Statistics and Computing (2020) 30:305–330
https://doi.org/10.1007/s11222-019-09884-y

Nudging the particle filter

Ömer Deniz Akyildiz[1,2] · Joaquín Míguez[3,4]

Received: 24 April 2018 / Accepted: 1 July 2019 / Published online: 13 July 2019

Abstract
We investigate a new sampling scheme aimed at improving the performance of particle filters whenever (a) there is a significant mismatch between the assumed model dynamics and the actual system, or (b) the posterior probability tends to concentrate in relatively small regions of the state space. The proposed scheme pushes some particles toward specific regions where the likelihood is expected to be high, an operation known as *nudging* in the geophysics literature. We reinterpret nudging in a form applicable to any particle filtering scheme, as it does not involve any changes in the rest of the algorithm. Since the particles are modified, but the importance weights do not account for this modification, the use of nudging leads to additional bias in the resulting estimators. However, we prove analytically that nudged particle filters can still attain asymptotic convergence with the same error rates as conventional particle methods. Simple analysis also yields an alternative interpretation of the nudging operation that explains its robustness to model errors. Finally, we show numerical results that illustrate the improvements that can be attained using the proposed scheme. In particular, we present nonlinear tracking examples with synthetic data and a model inference example using real-world financial data.

Keywords Particle filtering · Nudging · Robust filtering · Data assimilation · Model errors · Approximation errors.

1 Introduction

1.1 Background

State-space models (SSMs) are ubiquitous in many fields of science and engineering, including weather forecasting, mathematical finance, target tracking, machine learning, population dynamics, etc., where inferring the states of dynamical systems from data plays a key role.

A SSM comprises a pair of stochastic processes $(x_t)_{t\geq 0}$ and $(y_t)_{t\geq 1}$ called *signal process* and *observation process*, respectively. The conditional relations between these processes are defined with a transition and an observation model (also called *likelihood* model) where observations are conditionally independent given the signal process, and the latter is itself a Markov process. Given an observation sequence, $y_{1:t}$, the filtering problem in SSMs consists in the estimation of expectations with respect to the posterior probability distribution of the hidden states, conditional on $y_{1:t}$, which is also referred to as the filtering distribution.

Apart from a few special cases, neither the filtering distribution nor the integrals (or expectations) with respect to it can be computed exactly; hence, one needs to resort to numerical approximations of these quantities. Particle filters (PFs) have been a classical choice for this task since their introduction by Gordon et al. (1993); see also Kitagawa (1996), Liu and Chen (1998), Doucet et al. (2000, 2001). The PF constructs an empirical approximation of the posterior probability distribution via a set of Monte Carlo samples (usually termed *particles*) which are modified or killed sequentially as more data are taken into account. These samples are then used to

This work was partially supported by *Agencia Estatal de Investigación* of Spain (TEC2015-69868-C2-1-R ADVENTURE and RTI2018-099655-B-I00 CLARA), the Office of Naval Research (award no. N00014-19-1-2226), and the regional government of Madrid (program CASICAM-CM S2013/ICE-2845). Ö. D. A. is supported by the Lloyds Register Foundation programme on Data Centric Engineering through the London Air Quality project and supported by The Alan Turing Institute for Data Science and AI under EPSRC grant EP/N510129/1).

✉ Ömer Deniz Akyildiz
omer.akyildiz@warwick.ac.uk

1 University of Warwick, Coventry, UK

2 The Alan Turing Institute, London, UK

3 Universidad Carlos III de Madrid, Madrid, Spain

4 Instituto de Investigación Sanitaria Gregorio Marañón, Madrid, Spain

estimate the relevant expectations. The original form of the PF, often referred to as the bootstrap particle filter (BPF), has received significant attention due to its efficiency in a variety of problems, its intuitive appeal and its straightforward implementation. A large body of theoretical work concerning the BPF has also been compiled. For example, it has been proved that the expectations with respect to the empirical measures constructed by the BPF converge to the expectations with respect to the true posterior distributions when the number of particles is large enough (Del Moral and Guionnet 1999; Chopin 2004; Künsch 2005; Douc and Moulines 2008) or that they converge uniformly over time under additional assumptions related to the stability of the true distributions (Del Moral and Guionnet 2001; Del Moral 2004).

Despite the success of PFs in relatively low-dimensional settings, their use has been regarded impractical in models where $(x_t)_{t\geq 0}$ and $(y_t)_{t\geq 1}$ are sequences of high-dimensional random variables. In such scenarios, standard PFs have been shown to *collapse* (Bengtsson et al. 2008; Snyder et al. 2008). This problem has received significant attention from the data assimilation community. The high-dimensional models which are common in meteorology and other fields of Geophysics are often dealt with via an operation called *nudging* (Hoke and Anthes 1976; Malanotte-Rizzoli and Holland 1986, 1988; Zou et al. 1992). Within the particle filtering context, nudging can be defined as a transformation of the particles, which are pushed toward the observations using some observation-dependent map (van Leeuwen 2009, 2010; Ades and van Leeuwen 2013, 2015). If the dimensions of the observations and the hidden states are different, which is often the case, a gain matrix is computed in order to perform the nudging operation. In van Leeuwen (2009, 2010), Ades and van Leeuwen (2013, 2015), nudging is performed after the sampling step of the particle filter. The importance weights are then computed accordingly, so that they remain proper. Hence, nudging in this version amounts to a sophisticated choice of the importance function that generates the particles. It has been shown (numerically) that the schemes proposed by van Leeuwen (2009, 2010), Ades and van Leeuwen (2013, 2015) can track high-dimensional systems with a low number of particles. However, generating samples from the nudged proposal requires costly computations for each particle and the evaluation of weights becomes heavier as well. It is also unclear how to apply existing nudging schemes when non-Gaussianity and nontrivial nonlinearities are present in the observation model.

A related class of algorithms includes the so-called implicit particle filters (IPFs) (Chorin and Tu 2009; Chorin et al. 2010; Atkins et al. 2013). Similar to nudging schemes, IPFs rely on the principle of pushing particles to high-probability regions in order to prevent the collapse of the filter in high-dimensional state spaces. In a typical IPF, the region where particles should be generated is determined by solving an algebraic equation. This equation is model dependent, yet it can be solved for a variety of different cases (general procedures for finding solutions are given by Chorin and Tu 2009; Chorin et al. 2010). The fundamental principle underlying IPFs, moving the particles toward high-probability regions, is similar to nudging. Note, however, that unlike IPFs, nudging-based methods are not designed to *guarantee* that the resulting particles land on high-probability regions; it can be the case that nudged particles are moved to relatively low probability regions (at least occasionally). Since an IPF requires the solution of a model-dependent algebraic equation for every particle, it can be computationally costly, similar to the nudging methods by van Leeuwen (2009, 2010), Ades and van Leeuwen (2013, 2015). Moreover, it is not straightforward to derive the map for the translation of particles in general models; hence, the applicability of IPFs depends heavily on the specific model at hand.

1.2 Contribution

In this work, we propose a modification of the PF, termed *the nudged particle filter* (NuPF) and assess its performance in high-dimensional settings and with misspecified models. Although we use the same idea for nudging that is presented in the literature, our algorithm has subtle but crucial differences, as summarized below.

- First, we define the nudging step not just as a relaxation step toward observations but as a step that strictly increases the likelihood of a subset of particles. This definition paves the way for different nudging schemes, such as using the gradients of likelihoods or employing random search schemes to move around the state space. In particular, classical nudging (relaxation) operations arise as a special case of nudging using gradients when the likelihood is assumed to be Gaussian. Compared to IPFs, the nudging operation we propose is easier to implement as we only demand the likelihood to increase (rather than the posterior density). Indeed, nudging operators can be implemented in relatively straightforward forms, without the need to solve model-dependent equations.
- Second, unlike the other nudging-based PFs, we do not correct the bias induced by the nudging operation during the weighting step. Instead, we compute the weights in the same way they would be computed in a conventional (non-nudged) PF and the nudging step is devised to preserve the convergence rate of the PF, under mild standard assumptions, despite the bias. Moreover, computing biased weights is usually faster than computing proper (unbiased) weights. Depending on the choice of nudging scheme, the proposed algorithm can have an almost negligible computational overhead compared to the conventional PF from which it is derived.

– Finally, we show that a nudged PF for a given SSM (say $\mathcal{M}_0$) is equivalent to a standard BPF running on a modified dynamical model (denoted $\mathcal{M}_1$). In particular, model $\mathcal{M}_1$ is endowed with the same likelihood function as $\mathcal{M}_0$, but the transition kernel is observation driven in order to match the nudging operation. As a consequence, the implicit model $\mathcal{M}_1$ is "adapted to the data" and we have empirically found that, for any sufficiently long sequence $y_1, \ldots, y_t$, the evidence[1] (Robert 2007) in favor of $\mathcal{M}_1$ is greater than the evidence in favor of $\mathcal{M}_0$. We can show, for several examples, that this implicit adaptation to the data makes the NuPF robust to mismatches in the state equation of the SSM compared to conventional PFs. In particular, provided that the likelihoods are specified or calibrated reliably, we have found that NuPFs perform reliably under a certain amount of mismatch in the transition kernel of the SSM, while standard PFs degrade clearly in the same scenario.

In order to illustrate the contributions outlined above, we present the results of several computer experiments with both synthetic and real data. In the first example, we assess the performance of the NuPF when applied to a linear-Gaussian SSM. The aim of these computer simulations is to compare the estimation accuracy and the computational cost of the proposed scheme with several other competing algorithms, namely a standard BPF, a PF with optimal proposal function and a NuPF with proper weights. The fact that the underlying SSM is linear-Gaussian enables the computation of the optimal importance function (intractable in a general setting) and proper weights for the NuPF. We implement the latter scheme because of its similarity to standard nudging filters in the literature. This example shows that the NuPF suffers just from a slight performance degradation compared to the PF with optimal importance function or the NuPF with proper weights, while the latter two algorithms are computationally more demanding.

The second and third examples are aimed at testing the robustness of the NuPF when there is a significant misspecification in the state equation of the SSM. This is helpful in real-world applications because practitioners often have more control over measurement systems, which determine the likelihood, than they have over the state dynamics. We present computer simulation results for a stochastic Lorenz 63 model and a maneuvering target tracking problem.

In the fourth example, we present numerical results for a stochastic Lorenz 96 model, in order to show how a relatively high-dimensional system can be tracked without a major increase in the computational effort compared to the standard BPF. For this set of computer simulations, we have also compared the NuPF with the ensemble Kalman filter (EnKF), which is the de facto choice for tackling this type of systems.

Let us remark that, for the two stochastic Lorenz systems, the Markov kernel in the SSM can be sampled in a relatively straightforward way, yet transition probability densities cannot be computed (as they involve a sequence of noise variables mapped by a composition of nonlinear functions). Therefore, computing proper weights for proposal functions other than the Markov kernel itself is, in general, not possible for these examples.

Finally, we demonstrate the practical use of the NuPF on a problem where a real dataset is used to fit a stochastic volatility model using either particle Markov chain Monte Carlo (pMCMC) (Andrieu et al. 2010) or nested particle filters (Crisan and Miguez 2018).

1.3 Organization

The paper is structured as follows. After a brief note about notation, we describe the SSMs of interest and the BPF in Sect. 2. Then in Sect. 3, we outline the general algorithm and the specific nudging schemes we propose to use within the PF. We prove a convergence result in Sect. 4 which shows that the new algorithm has the same asymptotic convergence rate as the BPF. We also provide an alternative interpretation of the nudging operation that explains its robustness in scenarios where there is a mismatch between the observed data and the assumed SSM. We discuss the computer simulation experiments in Sect. 5 and present results for real data in Sect. 6. Finally, we make some concluding remarks in Sect. 7.

1.4 Notation

We denote the set of real numbers as $\mathbb{R}$, while $\mathbb{R}^d = \mathbb{R} \times \overset{d}{\cdots} \times \mathbb{R}$ is the space of d-dimensional real vectors. We denote the set of positive integers with $\mathbb{N}$ and the set of positive reals with $\mathbb{R}_+$. We represent the state space with $\mathsf{X} \subset \mathbb{R}^{d_x}$ and the observation space with $\mathsf{Y} \subset \mathbb{R}^{d_y}$.

In order to denote sequences, we use the shorthand notation $x_{i_1:i_2} = \{x_{i_1}, \ldots, x_{i_2}\}$. For sets of integers, we use $[n] = \{1, \ldots, n\}$. The p-norm of a vector $x \in \mathbb{R}^d$ is defined by $\|x\|_p = (x_1^p + \cdots + x_d^p)^{1/p}$. The L_p norm of a random variable z with probability density function (pdf) $p(z)$ is denoted $\|z\|_p = \left(\int |z|^p p(z) \mathrm{d}z\right)^{1/p}$, for $p \geq 1$. The Gaussian (normal) probability distribution with mean m and covariance matrix C is denoted $\mathcal{N}(m, C)$. We denote the identity matrix of dimension d with I_d.

The supremum norm of a real function $\varphi : \mathsf{X} \to \mathbb{R}$ is denoted $\|\varphi\|_\infty = \sup_{x \in \mathsf{X}} |\varphi(x)|$. A function is bounded if

[1] Given a dataset $\{y_1, \ldots, y_t\}$, the evidence in favor of a model $\mathcal{M}$ is the joint probability density of $y_1, \ldots, y_t$ conditional on $\mathcal{M}$, denoted $\mathsf{p}(y_{1:t}|\mathcal{M})$.

$\|\varphi\|_\infty < \infty$ and we indicate the space of real bounded functions $\mathsf{X} \to \mathbb{R}$ as $B(\mathsf{X})$. The set of probability measures on X is denoted $\mathcal{P}(\mathsf{X})$, the Borel σ-algebra of subsets of X is denoted $\mathcal{B}(\mathsf{X})$, and the integral of a function $\varphi : \mathsf{X} \to \mathbb{R}$ with respect to a measure μ on the measurable space $(\mathsf{X}, \mathcal{B}(\mathsf{X}))$ is denoted $(\varphi, \mu) := \int \varphi \mathrm{d}\mu$. The unit Dirac delta measure located at $x \in \mathbb{R}^d$ is denoted $\delta_x(\mathrm{d}x)$. The Monte Carlo approximation of a measure μ constructed using N samples is denoted as μ^N. Given a Markov kernel $\tau(\mathrm{d}x'|x)$ and a measure $\pi(\mathrm{d}x)$, we define the notation $\xi(\mathrm{d}x') = \tau\pi \triangleq \int \tau(\mathrm{d}x'|x)\pi(\mathrm{d}x)$.

2 Background

2.1 State-space models

We consider SSMs of the form

$$x_0 \sim \pi_0(\mathrm{d}x_0), \tag{2.1}$$

$$x_t|x_{t-1} \sim \tau_t(\mathrm{d}x_t|x_{t-1}), \tag{2.2}$$

$$y_t|x_t \sim g_t(y_t|x_t), \quad t \in \mathbb{N}, \tag{2.3}$$

where $x_t \in \mathsf{X}$ is the system state at time t, $y_t \in \mathsf{Y}$ is the tth observation, the measure π_0 describes the prior probability distribution of the initial state, τ_t is a Markov transition kernel on X and $g_t(y_t|x_t)$ is the (possibly non-normalized) pdf of the observation y_t conditional on the state x_t. We assume the observation sequence $\{y_t\}_{t\in\mathbb{N}_+}$ is arbitrary but fixed. Hence, it is convenient to think of the conditional pdf g_t as a likelihood function and we write $g_t(x_t) := g_t(y_t|x_t)$ for conciseness.

We are interested in the sequence of posterior probability distributions of the states generated by the SSM. To be specific, at each time $t = 1, 2, \ldots$ we aim at computing (or, at least, approximating) the probability measure π_t which describes the probability distribution of the state x_t conditional on the observation of the sequence $y_{1:t}$. When it exists, we use $\pi(x_t|y_{1:t})$ to denote the pdf of x_t given $y_{1:t}$ with respect to the Lebesgue measure, i.e., $\pi_t(\mathrm{d}x_t) = \pi(x_t|y_{1:t})\mathrm{d}x_t$.

The measure π_t is often termed the *optimal filter* at time t. It is closely related to the probability measure ξ_t, which describes the probability distribution of the state x_t conditional on $y_{1:t-1}$, and it is, therefore, termed the *predictive* measure at time t. As for the case of the optimal filter, we use $\xi(x_t|y_{1:t-1})$ to denote the pdf, with respect to the Lebesgue measure, of x_t given $y_{1:t-1}$.

2.2 Bootstrap particle filter

The BPF (Gordon et al. 1993) is a recursive algorithm that produces successive Monte Carlo approximations of ξ_t and π_t for $t = 1, 2, \ldots$. The method can be outlined as shown in Algorithm 1.

Algorithm 1 Bootstrap Particle Filter

1: Generate the initial particle system $\{x_0^{(i)}\}_{i=1}^N$ by drawing N times independently from the prior π_0.
2: **for** $t \geq 1$ **do**
3: Sampling: draw $\bar{x}_t^{(i)} \sim \tau_t(\mathrm{d}x_t|x_{t-1}^{(i)})$ independently for every $i = 1, \ldots, N$.
4: Weighting: compute $w_t^{(i)} = g_t(\bar{x}_t^{(i)})/\bar{Z}_t^N$ for every $i = 1, \ldots, N$, where $\bar{Z}_t^N = \sum_{i=1}^N g_t(\bar{x}_t^{(i)})$.
5: Resampling: draw $x_t^{(i)}$, $i = 1, \ldots, N$ from the discrete distribution $\sum_i w_t^{(i)} \delta_{\bar{x}_t^{(i)}}(\mathrm{d}x)$, independently for $i = 1, \ldots, N$.
6: **end for**

After an initialization stage, where a set of independent and identically distributed (i.i.d.) samples from the prior are drawn, it consists of three recursive steps which can be depicted as,

$$\pi_{t-1}^N \underbrace{\rightarrow}_{\text{sampling}} \xi_t^N \underbrace{\rightarrow}_{\text{weighting}} \tilde{\pi}_t^N \underbrace{\rightarrow}_{\text{resampling}} \pi_t^N. \tag{2.4}$$

Given a Monte Carlo approximation $\pi_{t-1}^N = \frac{1}{N}\sum_{i=1}^N \delta_{x_{t-1}^{(i)}}$ computed at time $t-1$, the sampling step yields an approximation of the predictive measure ξ_t of the form

$$\xi_t^N = \frac{1}{N}\sum_{i=1}^N \delta_{\bar{x}_t^{(i)}}$$

by propagating the *particles* $\{x_{t-1}^{(i)}\}_{i=1}^N$ via the Markov kernel $\tau_t(\cdot|x_{t-1}^{(i)})$. The observation y_t is assimilated via the importance weights $w_t^{(i)} \propto g_t(x_t^{(i)})$, to obtain the approximate filter

$$\tilde{\pi}_t^N = \sum_{i=1}^N w_t^{(i)} \delta_{\bar{x}_t^{(i)}},$$

and the resampling step produces a set of unweighted particles that completes the recursive loop and yields the approximation

$$\pi_t^N = \frac{1}{N}\sum_{i=1}^N \delta_{x_t^{(i)}}.$$

The random measures ξ_t^N, $\tilde{\pi}_t^N$ and π_t^N are commonly used to estimate *a posteriori* expectations conditional on the available observations. For example, if φ is a function $\mathsf{X} \to \mathbb{R}$, then the expectation of the random variable $\varphi(x_t)$ conditional on $y_{1:t-1}$ is $\mathbb{E}\left[\varphi(x_t)|y_{1:t-1}\right] = (\varphi, \xi_t)$. The latter integral can be approximated using ξ_t^N, namely

$$(\varphi, \xi_t) = \int \varphi(x_t)\xi_t(\mathrm{d}x_t) \approx (\varphi, \xi_t^N)$$

$$= \int \varphi(x_t)\xi_t^N(\mathrm{d}x_t) = \frac{1}{N}\sum_{i=1}^N \varphi(\bar{x}_t^{(i)}).$$

Similarly, we can have estimators $(\varphi, \tilde{\pi}_t^N) \approx (\varphi, \pi_t)$ and $(\varphi, \pi_t^N) \approx (\varphi, \pi_t)$. Classical convergence results are usually proved for real bounded functions, e.g., if $\varphi \in B(\mathsf{X})$ then

$$\lim_{N\to\infty} |(\varphi, \pi_t) - (\varphi, \pi_t^N)| = 0 \quad \text{almost surely (a.s.)}$$

under mild assumptions; see Del Moral (2004), Bain and Crisan (2009) and references therein.

The BPF can be generalized by using arbitrary proposal pdf's $q_t(x_t|x_{t-1}^{(i)}, y_t)$, possibly observation dependent, instead of the Markov kernel $\tau_t(\cdot|x_{t-1}^{(i)})$ in order to generate the particles $\{\bar{x}_t^{(i)}\}_{i=1}^N$ in the sampling step. This can lead to more efficient algorithms, but the weight computation has to account for the new proposal and we obtain (Doucet et al. 2000)

$$w_t^{(i)} \propto \frac{g_t\left(\bar{x}_t^{(i)}\right) \tau_t\left(\bar{x}_t^{(i)}|x_t^{(i)}\right)}{q_t\left(\bar{x}_t^{(i)}|x_{t-1}^{(i)}, y_t\right)}, \tag{2.5}$$

which can be more costly to evaluate. This issue is related to the nudged PF to be introduced in Sect. 3, which can be interpreted as a scheme to choose a certain observation-dependent proposal $q_t(x_t|x_{t-1}^{(i)}, y_t)$. However, the new method does not require that the weights be computed as in (2.5) in order to ensure convergence of the estimators.

3 Nudged particle filter

3.1 General algorithm

Compared to the standard BPF, the nudged particle filter (NuPF) incorporates one additional step right after the sampling of the particles $\{\bar{x}_t^{(i)}\}_{i=1}^N$ at time t. The schematic depiction of the BPF in (2.4) now becomes

$$\pi_{t-1}^N \underbrace{\rightarrow}_{\text{sampling}} \xi_t^N \underbrace{\rightarrow}_{\text{nudging}} \tilde{\xi}_t^N \underbrace{\rightarrow}_{\text{weighting}} \tilde{\pi}_t^N \underbrace{\rightarrow}_{\text{resampling}} \pi_t^N, \tag{3.1}$$

where the new *nudging step* intuitively consists in pushing a subset of the generated particles $\{\bar{x}_t^{(i)}\}_{i=1}^N$ toward regions of the state space X where the likelihood function $g_t(x)$ takes higher values.

When considered jointly, the sampling and nudging steps in (3.1) can be seen as sampling from a proposal distribution which is obtained by modifying the kernel $\tau_t(\cdot|x_{t-1})$ in a way that depends on the observation y_t. Indeed, this is the classical view of nudging in the literature (van Leeuwen 2009, 2010; Ades and van Leeuwen 2013, 2015). However, unlike in this classical approach, here the weighting step does not account for the effect of nudging. In the proposed NuPF, the weights are kept the same as in the original filter, $w_t^{(i)} \propto g_t(x_t^{(i)})$. In doing so, we save computations but, at the same time, introduce bias in the Monte Carlo estimators. One of the contributions of this paper is to show that this bias can be controlled using simple design rules for the nudging step, while practical performance can be improved at the same time.

In order to provide an explicit description of the NuPF, let us first state a definition for the nudging step.

Definition 1 A nudging operator $\alpha_t^{y_t} : \mathsf{X} \to \mathsf{X}$ associated with the likelihood function $g_t(x)$ is a map such that

$$\text{if} \quad x' = \alpha_t^{y_t}(x) \quad \text{then} \quad g_t(x') \geq g_t(x) \tag{3.2}$$

for every $x, x' \in \mathsf{X}$.

Intuitively, we define nudging herein as an operation that increases the likelihood. There are several ways in which this can be achieved and we discuss some examples in Sects. 3.2 and 3.3. The NuPF with nudging operator $\alpha_t^{y_t} : \mathsf{X} \to \mathsf{X}$ is outlined in Algorithm 2.

Algorithm 2 Nudged Particle Filter (NuPF)

1: Generate the initial particle system $\{x_0^{(i)}\}_{i=1}^N$ by drawing N times independently from the prior π_0.
2: **for** $t \geq 1$ **do**
3: Sampling: draw $\bar{x}_t^{(i)} \sim \tau_t(\mathrm{d}x_t|x_{t-1}^{(i)})$ independently for every $i = 1, \ldots, N$.
4: **Nudging**: choose a set of indices $\mathcal{I}_t \subset [N]$, then compute $\tilde{x}_t^{(i)} = \alpha_t^{y_t}(\bar{x}_t^{(i)})$ for every $i \in \mathcal{I}_t$. Keep $\tilde{x}_t^{(i)} = \bar{x}_t^{(i)}$ for every $i \in [N]\backslash\mathcal{I}_t$.
5: Weighting: compute $w_t^{(i)} = g_t(\tilde{x}_t^{(i)})/\tilde{Z}_t^N$ for every $i = 1, \ldots, N$, where $\tilde{Z}_t^N = \sum_{i=1}^N g(\tilde{x}_t^{(i)})$.
6: Resample: draw $x_t^{(i)}$ from $\sum_i w_t^{(i)} \delta_{\tilde{x}_t^{(i)}}(\mathrm{d}x)$ independently for $i = 1, \ldots, N$.
7: **end for**

It can be seen that the nudging operation is implemented in two stages.

- First, we choose a set of indices $\mathcal{I}_t \subset [N]$ that identifies the particles to be nudged. Let $M = |\mathcal{I}_t|$ denote the number of elements in $\mathcal{I}_t$. We prove in Sect. 4 that keeping $M \leq \mathcal{O}(\sqrt{N})$ allows the NuPF to converge with the same error rates $\mathcal{O}(1/\sqrt{N})$ as the BPF. In Sect. 3.2, we discuss two simple methods to build $\mathcal{I}_t$ in practice.
- Second, we choose an operator $\alpha_t^{y_t}$ that guarantees an increase in the likelihood of any particle. We discuss different implementations of $\alpha_t^{y_t}$ in Sect. 3.3.

We devote the rest of this section to a discussion of how these two steps can be implemented (in several ways).

3.2 Selection of particles to be nudged

The set of indices $\mathcal{I}_t$, which identifies the particles to be nudged in Algorithm 2, can be constructed in several different ways, either random or deterministic. In this paper, we describe two simple random procedures with little computational overhead.

- *Batch nudging*. Let the number of nudged particles M be fixed. A simple way to construct $\mathcal{I}_t$ is to draw indices $i_1, i_2, \ldots, i_M$ uniformly from $[N]$ without replacement, and then let $\mathcal{I}_t = i_{1:M}$. We refer to this scheme as *batch nudging*, referring to selection of the indices at once. One advantage of this scheme is that the number of particles to be nudged, M, is deterministic and can be set a priori.
- *Independent nudging*. The size and the elements of $\mathcal{I}_t$ can also be selected randomly in a number of ways. Here, we have studied a procedure in which, for each index $i = 1, \ldots, N$, we assign $i \in \mathcal{I}_t$ with probability $\frac{M}{N}$. In this way, the actual cardinality $|\mathcal{I}_t|$ is random, but its expected value is exactly M. This procedure is particularly suitable for parallel implementations, since each index can be assigned to $\mathcal{I}_t$ (or not) at the same time as all others.

3.3 How to nudge

The nudging step is aimed at increasing the likelihood of a subset of individual particles, namely those with indices contained in $\mathcal{I}_t$. Therefore, any map $\alpha_t^{y_t} : \mathsf{X} \to \mathsf{X}$ such that $(g_t \circ \alpha_t^{y_t})(x) \geq g_t(x)$ when $x \in \mathsf{X}$ is a valid nudging operator. Typical procedures used for optimization, such as gradient moves or random search schemes, can be easily adapted to implement (relatively) inexpensive nudging steps. Here we briefly describe a few of such techniques.

- *Gradient nudging*. If $g_t(x_t)$ is a differentiable function of x_t, one straightforward way to nudge particles is to take gradient steps. In Algorithm 3, we show a simple procedure with one gradient step alone, and where $\gamma_t > 0$ is a step-size parameter and $\nabla_x g_t(x)$ denotes the vector of partial derivatives of g_t with respect to the state variables, i.e.,

$$\nabla_{x_t} g_t = \begin{bmatrix} \frac{\partial g_t}{\partial x_{1,t}} \\ \frac{\partial g_t}{\partial x_{2,t}} \\ \vdots \\ \frac{\partial g_t}{\partial x_{d_x,t}} \end{bmatrix} \quad \text{for} \quad x_t = \begin{bmatrix} x_{1,t} \\ x_{2,t} \\ \vdots \\ x_{d_x,t} \end{bmatrix} \in \mathsf{X}.$$

  Algorithms can obviously be designed where nudging involves several gradient steps. In this work, we limit our study to the single-step case, which is shown to be effective and keeps the computational overhead to a minimum. We also note that the performance of gradient nudging can be sensitive to the choice of the step-size parameters $\gamma_t > 0$, which are, in turn, model dependent.[2]
- *Random nudging*. Gradient-free techniques inherited from the field of global optimization can also be employed in order to push particles toward regions where they have higher likelihoods. A simple stochastic-search technique adapted to the nudging framework is shown in Algorithm 4. We hereafter refer to the latter scheme as random search nudging.
- *Model-specific nudging*. Particles can also be nudged using the specific model information. For instance, in some applications the state vector x_t can be split into two subvectors, x_t^{obs} and x_t^{unobs} (observed and unobserved, respectively), such that $g_t(x_t) = g_t(x_t^{\text{obs}})$, i.e., the likelihood depends only on x_t^{obs} and not on x_t^{unobs}. If the relationship between x_t^{obs} and x_t^{unobs} is tractable, one can first nudge x_t^{obs} in order to increase the likelihood and then modify x_t^{unobs} in order to keep it coherent with x_t^{obs}. A typical example of this kind arises in object tracking problems, where positions and velocities have a special and simple physical relationship, but usually only position variables are observed through a linear or nonlinear transformation. In this case, nudging would only affect the position variables. However, using these position variables, one can also nudge velocity variables with simple rules. We discuss this idea and show numerical results in Sect. 5.

Algorithm 3 Gradient nudging

1: **for** every $i \in \mathcal{I}_t$ **do**

$$\tilde{x}_t^{(i)} = \bar{x}_t^{(i)} + \gamma_t \nabla_{x_t} g_t(\bar{x}_t^{(i)})$$

2: **end for**

Algorithm 4 Random search nudging

1: **repeat**
2: Generate $\tilde{x}_t^{(i)} = \bar{x}_t^{(i)} + \eta_t$ where $\eta_t \sim \mathcal{N}(0, C)$ for some covariance matrix C.
3: If $g_t(\tilde{x}_t^{(i)}) > g_t(\bar{x}_t^{(i)})$ then keep $\tilde{x}_t^{(i)}$, otherwise set $\tilde{x}_t^{(i)} = \bar{x}_t^{(i)}$.
4: **until** the particle is nudged.

3.4 Nudging general particle filters

In this paper, we limit our presentation to BPFs in order to focus on the key concepts of nudging and to ease presentation. It should be apparent, however, that nudging steps

[2] We have found, nevertheless, that fixed step sizes (i.e., $\gamma_t = \gamma$ for all t) work well in practice for the examples of Sects. 5 and 6.

can be plugged into general PFs. More specifically, since the nudging step is algorithmically detached from the sampling and weighting steps, it can be easily used within any PF, even if it relies on different proposals and different weighting schemes. We leave for future work the investigation of the performance of nudging within widely used PFs, such as auxiliary particle filters (APFs) (Pitt and Shephard 1999).

4 Analysis

The nudging step modifies the random generation of particles in a way that is not compensated by the importance weights. Therefore, we can expect nudging to introduce bias in the resulting estimators in general. However, in Sect. 4.1 we prove that, as long as some basic guidelines are followed, the estimators of integrals with respect to the filtering measure π_t and the predictive measure ξ_t converge in L_p as $N \to \infty$ with the usual Monte Carlo rate $\mathcal{O}(1/\sqrt{N})$. The analysis is based on a simple induction argument and ensures the consistency of a broad class of estimators. In Sect. 4.2, we briefly comment on the conditions needed to guarantee that convergence is attained uniformly over time. We do not provide a full proof, but this can be done by extending the classical arguments in Del Moral and Guionnet (2001) or Del Moral (2004) and using the same treatment of the nudging step as in the induction proof of Sect. 4.1. Finally, in Sect. 4.3, we provide an interpretation of nudging in a scenario with modeling errors. In particular, we show that the NuPF can be seen as a *standard* BPF for a modified dynamical model which is "a better fit" for the available data than the original SSM.

4.1 Convergence in L_p

The goal in this section is to provide theoretical guarantees of convergence for the NuPF under mild assumptions. First, we analyze a general NuPF (with arbitrary nudging operator $\alpha_t^{y_t}$ and an upper bound on the size M of the index set $\mathcal{I}_t$) and then we provide a result for a NuPF with gradient nudging.

Before proceeding with the analysis, let us note that the NuPF produces several approximate measures, depending on the set of particles (and weights) used to construct them. After the sampling step, we have the random probability measure

$$\xi_t^N = \frac{1}{N}\sum_{i=1}^{N} \delta_{\tilde{x}_t^{(i)}}, \tag{4.1}$$

which converts into

$$\tilde{\xi}_t^N = \frac{1}{N}\sum_{i=1}^{N} \delta_{\tilde{x}_t^{(i)}} \tag{4.2}$$

after nudging. Once the weights $w_t^{(i)}$ are computed, we obtain the approximate filter

$$\tilde{\pi}_t^N = \sum_{i=1}^{N} w_t^{(i)} \delta_{\tilde{x}_t^{(i)}}, \tag{4.3}$$

which finally yields

$$\pi_t^N = \frac{1}{N}\sum_{i=1}^{N} \delta_{x_t^{(i)}} \tag{4.4}$$

after the resampling step.

Similar to the BPF, simple Assumption 1 stated next is sufficient for consistency and to obtain explicit error rates (Del Moral and Miclo 2000; Crisan and Doucet 2002; Míguez et al. 2013) for the NuPF, as stated in Theorem 1.

Assumption 1 The likelihood function is positive and bounded, i.e.,

$$g_t(x_t) > 0 \quad \text{and} \quad \|g_t\|_\infty = \sup_{x_t \in \mathsf{X}} |g_t(x_t)| < \infty$$

for $t = 1, \ldots, T$.

Theorem 1 *Let $y_{1:T}$ be an arbitrary but fixed sequence of observations, with $T < \infty$, and choose any $M \leq \sqrt{N}$ and any map $\alpha_t^{y_t} : \mathsf{X} \to \mathsf{X}$. If Assumption 1 is satisfied and $|\mathcal{I}_t| = M$, then*

$$\|(\varphi, \pi_t^N) - (\varphi, \pi_t)\|_p \leq \frac{c_{t,p}\|\varphi\|_\infty}{\sqrt{N}} \tag{4.5}$$

for every $t = 1, 2, \ldots, T$, any $\varphi \in B(\mathsf{X})$, any $p \geq 1$ and some constant $c_{t,p} < \infty$ independent of N.

See "Appendix A" for a proof.

Theorem 1 is very general; it actually holds for any map $\alpha_t^{y_t} : \mathsf{X} \to \mathsf{X}$, i.e., not necessarily a nudging operator. We can also obtain error rates for specific choices of the nudging scheme. A simple, yet practically appealing, setup is the combination of batch and gradient nudging, as described in Sects. 3.2 and 3.3, respectively.

Assumption 2 The gradient of the likelihood is bounded. In particular, there are constants $G_t < \infty$ such that

$$\|\nabla_x g_t(x)\|_2 \leq G_t < \infty$$

for every $x \in \mathsf{X}$ and $t = 1, 2, \ldots, T$.

Lemma 1 *Choose the number of nudged particles, $M > 0$, and a sequence of step sizes, $\gamma_t > 0$, in such a way that $\sup_{1 \leq t \leq T} \gamma_t M \leq \sqrt{N}$ for some $T < 0$. If Assumption 2 holds and φ is a Lipschitz test function; then the error introduced*

by the batch gradient-nudging step with $|\mathcal{I}_t| = M$ can be bounded as,

$$\left\| (\varphi, \xi_t^N) - (\varphi, \tilde{\xi}_t^N) \right\|_p \leq \frac{LG_t}{\sqrt{N}},$$

where L is the Lipschitz constant of φ, for every $t = 1, \ldots, T$.

See "Appendix B" for a proof.

It is straightforward to apply Lemma 1 to prove convergence of the NuPF with a batch gradient-nudging step. Specifically, we have the following result.

Theorem 2 *Let $y_{1:T}$ be an arbitrary but fixed sequence of observations, with $T < \infty$, and choose a sequence of step sizes $\gamma_t > 0$ and an integer M such that*

$$\sup_{1 \leq t \leq T} \gamma_t M \leq \sqrt{N}.$$

Let π_t^N denote the filter approximation obtained with a NuPF with batch gradient nudging. If Assumptions 1 and 2 are satisfied and $|\mathcal{I}_t| = M$, then

$$\|(\varphi, \pi_t^N) - (\varphi, \pi_t)\|_p \leq \frac{c_{t,p}\|\varphi\|_\infty}{\sqrt{N}} \tag{4.6}$$

for every $t = 1, 2, \ldots, T$, any bounded Lipschitz function φ, some constant $c_{t,p} < \infty$ independent of N for any integer $p \geq 1$.

The proof is straightforward (using the same argument as in the proof of Theorem 1 combined with Lemma 1), and we omit it here. We note that Lemma 1 provides a guideline for the choice of M and γ_t. In particular, one can select $M = N^\beta$, where $0 < \beta < 1$, together with $\gamma_t \leq N^{\frac{1}{2}-\beta}$ in order to ensure that $\gamma_t M \leq \sqrt{N}$. Actually, it would be sufficient to set $\gamma_t \leq CN^{\frac{1}{2}-\beta}$ for some constant $C < \infty$ in order to keep the same error rate (albeit with a different constant in the numerator of the bound). Therefore, Lemma 1 provides a heuristic to balance the step size with the number of nudged particles.[3] We can increase the number of nudged particles, but in that case we need to shrink the step size accordingly, so as to keep $\gamma_t M \leq \sqrt{N}$. Similar results can be obtained using the gradient of the log-likelihood, $\log g_t$, if the g_t comes from the exponential family of densities.

4.2 Uniform convergence

Uniform convergence can be proved for the NuPF under the same standard assumptions as for the conventional BPF; see, e.g., Del Moral and Guionnet (2001), Del Moral (2004). The latter can be summarized as follows (Del Moral 2004):

(i) The likelihood function is bounded and bounded away from zero, i.e., $g_t \in B(\mathsf{X})$, and there is some constant $a > 0$ such that $\inf_{t>0, x \in \mathsf{X}} g_t(x) \geq a$.
(ii) The kernel mixes sufficiently well, namely, for any given integer m there is a constant $0 < \varepsilon < 1$ such that

$$\inf_{t>0;(x,x') \in \mathsf{X}^2} \frac{\tau_{t+m|t}(A|x)}{\tau_{t+m|t}(A|x')} > \varepsilon$$

for any Borel set A, where $\tau_{t+m|t}$ is the composition of the kernels $\tau_{t+m} \circ \tau_{t+m-1} \circ \cdots \circ \tau_t$.

When (i) and (ii) above hold, the sequence of optimal filters $\{\pi_t\}_{t \geq 0}$ is stable and it can be proved that

$$\sup_{t>0} \|(\varphi, \pi_t) - (\varphi, \pi_t^N)\|_p \leq \frac{c_p}{\sqrt{N}}$$

for any bounded function $\varphi \in B(\mathsf{X})$, where $c_p < \infty$ is constant with respect to N and t and π_t^N is the particle approximation produced by either the NuPF (as in Theorem 1 or, provided $\sup_{t>0} G_t < \infty$, as in Theorem 2) or the BPF algorithms. We skip a formal proof as, again, it is straightforward combination of the standard argument by Del Moral (2004) (see also, e.g., Oreshkin and Coates 2011; Crisan and Miguez 2017) with the same handling of the nudging operator as in the proofs of Theorem 1 or Lemma 1 .

4.3 Nudging as a modified dynamical model

We have found in computer simulation experiments that the NuPF is consistently more robust to model errors than the conventional BPF. In order to obtain some analytical insight of this scenario, in this section we reinterpret the NuPF as a standard BPF for a modified, observation-driven dynamical model and discuss why this modified model can be expected to be a better fit for the given data than the original SSM. In this way, the NuPF can be seen as an automatic adaptation of the underlying model to the available data.

The dynamic models of interest in stochastic filtering can be defined by a prior measure τ_0, the transition kernels τ_t and the likelihood functions $g_t(x) = g_t(y_t|x)$, for $t \geq 1$. In this section, we write the latter as $g_t^{y_t}(x) = g_t(y_t|x)$, in order to emphasize that g_t is parametrized by the observation y_t, and we also assume that every $g_t^{y_t}$ is a normalized pdf in y_t for the sake of clarity. Hence, we can formally represent the SSM defined by (2.1), (2.2) and (2.3) as $\mathcal{M}_0 = \{\tau_0, \tau_t, g_t^{y_t}\}$.

[3] Note that the step sizes may have to be kept small enough to ensure that $g_t(\tilde{x}_t^{(i)} + \gamma_t \nabla_x g_t(\tilde{x}_t^{(i)})) \geq g_t(\tilde{x}_t^{(i)})$, so that proper nudging, according to Definition 1, is performed.

Now, let us assume $y_{1:T}$ to be fixed and construct the alternative dynamical model $\mathcal{M}_1 = \{\tau_0, \tilde{\tau}_t^{y_t}, g_t^{y_t}\}$, where

$$\tilde{\tau}_t^{y_t}(\mathrm{d}x_t|x_{t-1}) := (1-\varepsilon_M)\tau_t(\mathrm{d}x_t|x_{t-1}) + \varepsilon_M \int \delta_{\alpha_t^{y_t}(\bar{x}_t)}(\mathrm{d}x_t)\tau_t(\mathrm{d}\bar{x}_t|x_{t-1}) \quad (4.7)$$

is an observation-driven transition kernel, $\varepsilon_M = \frac{M}{N}$ and the nudging operator $\alpha_t^{y_t}$ is a one-to-one map that depends on the (fixed) observation y_t. We note that the kernel $\tilde{\tau}_t^{y_t}$ jointly represents the Markov transition induced by the original kernel τ_t followed by an independent nudging transformation (namely, each particle is independently nudged with probability ε_M). As a consequence, the standard BPF for model $\mathcal{M}_1$ coincides exactly with a NuPF for model $\mathcal{M}_0$ with independent nudging and operator $\alpha_t^{y_t}$. Indeed, according to the definition of $\tilde{\tau}_t^{y_t}$ in (4.7), generating a sample $\tilde{x}_t^{(i)}$ from $\tilde{\tau}_t^{y_t}(\mathrm{d}x_t|x_{t-1}^{(i)})$ is a three-step process where

- We first draw $\bar{x}_t^{(i)}$ from $\tau_t(\mathrm{d}x_t|x_{t-1}^{(i)})$,
- Then generate a sample $u_t^{(i)}$ from the uniform distribution $\mathcal{U}(0,1)$, and
- If $u_t^{(i)} < \varepsilon_M$, then we set $\tilde{x}_t^{(i)} = \alpha_t^{y_t}(\bar{x}_t^{(i)})$, else we set $\tilde{x}_t^{(i)} = \bar{x}_t^{(i)}$.

After sampling, the importance weight for the BPF applied to model $\mathcal{M}_1$ is $w_t^{(i)} \propto g_t^{y_t}(\tilde{x}_t^{(i)})$. This is exactly the same procedure as in the NuPF applied to the original SSM $\mathcal{M}_0$ (see Algorithm 2).

Intuitively, one can expect that the observation-driven $\mathcal{M}_1$ is a better fit for the data sequence $y_{1:T}$ than the original model $\mathcal{M}_0$. Within the Bayesian methodology, a common approach to compare two competing probabilistic models ($\mathcal{M}_0$ and $\mathcal{M}_1$ in this case) for a given dataset $y_{1:t}$ is to evaluate the so-called *model evidence* (Bernardo and Smith 1994) for both $\mathcal{M}_0$ and $\mathcal{M}_1$.

Definition 2 The evidence (or likelihood) of a probabilistic model $\mathcal{M}$ for a given dataset $y_{1:t}$ is the probability density of the data conditional on the model that we denote as $\mathsf{p}(y_{1:t}|\mathcal{M})$.

We say that $\mathcal{M}_1$ is a better fit than $\mathcal{M}_0$ for the dataset $y_{1:t}$ when $\mathsf{p}(y_{1:t}|\mathcal{M}_1) > \mathsf{p}(y_{1:t}|\mathcal{M}_0)$. Since

$$\mathsf{p}(y_{1:t}|\mathcal{M}_0) = \int \cdots \int \prod_{l=1}^{t} g_l(x_l)\tau_l(\mathrm{d}x_l|x_{l-1})\tau_0(\mathrm{d}x_0),$$

and

$$\mathsf{p}(y_{1:t}|\mathcal{M}_1) = \int \cdots \int \prod_{l=1}^{t} g_l(x_l)\tilde{\tau}_l^{y_l}(\mathrm{d}x_l|x_{l-1})\tau_0(\mathrm{d}x_0),$$

the difference between the evidence of $\mathcal{M}_0$ and the evidence of $\mathcal{M}_1$ depends on the difference between the transition kernels $\tilde{\tau}_t$ and $\tilde{\tau}_t^{y_t}$.

We have empirically observed in several computer experiments that $\mathsf{p}(y_{1:t}|\mathcal{M}_1) > \mathsf{p}(y_{1:t}|\mathcal{M}_0)$ and we argue that the observation-driven kernel $\tilde{\tau}_t^{y_t}$ implicit to the NuPF is the reason why the latter filter is robust to modeling errors in the state equation, compared to standard PFs. This claim is supported by the numerical results in Sects. 5.2 and 5.3, which show how the NuPF attains a significant better performance than the standard BPF, the auxiliary PF Pitt and Shephard (1999) or the extended Kalman filter (Anderson and Moore 1979) in scenarios where the filters are built upon a transition kernel different from the one used to generate the actual observations.

While it is hard to show that $\mathsf{p}(y_{1:t}|\mathcal{M}_1) > \mathsf{p}(y_{1:t}|\mathcal{M}_0)$ for *every* NuPF, it is indeed possible to guarantee that the latter inequality holds for specific nudging schemes. An example is provided in "Appendix C", where we describe a certain nudging operator $\alpha_t^{y_t}$ and then proceed to prove that $\mathsf{p}(y_{1:t}|\mathcal{M}_1) > \mathsf{p}(y_{1:t}|\mathcal{M}_0)$, for that particular scheme, under some regularity conditions on the likelihoods and transition kernels.

5 Computer simulations

In this section, we present the results of several computer experiments. In the first one, we address the tracking of a linear-Gaussian system. This is a very simple model which enables a clearcut comparison of the NuPF and other competing schemes, including a conventional PF with optimal importance function (which is intractable for all other examples) and a PF with nudging and proper importance weights. Then, we study three nonlinear tracking problems:

- A stochastic Lorenz 63 model with misspecified parameters,
- A maneuvering target monitored by a network of sensors collecting nonlinear observations corrupted with heavy-tailed noise,
- And, finally, a high-dimensional stochastic Lorenz 96 model.[4]

We have used gradient nudging in all experiments, with either $M \leq \sqrt{N}$ (deterministically, with batch nudging) or $\mathbb{E}[M] \leq \sqrt{N}$ (with independent nudging). This ensures that the assumptions of Theorem 1 hold. For simplicity, the gradient steps are computed with fixed step sizes, i.e., $\gamma_t = \gamma$ for

[4] For the experiments involving Lorenz 96 model, simulation from the model is implemented in C++ and integrated into MATLAB. The rest of the simulations are fully implemented in MATLAB.

all t. For the object tracking experiment, we have used a large step size, but this choice does not affect the convergence rate of the NuPF algorithm either.

5.1 A high-dimensional, inhomogeneous linear-Gaussian state-space model

In this experiment, we compare different PFs implemented to track a high-dimensional linear-Gaussian SSM. In particular, the model under consideration is

$$x_0 \sim \mathcal{N}(0, I_{d_x}), \tag{5.1}$$

$$x_t|x_{t-1} \sim \mathcal{N}(x_{t-1}, Q), \tag{5.2}$$

$$y_t|x_t \sim \mathcal{N}(C_t x_t, R), \tag{5.3}$$

where $\{x_t\}_{t\geq 0}$ are hidden states, $\{y_t\}_{t\geq 1}$ are observations and Q and R are the process and the observation noise covariance matrices, respectively. The latter are diagonal matrices, namely $Q = q I_{d_x}$ and $R = I_{d_y}$, where $q = 0.1$, $d_x = 100$ and $d_y = 20$. The sequence $\{C_t\}_{t\geq 1}$ defines a time-varying observation model. The elements of this sequence are chosen as random binary matrices, i.e., $C_t \in \{0, 1\}^{d_y \times d_x}$ where each entry is generated as an independent Bernoulli random variable with $p = 0.5$. Once generated, they are fixed and fed into all algorithms we describe below for each independent Monte Carlo run.

We compare the NuPF with three alternative PFs. The first method we implement is the PF with the optimal proposal pdf $p(x_t|x_{t-1}, y_t) \propto g_t(y_t|x_t)\tau_t(x_t|x_{t-1})$, abbreviated as Optimal PF. The pdf $p(x_t|x_{t-1}, y_t)$ leads to an analytically tractable Gaussian density for model (5.1)–(5.3) (Doucet et al. 2000) but not in the nonlinear tracking examples below. Note, however, that at each time step, the mean and covariance matrix of this proposal have to be explicitly evaluated in order to compute the importance weights.

The second filter is a nudged PF with proper importance weights (NuPF-PW). In this case, we treat the generation of the nudged particles as a proposal function to be accounted for during the weighting step. To be specific, the proposal distribution resulting from the NuPF has the form

$$\tilde{\tau}_t(\mathrm{d}x_t|x_{t-1}) = (1-\epsilon_N)\tau_t(\mathrm{d}x_t|x_{t-1}) + \epsilon_N \bar{\tau}_t(\mathrm{d}x_t|x_{t-1}), \tag{5.4}$$

where $\epsilon_N = \frac{1}{\sqrt{N}}$ and

$$\bar{\tau}_t(\mathrm{d}x_t|x_{t-1}) = \int \delta_{\alpha_t^{y_t}(\bar{x}_t)}(\mathrm{d}x_t)\tau_t(\mathrm{d}\bar{x}_t|x_{t-1}).$$

The latter conditional distribution admits an explicit representation as a Gaussian for model (5.1)–(5.3) when the α_t operator is designed as a gradient step, but this approach is intractable for the examples in Sects. 5.2 and 5.4. Note that $\tilde{\tau}_t$ is a mixture of two time-varying Gaussians and this fact adds to the cost of the sampling and weighting steps. Specifically, computing weights for the NuPF-PW is significantly more costly, compared to the BPF or the NuPF, because mixture (5.4) has to be evaluated together with the likelihood and the transition pdf.

The third tracking algorithm implemented for model (5.1)–(5.3) is the conventional BPF.

For all filters, we have set the number of particles as[5] $N = 100$. In order to implement the NuPF and NuPF-PW schemes, we have selected the step size $\gamma = 2 \times 10^{-2}$. We have run 1000 independent Monte Carlo runs for this experiment. To evaluate different methods, we have computed the empirical normalized mean squared errors (NMSEs). Specifically, the NMSE for the jth simulation is

$$\overline{\mathrm{NMSE}}(j) = \frac{\sum_{t=1}^{t_f} \|\bar{x}_t - \hat{x}_t(j)\|_2^2}{\sum_{t=1}^{t_f} \|x_t\|_2^2}, \tag{5.5}$$

where $\bar{x}_t = \mathbb{E}[x_t|y_{1:t}]$ is the exact posterior mean of the state x_t conditioned on the observations up to time t and $\hat{x}_t(j)$ is the estimate of the state vector in the jth simulation run. Therefore, the notation $\overline{\mathrm{NMSE}}$ implies the normalized mean squared error is computed with respect to $\bar{x}_t$. In the figures, we usually plot the mean and the standard deviation of the sample of errors, $\overline{\mathrm{NMSE}}(1), \ldots, \overline{\mathrm{NMSE}}(1000)$.

The results are shown in Fig. 1. In particular, in Fig. 1a, we observe that the $\overline{\mathrm{NMSE}}$ performance of the NuPF compared to the optimal PF and NuPF-PW (which is similar to a classical PF with nudging) is comparable. However, Fig. 1b reveals that the NuPF is significantly less demanding compared to the optimal PF and the NuPF-PW method. Indeed, the run times of the NuPF are almost identical to the those of the plain BPF. As a result, the plot of the $\overline{\mathrm{NMSE}}$s multiplied by the running times displayed in Fig. 1b reveals that the proposed algorithm is as favorable as the optimal PF, which can be implemented for this model, but not for general models unlike the NuPF.

5.2 Stochastic Lorenz 63 model with misspecified parameters

In this experiment, we demonstrate the performance of the NuPF when tracking a misspecified stochastic Lorenz 63 model. The dynamics of the system is described by a stochastic differential equation (SDE) in three dimensions,

[5] When N is increased, the results are similar for the NuPF, the optimal PF and the NuPF-PW larger number particles, as they already perform close to optimally for $N = 100$, and only the BPF improves significantly.

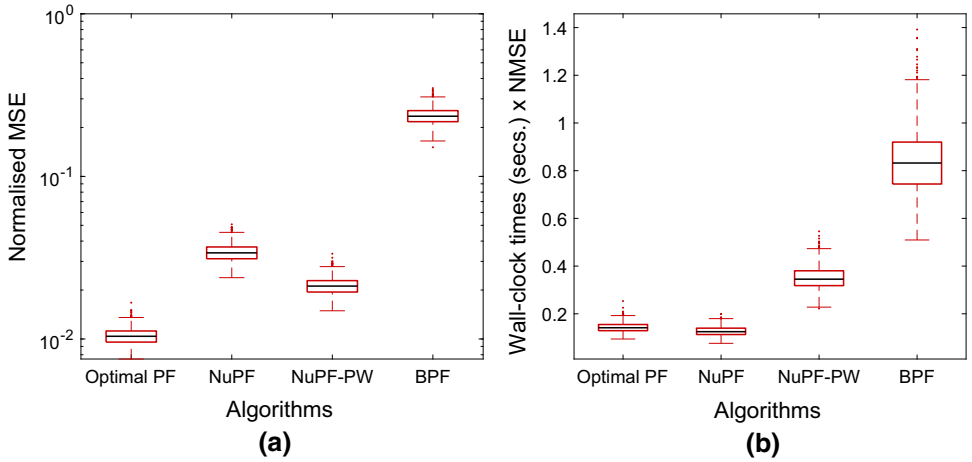

Fig. 1 **a** $\overline{\text{NMSE}}$ of the Optimal PF, NuPF-PW, NuPF, and BPF methods implemented for the high-dimensional linear-Gaussian SSM given in (5.1)–(5.3). The boxplots are constructed from 1,000 independent Monte Carlo runs. It can be seen that the $\overline{\text{NMSE}}$ of the NuPF is comparable to the error of the Optimal PF and the NuPF-PW methods. **b** Runtimes × $\overline{\text{NMSE}}$s of all methods. This experiment shows that, in addition to the fact that the NuPF attains a comparable estimation performance, which can be seen in **a**, it has a computational cost similar to the plain BPF. The figure demonstrates that the NuPF has a comparable performance to the optimal PF for this model

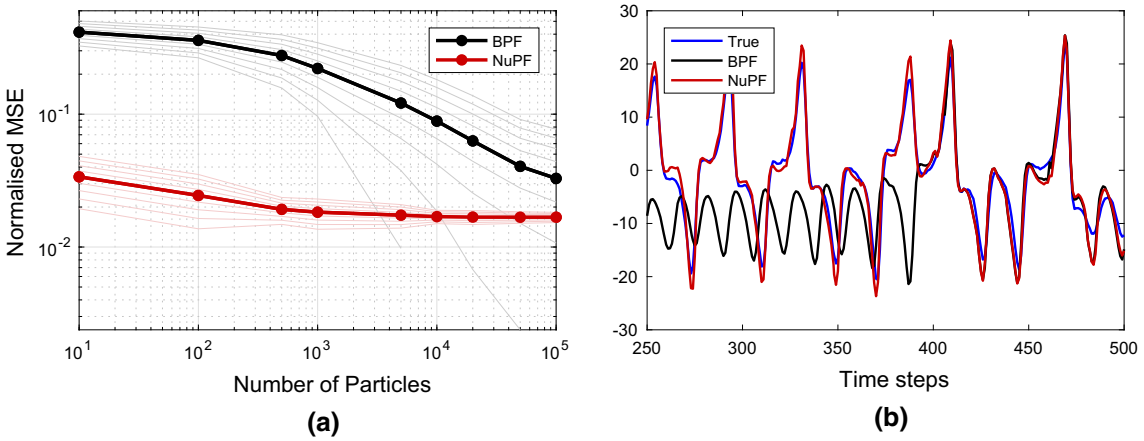

Fig. 2 **a** NMSE results of the BPF and NuPF algorithms for a misspecified Lorenz 63 system. The results have been obtained from 1000 independent Monte Carlo runs for each $N \in \{10, 100, 500, 1\,\text{K}, 5\,\text{K}, 10\,\text{K}, 20\,\text{K}, 50\,\text{K}, 100\,\text{K}\}$. The light-colored lines indicate the area containing up to one standard deviation from the empirical mean. The misspecified parameter is $\mathsf{b}_\epsilon = \mathsf{b} + \epsilon$, where $\mathsf{b} = 8/3$ and $\epsilon = 0.75$. **b** A sample path of the true state variable $x_{2,t}$ and its estimates in a run with $N = 500$ particles

$$
\begin{aligned}
\mathrm{d}x_1 &= -\mathsf{a}(x_1 - x_2)\mathrm{d}s + \mathrm{d}w_1,\\
\mathrm{d}x_2 &= (\mathsf{r}x_1 - x_2 - x_1x_3)\,\mathrm{d}s + \mathrm{d}w_2,\\
\mathrm{d}x_3 &= (x_1x_2 - \mathsf{b}x_3)\,\mathrm{d}s + \mathrm{d}w_3,
\end{aligned}
$$

where s denotes continuous time, $\{w_i(s)\}_{s\in(0,\infty)}$ for $i = 1, 2, 3$ are one-dimensional independent Wiener processes and $\mathsf{a}, \mathsf{r}, \mathsf{b} \in \mathbb{R}$ are fixed model parameters. We discretize the model using the Euler–Maruyama scheme with integration step $\mathsf{T} > 0$ and obtain the system of difference equations

$$
\begin{aligned}
x_{1,t} &= x_{1,t-1} - \mathsf{T}\mathsf{a}(x_{1,t-1} - x_{2,t-1}) + \sqrt{\mathsf{T}}u_{1,t},\\
x_{2,t} &= x_{2,t-1} + \mathsf{T}(\mathsf{r}x_{1,t-1} - x_{2,t-1} - x_{1,t-1}x_{3,t-1})\\
&\quad + \sqrt{\mathsf{T}}u_{2,t},\\
x_{3,t} &= x_{3,t-1} + \mathsf{T}(x_{1,t-1}x_{2,t-1} - \mathsf{b}x_{3,t-1}) + \sqrt{\mathsf{T}}u_{3,t}, \quad (5.6)
\end{aligned}
$$

where $\{u_{i,t}\}_{t\in\mathbb{N}}$, $i = 1, 2, 3$ are i.i.d. Gaussian random variables with zero mean and unit variance. We assume that we can only observe the variable $x_{1,t}$, contaminated by additive noise, every $t_s > 1$ discrete time steps. To be specific, we collect the sequence of observations

$$
y_n = k_o x_{1,nt_s} + v_n, \quad n = 1, 2, \ldots,
$$

where $\{v_n\}_{n\in\mathbb{N}}$ is a sequence of i.i.d. Gaussian random variables with zero mean and unit variance and the scale parameter $k_o = 0.8$ is assumed known.

In order to simulate both the state signal and the synthetic observations from this model, we choose the so-called standard parameter values

$$(\mathsf{a}, \mathsf{r}, \mathsf{b}) = \left(10, 28, \frac{8}{3}\right),$$

which make the system dynamics chaotic. The initial condition is set as

$$x_0 = [-5.91652, -5.52332, 24.5723]^\top.$$

The latter value has been chosen from a deterministic trajectory of the system (i.e., with no state noise) with the same parameter set $(\mathsf{a}, \mathsf{r}, \mathsf{b}) = \left(10, 28, \frac{8}{3}\right)$ to ensure that the model is started at a sensible point. We assume that the system is observed every $t_s = 40$ discrete time steps and for each simulation we simulate the system for $t = 0, 1, \ldots, t_f$, with $t_f = 20,000$. Since $t_s = 40$, we have a sequence of $\frac{t_f}{t_s} = 500$ observations overall.

Let us note here that the Markov kernel which takes the state from time $n-1$ to time n (i.e., from the time of one observation to the time of the next observation) is straightforward to simulate using Euler–Maruyama scheme (5.6); however, the associated transition probability density cannot be evaluated because it involves the mapping of both the state and a sequence of t_s noise samples through a composition of nonlinear functions. This precludes the use of importance sampling schemes that require the evaluation of this density when computing the weights.

We run the BPF and NuPF algorithms for the model described above, except that the parameter b is replaced by $\mathsf{b}_\epsilon = \mathsf{b} + \epsilon$, with $\epsilon = 0.75$ (hence $\mathsf{b}_\epsilon \approx 3.417$ vs. $\mathsf{b} \approx 2.667$ for the actual system). As the system underlying dynamics is chaotic, this mismatch affects the predictability of the system significantly.

We have implemented the NuPF with independent gradient nudging. Each particle is nudged with probability $\frac{1}{\sqrt{N}}$, where N is the number of particles (hence $\mathbb{E}[M] = \sqrt{N}$) and the size of the gradient steps is set to $\gamma = 0.75$ (see Algorithm 3).

As a figure of merit, we evaluate the NMSE for the three-dimensional state vector, averaged over 1000 independent Monte Carlo simulations. For this example (as well as in the rest of this section), it is not possible to compute the exact posterior mean of the state variables. Therefore, the NMSE values are computed with respect to the ground truth, i.e.,

$$\mathrm{NMSE}(j) = \frac{\sum_{t=1}^{t_f} \|x_t - \hat{x}_t(j)\|_2^2}{\sum_{t=1}^{t_f} \|x_t\|_2^2}, \tag{5.7}$$

where $(x_t)_{t\geq 1}$ is the ground truth signal.

Figure 2a displays the NMSE, attained for varying number of particles N, for the standard BPF and the NuPF. It is seen that the NuPF outperforms the BPF for the whole range of values of N in the experiment, in terms of both the mean and the standard deviation of the errors, although the NMSE values become closer for larger N. The plot on the right displays the values of $x_{2,t}$ and its estimates for a typical simulation. In general, the experiment shows that the NuPF can track the actual system using the misspecified model and a small number of particles, whereas the BPF requires a higher computational effort to attain a similar performance.

As a final experiment with this model, we have tested the robustness of the algorithms with respect to the choice of parameters in the nudging step. In particular, we have tested the NuPF with independent gradient nudging for a wide range of step sizes γ. Also, we have tested the NuPF with random search nudging using a wide range of covariances of the form $C = \sigma^2 I$ by varying σ^2.

The results can be seen in Fig. 3. This figure shows that the algorithm is robust to the choice of parameters for a range of step sizes and variances of the random search step. As expected, random search nudging takes longer running time compared to gradient steps. This difference in run times is expected to be larger in higher-dimensional models since random search is expected to be harder in such scenarios.

5.3 Object tracking with a misspecified model

In this experiment, we consider a tracking scenario where a target is observed through sensors collecting radio signal strength (RSS) measurements contaminated with additive heavy-tailed noise. The target dynamics are described by the model,

$$x_t = Ax_{t-1} + BL(x_{t-1} - x_\bullet) + u_t,$$

where $x_t \in \mathbb{R}^4$ denotes the target state, consisting of its position $r_t \in \mathbb{R}^2$ and its velocity, $v_t \in \mathbb{R}^2$; hence, $x_t = \begin{bmatrix} r_t \\ v_t \end{bmatrix} \in \mathbb{R}^4$. The vector $x_\bullet$ is a deterministic, pre-chosen state to be attained by the target. Each element in the sequence $\{u_t\}_{t\in\mathbb{N}}$ is a zero-mean Gaussian random vector with covariance matrix Q. The parameters A, B, Q are selected as

$$A = \begin{bmatrix} I_2 & \kappa I_2 \\ 0 & 0.99 I_2 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & I_2 \end{bmatrix}^\top,$$

and

$$Q = \begin{bmatrix} \frac{\kappa^3}{3} I_2 & \frac{\kappa^2}{2} I_2 \\ \frac{\kappa^2}{2} I_2 & \kappa I_2 \end{bmatrix},$$

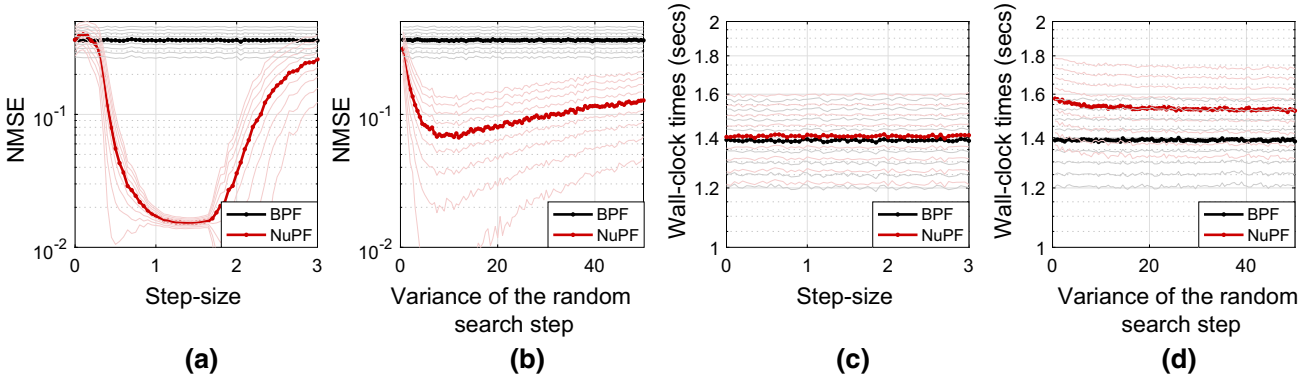

Fig. 3 A comparison of gradient nudging and random search nudging for a variety of parameter settings. From **a**, it can be seen that gradient nudging is robust within a large interval for γ. From **b**, one can see that the same is true for random search nudging with the covariance of the form $C = \sigma^2 I$ for a wide range of σ^2. From **c**, **d**, it can be seen that while gradient nudging causes negligible computational overhead, random search nudging is more demanding in terms of computation time and this behavior is expected to be more apparent in higher-dimensional spaces. Comparing **a**, **b**, it can also be seen that gradient nudging attains lower error rates in general. The lighter-colored lines indicate the area containing up to one standard deviation from the empirical means in each plot

where I_2 is the 2×2 identity matrix and $\kappa = 0.04$. The policy matrix $L \in \mathbb{R}^{2\times 4}$ determines the trajectory of the target from an initial position $x_0 = [140, 140, 50, 0]^\top$ to a final state $x_\bullet = [140, -140, 0, 0]^\top$ and it is computed by solving a Riccati equation (see Bertsekas 2001 for details), which yields

$$L = \begin{bmatrix} -0.0134 & 0 & -0.0381 & 0 \\ 0 & -0.0134 & 0 & -0.0381 \end{bmatrix}.$$

This policy results in a highly maneuvering trajectory. In order to design the NuPF, however, we assume the simpler dynamical model

$$x_t = Ax_{t-1} + u_t;$$

hence, there is a considerable model mismatch.

The observations are nonlinear and coming from 10 sensors placed in the region where the target moves. The measurement collected at the ith sensor, time t, is modeled as

$$y_{t,i} = 10 \log_{10}\left(\frac{P_0}{\|r_t - s_i\|^2} + \eta\right) + w_{t,i},$$

where $r_t \in \mathbb{R}^2$ is the location vector of the target, s_i is the position of the ith sensor and $w_{t,i} \sim \mathcal{T}(0, 1, \nu)$ is an independent t-distributed random variable for each $i = 1, \ldots, 10$. Intuitively, the closer the parameter ν to 1, the more explosive the observations become. In particular, we set $\nu = 1.01$ to make the observations explosive and heavy tailed. As for the sensor parameters, we set the transmitted RSS as $P_0 = 1$ and the sensitivity parameter as $\eta = 10^{-9}$. The latter yields a soft lower bound of -90 decibels (dB) for the RSS measurements.

We have implemented the NuPF with batch gradient nudging, with a large step size $\gamma = 5.5$ and $M = \lfloor \sqrt{N} \rfloor$. Since the observations depend on the position vector r_t only, an additional model-specific nudging step is needed for the velocity vector v_t. In particular, after nudging the $r_t^{(i)} = [x_{1,t}^{(i)}, x_{2,t}^{(i)}]^\top$, we update the velocity variables as

$$v_t^{(i)} = \frac{1}{\kappa}\left(r_t^{(i)} - r_{t-1}^{(i)}\right), \quad \text{where} \quad v_t^{(i)} = \left[x_{3,t}^{(i)}, x_{4,t}^{(i)}\right]^\top,$$

where $\kappa = 0.04$ as defined for the model. The motivation for this additional transformation comes from the physical relationship between position and velocity. We note, however, that the NuPF also works without nudging the velocities.

We have run 10, 000 Monte Carlo runs with $N = 500$ particles in the auxiliary particle filter (APF) (Pitt and Shephard 1999; Johansen and Doucet 2008; Douc et al. 2009), the BPF (Gordon et al. 1993) and the NuPF. We have also implemented the extended Kalman filter (EKF), which uses the gradient of the observation model.

Figure 4 shows a typical simulation run with each one of the four algorithms [on the left side, plots (a)–(d)] and a boxplot of the NMSEs obtained for the 10,000 simulations [on the right, plot (e)]. Plots (a)–(d) show that, while the EKF also uses the gradient of the observation model, it fails to handle the heavy-tailed noise, as it relies on Gaussian approximations. The BPF and the APF collapse due to the model mismatch in the state equation. Plot (d) shows that the NMSE of the NuPF is just slightly smaller in the mean than the NMSE of the EKF, but much more stable.

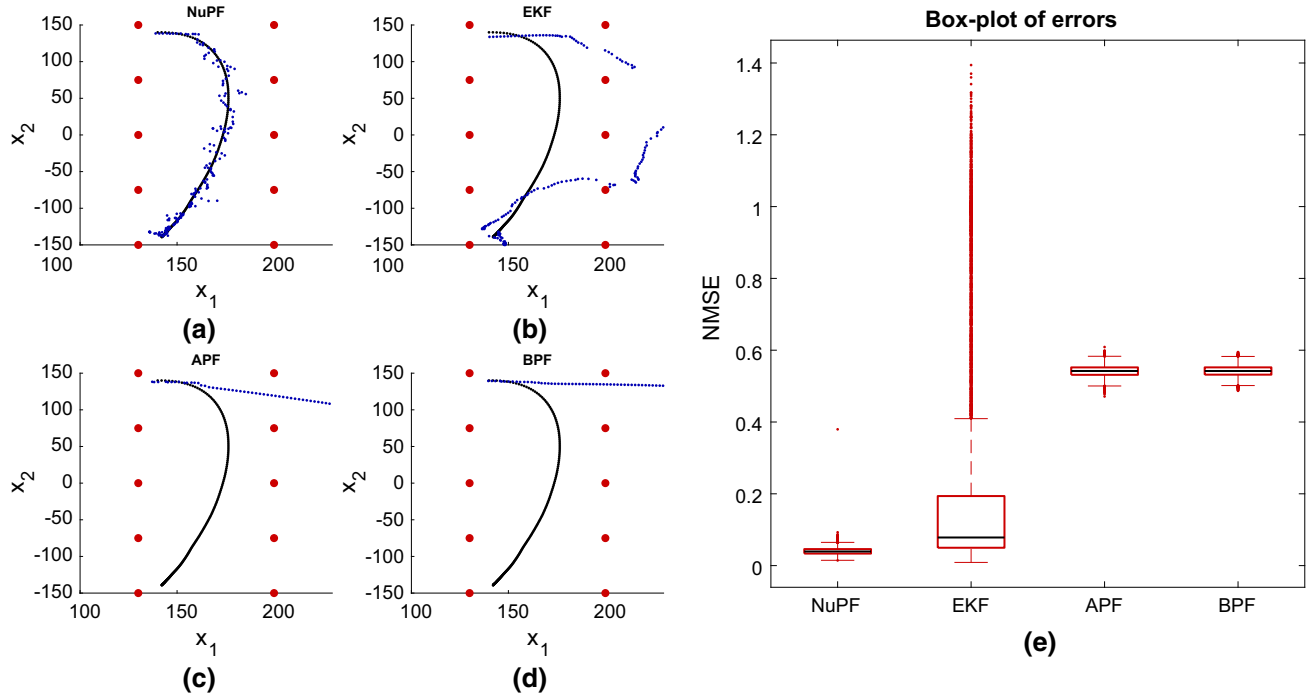

Fig. 4 Plots **a**–**d**: A typical simulation run for the BPF, APF, EKF and NuPF algorithms using $N = 500$ particles. The black dots denote the real trajectory of the object, the red dots are sensors, and the blue dots are position estimates as provided by the filters. Plot **e**: Boxplot of the errors $\mathrm{NMSE}(1), \ldots, \mathrm{NMSE}(10,000)$ obtained for the set of independent simulation runs. The NuPF achieves a low NMSE with a low variance, whereas the EKF exhibits a large variance

5.4 High-dimensional stochastic Lorenz 96 model

In this computer experiment, we compare the NuPF with the ensemble Kalman filter (EnKF) for the tracking of a stochastic Lorenz 96 system. The latter is described by the set of stochastic differential equations (SDEs)

$$\mathrm{d}x_i = \big[(x_{i+1} - x_{i-2})x_{i-1} - x_i + F\big]\,\mathrm{d}s + \mathrm{d}w_i, \quad i = 1, \ldots, d,$$

where s denotes continuous time, $\{w_i(s)\}_{s\in(0,\infty)}$, $1 \le i \le d$, are independent Wiener processes, d is the system dimension and the forcing parameter is set to $F = 8$, which ensures a chaotic regime. The model is assumed to have a circular spatial structure, so that $x_{-1} = x_{d-1}$, $x_0 = x_d$, and $x_{d+1} = x_1$. Note that each x_i, $i = 1, \ldots, d$, denotes a time-varying state associated to a different space location. In order to simulate data from this model, we apply the Euler–Maruyama discretization scheme and obtain the difference equations,

$$x_{i,t} = x_{i,t-1} + \mathsf{T}\big[(x_{i+1,t-1} - x_{i-2,t-1})x_{i-1,t-1} - x_{i,t-1} + F\big] + \sqrt{\mathsf{T}}u_{i,t},$$

where $u_{i,t}$ are zero-mean, unit-variance Gaussian random variables. We initialise this system by generating a vector from the uniform distribution on $(0, 1)^d$ and running the system for a small number of iterations and set x_0 as the output of this short run.

We assume that the system is only partially observed. In particular, half of the state variables are observed, in Gaussian noise, every $t_s = 10$ time steps, namely

$$y_{j,n} = x_{2j-1,nt_s} + u_{j,n},$$

where $n = 1, 2, \ldots$, $j = 1, 2, \ldots, \lfloor d/2 \rfloor$, and $u_{j,n}$ is a normal random variable with zero mean and unit variance. The same as in the stochastic Lorenz 63 example of Sect. 5.2, the transition pdf that takes the state from time $(n-1)t_s$ to time nt_s is simple to simulate but hard to evaluate, since it involves mapping a sequence of noise variables through a composition of nonlinearities.

In all the simulations for this system, we run the NuPF with batch gradient nudging (with $M = \lfloor\sqrt{N}\rfloor$ nudged particles and step size $\gamma = 0.075$). In the first computer experiment, we fixed the dimension $d = 40$ and run the BPF and the NuPF with increasing number of particles. The results can be seen in Fig. 5, which shows how the NuPF performs better than the BPF in terms of NMSE [plot (a)]. Since the run times of both algorithms are nearly identical, it can be seen that, when considered jointly with NMSEs, the NuPF attains a significantly better performance [plot (b)].

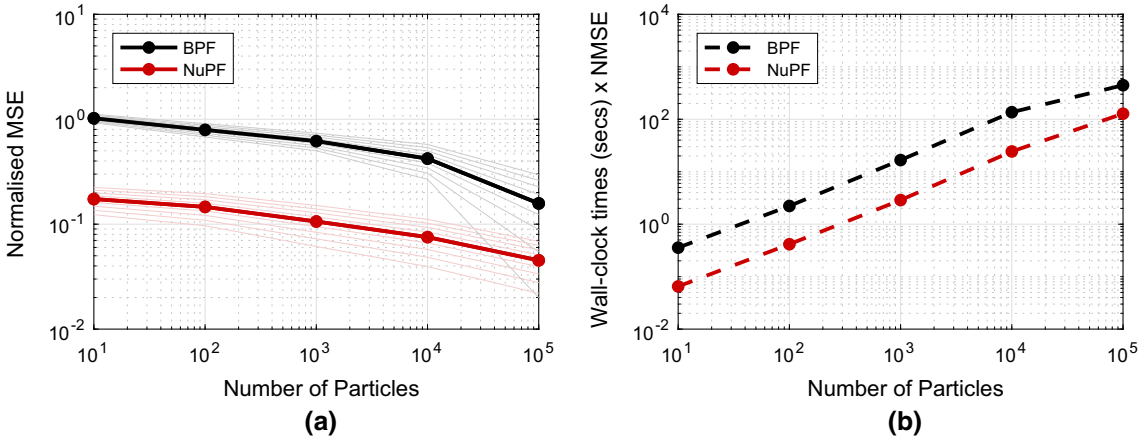

Fig. 5 Comparison of the NuPF and the BPF for the stochastic Lorenz 96 system with model dimension $d = 40$. The results have been averaged over a set of 1024 independent Monte Carlo runs. Plot **a**: evolution of the NMSE as the number of particles N is increased. The light-colored lines indicate the area containing up to one standard deviation from the empirical mean. Plot **b**: Run-$times \times NMSE$ for the BPF and the NuPF in the same set of simulations. Since the increase in computational cost of the NuPF, compared to the BPF, is negligible, it is seen from plot **b** that the NuPF performs better when errors and run times are considered jointly

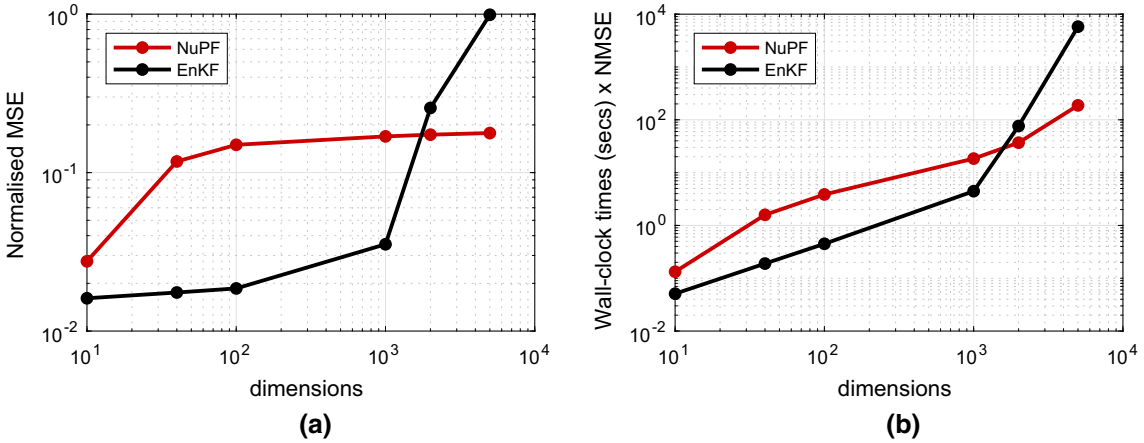

Fig. 6 Comparison of the NuPF with the EnKF for the stochastic Lorenz 96 model with increasing dimension d and fixed number of particles $N = 500$ (this is the same as the number of ensemble members in the EnKF). We have run 1000 independent Monte Carlo trials for this experiment. Plot **a**: NMSE versus dimension d. The EnKF attains a smaller error for lower dimensions, but then it explodes for $d > 10^3$, while the NuPF remains stable. Plot **b**: Running times × NMSE plot for the same set of simulations. It can be seen that the overall performance of the NuPF is better beyond 1K dimensions compared to the EnKF

In a second computer experiment, we compared the NuPF with the EnKF. Figure 6a shows how the NMSE of the two algorithms grows as the model dimension d increases and the number of particles N is kept fixed. In particular, the EnKF attains a better performance for smaller dimensions (up to $d = 10^3$); however, its NMSE blows up for $d > 10^3$ while the performance of the NuPF remains stable. The running time of the EnKF was also higher than the running time of the NuPF in the range of higher dimensions ($d \geq 10^3$).

5.5 Assessment of bias

In this section, we numerically quantify the bias of the proposed algorithm on a low-dimensional linear-Gaussian state-space model. To assess the bias, we compute the marginal likelihood estimates given by the BPF and the NuPF. The reason for this choice is that the BPF is known to yield unbiased estimates of the marginal likelihood (Del Moral 2004).[6] The NuPF leads to biased (typically overestimated) marginal likelihood estimates; hence, it is of interest to compare them with those of the BPF. To this end, we choose a simple linear-Gaussian state-space model for which the marginal

[6] Note that the estimates of integrals (φ, π_t) computed using the self-normalized importance sampling approximations (i.e., $(\varphi, \pi_t^N) \approx (\varphi, \pi_t)$) produced by the BPF and the NuPF methods are biased and the bias vanishes with the same rate for both algorithms as a result of Theorem 1. The same is true for the approximate predictive measures ξ_t^N.

likelihood can be exactly computed as a by-product of the Kalman filter. We then compare this exact marginal likelihood to the estimates given by the BPF and the NuPF.

Particularly, we define the state-space model,

$$x_0 \sim \mathcal{N}(x_0; \mu_0, P_0), \tag{5.8}$$

$$x_t|x_{t-1} \sim \mathcal{N}(x_t; x_{t-1}, Q), \tag{5.9}$$

$$y_t|x_t \sim \mathcal{N}(y_t; C_t x_t, R), \tag{5.10}$$

where $(C_t)_{t\geq 0} \in [0,1]^{1\times 2}$ is a sequence of observation matrices where each entry is generated as a realization of a Bernoulli random variable with $p = 0.5$, μ_0 is a zero vector and $x_t \in \mathbb{R}^2$ and $y_t \in \mathbb{R}$. The state variables are cross-correlated, namely

$$Q = \begin{bmatrix} 2.7 & -0.48 \\ -0.48 & 2.05 \end{bmatrix},$$

and $R = 1$. We have chosen the prior covariance as $P_0 = I_{d_x}$. We have simulated the system for $T = 100$ time steps. Given a fixed observation sequence $y_{1:T}$, the marginal likelihood for the system given in Eqs. (5.9)–(5.10) is

$$Z^\star = \mathsf{p}(y_{1:T}),$$

which can be exactly computed via the Kalman filter.

We denote the estimate of $Z^\star$ given by the BPF and the NuPF as Z^N_{BPF} and Z^N_{NuPF}, respectively. It is well known that the BPF estimator is unbiased (Del Moral 2004),

$$\mathbb{E}[Z^N_{\mathrm{BPF}}] = Z^\star, \tag{5.11}$$

where $\mathbb{E}[\cdot]$ denotes the expectation with respect to the randomness of the particles. Numerically, this suggests that as one runs identical, independent Monte Carlo simulations to obtain $\{Z^{N,k}_{\mathrm{BPF}}\}_{k=1}^K$ and compute the average

$$\bar{Z}^N_{\mathrm{BPF}} = \frac{1}{K}\sum_{k=1}^{K} Z^{N,k}_{\mathrm{BPF}}, \tag{5.12}$$

then it follows from unbiasedness property (5.11) that the ratio of the average in (5.12) and the true value $Z^\star$ should satisfy

$$\frac{\bar{Z}^N_{\mathrm{BPF}}}{Z^\star} \to 1 \quad \text{as} \quad K \to \infty.$$

Since the marginal likelihood estimates provided by the NuPF are not unbiased for the original SSM (and tend to attain higher values), if we define

$$\bar{Z}^N_{\mathrm{NuPF}} = \frac{1}{K}\sum_{k=1}^{K} Z^{N,k}_{\mathrm{NuPF}},$$

then as $K \to \infty$, we should see

$$\frac{\bar{Z}^N_{\mathrm{NuPF}}}{Z^\star} \to 1 + \epsilon \quad \text{as} \quad K \to \infty,$$

for some $\epsilon > 0$.

We have conducted an experiment aimed at quantifying the bias $\epsilon > 0$ above. In particular, we have run 20,000 independent simulations for the BPF and the NuPF with $N = 100$, $N = 1000$ and $N = 10,000$. For each value of N, we have computed running empirical means as in (5.12) and (6.1) for $K = 1, \ldots, 20,000$. The variance of $\bar{Z}^N_{\mathrm{BPF}}$ increases with T; hence, the estimators for small K display a relatively large variance and we need $K >> 1$ to clearly observe the bias. The NuPF filter performs independent gradient nudging with step size $\gamma = 0.1$.

The results of the experiment are displayed in Fig. 7, which shows how, as expected, the NuPF overestimates $Z^\star$. We can also see how the bias becomes smaller as N increases (because only and average of $\sqrt{N}$ particles are nudged per time step).

6 Experimental results on model inference

In this section, we illustrate the application of the NuPF to estimate the parameters of a financial time-series model. In particular, we adopt a stochastic volatility SSM and we aim at estimating its unknown parameters (and track its state variables) using the EURUSD log-return data from 2014-12-31 to 2016-12-31 (obtained from `www.quandl.com`). For this task, we apply two recently proposed Monte Carlo schemes: the nested particle filter (NPF) (Crisan and Miguez 2018) (a purely recursive, particle filter style Monte Carlo method) and the particle Metropolis–Hastings (pMH) algorithm (Andrieu et al. 2010) (a batch Markov chain Monte Carlo procedure). In their original forms, both algorithms use the marginal likelihood estimators given by the BPF to construct a Monte Carlo approximation of the posterior distribution of the unknown model parameters. Here, we compare the performance of these algorithms when the marginal likelihoods are computed using either the BPF or the proposed NuPF.

We assume the stochastic volatility SSM (Tsay 2005),

$$x_0 \sim \mathcal{N}\left(\mu, \frac{\sigma_v^2}{1-\phi^2}\right), \tag{6.1}$$

$$x_t|x_{t-1} \sim \mathcal{N}(\mu + \phi(x_{t-1} - \mu), \sigma_v^2), \tag{6.2}$$

$$y_t|x_t \sim \mathcal{N}(0, \exp(x_t)), \tag{6.3}$$

where $\mu \in \mathbb{R}$, $\sigma_v \in \mathbb{R}_+$ and $\phi \in [-1, 1]$ are fixed but unknown parameters. The states $\{x_t\}_{1\leq t\leq T}$ are log-volatilities, and the observations $\{y_t\}_{1\leq t\leq T}$ are log-returns.

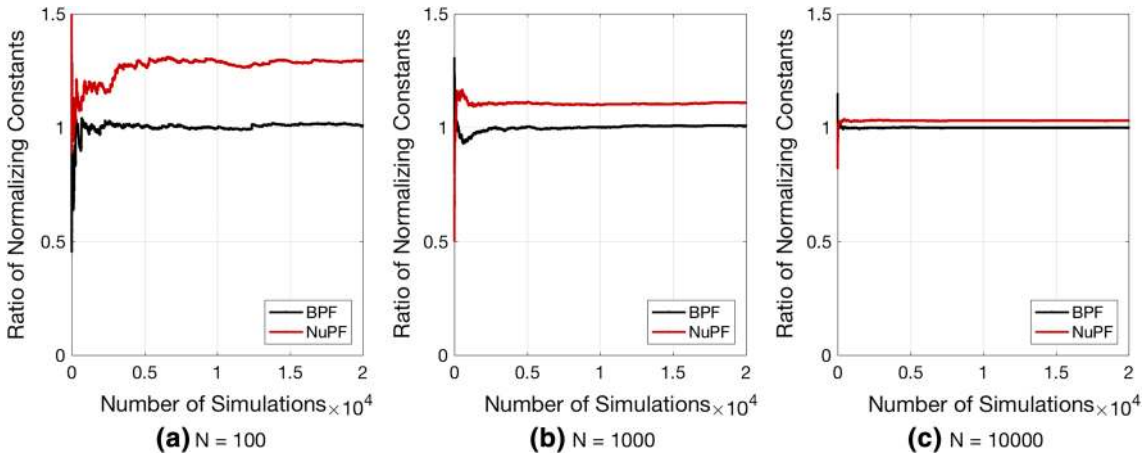

Fig. 7 Evolution of the running averages $\bar{Z}^N_{\mathrm{BPF}}/Z^\star$ (black) and $\bar{Z}^N_{\mathrm{NuPF}}/Z^\star$ (red) for $K = 1, \ldots,$ 20,000 independent simulations with $N = 100$, $N = 1000$ and N = 10,000 particles for both filters. The ratio $\bar{Z}^N_{\mathrm{BPF}}/Z^\star$ for the BPF is unbiased (Del Moral 2004) and hence converges to 1. The ratio $\bar{Z}^N_{\mathrm{NuPF}}/Z^\star$ for the NuPF converges to $1+\epsilon$, with $\epsilon > 0$ becoming smaller as N increases, showing that the estimator Z^N_{NuPF} is biased (yet asymptotically unbiased with $N \to \infty$; see Theorem 1). (Color figure online)

We follow the same procedure as Dahlin and Schön (2015) to preprocess the observations. Given the historical price sequence $s_0, \ldots, s_T$, the log-return at time t is calculated as

$$y_t = 100 \log(s_t/s_{t-1})$$

for $1 \leq t \leq T$. Then, given $y_{1:T}$, we tackle the joint Bayesian estimation of $x_{1:T}$ and the unknown parameters $\theta = (\mu, \sigma_v, \phi)$. In the next two subsections, we compare the conventional BPF and the NuPF as building blocks of the NPF and the pMH algorithms.

6.1 Nudging the nested particle filter

The NPF in Crisan and Miguez (2018) consists of two layers of particle filters which are used to jointly approximate the posterior distributions of the parameters and the states. The filter in the first layer builds a particle approximation of the marginal posterior distribution of the parameters. Then, for each particle in the parameter space, say $\theta^{(i)}$, there is an *inner* filter that approximates the posterior distribution of the states conditional on the parameter vector $\theta^{(i)}$. The inner filters are classical particle filters, which are essentially used to compute the importance weights (marginal likelihoods) of the particles in the parameter space. In the implementation of Crisan and Miguez (2018), the inner filters are conventional BPFs. We have compared this conventional implementation with an alternative one where the BPFs are replaced by the NuPFs. For a detailed description of the NPF, see Crisan and Miguez (2018).

In order to assess the performances of the nudged and classical versions of the NPF, we compute the model evidence estimate given by the nested filter by integrating out both the parameters and the states. In particular, if the set of particles in the parameter space at time t is $\{\theta_t^{(i)}\}_{i=1}^K$ and for each particle $\theta_t^{(i)}$ we have a set of particles in the state space $\{x_t^{(i,j)}\}_{j=1}^N$, we compute

$$\widehat{\mathrm{p}(y_{1:T})} = \prod_{t=1}^{T} \left[\frac{1}{KN} \sum_{i=1}^{K} \sum_{j=1}^{N} g_t(x_t^{(i,j)}) \right].$$

The model evidence quantifies the fitness of the stochastic volatility model for the given dataset; hence, we expect to see a higher value when a method attains a better performance (the intuition is that if we have better estimates of the parameters and the states, then the model will fit better). For this experiment, we compute the model evidence for the nudged NPF *before* the nudging step, so as to make the comparison with the conventional algorithm fair.

We have conducted 1000 independent Monte Carlo runs for each algorithm and computed the model evidence estimates. We have used the same parameters and the same setup for the two versions of the NPF (nudged and conventional). In particular, each unknown parameter is jittered independently. The parameter μ is jittered with a zero-mean Gaussian kernel variance $\sigma_\mu^2 = 10^{-3}$, the parameter σ_v is jittered with a truncated Gaussian kernel on $(0, \infty)$ with variance $\sigma_{\sigma_v}^2 = 10^{-4}$, and the parameter ϕ is jittered with a zero-mean truncated Gaussian kernel on $[-1, 1]$, with variance $\sigma_\phi^2 = 10^{-4}$. We have chosen a large step size for the nudging step, $\gamma = 4$, and we have used batch nudging with $M = \lfloor \sqrt{N} \rfloor$.

The results in Fig. 8 demonstrate empirically that the use of the nudging step within the NPF reduces the variance of the model evidence estimators; hence, it improves the numerical stability of the NPF.

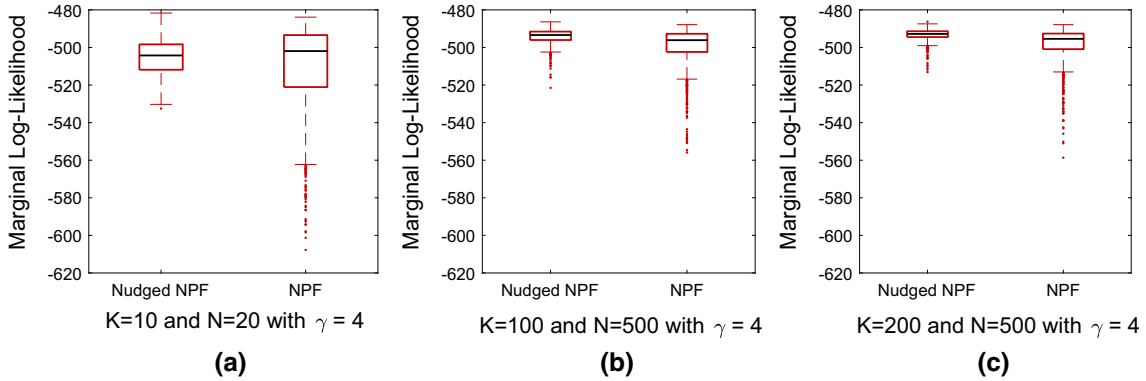

Fig. 8 Model evidence estimates produced by the nudged NPF and the conventional NPF with varying computational effort. From **a** to **c**, it can be seen that, as we increase the number of particles in the parameter space (K) and the state space (N), the variances of the estimates are smaller. The nudged NPF results in much more stable estimates, with lower variance and fewer extreme values

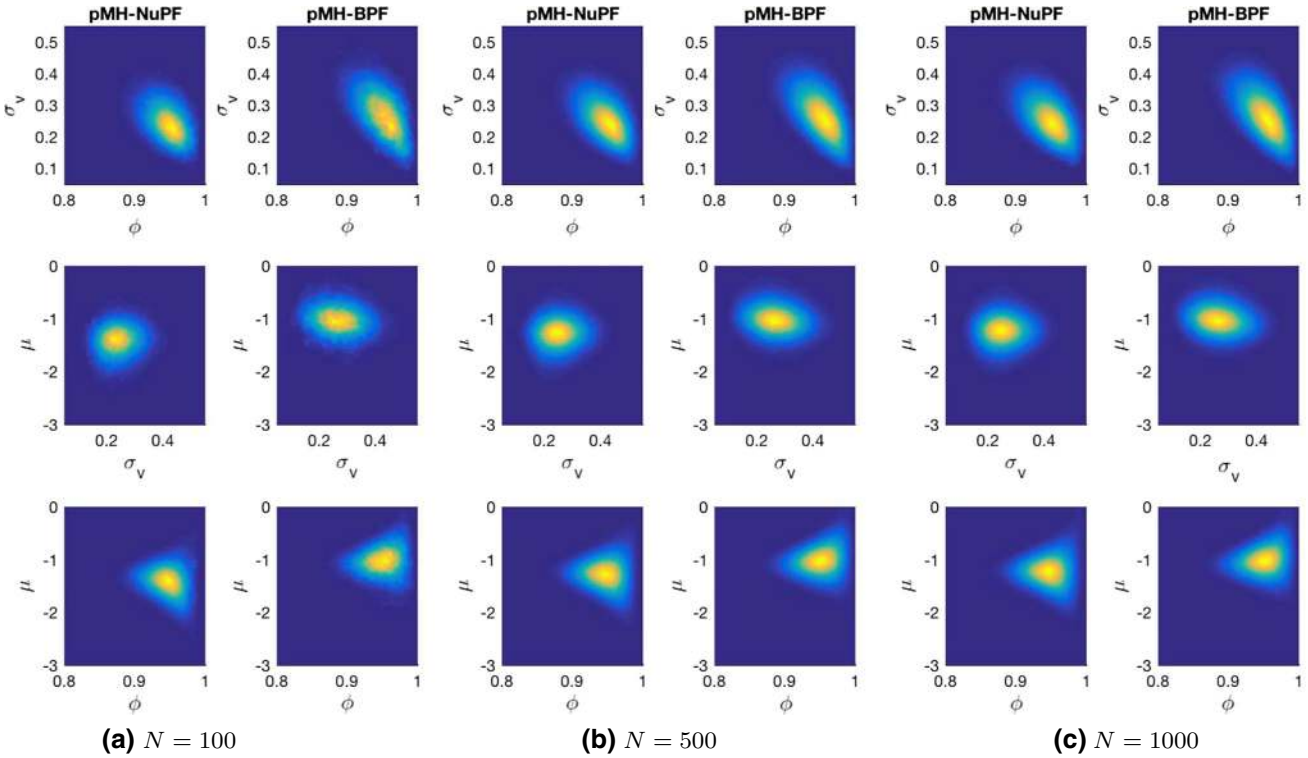

Fig. 9 The parameter posterior distributions found by the pMH-NuPF and the pMH-BPF for varying N. It can be seen that, as N increases, the impact of the nudging-induced bias on the posterior distributions vanishes

6.2 Nudging the particle Metropolis–Hastings

The pMH algorithm is a Markov chain Monte Carlo (MCMC) method for inferring parameters of general SSMs (Andrieu et al. 2010). The pMH uses PFs as auxiliary devices to estimate parameter likelihoods in a similar way as the NPF uses them to compute importance weights. In the case of the pMH, these estimates should be unbiased and they are needed to determine the acceptance probability for each element of the Markov chain. For the details of the algorithm, see Andrieu et al. (2010) (or Dahlin and Schön 2015 for a tutorial-style introduction). Let us note that the use of NuPF does not lead to an unbiased estimate of the likelihood with respect to the assumed SSM. However, as discussed in Sect. 4.3, one can view the use of nudging in this context as an implementation of pMH with an implicit dynamical model $\mathcal{M}_1$ derived from the original SSM $\mathcal{M}_0$.

We have carried out a computer experiment to compare the performance of the pMH scheme using either BPFs or NuPFs to compute acceptance probabilities. The two algorithms are

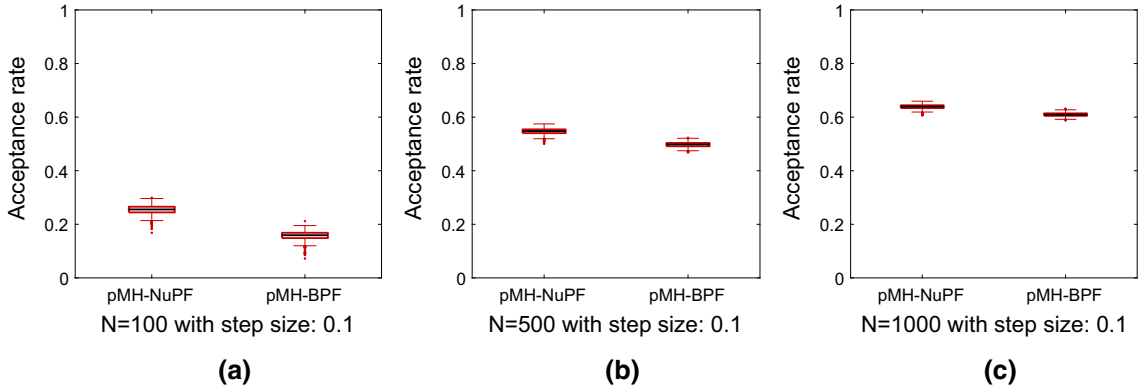

Fig. 10 Empirical acceptance rates computed for the pMH running BPF and the pMH running NuPF. From **a**, it can be seen that there is a significant increase in the acceptance rates when the number of particles are low, e.g., $N = 100$. From **b** and **c**, it can be seen that the pMH-NuPF is still better for increasing number of particles but the pMH-BPF is catching up with the performance of the pMH-NuPF

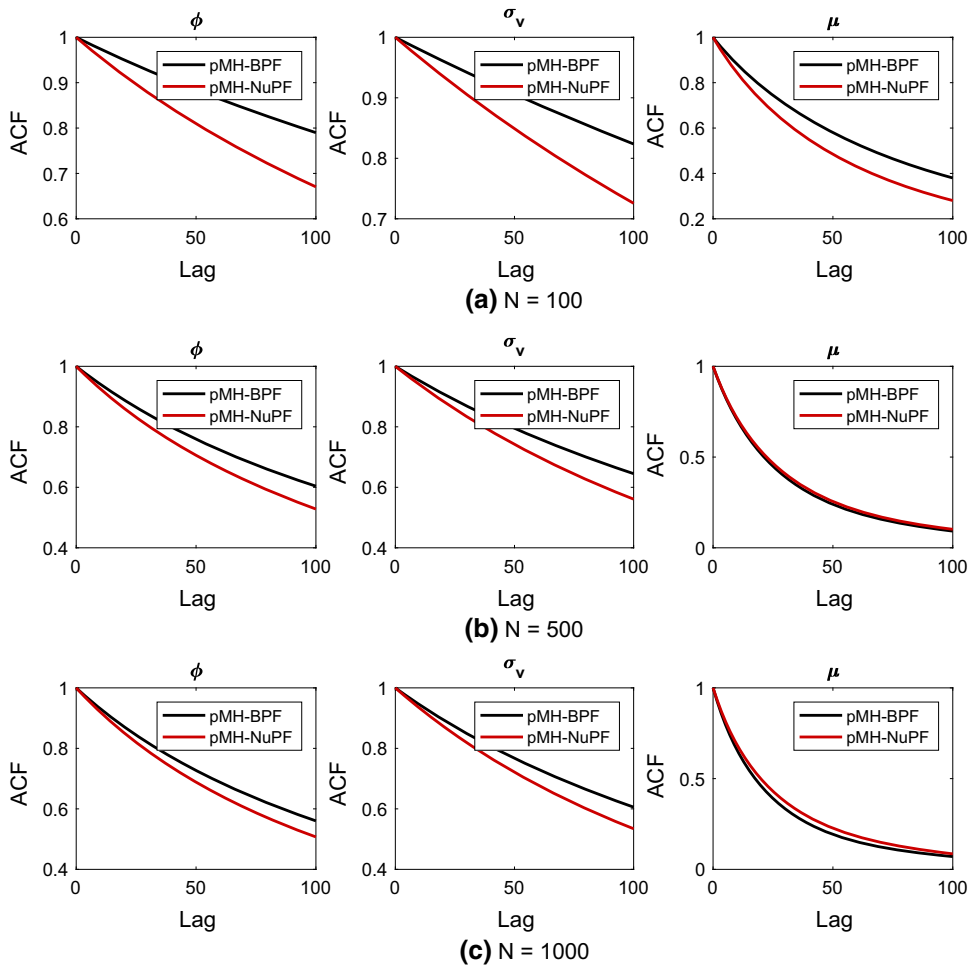

Fig. 11 Empirical autocorrelation functions (ACFs) computed for the pMH -BPF and the pMH-NuPF. From **a**–**c**, it can be seen that using the NuPF instead of BPF within the pMH causes faster autocorrelation decay. These results are obtained by averaging ACFs over 1000 Monte Carlo runs

labeled pMH-BPF and pMH-NuPF, respectively, hereafter. The parameter priors in the experiment are

$$p(\mu) = \mathcal{N}(0, 1), \qquad p(\sigma_v) = \mathcal{G}(2, 0.1),$$
$$p(\phi) = \mathcal{B}(120, 2),$$

where $\mathcal{G}(a, \theta)$ denotes the Gamma pdf with shape parameter a and scale parameter θ and $\mathcal{B}(\alpha, \beta)$ denotes the Beta pdf with shape parameters (α, β). Unlike Dahlin and Schön (2015), who use a truncated Gaussian prior centered on 0.95 with a small variance for ϕ, we use the Beta pdf, which is defined

on [0, 1], with mean $\alpha/(\alpha+\beta) = 0.9836$, which puts a significant probability mass on the interval [0.9, 1].

We have compared the pMH-BPF algorithm and the pMH-NuPF scheme (using a batch nudging procedure with $\gamma = 0.1$ and $M = \lfloor\sqrt{N}\rfloor$) by running 1000 independent Monte Carlo trials. We have computed the marginal likelihood estimates in the NuPF *after* the nudging step.

First, in order to illustrate the impact of the nudging on the parameter posteriors, we have run the pMH-NuPF and the pMH-BPF and obtained a long Markov chain (2×10^6 iterations) from both algorithms. Figure 9 displays the two-dimensional marginals of the resulting posterior distribution. It can be seen from Fig. 9 that the bias of the NuPF yields a perturbation compared to the posterior distribution approximated with the pMH-BPF. The discrepancy is small but noticeable for small N (see Fig. 9a for $N = 100$) and vanishes as we increase N (see Fig. 9b, c, for $N = 500$ and $N = 1000$, respectively). We observe that for a moderate number of particles, such as $N = 500$ in Fig. 9b, the error in the posterior distribution due to the bias in the NuPF is very slight.

Two common figures of merit for MCMC algorithms are the acceptance rate of the Markov kernel (desirably high) and the autocorrelation function of the chain (desirably low). Figure 10 shows the acceptance rates for the pMH-NuPF and the pMH-BPF algorithms with $N = 100$, $N = 500$ and $N = 1000$ particles in both PFs. It is observed that the use of nudging leads to noticeably higher acceptance rates, although the difference becomes smaller as N increases.

Figure 11 displays the average autocorrelation functions (ACFs) of the chains obtained in the 1000 independent simulations. We see that the autocorrelation of the chains produced by the pMH-NuPF method decays more quickly than the autocorrelation of the chains output by the conventional pMH-BPF, especially for lower values of N. Even for $N = 1000$ (which ensures an almost negligible perturbation of the posterior distribution, as shown in Fig. 9c), there is an improvement in the ACFs of the parameters ϕ and σ_v using the NuPF. Less correlation can be expected to translate into better estimates as well for a fixed length of the chain.

7 Conclusions

We have proposed a simple modification of the particle filter which, according to our computer experiments, can improve the performance of the algorithm (e.g., when tracking high-dimensional systems) or enhance its robustness to model mismatches in the state equation of a SSM. The modification of the standard particle filtering scheme consists of an additional step, which we term nudging, in which a subset of particles are pushed toward regions of the state space with a higher likelihood. In this way, the state space can be explored more efficiently while keeping the computational effort at nearly the same level as in a standard particle filter. We refer to the new algorithm as the "nudged particle filter" (NuPF). While, for clarity and simplicity, we have kept the discussion and the numerical comparisons restricted to the modification (nudging) of the conventional BPF, the new step can be naturally incorporated to most known particle filtering methods.

We have presented a basic analysis of the NuPF which indicates that the algorithm converges (in L_p) with the same error rate as the standard particle filter. In addition, we have also provided a simple reinterpretation of nudging that illustrates why the NuPF tends to outperform the BPF when there is some mismatch in the state equation of the SSM. To be specific, we have shown that, given a fixed sequence of observations, the NuPF amounts to a standard PF for a modified dynamical model which empirically leads to a higher model evidence (i.e., a higher likelihood) compared to the original SSM.

The analytical results have been supplemented with a number of computer experiments, with both synthetic and real data. In the latter case, we have tackled the fitting of a stochastic volatility SSM using Bayesian methods for model inference and a time-series dataset consisting of euro-to-US-dollar exchange rates over a period of two years. We have shown how different figures of merit (model evidence, acceptance probabilities or autocorrelation functions) improve when using the NuPF, instead of a standard BPF, in order to implement a nested particle filter (Crisan and Miguez 2018) and a particle Metropolis–Hastings (Andrieu et al. 2010) algorithm.

Since the nudging step is fairly general, it can be used with a wide range of differentiable or non-differentiable likelihoods. Besides, the new operation does not require any modification of the well-defined steps of the PF so it can be plugged into a variety of common particle filtering methods. Therefore, it can be adopted by a practitioner with hardly any additional effort. In particular, gradient-nudging steps (for differentiable log-likelihoods) can be implemented using automatic differentiation tools, currently available in many software packages, hence relieving the user from explicitly calculating the gradient of the likelihood.

Similar to the resampling step, which is routinely employed for numerical stability, we believe the nudging step can be systematically used for improving the performance and robustness of particle filters.

A Proof of Theorem 1

In order to prove Theorem 1, we need a preliminary lemma, which can be found, e.g., in Crisan (2001).

Lemma 2 *Let $\alpha, \beta, \bar{\alpha}, \bar{\beta} \in \mathcal{P}(\mathsf{X})$ be probability measures and $f, h \in B(\mathsf{X})$ be two real bounded functions on X such that $(h, \bar{\alpha}) > 0$ and $(h, \bar{\beta}) > 0$. If the identities,*

$$(f, \alpha) = \frac{(fh, \bar{\alpha})}{(h, \bar{\alpha})} \quad \text{and} \quad (f, \beta) = \frac{(fh, \bar{\beta})}{(h, \bar{\beta})}$$

hold, then we have,

$$\begin{aligned}|(f, \alpha) - (f, \beta)| \leq \frac{1}{(h, \bar{\alpha})} \left|(fh, \bar{\alpha}) - (fh, \bar{\beta})\right| \\ + \frac{\|f\|_\infty}{(h, \bar{\alpha})} \left|(h, \bar{\alpha}) - (h, \bar{\beta})\right|.\end{aligned}$$

Now we proceed with the proof of Theorem 1. We follow the same kind of induction argument as in, e.g., Del Moral and Miclo (2000) and Crisan and Miguez (2018).

For the base case, i.e., $t = 0$, we draw $x_0^{(i)}$, $i = 1, \ldots, N$, i.i.d. from π_0 and obtain,

$$\begin{aligned}&\|(\varphi, \pi_0^N) - (\varphi, \pi_0)\|_p \\ &= \mathbb{E}\left[\left|\frac{1}{N}\sum_{i=1}^N \left(\varphi(x_0^{(i)}) - (\varphi, \pi_0)\right)\right|^p\right]^{1/p}.\end{aligned}$$

We define $S_0^{(i)} = \varphi(x_0^{(i)}) - (\varphi, \pi_0)$ and note that $S_0^{(i)}$, $i = 1, \ldots, N$ are zero-mean and independent random variables. Using the Marcinkiewicz–Zygmund inequality (Shiryaev 1996), we arrive at,

$$\begin{aligned}\mathbb{E}\left[\left|\frac{1}{N}\sum_{i=1}^N S_0^{(i)}\right|^p\right] &\leq \frac{B_{0,p}}{N^p}\mathbb{E}\left[\left(\sum_{i=1}^N \left|S_0^{(i)}\right|^2\right)^{\frac{p}{2}}\right] \\ &\leq \frac{B_{0,p}}{N^p}\left(N 4\|\varphi\|_\infty^2\right)^{\frac{p}{2}},\end{aligned}$$

where $B_{0,p}$ is a constant independent of N and the last inequality follows from $\left|S_0^{(i)}\right| = \left|\varphi(x_0^{(i)}) - (\varphi, \pi_0)\right| \leq 2\|\varphi\|_\infty$. Therefore, we have proved that Eq. (C.6) holds for the base case,

$$\|(\varphi, \pi_0^N) - (\varphi, \pi_0)\|_p \leq \frac{c_{0,p}\|\varphi\|_\infty}{\sqrt{N}},$$

where $c_{0,p} = 2B_{0,p}^{1/p}$ is a constant independent of N.

The induction hypothesis is that, at time $t-1$,

$$\left\|(\varphi, \pi_{t-1}^N) - (\varphi, \pi_{t-1})\right\|_p \leq \frac{c_{t-1,p}\|\varphi\|_\infty}{\sqrt{N}}$$

for some constant $c_{t-1,p} < \infty$ independent of N.

We start analyzing the predictive measure ξ_t^N,

$$\xi_t^N(\mathrm{d}x) = \frac{1}{N}\sum_{i=1}^N \delta_{\bar{x}_t^{(i)}}(\mathrm{d}x),$$

where $\bar{x}_t^{(i)}$, $i = 1, \ldots, N$ are the particles sampled from the transition kernels $\tau_t^{x_{t-1}^{(i)}}(\mathrm{d}x_t) \triangleq \tau_t(\mathrm{d}x_t|x_{t-1}^{(i)})$. Since we have $\xi_t = \tau_t\pi_{t-1}$ (see Sect. 1.4), a simple triangle inequality yields,

$$\begin{aligned}\left\|(\varphi, \xi_t^N) - (\varphi, \xi_t)\right\|_p &= \left\|(\varphi, \xi_t^N) - (\varphi, \tau_t\pi_{t-1})\right\|_p \\ &\leq \left\|(\varphi, \xi_t^N) - (\varphi, \tau_t\pi_{t-1}^N)\right\|_p \\ &\quad + \left\|(\varphi, \tau_t\pi_{t-1}^N) - (\varphi, \tau_t\pi_{t-1})\right\|_p,\end{aligned} \tag{A.1}$$

where,

$$\left(\varphi, \tau_t\pi_{t-1}^N\right) = \frac{1}{N}\sum_{i=1}^N \left(\varphi, \tau_t^{x_{t-1}^{(i)}}\right). \tag{A.2}$$

For the sampling step, we aim at bounding the two terms on the rhs of (A.1).

For the first term, we introduce the σ-algebra generated by the random variables $x_{0:t}^{(i)}$ and $\bar{x}_{1:t}^{(i)}$, $i = 1, \ldots, N$, denoted $\mathcal{F}_t = \sigma(x_{0:t}^{(i)}, \bar{x}_{1:t}^{(i)}, i = 1, \ldots, N)$. Since π_{t-1}^N is measurable w.r.t. $\mathcal{F}_{t-1}$, we can write

$$\mathbb{E}[(\varphi, \xi_t^N)|\mathcal{F}_{t-1}] = \frac{1}{N}\sum_{i=1}^N \left(\varphi, \tau_t^{x_{t-1}^{(i)}}\right) = \left(\varphi, \tau_t\pi_{t-1}^N\right).$$

Next, we define the random variables $S_t^{(i)} = \varphi(\bar{x}_t^{(i)}) - (\varphi, \tau_t\pi_{t-1}^N)$ and note that, conditional on $\mathcal{F}_{t-1}$, $S_t^{(i)}$, $i = 1, \ldots, N$ are zero mean and independent. Then, the approximation error of ξ_t^N can be written as,

$$\begin{aligned}&\mathbb{E}[\left|(\varphi, \xi_t^N) - (\varphi, \tau_t\pi_{t-1}^N)\right|^p |\mathcal{F}_{t-1}] \\ &= \mathbb{E}\left[\left|\frac{1}{N}\sum_{i=1}^N S_t^{(i)}\right|^p \middle| \mathcal{F}_{t-1}\right].\end{aligned}$$

Resorting again to the Marcinkiewicz–Zygmund inequality, we can write,

$$\mathbb{E}\left[\left|\frac{1}{N}\sum_{i=1}^N S_t^{(i)}\right|^p \middle| \mathcal{F}_{t-1}\right]$$

$$\leq \frac{B_{t,p}}{N^p}\mathbb{E}\left[\left(\sum_{i=1}^{N}\left|S_t^{(i)}\right|^2\right)^{\frac{p}{2}}\middle|\mathcal{F}_{t-1}\right],$$

where $B_{t,p} < \infty$ is a constant independent of N. Moreover, since $\left|S_t^{(i)}\right| = \left|\varphi(\bar{x}_t^{(i)}) - (\varphi, \tau_t\pi_{t-1}^N)\right| \leq 2\|\varphi\|_\infty$, we have,

$$\mathbb{E}\left[\left|\frac{1}{N}\sum_{i=1}^{N}S_t^{(i)}\right|^p\middle|\mathcal{F}_{t-1}\right] \leq \frac{B_{t,p}}{N^p}\left(N4\|\varphi\|_\infty^2\right)^{\frac{p}{2}} = \frac{B_{t,p}}{N^{p/2}}2^p\|\varphi\|_\infty^p.$$

If we take unconditional expectations on both sides of the equation above, then we arrive at

$$\|(\varphi, \xi_t^N) - (\varphi, \tau_t\pi_{t-1}^N)\|_p \leq \frac{c_{1,p}\|\varphi\|_\infty}{\sqrt{N}}, \tag{A.3}$$

where $c_{1,p} = 2B_{t,p}^{1/p} < \infty$ is a constant independent of N.

To handle the second term in the rhs of (A.1), we define $(\bar{\varphi}, \pi_{t-1}) = (\varphi, \tau_t\pi_{t-1})$ where $\bar{\varphi} \in B(\mathsf{X})$ and given by,

$$\bar{\varphi}(x) = (\varphi, \tau_t^x).$$

We also write $(\bar{\varphi}, \pi_{t-1}^N) = (\varphi, \tau_t\pi_{t-1}^N)$. Since $\|\bar{\varphi}\|_\infty \leq \|\varphi\|_\infty$, the induction hypothesis leads,

$$\|(\varphi, \tau_t\pi_{t-1}^N) - (\varphi, \tau_t\pi_{t-1})\|_p = \|(\bar{\varphi}, \pi_{t-1}^N) - (\bar{\varphi}, \pi_{t-1})\|_p \leq \frac{c_{t-1,p}\|\varphi\|_\infty}{\sqrt{N}}, \tag{A.4}$$

where $c_{t-1,p}$ is a constant independent of N. Combining (A.1) and (A.4) yields,

$$\left\|(\varphi, \xi_t^N) - (\varphi, \xi_t)\right\|_p \leq \frac{c_{1,t,p}\|\varphi\|_\infty}{\sqrt{N}} \tag{A.5}$$

where $c_{1,t,p} = c_{t-1,p} + c_{1,p} < \infty$ is a constant independent of N.

Next, we have to bound the error between the predictive measure ξ_t^N and the nudged measure $\tilde{\xi}_t^N$. As the sets of samples $\{\bar{x}_t^{(i)}\}_{i=1}^N$, used to construct ξ_t^N, and $\{\tilde{x}_t^{(i)}\}_{i=1}^N$, used to construct $\tilde{\xi}_t^N$ as shown in (4.2), differ exactly in M particles, namely $\tilde{x}_t^{(j_1)}, \ldots, \tilde{x}_t^{(j_M)}$, where $\{j_1, \ldots, j_M\} = \mathcal{I}_t$, we readily obtain the relationship

$$\left\|(\varphi, \xi_t^N) - (\varphi, \tilde{\xi}_t^N)\right\|_p = \left\|\frac{1}{N}\sum_{i\in\mathcal{I}_t}\left(\varphi(\bar{x}_t^{(i)}) - \varphi(\tilde{x}_t^{(i)})\right)\right\|_p \leq \frac{2\|\varphi\|_\infty M}{N} \leq \frac{2\|\varphi\|_\infty}{\sqrt{N}} \tag{A.6}$$

where the first inequality holds trivially (since $|\varphi(x) - \varphi(x')| \leq 2\|\varphi\|_\infty$ for every $(x, x') \in \mathsf{X}^2$) and the second inequality follows from the assumption $M \leq \sqrt{N}$. Combining (A.5) and (C.8), we arrive at

$$\left\|(\varphi, \xi_t) - (\varphi, \tilde{\xi}_t^N)\right\|_p \leq \frac{\tilde{c}_{1,t}\|\varphi\|_\infty}{\sqrt{N}}, \tag{A.7}$$

where the constant $\tilde{c}_{1,t,p} = 2 + c_{1,t,p} < \infty$ is independent of N.

Next, we aim at bounding $\|(\varphi, \pi_t) - (\varphi, \tilde{\pi}_t^N)\|_p$ using (A.7). We note that, after the computation of weights, we define the weighted random measure,

$$\tilde{\pi}_t^N = \sum_{i=1}^{N} w_t^{(i)}\delta_{\tilde{x}_t^{(i)}} \quad \text{where} \quad w_t^{(i)} = \frac{g_t(\tilde{x}_t^{(i)})}{\sum_{i=1}^N g_t(\tilde{x}_t^{(i)})}.$$

The integrals computed with respect to the weighted measure $\tilde{\pi}_t^N$ takes the form,

$$(\varphi, \tilde{\pi}_t^N) = \frac{(\varphi g_t, \tilde{\xi}^N)}{(g_t, \tilde{\xi}_t^N)}. \tag{A.8}$$

On the other hand, using Bayes theorem, integrals with respect to the optimal filter can also be written in a similar form as,

$$(\varphi, \pi_t) = \frac{(\varphi g_t, \xi_t)}{(g_t, \xi_t)}. \tag{A.9}$$

Using Lemma 2 together with (A.8) and (A.9), we can readily obtain,

$$\left|(\varphi, \tilde{\pi}_t^N) - (\varphi, \pi_t)\right| \leq \frac{1}{(g_t, \xi_t)}\left(\|\varphi\|_\infty\left|(g_t, \xi_t) - (g_t, \xi_t^N)\right| + \left|(\varphi g_t, \xi_t) - (\varphi g_t, \xi_t^N)\right|\right), \tag{A.10}$$

where $(g_t, \xi_t) > 0$ by assumption. Using Minkowski's inequality, we can deduce from (A.10) that

$$\left\|(\varphi, \tilde{\pi}_t^N) - (\varphi, \pi_t)\right\|_p \leq \frac{1}{(g_t, \xi_t)}\left(\|\varphi\|_\infty\left\|(g_t, \xi_t) - (g_t, \xi_t^N)\right\|_p + \left\|(\varphi g_t, \xi_t) - (\varphi g_t, \xi_t^N)\right\|_p\right). \tag{A.11}$$

Noting that we have $\|\varphi g_t\|_\infty \leq \|\varphi\|_\infty\|g_t\|_\infty$, (A.7) and (A.11) together yield,

$$\left\|(\varphi, \pi_t) - (\varphi, \tilde{\pi}_t^N)\right\|_p \leq \frac{c_{2,t,p}\|\varphi\|_\infty}{\sqrt{N}}, \tag{A.12}$$

where

$$c_{2,t,p} = \frac{2\|g_t\|_\infty \tilde{c}_{1,t,p}}{(g_t, \xi_t)} < \infty$$

is a finite constant independent of N (the denominator is positive and the numerator is finite as a consequence of Assumption 1).

Finally, the analysis of the multinomial resampling step is also standard. We denote the resampled measure as π_t^N. Since the random variables which are used to construct π_t^N are sampled i.i.d from $\tilde{\pi}_t^N$, the argument for the base case can also be applied here to yield,

$$\left\|(\varphi, \tilde{\pi}_t^N) - (\varphi, \pi_t^N)\right\|_p \leq \frac{c_{3,t,p}\|\varphi\|_\infty}{\sqrt{N}}, \tag{A.13}$$

where $c_{3,t,p} < \infty$ is a constant independent of N. Combining bounds (A.12) and (A.13) to obtain inequality (C.6), with $c_{t,p} = c_{2,t,p} + c_{3,t,p} < \infty$, concludes the proof.

B Proof of Lemma 1

Since $\tilde{x}_t^{(i)} = \bar{x}_t^{(i)} + \gamma_t \nabla_{x_t} g_t(\bar{x}_t^{(i)})$, we readily obtain the relationships

$$\begin{aligned}\left|\varphi(\tilde{x}_t^{(i)}) - \varphi(\bar{x}_t^{(i)})\right| &\leq L\left\|\tilde{x}_t^{(i)} - \bar{x}_t^{(i)}\right\|_2 \\ &= L\gamma_t\left\|\nabla_{x_t} g(\bar{x}_t^{(i)})\right\|_2 \\ &\leq \gamma_t L G_t \end{aligned} \tag{B.1}$$

where the first inequality follows from the Lipschitz assumption, the identity is due to the implementation of the gradient-nudging step and the second inequality follows from Assumption 2. Then we bound the error $\|(\varphi, \xi_t^N) - (\varphi, \tilde{\xi}_t^N)\|_p$ as

$$\begin{aligned}\left\|(\varphi, \xi_t^N) - (\varphi, \tilde{\xi}_t^N)\right\|_p &= \left\|\frac{1}{N}\sum_{i\in\mathcal{I}_t}\left(\varphi(\bar{x}_t^{(i)}) - \varphi(\tilde{x}_t^{(i)})\right)\right\|_p \\ &\leq \frac{1}{N}\sum_{i\in\mathcal{I}}\left\|\varphi(\bar{x}_t^{(i)}) - \varphi(\tilde{x}_t^{(i)})\right\|_p \\ &\leq \frac{M}{N}\gamma_t L G_t \end{aligned} \tag{B.2}$$

where the identity is a consequence of the construction of $\mathcal{I}_t$ and we apply Minkowski's inequality, (B.1) and the assumption $|\mathcal{I}_t| = M$ to obtain (B.2). However, we have assumed that $\sup_{1\leq t\leq T}\gamma_t M \leq \sqrt{N}$, hence

$$\left\|(\varphi, \xi_t^N) - (\varphi, \tilde{\xi}_t^N)\right\|_p \leq \frac{L G_t}{\sqrt{N}}.$$

C Nudging scheme that increases the model evidence

Consider the SSM $\mathcal{M}_0 = \{\tau_0, \tau_t, g_t\}$ where the likelihoods $(g_t)_{t\geq 1}$ and the Markov kernels $(\tau_t)_{t\geq 1}$ satisfy the regularity assumptions below.

Assumption 3 The functions $\log g_t(x), t = 1, 2, \ldots$, are differentiable and the gradients $\nabla \log g_t(x)$ are Lipschitz with constant $\mathsf{L}_t^g < \infty$. To be specific,

$$\|\nabla \log g_t(x) - \nabla \log g_t(x')\|_2 \leq \mathsf{L}_t^g \|x - x'\|_2.$$

Assumption 4 The Markov kernels $\tau_t(\mathrm{d}x|x')$ are absolutely continuous with respect to the Lebesgue measure; hence, there are conditional pdf's $m_t(x|x')$ such that $\tau_t(\mathrm{d}x|x') = m_t(x|x')\mathrm{d}x$ for any $x' \in \mathsf{X}$. Moreover, the log-pdf's $\log m_t(x|x')$ are uniformly Lipschitz in x', i.e., there are non-negative bounded functions $L_t(x)$ such that

$$|\log m_t(x|x') - \log m_t(x|x'')| \leq L_t(x)\|x' - x''\|_2$$

and $\mathsf{L}_t^\tau = \sup_{x\in\mathsf{X}} L_t(x) < \infty$.

For any subset $A \subseteq \mathsf{X}$, let us introduce the indicator function

$$\mathbf{1}_A(x) := \begin{cases} 1, & \text{if } x \in A, \\ 0, & \text{otherwise.} \end{cases}$$

We construct the nudging operator $\alpha_t^{y_t} : \mathsf{X} \mapsto \mathsf{X}$ of the form

$$\alpha_t^{y_t}(x) := (x + \gamma_t \nabla \log g_t(x))\,\mathbf{1}_{S_{y_t}}(x) + x\mathbf{1}_{\overline{S_{y_t}}}(x), \tag{C.1}$$

where $S_{y_t} := \left\{x \in \mathsf{X} : \|\nabla \log g_t(x_t)\| \geq 2\mathsf{L}_t^\tau\right\}$ (recall that $g_t(x) = g_t(y_t|x)$), $\overline{S_{y_t}} = \mathsf{X}\backslash S_{y_t}$ is the complement of the set S_{y_t} and $\gamma_t > 0$ is small enough to guarantee that $g_t(x + \gamma_t \nabla \log g_t(x)) \geq g_t(x)$. Intuitively, this nudging scheme only takes a gradient step when the slope of the likelihood g_t is sufficient to insure an improvement of the likelihood with a small move of the state x.

Assume, for simplicity, that we apply the nudging operator (C.1) to every particle at every time step. The recursive step of the resulting NuPF can be outlined as follows:

1. For $i = 1, \ldots, N$,
   (a) Draw $\bar{x}_t^{(i)} \sim \tau_t(\mathrm{d}x_t|x_{t-1}^{(i)})$,
   (b) Nudge every particle, i.e., $\tilde{x}_t^{(i)} = \alpha_t^{y_t}(\bar{x}_t^{(i)})$,
   (c) And compute weights $w_t^{(i)} \propto g_t(\tilde{x}_t^{(i)})$.
2. Resample to obtain $\{x_t^{(i)}\}_{i=1,\ldots,N}$.

The asymptotic convergence of this algorithm can be insured whenever the step sizes γ_t are selected small enough to guarantee that $\sup_x \|x - \alpha_t^{y_t}(x)\| \leq \frac{1}{\sqrt{N}}$ (this is a consequence of Corollary 1 in Crisan and Miguez 2018). The implicit model for this NuPF is $\mathcal{M}_1 = \{\tau_0, \tilde{\tau}_t, g_t\}$ where the transition kernel is

$$\tilde{\tau}_t^{y_t}(\mathrm{d}x_t|x_{t-1}) = \int \delta_{\alpha_t^{y_t}(\bar{x})}(\mathrm{d}x_t)\tau_t(\mathrm{d}\bar{x}|x_{t-1}). \tag{C.2}$$

Note that for any integrable function $f : \mathsf{X} \mapsto \mathsf{X}$ we have

$$\begin{aligned}(f, \tilde{\tau}_t^{y_t})(x_{t-1}) &= \int\int f(x_t)\delta_{\alpha_t^{y_t}(\bar{x})}(\mathrm{d}x_t)\tau_t(\mathrm{d}\bar{x}|x_{t-1}) \\ &= \int (f \circ \alpha_t^{y_t})(\bar{x})\tau_t(\mathrm{d}\bar{x}|x_{t-1}) \\ &= (f \circ \alpha_t^{y_t}, \tau_t)(x_{t-1}),\end{aligned} \tag{C.3}$$

where $(f \circ \alpha_t^{y_t})(x) = f(\alpha_t^{y_t}(x))$ is the composition of f and $\alpha_t^{y_t}$. In particular, we note that $(g_t, \tilde{\tau}_t^{y_t})(x) = (g_t \circ \alpha_t^{y_t}, \tau_t)(x)$ and $(\mathbf{1}_\mathsf{X}, \tilde{\tau}_t^{y_t})(x) = \int \mathbf{1}_\mathsf{X}(\alpha_t^{y_t}(x))\tau_t(\mathrm{d}\bar{x}|x) = 1$ because $\alpha_t^{y_t}$ is $\mathsf{X} \mapsto \mathsf{X}$ and, therefore, $\mathbf{1}_\mathsf{X} \circ \alpha_t^{g_t} = 1$.

Let $\beta_t(x) := x + \gamma_t \nabla \log g_t(x)$. Assumptions 3 and 4 entail the following result.

Lemma 3 *If Assumptions 3 and 4 hold and the inequalities*

$$\gamma_t \leq \frac{1}{\mathsf{L}_t^g} \quad \textit{and} \quad \|\nabla \log g_t(x_t)\|_2 \geq 2\mathsf{L}_t^\tau,$$

are satisfied, then

$$\frac{(g_t \circ \beta_t)(x_t)}{g_t(x_t)} \geq \sup_{x_{t+1} \in \mathsf{X}} \frac{m_{t+1}(x_{t+1}|x_t)}{m_{t+1}(x_{t+1}|\beta_t(x_t))}. \tag{C.4}$$

Proof Assumption 3 implies that, for any pair $x, x' \in \mathsf{X}$,

$$\log g_t(x) \geq \log g_t(x') + \left\langle \nabla \log g_t(x'), x - x' \right\rangle - \frac{\mathsf{L}_t^g}{2}\|x - x'\|_2^2 \tag{C.5}$$

see, e.g., Bubeck et al. (2015, Lemma 3.4) for a proof. We can readily use (C.5) to obtain a lower bound for $\log g_t(\beta_t(x_t))$. Indeed,

$$\begin{aligned}\log g_t(\beta_t(x_t)) &\geq \log g_t(x_t) + \langle \nabla \log g_t(x_t), \gamma_t \nabla \log g_t(x_t)\rangle \\ &\quad - \frac{\mathsf{L}_t^g \gamma_t^2}{2}\|\nabla \log g_t(x_t)\|_2^2 \\ &= \log g_t(x_t) + \left(\gamma_t - \frac{\mathsf{L}_t^g \gamma_t^2}{2}\right)\|\nabla \log g_t(x_t)\|_2^2 \\ &\geq \log g_t(x_t) + \frac{\gamma_t}{2}\|\nabla \log g_t(x_t)\|_2^2\end{aligned} \tag{C.6}$$

where the last inequality follows from the assumption $\gamma_t \leq \frac{1}{\mathsf{L}_t^g}$. In turn, (C.6) implies

$$\frac{g_t(\beta_t(x_t))}{g_t(x_t)} \geq \exp\left\{\frac{\gamma_t}{2}\|\nabla \log g_t(x_t)\|_2^2\right\}. \tag{C.7}$$

We now turn to the problem of upper bounding the ratio of transition pdf's. From Assumption 4 and the definition of $\beta_t(x_t)$, we readily obtain that

$$\begin{aligned}\log m_{t+1}(x_{t+1}|x_t) &\quad - \log m_{t+1}(x_{t+1}|\beta_t(x_t)) \\ &\leq \mathsf{L}_t^\tau \gamma_t \|\nabla \log g_t(x_t)\|_2\end{aligned} \tag{C.8}$$

holds for any $x_{t+1} \in \mathsf{X}$. Taking exponentials on both sides of (C.8) we arrive at

$$\sup_{x_{t+1} \in \mathsf{X}} \frac{m_{t+1}(x_{t+1}|x_t)}{m_{t+1}(x_{t+1}|\beta_t(x_t))} \leq \exp\left\{\mathsf{L}_t^\tau \gamma_t \|\nabla \log g_t(x_t)\|_2\right\}. \tag{C.9}$$

If $\|\nabla \log g_t(x_t)\|_2 = 0$ then expressions (C.7) and (C.9) together readily yield desired relationship (C.4).

If $\|\nabla \log g_t(x_t)\|_2 > 0$, then the assumption $\|\nabla \log g_t(x_t)\| \geq 2\mathsf{L}_t^\tau$ implies that

$$\frac{\gamma_t}{2}\|\nabla \log g_t(x_t)\|_2^2 \geq \gamma_t \mathsf{L}_t^\tau \|\nabla \log g_t(x_t)\|_2$$

which, together with (C.7) and (C.9), again yield desired inequality (C.4). □

Finally, we prove that the evidence in favor of $\mathcal{M}_1$ is greater than the evidence in favor of $\mathcal{M}_0$.

Proposition 1 *Let the nudging scheme be defined as in* (C.1). *If Assumptions 3 and 4 hold and the inequality*

$$\gamma_t \leq \frac{1}{\mathsf{L}_t^g}$$

is satisfied for $t = 1, \ldots, T < \infty$, *then* $\mathsf{p}(y_{1:T}|\mathcal{M}_1) \geq \mathsf{p}(y_{1:T}|\mathcal{M}_0)$.

Proof From the definition of $\tilde{\tau}_t$ in (C.2) and ensuing relationship (C.3), the evidence of model $\mathcal{M}_1$ can be readily written down as

$$\begin{aligned}\mathsf{p}(y_{1:T}|\mathcal{M}_1) = \int \cdots \int g_T(\alpha_t^{y_T}(x_T)) \times \\ \times \prod_{t=1}^{T-1} m_{t+1}(x_{t+1}|\alpha_t^{y_t}(x_t))g_t(\alpha_t^{y_t}(x_t)) \\ \times m_1(x_1|x_0)m_0(x_0)\mathrm{d}x_0 \cdots \mathrm{d}x_T.\end{aligned} \tag{C.10}$$

It is apparent that $g_T(\alpha_T^{y_T}(x_T)) \geq g_T(x_T)$ for every x_T. Moreover, for any $x_t \in \mathsf{X}$, if $x_t \in S_{y_t}$ then

$$\|\nabla \log g_t(x_t)\| \geq 2\mathsf{L}_t^{\tau} \quad \text{and} \quad \alpha_t^{y_t}(x_t) = \beta_t(x_t);$$

hence, we can apply Lemma 3, which yields

$$m_{t+1}(x_{t+1}|\alpha_t^{y_t}(x_t))g_t(\alpha_t^{y_t}(x_t)) \geq m_{t+1}(x_{t+1}|x_t)g_t(x_t)$$

for every x_{t+1}. Alternatively, if $x_t \notin S_{y_t}$, then $\alpha_t^{y_t}(x_t) = x_t$ and, trivially,

$$m_{t+1}(x_{t+1}|\alpha_t^{y_t}(x_t))g_t(\alpha_t^{y_t}(x_t)) = m_{t+1}(x_{t+1}|x_t)g_t(x_t).$$

Therefore,

$$\begin{aligned}\mathsf{p}(y_{1:T}|\mathcal{M}_1) &\geq \int \cdots \int g_T(x_T) \prod_{t=1}^{T-1} m_{t+1}(x_{t+1}|x_t)g_t(x_t) \\ &\quad \times m_1(x_1|x_0)m_0(x_0)\mathrm{d}x_0 \cdots \mathrm{d}x_T \\ &= \mathsf{p}(y_{1:T}|\mathcal{M}_0).\end{aligned}$$

□

References

Ades, M., van Leeuwen, P.J.: An exploration of the equivalent weights particle filter. Q. J. R. Meteorol. Soc. **139**(672), 820–840 (2013)

Ades, M., van Leeuwen, P.J.: The equivalent-weights particle filter in a high-dimensional system. Q. J. R. Meteorol. Soc. **141**(687), 484–503 (2015)

Anderson, B.D.O., Moore, J.B.: Optimal Filtering. Prentice-Hall, Englewood Cliffs (1979)

Andrieu, C., Doucet, A., Holenstein, R.: Particle Markov chain Monte Carlo methods. J. R. Stat. Soc.: Ser. B (Stat. Methodol.) **72**(3), 269–342 (2010)

Atkins, E., Morzfeld, M., Chorin, A.J.: Implicit particle methods and their connection with variational data assimilation. Mon. Weather Rev. **141**(6), 1786–1803 (2013)

Bain, A., Crisan, D.: Fundamentals of Stochastic Filtering. Springer, Berlin (2009)

Bengtsson, T., Bickel, P., Li, B.: Curse-of-dimensionality revisited: collapse of the particle filter in very large scale systems. Probability and statistics: Essays in Honor of David A. Freedman, pp. 316–334. Institute of Mathematical Statistics, Beachwood (2008)

Bernardo, J.M., Smith, A.F.M.: Bayesian Theory. Wiley, New York (1994)

Bertsekas, D.P.: Dynamic Programming and Optimal Control, vol. I. Athena Scientific, Belmont (2001)

Bubeck, S., et al.: Convex optimization: algorithms and complexity. Found. Trends® Mach. Learn. **8**(3–4), 231–357 (2015)

Chopin, N.: Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference. Ann. Stat. **32**(6), 2385–2411 (2004)

Chorin, A.J., Tu, X.: Implicit sampling for particle filters. Proc. Natl. Acad. Sci. **106**(41), 17249–17254 (2009)

Chorin, A., Morzfeld, M., Tu, X.: Implicit particle filters for data assimilation. Commun. Appl. Math. Comput. Sci. **5**(2), 221–240 (2010)

Crisan, D.: Particle filters—a theoretical perspective. In: Doucet, A., de Freitas, N., Gordon, N. (eds.) Sequential Monte Carlo Methods in Practice. Statistics for Engineering and Information Science, pp. 17–41. Springer, New York (2001)

Crisan, D., Doucet, A.: A survey of convergence results on particle filtering. IEEE Trans. Signal Process. **50**(3), 736–746 (2002)

Crisan, D., Miguez, J.: Uniform convergence over time of a nested particle filtering scheme for recursive parameter estimation in state-space Markov models. Adv. Appl. Probab. **49**(4), 1170–1200 (2017)

Crisan, D., Miguez, J.: Nested particle filters for online parameter estimation in discrete-time state-space Markov models. Bernoulli **24**(4A), 3039–3086 (2018)

Dahlin, J., Schön, T.B.: Getting started with particle Metropolis–Hastings for inference in nonlinear dynamical models. arxiv:1511.01707 (2015)

Del Moral, P., Miclo, L.: Branching and interacting particle systems. Approximations of Feynman–Kac formulae with applications to non-linear filtering. In: Azéma J., Ledoux M., Émery M., Yor M. (eds) Séminaire de Probabilités XXXIV. Lecture Notes in Mathematics, Springer, Berlin, vol. **1729**, pp. 1–145 (2000)

Del Moral, P.: Feynman–Kac Formulae: Genealogical and Interacting Particle Systems with Applications. Springer, New York (2004)

Del Moral, P., Guionnet, A.: Central limit theorem for nonlinear filtering and interacting particle systems. Ann. Appl. Probab. **9**(2), 275–297 (1999)

Del Moral, P., Guionnet, A.: On the stability of interacting processes with applications to filtering and genetic algorithms. Ann. Inst. Henri Poincare (B) Probab. Stat. **37**(2), 155–194 (2001)

Douc, R., Moulines, E.: Limit theorems for weighted samples with applications to sequential Monte Carlo methods. Ann. Stat. **36**(5), 2344–2376 (2008)

Douc, R., Moulines, E., Olsson, J.: Optimality of the auxiliary particle filter. Probab. Math. Stat. **29**(1), 1–28 (2009)

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

Doucet, A., De Freitas, N., Gordon, N.: An introduction to sequential Monte Carlo methods. In: Doucet, A., de Freitas, N., Gordon, N. (eds.) Sequential Monte Carlo Methods in Practice. Statistics for Engineering and Information Science, pp. 3–14. Springer, New York (2001)

Gordon, N.J., Salmond, D.J., Smith, A.F.: Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In: IEEE Proceedings F (Radar and Signal Processing), vol. 140, pp. 107–113. IET (1993)

Hoke, J.E., Anthes, R.A.: The initialization of numerical models by a dynamic-initialization technique. Month. Weather Rev. **104**(12), 1551–1556 (1976)

Johansen, A.M., Doucet, A.: A note on auxiliary particle filters. Stat. Probab. Lett. **78**(12), 1498–1504 (2008)

Kitagawa, G.: Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. J. Comput. Graph. Stat. **5**(1), 1–25 (1996)

Künsch, H.R.: Recursive Monte Carlo filters: algorithms and theoretical analysis. Ann. Stat. **33**(5), 1983–2021 (2005)

Liu, J.S., Chen, R.: Sequential Monte Carlo methods for dynamic systems. J. Am. Stat. Assoc. **93**(443), 1032–1044 (1998)

Malanotte-Rizzoli, P., Holland, W.R.: Data constraints applied to models of the ocean general circulation. Part I: the steady case. J. Phys. Oceanogr. **16**(10), 1665–1682 (1986)

Malanotte-Rizzoli, P., Holland, W.R.: Data constraints applied to models of the ocean general circulation. Part II: the transient, eddy-resolving case. J. Phys. Oceanogr. **18**(8), 1093–1107 (1988)

Míguez, J., Crisan, D., Djurić, P.M.: On the convergence of two sequential Monte Carlo methods for maximum a posteriori sequence

estimation and stochastic global optimization. Stat. Comput. **23**(1), 91–107 (2013)
Oreshkin, B.N., Coates, M.J.: Analysis of error propagation in particle filters with approximation. Ann. Appl. Probab. **21**(6), 2343–2378 (2011)
Pitt, M.K., Shephard, N.: Filtering via simulation: auxiliary particle filters. J. Am. Stat. Assoc. **94**(446), 590–599 (1999)
Robert, C.P.: The Bayesian Choice. Springer, New York (2007)
Shiryaev, A.N.: Probability. Springer, Berlin (1996)
Snyder, C., Bengtsson, T., Bickel, P., Anderson, J.: Obstacles to high-dimensional particle filtering. Month. Weather Rev. **136**(12), 4629–4640 (2008)
Tsay, R.S.: Analysis of Financial Time Series. Wiley, New York (2005)
van Leeuwen, P.J.: Particle filtering in geophysical systems. Month. Weather Rev. **137**(12), 4089–4114 (2009)
van Leeuwen, P.J.: Nonlinear data assimilation in geosciences: an extremely efficient particle filter. Q. J. R. Meteorol. Soc. **136**(653), 1991–1999 (2010)
Zou, X., Navon, I., LeDimet, F.: An optimal nudging data assimilation scheme using parameter estimation. Q. J. R. Meteorol. Soc. **118**(508), 1163–1186 (1992)