Nonlin. Processes Geophys., 21, 633–643, 2014
www.nonlin-processes-geophys.net/21/633/2014/
doi:10.5194/npg-21-633-2014

Nonlinear Processes in Geophysics
Open Access

Monte Carlo fixed-lag smoothing in state-space models

A. Cuzol[1] **and E. Mémin**[2]

[1]University of Bretagne-Sud, UMR 6205, LMBA, 56000 Vannes, France
[2]INRIA Rennes-Bretagne Atlantique, Rennes, France

Correspondence to: A. Cuzol (anne.cuzol@univ-ubs.fr)

Received: 16 April 2013 – Revised: 13 March 2014 – Accepted: 14 March 2014 – Published: 28 May 2014

Abstract. This paper presents an algorithm for Monte Carlo fixed-lag smoothing in state-space models defined by a diffusion process observed through noisy discrete-time measurements. Based on a particle approximation of the filtering and smoothing distributions, the method relies on a simulation technique of conditioned diffusions. The proposed sequential smoother can be applied to general nonlinear and multidimensional models, like the ones used in environmental applications. The smoothing of a turbulent flow in a high-dimensional context is given as a practical example.

1 Introduction

The framework of this paper concerns state-space models described by general diffusions of the form

$$\mathrm{d}\boldsymbol{x}(t) = f(\boldsymbol{x}(t))\mathrm{d}t + \sigma(\boldsymbol{x}(t))\mathrm{d}\boldsymbol{B}(t), \qquad (1)$$

which are partially observed through noisy measurements at discrete times. Such models can describe many dynamical phenomena in the environmental sciences and physics, but also in finance or engineering applications. The main motivation of this work concerns environmental applications, where nonlinearity and high-dimensionality arise. Indeed, environmental models and data describe nonlinear phenomena over large domains, with high spatial resolution. The continuous dynamical model (Eq. 1) is defined from a priori physical laws, while observations are supplied by sensors (satellite data for instance) and can appear with very low time frequency. As an example, in the application presented in the last part of this paper, the dimension of the state and observations is of the order of many thousands, and the model is described by the nonlinear Navier–Stokes equation. Filtering and smoothing in such state-space models aim at coupling model and observations, which is called data assimilation. The goal of the filtering is to estimate the system state distribution knowing past and present observations. This allows us for instance to give proper initial conditions to forecast the future state of a system characterizing atmospheric or oceanographic flows. On the other hand, the smoothing aims at estimating the state distribution using past and future observations, and this retrospective state estimation allows us to analyze a spatio–temporal phenomenon over a given time period, for climatology studies for instance. Applications of data assimilation are numerous and interest is growing in environmental sciences with the increase in available data. However, it is still a challenge to develop filtering and smoothing methods that can be used within a general nonlinear and high-dimensional context.

Monte Carlo sequential methods, contrary to standard Kalman filters, are able to deal with the filtering problem in nonlinear state-space models. The particle filtering (Del Moral et al., 2001; Doucet et al., 2000) solves the whole filtering equations through Monte Carlo approximations of the state distribution. On the other hand, ensemble Kalman methods (Evensen, 2003) take into account in some way the nonlinearities in the system, but are based on a Gaussian assumption. For high-dimensional systems, ensemble Kalman methods are preferred in practice to particle filters (Stroud et al., 2010; Van Leeuwen, 2009) since they reach a better performance for a limited number of particles. In order to keep this advantage while alleviating the Gaussian assumption, both methods are combined in Papadakis et al. (2010), leading to a particle filter that can be applied to high-dimensional systems. We will use this technique for the filtering step in the high-dimensional application presented in Sect. 5.

The aim of this paper is to propose a new smoothing method. Within the particle filter framework, the smoothing can be computed backward, reweighting past particles

Published by Copernicus Publications on behalf of the European Geosciences Union & the American Geophysical Union.

using present observations (Briers et al., 2010; Godsill et al., 2004). There is however one main difficulty for continuous models of type Eq. (1). As a matter of fact, it is necessary to know the transition density of the process between observation times, which is not available for general diffusions. This transition density can be approximated through Monte Carlo simulations, as proposed by Durham and Gallant (2002) to solve inference problems for diffusion processes. However, these approximations are based on Brownian bridge (or modified versions of it) simulations that do not take into account the drift part of the model. For nonlinear and high-dimensional models with a drift term that dominates, such approximations will be inefficient. It is also possible to obtain an unbiased estimate of the transition density (see Beskos et al., 2006), but this approach is not adapted to a multi-dimensional context. As a matter of fact, the use of this technique in a multivariate setting imposes constraints on the diffusion drift (in particular, the drift function has to be of gradient type). In parallel, within the framework of ensemble Kalman methods, Evensen and van Leeuwen (2000) have proposed estimating backward the smoothing distribution in a recursive way, based on existing filtering trajectories; Stroud et al. (2010) presented and applied an ensemble Kalman smoothing method, relying on a linearization of the system dynamics.

All previously mentioned smoothing methods require us to perform specific assumptions or simplifications in order to deal with general nonlinear models of type Eq. (1) in a high-dimensional context. To the best of our knowledge, it remains a challenging problem to develop smoothing methods that can be used in this general setting. In this paper, we deal with this issue sequentially each time a new observation is available, by smoothing the hidden state from this new observation time up to the previous one. This approach, called fixed-lag smoothing, then constitutes a partial answer to the global smoothing problem that would take into account all available observations. Nevertheless, it is reasonable to assume that the distribution of the hidden state depends on future observations through the next observation only, as soon as the time step between measurements is long (which is typically the case in the environmental applications that motivate this work). Under this assumption, a new observation will impact the distribution of the hidden process up to the previous observation only. This point of view justifies the use of a fixed-lag smoothing in our setting as a reasonable approximation of the global smoothing problem.

Such a fixed-lag smoothing may be directly obtained from the particle filtering result, reweighting past trajectories. However, this smoothing will fail in two cases: when the number of particles is too small compared to the size of the system, or when existing trajectories do not correspond to plausible trajectories of the dynamical model. Unfortunately, these two situations have to be faced when smoothing in a high-dimensional state-space model. Firstly, the number of particles has to be reduced for computational reasons. Secondly, particle filters that have been proposed in this high-dimensional context require us to correct trajectories towards the observations (Papadakis et al., 2010; Van Leeuwen and Ades, 2013). This implies that filtering states are consistent at observation times, but also that filtering trajectories may not be plausible realizations of the underlying physical model. In that case, a smoothing based on existing trajectories will fail. Note that these remarks are not only valid for the fixed-lag smoothing, but also for previously mentioned global techniques relying on existing trajectories. In particular, a genealogical smoothing based on ancestral particle lines (Del Moral, 2004) will be deficient in a high-dimensional setting, since many trajectories will share only a few ancestral lines.

In contrast, our method does not rely on existing particles only. It is built on a conditional simulation technique of diffusions proposed by Delyon and Hu (2006) that provides new state trajectories at hidden times between observations. This simulation technique is adapted to a multivariate context where the drift dominates, contrary to techniques based on Brownian bridge sampling (Durham and Gallant, 2002). Moreover, it does not require constraining assumptions for multivariate models, contrary to other techniques based on the exact simulation of diffusions (Beskos and Roberts, 2005; Beskos et al., 2006). The proposed smoothing method can then be applied to high-dimensional systems. Finally, it does not require model linearization nor Gaussian hypotheses, and so is able to deal with general nonlinear models.

The remainder of the paper is organized as follows. Section 2 briefly describes sequential Monte Carlo filtering methods in state-space models, and presents the fixed-lag smoothing problem. Section 3 presents the conditional simulation technique of diffusions of Delyon and Hu (2006), and details the construction of the proposed Monte Carlo estimate of smoothing distributions. The method is then experimented on a one-dimensional example in Sect. 4. Finally, the method is applied in Sect. 5 to a practical nonlinear and high-dimensional case, similar to the problems that have to be faced in environmental applications. A discussion is given in Sect. 6.

2 Monte Carlo filtering and smoothing in state-space models

In this section we recall briefly the particle filtering and smoothing methods for models of type Eq. (1), where the hidden state vector $\boldsymbol{x} \in \mathbb{R}^n$ is observed through the observation vector $\boldsymbol{y} \in \mathbb{R}^m$ at discrete times $\{t_1, t_2, \ldots\}$:

$$\boldsymbol{y}(t_k) = g(\boldsymbol{x}(t_k)) + \boldsymbol{\gamma}_{t_k}. \tag{2}$$

The drift function f and observation operator g can be nonlinear. The dynamical model uncertainty is described by an n-dimensional Brownian motion with covariance $\boldsymbol{\Sigma} =$

$\sigma(\boldsymbol{x}(t))\sigma(\boldsymbol{x}(t))^T$. The functions f, g and σ are assumed to be known, as well as the law of the observation noise $\boldsymbol{\gamma}_{t_k}$.

In particular, we present the standard particle filter and the weighted ensemble Kalman filter, which can be used to face the filtering problem in high-dimensional systems.

2.1 Particle-based filtering methods

Filtering aims at estimating recursively the distribution $p(\boldsymbol{x}_{t_1:t_k}|\boldsymbol{y}_{t_1:t_k})$ (and in particular its marginal distribution $p(\boldsymbol{x}_{t_k}|\boldsymbol{y}_{t_1:t_k})$) at each observation time t_k. This filtering problem can be solved with a Monte Carlo sequential approach, called particle filtering (Del Moral et al., 2001; Doucet et al., 2000). The method relies on a Monte Carlo approximation of the filtering distribution over a set of weighted trajectories $\{\boldsymbol{x}^{(i)}_{t_1:t_k}\}_{i=1:N}$ (called particles):

$$\hat{p}(\boldsymbol{x}_{t_1:t_k}|\boldsymbol{y}_{t_1:t_k}) = \sum_{i=1}^{N} w^{(i)}_{t_k}\delta_{\boldsymbol{x}^{(i)}_{t_1:t_k}}(\boldsymbol{x}_{t_1:t_k}), \tag{3}$$

whose marginal distribution at time t_k is written as

$$\hat{p}(\boldsymbol{x}_{t_k}|\boldsymbol{y}_{t_1:t_k}) = \sum_{i=1}^{N} w^{(i)}_{t_k}\delta_{\boldsymbol{x}^{(i)}_{t_k}}(\boldsymbol{x}_{t_k}). \tag{4}$$

Particle filters rely on a sequential importance sampling scheme that recursively samples particles, and updates their weights at observation times. The weights correspond to the ratio between the target distribution and the importance sampling distribution $\pi(\boldsymbol{x}_{t_k}|\boldsymbol{x}_{t_0:t_{k-1}},\boldsymbol{y}_{t_1:t_k})$. They are recursively computed as follows:

$$w^{(i)}_{t_k} \propto w^{(i)}_{t_{k-1}}\frac{p(\boldsymbol{y}_{t_k}|\boldsymbol{x}^{(i)}_{t_k})p(\boldsymbol{x}^{(i)}_{t_k}|\boldsymbol{x}^{(i)}_{t_{k-1}})}{\pi(\boldsymbol{x}^{(i)}_{t_k}|\boldsymbol{x}^{(i)}_{t_0:t_{k-1}},\boldsymbol{y}_{t_1:t_k})}. \tag{5}$$

In practice, a resampling procedure is added in order to avoid degeneracy. This procedure duplicates trajectories with large weights and removes small weighted trajectories.

2.1.1 Standard particle filter

When the proposal distribution π is set to the prior (i.e., $\pi(\boldsymbol{x}_{t_k}|\boldsymbol{x}_{t_0:t_{k-1}},\boldsymbol{y}_{t_1:t_k}) = p(\boldsymbol{x}_{t_k}|\boldsymbol{x}_{t_{k-1}})$), the weight updating rule (Eq. 5) is simplified to the computation of the data likelihood $p(\boldsymbol{y}_{t_k}|\boldsymbol{x}^{(i)}_{t_k})$. This particular instance of the particle filter is called the *Bootstrap filter* or sequential importance resampling (SIR) filter (Gordon et al., 1993). Due to its simplicity it is the most commonly used particle filter. It is however a very inefficient distribution for high-dimensional space as it does not take into account the current observation and depends only weakly on the past data through the filtering distribution estimated at the previous instant. This distribution requires a great number of particles to explore meaningful areas of the state space.

2.1.2 Weighted ensemble Kalman filter (WEnKF)

One way to incorporate observation within the proposal distribution efficiently consists in relying on the ensemble Kalman filtering mechanism to define this distribution. This is the idea proposed in the WEnKF technique (Papadakis et al., 2010). In the WEnKF approach the importance sampling is taken as a Gaussian approximation of $p(\boldsymbol{x}_{t_k}|\boldsymbol{x}_{t_{k-1}},\boldsymbol{y}_{t_k})$. This approach is close to the technique proposed in Van Leeuwen (2010). A variation of a similar technique based on a deterministic square-root formulation is also described in Beyou et al. (2013). In order to make the estimation of the filtering distribution exact (up to the sampling), each member of the ensemble must be weighted at each observation instant t_k with appropriate weights $w^{(i)}_{t_k}$, defined from Eq. (5). Therefore, the weighted ensemble Kalman filter (WEnKF) procedure can be simply summarized by Algorithm 1.

Algorithm 1 The WEnKF algorithm

```
For each $t_k = t_1, t_2, \ldots$:
  • Start from particle set $\{\boldsymbol{x}^{(i)}_{t_{k-1}}, i = 1,\ldots,N\}$ and observation $\boldsymbol{y}_{t_k}$
  • Obtain particle set $\{\boldsymbol{x}^{(i)}_{t_k}, i = 1,\ldots,N\}$ from:
      – EnKF step: Get $\boldsymbol{x}^{(i)}_{t_k}, i = 1,\ldots,N$, from the assimilation of $\boldsymbol{y}_{t_k}$ with an EnKF procedure;
      – Computation of weights: $w^{(i)}_{t_k} \propto w^{(i)}_{t_{k-1}}\frac{p(\boldsymbol{y}_{t_k}|\boldsymbol{x}^{(i)}_{t_k})p(\boldsymbol{x}^{(i)}_{t_k}|\boldsymbol{x}^{(i)}_{t_{k-1}})}{p(\boldsymbol{x}^{(i)}_{t_k}|\boldsymbol{x}^{(i)}_{t_{k-1}},\boldsymbol{y}^{(i)}_{t_k})}$;
      – Resampling: For $j = 1,\ldots,N$, sample with replacement index $I(j)$ from discrete probability $\{w^{(i)}_{t_k}, i = 1,\ldots,N\}$ over $\{1,\ldots,N\}$ and set $\boldsymbol{x}^{(j)}_{t_k} = \boldsymbol{x}^{I(j)}_{t_k}$. Set $w^{(i)}_{t_k} = \frac{1}{N} \quad \forall i = 1,\ldots,N$.
```

Note that particle-based filtering techniques update the filtering distribution at observation times only. However, after the estimate $\hat{p}(\boldsymbol{x}_{t_k}|\boldsymbol{y}_{t_1:t_k})$ has been updated at observation time t_k, the filtering distribution can be predicted in order to have a continuous estimation of $\hat{p}(\boldsymbol{x}_t|\boldsymbol{y}_{t_1:t_k})$ for all $t \in]t_k, t_{k+1}[$ until the next observation time

$$\hat{p}(\boldsymbol{x}_t|\boldsymbol{y}_{t_1:t_k}) = \sum_{i=1}^{N} w^{(i)}_{t_k}\delta_{\boldsymbol{x}^{(i)}_t}(\boldsymbol{x}_t), \tag{6}$$

where, for all $i = 1,\ldots,N$, the state $\boldsymbol{x}^{(i)}_t$ is sampled from Eq. (1), starting from $\boldsymbol{x}^{(i)}_{t_k}$.

2.2 Fixed-lag smoothing problem

Contrary to the filtering approach that uses past and present observations, the smoothing in state-space models aims at estimating $p(\boldsymbol{x}_t|\boldsymbol{y}_{t_1:t_{\text{end}}})$ for all $t \in [t_1, t_{\text{end}}]$, using all past and future observations over a given time period. As raised in the introduction, existing smoothing methods do not apply directly to a general nonlinear model of type Eq. (1) in a high-dimensional context, since assumptions have to be made that may not be realistic. Instead of solving the global smoothing, we will concentrate in the rest of the paper on a fixed-lag smoothing, which constitutes a partial answer to the global smoothing problem.

The objective of the fixed-lag smoothing will be to replace the predictive distribution (Eq. 6) by its smoothed version $p(\boldsymbol{x}_t|\boldsymbol{y}_{t_1:t_{k+1}})$ $\forall t \in]t_k, t_{k+1}]$ sequentially each time a new observation $\boldsymbol{y}_{t_{k+1}}$ arrives. This will allow us to reduce the temporal discontinuities inherent in the filtering technique that successively predicts the distribution of the state between observations, and updates this distribution at observation times.

To achieve this, by construction of the particle filter that weights entire trajectories (see Eq. 3), it is known (see for instance Doucet et al., 2000) that the fixed-lag smoothing distribution $\hat{p}(\boldsymbol{x}_t|\boldsymbol{y}_{t_1:t_{k+1}})$ can be directly obtained from the marginal at time t of $\hat{p}(\boldsymbol{x}_{t_1:t_{k+1}}|\boldsymbol{y}_{t_1:t_{k+1}})$. The empirical smoothing distribution is then given by

$$\hat{p}(\boldsymbol{x}_t|\boldsymbol{y}_{t_1:t_{k+1}}) = \sum_{i=1}^{N} w_{t_{k+1}}^{(i)} \delta_{\boldsymbol{x}_t^{(i)}}(\boldsymbol{x}_t) \quad \forall t \in]t_k, t_{k+1}]. \tag{7}$$

However, this approximation is simply a reweighting of past existing particle trajectories, and relies on the support of the filtering distribution at time t_k. If the number of particles is too small with respect to the state dimension, the support may be greatly reduced by the correction step (assigning small weights to all particles except a few), leading in practice to a bad estimation of $p(\boldsymbol{x}_t|\boldsymbol{y}_{t_1:t_{k+1}})$. Moreover, if particle trajectories have been forced towards observations during the filtering step (like in the WEnKF procedure), a smoothing based on those particles will fail because it will not be able to correct discontinuities. Consequently, since we are interested in smoothing techniques that are efficient in a high-dimensional context, this direct smoothing technique can not be used in its basic form and has to be improved.

In the following, we propose to use a conditional simulation technique of diffusions that will enable the sampling of new smoothed trajectories between times t_k and t_{k+1}. The approximation of the smoothing distribution (Eq. 7) at each hidden time will then be improved. The conditional simulation technique is presented in the next section, before the resulting smoothing procedure we propose.

3 Fixed-lag smoothing with conditional simulation

The smoothing method we propose is based on a conditional simulation technique that is presented in Sect. 3.1. We then develop in Sect. 3.2 how this technique can be used to improve the estimation of the smoothing distribution (Eq. 7).

3.1 Conditional simulation

Conditional simulation of a diffusion aims at sampling trajectories from a given process

$$\mathrm{d}\boldsymbol{x}(t) = f(\boldsymbol{x}(t))\mathrm{d}t + \sigma(\boldsymbol{x}(t))\mathrm{d}\boldsymbol{B}(t) \tag{8}$$

between two times $t = 0$ and $t = T$, with the constraints $\boldsymbol{x}(0) = \boldsymbol{u}$ and $\boldsymbol{x}(T) = \boldsymbol{v}$. This simulation problem is treated by Delyon and Hu (2006), where the authors show how to obtain the law of the constrained process from a Girsanov theorem. In practice, the proposed algorithms consist in simulating trajectories according to another diffusion process, which is built to respect the constraints and is easy to simulate from. The conditional distribution of the constrained process (Eq. 8) is absolutely continuous with respect to the distribution of the auxiliary process, with explicitly given density. For instance, in the case where the drift is bounded (a similar algorithm is proposed in Delyon and Hu (2006) for the unbounded case) and with σ invertible, the algorithm is based on the simulation of trajectories from the following process:

$$\mathrm{d}\tilde{\boldsymbol{x}}(t) = \left(f(\tilde{\boldsymbol{x}}(t)) - \frac{\tilde{\boldsymbol{x}}(t) - \boldsymbol{v}}{T - t} \right) \mathrm{d}t + \sigma(\tilde{\boldsymbol{x}}(t))\mathrm{d}\boldsymbol{B}(t), \tag{9}$$

with initial condition $\tilde{\boldsymbol{x}}(0) = \boldsymbol{u}$. Note that Lemma 4 in Delyon and Hu (2006) deals with the existence of a unique solution for this equation. This process is a simple modification of Eq. (8), where a deterministic part is added to the drift. It is then easy to simulate unconditional trajectories from this process, and all simulated trajectories will satisfy $\tilde{\boldsymbol{x}}(T) = \boldsymbol{v}$ by construction. For simplicity we will assume in the following that σ is independent of $\boldsymbol{x}(t)$ (note however that this is not an assumption in Delyon and Hu (2006)). The law of the conditioned process is given by

$$\mathbb{E}[h(\boldsymbol{x})|\boldsymbol{x}(0) = \boldsymbol{u}, \boldsymbol{x}(T) = \boldsymbol{v}] = \mathbb{E}\left[h(\tilde{\boldsymbol{x}})\alpha(\tilde{\boldsymbol{x}})\right], \tag{10}$$

for all measurable functions h, where

$$\alpha(\tilde{\boldsymbol{x}}) = \exp\left(- \int_0^T \frac{(\tilde{\boldsymbol{x}}(t) - \boldsymbol{v})^T \boldsymbol{\Sigma}^{-1} f(\tilde{\boldsymbol{x}}(t))}{T - t} \mathrm{d}t \right) \tag{11}$$

is the density coming from the Girsanov theorem (see Delyon and Hu, 2006), with $\boldsymbol{\Sigma} = \sigma(\tilde{\boldsymbol{x}}(t))\sigma(\tilde{\boldsymbol{x}}(t))^T$.

Let us note that the presence of the drift part of model (Eq. 8) in the auxiliary process (Eq. 9) is crucial to making the simulation efficient. The same process had initially been proposed by Clark (1990) to solve the conditional simulation problem. On the other hand, standard Brownian bridges that could be used as auxiliary processes (Durham and Gallant, 2002) lead in practice to poor approximations of the original constrained diffusion in our high-dimensional setting, since Brownian bridge trajectories are too far away from trajectories of Eq. (8).

In the following, the conditional marginal of interest $p(\boldsymbol{x}_t|\boldsymbol{x}(0)=\boldsymbol{u},\boldsymbol{x}(T)=\boldsymbol{v})$ will then be approximated as follows:

$$\hat{p}(\boldsymbol{x}_t|\boldsymbol{x}(0)=\boldsymbol{u},\boldsymbol{x}(T)=\boldsymbol{v}) = \sum_{j=1}^{M} \alpha(\tilde{\boldsymbol{x}}^{(j)})\delta_{\tilde{\boldsymbol{x}}_t^{(j)}}(\boldsymbol{x}_t) \quad \forall t \in [0,T], \tag{12}$$

where the M trajectories $\{\tilde{\boldsymbol{x}}_t^{(j)}\}_{j=1:M}$ are simulated from Eq. (9) with $\tilde{\boldsymbol{x}}_0^{(j)}=\boldsymbol{u}$ for all $j=1,\ldots,M$.

3.2 Proposed fixed-lag smoothing method

We show in the following how the conditional simulation technique can be used to improve the estimation of the local smoothing distribution $p(\boldsymbol{x}_t|\boldsymbol{y}_{t_1:t_{k+1}})$ for all $t\in]t_k,t_{k+1}]$.

We first note that this distribution can be decomposed as

$$\begin{aligned} p(\boldsymbol{x}_t|\boldsymbol{y}_{t_1:t_{k+1}}) &= \int p(\boldsymbol{x}_t,\boldsymbol{x}_{t_k},\boldsymbol{x}_{t_{k+1}}|\boldsymbol{y}_{t_1:t_{k+1}})\mathrm{d}\boldsymbol{x}_{t_k}\mathrm{d}\boldsymbol{x}_{t_{k+1}} \\ &= \int p(\boldsymbol{x}_{t_k},\boldsymbol{x}_{t_{k+1}}|\boldsymbol{y}_{t_1:t_{k+1}})p(\boldsymbol{x}_t|\boldsymbol{x}_{t_k},\boldsymbol{x}_{t_{k+1}},\boldsymbol{y}_{t_1:t_{k+1}}) \\ &\quad \mathrm{d}\boldsymbol{x}_{t_k}\mathrm{d}\boldsymbol{x}_{t_{k+1}}. \end{aligned} \tag{13}$$

Then, from the state-space model properties, we obtain

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{t_1:t_{k+1}}) = \int p(\boldsymbol{x}_{t_k},\boldsymbol{x}_{t_{k+1}}|\boldsymbol{y}_{t_1:t_{k+1}})p(\boldsymbol{x}_t|\boldsymbol{x}_{t_k},\boldsymbol{x}_{t_{k+1}}) \mathrm{d}\boldsymbol{x}_{t_k}\mathrm{d}\boldsymbol{x}_{t_{k+1}}. \tag{14}$$

Moreover, from the particle filter Monte Carlo approximation described by Eq. (3), the joint law $p(\boldsymbol{x}_{t_k},\boldsymbol{x}_{t_{k+1}}|\boldsymbol{y}_{t_1:t_{k+1}})$ can be replaced by

$$\hat{p}(\boldsymbol{x}_{t_k},\boldsymbol{x}_{t_{k+1}}|\boldsymbol{y}_{t_1:t_{k+1}}) = \sum_{i=1}^{N} w_{t_{k+1}}^{(i)}\delta_{(\boldsymbol{x}_{t_{k+1}}^{(i)},\boldsymbol{x}_{t_k}^{(i)})}(\boldsymbol{x}_{t_{k+1}},\boldsymbol{x}_{t_k}), \tag{15}$$

where $w_{t_{k+1}}^{(i)}$ are the particle filter importance weights.

Plugging Eq. (15) into Eq. (14) then leads to the following approximation for the fixed-lag smoothing distribution:

$$\hat{p}(\boldsymbol{x}_t|\boldsymbol{y}_{t_1:t_{k+1}}) = \sum_{i=1}^{N} w_{t_{k+1}}^{(i)} p(\boldsymbol{x}_t|\boldsymbol{x}_{t_k}^{(i)},\boldsymbol{x}_{t_{k+1}}^{(i)}). \tag{16}$$

The conditional distribution $p(\boldsymbol{x}_t|\boldsymbol{x}_{t_k}^{(i)},\boldsymbol{x}_{t_{k+1}}^{(i)})$ can be estimated using Eq. (12) for each pair of initial and end points $\boldsymbol{x}_{t_k}^{(i)}$ and $\boldsymbol{x}_{t_{k+1}}^{(i)}$:

$$\hat{p}(\boldsymbol{x}_t|\boldsymbol{x}_{t_k}^{(i)},\boldsymbol{x}_{t_{k+1}}^{(i)}) = \sum_{j=1}^{M}\alpha(\tilde{\boldsymbol{x}}^{(i)(j)})\delta_{\tilde{\boldsymbol{x}}_t^{(i)(j)}}(\boldsymbol{x}_t), \tag{17}$$

where each $\tilde{\boldsymbol{x}}_t^{(i)(j)}$ is sampled from Eq. (9) with initial constraint $\tilde{\boldsymbol{x}}_{t_k}^{(i)(j)}=\boldsymbol{x}_{t_k}^{(i)}$ and final constraint $\boldsymbol{x}_{t_{k+1}}^{(i)}$.

The estimation of the smoothing distribution of interest is finally written as

$$\hat{p}(\boldsymbol{x}_t|\boldsymbol{y}_{t_1:t_{k+1}}) = \sum_{i=1}^{N} w_{t_{k+1}}^{(i)}\sum_{j=1}^{M}\alpha(\tilde{\boldsymbol{x}}^{(i)(j)})\delta_{\tilde{\boldsymbol{x}}_t^{(i)(j)}}(\boldsymbol{x}_t), \quad \forall t\in]t_k,t_{k+1}]. \tag{18}$$

The algorithm we propose to compute the fixed-lag smoothing distribution on a given time interval $[t_k,t_{k+1}]$ is therefore the following:

Algorithm 2 Fixed-lag conditional smoothing

```
For each $t_k = t_1, t_2, \ldots$:
  – Store $\{\boldsymbol{x}_{t_k}^{(i)}\}_{i=1:N}$ and compute $\{\boldsymbol{x}_{t_{k+1}}^{(i)}\}_{i=1:N}$ and associated weights $\{w_{t_{k+1}}^{(i)}\}_{i=1:N}$ from a particle filter algorithm;
  – For each pair $\{\boldsymbol{x}_{t_k}^{(i)},\boldsymbol{x}_{t_{k+1}}^{(i)}\}$, $i=1,\ldots,N$:
      – Simulate $M$ conditional trajectories $\{\tilde{\boldsymbol{x}}_t^{(i)(j)}\}_{j=1:M}$ for $t\in[t_k,t_{k+1}]$ from Eq. (9) with an Euler scheme, with the constraints $\tilde{\boldsymbol{x}}_{t_k}^{(i)(j)}=\boldsymbol{x}_{t_k}^{(i)}$ and $\tilde{\boldsymbol{x}}_{t_{k+1}}^{(i)(j)}=\boldsymbol{x}_{t_{k+1}}^{(i)}$,
      – Compute weights $\alpha(\tilde{\boldsymbol{x}}^{(i)(j)})$ from Eq. (11) for all $j=1,\ldots,M$ , with final constraint $\boldsymbol{x}_{t_{k+1}}^{(i)}$;
  – Compute $\hat{p}(\boldsymbol{x}_t|\boldsymbol{y}_{t_1:t_{k+1}}) =\sum_{i=1}^{N} w_{t_{k+1}}^{(i)}\sum_{j=1}^{M}\alpha(\tilde{\boldsymbol{x}}^{(i)(j)})\delta_{\tilde{\boldsymbol{x}}^{(i)(j)}}(\boldsymbol{x}_t)$ for all $t\in]t_k,t_{k+1}]$.
```

4 One-dimensional simulation study

In this section, the smoothing method is first experimented on a one-dimensional state space model. Since the proposed approach relies on a preliminary particle filtering step, filtering results are first presented in Sect. 4.2 (either considering a standard particle filter or the WEnKF). The results obtained with the standard fixed-lag smoothing method are then shown in Sect. 4.3. Finally, Sect. 4.4 presents the smoothing results obtained with the proposed technique.

4.1 State-space model

The one-dimensional state-space model of interest is a sine diffusion, partially observed with noise (used as an illustration by Fearnhead et al. (2008) for a particle filtering method):

$$\mathrm{d}x(t) = \sin(x(t))\mathrm{d}t + \sigma_x \mathrm{d}B(t), \tag{19}$$

$$y_{t_k} = x_{t_k} + \gamma_{t_k}, \tag{20}$$

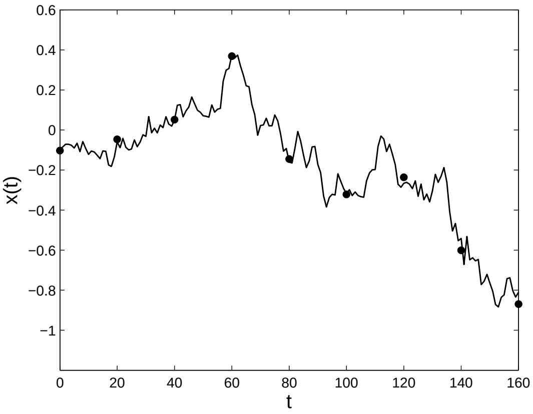
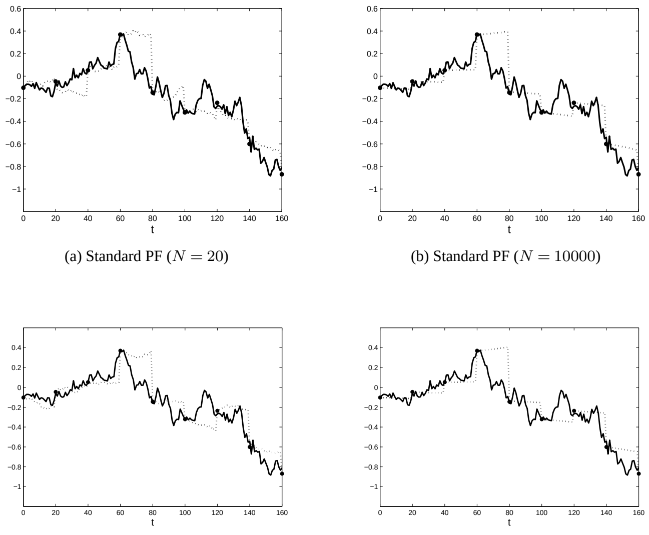


Figure 1. Simulated sine diffusion trajectory $x(t)$ and partial observations $y(t_k)$ (dots) with $t_k - t_{k-1} = 20\Delta t$.



(a) Standard PF ($N = 20$)

(b) Standard PF ($N = 10000$)

(c) WEnKF ($N = 20$)

(d) WEnKF ($N = 10000$)

Figure 2. Standard PF and WEnKF results. Thick line: hidden diffusion; dots: partial observations; dotted line: estimated filtering mean.

where $\sigma_x^2 = 0.5$ and $\gamma_{t_k} \sim \mathcal{N}(0, \sigma_y)$ with $\sigma_y^2 = 0.01$. One trajectory of the process is first simulated from Eq. (19) with an Euler-type discretization scheme of time step $\Delta t = 0.005$. This trajectory will constitute the hidden process, observed through y_{t_k} generated according to Eq. (20) at every time step t_k, with $t_k - t_{k-1} = 20\Delta t$. The trajectory is plotted in Fig. 1, together with the corresponding discrete observations at times t_k.

4.2 Particle filtering results

The filtering results are presented for the standard particle filter (denoted PF in the following) and the weighted ensemble Kalman filter (WEnKF). Two situations are shown, with a reduced ($N = 20$) and high number ($N = 10\,000$) of particles. The case with a high number of particles is shown as the reference for comparison purpose; note however that this ideal situation is not reachable in a high-dimensional context, since the number of particles has to be reduced for evident computational cost reasons.

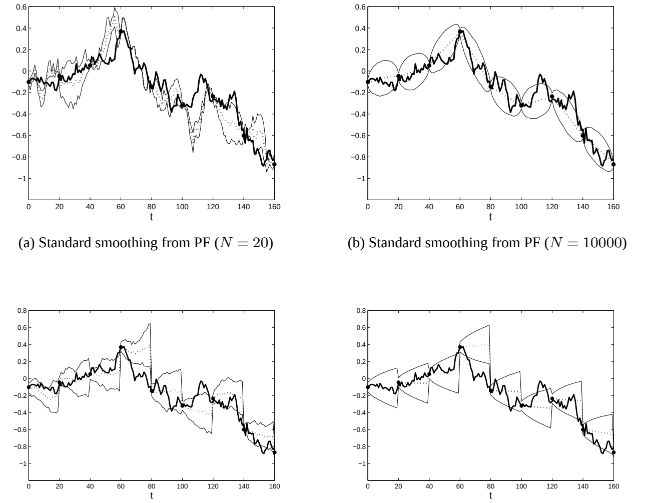
The results for the two configurations are presented in Fig. 2, where the dotted lines represent the filtering mean estimates. The filtering distribution $p(x_{t_k}|y_{t_1:t_k})$ is estimated at each observation time t_k using Eq. (4), and predicted between observation times from Eq. (6). The mean is then estimated from weighted particles as $\sum_{i=1}^N w_{t_k}^{(i)} x_t^{(i)}$, for all $t \in [t_k, t_{k+1}[$. Figure 2a–b shows that the standard particle filter results diverge from the reference solution between observation times, for low or high number of particles. As a matter of fact, when no observation is available, the state distribution is predicted from the dynamics only, so that particles trajectories are not guided towards the next observation. At observation times t_k, high weights are given to particles that are close to the observation, so that the estimated mean suddenly gets closer to the solution. These discontinuities between measurement times can also be observed in the WEnKF results (Fig. 2c–d), because particle trajectories are brutally corrected with the EnKF step at observation times. A smoothing will aim at reducing these temporal discontinuities, while providing dynamically consistent solutions.

4.3 Standard fixed-lag smoothing results

From the particle filtering results, we now present the results obtained with the direct particle smoothing procedure described in Sect. 2.2. This procedure relies on existing trajectories. The smoothing distribution $\hat{p}(x_t|y_{t_1:t_{k+1}})$ is computed backward for all $t \in]t_k, t_{k+1}]$ using expression Eq. (7) each time a new observation $y_{t_{k+1}}$ becomes available. The smoothing mean is computed as $\sum_{i=1}^N w_{t_{k+1}}^{(i)} x_t^{(i)}$ for all $t \in]t_k, t_{k+1}]$, and the standard deviation is computed in the same way from the weighted particles.

It can be observed in Fig. 3a that the smoothing based on the standard particle filter is not efficient when the number of particles N is small: only a few particles are close to the observation at time t_k and have nonzero weights, implying that the smoothing distribution is poorly estimated (see for instance between observation times $t = 100$ and $t = 120$, where the smoothing distribution is artificially peaked but far from the hidden trajectory). The smoothing result obtained from the reference configuration $N = 10\,000$ is plotted in Fig. 3b. In that situation, since many trajectories have high weights at observation times, the estimation of backward smoothing distributions is improved and includes the hidden trajectory.

Moreover, Fig. 3c–d shows that the standard smoothing based on the WEnKF result fails for a low or high number of particles. As a matter of fact, particle trajectories are artificially corrected by the EnKF step at each observation time.



(a) Standard smoothing from PF ($N = 20$)

(b) Standard smoothing from PF ($N = 10000$)

(c) Standard smoothing from WEnKF ($N = 20$)

(d) Standard smoothing from WEnKF ($N = 10000$)

Figure 3. Standard smoothing from PF and WEnKF results. Thick line: hidden diffusion; dots: partial observations; dotted line: estimated filtering mean.

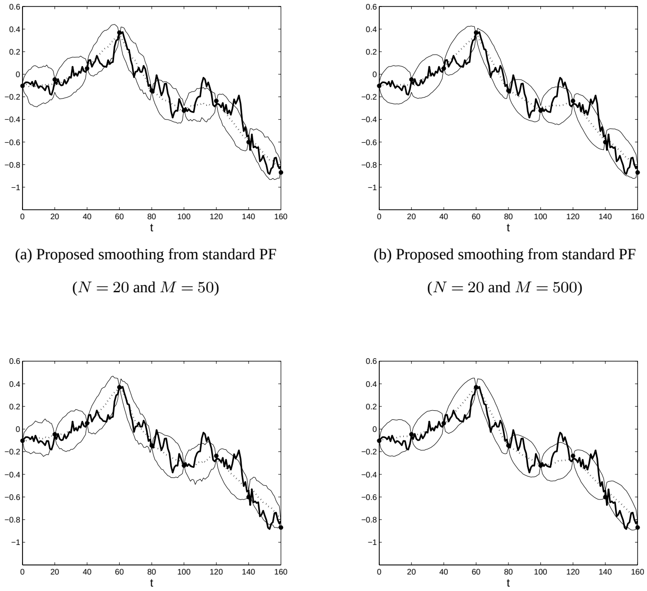
Resulting trajectories are highly non-plausible. Even for a huge number of particles, a smoothing based on those existing trajectories is not able to reduce the induced time discontinuities.

4.4 Proposed smoothing results

In this section, we show how the proposed method can improve the estimation of backward smoothing distributions when it is not adequate to rely on existing trajectories only. This is the case if the number of particles is too small, as demonstrated from the experiment presented in Fig. 3a, or if the existing trajectories do not correspond to plausible trajectories of the model (as shown for the WEnKF result in Fig. 3c–d).

Our smoothing is first applied using the filtering output of the standard particle filter with $N = 20$ particles. Figure 4a shows the result obtained with a sampling of $M = 50$ conditional trajectories between each pair $\{x_{t_k}^{(i)}, x_{t_{k+1}}^{(i)}\}$, $i = 1, \ldots, N$. The smoothing distribution $\hat{p}(x_t | y_{t_1:t_{k+1}})$ is computed from Eq. (18), so the smoothing mean is computed as $\sum_{i=1}^{N} w_{t_k}^{(i)} \sum_{j=1}^{M} \alpha(\tilde{x}^{(i)(j)}) \tilde{x}_t^{(i)(j)}$ for all $t \in]t_k, t_{k+1}]$, and similarly for the standard deviation. This result highlights the fact that since the proposed method creates new trajectories, it is able to correct the deficiencies of the standard smoothing approach presented in Fig. 3a when the initial number of filtering particles is too small. In Fig. 4b, the same experiment is presented using $M = 500$ conditional trajectories. In that case, the result is very similar to the reference particle smoothing result presented in Fig. 3b, obtained from a particle filter with $N = 10\,000$.

In parallel, the proposed smoothing has been tested using the output of the WEnKF filtering technique with $N = 20$ particles. Again, the smoothing is computed with $M = 50$ and $M = 500$ conditional trajectories, and the corresponding results are presented in Fig. 3c–d. Instead on relying on existing WEnKF trajectories that may not be plausible trajectories of the model (because of the EnKF correction step), the proposed method samples new trajectories between observation times. This leads to a good estimation of the smoothing distributions, contrary to the standard smoothing presented in Fig. 3c. Note that the smoothing results are very similar to the result obtained from the standard particle filter (Fig. 3a–b) because both filters have similar behavior at observation times.



(a) Proposed smoothing from standard PF
($N = 20$ and $M = 50$)

(b) Proposed smoothing from standard PF
($N = 20$ and $M = 500$)

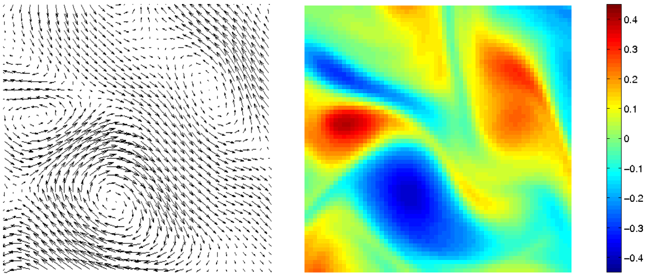
(c) Proposed smoothing from WEnKF
($N = 20$ and $M = 50$)

(d) Proposed smoothing from WEnKF
($N = 20$ and $M = 500$)

Figure 4. Proposed conditional smoothing result. Thick line: hidden diffusion; dots: partial observations; dotted line: estimated backward smoothing mean; thin line: estimated standard deviation.

5 Application to a high-dimensional assimilation problem

This section aims at illustrating the applicability of our method to a high-dimensional and nonlinear scenario, without extensive study at this stage. The method is applied to a turbulence assimilation problem, where the model of interest is of type Eq. (1). The goal is to recover temporal estimates of velocity/vorticity over a given spatial domain of size $n = 64 * 64$, from a sequence of noisy observations and a continuous a priori dynamical model based on a stochastic version of the Navier–Stokes equation. Within an environmental framework, a direct application would be the estimation of wind fields or sea surface currents from satellite data.



(a) (b)

Figure 5. State example. **(a)** Velocity field $\mathbf{w}_t$; **(b)** associated vorticity map $\boldsymbol{\xi}_t$.

5.1 State-space model

Let $\xi(\boldsymbol{x})$ denote the scalar vorticity at point $\boldsymbol{x}=(x,y)^T$, associated with the 2-D velocity $\boldsymbol{w}(\boldsymbol{x})=(w_x(\boldsymbol{x}),w_y(\boldsymbol{x}))^T$ through $\xi(\boldsymbol{x})=\frac{\partial w_y}{\partial x}-\frac{\partial w_x}{\partial y}$. Let $\boldsymbol{\xi}\in\mathbb{R}^n$ be the state vector describing the vorticity over an $n=64*64$ square domain, and $\boldsymbol{w}\in\mathbb{R}^{2n}$ the associated velocity field over the domain. We will focus on incompressible flows such that the divergence of the velocity field is null. A stochastic version of the Navier–Stokes equation in its velocity–vorticity form can then be written as

$$\mathrm{d}\boldsymbol{\xi}_t=-\nabla\boldsymbol{\xi}_t\cdot\boldsymbol{w}_t\mathrm{d}t+\frac{1}{Re}\Delta\boldsymbol{\xi}_t\mathrm{d}t+\sigma\mathrm{d}\boldsymbol{B}_t, \tag{21}$$

where Re denotes the flow Reynolds number ($Re=3000$). The uncertainty is modeled by a Brownian motion of size n, with covariance $\boldsymbol{\Sigma}=\sigma\sigma^T$, where $\sigma\in\mathbb{R}^n$. A velocity field example, generated from the model Eq. (21), is shown in Fig. 5a, together with the corresponding vorticity map (b).

We assume the hidden vorticity vector $\boldsymbol{\xi}$ is observed through noisy measurements $\boldsymbol{y}_{t_k}$ at discrete times t_k, where $t_k-t_{k-1}=100\Delta t$, and $\Delta t=0.1$ is the time step used to discretize Eq. (21). In our experimental setup, measurements correspond to PIV (particle image velocimetry) image sequences used in fluid mechanics applications. Note however that other kind of data can be used similarly within this state space model, like meteorological or oceanographic data for instance. The state and observation are related in our case through $\boldsymbol{y}_{t_k}=g(\boldsymbol{\xi}_{t_k})+\boldsymbol{\gamma}_{t_k}$, where g is a nonlinear function linking the vorticity to the image data, and $\boldsymbol{\gamma}_{t_k}$ is a Gaussian noise, uncorrelated in time.

5.2 Implementation details

We recall that the smoothing relies first on a particle filter step. Due to the high dimensionality of the state vector, the use of a standard particle filter is not adapted to solve the filtering problem, as discussed by Snyder et al. (2008) or Van Leeuwen (2009). We then make use of the method presented by Papadakis et al. (2010), which combines the benefits of the ensemble Kalman filter, known to perform well in practice for high-dimensional systems (Stroud et al., 2010), and the particle filter (which solves theoretically the true filtering problem, without approximating the filtering distributions with Gaussian distributions). Since the method of Papadakis et al. (2010) is intrinsically a particle filter, it then leads at each observation time t_k to a set of particles and weights $\{\boldsymbol{\xi}^{(i)}_{t_1:t_k},w_{t_k}\}_{i=1:N}$, as required by the algorithm proposed in Sect. 3.

The particle filter step requires simulations from the dynamical model (Eq. 21), and the conditional simulation step requires us to sample trajectories from its constrained version, which consists in a similar problem with modified drift (see process Eq. 9). The model is discretized in time with the time step $\Delta t=0.1$; more information about the discretization scheme may be obtained in Papadakis et al. (2010). The random perturbations are assumed to be realizations of Gaussian random fields that are correlated in space with the Gaussian covariance function $\boldsymbol{\Sigma}(\boldsymbol{x}_i,\boldsymbol{x}_j)=\eta\exp(-\frac{||\boldsymbol{x}_i-\boldsymbol{x}_j||^2}{\lambda})$, where $\eta=0.01$ and $\lambda=13$. In practice, the simulation of these perturbations is performed in Fourier space, with the method described in Evensen (2003).

Finally, the estimation of the smoothing distributions requires the computation of conditional trajectories weights, corresponding to Girsanov weights given by Eq. (11). After a Riemann sum approximation of the integral, the computation of weights requires the inversion of the matrix $\boldsymbol{\Sigma}$ of size (n,n), where $n=64*64$ is the number of grid points. We choose to compute $\boldsymbol{\Sigma}^{-1}$ empirically using a singular value decomposition computed from the M realizations of the perturbation fields used for the constrained trajectories simulations. Let $\mathbf{Z}$ be the matrix of size (n,M) containing the M-centered fields of size $n=64*64$; the SVD leads to $\mathbf{Z}=\mathbf{UDV}^T$, so that $\mathbf{ZZ}^T=\mathbf{UDD}^T\mathbf{U}^T$. The inverse of the covariance matrix $\boldsymbol{\Sigma}^{-1}$ is finally computed as

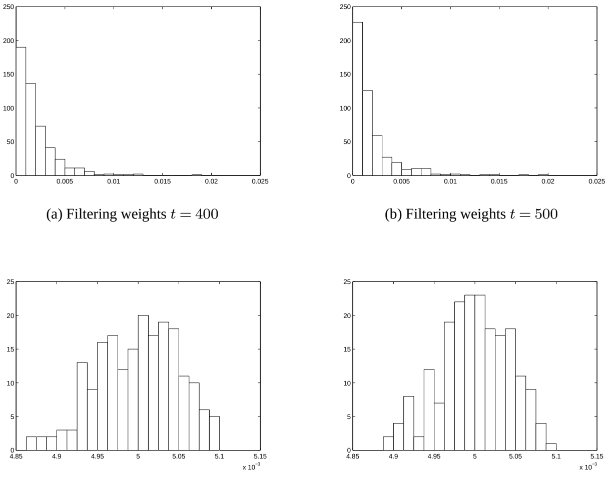
$$M(\mathbf{ZZ}^T)^{-1}=M\mathbf{U}(\mathbf{DD}^T)^{-1}\mathbf{U}^T, \tag{22}$$

which only requires the inversion of a diagonal. Note that more efficient procedures could be implemented in our case (homogeneous Gaussian covariance) since the covariance function is separable in the x and y directions. This means that the covariance matrix $\boldsymbol{\Sigma}$ can be written as the Kronecker product of smaller matrices and more easily inverted (Sun et al., 2012). However, the SVD inversion can be applied to any covariance structure, in particular it could deal with a non-homogeneous covariance matrix.

5.3 Results

In this section, we illustrate the capability of the proposed method to reduce the temporal discontinuities inherently introduced by the filtering in continuous–discrete state-space models.

The smoothing result relies on the output of the WEnKF filtering step, computed with $N=500$ particles. Compared



(a) Filtering weights $t = 400$

(b) Filtering weights $t = 500$

(d) Conditional simulation weights $t = 400$

(e) Conditional simulation weights $t = 500$

Figure 6. Filtering and conditional simulation weights.

to the size of the system, the number of particles is theoretically too small for the filter to be truly efficient. In practice, many filtering trajectories have close to zero weight at observation times. Histograms of filtering weights are given as illustration in Fig. 6a–b at two times $t = 400$ and $t = 500$. Note however that the filter is not degenerate and is able to provide results that get close to the hidden vorticity at measurement times. This can be observed in Fig. 7, where the mean square error is plotted with a full line, averaged at each time over the image domain of size $n = 64 * 64$. Since the ground truth vorticity sequence is known in our experimental setup, the mean square error is computed between the hidden vorticity and the estimated filtering mean, given by $\sum_{i=1}^{N} w_{t_k}^{(i)} \boldsymbol{\xi}_t^{(i)}$ for all $t \in [t_k, t_{k+1}[$. The correction steps lead to successive error decreases at observation times.

The proposed smoothing method has been applied with $M = 200$. Note that we take benefit from the fact that many filterings particles have close to zero weight. Indeed, the smoothing method relies in practice on a reduced number $\tilde{N}M$ of sampled conditional trajectories (with $\tilde{N} << N$), which makes the problem computationally tractable. In this experiment, we have retained around 5 % of initial filtering trajectories. The smoothing distribution $\hat{p}(\boldsymbol{\xi}_t | \boldsymbol{y}_{t_1:t_{k+1}})$ is computed for all $t \in]t_k, t_{k+1}]$ from Eq. (18), and its mean is computed as $\sum_{i=1}^{N} w_{t_{k+1}}^{(i)} \sum_{j=1}^{M} \alpha(\tilde{\boldsymbol{\xi}}^{(i)(j)}) \tilde{\boldsymbol{\xi}}_t^{(i)(j)}$. Histograms of conditional simulation weights $\alpha(\tilde{\boldsymbol{\xi}}^{(i)(j)})$ are given as an illustration in Fig. 6c–d for a given particle (i) at two times, $t = 400$ and $t = 500$.

The mean square error is computed between the true vorticity and the estimated smoothing mean, and plotted in Fig. 7 with a dotted line. As expected, the smoothing method reduces the error at hidden times between observations.

In addition, we present below a qualitative evaluation of the smoothing result for the same experiment, over a specific time interval.

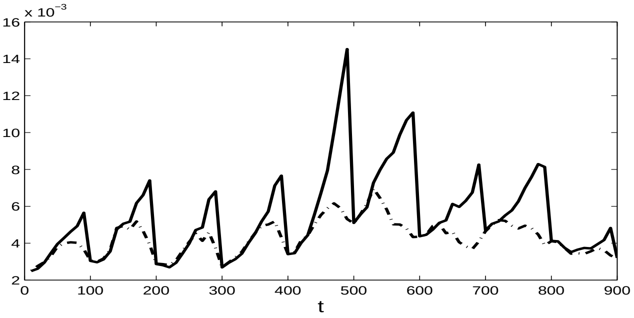


Figure 7. Full line: mean square error between ground truth vorticity and estimated filtering mean; dotted line: mean square error between ground truth vorticity and estimated backward smoothing mean.

The WEnKF result is first presented in Fig. 8 for the time interval $[400, 500]$ between two observations, where estimated mean vorticity maps are computed as $\sum_{i=1}^{N} w_{400}^{(i)} \boldsymbol{\xi}_t^{(i)}$ for all $t \in [400, 500[$, and as $\sum_{i=1}^{N} w_{500}^{(i)} \boldsymbol{\xi}_t^{(i)}$ for $t = 500$. The temporal discontinuity between estimations can be observed when reaching observation time $t = 500$: the vorticity map is suddenly modified in order to fit the observations, introducing inconsistencies in the vorticity temporal trajectories. Note that the application of the standard particle smoothing (described in Sect. 2.2) will fail here, and not only because the number of particles is too small. As a matter of fact, we recall that the filtering trajectories have been computed from the method presented in Papadakis et al. (2010), which uses the ensemble Kalman filter step as an importance distribution in the particle filter algorithm. The ensemble Kalman filter consists of a prediction step from the dynamical model Eq. (21), and a correction step that shifts particles towards the observation. Because of this correction step, the sampled filtering trajectories between two observation times do not correspond to trajectories of the dynamical model. This implies that from such a particle filter, the standard smoothing based on existing trajectories will not be able to reduce the temporal discontinuities observed in Fig. 8. This can be observed in Fig. 9, where smoothed vorticity maps are computed as $\sum_{i=1}^{N} w_{400}^{(i)} \boldsymbol{\xi}_t^{(i)}$ for $t = 400$, and as $\sum_{i=1}^{N} w_{500}^{(i)} \boldsymbol{\xi}_t^{(i)}$ for all $t \in]400, 500]$. The discontinuity at time $t = 500$ is still present.

The result obtained with the proposed method is plotted in Fig. 10. Estimated mean vorticity maps are computed as $\sum_{i=1}^{N} w_{500}^{(i)} \sum_{j=1}^{M} \alpha(\tilde{\boldsymbol{\xi}}^{(i)(j)}) \tilde{\boldsymbol{\xi}}_t^{(i)(j)}$ for all $t \in [400, 500]$. Spatio–temporal vorticity trajectories are gradually modified until observation time $t = 500$, preserving the fluid flow properties. As a matter of fact, since the proposed method samples new trajectories from the law of the physical process (Eq. 21), the smoothed vorticity trajectories are by construction consistent with the a priori dynamical model. In order to sample the smoothed trajectories, the method relies on the model and on filtering marginals at observation times,

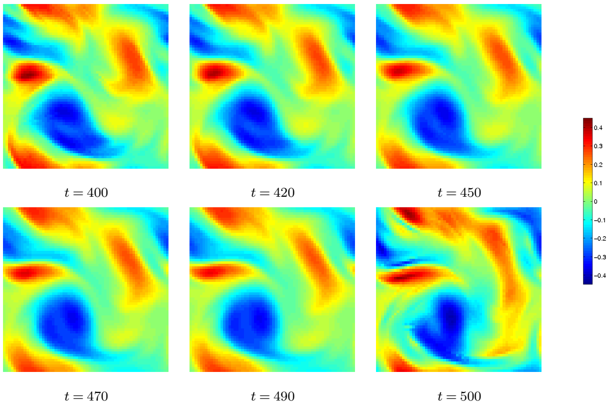


Figure 8. Filtering result with the method of Papadakis et al. (2010). Estimated mean vorticity maps for different times t between observation times $t = 400$ and $t = 500$.

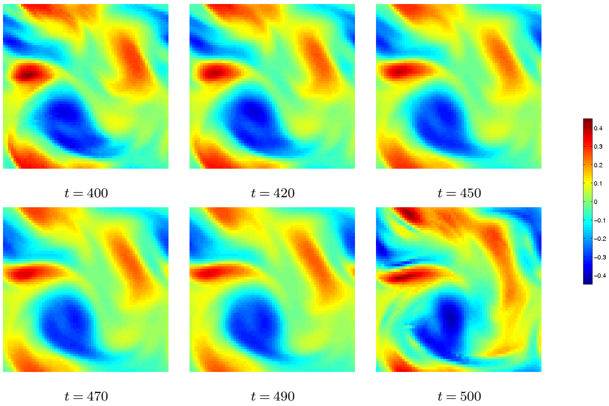


Figure 9. Standard particles smoothing result (see Sect. 2.2). Estimated mean vorticity maps for different times t between observation times $t = 400$ and $t = 500$.

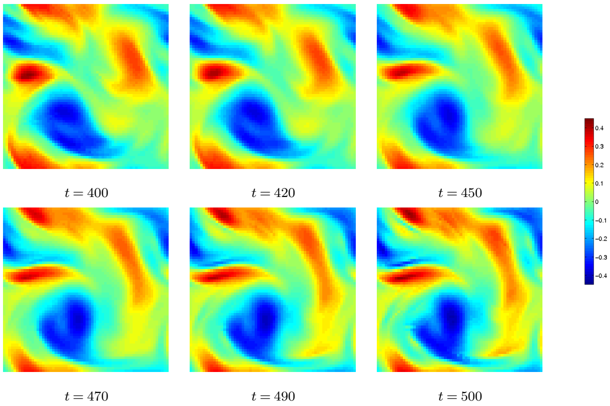


Figure 10. Smoothing result with the proposed method. Estimated mean vorticity maps for different times t between observation times $t = 400$ and $t = 500$.

but not on filtering trajectories at hidden times. It is then able to smooth the discontinuities inherent in the particle filtering technique we have used, contrary to the standard smoothing presented in Fig. 9.

6 Conclusion and discussion

In this paper we introduced a smoothing algorithm based on a conditional simulation technique of diffusions. The proposed smoothing is formulated as fixed-lag, in the sense that it is performed sequentially each time a new observation appears, in order to correct the state at hidden times up to the previous observation. Note that a decomposition similar to Eqs. (13) to (18) can be written from an integration up to a previous time t_{k-h}, with $h > 1$. This implies that the smoother can be formulated with a larger fixed lag, in order to correct the state backward not only up to the previous observation, but up to further measurement times. Yet, due to the successive resampling steps that have been performed in the filtering steps before time t_k, there are in practice only a few distinct filtering trajectories at times t_{k-h} if h is large. Consequently, the estimation of the joint law in Eq. (15) will not be reliable anymore if h is too large.

We have shown the practical applicability of the method to a high-dimensional problem. Nevertheless, the algorithm remains costly since a second Monte Carlo step is added to the Monte Carlo nature of particle filter algorithms. Yet, from an algorithmic point of view, the sequential nature of the proposed technique allows the smoothing to be implemented with a similar structure as filtering methods (sequential sampling and weighting of model trajectories). It is then easy to couple this smoothing to an operational filtering system and benefit from parallelization strategies, for instance.

Since the proposed smoothing uses the filtering result as input, it relies on the success of the underlying particle filter. For high-dimensional systems, a standard particle filter is not adapted and it is necessary to use filtering techniques that guide particles towards observations. In this paper, we use the WEnKF algorithm. In practice, any efficient particle filtering technique with such a guiding can be used within our framework. Note however that the construction of such techniques remains an open area of research.

We plan to work on the application of the smoothing method to a real high-dimensional case (for the estimation of sea surface currents from satellite image sequences). However, such a work will imply numerous difficulties which are not related to the smoothing technique but to the definition of the state-space model: definition of a suitable physical model, good characterization of state noise structure and model parameters. Therefore, this will be part of a future work.

Edited by: P. J. van Leeuwen

References

Beskos, A. and Roberts, G. O.: Exact simulation of diffusions, The Annal. Appl. Probabil., 15, 2422–2444, 2005.

Beskos, A., Papaspiliopoulos, O., Roberts, G. O., and Fearnhead, P.: Exact and computationally efficient likelihood-based estimation for discretely observed diffusion processes (with discussion), J. Roy. Stat. Soc. B, 68, 333–382, 2006.

Beyou, S., Cuzol, A., Gorthi, S., and Mémin, E.: Weighted Ensemble Transform Kalman Filter for Image Assimilation, Tellus A, 65, 18803, doi:10.3402/tellusa.v65i0.18803, 2013.

Briers, M., Doucet, A., and Maskell, S.: Smoothing algorithms for state-space models, Ann. Insti. Stat. Mathe., 62, 61–89, 2010.

Clark, J.: The simulation of pinned diffusions, in: Proceedings of the 29th IEEE Conference on Decision and Control, 1418–1420, 1990.

Del Moral, P.: Feynman-Kac Formulae. Genealogical and Interacting Particle Systems with Applications, Springer, 2004.

Del Moral, P., Jacod, J., and Protter, P.: The Monte Carlo Method for filtering with discrete-time observations, Probabil. Theor. Relat. Fields, 120, 346–368, 2001.

Delyon, B. and Hu, Y.: Simulation of conditioned diffusions and applications to parameter estimation, Stochast. Process. Applic., 116, 1660–1675, 2006.

Doucet, A., Godsill, S., and Andrieu, C.: On sequential Monte Carlo sampling methods for Bayesian filtering, Stat. Comput., 10, 197–208, 2000.

Durham, G. and Gallant, A.: Numerical techniques for maximum likelihood estimation of continuous-time diffusion processes, J. Business Econom. Stat., 20, 297–316, 2002.

Evensen, G.: The ensemble Kalman filter: theoretical formulation and practical implementation, Ocean Dynam., 53, 343–367, 2003.

Evensen, G. and van Leeuwen, P.: An ensemble Kalman Smoother for nonlinear dynamics, Mon. Weather Rev., 128, 1852–1867, 2000.

Fearnhead, P., Papaspiliopoulos, O., and Roberts, G.: Particle filters for partially observed diffusions, J. Roy. Stat. Soc. B, 70, 755–777, 2008.

Godsill, S. J., Doucet, A., and West, M.: Monte Carlo smoothing for nonlinear time series, J. Am. Stat. Assoc., 99, 156–168, 2004.

Gordon, N., Salmond, D., and Smith, A.: Novel approach to nonlinear/non-Gaussian Bayesian state estimation, IEEE Proc. F, 140, 107–113, 1993.

Papadakis, N., Mémin, E., Cuzol, A., and Gengembre, N.: Data assimilation with the weighted ensemble Kalman filter, Tellus A: Dynam. Meteorol. Oceanogr., 62, 673–697, 2010.

Snyder, C., Bengtsson, T., Bickel, P., and Anderson, J.: Obstacles to high-dimensional particle filtering, Mon. Weather Rev., 136, 4629–4640, 2008.

Stroud, J. R., Stein, M. L., Lesht, B. M., Schwab, D. J., and Beletsky, D.: An ensemble Kalman filter and smoother for satellite data assimilation, J. Am. Stat. Assoc., 105, 978–990, 2010.

Sun, Y., Bo, L., and Genton, M. G.: Geostatistics for large datasets, in: Advances and challenges in space-time modelling of natural events, 55–77, Springer, 2012.

Van Leeuwen, P. J.: Particle filtering in Geophysical systems, Mon. Weather Rev., 137, 4089–4114, 2009.

Van Leeuwen, P. J.: Nonlinear data assimilation in geosciences: an extremely efficient particle filter, Q. J. Roy. Meteorol. Soc., 136, 1991–1999, 2010.

Van Leeuwen, P. J. and Ades, M.: Efficient fully nonlinear data assimilation for geophysical fluid dynamics, Comput. Geosci., 55, 16–27, 2013.