard, N. (2004). Likelihood based inference for diffusion driven models. 13

Chopin, N., Jacob, P. E., and Papaspiliopoulos, O. (2013). SMC2: an efficient algorithm for sequential analysis of state space models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75(3):397–426. 5

Dahlhaus, R. and Neddermeyer, J. C. (2010). Particle filter-based on-line estimation of spot volatility with nonlinear market microstructure noise models. *arXiv preprint arXiv:1006.1860.* 8

Del Moral, P., Doucet, A., and Singh, S. (2010). Forward smoothing using sequential Monte Carlo. *arXiv preprint arXiv:1012.5390.* 6, 8, 9, 10, 11, 12, 24

Dellaportas, P., Friel, N., and Roberts, G. O. (2006). Bayesian model selection for partially observed diffusion models. *Biometrika*, 93(4):809–825. 15, 32, 33

Delyon, B. and Hu, Y. (2006). Simulation of conditioned diffusion and application to parameter estimation. *Stochastic Processes and their Applications*, 116(11):1660–1675. 13, 14, 15, 18, 34

Douc, R., Garivier, A., Moulines, E., and Olsson, J. (2011). Sequential Monte Carlo smoothing for general state space hidden Markov models. *The Annals of Applied Probability*, 21(6):2109–2145. 44

Douc, R., Moulines, E., and Stoffer, D. (2014). *Nonlinear time series: Theory, methods and applications with R examples.* Chapman and Hall/CRC. 3

Doucet, A., Godsill, S., and Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208. 9

Durham, G. B. (2003). Likelihood-based specification analysis of continuous-time models of the short-term interest rate. *Journal of Financial Economics*, 70(3):463–487. 32, 33

Durham, G. B. and Gallant, A. R. (2002). Numerical techniques for maximum likelihood estimation of continuous-time diffusion processes. *Journal of Business & Economic Statistics*, 20(3):297–338. 4, 35

Eguchi, S. and Masuda, H. (2018). Schwarz type model comparison for LAQ models. *Bernoulli*, 24(3):2278–2327. 30

Eraker, B., Johannes, M., and Polson, N. (2003). The impact of jumps in volatility and returns. *The Journal of Finance*, 58(3):1269–1300. 30, 31

Fearnhead, P., Papaspiliopoulos, O., and Roberts, G. O. (2008). Particle filters for partially observed diffusions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(4):755–777. 6, 11, 15

Gloaguen, P., Étienne, M.-P., and Le Corff, S. (2018). Online sequential Monte Carlo smoother for partially observed diffusion processes. *EURASIP Journal on Advances in Signal Processing*, 2018(1):9. 6

Golightly, A. and Wilkinson, D. J. (2008). Bayesian inference for nonlinear multivariate diffusion models observed with error. *Computational Statistics & Data Analysis*, 52(3):1674–1693. 13

Gonçalves, F. B. and Roberts, G. O. (2014). Exact simulation problems for jump-diffusions. *Methodology and Computing in Applied Probability*, 16(4):907–930. 17

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT press. 25

Gordon, N. J., Salmond, D. J., and Smith, A. F. (1993). Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE Proceedings F (Radar and Signal Processing)*, volume 140, pages 107–113. IET. 8, 12

Hansen, L. P. and Scheinkman, J. A. (1993). Back to the future: Generating moment implications for continuous-time Markov processes. Technical report, National Bureau of Economic Research. 4

Hansen, P. R. and Lunde, A. (2006). Realized variance and market microstructure noise. *Journal of Business & Economic Statistics*, 24(2):127–161. 3

Heston, S. L. (1993). A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The Review of Financial Studies*, 6(2):327–343. 28

Johannes, M. S., Polson, N. G., and Stroud, J. R. (2009). Optimal filtering of jump diffusions: Extracting latent states from asset prices. *The Review of Financial Studies*, 22(7):2759–2799. 30, 31

Jones, C. S. (2003). Nonlinear mean reversion in the short-term interest rate. *The Review of Financial Studies*, 16(3):793–843. 32

Kalogeropoulos, K., Roberts, G. O., and Dellaportas, P. (2010). Inference for stochastic volatility models using time change transformations. *The Annals of Statistics*, 38(2):784–807. 15

Kantas, N., Doucet, A., Singh, S. S., Maciejowski, J., and Chopin, N. (2015). On particle methods for parameter estimation in state-space models. *Statistical Science*, 30(3):328–351. 9

Kessler, M. (1997). Estimation of an ergodic diffusion from discrete observations. *Scandinavian Journal of Statistics*, 24(2):211–229. 4

Kessler, M., Lindner, A., and Sorensen, M. (2012). *Statistical methods for stochastic differential equations.* Chapman and Hall/CRC. 4

Kessler, M. and Sørensen, M. (1999). Estimating equations based on eigenfunctions for a discretely observed diffusion process. *Bernoulli*, 5(2):299–314. 4

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980.* 24, 25

Kitagawa, G. (1987). Non-gaussian state space modeling of nonstationary time series. *Journal of the American Statistical Association*, 82(400):1032–1041. 9

Kitagawa, G. (1996). Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5(1):1–25. 21

Kloeden, P. E. and Platen, E. (2013). *Numerical solution of stochastic differential equations*, volume 23. Springer Science & Business Media. 5, 8

LeGland, F. and Mevel, L. (1997). Recursive estimation in hidden Markov models. In *Decision and Control, 1997, Proceedings of the 36th IEEE Conference*, volume 4, pages 3468–3473. IEEE. 23

Nishii, R. (1988). Maximum likelihood principle and model selection when the true model is unspecified. *Journal of Multivariate Analysis*, 27(2):392–403. 30

Øksendal, B. and Sulem, A. (2007). *Applied Stochastic Control of Jump Diffusions*. Springer Science & Business Media. 3

Olsson, J. and Westerborn, J. (2017). Efficient particle-based online smoothing in general hidden Markov models: the PaRIS algorithm. *Bernoulli*, 23(3):1951–1996. 44

Papaspiliopoulos, O. and Roberts, G. (2012). Importance sampling techniques for estimation of diffusion models. *Statistical methods for stochastic differential equations*, 124:311–340. 14, 35

Papaspiliopoulos, O., Roberts, G. O., and Stramer, O. (2013). Data augmentation for diffusions. *Journal of Computational and Graphical Statistics*, 22(3):665–688. 14

Poyiadjis, G., Doucet, A., and Singh, S. S. (2011). Particle approximations of the score and observed information matrix in state space models with application to parameter estimation. *Biometrika*, 98(1):65–80. 6, 23

Reddi, S. J., Kale, S., and Kumar, S. (2019). On the convergence of Adam and beyond. *arXiv preprint arXiv:1904.09237*. 25

Roberts, G. O. and Stramer, O. (2001). On inference for partially observed nonlinear diffusion models using the Metropolis–Hastings algorithm. *Biometrika*, 88(3):603–621. 13

Särkkä, S. and Sottinen, T. (2008). Application of Girsanov theorem to particle filtering of discretely observed continuous-time non-linear systems. *Bayesian Analysis*, 3(3):555–584. 6, 11

Schauer, M., Van Der Meulen, F., and Van Zanten, H. (2017). Guided proposals for simulating multi-dimensional diffusion bridges. *Bernoulli*, 23(4A):2917–2950. 34, 35

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, 6(2):461–464. 30

Sin, C.-Y. and White, H. (1996). Information criteria for selecting possibly misspecified parametric models. *Journal of Econometrics*, 71(1-2):207–225. 30

Stanton, R. (1997). A nonparametric model of term structure dynamics and the market price of interest rate risk. *The Journal of Finance*, 52(5):1973–2002. 32

Ströjby, J. and Olsson, J. (2009). Efficient particle-based likelihood estimation in partially observed diffusion processes. *IFAC Proceedings Volumes*, 42(10):1364–1369. 6

Sutskever, I., Martens, J., Dahl, G., and Hinton, G. (2013). On the importance of initialization and momentum in Deep Learning. In *International Conference on Machine Learning*, pages 1139–1147. 35

Tadic, V. Z. and Doucet, A. (2018). Asymptotic properties of recursive maximum likelihood estimation in non-linear state-space models. *arXiv preprint arXiv:1806.09571*. 23

van der Meulen, F. and Schauer, M. (2017). Bayesian estimation of discretely observed multi-dimensional diffusion processes using guided proposals. *Electronic Journal of Statistics*, 11(1):2358–2396. 34

Wagner, W. (1989). Undiased Monte Carlo estimators for functionals of weak solutions of stochastic differential equations. *Stochastics: An International Journal of Probability and Stochastic Processes*, 28(1):1–20. 4

Yonekura, S., Beskos, A., and Singh, S. (2018). Asymptotic analysis of model selection criteria for general hidden Markov models. *arXiv preprint arXiv:1811.11834*. 30

Yuan, K., Ying, B., and Sayed, A. H. (2016). On the influence of momentum acceleration on online learning. *The Journal of Machine Learning Research*, 17(1):6602–6667. 35

A Proof of Proposition 1

Proof. We have the integral,

$$\int T_{\theta,n}(\mathbf{x}_n)\, p_\theta(d\mathbf{x}_n|y_{0:n}) = \int S_\theta(\mathbf{x}_{0:n})\, p_\theta(d\mathbf{x}_{0:n-1}|y_{0:n-1}, \mathbf{x}_n) \otimes p_\theta(d\mathbf{x}_n|y_{0:n}). \qquad (A.1)$$

Also, simple calculations give,

$$p_\theta(d\mathbf{x}_{0:n-1}|y_{0:n-1},\mathbf{x}_n) \otimes p_\theta(\mathbf{x}_n|y_{0:n})$$
$$\equiv p_\theta(d\mathbf{x}_{0:n-1}|y_{0:n}, \mathbf{x}_n) \otimes p_\theta(\mathbf{x}_n|y_{0:n}) = p_\theta(d\mathbf{x}_{0:n}|y_{0:n}).$$

Using this expression in the integral on the right side of (A.1) completes the proof. $\qquad\square$

B Proof of Proposition 2

Proof. Simply note that,

$$\int T_{\theta,n-1}(\mathbf{x}_{n-1})\, p_\theta(d\mathbf{x}_{n-1}|y_{0:n-1}, \mathbf{x}_n) =$$
$$= \int S_{\theta,n-1}(\mathbf{x}_{0:n-1})\, p_\theta(d\mathbf{x}_{0:n-2}|y_{0:n-2}, \mathbf{x}_{n-1})\, p_\theta(d\mathbf{x}_{n-1}|y_{0:n-1}, \mathbf{x}_n).$$

Then, we have that,

$$p_\theta(d\mathbf{x}_{0:n-2}|y_{0:n-2}, \mathbf{x}_{n-1})\, p_\theta(d\mathbf{x}_{n-1}|y_{0:n-1}, \mathbf{x}_n)$$
$$\equiv p_\theta(d\mathbf{x}_{0:n-2}|\mathbf{x}_{n-1}, y_{0:n-1}, \mathbf{x}_n)\, p_\theta(d\mathbf{x}_{n-1}|y_{0:n-1}, \mathbf{x}_n)$$
$$\equiv p_\theta(d\mathbf{x}_{0:n-1}|y_{0:n-1}, \mathbf{x}_n).$$

Replacing the probability measure on the left side of the above equality with its equal on the right side, and using the latter in the integral above completes the proof for the first equation in the statement of the proposition. The second equation follows from trivial use of Bayes rule. $\qquad\square$

C Proof of Proposition 3

Proof. Part (i) follows via the same arguments as in Olsson and Westerborn (2017, Corollary 2), based on Azuma (1967) and Douc et al. (2011, Lemma 4). For part (ii), one can proceed as follows. Given $n \geq 0$, we define the event $A_N(1/j) := \{|\mathcal{S}_{\theta,n} - \hat{\mathcal{S}}_{\theta,n}^N| > \frac{1}{j}\}$, where we added superscript N to stress the dependency of the estimate on the number of particles. Then,

$$\text{Prob}\left[\lim_{N\to\infty} \hat{\mathcal{S}}_{\theta,n}^N = \mathcal{S}_{\theta,n}\right] = 1 - \text{Prob}\left[\cup_{j=1}^\infty \limsup_{N\to\infty} A_N(1/j)\right]$$
$$\geq 1 - \sum_{j=1}^\infty \text{Prob}\left[\limsup_{N\to\infty} A_N(1/j)\right]. \qquad (\text{C.1})$$

From (i), we get $\text{Prob}\left[A_N(1/j)\right] \leq b_n e^{-c_n N\left(\frac{1}{j}\right)^2}$, so $\sum_{N=1}^\infty \text{Prob}\left[A_N(1/j)\right] < \infty$. The Borel–Cantelli lemma gives $\text{Prob}\left[\limsup_{N\to\infty} A_N(1/j)\right] = 0$, therefore the result follows from (C.1). $\qquad\square$