Online Smoothing for Diffusion Processes Observed with Noise

Shouto Yonekura and Alexandros Beskos

Department of Statistical Science, University College London, Alan Turing Institute, London, UK

ABSTRACT

We introduce a methodology for online estimation of smoothing expectations for a class of additive functional processes, 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.

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 $X_{t-} := \lim_{s \uparrow t} X_s$,

$$dx_t = b(X_{t-})dt + \sigma(X_{t-})dW_t + df_t, \quad X_0 = x_0 \in \mathbb{R}^{d_x}, \quad t \geq 0.$$  

(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 $b = b_0 : \mathbb{R}^{d_x} \mapsto \mathbb{R}^{d_x}$ and coefficient matrix $\sigma = \sigma_0 : \mathbb{R}^{d_x} \mapsto \mathbb{R}^{d_x \times d_w}$, for parameter $\theta \in \mathbb{R}^p$, $p \geq 1$. Let $\{N_t; t \geq 0\}$ be a Poisson process with intensity function $\lambda_0(\cdot)$, and $\{\xi_k\}_{k \geq 1}$ iid sequence of random variables with Lebesgue density $h_0(\cdot)$; the càdlàg process $J$ is determined as $J_t = J_{0,\theta} = \sum_{i=1}^{N_t} \xi_i$. We work under standard assumptions (e.g., linear growth, Lipschitz continuity for $b, \sigma$) that guarantee a unique global solution of (1.1), in a weak or strong sense; see, for example, 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 $\Delta := t_{i-} - t_0$. 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 $s \in [t_{i-}, t_i], 0 < i \leq n$. We assume

$$\left[ Y_{t_i} \mid \{Y_{t_j}; j < i\}, \{X_s; s \in [0, t_i]\} \right] \sim g_0(dy_{t_i} \mid Y_{t_{i-}}, \mathcal{F}_i),$$

$$0 \leq i \leq n,$$  

(1.2)
for conditional distribution \( g_0 (\cdot | Y_{t-1}, \mathcal{F}_t) \) on \( \mathbb{R}^d_y \), \( d_y \geq 1 \), under convention \( Y_{t-1} = y_{t-1} = \emptyset \). We write,

\[
[x_i | x_{i-1}] \sim f_0(dx_i | x_{i-1}), \quad (1.3)
\]

where \( f_0(dx_i | x_{i-1}) \) is the transition distribution of the driving SDE process \((1.1)\). We consider the density functions of 

\[
g_0(dy_j | y_{t-1}, \mathcal{F}_t) = f_0(dy_j | x_{t-1}, \mathcal{F}_t),
\]

and – with some abuse of notation – we write \( g_0(dy_j | y_{t-1}, \mathcal{F}_t) = g_0(y_j | y_{t-1}, \mathcal{F}_t)dy_j \), \( f_0(dx_i | x_{i-1}) = f_0(x_i | x_{i-1}, \mathcal{F}_i)dx_i \), where \( dy_j, dx_i \) denote Lebesgue measures. Our work develops under the following regime.

**Assumption 1.** The transition density \( f_0(\cdot | \cdot) \) is intractable; the density \( g_0(y' | y, \mathcal{F}) \) is analytically available – for appropriate \( x', y, \mathcal{F} \) consistent with preceding definitions.

The intractability of the transition density \( f_0(\cdot | \cdot) \) will pose challenges for the main problems this article aims to address.

Models defined via \((1.1)-(1.3)\) are extensively used, for example, in finance and econometrics, for instance for capturing the market microstructure noise, see Aït-Sahalia, Mykland, and Zhang (2005), Aït-Sahalia and Yu (2008), and Hansen and Lunde (2006). The above setting belongs to the general class of HMMs, with a signal defined in continuous-time. See Cappé, Moulines, and Ryden (2005) and Douc, Moulines, and Stoffer (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 (Kessler 1997; Aït-Sahalia 2002, 2008; Aït-Sahalia, Mykland, and Zhang 2005); 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, for example, Kessler, Lindner, and Sorensen (2012) for a detailed review.

For a given sequence \( \{a_{ij}\}_{i,j} \), we use the notation \( a_{ij} := (a_j, \ldots, a_i) \), 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 modeling 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,

\[
S_{\theta,n} = \int S_\theta(x_{0:n})p_\theta(dx_{0:n} | y_{0:n}), \quad n \geq 1, \quad (1.4)
\]

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

\[
S_\theta(x_{0:n}) = \sum_{k=0}^n S_{\theta,k}(x_{k-1}, x_k), \quad (1.5)
\]

under the conventions \( x_{-1} = \emptyset, x_0 = x_0 \). The bold type notation \( x_k, k \geq 0 \), is reserved for carefully defined variables involving elements of the infinite-dimensional pathspace. The specific construction of \( x_k \) will depend on the model at hand, and will be explained in the main part of the article. 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, for example, (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 favorable algorithmic characteristic, when contrasted with alternative algorithms with similar objectives, such as, for example, Particle MCMC (Andrieu, Doucet, and Holenstein 2010), or SMC\(^2\) (Chopin, Jacob, and Papaspiliopoulos 2013).

(iii) In this work, we will not characterize 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-Matuyama one, or some other Taylor scheme, see, for example, 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,

\[
\Sigma_\theta(\nu) := \sigma_\theta(\nu)\sigma_\theta(\nu)^\top \in \mathbb{R}^{d_x \times d_x},
\]

is invertible, for all relevant \( \nu, \theta \).
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 \( f_0(x'|x) \), is given in Del Moral, Doucet, and Singh (2010) and Poyiadjis, Doucet, and Singh (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, Papaspiliopoulos, and Roberts (2008), Ströby and Olsson (2009) and Gloopuen, Étienne, and Le Corff (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: Figure 1 (top-panel) shows estimates of the score function, evaluated at the true parameter value \( \theta = \theta^\dagger \), for parameter \( \theta^\dagger \) of the Ornstein–Uhlenbeck (O–U) process, \( dX_t = \theta_1(\theta_2 - X_t)dt + \theta_3dW_t, X_0 = 0.0 \), for \( n = 100 \) observations \( y_i = x_i + \epsilon_i, \epsilon_i \sim \mathcal{N}(0, 0.1^2) \), \( 1 \leq i \leq n \). Data were simulated from \( \theta^\dagger = (0.5, 0.0, 0.4) \) with an Euler–Maruyama scheme of \( M^\dagger \) = 1000 grid points per unit of time. Figure 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 \( M = 10, 50, 100, 200 \)—that is, as it approaches the "true" resolution used for the data generation. In contrast, Figure 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.

The rest of the article is organized 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, Doucet, and Singh (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, Salmond, and Smith 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 \( p_0(dx_0) \), transition density \( f_0(x'|x) \), and likelihood \( g_0(y|x) \), for appropriate \( x, x' \in \mathbb{R}^d, y \in \mathbb{R}^d \), and review the online algorithm developed in Del Moral, Doucet, and Singh (2010) for this setting. Implementation of the bootstrap filter provides an approximation of the smoothing distribution \( p_0(dx_{0:n}|y_{0:n}) \) by following the genealogy of the particles. This method is studied, for example, in Cappe (2009) and 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 \( p_0(dx_{0:n}|y_{0:n}) \), in the sense that we have the estimate,

\[
\tilde{p}_0(dx_{0:n}|y_{0:n}) = \sum_{i=1}^N W_n^{(i)} \delta_{x_{0:n}^{(i)}}(dx_{0:n}), \quad \sum_{i=1}^N W_n^{(i)} = 1, \tag{2.1}
\]

with \( \delta_{x_{0:n}^{(i)}}(dx_{0:n}) \) the Dirac measure with an atom at \( x_{0:n}^{(i)} \). Then, replacing \( p_0(dx_{0:n}|y_{0:n}) \) 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 \( O(N) \), it typically suffers from the well-documented path-degeneracy problem—as illustrated via theoretical results or numerically (Del Moral, Doucet, and Singh 2010; Kantas et al. 2013). That is, as \( n \) increases, the particles representing \( p_0(dx_{0:n}|y_{0:n}) \) 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, Godsill, and Andrieu (2000). We briefly review the method here, following closely the notation and development in Del Moral, Doucet, and Singh (2010). In the forward direction, assume that a filtering algorithm (e.g., bootstrap) has provided a particle approximation of the filtering distribution \( p_0(dx_{k-1}|y_{0:k-1}) \)—assuming a relevant \( k \)—

\[
\tilde{p}_0(dx_{k-1}|y_{0:k-1}) = \sum_{i=1}^N W_{k-1}^{(i)} \delta_{x_{k-1}^{(i)}}(dx_{k-1}), \tag{2.2}
\]

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 \( p_0(dx_k|y_{0:n}) \),

\[
\tilde{p}_0(dx_k|y_{0:n}) = \sum_{i=1}^N W_n^{(i)} \delta_{x_k^{(i)}}(dx_k). \tag{2.3}
\]

One has that (Kitagawa 1987),

\[
p_0(dx_{k-1}|y_{0:n}) = \tilde{p}_0(dx_{k-1}|y_{0:n}) = \tilde{p}_0(dx_{k-1}|x_k, y_{0:k-1}) = \tilde{p}_0(dx_k|y_{0:n}) \otimes \tilde{p}_0(dx_{k-1}|y_{0:k-1}) \,
\]

\[
\int f_0(x_k|x_{k-1})p_0(x_{k-1}|y_{0:k-1})dx_{k-1}. \tag{2.4}
\]
Using (2.2)–(2.3), and based on Equation (2.4), we obtain the approximation,
\[
\hat{p}_\theta(dx_{k-1}|y_{0:n}) = \sum_{i=1}^{N} W_{k|i}^{(i)} \delta_{x_{k-1}^{(i)}}(dx_{k-1}) / \sum_{i=1}^{N} W_{k|i}^{(i)},
\]
for the normalized weights,
\[
W_{k-1|i}^{(i)} \propto \frac{\int \hat{p}_\theta(dx_{k-1}|y_{0:n})}{\sum_{j=1}^{N} \int \hat{p}_\theta(dx_{k-1}|y_{0:n}) / \sum_{i=1}^{N} W_{k|i}^{(i)}},
\]
for 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,
\[
\hat{p}_\theta(dx_{k-1}|y_{0:n}) = \sum_{i=1}^{N} W_{k-1|i}^{(i)} \delta_{x_{k-1}^{(i)}}(dx_{k-1}),
\]
for the normalized weights,
\[
W_{k-1|i}^{(i)} \propto \frac{\int \hat{p}_\theta(dx_{k-1}|y_{0:n})}{\sum_{j=1}^{N} \int \hat{p}_\theta(dx_{k-1}|y_{0:n}) / \sum_{i=1}^{N} W_{k|i}^{(i)}},
\]
Notice that—in this version of FFBS—the same particles \( \{x_{k-1}^{(i)}\}_{i=1}^{N} \) are used in both directions (the ones before resampling at the forward filter), but with different weights.

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

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

Notice that, first,

\[ S_{\theta,n} = \int T_{\theta,n}(x_n)p_\theta(dx_n|y_{0:n}). \]

We also have that,

\[ \int \left[ T_{\theta,n-1}(x_{n-1}) + s_{\theta,n}(x_{n-1}, x_n) \right]p_\theta(dx_{n-1}|y_{0:n-1}, x_n) \]
\[ \equiv \int f_\theta(x_n|x_{n-1})p_\theta(dx_{n-1}|y_{0:n-1}), \quad (2.6) \]

see Proposition 2.1 of Del Moral, Doucet, and Singh (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 summarized 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 \( f_\theta(\cdot|x) \) is considered intractable, thus, Algorithm 1—apart from serving as a review of the method in Del Moral, Doucet, and Singh (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 \( f_\theta(\cdot|x) \) 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, Papaspiliopoulos, and Roberts (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 fact that our approach appears as the relatively most “natural” extension (compared to alternatives) of the standard discrete-time algorithm of Del Moral, Doucet, and Singh (2010).

**Algorithm 1** Online Forward-Only Smoothing (Del Moral, Doucet, and Singh 2010)

1. Initialize particles \( \{x_0^{(i)}, W_0^{(i)} \}_{i=1}^N \), with \( x_0^{(i)} \equiv x_0 \sim p_\theta(dx_0) \), \( W_0^{(i)} \propto g_\theta(y_0|x_0^{(i)}) \), and functionals \( \tilde{T}_{\theta,0}(x_0^{(i)}) = s_{\theta,0}(x_0^{(i)}) \), for \( 1 \leq i \leq N \).
2. 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 \( \tilde{T}_{\theta,n-1}(x_{n-1}^{(i)}) \) of \( T_{\theta,n-1}(x_{n-1}^{(i)}) \), for \( 1 \leq i \leq N \).
3. At time \( n \), sample \( x_n^{(i)} \), for \( 1 \leq i \leq N \), from the mixture (Gordon, Salmond, and Smith 1993),

\[ x_n^{(i)} \sim \tilde{p}_\theta(x_n|x_{n-1}^{(i)}), \quad 1 \leq i \leq N. \]

4. Then set, for \( 1 \leq i \leq N \),

\[ \tilde{T}_{\theta,n}(x_n^{(i)}) = \sum_{j=1}^N W_{n-1}^{(j)}f_\theta(x_n^{(i)}|x_{n-1}^{(j)}). \]

5. Obtain an estimate of \( S_{\theta,n} \) as,

\[ \hat{S}_{\theta,n} = \sum_{i=1}^N W_n^{(i)} \tilde{T}_{\theta,n}(x_n^{(i)}). \]

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 \( p_\theta(dx'|x) \), for a properly defined variate \( x' \) that involves information about the continuous-time path for moving from \( x \) to \( x' \) within time \( \Delta \). We will give the complete algorithm in Section 4. In this section, we prepare the ground via carefully determining \( x' \) 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, that is, of dynamics,

\[ dX_t = b_\theta(X_t)dt + \sigma_\theta(X_t)dW_t. \]

We adopt an approach motivated by techniques used for MCMC algorithms (Roberts and Stramer 2001; Chib, Pitt, and Shephard 2004; Golightly and Wilkinson 2008). Assume we are given starting point \( x \in \mathbb{R}^d \), ending point \( x' \in \mathbb{R}^d \), 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

\[ \{X_t; t \in [0, T]\} \mid X_0 = x, X_T = x'. \]

Let \( \mathbb{P}_{x,x'} \) 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,


\[ d\tilde{X}_t = \left\{ b_0(\tilde{X}_t) + \frac{x' - \tilde{X}_t}{T-t} \right\} dt + \sigma_0(\tilde{X}_t) dW_t, \]

\[ \tilde{X}_0 = x, \quad t \in [0, T], \] (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 \( F^{\theta,x,x'}_\theta, \mathbb{Q}_{\theta,x,x} \) 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 \( X = \{ \tilde{X}_t; t \in [0, T] \} \), 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—it is invertible,

\[ Z \mapsto X =: F_\theta(Z;x,x'), \quad Z = F^{-1}_\theta(X;x,x'). \] (3.4)

More analytically, \( F^{-1}_\theta \) is given via the transform,

\[ dZ_t = \sigma_0(X_t)^{-1} \left\{ dX_t - b_0(X_t) dt - \frac{x' - X_t}{T-t} dt \right\}. \]

In this case we define,

\[ x' := (x', Z), \]

and the probability measure of interest is,

\[ p_\theta(dx'|x) := f_\theta(dx'|x) \otimes p_\theta(dZ|x',x). \] (3.5)

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

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

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

Such a Radon-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 normalizing constant which is important in our case, as it involves the parameter \( \theta \) in our applications later in the article, we aim to infer about \( \theta \). Papaspiliopoulos, Roberts, and Stramer (2013) and 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 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) \( v \mapsto \sigma_0(v) \) is in \( C^2_b \) (i.e., twice continuously differentiable and bounded, with bounded first and second derivatives), and it is invertible, with bounded inverse;
(iii) \( v \mapsto b_0(v) \) is locally Lipschitz, locally bounded.

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

\[ \frac{d\mathbb{P}_{\theta,x,x'}}{d\mathbb{Q}_{\theta,x,x'}}(X) = \frac{|\Sigma_0(x')|^{1/2}}{|\Sigma_0(x)|^{1/2}} \times \frac{N(x';x,T\Sigma_0(x))}{f_\theta(x'|x)} \times \psi_0(X;x,x'), \] (3.6)

where \( \mathcal{N}(\nu;\mu,V) \) is the density function of the Gaussian law on \( \mathbb{R}^d \) with mean \( \mu \), variance \( V \), evaluated at \( \nu, \| \cdot \| \) is matrix determinant, and \( \psi_0(X;x,x') \) is such that,

\[ \log \psi_0(X;x,x') = \int_0^T \left\{ b_0(X_t), \Sigma_0^{-1}(X_t) dX_t \right\} \]

\[-\frac{1}{2} \int_0^T \left\{ b_0(X_t), \Sigma_0^{-1}(X_t) b_0(X_t) dt \right\} \]

\[-\frac{1}{2} \int_0^T \left\{ (x' - X_t), \Sigma_0^{-1}(X_t)(x' - X_t) \right\} \]

\[-\frac{1}{2} \int_0^T \frac{\sigma_0^2(x')}{\Sigma_0^2(x)} dW_t dt. \]

Here, \( \langle \cdot, \cdot \rangle \) denotes the quadratic variation process for semimartingales; also, \( \langle \cdot, \cdot \rangle \) is the standard inner-product on \( \mathbb{R}^d \).

We note that transforms different from (3.4) have been proposed in the literature (Dellaportas, Friel, and Roberts 2006; Kalogeropoulos, Roberts, and Dellaportas 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 \( \theta \). 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, Papaspiliopoulos, and Roberts (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, for example, as in Assumption 3. Thus, continuing from (3.5) we have obtained that,

\[ p_\theta(dx'|x) = f_\theta(F_\theta(Z;x,x');x,x') \times \mathcal{N}(x';x,T\Sigma_0(x)) \times \frac{|\Sigma_0(x')|^{1/2}}{|\Sigma_0(x)|^{1/2}} \]

\[ =: p_\theta(x'|x) \equiv p_\theta(x'|x;T). \] (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 rewrite here for convenience,

\[ dX_t = b_0(X_{t-}) dt + \sigma_0(X_{t-}) dW_t + dJ_t, \]

\[ X_0 = x \in \mathbb{R}^d, \quad t \in [0, T]. \] (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 \( \mathbb{P}_{\theta,x}(-) \) denote the law of the unconditional process (3.8) and \( \mathbb{I}_\theta \) the law of the involved compound Poisson process. We write \( J = \{(t_1,b_1), \ldots, (t_\kappa,b_\kappa)\} \) to denote the jump process, where \( \{t_i\} \) 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 $\otimes_{i=1}^{t+1} \mathbb{L}_{\text{Leb}}^{\otimes dx_i}$.

**Construct One**

We consider the random variate,

$$x' = \left( J, \left( x_{t_{i-1}} \right)^{t+1}_{i=1}, \left( Z(i) \right)^{t+1}_{i=1} \right),$$

under the conventions $x_{t_0} = x$, $x_{t_{k+1}} = x'$, where we have defined,

$$Z(i) = F_{t_i}^{-1}(X(i) ; x_{t_{i-1}}, x_{t_i}),$$

$$X(i) = \{ X \in \mathbb{R} \mid t \in [x_{t_{i-1}}, x_{t_i}) \}, \quad 1 \leq i \leq k + 1.$$

We have that,

$$p_0(dx' | x) := \mathbb{L}_0(dJ) \otimes \left( \otimes_{i=1}^{t+1} \left[ f_0(dx_{t_i-1}|x_{t_{i-1}}, x_{t_i}) \otimes p_0(dZ(i)|x_{t_{i-1}}, x_{t_i}) \right] \right).$$

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

$$x'(i) := (x_{t_{i-1}}, Z(i)),$$

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

$$\frac{f_0(dx_{t_i-1}|x_{t_{i-1}}, x_{t_i}) \otimes p_0(dZ(i)|x_{t_{i-1}}, x_{t_i})}{\mathbb{L}_{\text{Leb}}^{\otimes dx_i}(dx_{t_i-1} \otimes \mathbb{W}(dZ(i)))} = p_0(x'(i)|x_{t_{i-1}}, \tau_i - t_{i-1}, t_i),$$

with the latter density $p_0(x'(i)|x_{t_{i-1}})$ determined as in (3.7), given clear adjustments. Thus, the density of $p_0(dx'|x)$ with respect to the reference measure, $\mu(dx') := \mathbb{L}(dJ) \otimes \left( \otimes_{i=1}^{t+1} \left[ \mathbb{L}_{\text{Leb}}^{\otimes dx_i}(dx_{t_i-1} \otimes \mathbb{W}(dZ(i))) \right] \right)$, is equal to,

$$\frac{p_0(dx'|x)}{\mu(dx')} = e^{-\int_{T}^{T-	au} \lambda_0(t) dt} \prod_{i=1}^{k} \{ \lambda_0(\tau_i) \} \int_{T}^{T-	au} \prod_{i=1}^{k+1} p_0(x'(i)|x_{t_{i-1}}, \tau_i - t_{i-1}).$$

**Construct Two**

We adopt an idea used—for a different problem—in Gonçalves and Roberts (2014). Given $x, x' \in \mathbb{R}^{d_1}$, we define an auxiliary process $\tilde{X}_t$, as follows,

$$d\tilde{X}_t = \left[ b_0(\tilde{X}_t) + \frac{x' - J_T - \tilde{X}_t}{T - t} \right] dt + \sigma_0(\tilde{X}_t) dW_1 + dJ_t,$$

$$\quad X_0 = x,$$

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,

$$(J, Z) \mapsto X = : G_0(J, Z; x, x'), \quad (J, Z) = G_0^{-1}(X; x, x').$$

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—

$$dZ_t = \sigma_0(X_t)^{-1} \left[ dX_t - dJ_t - b_0(X_t) dt - \frac{x' - J_T - X_t + J_t}{T - t} dt \right].$$

We denote by $\overline{P}_{\theta, x, x'}$ 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 $\overline{Q}_{\theta, x, x'}$. Consider the variate,

$$x' = (x', J, Z),$$

so that,

$$p_0(dx'|x) := f_0(dx'|x) \otimes p_0(d(J, Z)|x', x).$$

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

$$p_0(dx'|x) = f_0(x'|x) \times \frac{d\overline{P}_{\theta, x, x'}}{d\overline{Q}_{\theta, x, x'}}(G_0(J; Z; x, x')).$$

Thus, using the parameter-free reference measure, $\mu(dx') := \mathbb{L}_{\text{Leb}}^{\otimes dx} \otimes \mathbb{L} \otimes \mathbb{W}$, one obtains that,

$$p_0(dx'|x) = \frac{f_0(x'|x) \times e^{\int_{0}^{T} \lambda_0(t) dt} \prod_{i=1}^{k} \{ \lambda_0(\tau_i) \} \lambda_0(\tau_1) \lambda_0(\tau_2) \lambda_0(\tau_3) \lambda_0(\tau_4) \lambda_0(\tau_5)}{\prod_{i=1}^{k+1} p_0(x'(i)|x_{t_{i-1}}, \tau_i - t_{i-1})} \times \frac{d\overline{P}_{\theta, x, x'}}{d\overline{Q}_{\theta, x, x'}}(G_0(J; Z; x, x')).$$

(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 $\overline{P}_{\theta, x, x'}^{M}(X_{\delta}, \ldots, X_{(M-1)\delta}| X_0 = x, X_{M\delta} = x')$ denote the time-discretised Lebesgue density of the $(M - 1)$-positions of the conditioned diffusion with law $\overline{P}_{\theta, x, x'}$. Once (3.10) is obtained, a time-discretization approach will give,

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

$$(x_{k-1}, x_k), \quad 1 \leq k \leq n,$$

with $x_k$ as defined in Section 3, that is, containing pathspace elements, and given by an 1–1 transform of $\{X_t; t \in [t_{k-1}, t_k]\}$, such that we can obtain a density for $p_0(x_k|x_{k-1})$ with respect to a reference measure that does not involve $\theta$. Recall that $p_0(dx_k|x_{k-1})$ denotes the probability law for the augmented variable $x_k$ given $x_{k-1}$. We also write the corresponding density as,

$$p_0(x_k|x_{k-1}) := \frac{p_0(dx_k|x_{k-1})}{\mu(dx_k)}.$$

The quantity of interest is now,

$$S_{\theta,n} = \int S_0(x_{0:n}) p_\theta(dx_{0:n}|y_{0:n}), \quad n \geq 1,$$

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

$$S_0(x_{0:n}) = \sum_{k=0}^{n} s_{0:k}(x_{k-1}, x_k),$$

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 $x_k$; 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 $x_k$.

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

$$T_{\theta,n}(x_n) := \int S_0(x_{0:n}) p_\theta(dx_{0:n-1}|y_{0:n-1}, x_n).$$

**Proposition 1.** We have that,

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

**Proof.** See Appendix A.

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

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

$$T_{\theta,n}(x_n) = \int \left[ T_{\theta,n-1}(x_{n-1}) + s_{\theta,n}(x_{n-1}, x_n) \right] p_\theta(dx_{n-1}|y_{0:n-1}, x_n)$$

$$= \int \left[ T_{\theta,n-1}(x_{n-1}) + s_{\theta,n}(x_{n-1}, x_n) \right] p_\theta(dx_{n-1}|y_{0:n-1}, x_n)$$

$$= \frac{\int \left[ T_{\theta,n}(x_{n-1}) + s_{\theta,n}(x_{n-1}, x_n) \right] p_\theta(dx_{n-1}|y_{0:n-1}, x_n)}{\int p_\theta(dx_n|y_{0:n}) p_\theta(dx_{n-1}|y_{0:n-1})}. $$

**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

1. Initialise the particles $\{x_n^{(i)}, W_n^{(i)}\}_{i=1}^{N}$, with $x_0^{(i)} \overset{\text{i.i.d.}}{\sim} p_0(dx_0)$, $W_0^{(i)} \sim g_0(y_0|x_0^{(i)})$, and functionals $\tilde{T}_{\theta,0}(x_0^{(i)}) = s_{\theta,0}(x_0^{(i)})$, where $x_0^{(i)} = x_0^{j}$, $1 \leq i \leq N$.

2. 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 $\tilde{T}_{\theta,n-1}(x_{n-1}^{(i)})$ of $T_{\theta,n-1}(x_{n-1}^{(i)})$, $1 \leq i \leq N$.

3. At time $n$, sample $x_n^{(i)}$ for $1 \leq i \leq N$, from

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

and assign particle weights $W_n^{(i)} \sim g_0(y_n|x_n^{(i)}, \mathcal{F}_n^{(i)})$, $1 \leq i \leq N$.

4. Then set, for $1 \leq i \leq N$,

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

5. Obtain an estimate of $S_{\theta,n}$ as,

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

**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, Clifford, and Fearnhead 1999), stratified resampling (Kitagawa 1996)); dynamic resampling via use of Effective Sample Size; nonblind 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 $X' = (x_{0}, x_{1}, \ldots, x_{M})$, and using, for example, an Euler–Maruyama time-discretisation scheme to obtain the joint density of such an $X'$ given $x = x_0$, a Radon–Nikodym derivative, $p_\theta^M(X'|x)$ on $\mathbb{R}^{M \times d}$, can be obtained with respect to the Lebesgue measure $\mu(dx_0 \times \cdots \times dx_M)$, as a product of $M$ conditionally Gaussian densities. As shown for example, 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 rigor, 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 $X$ and $x$ denote the state spaces of $x$ and $x$, respectively.

(i) For any relevant $y'$, $y$, $F$, $x'$, we have that $g_0(y'|y, F) \equiv g_0(y'|x')$, where the latter is a positive function such that, for any $y$, $\sup_{x \in \mathcal{X}} g_0(y|x) < \infty$.

(ii) $\sup_{(x,x') \in \mathcal{X} \times \mathcal{X}} p_0(x'|x) < \infty$.

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

$$\text{Prob} \left[ |S_{\theta,n} - \hat{S}_{\theta,n}| > \epsilon \right] \leq b_n e^{-c_n n^2}.$$

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

**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, Doucet, and Singh (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,

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

$$\theta_{n+1} = \theta_n + y_{n+1} \{ \nabla \log p_{0,n}(y_{0:n}) - \nabla \log p_{0,n-1}(y_{0:n-1}) \}$$

where $\{ \gamma_k \}$ 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 $\nabla \log p_{0,n}(y_{0:n})$ is that—given a recursive method (in $n$) for the estimation of $\theta \rightarrow \nabla \log p_0(y_{0:n})$ as we describe below and based on the methodology of Algorithm 2—one uses $\theta_{n+1}$ when incorporating $y_{n+1}$, then $\theta_n$ 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 $\theta^*$, as $n \rightarrow \infty$.

Observe that, from Fisher’s identity (see, e.g., Poyiadjis, Doucet, and Singh 2011) we have that,

$$\nabla \log p_0(y_{0:n}) = \int \nabla \log p_0(x_{0:n}, y_{0:n}) p_0(dx_{0:n}|y_{0:n}).$$

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

$$S_0(x_{0:n}) = \nabla \log p_0(x_{0:n}, y_{0:n}) \equiv \sum_{k=0}^{n} \nabla \log p_0(x_k, y_k|x_{k-1}).$$

and,

$$s_{0,k}(x_{k-1}, x_k) \equiv \nabla \log p_0(x_k, y_k|x_{k-1}).$$

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

**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 $[x_n^{(i)}, W_n^{(i)}]_{i=1}^N$ of the “filter” $p_{k,n}(x_n|y_{0:n})$ and estimators $\hat{T}_{\theta,n}(x_n^{(i)})$ of $T_{\theta,n}(x_n^{(i)})$, for $1 \leq i \leq N$.

(ii) Apply the iteration,

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

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

$$x_{n+1}^{(i+1)} \sim \hat{p}_{\hat{\theta}_{n+1}}(dx_{n+1}|y_{0:n}) = \sum_{j=1}^{N} W_n^{(j)} p_{\hat{\theta}_{n+1}}(x_{n+1}|x_n^{(j)})$$

and assign particle weights $W_{n+1}^{(i)} \propto g_0^{(i)}(y_{n+1}|y_n, x_n^{(i)})$, $1 \leq i \leq N$.

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

$$\hat{T}_{\hat{\theta}_{n+1},n+1}^{(i)}(x_n^{(i)}) = \sum_{j=1}^{N} W_n^{(j)} p_{\hat{\theta}_{n+1}}(x_n^{(j)}|x_n^{(i)}) \left[ \hat{T}_{\hat{\theta}_{n+1}}(x_n^{(j)}) + s_{0,n}(x_n^{(j)}, x_{n+1}^{(i)}) \right],$$

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

$$\hat{\theta}_{n+1} = \hat{\theta}_n + y_{n+1} \{ \nabla \log p_{0,n}(y_{0:n}) - \nabla \log p_{0,n-1}(y_{0:n-1}) \}.$$
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 $\{g_n\}$ in (5.1), we use the standard adaptive method termed ADAM, by Kingma and Ba (2014). Assume that after $n$ steps, we have $c_n := -(\nabla \log p_{\hat{\theta}_n}(y_{0:n}) - \nabla \log p_{\theta_0}(y_{0:n-1}))$. ADAM applies the following iterative procedure,

$$m_n = m_{n-1} + (1 - \beta_1)c_n, \quad v_n = v_{n-1} + (1 - \beta_2)c_n^2,$$

$$\hat{m}_n = m_n/(1 - \beta_1^n), \quad \hat{v}_n = v_n/(1 - \beta_2^n),$$

$$\hat{\theta}_{n+1} = \hat{\theta}_n - \alpha \hat{m}_n/(\sqrt{\hat{v}_n} + \epsilon),$$

where $(\beta_1, \beta_2, \alpha, \epsilon)$ are tuning parameters. Convergence properties of ADAM have been widely studied (Kingma and...
Ba 2014; Goodfellow, Bengio, and Courville 2016; Reddi, Kale, and Kumar 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,

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

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

**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 $n = 2500, 5000, 7500, 10,000$ under true remaining parameter values $(\theta_1^*, \theta_3^*) = (0.4, 0.5)$. We executed Algorithm 2 to approximate the score function with $N = 50, 100, 150$ particles, for all above datasets. Figure 2 summarizes results of $R = 50$ replicates of estimates of the unknown parameters.

**Figure 4.** O–U: Experiment 3: boxplots of estimated score function of $\theta_1$ 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 $(\theta_1^*, \theta_2^*, \theta_3^*) = (0.3, 0.0, 0.2)$, used to generate $n = 10$ observations. Boxplots on the right (left) of each panel correspond to Construct 1 (Construct 2).

**Figure 5.** Left panel: data generated from model (6.2) for true values $\theta_1^* = (\pi/4, 0.9)$. Right Panel: Trajectories obtained from Algorithm 3, with $N = 100$ and initial values $(\hat{\theta}_1, \hat{\theta}_2, 0) = (0.1, 2)$. Solid lines show the results for $M = 10$ and dotted lines for $M = 100$. 

$$
    y_i = x_i + \epsilon_i, \quad i \geq 1, \quad \epsilon_i \overset{\text{iid}}{\sim} \mathcal{N}(0, 0.1^2), \quad \Delta = 1, \quad (6.1)
$$
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 \( \lambda, \zeta \), in both scenarios we generated \( n = 20,000 \) data points using the true parameters values \( \theta^* = (\theta_1^*, \theta_2^*, \theta_3^*) = (0.2, 0.0, 0.2) \). We applied Algorithm 3 with \( N = 100 \) particles, and initial parameter values \( \theta_0 = (1.0, 1.0, 1.0) \) for both scenarios. We used Construct 1 for the scenario with jumps. Figure 3 summarizes 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 \( (\theta_1^*, \theta_2^*, \theta_3^*) = (0.3, 0.0, 0.2) \). Figure 4 shows boxplots summarizing 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.

### 6.2. Periodic Drift SDE

We consider the (highly) nonlinear model,

\[
\begin{align*}
dX_t &= \sin(X_t - \theta_1) dt + \theta_2 dW_t, \quad X_0 = 0, \\
y_i &= x_i + \epsilon_i, \quad i \geq 1, \quad \epsilon_i \overset{\text{iid}}{\sim} N(0,0.1^2), \quad \Delta = 1, \quad (6.2)
\end{align*}
\]

for \( \theta_1 \in [0,2\pi), \theta_2 > 0 \). We used true values \( (\theta_1^*, \theta_2^*) = (\pi/4, 0.9) \), generated \( n = 10^4 \) data, and applied Algorithm 3 with \( N = 100 \) particles and initial values \( (\theta_{1,0}, \theta_{2,0}) = (0.1, 2) \). 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 5 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.

### 6.3. Heston Model

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

\[
\begin{align*}
dS_t &= \theta_1 S_t dt + \sqrt{\theta_2} S_t^{1/2} dW_t, \\
dX_t &= \theta_1 (X_t - \theta_2) dt + \theta_3 X_t^{1/2} dW_t, \quad X_0 = \theta_2,
\end{align*}
\]

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,

\[
\begin{align*}
dU_t &= (\theta_1 - \frac{\theta_2}{2}) dt + \frac{1}{2} \theta_3 S_t^{1/2} dW_t, \\
dX_t &= \theta_1 (X_t - \theta_2) dt + \theta_3 X_t^{1/2} dW_t, \quad X_0 = \theta_2. \quad (6.3)
\end{align*}
\]

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,

\[
\begin{align*}
\left[ y_i \mid y_{i-1}, \{X_s, s \in [t_{i-1}, t_i]\} \right] &\sim N(\mu_i, \Sigma_i),
\end{align*}
\]

where we have set,

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

We chose true parameter value \( \theta^* = (0.1, 0.1, 0.2, 0.45) \), and generated \( n = 10^4 \) observations, with \( \Delta = 1 \). We applied Algorithm 3 with \( \theta_0 = (0.005, 0.1, 0.4, 0.3) \) and \( N = 100 \) particles. The trajectories of the estimated parameters are given in Figure 6.

### 6.4. Sequential Model Selection—Real Data

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

\[
\text{BIC}(M) := -2\ell_{\text{MLE}}(y_{1:n}) + \operatorname{dim}(\theta) \log n,
\]

where \( \ell_{\text{MLE}} \) denotes the MLE, and \( \ell(\cdot) \) the log-likelihood. BIC and the (closely related) Bayes Factor are known to have good asymptotic properties, for example, 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., Nishii (1988); Sin and White (1996); Chib and Kuffner (2016); Yonekura, Beskos, and Singh (2018), for the case of discrete-time nested HMMs). One can plausibly conjecture such criteria also will perform well for the type of continuous-time HMMs we consider here. See also Eguchi and Maeda (2018) for a rigorous analysis of BIC for diffusion-type models. Given models \( M_a, M_b, \) with \( M_a \subset M_b \), we use the difference,

\[
\text{BIC}(M_{ab}) := -2(\ell_{\text{MLE}_{ab}}(y_{1:n}) + \ell_{\text{MLE}_{ab}}(y_{1:n})) + (\operatorname{dim}(\theta_a) - \operatorname{dim}(\theta_b)) \log n,
\]

with involved terms defined as obviously, to choose between \( M_a \) and \( M_b \).

We note that Eraker, Johannes, and Polson (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, Polson, and Stroud (2009) use the sequential likelihood ratio for model comparison, but such quantity can overshoot, that is, 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,

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

where

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

$$\mathcal{M}_2 : b^{(2)}_{\theta}(X_t) = \theta_0 + \theta_1 X_t + \theta_2 X^2_t,$$

$$\mathcal{M}_3 : b^{(3)}_{\theta}(X_t) = \theta_0 + \theta_1 X_t + \theta_2 X^2_t + \frac{\theta_3}{X_t}.$$

Such models have been used for short-term interest rates; see Jones (2003), Durham (2003), Aït-Sahalia (1996) and Bali and Wu (2006) and references therein for more details. Motivated by Dellaportas, Friel, and Roberts (2006) and Stanton (1997), we applied our methodology to daily 3-month treasury bill rates, from January 2, 1970, to December 29, 2000, providing $n = 7739$ observations. The data can be obtained from the Federal...
Reserve Bank of St. Louis, at webpage [https://fred.stlouisfed.org/series/TB3MS](https://fred.stlouisfed.org/series/TB3MS). The dataset is shown at Figure 7.

Figure 8 shows the BIC differences, obtained online, for each of the three pairs of models. The results suggest that $M_2$ does not provide a good fit to the data—relatively to $M_1$, $M_3$—for almost all period under consideration. In terms of $M_1$ against $M_3$, there is evidence that once the data from around 1993 onwards are taken under consideration, $M_3$ is preferred to $M_1$. In general, one can claim that models with nonlinear drift should be taken under consideration for fitting daily 3-month treasury bill rates, without strong evidence in favor or against linearity. This nondefinite conclusion is in some agreement with the empirical studies in Chapman and Pearson (2000), Durham (2003), and Dellaportas, Friel, and Roberts (2006).

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 generalization here, and one can extend the algorithm to different cases of jump processes, also characterized 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 $b(\nu) = b(t, \nu)$, $\sigma(\nu) = \sigma(t, \nu)$, 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, for example, the works of Schauer, Van Der Meulen, and Van Zanten (2017) and van der Meulen and Schauer (2017). Indicatively, the work in Schauer, Van Der Meulen, and Van Zanten (2017) considers bridges of the form (in one of the many options they consider),

$$d\tilde{X}_t = \left[ b(\tilde{X}_t) + \Sigma(\tilde{X}_t) \Sigma^{-1}(\tilde{X}_t) \left( X_t - \tilde{X}_t \right) T - t \right] dt + \sigma(\tilde{X}_t) dW_t. \tag{7.1}$$

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, Van Der Meulen, and Van Zanten (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, for example, the numerical examples in Durham and Gallant (2002) and the discussion in (Papaspiliopoulos and Roberts 2012, sec. 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 optimization. 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. For example, recent works in the Machine Learning community have pointed at the use of “velocity” components in the recursion to speed up convergence, see, for example, Sutskever et al. (2013) and Yuan, Ying, and Sayed (2016).

(iv) We mentioned in the main text several modifications that can improve algorithmic performance: dynamic resampling, stratified resampling, nonblind proposals in the filtering steps, choice of auxiliary processes. Parallelization 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 x′ the location of the path at all times of jumps; thus, when the algorithm “mixes” all pairs of \{x_k^{(i−1)}\}, \{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 \(x_k^{(i)}\) has in fact been generated conditionally on \(x_{k−1}^{(i−1)}\)). 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, x′ contains less information about the underlying paths, thereby improving the compatibility of pairs selected from particle populations \(\{x_k^{(i−1)}\}, \{x_k^{(i)}\}\), 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 5. In all cases, the code was written in Julia, was not parallelized or optimized 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 min). 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.

**Appendix A. Proof of Proposition 1**

**Proof.** We have the integral,

\[
\int T_{\theta,N}(x_1) p_0(dx_1|y_0:n) = \int S_{\theta}(x_0) p_0(dx_{0:n−1}|y_0:n−1, x_n) \otimes p_0(dx_n|y_0:n).
\]

(A.1)

Also, simple calculations give,

\[
p_0(dx_{0:n−1}|y_0:n−1, x_n) \otimes p_0(dx_n|y_0:n) = p_0(dx_{0:n}|y_0:n) = p_0(dx_{0:n}|y_0:n).
\]

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

**Appendix B. Proof of Proposition 2**

**Proof.** Simply note that,

\[
\int T_{\theta,N}(x_1) p_0(dx_1|y_0:n) = \int S_{\theta}(x_0) p_0(dx_{0:n−1}|y_0:n, x_n) \otimes p_0(dx_n|y_0:n).
\]

Then, we have that,

\[
p_0(dx_{0:n−1}|y_0:n−1, x_n) \otimes p_0(dx_n|y_0:n−1, x_n) = p_0(dx_{0:n−1}|y_0:n−1, x_n).
\]

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.

**Appendix 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 \(A^n_N(1/j) := \{|S_{\theta,N} − S_N^{\theta} > 1/2\}\), where we added superscript \(N\) to stress the dependency of the estimate on the number of particles. Then,

\[
\text{Prob}\left[\lim_{N \to \infty} S_{\theta,N} = S_{\theta,N}\right] = 1 − \text{Prob}\left[\bigcup_{j=1}^{\infty} \limsup_{N \to \infty} A_N^N(1/j)\right] \geq 1 − \sum_{j=1}^{\infty} \text{Prob}\left[\limsup_{N \to \infty} A_N^N(1/j)\right].
\]

(C.1)

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

**Acknowledgments**

We are grateful to the AE and the referees for several useful comments that helped us to improve the content of the paper.

**Funding**

Shouto Yonekura acknowledges funding from the Alan Turing Institute under grant number TU/C/000013. Alexandros Beskos acknowledges funding via a Leverhulme Trust Prize.

**References**

Aït-Sahalia, Y. (1996), “Testing Continuous-Time Models of the Spot Interest Rate,” The Review of Financial Studies, 9, 385–426. [1356]

——— (2002), “Maximum Likelihood Estimation of Discretely Sampled Diffusions: A Closed-Form Approximation Approach,” Econometrica, 70, 223–262. [1345]
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, 65–80. [1346,1352]

Reddi, S. J., Kale, S., and Kumar, S. (2019), “On the Convergence of Adam and Beyond,” arXiv preprint arXiv:1904.09237. [1354]

Roberts, G. O., and Stramer, O. (2001), “On Inference for Partially Observed Nonlinear Diffusion Models Using the Metropolis–Hastings Algorithm,” *Biometrika*, 88, 603–621. [1348]

Särkkä, S., and Sottinen, T. (2008), “Application of Girsanov Theorem to Particle Filtering of Discretely Observed Continuous-Time Non-linear Systems,” *Bayesian Analysis*, 3, 555–584. [1346,1348]

Schauer, M., Van Der Meulen, F., and van Zanten, H. (2017), “Guided Proposals for Simulating Multi-Dimensional Diffusion Bridges,” *Bernoulli*, 23, 2917–2950. [1357]

Schwarz, G. (1978), “Estimating the Dimension of a Model,” *Annals of Statistics*, 6, 461–464. [1355]

Sin, C.-Y., and White, H. (1996), “Information Criteria for Selecting Possibly Misspecified Parametric Models,” *Journal of Econometrics*, 71, 207–225. [1355]

Stanton, R. (1997), “A Nonparametric Model of Term Structure Dynamics and the Market Price of Interest Rate Risk,” *The Journal of Finance*, 52, 1973–2002. [1356]

Ströjby, J., and Olsson, J. (2009), “Efficient Particle-Based Likelihood Estimation in Partially Observed Diffusion Processes,” *IFAC Proceedings Volumes*, 42, 1364–1369. [1346]

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*, pp. 1139–1147. [1358]

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. [1352]

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, 2358–2396. [1357]

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, 1–20. [1345]

Yonekura, S., Beskos, A., and Singh, S. (2018), “Asymptotic Analysis of Model Selection Criteria for General Hidden Markov Models,” arXiv preprint arXiv:1811.11834. [1355]

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, 6602–6667. [1358]