Electronic Journal of Statistics
Vol. 15 (2021) 4295–4342
ISSN: 1935-7524
https://doi.org/10.1214/21-EJS1894

Continuous-discrete smoothing of diffusions*

Marcin Mider[1], **Moritz Schauer**[2] and **Frank van der Meulen**[3]

[1] *Trium Analysis Online GmbH, Hohenlindener Str. 1, 81677 München, Germany*
e-mail: marcin.mider@gmail.com

[2] *Department of Mathematical Sciences, Chalmers University of Technology and University of Gothenburg, Chalmers Tvärgata 3, 41296 Göteborg, Sweden*
e-mail: smoritz@chalmers.se

[3] *Delft Institute of Applied Mathematics (DIAM), Delft University of Technology, Mekelweg 4, 2628CD Delft, The Netherlands*
e-mail: f.h.vandermeulen@tudelft.nl

Abstract: Suppose X is a multivariate diffusion process that is observed discretely in time. At each observation time, a transformation of the state of the process is observed with noise. The smoothing problem consists of recovering the path of the process, consistent with the observations. We derive a novel Markov Chain Monte Carlo algorithm to sample from the exact smoothing distribution. The resulting algorithm is called the *Backward Filtering Forward Guiding (BFFG) algorithm*. We extend the algorithm to include parameter estimation. The proposed method relies on guided proposals introduced in [53]. We illustrate its efficiency in a number of challenging problems.

MSC2020 subject classifications: Primary 60J60, 65C05; secondary 62F15.
Keywords and phrases: Chemical reaction network, data assimilation, diffusion bridge, filtering, guided proposal, Lorenz system, Markov Chain Monte Carlo, partial observations, stochastic heat equation on a graph.

Received September 2020.

1. Introduction

Suppose X is a diffusion process with dynamics governed by the stochastic differential equation (SDE)

$$\mathrm{d}X_t = b(t, X_t)\,\mathrm{d}t + \sigma(t, X_t)\,\mathrm{d}W_t \tag{1.1}$$

with prescribed initial density $X_{t_0} \sim \pi$. Here $b\colon \mathbb{R}_+ \times \mathbb{R}^d \to \mathbb{R}^d$ and $\sigma\colon \mathbb{R}_+ \times \mathbb{R}^d \to \mathbb{R}^{d\times d'}$ are the drift and dispersion coefficient respectively. The process W is a vector valued process in $\mathbb{R}^{d'}$ consisting of independent Brownian motions.

arXiv: 1712.03807

*The research leading to the results in this paper has received funding from the European Research Council under ERC Grant Agreement 320637. M.M. was sponsored by the Max Planck Institute for Mathematics in the Sciences, Leipzig and the EPSRC [grant number EP/L016710/1].

It is assumed that the required conditions for existence of a strong solution are satisfied (cf. [36]). We assume observation times $0 = t_0 < t_1 < \cdots < t_n$ and observations

$$V_i \mid X_{t_i} \sim k_i(X_{t_i}; \cdot), \qquad i = 0, \dots, n, \tag{1.2}$$

with conditional density k_i. This includes as special case

$$V_i \mid X_{t_i} \sim \varphi(\cdot; L_i X_{t_i}, \Sigma_i), \tag{1.3}$$

where L_i is a $m_i \times d$-matrix, Σ_i an $m_i \times m_i$ positive definite matrix and $\varphi(x; \mu, \Sigma)$ denotes the density of the $\mathrm{N}(\mu, \Sigma)$-distribution, evaluated at x. For $t \geq 0$, denote the set of non-past observations by $\mathcal{V}_t$, i.e. $\mathcal{V}_t = \{V_i \colon t_i \geq t\}$. With slight abuse of notation, set $\mathcal{V}_i = \mathcal{V}_{t_i}$.

In this article we address two related problems: *smoothing* and *inference*. For the first one we assume that b, σ and $\{k_i,\, i = 0, \dots, n\}$ are known and aim to reconstruct the path $\{X_t,\, t \in [0, t_n]\}$ based on $\mathcal{V}_0$, which here is given precise meaning as sampling from the conditional distribution of X on $[0, t_n]$ conditional on $\mathcal{V}_0$. Because we are dealing with "continuous dynamics-discrete observations", this setup is also referred to as continuous-discrete smoothing or hybrid smoothing. This is an interesting problem in its own right that arises naturally for instance in statistical analysis of tracking systems (cf. [49]).

Additionally, smoothing constitutes a central component of Bayesian inference algorithms for discretely observed diffusions. If in (1.1) the coefficients of the diffusion X or the observation scheme $\{k_i,\, i = 0, \dots, n\}$ depend on some unknown parameters, inference can be performed by employing a data augmentation algorithm, where one iterates between *(i)* updating the parameter conditional on the smoothed path and *(ii)* continuous-discrete smoothing conditional on the value of the parameter (cf. [48]). The relevance of smoothing and parameter estimation for diffusions is illustrated from the problem reappearing in a wide variety of application fields, see for instance [60], [3], [1], [17] and [63]. There is extensive research about this topic, in Section A we put this work further into context of related research. The setup (1.2) includes the practically relevant case that a highly non-linear process X is only observed at start time 0 and at a final time, say 1, see for instance [4] for an application in shape analysis.

1.1. Data augmentation for continuous-discrete smoothing

In this section we give intuition on the method that we will give for sampling from the smoothing distribution. Assume for the moment that x_0 is known and denote x_{t_i} by x_i. Using Bayesian notation (writing π for a density), we have the following recursive relation

$$\pi(x_1, \dots, x_i \mid \mathcal{V}_1) = \pi(x_i \mid x_{i-1}, \mathcal{V}_1) \pi(x_1, \dots, x_{i-1} \mid \mathcal{V}_1).$$

Here the first term on the right hand side captures the dynamics of the conditional process, which satisfies

$$\pi(x_i \mid x_{i-1}, \mathcal{V}_1) = \pi(x_i \mid x_{i-1}, \mathcal{V}_i) = \pi(x_i \mid x_{i-1}) w_i,$$

where we write suggestively

$$w_i = \frac{\pi(\mathcal{V}_i \mid x_i)}{\pi(\mathcal{V}_i \mid x_{i-1})} = \exp\left(\log \pi(\mathcal{V}_i \mid x_i) - \log \pi(\mathcal{V}_i \mid x_{i-1})\right).$$

In the (non-Bayesian) notation used in the next subsection, $\pi(\mathcal{V}_i \mid x_i)$, the likelihood of x_i given the present and future observations $\mathcal{V}_i$, is denoted as $\rho(t_i, x_i)$. Up to a scaling factor, ρ can be related to a backward filtered marginal density of X. In a simplified view of our approach, firstly we compute the scaled filtering density backward in time.

One can now show using the arguments laid out in Appendix C that the weight equals the exponential process of a change of measure:

$$w_i = \exp\left(\int_{t_{i-1}}^{t_i} \sigma(s, X_s)'\nabla \log \rho(s, X_s)\, \mathrm{d}W_s \right.$$
$$\left. -\frac{1}{2}\int_{t_{i-1}}^{t_i} \|\sigma(s, X_s)'\nabla \log \rho(s, X_s)\|^2\, \mathrm{d}s\right),$$

which reveals the changed drift $b + \sigma\sigma'\nabla \log \rho$ of the conditional (smoothed) process by Girsanov's theorem. Thus we may substitute sampling $X_i \mid X_{i-1}, \mathcal{V}_i$ by simulating a process X with drift $b+\sigma\sigma'\nabla \log \rho$, understood as *data augmentation* in the sense of [54], and obtain a sample of the smoothing distribution of X_i (and in fact the entire smoothed segment $\{X_t, t_{i-1} \le t \le t_i\}$) in this way.

Finally, if ρ cannot be computed efficiently, we find a substitute $\tilde{\rho}$ and correct for the difference through Monte Carlo methods: this is the technique of guided proposals as introduced in the following section.

1.2. Approach

Conceptually, we follow closely the literature on *guided proposals*. We explain the main idea here, more technical details can be found in Appendix B.

Assume X admits smooth transition densities p, i.e. for $s < \tau$, $\mathrm{P}^{(s,x)}(X_\tau \in \mathrm{d}y) = p(s, x; \tau, y)\,\mathrm{d}y$. For $i \in \{1, \dots n\}$, $t \in (t_{i-1}, t_i]$, the process X, conditioned on $V_i, \dots, V_n$ and $X_{t_{i-1}} = x_{t_{i-1}}$ satisfies the SDE

$$\mathrm{d}X^\star_t = b(t, X^\star_t)\,\mathrm{d}t + a(t, X^\star_t)r(t, X^\star_t)\,\mathrm{d}t + \sigma(t, X^\star_t)\,\mathrm{d}W_t, \qquad X^\star_{t_i} = x_{t_i}, \quad (1.4)$$

with $a(t, x) = \sigma(t, x)\sigma(t, x)'$, $r(t, x) = \nabla_x \log \rho(t, x)$ (considered as a column vector) and the likelihood term ρ defined by integrating out the latent future states,

$$\rho(t, x) = \int p(t, x; t_i, \xi_i)k_n(\xi_n; v_n) \prod_{j=i}^{n-1} p(t_j, \xi_j; t_{j+1}, \xi_{j+1})k_j(\xi_j; v_j)\,\mathrm{d}\xi_i \cdots \mathrm{d}\xi_n. \quad (1.5)$$

Define $\rho(0,x) = k_0(x;v_0)\rho(0+,x)$. If $X_{t_0} \sim \pi^\star$, with

$$\pi^\star(x_0) = \frac{\pi(x_0)\rho(0,x_0)}{\int \pi(x_0)\rho(0,x_0)\,\mathrm{d}x_0} \tag{1.6}$$

being the smoothed (conditional) density of x_0, then a trajectory $X^\star$ constitutes a sample from the smoothing distribution. This expression was derived in [41] (Theorem 2.3.4) in a special case of (1.3) with $\Sigma_j \equiv 0$ for all j). In Appendix C we derive it using Doob's h-transform for the more general setting considered in this paper. As p is tractable only in very specific instances, so are ρ and $\pi^\star$. Hence, sampling $X^\star$ directly from (1.4) is generally not possible. The key idea in [53] (which addressed a simpler problem of diffusion bridge simulation) is to introduce the so-called *guided proposal process*, defined as the (strong) solution to the SDE

$$\mathrm{d}X^\circ_t = b(t,X^\circ_t)\,\mathrm{d}t + a(t,X^\circ_t)\tilde{r}(t,X^\circ_t)\,\mathrm{d}t + \sigma(t,X^\circ_t)\,\mathrm{d}W_t, \qquad X^\circ_{t_i} = x_{t_i}, \tag{1.7}$$

where $\tilde{r}(t,x) = \nabla_x \log\tilde{\rho}(t,x)$ and $\tilde{\rho}(t,x)$ is defined in a similar manner as $\rho(t,x)$, but with p and $\{k_i\}$ replaced by tractable approximations $\tilde{p}$ and $\{\tilde{k}_i\}$ respectively. Here $\tilde{p}$ can be taken to be the transition density of some auxiliary diffusion $\tilde{X}$ and $\{\tilde{k}_i\}$ can be based on (1.3). The choice of $\tilde{X}$ and $\{\tilde{k}_i\}$ impacts the quality with which the law of X° approximates the law of the target process $X^\star$. Intuitively, the drift and dispersion of $\tilde{X}$ should be chosen such that $\tilde{X}$ is similar to X in areas visited by the true conditional process. At the same time, the two functions need to be simple enough, so as to make it possible to compute $\tilde{\rho}(t,x)$ and $\tilde{r}(t,x)$. [53] showed that the linear processes, defined by the SDE

$$\mathrm{d}\tilde{X}_t = \left(\beta(t) + B(t)\tilde{X}_t\right)\,\mathrm{d}t + \tilde{\sigma}(t)\,\mathrm{d}W_t, \tag{1.8}$$

give rise to a family of auxiliary diffusions that grant a generous degree of flexibility (as detailed in Section 3.2) for crafting suitable proposals X°. For instance, in [53, Section 1.3] the authors showed how in a setting of a highly non-linear target diffusion $X^\star$, a simple shift by a constant in $\beta(t)$ may lead to a dramatic improvement of simulated proposal bridges X°, leading e.g. to lowering of rejection rates in Metropolis–Hastings algorithms.

Assume for now that x_0 is known and $k_i = \tilde{k}_i$ with $\tilde{k}_i$ derived from observation scheme (1.3) and transition density $\tilde{p}$ derived from (1.8). We defer the general case to Section 4. Let $\mathbb{P}^\star$ and $\mathbb{P}^\circ$ denote the laws of $X^\star$ and X° on $[0,t_n]$ (started at x_0) respectively. Under regularity conditions on the auxiliary process $\tilde{X}$ listed in Appendix B, it follows from Theorem 1 in [53] and Theorem 2.14 in [12] that

$$\frac{\mathrm{d}\mathbb{P}^\star}{\mathrm{d}\mathbb{P}^\circ}(X^\circ) = \frac{\tilde{\rho}(0+,x_0)}{\rho(0+,x_0)}\Psi(X^\circ), \tag{1.9}$$

where

$$\Psi(X^\circ) = \exp\left(\int_0^{t_n} G(s,X^\circ_s)\,\mathrm{d}s\right), \tag{1.10}$$

and

$$\begin{aligned} G(s,x) =& (b(s,x) - \tilde{b}(s,x))'\tilde{r}(s,x) \\ & - \frac{1}{2}\operatorname{tr}\left([a(s,x) - \tilde{a}(s)]\,[H(s) - \tilde{r}(s,x)\tilde{r}(s,x)']\right). \end{aligned} \tag{1.11}$$

The factor in front of Ψ is written to indicate that in this expression ρ and $\tilde{\rho}$ are to be evaluated at time $0+$, which is of importance when considering the case where x_0 is not known. Here, $\tilde{b}$ and $\tilde{\sigma}$ are the coefficients of the SDE defining $\tilde{X}$, $\tilde{a}(s) = \tilde{\sigma}(s)\tilde{\sigma}(s)'$ and $H(s)$ is the negative of the Hessian matrix of $x \mapsto \nabla_x \log \tilde{\rho}(s,x)$ (which turns out not to depend on x).

Since all terms in (1.7) and (1.9) are in principle tractable, one may define an importance sampler that targets the law $\mathbb{P}^\star$: proposals are drawn from X° (by forward simulating paths via some discretisation scheme for SDEs, say, the Euler–Maruyama scheme) and importance weights are computed based on (1.9). As the SDE for X° is obtained from the SDE for $X^\star$ by superimposing a guiding term, the paths distributed according to the former law are called *guided proposals*. The process X° is typically used as a proposal (the origin of the terminology "guided proposal") in a "latent" path-space Metropolis–Hastings step such as the non-centred path-space Crank-Nicolson scheme (cf. [16] and [9]).

However, our results can also be used in other Monte Carlo procedures, for example in an auxiliary particle filter. Essentially our backward filtering step is an approximation to the backwards information filter. Various related "optimal policy finding" strategies for finding a good approximation have been proposed in the particle filtering literature, such as the iterated auxiliary particle filter (cf. [30]) and controlled SMC (cf. [34]). Guided proposals can also be used to create flexible variational classes for latent diffusion paths (cf. recent work [7]). In fact guided proposals are agnostic to the choice of inferential algorithm, a point highlighted in the preprint [58].

Whereas the computations of $\tilde{r}$, H and $\tilde{\rho}(0,x_0)$ for (1.8) are in principle tractable, it is not trivial to derive an efficient numerical scheme. The numerical methods presented in [53] are restricted to simpler cases and to diffusion bridge simulation on a single time segment. *In this paper we derive an efficient scalable scheme (both in the number of observations as dimension of the state space of the diffusion) using guided proposals for sampling high-dimensional, non-linear latent diffusion processes under the general observation scheme* (1.2)*, and to estimate unknown parameters.* This is illustrated in a number of challenging numerical examples.

1.3. Innovation

We derive simple systems of ordinary differential equations that can be used to compute all terms needed for implementation of guided proposals, i.e. terms $\tilde{r}$, H and $\tilde{\rho}(0,x_0)$. These results imply that in order to sample paths of the proposal diffusion X° and embed them in the Metropolis–Hastings algorithm only the following steps need to be performed

- a system of ordinary differential equations, akin to the updating equations in hybrid Kalman filtering, has to be solved backwards in time (suitable systems are given in theorems 2.4, 2.5 and 2.6);
- paths of X° need to be simulated forward in time, using standard simulation techniques based on stochastic Taylor expansions (cf. [38]);
- the Radon–Nikodym derivative between the laws of $X^\star$ and X° has to be evaluated and then used in the acceptance probability of the Metropolis–Hastings algorithm.

In particular, in this scheme, forward simulating the guided proposal X° is comparable in computational effort to simulating X itself. This resolves objections about (perceived) computational effort required to simulate guided proposals using a closed form expression for ρ. The systems of ODEs make use only of the matrix addition, multiplication and inversion operations and scale much better to high dimensional problems compared to for example [55]. For the smoothing problem, the first step needs to be executed only once.

We call the resulting algorithm the *Backward Filtering Forward Guiding* (BFFG) algorithm, because (as we will show) it combines backward filtering under the auxiliary process $\tilde{X}$ and forward simulation of the guiding process X°. It may also be characterised as a variant of the forward filtering-backward sampling algorithm for linear state space models with (possibly) non-linear continuous time processes. Reversing the time-direction and sampling the process forward in time is not strictly necessary, but turns out to be more convenient.

We believe the method derived in this paper has a number of attractive properties:

1. It is a computationally simple algorithm that provides a unified approach to smoothing of both hypo-elliptic and uniformly elliptic diffusions.
2. It allows for taking into account nonlinearities in the drift efficiently (by choice of the auxiliary process $\tilde{X}$).
3. It can deal with a large class of diffusions with state-dependent diffusion coefficient.
4. The algorithm targets the exact smoothing distribution and does not rely on Gaussian approximations.
5. It can deal with nonlinear observation schemes or non-Gaussian error densities while still sampling from the exact smoothing distribution.

Regarding being exact, we derive our algorithm in continuous time, but ultimately, in any implementation the SDE for X° needs to be discretised. However, the mesh-width for the discretisation can be controlled by the user.

1.4. Outline

In Section 2 we derive ODE-systems for backward filtering which result in efficient ways for computing $\tilde{r}$, H and $\tilde{\rho}(0, x_0)$. In Section 3 we discuss strategies for choosing the auxiliary process $\tilde{X}$ and auxiliary observation scheme to be used in backward filtering. The backward filtering forward guiding algorithm for diffusions is subsequently given in Section 4. In Section 5 we address applications of the proposed algorithms to problems in which the dimension of the state-space of the diffusion is high. In such a setting, the backward filtering can for example be carried out using an ensemble Kalman filter. We illustrate our results on 3 challenging examples in Section 6. The appendix contains sections on related literature, a derivation of the guiding term for guided proposals together with deferred proofs and remarks.

2. Backward filtering

To sample the guided proposal, we need to define $\tilde{\rho}$ as a proxy to ρ defined in (1.5). For that, we choose $\tilde{X}$ as in (1.8). For $i = 0, \dots, n$, we assume $\tilde{k}_i$ to correspond to the observation scheme (1.3) which is parametrised by L_i and Σ_i.

Definition 2.1. For a specified (strictly positive definite) covariance matrix P_{n+}, define $\tilde{k}_{n+}(x) = \varphi(x; 0, P_{n+})$. We define $\tilde{\rho}$ by

$$\begin{aligned}\tilde{\rho}(t,x) = \int & \tilde{k}_{n+}(\xi_n)\tilde{p}(t,x;t_i,\xi_i)\tilde{k}_n(\xi_n;v_n) \\ & \times \prod_{j=i}^{n-1} \tilde{p}(t_j,\xi_j;t_{j+1},\xi_{j+1})\tilde{k}_j(\xi_j;v_j)\,\mathrm{d}\xi_i \cdots \mathrm{d}\xi_n.\end{aligned} \tag{2.1}$$

Note the inclusion of k_{n+} which does not appear in (1.5). It ensures regularisation, where we may intuitively think of $P_{n+} = \varepsilon^{-1} I$ with ϵ small, though any choice of P_{n+} will give rise to a valid algorithm targeting the exact smoothing distribution.

Naturally BFFG will work best if $\{\tilde{k}_i\}$ and $\tilde{X}$ approximate $\{k_i\}$ and X well, most importantly in those areas where the smoothing distribution puts its weight. We comment on good choices in Section 3. Theorems 2.4, 2.5 and 2.6 in this section give backward recursions for evaluating $\tilde{r}$, H and $\tilde{\rho}$ which are needed for application of guided proposals. All proofs are in Section D of the appendix.

We impose

Assumption 2.2. For each $i \in \{0, \dots, n\}$, the matrix Σ_i is strictly positive definite.

Assumption 2.3. The maps $t \mapsto \beta(t)$, $t \mapsto B(t)$ and $t \mapsto \tilde{\sigma}(t)$ are bounded on $[0, T]$.

By Theorem 1.184 in [14], Assumption 2.3 ensures existence and uniqueness of the ODEs for $t \mapsto L(t)$, $t \mapsto P(t)$ and $t \mapsto \nu(t)$ appearing in Theorem 2.4 and Theorem 2.6 below. Uniqueness and existence of ν and P then translates to the ODEs for $t \mapsto F(t)$ and $t \mapsto H(t)$ appearing in Theorem 2.5 below.

For $i \in \{1, \dots, n\}$, let $v(t)$ be defined by the concatenated vector of all non-past observations:

$$v(t) = [v_i', \dots, v_n']' \qquad t \in (t_{i-1}, t_i],$$

where $i \in \{1, \dots, n\}$. Define $v(0) = [v_0', \dots, v_n']'$, let $m(t) = \dim(v(t))$ and recall $m_i = \dim(v_i)$. In the following, we give three results characterising $\tilde{\rho}$, first directly as marginal likelihood, and then in a form more suitable for implementation: as (scaled) backward ODE for the filtering density in two parametrisations.

Theorem 2.4. Let the triplet $(L(t), M^\dagger(t), \mu(t))$ be defined on each interval $(t_{i-1}, t_i]$ $(i = 1, \dots, n)$ as solutions to the backward ODE systems

$$\mathrm{d}L(t) = -L(t)B(t)\,\mathrm{d}t \tag{2.2}$$

$$\mathrm{d}M^\dagger(t) = -L(t)\tilde{a}(t)L(t)'\,\mathrm{d}t \tag{2.3}$$

$$\mathrm{d}\mu(t) = -L(t)\beta(t)\,\mathrm{d}t. \tag{2.4}$$

Define

$$L(t_n+) = I \quad M^\dagger(t_n+) = P_{n+} \quad \mu(t_n+) = 0$$

and for $i = 0, \dots, n$,

$$L(t_i) = \begin{bmatrix} L_i \\ L(t_i+) \end{bmatrix}, \quad M^\dagger(t_i) = \begin{bmatrix} \Sigma_i & 0_{m_i \times m(t_i+)} \\ 0_{m(t_i+) \times m_i} & M^\dagger(t_i+) \end{bmatrix}, \quad \mu(t_i) = \begin{bmatrix} 0_{m_i \times 1} \\ \mu(t_i+) \end{bmatrix}. \tag{2.5}$$

If $M(t) = M^\dagger(t)^{-1}$, then for all $t \in [0, t_n]$, we have

$$\tilde{r}(t, x) = F(t) - H(t)x,$$
$$\tilde{\rho}(t, x) = \varphi(v; \mu(t) + L(t)x, M^\dagger(t))$$

where

$$H(t) = L(t)'M(t)L(t), \tag{2.6}$$

$$F(t) = L(t)'M(t)(v(t) - \mu(t)). \tag{2.7}$$

Note that the equations for $M^\dagger$ and μ directly give the time derivative and that the values of $M^\dagger(t)$ and $\mu(t)$ can be found by using a numerical quadrature rule such as for example the trapezoid rule. This theorem shows that H and $\tilde{r}$ can be computed by solving the systems (2.2), (2.3) and (2.4). Additionally, by computing the term $|M(t)|$, we may evaluate $\tilde{\rho}$ as well. In Theorem 2.5 below we will see that computation of this determinant can be avoided by solving another (scalar-valued) backward ODE.

The dimensions of the equations in Theorem 2.4 are larger on $(t_{i-1}, t_i]$ than on $(t_{j-1}, t_j]$ for all $j > i$. The dimension on the segment closest to zero (i.e. the leftmost segment) grows rapidly with a large number of observations. It turns out that other sets of backward differential equations can be derived, which are constant in dimensions determined by the the dimension of the state space of the diffusion, no matter the number of observations.

Theorem 2.5 (Information filter)**.** For $(t, x) \in [0, T] \times \mathbb{R}^d$, $\tilde{\rho}$ admits the following decomposition

$$\log \tilde{\rho}(t, x) = -c(t) - \frac{1}{2}x'H(t)x + F(t)'x,$$

where on each interval $(t_{i-1}, t_i]$ $(i = 1, \dots, n)$, H, F and c solve the backward ODEs

$$\begin{aligned} \mathrm{d}H(t) &= (-B(t)'H(t) - H(t)B(t) + H(t)\tilde{a}(t)H(t))\ \mathrm{d}t, \\ \mathrm{d}F(t) &= (-B(t)'F(t) + H(t)\tilde{a}(t)F(t) + H(t)\beta(t))\ \mathrm{d}t, \\ \mathrm{d}c(t) &= \left(\beta(t)'F(t) + \frac{1}{2}F(t)'\tilde{a}(t)F(t) - \frac{1}{2}\mathrm{tr}\,(H(t)\tilde{a}(t))\right)\ \mathrm{d}t. \end{aligned} \tag{2.8}$$

At observations times, we have for $i = 0, \dots, n$

$$\begin{aligned} H(t_i) &= H(t_i+) + L_i'\Sigma_i^{-1}L_i, \\ F(t_i) &= F(t_i+) + L_i'\Sigma_i^{-1}v_i, \qquad , \\ c(t_i) &= c(t_i+) - \log\varphi(v_i; 0, \Sigma_i). \end{aligned} \tag{2.9}$$

to be initialised from

$$H(t_n+) = P_{n+}^{-1}, \quad F(t_n+) = 0, \quad c(t_n+) = \varphi(0; 0, P_{n+}) \tag{2.10}$$

Note that $H(t) \in \mathbb{R}^{d\times d}$, $F(t) \in \mathbb{R}^d$ and that $c(t)$ is scalar-valued throughout $[0, t_n]$.

Theorem 2.6 (Covariance filter)**.** Let $\nu(t) = P(t)F(t)$ with $P(t) = H(t)^{-1}$. The mapping $\tilde{r}$ satisfies

$$P(t)\tilde{r}(t, x) = \nu(t) - x, \tag{2.11}$$

On each interval $(t_{i-1}, t_i]$ $(i = 1, \dots, n)$, P and ν satisfy the backward ODEs

$$\begin{aligned} \mathrm{d}P(t) &= (B(t)P(t) + P(t)B(t)' - \tilde{a}(t))\ \mathrm{d}t, \qquad (2.12) \\ \mathrm{d}\nu(t) &= (B(t)\nu(t) + \beta(t))\ \mathrm{d}t. \end{aligned}$$

Furthermore, if we define $P(t_n+) = P_{n+}$ and $\nu(t_n+) = 0$ then for $i = 0, \dots, n$

$$P(t_i) = P(t_i+) - P(t_i+)L_i'\left(\Sigma_i + L_iP(t_i+)L_i'\right)^{-1}L_iP(t_i+) \tag{2.13}$$

$$\nu(t_i) = P(t_i)\left(L_i'\Sigma_i^{-1}v_i + H(t_i+)\nu(t_i+)\right). \tag{2.14}$$

From (2.13) it is seen that Assumption 2.3 can be weakened to invertibility of $\Sigma_i + L_iP(t_i+)L_i'$.

In case $\tilde{\sigma}$ and B are constant, $P(t)$ can be computed directly.

Proposition 2.7. Assume B and $\tilde{a}$ are constant and Σ solves the continuous time Lyapunov equation

$$B\Sigma + \Sigma B' + \tilde{a} = 0.$$

Then the solution to

$$\mathrm{d}P(t) = (BP(t) + P(t)B' - \tilde{a})\ \mathrm{d}t, \qquad P(T) = P_T$$

is given by

$$P(t) = \Phi(t, T)(\Sigma + P_T)\Phi(t, T)' - \Sigma, \tag{2.15}$$

where $\Phi(t, T) = \exp(-(T - t)B)$.

Remark 2.8. It is natural to ask which of the 3 recursions to use. In case of only one future observation at time t_1, the ODEs in Theorem 2.4 are advantageous when $m_1 < d$. Moreover, these are also applicable in case of a noiseless observation at time t_1 by taking $M^{\dagger}(t_1)$ to be equal to a zero matrix. With multiple observations, usually, the recursions from Theorem 2.5 or Theorem 2.6 are preferable and can lead to large computational gains.

Remark 2.9. Let $\mathcal{V}_t = \{V_i, \dots, V_n\}$ if $t \in (t_{i-1}, t_i]$. The interpretation of ν and P (appearing in Theorem 2.6) follows from $\tilde{X}_t \mid \mathcal{V}_t \sim \mathrm{N}(\nu(t), P(t))$, i.e. these are the mean and covariance of the backward filtered process. The update formula at observations times can now be seen from the following argument. Denote $\tilde{X}_i \equiv \tilde{X}_{t_i}$. Suppose $(\nu(t), P(t))$ has been computed for $t \in (t_i, t_n]$ for $i \in \{0, \dots, n-1\}$. At time t_i the observation $\tilde{X}_i$ can be incorporated using the following relation:

$$p(\tilde{X}_i \mid \mathcal{V}_i) \propto p(V_i \mid \tilde{X}_i, \mathcal{V}_{i+1}) p(\tilde{X}_i \mid \mathcal{V}_{i+1}) = p(V_i \mid \tilde{X}_i) p(\tilde{X}_i \mid \mathcal{V}_{i+1}). \tag{2.16}$$

Since $\tilde{X}_i \mid \mathcal{V}_{i+1} \sim \mathrm{N}(\nu(t_i+), P(t_i+))$ and $V_i \mid \tilde{X}_i \sim \mathrm{N}(L_i \tilde{X}_i, \Sigma_i)$, the joint distribution of $(\tilde{X}_i, V_i) \mid \mathcal{V}_{i+1}$ is Normal. Hence there is a closed form expression for the distribution of $\tilde{X}_i \mid (V_i, \mathcal{V}_{i+1})$ which gives $\nu(t_i)$ and $P(t_i)$.

3. Choice in auxiliary process and auxiliary observation scheme

Application of the tractable backward filtering updates requires choosing $(\{L_i\}, \{\Sigma_i\})$ for the observation scheme and $t \mapsto (\beta(t), B(t), \tilde{\sigma}(t))$ for the auxiliary process $\tilde{X}$.

3.1. Choice of the auxiliary observation scheme

The backward filtering formulas in Section 2 are used with the auxiliary observation scheme of the form implied by (1.3), leaving open the choice of (L_i, Σ_i) in case the actual observation scheme is of the more general form (1.2).

The simplest thing to do is choose L_i and Σ_i such that $\varphi(v_i; L_i X_{t_i}, \Sigma_i) \approx k_i(X_{t_i}; v_i)$. Ultimately, the best choice of (L_i, Σ_i) is that the discrepancy between $\mathbb{P}^{\circ}$ and $\mathbb{P}^{\star}$ is minimal (for example measured by (reverse) Kullback-Leibler divergence).

Example 3.1. Assume that instead of $V_i \sim \mathrm{N}(L_i X_{t_i}, \Sigma_i)$ one observes $V_i \sim \mathrm{N}(g_i(X_{t_i}), \Sigma_i)$ with g_i a nonlinear map. Consider the linearisation (with ν as in the covariance filter)

$$g_i(x) \approx g_i(\nu(t_i+)) + G_i(x - \nu(t_i+)) = \zeta_i + G_i x.$$

where $G_i = (Dg_i)(\nu(t_i+))$ and $\zeta_i = g_i(\nu(t_i+)) - G_i \nu(t_i+)$. Under this linearisation, backward filtering can be carried out exactly as in our original setting by assuming to observe $V_i - \zeta_i$ instead of V_i and setting L_i to be equal to G_i. This choice of auxiliary scheme is alike the extended Kalman filter (Algorithm 5.4 in [49]).

We refer to Chapters 5 and 6 in [49] for other approximations proposed in the literature that can be used as auxiliary observation scheme.

3.2. Choice of the auxiliary process $\tilde{X}$

The auxiliary process $\tilde{X}$ needs to be chosen by a user, i.e. the parameters $\beta(t)$, $B(t)$ and $\tilde{\sigma}(t)$ need to satisfy matching and regularity conditions, but are otherwise free. A number of practical ways for choosing them are discussed in [56] (Section 4.4); below, we list a number of reasonable options. Determining an optimal auxiliary law in an automated fashion deserves its own study and is the subject of an ongoing research (but cf. also considerations in [53]).

A Waive the freedom and simply take $\tilde{\sigma}$ constant, $B(t) \equiv 0$ and $\beta = 0$. If X is either elliptic or hypo-elliptic with nonzero noise on each coordinate, this yields a valid algorithm. It still takes local nonlinearity into account through the presence of b in (1.7) and, in case of partial observations, is informed by multiple future observations.

B On each $t \in (t_{i-1}, t_i]$ compute a first order Taylor expansion of the drift evaluated at a specified $\tilde{x}(t_i)$. That is, compute $\tilde{b}(t,x) = b(t, \tilde{x}(t_i)) + J_b(t, \tilde{x}(t_i))(x - \tilde{x}(t_i))$, where J_b denotes the Jacobian matrix of b. In case the diffusion is fully observed, $\tilde{x}(t_i)$ is taken equal to X_{t_i}, else it is specified by a user-chosen guess for it. The guess can be adaptively refined by using empirical averages of the imputed path. Otherwise put, the auxiliary process comes from linearisations of the target diffusion at the times of subsequent observations.

C The aim is to employ a first order Taylor expansion $\tilde{b}(t,x) = b(t, \bar{x}(t)) + J_b(t, \bar{x}(t))(x - \bar{x}(t))$ evaluated at $\bar{x}(t) = \mathbb{E}[X_t \mid D]$. Of course, $\bar{x}(t)$ is unknown, but information becomes available during MCMC-iterations, and thus, just as in **B**, empirical averages of the imputed path can be used as its proxy. More specifically, we propose to first use the auxiliary process constructed by one of the preceding methods to get $B^{(0)}(t,x) = \beta(t) + B(t)x$, and then, run the algorithm for k iterations and compute the average of these k paths. Denote this average by $\{\bar{X}^{(0)}(t),\, t \in [0, t_n]\}$. Next, starting at $i = 1$, repeat the following steps until the prescribed total number of iterations for adaptation has been reached.

(a) Set

$$B^{(i)}(t,x) = b(t, \bar{X}^{(i-1)}(t)) + J_b(t, \bar{X}^{(i-1)}(t)) \left(x - \bar{X}^{(i-1)}(t) \right).$$

(b) Recompute $H(t)$, $F(t)$ and $c(t)$ on $[0, t_n]$ based on $B^{(i)}(t,x)$.

(c) Perform k MCMC-iterations based on $H(t)$, $F(t)$ and $c(t)$. Compute the average of all simulated paths to obtain $\{\bar{X}^{(i)}(t),\, t \in [0, t_n]\}$

To avoid common problems with adaptive schemes we stop the adaptation after a fixed number of steps.

D Choose $\tilde{\sigma} = \sigma$, $B(t) \equiv 0$ and take β nonzero to make the pulling term itself take into account the nonlinearity of the system. To determine β, first obtain $F(t_n+)$ and $H(t_n+)$ as in (2.10). Next, for $i = n$ to 1:

(a) For $t \in (t_{i-1}, t_i]$, backwards solve

$$\mathrm{d}x(t) = b(t, x(t))\,\mathrm{d}t, \qquad x(t_i) = H(t_i)F(t_i).$$

Set $\beta(t) = b(t, x(t))$.

(b) Using β, compute $(F(t), H(t))$ for all $t \in (t_{i-1}, t_i]$. Compute $(F(t_{i-1}), H(t_{i-1}))$ using $(F(t_{i-1}+), H(t_{i-1}+))$ using the formulas of the information filter (Theorem 2.5).

E Use a linear combination of the schemes above. This is a risk-averse, robust strategy, which aims to guard against unexpected artifacts that some of the adaptive strategies above might exhibit (say, for strategy **C** such behaviour could appear as a result of multimodality on a path space). In Section 6 we use: $B = wb^{(B)} + (1-w)b^{(C)}$, where $b^{(B)}$ and $b^{(C)}$ are defined by strategies **B** and **C** respectively and $w \in [0, 1]$ are some user-chosen weights. Another option (not considered here) consists of sampling w in each iteration from the Bernoulli distribution with fixed success probability.

F Finally, the auxiliary process may take values in a higher dimensional space, with only its first d coordinates used in defining the proposal diffusion. Say, we want to target a process X conditioned on the observations $V_i = LX_{t_i} + \eta_i$. Consider an auxiliary process $[\tilde{X}, Y]$ with $\tilde{X}$ solving $\mathrm{d}\tilde{X}_t = (B\tilde{X}_t + \beta(t) + CY_t)\,\mathrm{d}t + \tilde{\sigma}\,\mathrm{d}W_t$ and Y a second, independent linear process $\mathrm{d}Y_t = DY_t\,\mathrm{d}t + \mathrm{d}W_t^Y$ driven by an independent Brownian motion W^Y. Here B, C and D are matrices of suitable dimension. Then, we may use an augmented sampler that targets the joint process $[X_t, Y_t]$ conditional on the observations V_i. The augmented observation operator is given by $L_{aug} = [L\ 0]$, the augmented drift for the target process is $b_{aug}(t, [x, y]) = [b(t, x), Dy]$ and the block diagonal dispersion coefficient is $\left[\begin{smallmatrix}\sigma & 0\\ 0 & I\end{smallmatrix}\right]$; the sampler uses the augmented guided proposal with the auxiliary linear process $[\tilde{X}, Y]$ that has a drift $\tilde{b}_{aug}(t, [x, y]) = [Bx + Cy, Dy]$ and the block diagonal dispersion coefficient $\left[\begin{smallmatrix}\tilde{\sigma} & 0\\ 0 & I\end{smallmatrix}\right]$. Samples of the conditional process X are obtained by marginalisation. To give a simple example, $\tilde{X}$ can be taken to be a sum of a Brownian motion and an integrated Brownian motion.

4. Backward filtering forward guiding for diffusions

In this section we present a novel Gibbs sampling algorithm for joint continuous-discrete smoothing of diffusions and parameter estimation. We will first precisely derive the posterior distribution that we target in Section 4.1. Next, in Section 4.5 we present details on initialisation of the algorithms, whereas steps for updating the path, initial state and parameters and given in sections 4.2, 4.3 and 4.4 respectively.

4.1. Likelihood calculations

The law $\mathbb{P}^\circ$ of the guided proposal approximates $\mathbb{P}^\star$ with tractable likelihood ratio. In the simple setting of known starting point x_0 and $k_i = \tilde{k}_i$ with $\tilde{k}_i$ derived from observation scheme (1.3) we have specified this ratio in (1.9). Dropping the requirement that $k_i = \tilde{k}_i$, we have for bounded measurable test-functions g

$$\mathbb{E}\left[g(X^\star) \mid X_0^\star = x_0\right] = \frac{\tilde{\rho}(0+, x_0)}{\rho(0+, x_0)} \mathbb{E}\left[g(X^\circ)\Psi(X^\circ)\left(\prod_{i=1}^{n} C_i(X_{t_i}^\circ)\right) \big| X_0^\circ = x_0\right], \tag{4.1}$$

where

$$C_i(x) = \begin{cases} k_i(x; v_i)/\tilde{k}_i(x; v_i) & \text{if} \quad 1 \le i \le n-1 \\ k_i(x; v_i)/(\tilde{k}_i(x; v_i)\tilde{k}_{i+}(x)) & \text{if} \quad i = n \end{cases}.$$

This follows from (1.9) upon correcting for the discrepancy between $\tilde{k}_i$ and k_i. Now we assume b, σ, $\tilde{p}$, $\{\tilde{k}_i\}$, $\{k_i\}$, $\pi(x_0)$ depend on an unknown parameter θ with prior $\kappa(\theta)$. We add a subscript θ if we wish to highlight this dependency.

A naive approach for sampling jointly the latent path $X^\star$ and parameter θ would consist of data-augmentation where one iteratively updates the latent path $X^\star$ proposing from X° conditionally on the parameter θ and θ conditionally on $X^\star$. It is well known that this yields a reducible algorithm if unknown parameters appear in the diffusivity σ (cf. [48]). The key idea put forward in [27] (see also [56] and [46]) is to use a noncentred parametrisation where the law of X° is casted as a pushforward of Wiener measure. It is this approach that we adopt here as well: as we assume existence of a strong solution to the SDE (1.7), there exists of a measurable map $\mathcal{GP}_\theta$ such that

$$X = \mathcal{GP}_\theta(X_0, Z),$$

where Z is a Wiener process in $\mathbb{R}^{d'}$. The process Z is referred to as the *innovation process*.

Below, we define algorithms 4.1, 4.3 and 4.4. These are to be combined in a Gibbs sampler to target the joint posterior distribution of (θ, x_0, Z) specified by the proper density

$$\frac{\kappa(\theta)\pi_\theta(x_0)\tilde{\rho}_\theta(0, x_0)}{\int \kappa(\theta)\pi_\theta(x_0)\rho_\theta(0, x_0)\,\mathrm{d}(x_0, \theta)} \Psi_\theta(\mathcal{GP}_\theta(x_0, Z)) \prod_{i=0}^{n} C_i(\mathcal{GP}_\theta(x_0, Z)_{t_i}), \tag{4.2}$$

with respect to the product measure of Lebesgue measure on (θ, x_0) and d'-dimensional Wiener measure on Z. In particular, samples of the latent path are obtained from samples (θ, x_0, Z) through $X = \mathcal{GP}_\theta(x_0, Z)$.

To see why (4.2) is true, if g is a bounded measurable function, then, with $\mathbb{W}$ denoting Wiener measure and X_0, Θ random variables drawn from the prior,

$$\begin{aligned} \mathbb{E}[g(\Theta, X_0, Z) \mid \mathcal{V}_0] &= \mathbb{E}[\mathbb{E}[g(\Theta, X_0, Z) \mid \mathcal{V}_0, \Theta, X_0] \mid \mathcal{V}_0] \\ &= \int \mathbb{E}[g(\Theta, X_0, Z) \mid \mathcal{V}_0, X_0 = x_0, \Theta = \theta]\xi(x_0, \theta; \mathcal{V}_0)\,\mathrm{d}(\theta, x_0) \end{aligned}$$

$$= \int g(\theta, x_0, z)\xi(x_0, \theta; \mathcal{V}_0)\frac{\tilde{\rho}_\theta(0+, x_0)}{\rho_\theta(0+, x_0)}\Psi_\theta(GP_\theta(x_0, z)) \\ \times \prod_{i=1}^{n} C_i(\mathcal{G}\mathcal{P}_\theta(x_0, z)_{t_i}) \,\mathrm{d}\mathbb{W}(z)\,\mathrm{d}\theta\,\mathrm{d}x_0$$

where

$$\xi(x_0, \theta; \mathcal{V}_0) = \frac{\kappa(\theta)\pi_\theta(x_0)k_0(x_0, v_0)\rho_\theta(0+, x_0)}{\int \kappa(\theta)\pi_\theta(x_0)k_0(x_0, v_0)\rho_\theta(0+, x_0)\,\mathrm{d}(\theta, x_0)}$$

is the density of (x_0, θ) conditional on $\mathcal{V}_0$. Finally, use $\rho_\theta(0, x_0) = k_0(x_0, v_0)$ $\rho_\theta(0+, x_0)$ and likewise for $\tilde{k}_0$, $\tilde{\rho}_\theta$ and multiply both the numerator and denominator by $\tilde{k}_0(x_0; v_0)$ and "absorb" the ratio $k_0(x_0; v_0)/\tilde{k}_0(x_0; v_0)$ into the product by starting at $i = 0$ rather than $i = 1$.

4.2. Smoothing from known initial state and parameter

In this conditional update step we assume that the initial state and parameter are either known or conditioned upon. This means sampling from the smoothing distribution which requires the likelihood ratio of $\mathbb{P}^\star$ with respect to $\mathbb{P}^\circ$ as specified in (4.1). Recall the definition of Ψ in (1.10).

For this step, we choose a tuning parameter $\lambda \in [0, 1)$.

Algorithm 4.1. *Smoothing step for fixed θ and fixed initial state x_0.* Suppose the current iterate is (θ, x_0, Z) and $X = \mathcal{G}\mathcal{P}_\theta(x_0, Z)$.

1. **Compute the guiding term**. Initialise $H(t_n+)$, $F(t_n+)$, $c(t_n+)$ via (2.10).
   For $i = n$ to 0
   (a) For $t \in (t_i, t_{i+1}]$, backwards solve the ordinary differential equations (2.8)
   (b) Compute $H(t_i)$, $F(t_i)$, $c(t_i)$ from (2.9)
2. **Crank-Nicolson step**.
   (a) Sample independently a Wiener process W and set
   $$Z^\circ = \lambda Z + \sqrt{1 - \lambda^2} W.$$
   Compute
   $$X^\circ = \mathcal{G}\mathcal{P}_\theta(x_0, Z^\circ).$$
   (b) Compute
   $$A = \frac{\Psi(X^\circ)}{\Psi(X)} \prod_{i=1}^{n} \frac{C_i(X^\circ_{t_i})}{C_i(X_{t_i})}.$$
   Draw $U \sim \mathcal{U}(0, 1)$. If $U < A$ replace $X = X^\circ$ and $Z = Z^\circ$.

Repeating step 2 multiple times constitutes a procedure to sample from the smoothing distribution, in which case step 1 only needs to be done once. We have used the information filter from Theorem 2.5. Clearly, with straightforward modifications it can be adjusted to other filtering equations from Section 2.

ODEs for $H(t)$ and $F(t)$ are calculated on $t \in [t_0, t_n]$ using time discretisation. Once these have been calculated and stored (on a grid of timepoints), the remainder of the algorithm consists of preconditioned Crank–Nicolson steps on the Wiener increments. After "burn in", the sample paths generated by this algorithm are from the smoothing distribution. Note that since we assume the initial state x_0 to be known and the parameter θ to be fixed, $c(t)$, in fact, does not need to be computed. As we will shortly extend the presented algorithm to include uncertatinty over the initial state and parameter estimation we have already included the ODE and the update formula for $c(t)$.

While Algorithm 4.1 is theoretically valid for any fixed value of λ, its efficiency strongly depends on the particular choice of this parameter. Instead of a fixed value of λ, we can choose it randomly at each iteration of step (3a): the acceptance probability in step (3b) is not affected by its value (cf. the discussion after Algorithm 1 in [56]).

Remark 4.2. If the measurement error is close to zero, then the (Euler) discretisation of the guided proposal is a delicate matter due to the behaviour of the guiding term just prior to observation times. As discussed in Section 5 of [56] a time change and scaling of the process can alleviate discretisation errors. For completeness, we present this approach in Appendix E.

4.3. Initial state updating

We choose to update the initial state conditional on (Z, θ).

The expression for A follows from the target density in (4.2). We choose a Markov kernel q (which may depend on θ) for proposing a new state x_0°.

Algorithm 4.3. *Initial state updating step*. Suppose the current iterate is (θ, x_0, Z) and $X = \mathcal{GP}_\theta(x_0, Z)$.

1. Propose $x_0^\circ \sim q_\theta(\cdot \mid x_0)$.
2. Compute the corresponding guided proposal $X^\circ = \mathcal{GP}_\theta(x_0^\circ, Z)$.
3. Compute

$$A = \frac{q_\theta(x_0 \mid x_0^\circ)}{q_\theta(x_0^\circ \mid x_0)} \frac{\pi(x_0^\circ)\tilde{\rho}(0, x_0^\circ)}{\pi(x_0)\tilde{\rho}(0, x_0)} \frac{\Psi(X^\circ)}{\Psi(X)} \prod_{i=0}^{n} \frac{C_i(X_{t_i}^\circ)}{C_i(X_{t_i})}.$$

   Draw $U \sim \mathcal{U}(0,1)$. If $U < A$ replace $X = X^\circ$ and $x_0 = x_0^\circ$.

An independence sampler can be obtained by taking $q_\theta(\cdot \mid x_0) = \pi^\circ(\cdot)$ defined by

$$\pi^\circ(x_0) = \frac{\tilde{\pi}(x_0)\tilde{\rho}(0, x_0)}{\int \tilde{\pi}(x_0)\tilde{\rho}(0, x_0)\, \mathrm{d}x_0} \tag{4.3}$$

for specified nonnegative $\tilde{\pi}$. Naturally, in case π is itself Gaussian an obvious choice is to take $\tilde{\pi} = \pi$.

4.4. Parameter updating

Choose a Markov kernel q for proposing new value for θ.

Algorithm 4.4. *Parameter updating step.* Suppose the current iterate is (θ, x_0, Z) and $X = \mathcal{GP}_\theta(x_0, Z)$.

1. Propose θ° from the proposal kernel q.
2. If (H, F, c) (defining the guiding term) depend on the parameter, then recompute the guiding term with parameter θ° using step 1 of Algorithm 4.1.
3. Compute the corresponding guided proposal $X^\circ = \mathcal{GP}_{\theta^\circ}(x_0, Z)$.
4. Compute

$$A = \frac{q(\theta \mid \theta^\circ)}{q(\theta^\circ \mid \theta)} \frac{\kappa(\theta^\circ)\tilde{\rho}_{\theta^\circ}(0, x_0)}{\kappa(\theta)\tilde{\rho}_\theta(0, x_0)} \frac{\Psi_{\theta^\circ}(X^\circ)}{\Psi_\theta(X)} \prod_{i=0}^{n} \frac{C_{i,\theta^\circ}(X^\circ_{t_i})}{C_{i,\theta}(X_{t_i})}.$$

Draw $U \sim \mathcal{U}(0, 1)$. If $U < A$ replace $X = X^\circ$ and $\theta = \theta^\circ$.

The following Gaussian conjugacy result is sometimes useful; its proof is an elementary consequence of Girsanov's theorem.

Proposition 4.5. Assume a diffusion X with drift of the form

$$b(t, x) = \varphi_0(t, x) + \sum_{k=1}^{K} \theta_k \varphi_k(t, x),$$

where both the diffusion coefficient and the initial distribution of X_0 are independent of θ. If the full path $(X_t, t \in [0, t_n])$ is observed and a priori $\theta \sim \mathrm{N}(0, \Gamma_0^{-1})$, then $\theta \mid X \sim \mathrm{N}(\Gamma^{-1}\mu, \Gamma^{-1})$ with

$$\mu = \int_0^{t_n} \Phi(t, X_t)'(\,\mathrm{d}X_t - \varphi_0(t, X_t)\,\mathrm{d}t)$$
$$\Gamma = \int_0^{t_n} \Phi(t, X_t)' a^{-1}(t, X_t)\Phi(t, X_t)\,\mathrm{d}t + \Gamma_0,$$

where $\Phi(t, x) = (\varphi_k(t, x))_{1 \le k \le K}$.

4.5. Initialisation

While any value of (θ, x_0) within the support of their prior is possible, we propose to choose an initial value θ and subsequently sample $x_0 \sim \pi^\circ$ for specified nonnegative $\tilde{\pi}$. Next, we sample a Wiener process Z on $[0, t_n]$. The corresponding guided proposal is obtained from $X^\circ = \mathcal{GP}_\theta(x_0, Z)$, which can be computed using

$$\mathrm{d}X^\circ_t = b(t, X^\circ_t)\,\mathrm{d}t + a(t, X^\circ_t)\left(F(t) - H(t)X^\circ_t\right)\,\mathrm{d}t + \sigma(t, X^\circ_t)\,\mathrm{d}Z_t.$$

Next, we initialise X by defining $X = (X^\circ_t,\, t \in [0, t_n])$.

4.6. Blocking strategy

In this article, for comparisons of the effects of blocking we always adopt a "chequerboard" updating strategy. More precisely, we say that we use "blocks of length k" (for an even k) to mean that the following strategy is adopted for updating the path:

1. Initialise $X_{(0:n)}$.
2. For $i = 1, \dots, \lfloor n/k \rfloor$, sample bridges $X_{(ki-k:ki)}$, conditional on X_{ki-k}, X_{ki} and $\{V_{ki-j}; j = 1, \dots, k-1\}$.
3. Sample the last segment $X_{(\lfloor n/k \rfloor k:n)}$ conditional on $X_{\lfloor n/k \rfloor k}$ and $\{V_j; j = \lfloor n/k \rfloor k, \dots, n\}$.
4. For $i = 1, \dots, \lfloor n/k \rfloor - 1$, sample bridges $X_{(ki-k/2:ki+k/2)}$, conditional on $X_{ki-k/2}$, $X_{ki+ki/2}$ and $\{V_{ki+j}; j = -k/2+1, \dots, k/2-1\}$.
5. Sample the first segment $X_{(0:k/2)}$ conditional on X_0, $X_{k/2}$ and $\{V_j; j = 1, \dots, k/2\}$.
6. Sample the last segment $X_{(\lfloor n/k \rfloor k-k/2:n)}$ conditional on $X_{\lfloor n/k \rfloor k-k/2}$ and $\{V_j; j = \lfloor n/k \rfloor k - k/2 + 1, \dots, n\}$.

Naturally, the algorithm above describes only smoothing. In order to transform the above into an inference algorithm a parameter update step described in Algorithm 4.4 may be introduced in between steps 3 and 4 or after step 6 or both.

5. Continuous-discrete smoothing for high dimensional diffusion processes

For high dimensional SDEs with $d \gg 1$, the computation of the $d \times d$ backward filtered covariance $P(t)$ in (2.11) becomes prohibitive in an implementation in dense matrix algebra (and likewise the equations for its inverse). Another numerical difficulty is the computation of the likelihood accounting for the difference in a and $\tilde{a}$ in case these are unequal. In this section we do not consider the latter problem, and therefore, restrict our attention to the high dimensional situation where $a = \tilde{a}$. In this case the linear filtering step is the numerical bottleneck and not the sampling step. Specifically, if the process X and the observations are of the form

$$\begin{aligned} \mathrm{d}X_t &= (B(t)X_t + F(t, X_t))\,\mathrm{d}t + \sigma_t\,\mathrm{d}W_t, \quad X_0 \sim p(x_0), \\ Y_i &= LX_{t_i} + \eta_i, \quad \eta_i \sim \mathrm{N}(0, \Sigma_i), \end{aligned} \tag{5.1}$$

then a natural choice is to take $\tilde{B}(t) = B(t)$ and $\tilde{\sigma}(t) = \sigma(t)$.

Even if $\tilde{B}$ and $\tilde{\sigma}$ are sparse, $P(t)$ typically is not. But if those operators that describe the drift and dispersion of $\tilde{X}$ are *local*, then one may be able to approximate P by a sparse matrix. Instead of stating a formal definition we provide an illustrative example, which can be seen as a variant of the stochastic heat equation on a graph with drift. Consider the d-dimensional diffusion (X_t) for which a state x is understood as a discrete approximation to a function $f\colon [0,1] \to \mathbb{R}$, so that in particular, a coordinate $X^{(i)}$ corresponds to an evaluation of f in $\frac{i-1}{d-1} \in [0,1]$. More concretely, let X be defined as a solution to

$$\mathrm{d}X_t = -\tfrac{\sigma^2}{2}(\Lambda + cI)X_t \,\mathrm{d}t + \sigma \,\mathrm{d}W_t, \tag{5.2}$$

with $\sigma, c > 0$ and

$$\Lambda = \begin{bmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 1 \end{bmatrix}, \tag{5.3}$$

the graph Laplacian of the linear graph with d vertices corresponding to the coordinates. Λ acts as a discrete approximation to the second derivative. Λ is a local operator in the sense of having a representation by a banded matrix (this corresponds to all interactions in the linear graph being limited to only those between the neighbours). As a result, matrix-vector operations (matvecs) Λx can be computed efficiently in $\mathcal{O}(d)$ time. X is a Gauss–Markov process with stationary distribution $\mathrm{N}(0, (\Lambda + cI)^{-1})$ (as $\Sigma = (\Lambda + cI)^{-1}$ is a solution to the continuous time Lyapunov equation $B\Sigma + \Sigma B' + C = 0$, where $B = -\sigma^2/2(\Lambda + cI)$). Thus, in the stationary regime, $X(t)$ at a fixed time t is a draw from a high-dimensional Gaussian law approximating a spatial Gaussian process on $[0,1]$ with continuous realisations of Hölder regularity just below $\frac{1}{2}$. While $(\Lambda + cI)^{-1}$ is not a sparse matrix, the entries decay exponentially moving away from the diagonal. By Proposition 2.7, the same holds for $P(t)$. Good, sparse approximations are obtained by setting $(\Lambda + cI)^{-1}$ or $P(t)$ to zero outside of a fixed band.

In contrast, the forward simulation and the likelihood evaluation happen in $\mathbb{R}^d$ and are fast, as long as fast matvec operations with the operator H are available. This implies that the actual bottleneck of our approach is in solving a continuous-discrete linear filtering problem and we can rely on a long sequence of work in this direction.

From the various possibilities we consider but two: a sampling approach related to the ensemble Kalman filter and a sparsity enforcing solver for (2.12).

5.1. Ensemble backward filter

Instead of solving (2.12) one can exploit that for the linear backward process with a reversed time axis

$$\mathrm{d}\bar{X}_s = (\bar{B}(s)\bar{X}_s + \bar{\beta}(s))\,\mathrm{d}s + \bar{\sigma}(s)\,\mathrm{d}\bar{W}_s, \bar{X}_0 \sim \mathrm{N}(\nu_{t_{i+1}}, P_{t_{i+1}})$$

for $T - s \in (t_i, t_{i+1})$ with $\bar{B}(s) = -\tilde{B}(T-s)$, $\bar{\beta}(s) = -\tilde{\beta}(T-s)$ and $\bar{\sigma}(s) = \tilde{\sigma}(T-s)$, it holds

$$\mathbb{E}\left[\bar{X}_s\right] = \nu(T-s) \quad \text{and} \quad \operatorname{Var}(\bar{X}_s) = P(T-s).$$

Here, $\mathrm{d}\bar{W}_s$ can be taken to be an Itô integral; as $\tilde{\sigma}$ does not depend on space, there is no need for an Itô correction.

This can be used to obtain estimates of $\nu(s)$ and a low rank approximation of $P(s)$ by sampling $\bar{X}^{(k)}$ a number of K times with $K \ll d$ and approximating by the empirical covariance $\Sigma = \frac{1}{K-1}\sum \bar{X}^{(k)}_{T-s} X^{(k)\prime}_{T-s}$. To obtain a positive definite (and invertible) estimate of P, one employs localization and sets $\hat{P} = \Sigma \circ \Lambda$ where $\circ$ is the Hadamard (elementwise) product with a positive definite sparse localisation matrix Λ chosen to ignore long range dependencies, see [35]. The observations are taken into account by way of the usual ensemble Kalman filter

$$\bar{X}^{(k)}_{T-t_i} = X^{(k)}_{T-t_i+} + \hat{P}(t+)L_i'\left(L_i\hat{P}(t+)L_i' + \Sigma\right)^{-1}(v_i - L_i X^{(k)}_{T-t_i+}),$$

see [23].

5.2. *Forced sparsity*

Secondly, we consider sparsity enforcing backward integrators, where we set

$$P(t-h) = \operatorname{drop}_\epsilon\left(P(t) + (-\tilde{B}(t)P(t) - P(t)\tilde{B}(t)' + \tilde{a}(t))h\right).$$

Here

$$(\operatorname{drop}_\epsilon A)_{ij} = \begin{cases} A_{ij} & |A_{ij}| > \varepsilon \\ 0 & \text{otherwise} \end{cases}.$$

If P is well-approximated by a sparse matrix – as illustrated for the case (5.2) – then $\operatorname{drop}_\epsilon P_h$ will be sparse. We explore this approach in section 6.3.

Remark 5.1. A particularly relevant case of (5.1) is a process X that is obtained from a spatial discretisation of the stochastic partial differential equation

$$\mathrm{d}\mathbb{X}_t = (-\mathbf{A}\mathbb{X}_t + \mathbf{F}(t, \mathbb{X}_t))\,\mathrm{d}t + \boldsymbol{\sigma}_t\,\mathrm{d}\mathbb{W}_t, \quad \mathbb{X}_0 = \mathbb{x}_0, \tag{5.4}$$

taking solutions in a Hilbert space $\mathbb{H}$, where $\mathbf{A}$ is a densely defined, positive definite operator and $\mathbb{W}$ is a $\mathbf{Q}$-Wiener process with a covariance operator $\mathbf{Q}$ given by a positive definite trace class operator, see [18].

6. Examples

In all examples, the differential equations for $H(t)$, $F(t)$ and $c(t)$ have been solved using the 7th order Runge–Kutta solver (cf. [61]). The guided proposal is obtained by solving its SDE using Euler discretisation. The code is available in the folders `scripts/papers` of the package `BridgeSDEInference.jl`[1] [42] and `scripts` of the package `BridgeSPDE.jl`[2]. Both packages build upon [52], for the programming language Julia [11].

6.1. Lorenz system

The Lorenz system is notable for having chaotic solutions for certain parameter values and initial conditions. It is described by the SDE with the drift and dispersion coefficient given by

$$b(x) = \begin{bmatrix} \theta_1(x_2 - x_1) \\ \theta_2 x_1 - x_2 - x_1 x_3 \\ x_1 x_2 - \theta_3 x_3 \end{bmatrix} \qquad \text{and} \qquad \sigma = \sigma_0 I_{3\times 3}.$$

Data: We initialised the process at $x_0 = \begin{bmatrix}1.5 & -1.5 & 25\end{bmatrix}'$ and simulated it on $[0,2]$ with mesh-width 2e-4 and parameters $\theta = \begin{bmatrix}10 & 28 & 8/3\end{bmatrix}'$, $\sigma_0 = 3$. Only the 2nd and 3rd coordinate were observed so that L_i is a 2×3 matrix with the first row [0, 1, 0] and the second row [0, 0, 1]. We then defined two datasets.
Dataset 1: 200 observations were retained at times $0.01, 0.02, \dots, 2$, where we took $\Sigma_i = 5 \cdot I$ as the covariance matrix for the noise.
Dataset 2: 10 observations were retained at times $0.2, 0.4, \dots, 2$, where we took $\Sigma_i = 0.05 \cdot I$. The observed data are shown in figures 1 and 2. Note that the first coordinate process $\{X_{1,t}; t \in [0,T]\}$ remains latent in both cases.

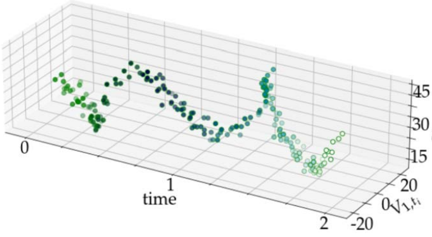

Fig 1: Dataset 1

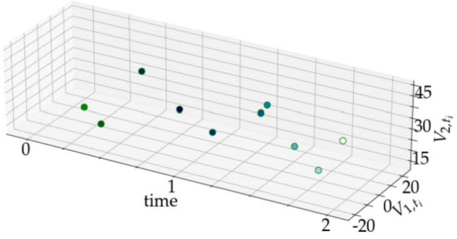

Fig 2: Dataset 2

Algorithm details: Overall, six inference algorithms have been run, three on each dataset. The procedures were inferring parameter θ, whereas parameter σ_0 was assumed to be known. All four parameters could have been inferred simultaneously; however, by fixing σ_0 it was possible to illustrate the differences in performances of various algorithms more clearly. Indeed, a varying σ_0 locally influences the speed at which chains on a path space, as well as parameter chains mix and this makes it more difficult to disentangle the contribution due to efficiency of the algorithm from that of a locally elevated or decreased volatility.

[1] https://github.com/mmider/BridgeSDEInference.jl/tree/master/scripts/papers/cts_discr_smooth

[2] https://github.com/mschauer/BridgeSPDE.jl/blob/master/scripts/smoothing.jl

We illustrate the problem of inference for the volatility coefficient on a more challenging example in the following section.

We ran the following algorithms:

- Dataset 1
  - The algorithm of [55]. It is a special case of the algorithm proposed in this article, in which a path—instead of being imputed in full—is updated in blocks of length 2, as described in Section 4.6.
  - Inference algorithm with blocking, with blocks of length 8.
  - Inference algorithm with no blocking (where each proposal path is imputed in full).
- Dataset 2
  - Inference algorithm with no blocking.
  - Inference algorithm with blocks of length 2.
  - Inference algorithm with blocks of length 2 and adaptation of proposals laws.

We set the following Gaussian priors over the parameter θ and the value of the starting point:

$$\theta \sim \mathrm{N}(0, \mathrm{diag}(10^3, 10^3, 10^3)), \quad X_0 \sim \mathrm{N}(x_0, \mathrm{diag}(400, 20, 20)).$$

Setting a Gaussian prior over θ made it possible to implement conjugate updates for θ (cf. Proposition 4.5 or [59]). Each block had its own persistence parameter λ_i of the preconditioned Crank–Nicolson scheme (in case of no-blocking a single parameter λ was used). For each block, λ_i was tuned adaptively so as to target the acceptance probability of the imputation step for a given block to be 0.234 (cf. Section 3 of [47]). The updates of the initial position were always done jointly with the updates of the path in the first block (or jointly with the entire path updates in case of no blocking).

We chose the auxiliary process according to the strategy **B**, where the initial guess for $\tilde{x}(t_i)$ was taken to be $\begin{bmatrix}\Xi_i & v_{i,1} & v_{i,2}\end{bmatrix}'$, $\Xi_i = 25$, $(i = 1, \dots, n)$, and where the values of Ξ_i were updated over the course of running an MCMC sampler: during the first 2500 iterations, once in every 500 steps we used an empirical average of X_{1,t_i} to re-define Ξ_i. Note that this was done in all six experiments.

Additionally, in the experiment with the “adaptation”, we used strategy **E**, where the adaptation time was set to 2500 iterations and adaptation updates were done once in every 500 steps. The weights were changed from, initially, $(1, 0)$ (the first value representing the weight for strategy **B**, the second one for strategy **C**), through $(0.7, 0.3)$ and $(0.4, 0.6)$ up to $(0.2, 0.8)$ at iteration 1500 (from then on the weights remained unchanged).

We set the density of the imputation grid to $2 \cdot 10^{-4}$. The simulations took respectively: 784s, 570s, 198s, 137s, 260s and 777s to complete on an i5-5250U CPU, 1.60GHz.

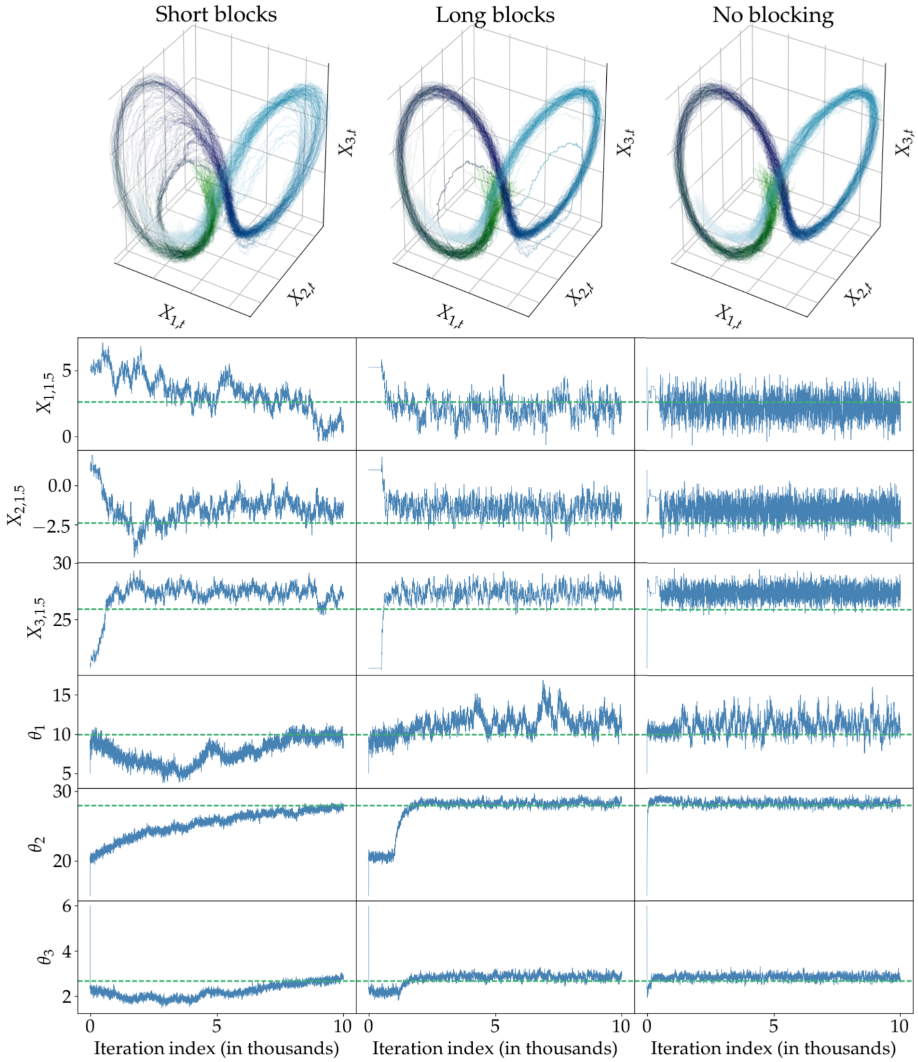

Fig 3: Lorenz system: results on **dataset 1**. Left column: results of a single MCMC run with blocks set to comprise of two inter-observation intervals (as in [55]); middle column: blocks comprising of eight inter-observation intervals; right column: no blocking. Top plots: paths sampled by the MCMC samplers (1 in every 100 sampled paths); Rows 2–4: marginal distribution of the coordinates of X at time $t = 1.5$. Rows 5–7: traceplots for the updated parameters θ.

Results: The results of the experiments associated with the first dataset are shown in Figure 3 and they illustrate the outcome of only the first 10^4 iterations. The summary concerning time-adjusted effective sample size that is based on the entire chain is given in Table 1.

The green dashed lines in the top-most three traceplots indicate the values taken by the diffusion trajectory that was used to produce the data. As we only have one realisation of the path and finitely many observations, the conditional distribution (given the data) is not centred at this line; however, we can expect the samples to be close to them. The green dashed lines in the bottom three traceplots indicate the values of the parameters that were set to simulate the data.

Under the first observational setting the discrepancy between an approximate and a true guiding term is small and thus blocking becomes a hindrance that decelerates mixing of the parameter chains and the path chain. This is clearly illustrated in the left-most column of Figure 3, which corresponds to the algorithm of [55], where the block-length is set to 2. By increasing the block-length to 8 (centre column) the mixing is improved and by removing it altogether and updating the entire path at once even better mixing can be achieved (right column). The same conclusions are strongly supported by the results summarised in Table 1, where the difference in the time-adjusted effective sample size is in excess of 2 orders of magnitude (we used a standard definition of the effective sample size, recommended by R. Neal in the panel discussion of [37]: $\mathtt{ESS}(\theta) := n_{\mathtt{ps}}/(1+2\sum_{k=1}^{\infty}\rho_k(\theta))$, with $n_{\mathtt{ps}}$ denoting the number of posterior samples and ρ_k the autocorrelation at lag k; it is implemented as a function `ESS` in `R` programming language).

In the top row of Figure 3 we plotted a thinned chain of the imputed trajectories (1 in every 100 sampled paths is given; note that unlike figures 1 and 2, the temporal component is represented only by the changing colour of the trajectories, whereas each of the three axes have purely spatial meaning and correspond to the coordinates of the process X). Notice how short blocks cause a slowdown in mixing on a path space, effectively disallowing the sampler to perform large moves. This is expected to be of a particular hindrance to sampling of multimodal diffusions.

The results from the first dataset clearly illustrate that removing the restriction on the block length can have a positive effect on the efficiency of sampling on a path space. Nonetheless, some caution must be executed in extrapolating those conclusions. To see this, consider the second dataset. It is qualitatively different from the first one, in that the inter-observation distance is large enough for the non-linear dynamics of the target process to become pronounced between any two observations, whereas the decreased observational noise elevates the impact of the guiding term (for this reason it is also considered to be a substantially more difficult problem, for which a smaller number of numerical schemes can be applied to).

The inference results obtained on the second dataset are gathered in Figure 4. In this setting, using blocking yields similar results as imputing the path in full. In fact, as summarised by Table 2, the values of taESS corresponding to parameter chains are approximately twice as large when blocking is used (as compared to a no-blocking scheme). However, it is possible to improve the mixing of the chains even further—as depicted in the centre column of Figure 4—by designing better proposal laws. We used a scheme from **E** simply to

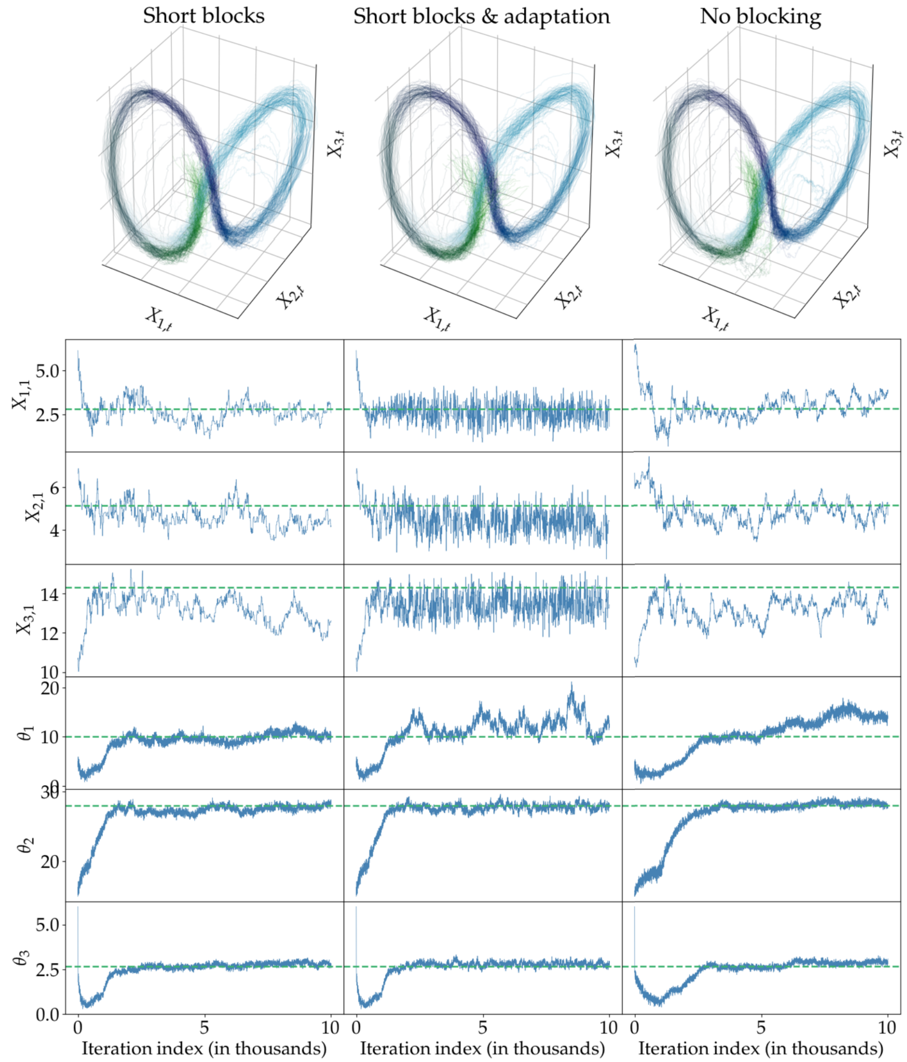

Fig 4: Lorenz system: results on **dataset 2**. Left column: results of a single MCMC run with no blocking; middle column: blocks comprising of two inter-observation intervals; right column: blocks of length 2 and adaptive tuning of the auxiliary law according to $b^{(aux)} = wb^{(7.1)} + (1-w)b^{(C)}$ (see Section 4.6 for details). Top plots: paths sampled by the MCMC samplers (1 in every 100 sampled paths); Rows 2–4: marginal distribution of the coordinates of X at time $t = 1$. Rows 5–7: traceplots for the updated parameters θ.

motivate the usefulness of making the coefficients of the auxiliary diffusion time-dependent. The topic of optimal choice of the auxiliary law is beyond the scope of this paper.

Table 1

Time-adjusted effective sample size (taESS) and effective sample size (ESS) for the experiments with the first dataset of the Lorenz example. Based on chains of length 10^5*. Best results are written in bold.*

		taESS (ESS)	
	short blocks	long blocks	no blocking
$X_{1,1.5}$	0.164 (128.6)	1.991 (1134.9)	**52.932** (**10480.3**)
$X_{2,1.5}$	0.447 (350.1)	5.164 (2943.4)	**115.610** (**22890.5**)
$X_{3,1.5}$	0.545 (427.0)	1.468 (836.6)	**121.568** (**24070.2**)
θ_1	0.039 (30.6)	1.643 (936.6)	**23.194** (**4592.4**)
θ_2	0.081 (63.7)	0.366 (208.6)	**77.675** (**15379.5**)
θ_3	0.057 (44.5)	3.108 (1771.7)	**55.141** (**10917.7**)

Table 2

Time-adjusted effective sample size (taESS) and effective sample size (ESS) for the experiments with the second dataset of the Lorenz example. Additional comparison for the experiments with the auxiliary law chosen according to the strategy ***A****. Based on chains of length* 10^5*. Best results are written in bold.*

		taESS (ESS)		
	short blocks	short blocks & adpt	no blocking	strategy **A**
$X_{1,1}$	1.328 (332.1)	**4.140** (**3216.6**)	1.271 (175.3)	1.17 (56.2)
$X_{2,1}$	1.566 (391.4)	**5.162** (**4011.0**)	1.618 (223.3)	0.600 (28.8)
$X_{3,1}$	1.366 (341.2)	**4.608** (**3580.4**)	1.469 (202.7)	0.346 (16.5)
θ_1	0.416 (103.9)	**0.602** (**467.5**)	0.280 (38.6)	0.330 (15.8)
θ_2	**1.307** (326.8)	0.963 (**748.5**)	0.589 (81.2)	0.901 (43.3)
θ_3	**0.76** (190.6)	0.563 (**437.2**)	0.488 (67.3)	0.196 (9.39)

Problems encountered in practice might fall anywhere on the spectrum spanned by the two datasets above or exhibit peculiarities of its own. We therefore believe that removing restriction on the block length—as it is done in this article—is a particularly practical improvement.

For reference we also ran the algorithm on the second dataset with the choice of the simplest strategy **A** and reported the result in Table 2.

6.2. Prokaryotic auto-regulatory gene network

We consider a simplified model of the auto-regulated protein transcription studied in [26, 28]. It is based on the stochastic model describing λ repressor protein cI of phage λ in Escherichia coli introduced in [2]. The simplified model is characterised by eight biomolecular reactions: the reversible repression of transcription in which a protein dimer $\mathtt{P}_2$ attaches to certain sites on the DNA:

$$\mathcal{R}_1\colon\ \mathtt{DNA}+\mathtt{P}_2\to\mathtt{DNA}\cdot\mathtt{P}_2,\qquad \mathcal{R}_2\colon\ \mathtt{DNA}\cdot\mathtt{P}_2\to\mathtt{DNA}+\mathtt{P}_2;$$

the transcription of DNA into mRNA and the subsequent translation and protein folding:

$$\mathcal{R}_3\colon\ \mathtt{DNA}\to\mathtt{DNA}+\mathtt{RNA},\qquad \mathcal{R}_4\colon\ \mathtt{RNA}\to\mathtt{RNA}+\mathtt{P};$$

the reversible protein dimerisation:

$$\mathcal{R}_5\colon\ 2\mathtt{P} \to \mathtt{P}_2, \qquad \mathcal{R}_6\colon\ \mathtt{P}_2 \to 2\mathtt{P};$$

and the mRNA and protein degradation:

$$\mathcal{R}_7\colon\ \mathtt{RNA} \to \emptyset, \qquad \mathcal{R}_8\colon\ \mathtt{P} \to \emptyset.$$

The total number of DNA strands is assumed to be fixed: $\mathtt{DNA} + \mathtt{DNA}\cdot\mathtt{P}_2 = K$. It is well known that such system admits a diffusion approximation (called the chemical Langevin equation) and for the given system above, an associated SDE is given by:

$$\mathrm{d}X_t = S(\theta \circ h(X_t))\,\mathrm{d}t + S \odot \gamma(\theta \circ h(X_t))\,\mathrm{d}W_t, \quad t \in [0,T], \quad X_0 = x_0,$$

where $\circ\colon \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ is a component-wise multiplication:

$$(\mu \circ \nu)_i = \mu_i \nu_i, \quad i = 1, \dots, d,$$

the operation $\odot\colon \mathbb{R}^{d\times d'} \times \mathbb{R}^{d'} \to \mathbb{R}^{d\times d'}$ is defined via:

$$(M \odot \mu)_{i,j} = M_{i,j}\mu_j, \quad i = 1, \dots, d;\ j = 1, \dots, d',$$

the function $\gamma\colon \mathbb{R}^d \to \mathbb{R}^d$ is a component-wise square root:

$$(\gamma(\mu))_i = \sqrt{\mu_i}, \quad i = 1, \dots, d,$$

S is the stoichiometry matrix:

$$S = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & -2 & 2 & 0 & -1 \\ -1 & 1 & 0 & 0 & 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

and the function h is given by:

$$h(x) = (x_3x_4, K - x_4, x_4, x_1, x_2(x_2-1)/2, x_3, x_1, x_2)'.$$

For this SDE $X_t = (\mathtt{RNA}_t, \mathtt{P}_t, (\mathtt{P}_2)_t, \mathtt{DNA}_t)$ and W is an 8-dimensional Brownian motion. For more details about the model and how to derive chemical Langevin equations see [26].

We simulated three datasets using the Gillespie algorithm. In all three cases the process was started from $x_0 = (8, 8, 8, 5)$ and we set $K = 10$.

Dataset 1: First, we reproduced the observational scheme of [28], where noisy estimates of the total number of proteins that are not attached to any DNA strands are recorded at integer times:

$$V_i = \begin{bmatrix} 0 & 1 & 2 & 0 \end{bmatrix} X_i + \eta_i, \quad \eta_i \sim \mathrm{N}(0, 2^2), \quad i = 1, \dots, 100. \tag{6.1}$$

Dataset 2: Next, in addition to observations from (6.1), we assumed that for each 8 observations of protein counts it is possible to obtain one noisy count of `RNA`, i.e. that:

$$V_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 2 & 0 \end{bmatrix} X_i + \eta_i, \quad \eta_i \sim N\left(0, \begin{bmatrix} 1^2 & 0 \\ 0 & 2^2 \end{bmatrix}\right), \quad i = 8, 16, \dots, 96, \tag{6.2}$$

and that for $i \in \{1, 2, \dots, 100\} \backslash \{8, 16, \dots, 96\}$, V_i is given by (6.1).

Dataset 3: Finally, we took the second dataset and permuted the noisy `RNA` counts randomly. More precisely, if $V_i^{(j)}$ denotes the i^{th} observations from the j^{th} dataset ($i = 1, \dots, 100$; $j = 2, 3$) then:

$$V_i^{(3)} = \begin{cases} V_i^{(2)}, & i \in \{1, 2, \dots, 100\} \backslash \{8, 16, \dots, 96\}, \\ \begin{bmatrix} V_{1,\zeta(i)}^{(2)} & V_{2,i}^{(2)} \end{bmatrix}, & i \in \{8, 16, \dots, 96\}, \end{cases}$$

with $\zeta(i)$ denoting the i-th element of a random permutation ζ of $(8, 16, \dots, 96)$.

Algorithm details: We ran the inference algorithm using guided proposals without blocking. Note that due to a rectangular shape of the volatility coefficient it is impossible to employ guided proposals with blocking and a non-zero persistence parameter λ of the preconditioned Crank–Nicolson scheme. Consequently, one can employ the algorithm of [55] only in a special case when the observations are made on a sufficiently dense time-grid, for which it is possible to set $\lambda = 0$.

We set the prior distribution over the starting position to be $X_0 \sim \mathrm{N}(x_0, I_{4\times 4})$ and, following [28], a prior distribution over the logarithm of each inferred parameter to be $\texttt{Unif}([-7, 2])$. We remark that the priors chosen by [28] that truncate the natural support of the parameters might be problematic, especially for the first dataset, as the data are not informative enough to narrow down the support of some of the posteriors, causing the parameter chains to hit against the boundaries of the priors. Instead, using an exponential prior over each θ_i—this being the maximum entropy distribution of all continuous distributions supported on $[0, \infty)$—might be a more suitable choice. For the sake of fair comparison we stick with the choice made by [28].

The MCMC sampler started off with updating each log-transformed parameter separately using Metropolis–Hastings algorithm with random walk proposals and adaptation as described in Section 3 of [47]. After the first $12 \cdot 10^3$ steps of parameter updates we computed the empirical covariance of the parameter chains and switched to joint proposals of [31] (we used a modified version from Section 2 of [47]). We kept updating the empirical covariance matrix once in every 100 steps of the algorithm. We chose the auxiliary law according to the strategy **B**, where during the first 10^4 iterations of the Markov chain we updated our guesses for the latent positions of the path at the observation times once in every 200 iterations.

Results: Figure 5 summarises the results from running inference on each of the three datasets. For each log-transformed parameter we give traceplots of the Markov chain and the resulting marginal density plots (with burn-in set to

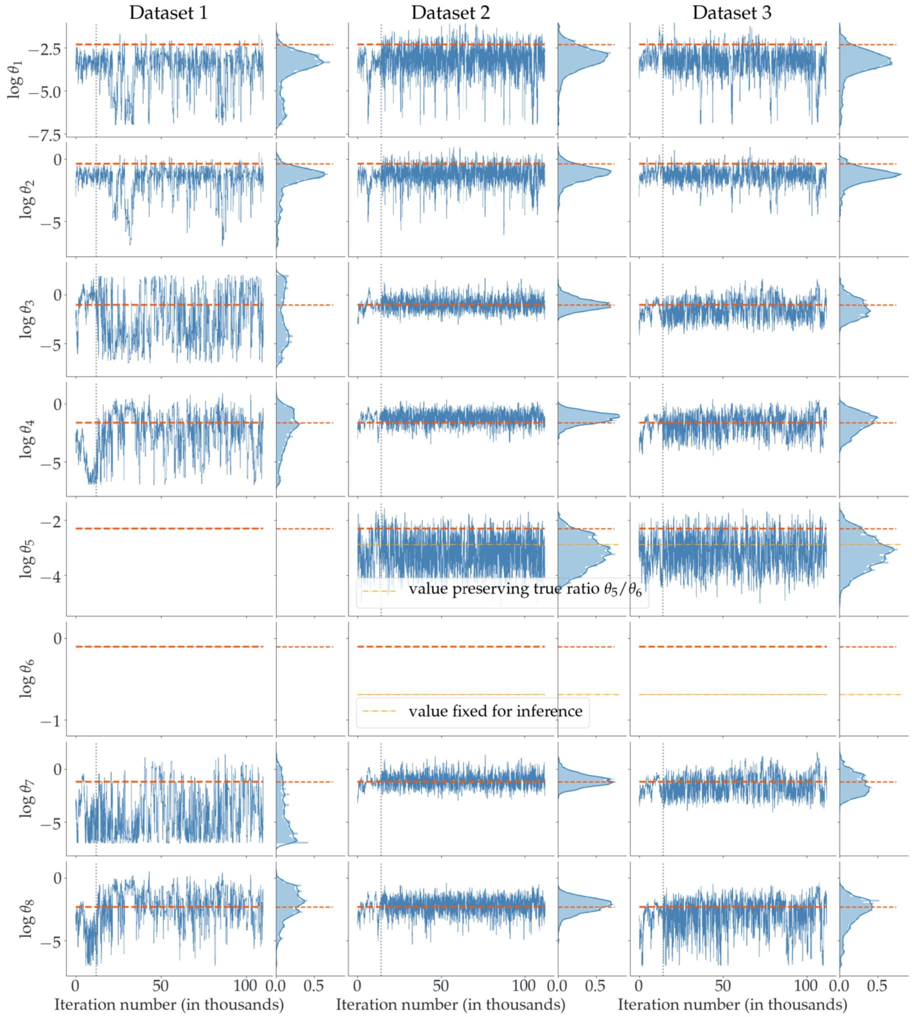

Fig 5: Inference results for the prokaryotic example. Each iteration refers to a single step of parameter update. Left column gives the results of inference run on dataset 1; middle column: results from dataset 2; right column: results from dataset 3

$2 \cdot 10^4$ iterations). The true parameter values are marked with orange, dashed, horizontal lines, whereas grey, dotted, vertical lines mark the change in the type of the transition kernel used (from adaptive, single-site updates to joint updates of [31]).

For the dataset 1 we fixed θ_5, θ_6 and K to their true values and we aimed at inferring the remaining parameters. This is the same setup as in [28]. The results are given in the left column of Figure 5.

Note that the marginal posteriors over $\log(c_3)$ and $\log(c_8)$ presented in Figure 3 in [28] (corresponding to $\log(\theta_3)$ and $\log(\theta_8)$) concentrate decisively around incorrect values of the parameters, possibly pointing to some undiagnosed problems with the methodology. In our case, the true values of the parameters that were used to generate the data lie within the support of the posterior. In particular, the posteriors over $\log(\theta_3)$ and $\log(\theta_7)$ do not depart much from the priors, indicating that the data are not informative enough to identify those two parameters. Indeed, an examination of the reaction equations reveals that θ_3 and θ_7 regulate the production and the degradation of the `RNA` respectively and the observation scheme that provides only noisy counts of `P` might not provide sufficient information.

We generated the second dataset to examine the degree of improvement in identifying the true parameters that could be brought about by adding sparse observations of the `RNA`. We assumed that only K is known and we fixed θ_6 to an *incorrect* but *reasonable* value of $\theta_6^\star = 0.5$. This was done to reproduce a real setting in which none of the θ parameters are known, so fixing any θ_i parameter to its true value is impossible. By fixing θ_6 to an incorrect value and running the inference algorithm we hope to estimate θ_5/θ_6. Ideally, one would want to estimate all 8 parameters simultaneously, but the dataset is not informative enough to conduct such inference. The results are summarised in the middle column of Figure 5. Parameters θ_i, $i \in \{1, 2, 3, 4, 7, 8\}$ have been successfully identified and the posterior over θ_5 concentrates around the value $\theta_5^\star$ for which $\theta_5^\star/\theta_6^\star$ is equal to the true ratio θ_5/θ_6 of the parameters that were used to generate the data.

Naturally, the problem with the observational setting of dataset 2 is that noisy counts of $\texttt{RNA}_t$ need not be easily available in practice. However, it is conceivable that it is possible to collect information about the marginal distribution of `RNA`. To make use of such information, at each time $t \in \{8, 16, \dots, 96\}$ instead of observing a true, noisy count of $\texttt{RNA}_t$ we could instead observe a realisation from the marginal density of `RNA`. We reproduce such observational setting by randomly permuting the time-labels of the generated $\texttt{RNA}_t$, $t \in \{8, 16, \dots, 96\}$, resulting in the dataset 3. The inference results in this setting (assuming K known and θ_6 fixed to an incorrect value of $\theta_6^\star = 0.5$) are summarised in the right column of Figure 5. Some of the marginal posteriors appear flatter than the ones in the middle column, but the loss of information is not substantial and the posteriors still identify the true parameter values.

6.3. Tracking the translation of a dynamic Gaussian random field

We consider the problem of inferring a latent spatial process from noisy low frequency observations. As an example we take a few pictures from a sequence of satellite images depicting the evolution of a convective cloud system taken 15 minutes apart. These are shown in Figure 6. The sequence was analysed by [5] from the perspective of curve tracking.[3] A visual inspection indicates that the

[3] The images were processed to remove annotated curves.

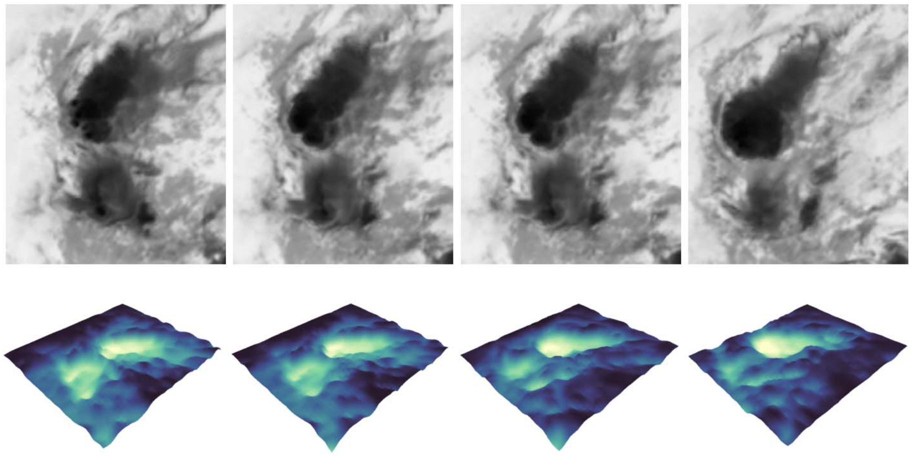

Fig 6: Sequence of infrared Meteosat-MSG2 satellite images depicting the evolution of a convective cloud system. The time in between each image is 15 minutes. Images from [5]. Top: black/white image. Bottom: surface plot.

system moves south–east and we are interested in estimating the speed of the system, as well as its location and extent at intermediate times.

No attempt is made to physically model a convective system. This example is merely meant to serve as a starting point for applications in high dimensional settings. The statistical analysis is made on the minimalistic assumptions that we observe a large scale phenomenon moving at constant speed through space, but the images show additional temporal and spatial high frequency phenomena we consider as random and it is the latent, large scale evolution that we are interested in. In this way we develop a generic model to track deformations and translations in time not bound to this particular meteorological example. Both Wiener and formal observation noise then capture all dynamics, observation errors and local physical phenomena which cannot be explained by our simple model.

We set this numerical experiment up in the way of a 2-d generalisation of (5.2). It is convenient to describe the model as a discretely observed, matrix-valued Langevin process (stochastic heat equation) on the graph with a stationary distribution that shows large scale variation in the distribution of its samples. This is similar in spirit to [33], but we take a more dynamical view with discrete time observations and continuous latent dynamics and a drift term. Furthermore, via forward guiding our approach has a natural non-Gaussian extension. To avoid working with covariance tensors describing the uncertainty of a matrix valued process we move to the equivalent, vector-valued process, obtained through the application of the vec (vectorisation) operation.

The images have a single channel with values in $[0, 1]$, so each pixel corresponds to one entry of a real $m \times n$ matrix, but we do not take the boundedness into account and model them as real-valued matrices. The matrix-valued observations are denoted by $Y_0, \ldots, Y_3$ corresponding to pictures with indices $33, 35, 40, 45$. The chosen images are not equally spaced in time, but this poses no difficulty to our method. The last three pictures are taken 75 minutes apart, the first two pictures are 30 minutes apart. We consider 75 min as 1 time unit. The original image size was 196×166 and it was downsampled to 65×55.

We assume that

$$Y_i = X_{t_i} + \eta_i, \eta_i \sim \mathrm{N}(0, \Sigma),$$

where η is matrix-valued Gaussian noise with covariance tensor Σ of the appropriate dimension (corresponding to the covariance matrix of $\operatorname{vec}(\eta)$). We took $\Sigma = \sigma_\epsilon^2 I$

The latent process is a stochastic process $X = (X_t)$ as well taking values in the set of $m \times n$-real matrices, so nominally a 3575 dimensional diffusion process solving

$$\mathrm{d}X_t = -\tfrac{\sigma^2}{2}(\rho\Lambda + cI)X_t \,\mathrm{d}t + F_\theta(X_t)\,\mathrm{d}t + \sigma\,\mathrm{d}W_t, \tag{6.3}$$

with W_t an uncorrelated, matrix-valued Brownian motion, $\sigma, \rho, c > 0$ and Λ the graph Laplacian tensor defined in (6.4) below.

The pixel indices i, j are identified with the vertices $V = \{(i,j), i \in \{1, \ldots, m\}, j \in \{1, \ldots, n\}\}$ of the $m \times n$-lattice (V, E) with edges $E = \{\{v, v'\}\colon v = (i,j), v' = (i', j') \in V, |i - i'| + |j - j'| = 1\}$ (using the set notation for edges). Thus, edges connect a pixel to its vertical and horizontal neighbours. Then

$$\Lambda_{v,v'} = \begin{cases} \text{degree}(v) & v = v' \\ -1 & \{v, v'\} \in E \\ 0 & \text{otherwise,} \end{cases} \tag{6.4}$$

which generalises (5.3), and F_θ, $\theta = (\theta_1, \theta_2)$ is a lateral translation of θ_1 pixels south and θ_2 pixels east per unit of time, i.e.

$$F_\theta(x) = \theta_1 \rho \nabla^{\downarrow} x + \theta_2 \rho \nabla^{\leftarrow} x,$$

where $\nabla^{\downarrow} = \nabla^{(0,-1)}$ and $\nabla^{\leftarrow} = \nabla^{(-1,0)}$ are given by the mass transport operators

$$\nabla^{(\Delta i, \Delta j)}_{v,v'} = \begin{cases} -1 & v = v' \\ 1 & v = (i,j), v' = (i + \Delta i, j + \Delta j) \\ 0 & \text{otherwise} \end{cases}$$

with $(\Delta i, \Delta j) \in \{(-1,0), (1,0), (0,1), (0,-1)\}$. Thus $b_\theta(t,x) = -\frac{\sigma^2}{2}(\rho\Lambda + cI)x + F_\theta(x)$ with θ considered as unknown parameter is of the form prescribed by Proposition 4.5. Choosing ρ and σ proportional to image scale makes the model roughly scale invariant, which informally can be seen from considering the stationary distribution given by the Lyapunov equation under transformation of the SDE by a linear downsampling operator $R\colon \mathbb{R}^{m \times n} \to \mathbb{R}^{m/2 \times n/2}$, assuming n, m are even. We then chose parameters which worked well at 20×24 pixel resolution and scaled them to other resolutions 41×49 and 55×65 that we use.

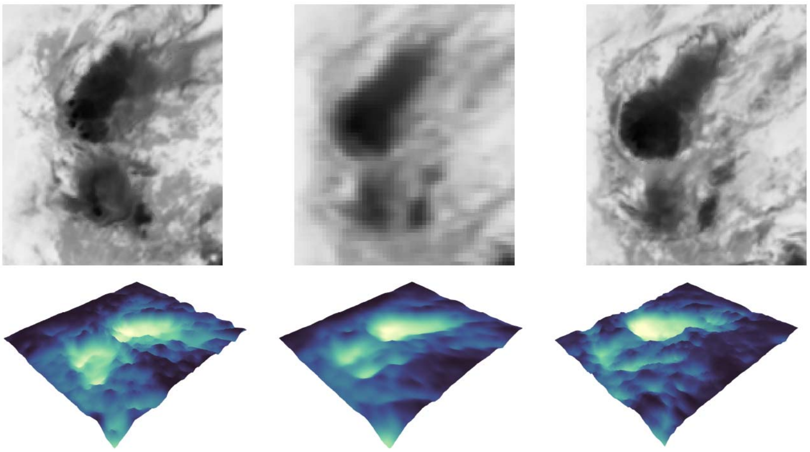

Fig 7: The center panel shows the posterior mean after approximately 20 minutes. The left and right frames show the first and last observation respectively of the four observations in the same resolution. The supplement [43] contains an animation showing that the latent convective system bridge moves smoothly from the initial to the final state. Top: black/white image. Bottom: surface plot.

While the model works for finite dimensional problem at hand, the spatial white observation noise has no infinite dimensional scaling limit and for larger applications one would consider smoother observation noise with spatial correlation given by a non-diagonal covariance structure.

Parameter choices: We set a scale $\rho = 8m/195$ (which amounts to 8/3 for the highest resolution level) and take $c = 0.1\rho$, We set $\sigma = 0.08\rho$. The noise level is set to $\sigma_\epsilon = 0.05\sqrt{\rho}$. We took the time step as $\delta t = 1/(35\rho)$. We assume that X_T is a priori Gaussian with mean 0.5 (a grey value) and variance $2.5I$ and equip θ with an improper uniform prior.

Algorithm details: The SDE in our model is linear, so with $\tilde{X} = X$, the proposal is a sample from the conditional distribution and no change of measure is needed. We solve the backward filtering equations from Theorem 2.6 for ν and P using the Euler scheme in combination with the enforced sparsity method from Section 5.2 using Julia's native sparse matrix type (CSC). We took $\varepsilon = 10^{-8}$.

We employed Rao-Blackwellisation to obtain an estimate of the mean of the joint posterior of parameters and latent path, where in one step, the mean of $X^\star$ given observations and θ is given by

$$\mathrm{d}x_t^\star = b_\theta(t, x_t^\star)\,\mathrm{d}t + \sigma^2 P(t)^{-1}(x_t^\star - \nu(t))\,\mathrm{d}t, \quad x_0^\star = \nu(0)$$

and in the second step the mean of θ given $X^\star$ is obtained using Proposition 4.5.

For the correction formulas (2.13) we opted for an implementation based on the sparse Cholesky factorisation provided in Julia.

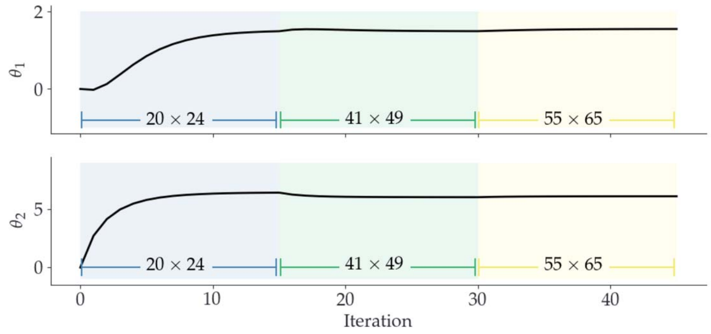

Fig 8: Iterates of θ_1, θ_2 starting in $\theta_1 = \theta_2 = 0$ from Rao-Blackwellisation with increasing dimension. First 15 iterates (blue): running with resolution 20 × 24. Next 15 iterates (green): continuing with resolution 41 × 49. Last 15 iterates (yellow): full resolution 55 × 65.

Results: The first 15 steps of the algorithm were run on a model downscaled to lower resolution, for the subsequent 15 steps the resolution was increased and the final 15 steps were run on a model with the chosen, target resolution. Figure 8 shows the trace of θ_1, and θ_2 for 45 iterates of the Rao-Blackwellised chain. After the last 15 steps the final estimates of the posterior means of the parameters are obtained as $\hat{\theta}_1 = 1.55$ and $\hat{\theta}_2 = 6.12$. These are in good agreement with visual assessment of the movement. The estimates of θ appear mostly stable at different model scales. (See the small changes in the estimates after each change of resolution at steps 15 and 30 in Figure 8.) The animation in supplement [43] shows the estimated posterior mean trajectory. The incorporation of discrete time observations with continuous latent dynamics and a drift term allows the algorithm to recognise the convective system as one object even if it has moved a substantial amount of pixels.

We only report rough timings: backward filtering step took half a minute for the correction steps and about two minutes for propagating the uncertainty at full resolution. Forward guiding took about half a minute and the parameter update about 1 second. (Timings are variable and depend on the current value of θ, as θ influences the sparsity.) The Kalman gain had 3 % non-zero entries with $\epsilon = 10^{-8}$.

Appendix A: Related work

If the full state of the diffusion process is observed at all times $\{t_i,\, 0 \le i \le n\}$ without noise, i.e. if $L_i = I$ and $\Sigma_i \equiv 0$ for all i in (1.3), then the smoothing problem reduces to the sampling of n independent diffusion bridges. This problem has attracted considerable attention over the past two decades, see for instance [22], [21], [20], [15], [13], [9], [32], [6], [39]), [8], [19], [40], [53], [62] and [12]. In the general case however, the connecting bridges cannot be sampled independently between adjacent observation times. In fact, at time $t \in (t_{i-1}, t_i)$ the process X, conditioned on $\mathcal{V}_0$, depends on all future conditionings $V_i, \dots, V_n$. To resolve this problem, subsequent simulation of bridges on overlapping intervals has been proposed by [27], [25] and [55].

[51] considered *filtering* (instead of smoothing) of diffusions under the assumption that the dispersion coefficient is only allowed to depend on time. If the diffusion can be transformed to unit diffusion coefficient, then filtering can also be accomplished using the exact algorithm for simulation of diffusions, as introduced by [10]. This algorithm forms the basis for the methods presented in [24] and [45]. Various solutions to the filtering and smoothing problem are further discussed in [50]. Key to the proposed algorithms therein is the assumption that the distribution of X_t, conditional on the data $\mathcal{V}_0$ can be approximated by the normal distribution.

Recently, [29] considered the same problem as considered here using a sampler based on constrained Hamiltonian dynamics. The basic idea is to consider the solution to the SDE as a forward map of Wiener-innovations that are constrained to a manifold implied by the observations. Our approach is rather different and based on forward simulating paths that directly take all observations into account.

Appendix B: Technical background on guided proposals

Previous work on guided proposals includes:

(A) In [53] the basic idea was introduced in the setting of uniformly elliptic diffusions (i.e. the case where for each $y \in \mathbb{R}^d$ there exists an $\varepsilon > 0$ such that $\|\sigma(t,x)'y\| \geq \varepsilon\|y\|^2$), with one segment and a "full" observation at the time of conditioning. More precisely, the aim is to simulate a diffusion, conditioned on $X_0 = x_0$ and $X_T = x_T$, with $0 < T$.

(B) In [55] this was extended to the setting of two future conditionings. More precisely, in this paper it was shown how to define guided proposals to sample from a diffusion starting in $X_0 = x_0$ and conditioned on $v_S \sim \mathrm{N}(LX_S, \Sigma_S)$ and $X_T = x_T$, where $0 < S < T$.

(C) In [12] the work on [53] was generalised in a different direction by incorporating hypo-elliptic diffusions, observed partially and without noise. More precisely, here it is assumed that $X_0 = x_0$ and the conditioning is given by $LX_T = v_T$.

In all cases, the SDE for the conditioned process is of the form (1.4). The function ρ is specific to the type of conditioning and depends on the unknown transition densities p of the diffusion X.

The results of Theorem 1 in [53] and Theorem 2.14 in [12] (and thus, the absolute continuity of the laws $\mathbb{P}^\star$ and $\mathbb{P}^\circ$ and the expression (1.9)) are proven under the following conditions: *(i)* boundedness and smoothness of the drift coefficient b and dispersion coefficient σ; *(ii)* "matching conditions on the diffusivity (and possibly drift)". In setting (A), the matching condition is given by $\tilde{a}(T) =$

$a(T, x_T)$ which can always be ensured. In setting (C) the matching conditions are more subtle and more complicated. The results in [12] strongly suggest that for the diffusivity of the auxiliary process the condition $La(T, x_T)L' = L\tilde{a}(T)L'$ suffices. In case of hypo-ellipticity, additionally the drift of the auxiliary process, $\tilde{b}(t,x) = B(t)x + \beta(t)$, needs to satisfy the equation $Lb(T, x_T) = L\tilde{b}(T, x_T)$. These conditions are particularly important in the noiseless limit of the discrete time (partial) observations.

Appendix C: Derivation of the guiding term in $X^\star$

Here we explain the type of conditioning induced by the guiding term in $X^\star$. In particular, the specific form of ρ in (1.5).

Let p denote the transition density of a diffusion process Y solving an SDE with drift b and diffusivity σ on the (canonical) filtered probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}, \mathbb{P})$ with $\Omega = C([0, t_n], \mathbb{R}^d)$. Assume the hierarchical model

$$\begin{aligned} v_i \mid \xi_i &\overset{\text{ind}}{\sim} k_i(\xi_i; v_i) \qquad 1 \le i \le n \\ \xi_1, \dots, \xi_n &\sim \prod_{i=1}^n p(t_{i-1}, \xi_{i-1}; t_i, \xi_i) \end{aligned} \tag{C.1}$$

with ξ_0 assumed fixed (known). Without loss of generality we assume $t \in [0, t_1)$. We will assume $\xi_j \in \mathbb{R}^d$, $L_j \in \mathbb{R}^{m_j \times d}$ so that $v_j \in \mathbb{R}^{m_j}$. Define $\boldsymbol{\xi} = (\xi_1, \dots, \xi_n)$ and $\boldsymbol{v} = (v_1, \dots, v_n)$. Let

$$\pi_{t,y}(\boldsymbol{\xi}) = p(t, y; t_1, \xi_1) \prod_{j=1}^n p(t_j, \xi_j; t_{j+1}, \xi_{j+1})$$

and denote the density of $(\boldsymbol{\xi}, \boldsymbol{v})$, conditional on $Y_t = y$, by

$$\zeta_{t,y}(\boldsymbol{\xi}, \boldsymbol{v}) = \pi_{t,y}(\boldsymbol{\xi}) \prod_{j=1}^n k_j(\xi_j, v_j).$$

We apply Doob's h-transform with

$$h(t, y) = \frac{\int \zeta_{t,y}(\boldsymbol{\xi}, \boldsymbol{v}) \,\mathrm{d}\boldsymbol{\xi}}{\int \zeta_{0,\xi_0}(\boldsymbol{\xi}, \boldsymbol{v}) \,\mathrm{d}\boldsymbol{\xi}}.$$

Set $Z_t = h(t, Y_t)$. It is easily verified that for any $t \in [0, t_1)$, Z_t is a martingale with mean 1. Define a new probability measure $\mathbb{Q}$ on Ω by the change of measure $\mathrm{d}\mathbb{Q}|_{\mathcal{F}_t} = Z_t \,\mathrm{d}\mathbb{P}|_{\mathcal{F}_t}$, $\forall t \in [0, t_1)$. By Girsanov's theorem it follows that under $\mathbb{Q}$

$$\mathrm{d}Y_t = b(t, Y_t)\,\mathrm{d}t + \sigma(t, Y_t)\,\mathrm{d}\tilde{W}_t + a(t, Y_t)'\nabla \log h(t, Y_t)\,\mathrm{d}t,$$

where $\tilde{W}$ is Brownian motion under $\mathbb{Q}$. We have

$$\mathbb{E}_{\mathbb{Q}} f(Y_t) = \mathbb{E}_{\mathbb{P}}[Z_t f(Y_t)] = \mathbb{E}_{\mathbb{P}}[h(t, Y_t) f(Y_t)] = \int f(y) p(0, \xi_0; t, y) h(t, y) \,\mathrm{d}y$$

$$= \int f(y)p(0,\xi_0;t,y)\frac{\int \zeta_{t,y}(\boldsymbol{\xi},\boldsymbol{v})\,\mathrm{d}\boldsymbol{\xi}}{\int \zeta_{0,\xi_0}(\boldsymbol{\xi},\boldsymbol{v})\,\mathrm{d}\boldsymbol{\xi}}\,\mathrm{d}y$$
$$= \int \left(\int f(y)\frac{p(0,\xi_0;t,y)\zeta_{(t,y)}(\boldsymbol{\xi},\boldsymbol{v})}{\zeta_{(0,\xi_0)}(\boldsymbol{\xi},\boldsymbol{v})}\,\mathrm{d}y \right) \psi(\boldsymbol{\xi} \mid \boldsymbol{v})\,\mathrm{d}\boldsymbol{\xi}$$

where we multiplied by $1 = \zeta_{(0,\xi_0)}(\boldsymbol{\xi},\boldsymbol{v})/\zeta_{(0,\xi_0)}(\boldsymbol{\xi},\boldsymbol{v})$, used Fubini to arrive at the final equality and ψ is defined by

$$\psi(\boldsymbol{\xi} \mid \boldsymbol{v}) = \frac{\zeta_{0,\xi_0}(\boldsymbol{\xi},\boldsymbol{v})}{\int \zeta_{0,\xi_0}(\boldsymbol{\xi},\boldsymbol{v})\,\mathrm{d}\boldsymbol{\xi}}.$$

Therefore, we have

$$\mathbb{E}_{\mathbb{Q}} f(Y_t) = \int \mathbb{E}_{\mathbb{P}}[f(Y_t) \mid Y_{t_1} = \xi_1, \ldots, Y_{t_n} = \xi_n]\,\psi(\boldsymbol{\xi} \mid \boldsymbol{v})\,\mathrm{d}\boldsymbol{\xi}.$$

This shows that under $\mathbb{Q}$, first $\boldsymbol{\xi}$ is sampled from the density ψ, and subsequently the process Y is conditioned on the event $\{Y_{t_1} = \xi_1, \ldots, Y_{t_n} = \xi_n\}$. Note that sampling from ψ corresponds to sampling from the posterior of $\boldsymbol{\xi}$ in the statistical model specified by (C.1), with data $\boldsymbol{v}$. In this model, π_{0,ξ_0} can be viewed as the prior density of $\boldsymbol{\xi}$.

To connect to the main text, simply note that Y under the measure $\mathbb{P}$ is X, whereas Y under the measure $\mathbb{Q}$ is $X^\star$ and that the h transform is in our case ρ.

Appendix D: Proofs of theorems in Section 2 and additional remarks

In the proofs we will assume observation times $0 < S < T$ and observations $V_S \sim \mathrm{N}(L_S X_S, \Sigma_S)$ and $V_T \sim \mathrm{N}(L_T X_T, \Sigma_T)$ with $L_S \in \mathbb{R}^{m_S \times d}$ and $L_T \in \mathbb{R}^{m_T \times d}$; the extension to multiple future conditionings being trivial.

D.1. *Proof of Theorem 2.4*

Define Φ to be the solution to $\mathrm{d}\Phi(t) = B(t)\Phi(t)\,\mathrm{d}t$ with $\Phi(0) = I$. Set $\Phi(t,s) = \Phi(t)\Phi(s)^{-1}$. Define

$$\Upsilon(t) = \begin{cases} \begin{bmatrix} \Sigma_S & 0_{m_S \times m_T} \\ 0_{m_T \times m_S} & \Sigma_T \end{bmatrix} & t \in (0,S] \\ \Sigma_T & t \in (S,T] \end{cases},$$

and notice (upon differentiation) that

$$L(t) = \begin{cases} \begin{bmatrix} L_S \Phi(S,t) \mathbf{1}_{(0,S]}(t) \\ L_T \Phi(T,t) \end{bmatrix} & t \in (0,S] \\ L_T \Phi(T,t) & t \in (S,T] \end{cases}, \tag{D.1}$$

$$M^{\dagger}(t) = \int_t^T L(\tau)\tilde{a}(\tau)L(\tau)' \,\mathrm{d}\tau + \Upsilon(t), \tag{D.2}$$

and

$$\mu(t) = \int_t^T L(\tau)\beta(\tau)\,\mathrm{d}\tau, \qquad t \in [0,T], \tag{D.3}$$

are the solutions to ODEs (2.2), (2.3) and (2.4) respectively. Note that $\mu(t) \in \mathbb{R}^{m(t)}$ and that $\Upsilon(t)$, $L(t)$, $M^{\dagger}(t)$ and $v(t)$ have similar structures when t is either in $(0,S]$ or $(S,T]$. The relations in (2.5) can be verified by evaluating $L(t)$ for both $t = S$ and $t = S+$ (and similarly for $M^{\dagger}(t)$ and $\mu(t)$).

As we assume Σ_T to be strictly positive definite, matrix $M(t) = M^{\dagger}(t)^{-1}$ exists for $t \in [0,T]$ (see however Remark D.2 in case of no noise on the observations).

The expression for $\tilde{r}$ for $t \in (0,S]$ follows by extending Lemma 2.5 in [55] to the case where not necessarily $\Sigma_T = 0$ and $L_T = I$. The expressions for $t \in (S,T]$ follow from equation (4.1) in [55].

To derive $\tilde{\rho}$, note by Section 5 in [53] that $\tilde{X}$ is a Gaussian process with conditional mean $\mu_t(s,x) = \mathbb{E}[\tilde{X}_t \mid \tilde{X}_s = x]$ and covariance $K_t(s) = \mathrm{Cov}(\tilde{X}_s, \tilde{X}_t)$, where:

$$\mu_t(s,x) = \Phi(t,s)x + \int_s^t \Phi(s,\tau)\beta(\tau)\,\mathrm{d}\tau, \qquad K_t(s) = \int_s^t \Phi(t,\tau)\tilde{a}(\tau)\Phi(t,\tau)'\,\mathrm{d}\tau. \tag{D.4}$$

Then, clearly, $(V_S', V_T')' = ((L_S\tilde{X}_S + \eta_S)', (L_T\tilde{X}_T + \eta_T)')'$ is Gaussian as well and its mean (denoted with $\bar{v}(t)$) and covariance (denoted with $\bar{\Omega}(t)$) are given by:

$$\bar{v}(t) = \begin{pmatrix} L_S\mu_S(t,x) \\ L_T\mu_T(t,x) \end{pmatrix}, \quad \bar{\Omega}(t) = \begin{pmatrix} L_SK_{SS}L_S' + \Sigma_S & L_SK_{ST}L_T' \\ L_TK_{TS}L_S' & L_TK_{TT}L_T' + \Sigma_T \end{pmatrix},$$

where $K_{\tau\nu} = K_{\tau\wedge\nu}(t)$, $\tau, \nu \in [t,T]$. Careful comparison of $\bar{v}(t)$ and $\bar{\Omega}(t)$ with the definitions (D.1), (D.2) and (D.3) reveals that

$$\bar{v}(t) = L(t)x + \mu(t), \qquad \bar{\Omega}(t) = M^{\dagger}(t),$$

and the expression for $\tilde{\rho}$ follows.

Remark D.1. The value of Theorem 2.4 lies in recognition of the structure on both H and $\tilde{r}$, something which was not noticed in [55]. The theorem shows that both quantities can be written in a unified way on both $(0,S]$ and $(S,T]$. The key to this is the proper definition of L (including the indicator).

Remark D.2. Suppose $L_T = I$ and $\Sigma_T \equiv 0$. For $t \in (S,T]$, $M(t)$ exists if and only if $\int_t^T \Phi(T,t)\tilde{a}(\tau)\Phi(T,t)'\,\mathrm{d}\tau$ is invertible. This matrix is the controllability Grammian. Systems theory provides sufficient conditions for controllability. In case $\tilde{a}(t)$ is not invertible, then $M(t)$ exists if and only if the pair of functions $(B,\tilde{\sigma})$ is controllable on $[t,T]$ for any $t \in [0,T)$ (cf. Section 5.6 in [36]).

Remark D.3. Suppose $L_S = I$ and $\Sigma_S = 0$. This corresponds to the case where the diffusion is fully observed at time S without noise. By the Markov property, the pulling term should only depend on v_S (which is then in fact x_S) and not on v_T. To verify this, first note that we can write

$$M^\dagger(t) = \begin{bmatrix} A & C \\ C' & D \end{bmatrix},$$

where for $t \in [0, S)$

$$\begin{aligned} A &= \int_t^S L_S \Phi(S, \tau) \tilde{a}(\tau) \Phi(S, \tau)' \,\mathrm{d}\tau, \\ C &= \int_t^S L_S \Phi(S, \tau) \tilde{a}(\tau) \Phi(T, \tau)' L_T' \,\mathrm{d}\tau = A \Phi(T, S)' L_T', \\ D &= \int_t^T L_T \Phi(T, \tau) \tilde{a}(\tau) \Phi(T, \tau)' L_T' \,\mathrm{d}\tau. \end{aligned}$$

Now assume $L_S = I$ and that A is invertible. Then we have

$$L(t) = \begin{bmatrix} L_S \\ L_T \Phi(T, S) \end{bmatrix} \Phi(S, t) = \begin{bmatrix} I \\ C' A^{-1} \end{bmatrix} \Phi(S, t).$$

Let $Z = (D - C' A^{-1} C)^{-1}$. Using the formula for the inverse of a block matrix, we get

$$\begin{aligned} H(t) &= \Phi(S, t)' \begin{bmatrix} I & A^{-1} C \end{bmatrix} \begin{bmatrix} A^{-1} + A^{-1} C Z C' A^{-1} & -A^{-1} C Z \\ -Z C' A^{-1} & Z \end{bmatrix} \begin{bmatrix} I \\ C' A^{-1} \end{bmatrix} \Phi(S, t) \\ &= \Phi(S, t)' A^{-1} \Phi(S, t) = \left(\int_t^S \Phi(t, \tau) \tilde{a}(\tau) \Phi(t, \tau)' \,\mathrm{d}\tau \right)^{-1}. \end{aligned}$$

This is exactly as in Lemma 6 of [53].

Remark D.4. Suppose $t \in (t_{i-1}, t_i]$ and we condition on future incomplete observations $(v_i, \ldots, v_n)$. In that setting the correct definitions for Υ and $L(t)$ are

$$\Upsilon(t) = \operatorname{diag}(\Sigma_i, \ldots, \Sigma_n) \qquad \text{and} \qquad L(t) = \begin{bmatrix} L_i \Phi(t_i, t) \\ L_{i+1} \Phi(t_{i+1}, t) \\ \cdots \\ L_n \Phi(t_n, t) \end{bmatrix}$$

.

D.2. *Proof of Theorem 2.5*

The expression for $\log \tilde{\rho}$ follows from Theorem 2.4 by simple algebra, with

$$c(t) = \frac{1}{2} C_t^{(1)} + C_t^{(2)}, \quad \text{where}$$

$$C_t^{(1)} = (v(t) - \mu(t))' M(t) (v(t) - \mu(t)) \quad \text{and} \quad C_t^{(2)} = \log \left[(2\pi)^{m(t)/2} \left| M^\dagger(t) \right|^{1/2} \right]. \tag{D.5}$$

Backward ordinary differential equations We derive the ordinary differential equations solved by $H(t)$, $F(t)$ and $c(t)$ in a neighbourgood disjoint from $\{S,T\}$. We start with $H(t)$.

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}H(t) &= \left(\frac{\mathrm{d}}{\mathrm{d}t}L(t)\right)' M(t)L(t) + L(t)'\left(\frac{\mathrm{d}}{\mathrm{d}t}M(t)\right)L(t) + L(t)'M(t)\left(\frac{\mathrm{d}}{\mathrm{d}t}L(t)\right)\\ &= -B(t)'L(t)'M(t)L(t) + L(t)'\left(\frac{\mathrm{d}}{\mathrm{d}t}M(t)\right)L(t) - L(t)'M(t)L(t)B(t)\\ &\quad - B(t)'H(t) - H(t)B(t) + L(t)'\left(\frac{\mathrm{d}}{\mathrm{d}t}M(t)\right)L(t).\end{aligned} \tag{D.6}$$

Since $M(t) = (M^\dagger(t))^{-1}$:

$$\frac{\mathrm{d}}{\mathrm{d}t}M(t) = -M(t)\left(\frac{\mathrm{d}}{\mathrm{d}t}M^\dagger(t)\right)M(t) = M(t)L(t)\tilde{a}(t)L(t)'M(t), \tag{D.7}$$

and thus:

$$L(t)'\left(\frac{\mathrm{d}}{\mathrm{d}t}M(t)\right)L(t) = H(t)\tilde{a}(t)H(t).$$

Substituting this back into (D.6) yields:

$$\mathrm{d}H(t) = \left(-B(t)'H(t) - H(t)B(t) + H(t)\tilde{a}(t)H(t)\right)\,\mathrm{d}t.$$

For $F(t)$ notice:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}F(t) &= \left(\frac{\mathrm{d}}{\mathrm{d}t}L(t)\right)' M(t)(v(t)-\mu(t)) + L(t)'\left(\frac{\mathrm{d}}{\mathrm{d}t}M(t)\right)(v(t)-\mu(t))\\ &\quad - L(t)'M(t)\left(\frac{\mathrm{d}}{\mathrm{d}t}\mu(t)\right)\\ &= -B(t)'L(t)'M(t)\left(v(t)-\mu(t)\right)\\ &\quad + L(t)'M(t)L(t)\tilde{a}(t)L(t)'M(t)(v(t)-\mu(t)) + L(t)'M(t)L(t)\beta(t)\\ &= -B(t)'F(t) + H(t)\tilde{a}(t)F(t) + H(t)\beta(t).\end{aligned}$$

For $c(t)$ we differentiate $C_t^{(1)}$ and $C_t^{(2)}$ in turn. By (2.4), (D.7) and the definition of $F(t)$:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}C_t^{(1)} &= -2\left(\frac{\mathrm{d}}{\mathrm{d}t}\mu(t)\right)' M(t)(v(t)-\mu(t))\\ &\qquad + (v(t)-\mu(t))'\left(\frac{\mathrm{d}}{\mathrm{d}t}M(t)\right)(v(t)-\mu(t))\\ &= 2\beta(t)'L(t)'M(t)(v(t)-\mu(t))\\ &\qquad + (v(t)-\mu(t))'M(t)L(t)\tilde{a}(t)L(t)'M(t)(v(t)-\mu(t))\\ &= 2\beta(t)'F(t) + F(t)'\tilde{a}(t)F(t).\end{aligned} \tag{D.8}$$

To find $\frac{\mathrm{d}}{\mathrm{d}t}C^{(2)}_t$, first, note by eq. (10) in [44] that for a matrix-valued function Y:

$$\frac{\partial |Y|}{\partial x} = |Y| \mathrm{tr}\left(Y^{-1}\frac{\partial Y}{\partial x}\right),$$

which gives

$$\begin{aligned}\frac{\mathrm{d}|M^{\dagger}(t)|}{\mathrm{d}t} &= |M^{\dagger}(t)|\mathrm{tr}\left(M(t)\frac{\mathrm{d}M^{\dagger}(t)}{\mathrm{d}t}\right)\\ &= |M^{\dagger}(t)|\mathrm{tr}\left(-M(t)L(t)\tilde{a}(t)L(t)'\right)\\ &= -|M^{\dagger}(t)|\mathrm{tr}\left(L(t)'M(t)L(t)\tilde{a}(t)\right) = -|M^{\dagger}(t)|\mathrm{tr}\left(H(t)\tilde{a}(t)\right).\end{aligned}$$

Since at non-observation times t, $m(t)$ is constant, the chain rule implies

$$\frac{\mathrm{d}}{\mathrm{d}t}C^{(2)}_t = -\frac{1}{2}\mathrm{tr}(H(t)\tilde{a}(t)). \tag{D.9}$$

Combining (D.8) and (D.9) yields the ODE for $c(t)$ stated in Theorem 2.5.

Update equations The boundary conditions for $H(T)$, $F(T)$, $C^{(1)}_T$ and $C^{(2)}_T$:

$$H(T) = L_T'\Sigma_T^{-1}L_T, \quad F(T) = L_T'\Sigma_T^{-1}v_T,$$

$$C^{(1)}_T = v_T\Sigma_T^{-1}v_T, \quad C^{(2)}_T = \frac{m_T}{2}\log(2\pi) + \frac{1}{2}\log|\Sigma_T|,$$

follow directly from the definitions: $L(T) = L_T$, $M(T) = \Sigma_T^{-1}$ and $\mu(T) = 0$; as well as the expressions for $H(t)$, $F(t)$, $C^{(1)}_t$ and $C^{(2)}_t$ given in (2.6), (2.7) and (D.5) respectively. The boundary condition for $c(T)$ follows immediately

$$c(T) = \frac{1}{2}C^{(1)}_T + C^{(2)}_T = \frac{1}{2}v_T\Sigma_T^{-1}v_T + \frac{m_T}{2}\log(2\pi) + \frac{1}{2}\log|\Sigma_T|.$$

Now, combining the expressions for $H(t)$, $F(t)$ and $C^{(1)}_t$, given in (2.6), (2.7) and (D.5) respectively, with the update equations for $L(S)$, $M(S)$ and $\mu(S)$ at time S, given in Theorem 2.4, yields the update equations for $H(S)$, $F(S)$ and $C^{(1)}_S$:

$$\begin{aligned} H(S) &= H(S+) + L_S'\Sigma_S^{-1}L_S\\ F(S) &= F(S+) + L_S'\Sigma_S^{-1}v_S\\ C^{(1)}_S &= C^{(1)}_{S+} + v_S\Sigma_S^{-1}v_S.\end{aligned} \tag{D.10}$$

To derive the update equations for $C^{(2)}_S$, notice that at observation time S we have

$$\begin{aligned}\tilde{\rho}(S,x) &= \varphi(v_S; L_Sx, \Sigma_S)\tilde{\rho}(S+,x)\\ &= (2\pi)^{-m_S/2}|\Sigma_S|^{-1/2}\exp\left(-\frac{1}{2}(v_S - L_Sx)'\Sigma_S^{-1}(v_S - L_Sx)\right)\times\exp\left(-C^{(2)}_{S+}\right)\end{aligned}$$

$$\times \exp\left(-\frac{1}{2}(v(S+)-\mu(S+)-L(S+)x)'M(S+)(v(S+)-\mu(S+)-L(S+)x)\right).$$

This can be written as

$$\exp\left(-C_S^{(2)}\right)\exp\left(-\frac{1}{2}(v(S)-\mu(S)-L(S)x)'M(S)(v(S)-\mu(S)-L(S)x)\right),$$

upon defining

$$C_S^{(2)} = C_{S+}^{(2)} + \frac{m_S}{2}\log(2\pi) + \frac{1}{2}\log|\Sigma_S| \tag{D.11}$$

and because we have

$$v(S)=\begin{bmatrix} v_S \\ v(S+)\end{bmatrix} \quad \mu(S)=\begin{bmatrix} 0 \\ \mu(S+)\end{bmatrix} \quad L(S)=\begin{bmatrix} L_S \\ L(S+)\end{bmatrix} \quad M(S)=\begin{bmatrix} \Sigma_S^{-1} & 0 \\ 0 & M(S+)\end{bmatrix}.$$

Cobining (D.10) and (D.11) yields the update equation for $c(S)$

$$c(S) = \frac{1}{2}C_S^{(1)} + C_S^{(2)} = c(S+) + \frac{1}{2}v_S\Sigma_S^{-1}v_S + \frac{m_S}{2}\log(2\pi) + \frac{1}{2}\log|\Sigma_S|.$$

D.3. Proof of Theorem 2.6

We first give the derivation for the ODE of $P(t)$. The derivation of the differential equation is the same whether we consider $t \in [0,S]$ or $t \in (S,T]$. In particular, the ODE for H from Theorem 2.5 and the definition of P imply that

$$\frac{\mathrm{d}P(t)}{\mathrm{d}t} = -P(t)\left(\frac{\mathrm{d}}{\mathrm{d}t}H(t)\right)P(t) = P(t)B(t)' + B(t)P(t) - \tilde{a}(t).$$

The update formula from time $S+$ to S follows from the corresponding one in terms of H and applying Woodbury's formula.

The derivation of the differential equation for ν is the same whether we consider $t \in [0,S]$ or $t \in (S,T]$. We have, using (D.12), (D.7) and (2.2)

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(P(t)L(t)'M(t)\right) = B(t)P(t)L(t)'M(t). \tag{D.12}$$

Using (2.4) we get

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(v(t)-\mu(t)\right) = L(t)\beta(t).$$

The previous two equations together yield

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\nu(t) &= B(t)P(t)L(t)'M(t)\left(v(t)-\mu(t)\right) + P(t)L(t)'M(t)L(t)\beta(t)\\ &= B(t)\nu(t) + \beta(t).\end{aligned}$$

The value of $\nu(T)$ follows from $P(T) = (L_T'\Sigma_T^{-1}L_T)^{-1}$, $L(T) = L_T$, $M(T) = \Sigma_T^{-1}$, $v(T) = v_T$ and $\mu(T) = 0$.

To obtain the expression for $\nu(S)$, note that

$$\begin{aligned}
\nu(S) &= P(S)L'_S M(S)\left(v(S)-\mu(S)\right)\\
&= P(S)\begin{bmatrix} L_S \\ L(S+)\end{bmatrix}\begin{bmatrix}\Sigma_S & 0\\ 0 & M(S+)\end{bmatrix}^{-1}\left(\begin{bmatrix} v_S \\ v(S+)\end{bmatrix} - \begin{bmatrix} 0 \\ \int_s^T L_T\Phi(T,\tau)\beta(\tau)\,\mathrm{d}\tau\end{bmatrix}\right)\\
&= P(S)\left(L'_S\Sigma_S^{-1}v_S + L(S+)M(S+)\left(v(S+)-\mu(S+)\right)\right)\\
&= P(S)\left(L'_S\Sigma_S^{-1}v_S + H(S+)\nu(S+)\right).
\end{aligned}$$

Proof of Proposition 2.7.

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}P(t) &= \frac{\mathrm{d}}{\mathrm{d}t}\left(\Phi(t,T)(\Sigma+P(T))\Phi(t,T)-\Sigma\right)\\
&= \left(\frac{\mathrm{d}}{\mathrm{d}t}\Phi(t,T)\right)(\Sigma+P(T))\Phi(t,T) + \Phi(t,T)(\Sigma+P(T))\frac{\mathrm{d}}{\mathrm{d}t}\Phi(t,T)'\\
&= B\Phi(t,T)(\Sigma+P(T))\Phi(t,T)' + \Phi(t,T)(\Sigma+P(T))\Phi(t,T)'B'\\
&= B(P(t)+\Sigma)+(P(t)+\Sigma)B' = BP(t)+P(t)B'-\tilde{a},
\end{aligned}$$

where we used (2.15) at the third equality and Σ solving the Lyapunow equation at the final equality. □

Remark D.5. We investigate the behaviour of $P(S)$ and $\nu(S)$ when the noise level tends to zero. *Assume* $L_S P(S+)L'_S$ *is invertible.* Then it follows from (2.13) that the expression for $P(S)$ is also well defined when $\|\Sigma\| \to 0$. Moreover, when $\Sigma = 0$ we have

$$L_S P(S) = L_S P(S+) - L_S P(S+)L'_S\left(L_S P(S+)L'_S\right)^{-1}L_S P(S+) = 0.$$

To evaluate the limiting behaviour of $\nu(S)$, we write

$$L_S\nu(S) = L_S P(S)L'_S\Sigma_S^{-1}v_S + L_S P(S)H(S+)\nu(S+).$$

The second term on the right-hand-side is easily seen to tend to zero when $\|\Sigma_S\| \to 0$. For deriving the limit of the first term on the right-hand-side we define $C = L_S P(S+)L'_S\Sigma_S^{-1}$ and rewrite

$$\begin{aligned}
L_S P(S)L'_S\Sigma_S^{-1} &= L_S P(S+)L'_S\Sigma_S^{-1}\\
&\quad - L_S P(S+)L'_S\left(\Sigma_S + L_S P(S+)L'_S\Sigma_S^{-1}\Sigma_S\right)^{-1}L_S P(S+)L'_S\Sigma_S^{-1}\\
&= C - C(I+C)^{-1}C = \left(I+C^{-1}\right)^{-1},
\end{aligned}$$

where we used Woodbury's formula at the final equality. Now

$$C^{-1} = \Sigma_S\left(L_S P(S+)L'_S\right)^{-1} \to 0, \quad \text{when} \quad \|\Sigma_S\| \to 0.$$

This implies that $L_S P(S)L'_S\Sigma_S^{-1} \to I$. Combining these results we obtain that

$$L_S\nu(S) \to v_S, \quad \text{when} \quad \|\Sigma_S\| \to 0.$$

In particular, if we have full observations at time S, i.e. $L_S = I$, then $P(S) \to 0$ and $\nu(S) \to v_S$ if $\|\Sigma_S\| \to 0$. As a consequence, in case of full observations and no noise, we recover the full observation without noise case (in this simpler setting, the differential equations for ν and P were derived in [57]). However, in the general case where L_S is not of full rank, we need that $\det \Sigma_S \neq 0$ to obtain the value of $\nu(S)$ from $\nu(S+)$.

Appendix E: Implementation using a time-change and scaling

If the noise level on the observation is small, care is required in the discretisation of guided proposals near the conditioning points. For this reason, a time-change and scaling was introduced in Section 5 of [56]. Here, we explain it for the case of 2 future conditionings. Define the mapping $\tau : [0, S+T] \to [0, S+T]$ by

$$\tau(s) = \begin{cases} s\left(2 - \frac{s}{S}\right) & \text{if} \quad s \in [0, S] \\ S + (s - S)\left(2 - \frac{s-S}{T-S}\right) & \text{if} \quad s \in [S, T] \end{cases}.$$

For a mapping $f : \mathbb{R} \to \mathbb{R}$ we write $f_\tau(s) = f(\tau(s))$. We propose to discretise the process U_s, defined by

$$U_s = \frac{\nu_\tau(s) - X^\circ_{\tau(s)}}{\dot\tau(s)},$$

instead on X°_s. This process satisfies the SDE

$$\begin{aligned} \mathrm{d}U_s =& \left(B_\tau(s) v_\tau(s) + \beta_\tau(s) - b_\tau(s, U_s))\right) \mathrm{d}s \\ &- \left(\frac{\ddot\tau(s)}{\dot\tau(s)} I + a_\tau(s, U_s) J(s)\right) U_s \,\mathrm{d}s \\ &+ \frac{1}{\sqrt{\dot\tau(s)}} \sigma_\tau(s, U_s)\,\mathrm{d}W_s, \qquad U_0 = \frac{\nu_\tau(0) - x_0}{2} \end{aligned}$$

Here, J is defined by

$$J(s) = \dot\tau(s) H_\tau(s).$$

Furthermore, we have used the notation $b_\tau(s, y) = b(\tau(s), v_\tau(s) - \dot\tau(s) y)$ and similarly for σ_τ and a_τ. For the likelihood ratio (1.9), note that $\tilde r(\tau(s), X^\circ_{\tau(s)}) = J(s) U_s$.

$$\begin{aligned} \int_0^T G(s, X^\circ_s)\,\mathrm{d}s =& \int_0^T \left(b_\tau(s, U_s) - B_\tau(s, U_s)\right) J(s) U_s \dot\tau(s)\,\mathrm{d}s \\ &- \frac{1}{2}\int_0^T \operatorname{tr}\left(\{a_\tau(s, U_s) - \tilde a_\tau(s)\}\left\{J(s) - J(s) U_s U_s' J(s) \dot\tau(s)\right\}\right) \mathrm{d}s \end{aligned}$$

Finally, by the chain-rule, $P_\tau(s)$ and $\nu_\tau(s)$ satisfy the backward differential equations

$$\frac{\mathrm{d}}{\mathrm{d}t} P_\tau(t) = \left(B_\tau(t) P_\tau(t) + P_\tau B_\tau(t)' - \tilde a_\tau(t)\right) \dot\tau(s)$$

and

$$\frac{\mathrm{d}}{\mathrm{d}t}\nu_\tau(t) = \left(B_\tau(t)\nu_\tau(t) + \beta_\tau(t)\right)\dot{\tau}(s).$$

Acknowledgments

We thank the associate editor for valuable comments that considerably improved the contents and structure of the paper.

Supplementary Material

Animation for tracking the translation of a dynamic Gaussian random field

(doi: 10.1214/21-EJS1894SUPP; .zip). The supplement contains an animation (QuickTime movie) corresponding to Figure 7 showing that the latent convective system bridge moves smoothly from the initial to the final state.

References

[1] Apte, A., Hairer, M., Stuart, A. M. and Voss, J. (2007). Sampling the posterior: An approach to non-Gaussian data assimilation. *Physica D: Nonlinear Phenomena* **230** 50–64. Data Assimilation. MR2345202

[2] Arkin, A., Ross, J. and McAdams, H. H. (1998). Stochastic kinetic analysis of developmental pathway bifurcation in phage λ-infected Escherichia coli cells. *Genetics* **149** 1633–1648.

[3] Arnaudon, A., Holm, D. D. and Sommer, S. (2017). A Geometric Framework for Stochastic Shape Analysis. *ArXiv e-prints 1703.09971*. MR3958804

[4] Arnaudon, A., van der Meulen, F., Schauer, M. and Sommer, S. (2020). Diffusion bridges for stochastic Hamiltonian systems with applications to shape analysis.

[5] Avenel, C., Mémin, E. and Pérez, P. (2014). Stochastic Level Set Dynamics to Track Closed Curves Through Image Data. *Journal of Mathematical Imaging and Vision* **49** 296–316. MR3197342

[6] Bayer, C. and Schoenmakers, J. (2013). Simulation of forward-reverse stochastic representations for conditional diffusions. *Ann. Appl. Probab.* to appear. MR3226170

[7] Beskos, A., Crisan, D., Jasra, A., Kantas, N. and Ruzayqat, H. (2021). Score-Based Parameter Estimation for a Class of Continuous-Time State Space Models. MR4285778

[8] Beskos, A., Papaspiliopoulos, O., Roberts, G. O. and Fearnhead, P. (2006). Exact and computationally efficient likelihood-based estimation for discretely observed diffusion processes. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **68** 333–382. With discussions and a reply by the authors. MR2278331

[9] Beskos, A., Roberts, G., Stuart, A. and Voss, J. (2008). MCMC Methods for diffusion bridges. *Stochastics and Dynamics* **08** 319–350. MR2444507

[10] Beskos, A. and Roberts, G. O. (2005). Exact simulation of diffusions. *Ann. Appl. Probab.* **15** 2422–2444. MR2187299 (2006e:60111)

[11] Bezanson, J., Edelman, A., Karpinski, S. and Shah, V. B. (2017). Julia: a fresh approach to numerical computing. *SIAM Rev.* **59** 65–98. MR3605826

[12] Bierkens, J., van der Meulen, F. and Schauer, M. (2020). Simulation of elliptic and hypo-elliptic conditional diffusions. *Advances in Applied Probability* **52** 173–212. MR4092811

[13] Bladt, M., Finch, S. and Sørensen, M. (2016). Simulation of multivariate diffusion bridges. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **78** 343–369. MR3454200

[14] Chicone, C. (2006). *Ordinary Differential Equations with Applications. Texts in Applied Mathematics.* Springer New York. MR2224508

[15] Clark, J. M. C. (1990). The simulation of pinned diffusions. In *Decision and Control, 1990., Proceedings of the 29th IEEE Conference on* 1418–1420. IEEE.

[16] Cotter, S. L., Roberts, G. O., Stuart, A. M. and White, D. (2013). MCMC Methods for Functions: Modifying Old Algorithms to Make Them Faster. *Statist. Sci.* **28** 424–446. MR3135540

[17] Cuenod, C.-A., Favetto, B., Genon-Catalot, V., Rozenholc, Y. and Samson, A. (2011). Parameter estimation and change-point detection from Dynamic Contrast Enhanced {MRI} data using stochastic differential equations. *Mathematical Biosciences* **233** 68–76. MR2865652

[18] Da Prato, G. and Zabczyk, J. (2014). *Stochastic Equations in Infinite Dimensions*, 2 ed. *Encyclopedia of Mathematics and its Applications.* Cambridge University Press. MR3236753

[19] Delyon, B. and Hu, Y. (2006). Simulation of conditioned diffusion and application to parameter estimation. *Stochastic Processes and their Applications* **116** 1660–1675. MR2269221

[20] Durham, G. B. and Gallant, A. R. (2002). Numerical techniques for maximum likelihood estimation of continuous-time diffusion processes. *J. Bus. Econom. Statist.* **20** 297–338. With comments and a reply by the authors. MR1939904

[21] Elerian, O., Chib, S. and Shephard, N. (2001). Likelihood inference for discretely observed nonlinear diffusions. *Econometrica* **69** 959–993. MR1839375 (2002e:62078)

[22] Eraker, B. (2001). MCMC analysis of diffusion models with application to finance. *J. Bus. Econom. Statist.* **19** 177–191. MR1939708 (2003j:60107)

[23] Evensen, G. (2003). The Ensemble Kalman Filter: theoretical formulation and practical implementation. *Ocean Dynamics* **53** 343–367.

[24] Fearnhead, P., Papaspiliopoulos, O. and Roberts, G. O. (2008). Particle filters for partially observed diffusions. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **70** 755–777. MR2523903

[25] Fuchs, C. (2013). *Inference for diffusion processes*. Springer, Heidelberg With applications in life sciences, With a foreword by Ludwig Fahrmeir. MR3015023

[26] Golightly, A. and Wilkinson, D. J. (2005). Bayesian inference for stochastic kinetic models using a diffusion approximation. *Biometrics* **61** 781–788. MR2196166

[27] Golightly, A. and Wilkinson, D. J. (2008). Bayesian inference for nonlinear multivariate diffusion models observed with error. *Comput. Statist. Data Anal.* **52** 1674–1693. MR2422763

[28] Golightly, A. and Wilkinson, D. J. (2011). Bayesian parameter inference for stochastic biochemical network models using particle Markov chain Monte Carlo. *Interface focus* **1** 807–820.

[29] Graham, M. M., Thiery, A. H. and Beskos, A. (2019). Manifold Markov chain Monte Carlo methods for Bayesian inference in a wide class of diffusion models.

[30] Guarniero, P., Johansen, A. M. and Lee, A. (2017). The Iterated Auxiliary Particle Filter. *Journal of the American Statistical Association* **112** 1636–1647. MR3750887

[31] Haario, H., Saksman, E., Tamminen, J. et al. (2001). An adaptive Metropolis algorithm. *Bernoulli* **7** 223–242. MR1828504

[32] Hairer, M., Stuart, A. M. and Voss, J. (2009). Sampling Conditioned Diffusions. In *Trends in Stochastic Analysis*. *London Mathematical Society Lecture Note Series* **353** 159–186. Cambridge University Press. MR2562154

[33] Hartog, J. and van Zanten, H. (2019). Nonparametric Bayesian label prediction on a large graph using truncated Laplacian regularization. *Communications in Statistics – Simulation and Computation* **0** 1–18. MR3742212

[34] Heng, J., Bishop, A. N., Deligiannidis, G. and Doucet, A. (2020). Controlled sequential Monte Carlo. *The Annals of Statistics* **48** 2904 – 2929. MR4152628

[35] Houtekamer, P. L. and Mitchell, H. L. (2001). A sequential ensemble Kalman filter for atmospheric data assimilation. *Monthly Weather Review* **129** 123–137.

[36] Karatzas, I. and Shreve, S. E. (1991). *Brownian motion and stochastic calculus*, second ed. *Graduate Texts in Mathematics* **113**. Springer-Verlag, New York. MR1121940 (92h:60127)

[37] Kass, R. E., Carlin, B. P., Gelman, A. and Neal, R. M. (1998). Markov chain Monte Carlo in practice: a roundtable discussion. *The American Statistician* **52** 93–100. MR1628427

[38] Kloeden, P. E. and Platen, E. (2013). *Numerical Solution of Stochastic Differential Equations* **23**. Springer Science & Business Media. MR1214374

[39] Lin, M., Chen, R. and Mykland, P. (2010). On generating Monte Carlo samples of continuous diffusion bridges. *J. Amer. Statist. Assoc.* **105** 820–838. MR2724864 (2012b:62109)

[40] Lindström, E. (2012). A regularized bridge sampler for sparsely sampled diffusions. *Stat. Comput.* **22** 615–623. MR2865039

[41] Marchand, J.-L. (2012). *Conditionnement de processus markoviens. IRMAR, Ph.d. Thesis Université de Rennes 1.*

[42] Mider, M., Schauer, M. and Grazzi, S. (2019). BridgeSDEInference v0.2.0. *Zenodo* **10.5281/zenodo.3525435**.

[43] Mider, M., Schauer, M. and van der Meulen, F. (2021). Supplement to "Continuous-discrete smoothing of diffusions". DOI: 10.1214/21-EJS1894SUPP.

[44] Minka, T. P. (2000). Old and new matrix algebra useful for statistics. *Notes*.

[45] Olsson, J. and Ströjby, J. (2011). Particle-based likelihood inference in partially observed diffusion processes using generalised Poisson estimators. *Electron. J. Stat.* **5** 1090–1122. MR2836770

[46] Papaspiliopoulos, O., Roberts, G. O. and Stramer, O. (2013). Data Augmentation for Diffusions. *J. Comput. Graph. Statist.* **22** 665–688. MR3173736

[47] Roberts, G. O. and Rosenthal, J. S. (2009). Examples of adaptive MCMC. *Journal of Computational and Graphical Statistics* **18** 349–367. MR2749836

[48] Roberts, G. O. and Stramer, O. (2001). On inference for partially observed nonlinear diffusion models using the Metropolis-Hastings algorithm. *Biometrika* **88** 603–621. MR1859397 (2002g:62122)

[49] Särkkä, S. (2013). *Bayesian Filtering and Smoothing*. Cambridge University Press, New York, NY, USA. MR3154309

[50] Särkkä, S. and Sarmavuori, J. (2013). Gaussian filtering and smoothing for continuous-discrete dynamic systems. *Signal Processing* **93** 500–510.

[51] Särkkä, S. and Sottinen, T. (2008). Application of Girsanov theorem to particle filtering of discretely observed continuous-time non-linear systems. *Bayesian Anal.* **3** 555–584. MR2434403

[52] Schauer, M. and contributors (2017). Bridge 0.7. *Zenodo* **10.5281/zenodo.1100978**.

[53] Schauer, M., van der Meulen, F. and van Zanten, H. (2017). Guided proposals for simulating multi-dimensional diffusion bridges. *Bernoulli* **23** 2917–2950. MR3648050

[54] Tanner, M. A. and Wong, W. H. (1987). The calculation of posterior distributions by data augmentation. *J. Amer. Statist. Assoc.* **82** 528–550. With discussion and with a reply by the authors. MR898357 (88h:62050)

[55] van der Meulen, F. and Schauer, M. (2017). Bayesian estimation of incompletely observed diffusions. *Stochastics* **90** 641–662. MR3810575

[56] van der Meulen, F. and Schauer, M. (2017). Bayesian estimation of discretely observed multi-dimensional diffusion processes using guided proposals. *Electron. J. Stat.* **11** 2358–2396. MR3656495

[57] van der Meulen, F. and Schauer, M. (2017c). On residual and guided proposals for diffusion bridge simulation. *arXiv:1708.04870*. MR3648050

[58] van der Meulen, F. and Schauer, M. (2021). Automatic Backward Filtering Forward Guiding for Markov processes and graphical models.

[59] van der Meulen, F., Schauer, M. and van Zanten, H. (2014). Re-

versible jump MCMC for nonparametric drift estimation for diffusion processes. *Comput. Statist. Data Anal.* **71** 615–632. MR3131993

[60] van Kampen, N. G. (1981). *Stochastic processes in physics and chemistry. Lecture Notes in Mathematics* **888**. North-Holland Publishing Co., Amsterdam-New York. MR648937

[61] Verner, J. H. (1978). Explicit Runge–Kutta methods with estimates of the local truncation error. *SIAM Journal on Numerical Analysis* **15** 772–790. MR0483471

[62] Whitaker, G. A., Golightly, A., Boys, R. J. and Sherlock, C. (2017). Improved bridge constructs for stochastic differential equations. *Statistics and Computing* **27** 885–900. MR3627552

[63] Wilkinson, D. J. (2006). *Stochastic modelling for systems biology. Chapman & Hall/CRC Mathematical and Computational Biology Series*. Chapman & Hall/CRC, Boca Raton, FL. MR2222876