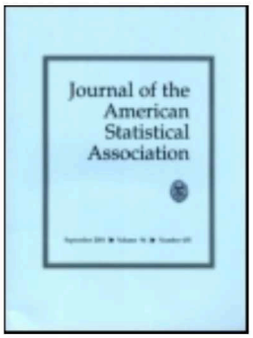


Journal of the American Statistical Association

ISSN: 0162-1459 (Print) 1537-274X (Online) Journal homepage: https://amstat.tandfonline.com/loi/uasa20

Smoothing with Couplings of Conditional Particle Filters

Pierre E. Jacob, Fredrik Lindsten & Thomas B. Schön

To cite this article: Pierre E. Jacob, Fredrik Lindsten & Thomas B. Schön (2019): Smoothing with Couplings of Conditional Particle Filters, Journal of the American Statistical Association, DOI: 10.1080/01621459.2018.1548856

To link to this article: https://doi.org/10.1080/01621459.2018.1548856

Smoothing with Couplings of Conditional Particle Filters

Pierre E. Jacob[1]*, Fredrik Lindsten[2] and Thomas B. Schön[2]

[1]Department of Statistics, Harvard University, USA.

[2]Department of Information Technology, Uppsala University, Sweden.

*The authors gratefully acknowledge the Swedish Foundation for Strategic Research (SSF) via the projects *Probabilistic Modeling and Inference for Machine Learning* (contract number: ICA16-0015) and ASSEMBLE (contract number: RIT15-0012), the Swedish Research Council (VR) via the projects *Learning of Large-Scale Probabilistic Dynamical Models* (contract number: 2016-04278) and *NewLEADS - New Directions in Learning Dynamical Systems* (contract number: 621-2016-06079), and the National Science Foundation through grant DMS-1712872.

Abstract

In state space models, smoothing refers to the task of estimating a latent stochastic process given noisy measurements related to the process. We propose an unbiased estimator of smoothing expectations. The lack-of-bias property has methodological benefits: independent estimators can be generated in parallel, and confidence intervals can be constructed from the central limit theorem to quantify the approximation error. To design unbiased estimators, we combine a generic debiasing technique for Markov chains, with a Markov chain Monte Carlo algorithm for smoothing. The resulting procedure is widely applicable and we show in numerical experiments that the removal of the bias comes at a manageable increase in variance. We establish the validity of the proposed estimators under mild assumptions. Numerical experiments are provided on toy models, including a setting of highly-informative observations, and for a realistic Lotka-Volterra model with an intractable transition density.

Keywords: couplings, particle filtering, particle smoothing, debiasing techniques, parallel computation.

1 Introduction

1.1 Goal and content

In state space models, the observations are treated as noisy measurements related to an underlying latent stochastic process. The problem of smoothing

refers to the estimation of trajectories of the underlying process given the observations (Cappé et al., 2005). For finite state spaces and linear Gaussian models, smoothing can be performed exactly. In general models, numerical approximations are required, and many state-of-the-art methods are based on particle methods (Douc et al., 2014; Kantas et al., 2015). Following this line of work, we propose a new method for smoothing in general state space models. Unlike existing methods, the proposed estimators are unbiased, which has direct benefits for parallelization and for the construction of confidence intervals.

The proposed method combines recently proposed conditional particle filters (Andrieu et al., 2010) with debiasing techniques for Markov chains (Glynn and Rhee, 2014). Specifically, we show in Section 2 how to remove the bias of estimators constructed with conditional particle filters, in exchange for an increase of variance; this variance can then be controlled with tuning parameters, and arbitrarily reduced by averaging over independent replicates. The validity of the proposed approach relies on the finiteness of the computational cost and of the variance of the proposed estimators, which we establish under mild conditions in Section 3. Methodological improvements are presented in Section 4, and comparisons with other smoothers in Section 5. Numerical experiments are provided in Section 6, and Section 7 concludes.

1.2 Smoothing in state space models

The latent stochastic process $(x_t)_{t\geq 0}$ takes values in $\mathbb{X}\subset\mathbb{R}^{d_x}$, and the observations $(y_t)_{t\geq 1}$ are in $\mathbb{Y}\subset\mathbb{R}^{d_y}$ for some $d_x, d_y \in \mathbb{N}$. A model specifies an initial distribution $m_0(dx_0 \mid \theta)$ and a transition kernel $f(dx_t \mid x_{t-1}, \theta)$ for the latent process. We will assume that we have access to deterministic functions *M* and *F*, and random variables U_t for $t \geq 0$, such that $M(U_0, \theta)$ follows $m_0(dx_0 \mid \theta)$ and $F(x_{t-1}, U_t, \theta)$ follows $f(dx_t \mid x_{t-1}, \theta)$; we refer to these as random function representations of the process (see Diaconis and Freedman, 1999). Conditionally upon the latent process, the observations are independent and their distribution is given by a measurement kernel $g(dy_t \mid x_t, \theta)$. The model is

parameterized by $\theta \in \Theta \subset \mathbb{R}^{d_\theta}$, for $d_\theta \in \mathbb{N}$. Filtering consists in approximating the distribution $p(dx_t \mid y_{1:t}, \theta)$ for all times $t \geq 1$, whereas smoothing refers to the approximation of $p(dx_{0:T} \mid y_{1:T}, \theta)$ for a fixed time horizon T, where for $s, t \in \mathbb{N}$, we write $s:t$ for the set $\{s, \ldots, t\}$, and $v_{s:t}$ for the vector $(v_s, \ldots, v_t)$. The parameter θ is hereafter fixed and removed from the notation, as is usually done in the smoothing literature (see Section 4 in Kantas et al., 2015); we discuss unknown parameters in Section 7. Denote by h a test function from $\mathbb{X}^{T+1}$ to $\mathbb{R}$, of which we want to compute the expectation with respect to the smoothing distribution $\pi(dx_{0:T}) = p(dx_{0:T} \mid y_{1:T})$; we write $\pi(h)$ for $\int_{\mathbb{X}^{T+1}} h(x_{0:T}) \pi(dx_{0:T})$. For instance, with $h : x_{0:T} \mapsto x_t$ where $t \in 0:T$, $\pi(h)$ is the smoothing expectation $\mathbb{E}[x_t \mid y_{1:T}]$.

Postponing a discussion on existing smoothing methods to Section 5, we first describe the conditional particle filter (CPF, Andrieu et al., 2010), which is a variant of the particle filter (Doucet et al., 2001). Given a "reference" trajectory $X = x_{0:T}$, a CPF generates a new trajectory $X' = x'_{0:T}$ as described in Algorithm 1, which defines a Markov kernel on the space of trajectories; we will write $x'_{0:T} \sim \mathrm{CPF}(x_{0:T}, \cdot)$. This Markov kernel leaves π invariant and ergodic averages of the resulting chains consistently estimate integrals with respect to π, under mild conditions (Andrieu et al., 2010; Chopin and Singh, 2015; Lindsten et al., 2015; Andrieu et al., 2018; Kuhlenschmidt and Singh, 2018; Lee et al., 2018). We denote by $(X^{(n)})_{n \geq 0}$ a chain starting from a path $X^{(0)}$, and iterating through $X^{(n)} \sim \mathrm{CPF}(X^{(n-1)}, \cdot)$ for $n \geq 1$.

Algorithm 1 Conditional particle filter (CPF), given a trajectory $x_{0:T}$ and θ.

At step $t = 0$.

1.1 Draw U_0^j and compute $x_0^j = M(U_0^j, \theta)$, for all $j = 1, \ldots, N-1$, and $x_0^N = x_0$.

1.2 Set $w_0^j = N^{-1}$, for $j = 1, \ldots, N$.

At step $t = 1, \ldots, T$.

2.1 Draw ancestors $a_{t-1}^{1:N-1} \sim r(da^{1:N-1} \mid w_{t-1}^{1:N})$, and set $a_{t-1}^{N} = N$.

2.2 Draw U_t^j and compute $x_t^j = F(x_{t-1}^{a_{t-1}^j}, U_t^j, \theta)$, for all $j = 1, \ldots, N-1$, and $x_t^N = x_t$.

2.3 Compute $w_t^j = g(y_t \mid x_t^j, \theta)$, for all $j = 1, \ldots, N$, and normalize the weights.

After the final step.

3.1 Draw b_T from a discrete distribution on $1:N$, with probabilities $w_T^{1:N}$.

3.2 For $t = T-1, \ldots, 0$, set $b_t = a_t^{b_{t+1}}$.

Return $x_{0:T}' = (x_0^{b_0}, \ldots, x_T^{b_T})$.

In step 2.1. of Algorithm 1, the resampling distribution $r(da^{1:N-1} \mid w^{1:N})$ refers to a distribution on $\{1, \ldots, N\}^{N-1}$ from which "ancestors" are drawn according to particle weights. The resampling distribution is an algorithmic choice; specific schemes for the conditional particle filter are described in Chopin and Singh (2015). Here we will use multinomial resampling throughout. In step 2.3., "normalize the weights" means dividing them by their sum. Instead of bootstrap particle filters (Gordon et al., 1993), where particles are propagated from the model transition, more sophisticated filters can readily be used in the CPF procedure. For instance, performance gains can be obtained with auxiliary particle filters (Pitt and Shephard, 1999; Johansen and Doucet, 2008), as illustrated in Section 6.1. In presenting algorithms we focus on bootstrap particle filters for simplicity. When the transition density is tractable, extensions of the CPF include backward sampling (Whiteley, 2010; Lindsten and Schön, 2013) and ancestor sampling (Lindsten et al., 2014), which is beneficial in the proposed approach as illustrated in Section 6.1. The complexity of a standard CPF update is of order *NT*, and the memory requirements are of order $T + N \log N$ (Jacob et al., 2015).

The proposed method relies on CPF kernels but is different from Markov chain Monte Carlo (MCMC) estimators: it involves independent copies of unbiased estimators of $\pi(h)$. Thus it will be amenable to parallel computation and confidence intervals will be constructed in a different way than with standard MCMC output (e.g. Chapter 7 in Gelman et al., 2010); see Section 5 for a comparison with existing smoothers.

1.3 Debiasing Markov chains

We briefly recall the debiasing technique of Glynn and Rhee (2014), see also McLeish (2011); Rhee and Glynn (2012); Vihola (2017) and references therein. Denote by $(X^{(n)})_{n\geq 0}$ and $(\tilde{X}^{(n)})_{n\geq 0}$ two Markov chains with invariant distribution π, initialized from a distribution π_0. Assume that, for all $n \geq 0$, $X^{(n)}$ and $\tilde{X}^{(n)}$ have the same marginal distribution, and that $\lim_{n\to\infty} \mathbb{E}[h(X^{(n)})] = \pi(h)$. Writing limit as a telescopic sum, and swapping infinite sum and expectation, which will be justified later on, we obtain

$$\pi(h) = \mathbb{E}[h(X^{(0)})] + \sum_{n=1}^{\infty} \mathbb{E}[h(X^{(n)}) - h(\tilde{X}^{(n-1)})] = \mathbb{E}[h(X^{(0)}) + \sum_{n=1}^{\infty} (h(X^{(n)}) - h(\tilde{X}^{(n-1)}))].$$

Then, if it exists, the random variable $H_0 = h(X^{(0)}) + \sum_{n=1}^{\infty} (h(X^{(n)}) - h(\tilde{X}^{(n-1)}))$, is an unbiased estimator of $\pi(h)$. Furthermore, if the chains are coupled in such a way that there exists a time τ, termed the *meeting time*, such that $X^{(n)} = \tilde{X}^{(n-1)}$ almost surely for all $n \geq \tau$, then H_0 can be computed as

$$H_0 = h(X^{(0)}) + \sum_{n=1}^{\tau-1} (h(X^{(n)}) - h(\tilde{X}^{(n-1)})). \qquad (1)$$

We refer to H_0 as a Rhee–Glynn estimator. Given that the cost of producing H_0 increases with τ, it will be worth keeping in mind that we would prefer τ to take small values with large probability. The main contribution of the present article is to couple CPF chains and to use them in a Rhee–Glynn estimation procedure. Section 3 provides guarantees on the cost and the variance of H_0

under mild conditions, and Section 4 contains alternative estimators with reduced variance and practical considerations.

2 Unbiased smoothing

2.1 Coupled conditional particle filters

Our goal is to couple CPF chains $(X^{(n)})_{n\geq 0}$ and $(\tilde{X}^{(n)})_{n\geq 0}$ such that the meeting time has finite expectation, in order to enable a Rhee–Glynn estimator for smoothing. A coupled conditional particle filter (CCPF) is a Markov kernel on the space of pairs of trajectories, such that $(X', \tilde{X}') \sim \mathrm{CCPF}((X, \tilde{X}), \cdot)$ implies that $X' \sim \mathrm{CPF}(X, \cdot)$ and $\tilde{X}' \sim \mathrm{CPF}(\tilde{X}, \cdot)$.

Algorithm 2 describes CCPF in pseudo-code, conditional upon $X = x_{0:T}$ and $\tilde{X} = \tilde{x}_{0:T}$. Two particle systems are initialized and propagated using common random numbers. The resampling steps and the selection of trajectories at the final step are performed jointly using couplings of discrete distributions. To complete the description of the CCPF procedure, we thus need to specify these couplings (for steps 2.1. and 3.1. in Algorithm 2). With the Rhee–Glynn estimation procedure in mind, we aim at achieving large meeting probabilities $\mathbb{P}(X' = \tilde{X}' \mid X, \tilde{X})$, so as to incur short meeting times on average.

Algorithm 2 Coupled conditional particle filter (CCPF), given trajectories $x_{0:T}$ and $\tilde{x}_{0:T}$.

At step $t = 0$.

1.1 Draw U_0^j, compute $x_0^j = M(U_0^j, \theta)$ and $\tilde{x}_0^j = M(U_0^j, \theta)$ for $j = 1, \ldots, N-1$.

1.2 Set $x_0^N = x_0$ and $\tilde{x}_0^N = \tilde{x}_0$.

1.3 Set $w_0^j = N^{-1}$ and $\tilde{w}_0^j = N^{-1}$, for $j = 1, \ldots, N$.

At step $t = 1, \ldots, T$.

2.1 Draw ancestors $a_{t-1}^{1:N}$ and $\tilde{a}_{t-1}^{1:N}$ from a coupling of $r(da^{1:N-1} \mid w_{t-1}^{1:N})$ and $r(da^{1:N-1} \mid \tilde{w}_{t-1}^{1:N})$, and set $a_{t-1}^{N} = N$ and $\tilde{a}_{t-1}^{N} = N$.

2.2 Draw U_t^j, and compute $x_t^j = F(x_{t-1}^{a_{t-1}^j}, U_t^j, \theta)$ and $\tilde{x}_t^j = F(\tilde{x}_{t-1}^{\tilde{a}_{t-1}^j}, U_t^j, \theta)$, for all $j = 1, \ldots, N-1$. Set $x_t^N = x_t$ and $\tilde{x}_t^N = \tilde{x}_t$.

2.3 Compute $w_t^j = g(y_t \mid x_t^j, \theta)$ and $\tilde{w}_t^j = g(y_t \mid \tilde{x}_t^j, \theta)$, for all $j = 1, \ldots, N$, and normalize the weights.

After the final step.

3.1 Draw $(b_T, \tilde{b}_T)$ from a coupling of two distributions on $1:N$, with probabilities $w_T^{1:N}$ and $\tilde{w}_T^{1:N}$ respectively.

3.2 For $t = T-1, \ldots, 0$, set $b_t = a_t^{b_{t+1}}$ and $\tilde{b}_t = \tilde{a}_t^{\tilde{b}_{t+1}}$.

Return $x_{0:T}' = (x_0^{b_0}, \ldots, x_T^{b_T})$ and $\tilde{x}_{0:T}' = (\tilde{x}_0^{\tilde{b}_0}, \ldots, \tilde{x}_T^{\tilde{b}_T})$.

2.2 Coupled resampling

The temporal index t is momentarily removed from the notation: the task is that of sampling pairs $(a, \tilde{a})$ such that $\mathbb{P}(a = j) = w^j$ and $\mathbb{P}(\tilde{a} = j) = \tilde{w}^j$ for all $j \in 1:N$; this is a sufficient condition for CPF kernels to leave π invariant (Andrieu et al., 2010).

A joint distribution on $\{1, \ldots, N\}^2$ is characterized by a matrix P with non-negative entries P^{ij}, for $i, j \in \{1, \ldots, N\}$, that sum to one. The value P^{ij} represents the probability of the event $(a, \tilde{a}) = (i, j)$. We consider the set $\mathcal{J}(w, \tilde{w})$ of matrices P such that $P1 = w$ and $P^{\mathsf{T}}1 = \tilde{w}$, where 1 denotes a column vector of N ones, $w = w^{1:N}$ and $\tilde{w} = \tilde{w}^{1:N}$. Matrices $P \in \mathcal{J}(w, \tilde{w})$ are such that $\mathbb{P}(a = j) = w^j$ and $\mathbb{P}(\tilde{a} = j) = \tilde{w}^j$ for $j \in 1:N$.

Any choice of probability matrix $P \in \mathcal{J}(w, \tilde{w})$, and of a way of sampling $(a, \tilde{a}) \sim P$, leads to a *coupled* resampling scheme. In order to keep the

complexity of sampling N pairs from P linear in N, we focus on a particular choice. Other choices of coupled resampling schemes are given in Deligiannidis et al. (2018); Jacob et al. (2016); Sen et al. (2018), following earlier works such as Pitt (2002); Lee (2008).

We consider the *index-coupled* resampling scheme, used by Chopin and Singh (2015) in their theoretical analysis of the CPF, and by Jasra et al. (2017) in a multilevel Monte Carlo context, see also Section 2.4 in Jacob et al. (2016). The scheme amounts to a maximal coupling of discrete distributions on $\{1,\ldots,N\}$ with probabilities $w^{1:N}$ and $\tilde{w}^{1:N}$, respectively. This coupling maximizes the probability of the event $\{a=\tilde{a}\}$ under the marginal constraints. How to sample from a maximal coupling of discrete distributions is described e.g. in Lindvall (2002). The scheme is intuitive at the initial step of the CCPF, when $x_0^j = \tilde{x}_0^j$ for all $j=1,\ldots,N-1$: one would want pairs of ancestors $(a_0,\tilde{a}_0)$ to be such that $a_0=\tilde{a}_0$, so that pairs of resampled particles remain identical. At later steps, the number of identical pairs across both particle systems might be small, or even null. In any case, at step 2.2. of Algorithm 2, the same random number U_t^j is used to compute x_t^j and $\tilde{x}_t^j$ from their ancestors. If $a_{t-1}^j = \tilde{a}_{t-1}^j$, we select ancestor particles that were, themselves, computed with common random numbers at the previous step, and we give them common random numbers again. Thus this scheme maximizes the number of consecutive steps at which common random numbers are used to propagate each pair of particles.

We now discuss why propagating pairs of particles with common random numbers might be desirable. Under assumptions on the random function representation of the latent process, using common random numbers to propagate pairs of particles results in the particles contracting. For instance, in an auto-regressive model where $F(x,U,\theta)=\theta x+U$, where $\theta\in(-1,1)$ and U is the innovation term, we have $|F(x,U,\theta)-F(\tilde{x},U,\theta)|=|\theta||x-\tilde{x}|$, thus a pair of particles propagated with common variables U contracts at a geometric rate. We can formulate assumptions directly on the function $x\mapsto\mathbb{E}_U[F(x,U,\theta)]$, such as Lipschitz conditions with respect to x, after having integrated U out,

for fixed θ. Discussions on these assumptions can be found in Diaconis and Freedman (1999), and an alternative method that would not require them is mentioned in Section 7.

2.3 Rhee–Glynn smoothing estimator

We now put together the Rhee–Glynn estimator of Section 1.3 with the CCPF algorithm of Section 2.1. In passing we generalize the Rhee–Glynn estimator slightly by starting the telescopic sum at index $k \geq 0$ instead of zero, and denote it by H_k; k becomes a tuning parameter, discussed in Section 4. The procedure is fully described in Algorithm 3; CPF and CCPF refer to Algorithms 1 and 2 respectively.

By convention the sum from $k+1$ to $\tau-1$ in the definition of H_k is set to zero whenever $k+1 > \tau-1$. Thus the estimator H_k is equal to $h(X^{(k)})$ on the event $\{k+1 > \tau-1\}$. Recall that $h(X^{(k)})$ is in general a biased estimator of $\pi(h)$, since there is no guarantee that a CPF chain reaches stationarity within k iterations. Thus the term $\sum_{n=k+1}^{\tau-1} (h(X^{(n)}) - h(\tilde{X}^{(n-1)}))$ acts as a bias correction.

Algorithm 3 Rhee–Glynn smoothing estimator, with initial π_0 and tuning parameter k.

---

1. Draw $X^{(0)} \sim \pi_0$, $\tilde{X}^{(0)} \sim \pi_0$ and draw $X^{(1)} \sim \mathrm{CPF}(X^{(0)}, \cdot)$.
2. Set $n = 1$. While $n < \max(k, \tau)$, where $\tau = \inf\{n \geq 1 : X^{(n)} = \tilde{X}^{(n-1)}\}$,
   2.1 Draw $(X^{(n+1)}, \tilde{X}^{(n)}) \sim \mathrm{CCPF}((X^{(n)}, \tilde{X}^{(n-1)}), \cdot)$.
   2.2 Set $n \leftarrow n+1$.
3. Return $H_k = h(X^{(k)}) + \sum_{n=k+1}^{\tau-1} (h(X^{(n)}) - h(\tilde{X}^{(n-1)}))$.

At step 1. of Algorithm 3, the paths $X^{(0)}$ and $\tilde{X}^{(0)}$ can be sampled independently or not from π_0. In the experiments we will initialize chains independently and π_0 will refer to the distribution of a path randomly chosen among the trajectories of a particle filter.

3 Theoretical properties

We give three sufficient conditions for the validity of Rhee–Glynn smoothing estimators.

Assumption 1. *The measurement density of the model is bounded from above: there exists* $\overline{g} < \infty$ *such that, for all* $y \in \mathbb{Y}$ *and* $x \in \mathbb{X}, g(y \mid x) \leq \overline{g}$.

Assumption 2. *The resampling probability matrix P, with rows summing to* $w^{1:N}$ *and columns summing to* $\tilde{w}^{1:N}$, *is such that, for all* $i \in \{1, \ldots, N\}, P^{ii} \geq w^i \tilde{w}^i$. *Furthermore, if* $w^{1:N} = \tilde{w}^{1:N}$, *then P is a diagonal matrix with entries given by* $w^{1:N}$.

Assumption 3. *Let* $(X^{(n)})_{n \geq 0}$ *be a Markov chain generated by the conditional particle filter and started from* π_0, *and h a test function of interest. Then* $\mathbb{E}\left[h(X^{(n)})\right] \xrightarrow[n \to \infty]{} \pi(h)$. *Furthermore, there exists* $\delta > 0, n_0 < \infty$ *and* $C < \infty$ *such that, for all* $n \geq n_0, \mathbb{E}\left[h(X^{(n)})^{2+\delta}\right] \leq C$.

The first assumption is satisfied for wide classes of models where the measurements are assumed to be some transformation of the latent process with added noise. However, it would not be satisfied for instance in stochastic volatility models where it is often assumed that $Y \mid X = x \sim \mathcal{N}(0, \exp(x)^2)$ or variants thereof (e.g. Fulop and Li, 2013). There, the measurement density would diverge when y is exactly zero and $x \to -\infty$. A similar assumption is discussed in Section 3 of Whiteley (2013). One can readily check that the second assumption always holds for the index-coupled resampling scheme. The third assumption relates to the validity of MCMC estimators generated by the CPF algorithm, addressed under general assumptions in Chopin and Singh (2015); Lindsten et al. (2015); Andrieu et al. (2018).

Our main result states that the proposed estimator is unbiased, has a finite variance, and that the meeting time τ has tail probabilities bounded by those of a geometric variable, which implies in particular that the estimator has a finite expected cost.

Theorem 3.1. *Under Assumptions 1 and 2, for any initial distribution π_0, any number of particles $N \geq 2$ and time horizon $T \geq 1$, there exists $\varepsilon > 0$, which might depend on N and T, such that for all $n \geq 2$,*

$$\mathbb{P}(\tau > n) \leq (1-\varepsilon)^{n-1},$$

and therefore $\mathbb{E}[\tau] < \infty$. Under the additional Assumption 3, the Rhee–Glynn smoothing estimator H_k of Algorithm 3 is such that, for any $k \geq 0$, $\mathbb{E}[H_k] = \pi(h)$ and $\mathbb{V}[H_k] < \infty$.

The proof is in the supplementary materials. Some aspects of the proof, not specific to the smoothing setting, are similar to the proofs of Theorem 1 in Rhee (2013), Theorem 2.1 in McLeish (2011), Theorem 7 in Vihola (2017), and results in Glynn and Rhee (2014). It is provided in univariate notation but the Rhee–Glynn smoother can estimate multivariate smoothing functionals, in which case the theorem applies component-wise.

4 Improvements and tuning

Since H_ℓ is unbiased for all $\ell \geq 0$, we can compute H_ℓ for various values of ℓ between two integers $k \leq m$, and average these estimators to obtain $H_{k:m}$ defined as

$$\begin{aligned} H_{k:m} &= \frac{1}{m-k+1}\sum_{n=k}^{m}\{h(X^{(n)}) + \sum_{\ell=n+1}^{\tau-1}(h(X^{(\ell)}) - h(\tilde{X}^{(\ell-1)}))\} \\ &= \frac{1}{m-k+1}\sum_{n=k}^{m} h(X^{(n)}) + \sum_{n=k+1}^{\tau-1}\frac{\min(m-k+1, n-k)}{m-k+1}(h(X^{(n)}) - h(\tilde{X}^{(n-1)})). \end{aligned} \tag{2}$$

The term $(m-k+1)^{-1}\sum_{n=k}^{m} h(X^{(n)})$ is a standard ergodic average of a CPF chain, after m iterations and discarding the first $k-1$ steps as burn-in. It is a biased

estimator of $\pi(h)$ in general since π_0 is different from π. The other term acts as a bias correction. On the event $\tau - 1 < k + 1$ the correction term is equal to zero.

As k increases the bias of the term $(m-k+1)^{-1}\sum_{n=k}^{m} h(X^{(n)})$ decreases. The variance inflation of the Rhee–Glynn estimator decreases too, since the correction term is equal to zero with increasing probability. On the other hand, it can be wasteful to set k to an overly large value, in the same way that it is wasteful to discard too many iterations as burn-in when computing MCMC estimators. In practice we propose to choose k according to the distribution of τ, which can be sampled from exactly by running Algorithm 3, as illustrated in the numerical experiments of Section 6. Conditional upon a choice of k, by analogy with MCMC estimators we can set m to a multiple of k, such as $2k$ or $5k$. Indeed the proportion of discarded iterations is approximately k / m, and it appears desirable to keep this proportion low. We stress that the proposed estimators are unbiased and with a finite variance for any choice of k and m; tuning k and m only impacts variance and cost.

For a given choice of k and m, the estimator $H_{k:m}$ can be sampled R times independently in parallel. We denote the independent copies by $H_{k:m}^{(r)}$ for $r \in 1:R$. The smoothing expectation of interest $\pi(h)$ can then be approximated by $\bar{H}_{k:m}^{R} = R^{-1}\sum_{r=1}^{R} H_{k:m}^{(r)}$, with a variance that decreases linearly with R. From the central limit theorem the confidence interval $[\bar{H}_{k:m}^{R} + z_{\alpha/2}\hat{\sigma}^{R} / \sqrt{R}, \bar{H}_{k:m}^{R} + z_{1-\alpha/2}\hat{\sigma}^{R} / \sqrt{R}]$, where $\hat{\sigma}^{R}$ is the empirical standard deviation of $(H_{k:m}^{(r)})_{r=1}^{R}$ and z_a is the a-th quantile of a standard Normal distribution, has $1-\alpha$ asymptotic coverage as $R \to \infty$. The central limit theorem is applicable as a consequence of Theorem 3.1.

The variance of the proposed estimator can be further reduced by Rao–Blackwellization. In Eq. (2), the random variable $h(X^{(n)})$ is obtained by applying the test function h of interest to a trajectory drawn among N

trajectories, denoted by say $x_{0:T}^k$ for $k = 1, \ldots, N$, with probabilities $w_T^{1:N}$; see step 3 in Algorithms 1 and 2. Thus the random variable $\sum_{k=1}^{N} w_T^k h(x_{0:T}^k)$ is the conditional expectation of $h(X^{(n)})$ given the trajectories and $w_T^{1:N}$, which has the same expectation as $h(X^{(n)})$. Thus any term $h(X^{(n)})$ or $h(\tilde{X}^{(n)})$ in $H_{k:m}$ can be replaced by similar conditional expectations. This enables the use of all the paths generated by the CPF and CCPF kernels, and not only the selected ones.

As in other particle methods the choice of the number of particles N is important. Here, the estimator $\bar{H}_{k:m}^R$ is consistent as $R \to \infty$ for any $N \geq 2$, but N plays a role both on the cost and of the variance of each $H_{k:m}^{(r)}$. We can generate unbiased estimators for different values of N and compare their costs and variances in preliminary runs. The scaling of N with the time horizon T is explored numerically in Section 6.1. If possible, one can also employ other algorithms than the bootstrap particle filter, as illustrated in Section 6.1 with the auxiliary particle filter.

5 Comparison with existing smoothers

The proposed method combines elements from both particle smoothers and MCMC methods, but does not belong to either category. We summarize advantages and drawbacks below, after having discussed the cost of the proposed estimators.

Each estimator $H_{k:m}$ requires two draws from π_0, here taken as the distribution of a trajectory selected from a particle filter with N particles. Then, the estimator as described in Algorithm 3 requires a draw from the CPF kernel, $\tau - 1$ draws from the CCPF kernel, and finally $m - \tau$ draws of the CPF kernel on the events $\{m > \tau\}$. The cost of a particle filter and of an iteration of CPF is usually dominated by the propagation of N particles and the evaluation of their weights. The cost of an iteration of CCPF is approximately twice larger. Overall the cost of $H_{k:m}$ is thus of order

$C(\tau, m, N) = N \times (3 + 2(\tau - 1) + \max(0, m - \tau))$, for fixed T. The finiteness of the expected cost $\mathbb{E}[C(\tau, m, N)]$ is a consequence of Theorem 3.1. The average $\bar{H}^R_{k:m}$ satisfies a central limit theorem parametrized by the number of estimators R, as discussed in Section 4; however, since the cost of $H_{k:m}$ is random, it might be more relevant to consider central limit theorems parametrized by computational cost, as in Glynn and Whitt (1992). The asymptotic inefficiency of the proposed estimators can be defined as $\mathbb{E}[C(\tau, m, N)] \times \mathbb{V}[H_{k:m}]$, which can be approximated with independent copies of $H_{k:m}$ and τ, obtained by running Algorithm 3.

State-of-the-art particle smoothers include fixed-lag approximations (Kitagawa and Sato, 2001; Cappé et al., 2005; Olsson et al., 2008), forward filtering backward smoothers (Godsill et al., 2004; Del Moral et al., 2010; Douc et al., 2011; Taghavi et al., 2013), and smoothers based on the two-filter formula (Briers et al., 2010; Kantas et al., 2015). These particle methods provide consistent approximations as $N \to \infty$, with associated mean squared error decreasing as $1/N$ (Section 4.4 of Kantas et al., 2015); except for fixed-lag approximations for which some bias remains. The cost is typically of order N with efficient implementations described in Fearnhead et al. (2010); Kantas et al. (2015); Olsson and Westerborn (2017), and is linear in T for fixed N. Parallelization over the N particles is mostly feasible, the main limitation coming from the resampling step (Murray et al., 2016a; Lee and Whiteley, 2015a; Whiteley et al., 2016; Paige et al., 2014; Murray et al., 2016b). The memory cost of particle filters is of order N, or $N \log N$ if trajectories are kept (Jacob et al., 2015), see also Koskela et al. (2018). Assessing the accuracy of particle approximations from a single run of these methods remains a major challenge; see Lee and Whiteley (2015b); Olsson and Douc (2017) for recent breakthroughs. Furthermore, we will see in Section 6.2 that the bias of particle smoothers cannot always be safely ignored. On the other hand, we will see in Section 6.3 that the variance of particle smoothers can be smaller than that of the proposed estimators, for a given computational cost. Thus, in terms of mean squared error per unit of computational cost, the proposed method is not expected to provide benefits.

The main advantage of the proposed method over particle smoothers lies in the construction of confidence intervals, and the possibility of parallelizing over independent runs as opposed to interacting particles. Additionally, a user of particle smoothers who would want more precise results would increase the number of particles N, if enough memory is available, discarding previous runs. On the other hand, the proposed estimator $\bar{H}^R_{k:m}$ can be refined to arbitrary precision by drawing more independent copies of $H_{k:m}$, for a constant memory requirement.

Other popular smoothers belong to the family of MCMC methods. Early examples include Gibbs samplers, updating components of the latent process conditionally on other components and on the observations (e.g. Carter and Kohn, 1994). The CPF kernel described in Section 1.2 can be used in the standard MCMC way, averaging over as many iterations as possible (Andrieu et al., 2010). The bias of MCMC estimators after a finite number of iterations is hard to assess, which makes the choice of burn-in period difficult. Asymptotically valid confidence intervals can be produced in various ways, for instance using the CODA package (Plummer et al., 2006); see also Vats et al. (2018). On the other hand, parallelization over the iterations is intrinsically challenging with MCMC methods (Rosenthal, 2000).

Therefore the proposed estimators have some advantages over existing methods, the main drawback being a potential increase in mean squared error for a given (serial) computational budget, as illustrated in the numerical experiments.

6 Numerical experiments

We illustrate the tuning of the proposed estimators, their advantages and their drawbacks through numerical experiments. All estimators of this section employ the Rao–Blackwellization technique described in Section 4, and multinomial resampling is used within all filters.

6.1 Hidden auto-regressive model

Our first example illustrates the proposed method, the impact of the number of particles N and that of the time horizon T, and the benefits of auxiliary particle filters. We consider a linear Gaussian model, with $x_0 \sim \mathcal{N}(0,1)$ and $x_t = \eta x_{t-1} + \mathcal{N}(0,1)$ for all $t \geq 1$, with $\eta = 0.9$. We assume that $y_t \sim \mathcal{N}(x_t, 1)$ for all $t \geq 1$.

We first generate $T = 100$ observations from the model, and consider the task of estimating all smoothing means, which corresponds to the test function $h : x_{0:T} \mapsto x_{0:T}$. With CPF kernels using bootstrap particle filters, with $N = 256$ particles and ancestor sampling (Lindsten et al., 2014), we draw meeting times τ independently, and represent a histogram of them in Figure 1(a). Based on these meeting times, we can choose k as a large quantile of the meeting times, for instance $k = 10$, and m as a multiple of k, for instance $m = 2k = 20$. For this choice, we find the average compute cost of each estimator to approximately equal that of a particle filter with 28 × 256 particles, with a memory usage equivalent to 2 × 256 particles. How many of these estimators can be produced in a given wall-clock time depends on available hardware. With $R = 100$ independent estimators, we obtain 95% confidence intervals indicated by black error bars in Figure 7. The true smoothing means, obtained by Kalman smoothing, are indicated by a line.

The method is valid for all N, which prompts the question of the optimal choice of N. Intuitively, larger values of N lead to smaller meeting times. However, the meeting time cannot be less than 2 by definition, which leads to a trade-off. We verify this intuition by numerical simulations with 1, 000 independent runs. For $N = 16$, $N = 128$, $N = 256$, $N = 512$ and $N = 1, 024$, we find average meeting times of 97, 15, 7, 4 and 3 respectively. After adjusting for the different numbers of particles, the expected cost of obtaining a meeting is approximately equivalent with $N = 16$ and $N = 512$, but more expensive for $N = 1, 024$. In practice, for specific integrals of interest, one can approximate the cost and the variance of the proposed estimators for various values of N, k and m using independent runs, and use the most favorable configuration in subsequent, larger experiments.

Next we investigate the effect of the time horizon T. We expect the performance of the CPF kernel to decay as T increases for a fixed N. We compensate by increasing N linearly with T. Table 1 reports the average meeting times obtained from $R = 500$ independent runs. We see that the average meeting times are approximately constant or slightly decreasing over T, implying that the linear scaling of N with T is appropriate or even conservative, in agreement with the literature (e.g. Huggins and Roy, 2018). The table contains the average meeting times obtained with and without ancestor sampling (Lindsten et al., 2014); we observe significant reductions of average meeting times with ancestor sampling, but it requires tractable transition densities. Finally, for the present model we can employ an auxiliary particle filter, in which particles are propagated conditionally on the next observation. Table 1 shows a significant reduction in expected meeting time. The combination of auxiliary particle filter and ancestor sampling naturally leads to the smallest expected meeting times.

6.2 A hidden auto-regressive model with an unlikely observation

We now illustrate the benefits of the proposed estimators in an example taken from Ruiz and Kappen (2017) where particle filters exhibit a significant bias. The latent process is defined as $x_0 \sim \mathcal{N}(0, 0.1^2)$ and $x_t = \eta x_{t-1} + \mathcal{N}(0, 0.1^2)$; we take $\eta = 0.9$ and consider $T = 10$ time steps. The process is observed only at time $T = 10$, where $y_T = 1$ and we assume $y_T \sim \mathcal{N}(x_T, 0.1^2)$. The observation y_T is unlikely under the model. Therefore the filtering distributions and the smoothing distributions have little overlap, particularly for times t close to T. This toy model is a stylized example of settings with highly-informative observations (Ruiz and Kappen, 2017; Del Moral and Murray, 2015).

We consider the task of estimating the smoothing mean $\mathbb{E}[x_9 \mid y_{10}]$. We run particle filters for different values of N, 10, 000 times independently, and plot kernel density estimators of the distributions of the estimators of $\mathbb{E}[x_9 \mid y_{10}]$ in Figure 2(a). The dashed vertical line represents the estimand $\mathbb{E}[x_9 \mid y_{10}]$, obtained analytically. We see that the bias diminishes when N increases, but

that it is still significant with $N = 16,384$ particles. For any fixed N, if we were to ignore the bias and produce confidence intervals using the central limit theorem based on independent particle filter estimators, the associated coverage would go to zero as the number of independent runs would increase.

In contrast, confidence intervals obtained with the proposed unbiased estimators are shown in Figure 7. For each value of N, the average meeting time was estimated from 100 independent runs (without ancestor sampling), and then k was set to that estimate, and m equal to k. Then, $R = 10,000$ independent estimators were produced, and confidence intervals were computed as described in Section 4. This leads to precise intervals for each choice of N. The average costs associated with $N = 128$, $N = 256$, $N = 512$ and $N = 1024$ were respectively matching the costs of particle filters with 3814, 4952, 9152 and 13,762 particles. To conclude, if we match computational costs and compare mean squared errors, the proposed method is not necessarily advantageous. However, if the interest lies in confidence intervals with adequate coverage, the proposed approach comes with guarantees thanks to the lack of bias and the central limit theorem for i.i.d. variables.

6.3 Prey-predator model

Our last example involves a model of plankton–zooplankton dynamics taken from Jones et al. (2010), in which the transition density is intractable (Bretó et al., 2009; Jacob, 2015). The bootstrap particle filter is still implementable, and one can either keep the entire trajectories of the particle filter, or perform fixed-lag approximations to perform smoothing. On the other hand, backward and ancestor sampling are not implementable.

The hidden state $x_t = (p_t, z_t)$ represents the population size of phytoplankton and zooplankton, and the transition from time t to $t + 1$ is given by a Lotka–Volterra equation,

$$\frac{dp_t}{dt} = \alpha p_t - c p_t z_t, \quad \text{and} \quad \frac{dz_t}{dt} = e c p_t z_t - m_l z_t - m_q z_t^2,$$

where the stochastic daily growth rate α is drawn from $\mathcal{N}(\mu_\alpha, \sigma_\alpha^2)$ at every integer time t. The propagation of each particle involves solving the above equation numerically using a Runge-Kutta method in the `odeint` library (Ahnert and Mulansky, 2011). The initial distribution is given by $\log p_0 \sim \mathcal{N}(\log 2, 1)$ and $\log z_0 \sim \mathcal{N}(\log 2, 1)$. The parameters c and e represent the clearance rate of the prey and the growth efficiency of the predator. Both m_l and m_q parameterize the mortality rate of the predator. The observations y_t are noisy measurements of the phytoplankton p_t, $\log y_t \sim \mathcal{N}(\log p_t, 0.2^2)$; z_t is not observed. We generate $T = 365$ observations using $\mu_\alpha = 0.7, \sigma_\alpha = 0.5$, $c = 0.25$, $e = 0.3$, $m_l = 0.1, m_q = 0.1$. We consider the problem of estimating the mean population of zooplankton at each time $t \in 0:T$, denoted by $\mathbb{E}[z_t \mid y_{1:T}]$, given the data-generating parameter.

The distribution of meeting times obtained with $N = 4,096$ particles over $R = 1,000$ experiments is shown in Figure 3(a). Based on this graph, we choose $k = 7$, $m = 2k = 14$, and produce $R = 1,000$ independent estimators of the smoothing means $\mathbb{E}[z_t \mid y_{1:T}]$. We compute the smoothing means with a long CPF chain, taken as ground truth. We then compute the relative variance of our estimators, defined as their variance divided by the square of the smoothing means. We find the average cost of the proposed estimator to be equivalent to that of a particle filter with 78, 377 particles. To approximately match the cost, we thus run particle filters with $2^{16} = 65,536$ particles, with and without fixed-lag smoothing with a lag of 10. The resulting relative variances are shown in Figure 3(b). We see that the proposed estimators yield a larger variance than particle filters, but that the difference is manageable. Fixed-lag smoothing provides significant variance reduction, particularly for earlier time indices. We can also verify that the bias of fixed-lag smoothing is negligible in the present example; this would however be hard to assess with fixed-lag smoothers alone.

7 Discussion

The performance of the proposed estimator is tied to the meeting time. As in Chopin and Singh (2015), the coupling inequality (Lindvall, 2002) can be used to relate the meeting time with the mixing of the underlying conditional particle filter kernel. The proposed approach can be seen as a framework to parallelize CPF chains and to obtain reliable confidence intervals over independent replicates. Any improvement in the CPF directly translates into more efficient Rhee–Glynn estimators, as we have illustrated in Section 6.1 with auxiliary particle filters and ancestor sampling. The methods proposed e.g. in Singh et al. (2017); Del Moral and Murray (2015); Guarniero et al. (2017); Gerber and Chopin (2015); Heng et al. (2017) could also be used in Rhee–Glynn estimators, with the hope of obtaining shorter meeting times and smaller variance.

We have considered the estimation of latent processes given known parameters. In the case of unknown parameters, joint inference of parameters and latent processes can be done with MCMC methods, and particle MCMC methods in particular (Andrieu et al., 2010). Couplings of generic particle MCMC methods could be achieved by combining couplings proposed in the present article with those described in Jacob et al. (2017) for Metropolis–Hastings chains. Furthermore, for fixed parameters, coupling the particle independent Metropolis–Hastings algorithm of Andrieu et al. (2010) would lead to unbiased estimators of smoothing expectations that would not require coupled resampling schemes (see Section 2.2).

The appeal of the proposed smoother, namely parallelization over independent replicates and confidence intervals, would be shared by perfect samplers. These algorithms aim at the more ambitious task of sampling exactly from the smoothing distribution (Lee et al., 2014). It remains unknown whether the proposed approach could play a role in the design of perfect samplers. We have established the validity of the Rhee–Glynn estimator under mild conditions, but its theoretical study as a function of the time horizon and the number of particles deserves further analysis (see Lee

et al., 2018, for a path forward). Finally, together with Fisher's identity (Douc et al., 2014), the proposed smoother provides unbiased estimators of the score for models where the transition density is tractable. This could help maximizing the likelihood via stochastic gradient ascent.

Acknowledgements. The authors thank Marco Cuturi, Mathieu Gerber, Jeremy Heng and Anthony Lee for helpful discussions. This work was initiated during the workshop on *Advanced Monte Carlo methods for complex inference problems* at the Isaac Newton Institute for Mathematical Sciences, Cambridge, UK held in April 2014. We would like to thank the organizers for a great event which led to this work.

References

Ahnert, K. and Mulansky, M. (2011). Odeint – solving ordinary differential equations in C++. *AIP Conference Proceedings*, 1389(1):1586–1589.

Andrieu, C., Doucet, A., and Holenstein, R. (2010). Particle Markov chain Monte Carlo (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):357–385.

Andrieu, C., Lee, A., Vihola, M., et al. (2018). Uniform ergodicity of the iterated conditional SMC and geometric ergodicity of particle Gibbs samplers. *Bernoulli*, 24(2):842–872.

Bretó, C., He, D., Ionides, E. L., and King, A. A. (2009). Time series analysis via mechanistic models. *The Annals of Applied Statistics*, 3(1):319–348.

Briers, M., Doucet, A., and Maskell, S. (2010). Smoothing algorithms for state-space models. *Annals of the Institute of Statistical Mathematics*, 62(1):61–89.

Cappé, O., Moulines, E., and Rydén, T. (2005). *Inference in Hidden Markov Models*. Springer-Verlag, New York.

Carter, C. K. and Kohn, R. (1994). On Gibbs sampling for state space models. *Biometrika*, 81(3):541–553.

Chopin, N. and Singh, S. S. (2015). On particle Gibbs sampling. *Bernoulli*, 21(3):1855–1883.

Del Moral, P., Doucet, A., and Singh, S. (2010). Forward smoothing using sequential Monte Carlo. *arXiv preprint arXiv:1012.5390*.

Del Moral, P. and Murray, L. M. (2015). Sequential Monte Carlo with highly informative observations. *SIAM/ASA Journal on Uncertainty Quantification*, 3(1):969–997.

Deligiannidis, G., Doucet, A., and Pitt, M. K. (2018). The correlated pseudo-marginal method. *Journal of the Royal Statistical Society: Series B (Statistical Methodology), to appear; arXiv preprint arXiv:1511.04992*.

Diaconis, P. and Freedman, D. (1999). Iterated random functions. *SIAM review*, 41(1):45–76.

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

Douc, R., Moulines, E., and Stoffer, D. (2014). *Nonlinear Time Series: Theory, Methods and Applications with R Examples*. Chapman and Hall, New York.

Doucet, A., de Freitas, N., and Gordon, N. (2001). *Sequential Monte Carlo methods in practice*. Springer-Verlag, New York.

Fearnhead, P., Wyncoll, D., and Tawn, J. (2010). A sequential smoothing algorithm with linear computational cost. *Biometrika*, 97(2):447–464.

Fulop, A. and Li, J. (2013). Efficient learning via simulation: A marginalized resample-move approach. *Journal of Econometrics*, 176(2):146–161.

Gelman, A., Brooks, S., Jones, G., and Meng, X. (2010). *Handbook of Markov chain Monte Carlo: methods and applications*. Chapman & Hall/CRC handbooks of modern statistical methods. CRC Press.

Gerber, M. and Chopin, N. (2015). Sequential quasi Monte Carlo. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 77(3):509–579.

Glynn, P. W. and Rhee, C.-H. (2014). Exact estimation for Markov chain equilibrium expectations. *Journal of Applied Probability*, 51A:377–389.

Glynn, P. W. and Whitt, W. (1992). The asymptotic efficiency of simulation estimators. *Operations Research*, 40(3):505–520.

Godsill, S. J., Doucet, A., and West, M. (2004). Monte Carlo smoothing for nonlinear time series. *Journal of the American Statistical Association*, 99(465):156–168.

Gordon, N., Salmond, J., and Smith, A. (1993). A novel approach to non-linear/non-Gaussian Bayesian state estimation. *IEE Proceedings on Radar and Signal Processing*, 140:107–113.

Guarniero, P., Johansen, A. M., and Lee, A. (2017). The iterated auxiliary particle filter. *Journal of the American Statistical Association*, 112(520):1636–1647.

Heng, J., Bishop, A. N., Deligiannidis, G., and Doucet, A. (2017). Controlled Sequential Monte Carlo. *arXiv preprint arXiv:1708.08396*.

Huggins, J. H. and Roy, D. M. (2018). Sequential Monte Carlo as approximate sampling: bounds, adaptive resampling via ∞-ESS, and an application to Particle Gibbs. *To appear in Bernoulli*.

Jacob, P. E. (2015). Sequential Bayesian inference for implicit hidden Markov models and current limitations. *ESAIM: Proceedings and Surveys*, 51:24–48.

Jacob, P. E., Lindsten, F., and Schön, T. B. (2016). Coupling of particle filters. *arXiv preprint arXiv:1606.01156*.

Jacob, P. E., Murray, L. M., and Rubenthaler, S. (2015). Path storage in the particle filter. *Statistics and Computing*, 25(2):487–496.

Jacob, P. E., O'Leary, J., and Atchadé, Y. F. (2017). Unbiased Markov chain Monte Carlo with couplings. *arXiv preprint arXiv:1708.03625*.

Jasra, A., Kamatani, K., Law, K. J. H., and Zhou, Y. (2017). Multilevel particle filters. *SIAM Journal on Numerical Analysis*, 55(6):3068–3096.

Johansen, A. M. and Doucet, A. (2008). A note on auxiliary particle filters. *Statistics & Probability Letters*, 78(12):1498–1504.

Jones, E., Parslow, J., and Murray, L. (2010). A Bayesian approach to state and parameter estimation in a phytoplankton-zooplankton model. *Australian Meteorological and Oceanographic Journal*, 59:7–16.

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

Kitagawa, G. and Sato, S. (2001). Monte Carlo smoothing and self-organising state-space model. In *Sequential Monte Carlo methods in practice*, pages 177–195. Springer.

Koskela, J., Jenkins, P. A., Johansen, A. M., and Spano, D. (2018). Asymptotic genealogies of interacting particle systems with an application to sequential Monte Carlo. *arXiv preprint arXiv:1804.01811*.

Kuhlenschmidt, B. and Singh, S. S. (2018). Stability of conditional sequential Monte Carlo. *arXiv preprint arXiv:1806.06520*.

Lee, A. (2008). Towards smooth particle filters for likelihood estimation with multivariate latent variables. Master's thesis, University of British Columbia.

Lee, A., Doucet, A., and Łatuszyński, K. (2014). Perfect simulation using atomic regeneration with application to sequential Monte Carlo. *arXiv preprints arXiv:1407.5770*.

Lee, A., Singh, S. S., and Vihola, M. (2018). Coupled conditional backward sampling particle filter. *arXiv preprint arXiv:1806.05852*.

Lee, A. and Whiteley, N. (2015a). Forest resampling for distributed sequential Monte Carlo. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 9(4):230–248.

Lee, A. and Whiteley, N. (2015b). Variance estimation and allocation in the particle filter. *arXiv preprint arXiv:1509.00394*.

Lindsten, F., Douc, R., and Moulines, E. (2015). Uniform ergodicity of the particle Gibbs sampler. *Scandinavian Journal of Statistics*, 42(3):775–797.

Lindsten, F., Jordan, M. I., and Schön, T. B. (2014). Particle Gibbs with ancestor sampling. *Journal of Machine Learning Research (JMLR)*, 15:2145–2184.

Lindsten, F. and Schön, T. B. (2013). Backward simulation methods for Monte Carlo statistical inference. *Foundations and Trends in Machine Learning*, 6(1):1–143.

Lindvall, T. (2002). *Lectures on the coupling method*. Courier Corporation.

McLeish, D. (2011). A general method for debiasing a Monte Carlo estimator. *Monte Carlo methods and applications*, 17(4):301–315.

Murray, L. M., Lee, A., and Jacob, P. E. (2016a). Parallel resampling in the particle filter. *Journal of Computational and Graphical Statistics*, 25(3):789–805.

Murray, L. M., Singh, S., Jacob, P. E., and Lee, A. (2016b). Anytime Monte Carlo. *arXiv preprint arXiv:1612.03319*.

Olsson, J., Cappé, O., Douc, R., and Moulines, E. (2008). Sequential Monte Carlo smoothing with application to parameter estimation in nonlinear state space models. *Bernoulli*, 14(1):155–179.

Olsson, J. and Douc, R. (2017). Numerically stable online estimation of variance in particle filters. *arXiv preprint arXiv:1701.01001*.

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

Paige, B., Wood, F., Doucet, A., and Teh, Y. W. (2014). Asynchronous anytime sequential Monte Carlo. In *Advances in Neural Information Processing Systems*, pages 3410–3418.

Pitt, M. K. (2002). Smooth particle filters for likelihood evaluation and maximisation. *Technical report, University of Warwick, Department of Economics*.

Pitt, M. K. and Shephard, N. (1999). Filtering via simulation: Auxiliary particle filters. *Journal of the American statistical association*, 94(446):590–599.

Plummer, M., Best, N., Cowles, K., and Vines, K. (2006). CODA: convergence diagnosis and output analysis for MCMC. *R news*, 6(1):7–11.

Rhee, C. (2013). *Unbiased Estimation with Biased Samplers*. PhD thesis, Stanford University.

Rhee, C. and Glynn, P. W. (2012). A new approach to unbiased estimation for SDE's. In *Proceedings of the Winter Simulation Conference*, pages 17:1–17:7.

Rosenthal, J. S. (2000). Parallel computing and Monte Carlo algorithms. *Far East Journal of Theoretical Statistics*, 4(2):207–236.

Ruiz, H.-C. and Kappen, H. J. (2017). Particle smoothing for hidden diffusion processes: Adaptive path integral smoother. *IEEE Transactions on Signal Processing*, 65(12):3191–3203.

Sen, D., Thiery, A. H., and Jasra, A. (2018). On coupling particle filter trajectories. *Statistics and Computing*, 28(2):461–475.

Singh, S. S., Lindsten, F., and Moulines, E. (2017). Blocking strategies and stability of particle Gibbs samplers. *Biometrika*, 104(4):953–969.

Taghavi, E., Lindsten, F., Svensson, L., and Schön, T. B. (2013). Adaptive stopping for fast particle smoothing. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6293–6297, Vancouver, Canada.

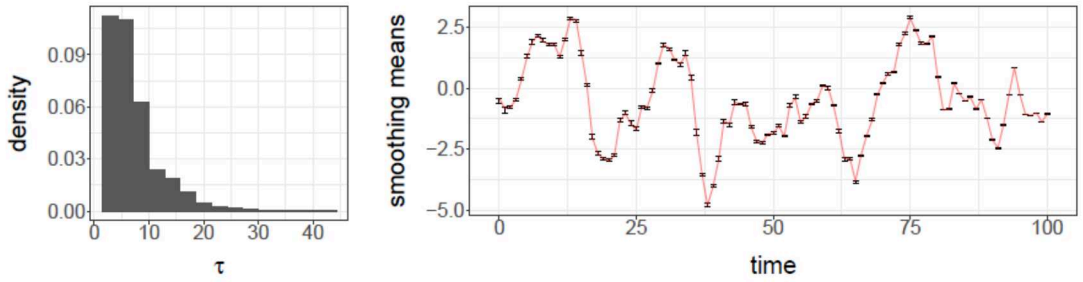
Vats, D., Flegal, J. M., Jones, G. L., et al. (2018). Strong consistency of multivariate spectral variance estimators in Markov chain Monte Carlo. *Bernoulli*, 24(3):1860–1909.

Vihola, M. (2017). Unbiased estimators and multilevel Monte Carlo. *Operations Research*.

Whiteley, N. (2010). Comment on Particle Markov chain Monte Carlo by Andrieu, Doucet and Holenstein. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):357–385.

Whiteley, N. (2013). Stability properties of some particle filters. *The Annals of Applied Probability*, 23(6):2500–2537.

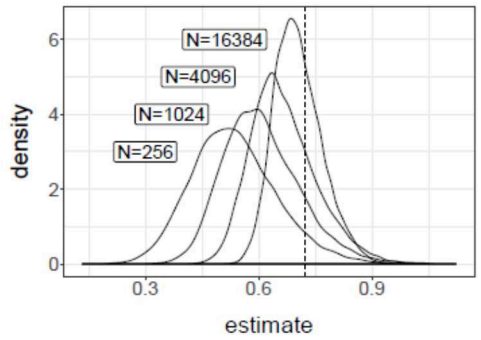
Whiteley, N., Lee, A., and Heine, K. (2016). On the role of interaction in sequential Monte Carlo algorithms. *Bernoulli*, 22(1):494–529.



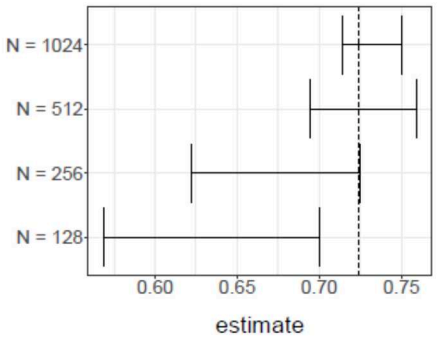
(a) Meeting times.

(b) Smoothing means: 95% confidence intervals as error bars in black, and exact values connected by a red line.

Fig. 1 Experiments with the auto-regressive model of Section 6.1, with $T = 100$ observations. Here the CPF kernel employs $N = 256$ particles and ancestor sampling. The distribution of the meeting times of coupled chains is shown on the left, and error bars for the estimation of smoothing means, obtained with $k = 10$ and $m = 20$ over $R = 100$ estimators, are shown on the right.

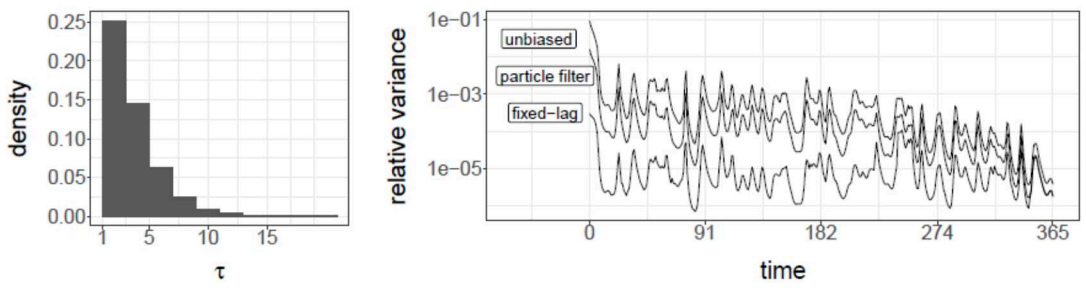


(a) Distributions of smoothing estimates produced by particle filters.



(b) 95% confidence intervals produced by the proposed method.

Fig. 2 Experiments with the model of Section 6.2. On the left, kernel density estimates based on 10, 000 runs show the distributions of estimates of $\mathbb{E}[x_9 \mid y_{10}]$, using particle filters with various N. The exact smoothing mean is represented by a vertical dashed line. On the right, 95% confidence intervals constructed from 10, 000 unbiased estimators are shown for various N.



(a) Meeting times.

(b) Relative variance of smoothing mean estimators, for the population of zooplankton at each time.

Fig. 3 Experiments with phytoplankton-zooplankton model of Section 6.3. On the left, histogram of 1, 000 independent meeting times, obtained with CCPF chains using $N = 4,096$ particles. On the right, relative variance of estimators of $\mathbb{E}[z_t \mid y_{1:T}]$ for all t. The proposed unbiased estimators ("unbiased") use $N = 4,096$, $k = 7$, $m = 14$; the particle filters ("particle filter") and fixed-lag smoothers ("fixed-lag") use $N = 65,536$, which makes all costs comparable in terms of numbers of particle propagations and weight evaluations.

Table 1 Average meeting time, as a function of the number of particles N and the time horizon T, with bootstrap particle filters and auxiliary particle filters, with and without ancestor sampling (AS), computed over $R = 500$ experiments. Standard deviations are between brackets. Results obtained in the hidden auto-regressive model of Section 6.1.

		Bootstrap PF		Auxiliary PF	
		without AS	with AS	without AS	with AS
N = 128	T = 50	17.84 (17.13)	7.73 (5.11)	3.96 (2.3)	3.37 (1.42)
N = 256	T = 100	13.16 (11.09)	7.59 (5.05)	3.78 (1.99)	3.16 (1.09)
N = 512	T = 200	12.52 (10.64)	6.77 (3.85)	3.52 (1.75)	2.97 (0.94)
N = 1024	T = 400	12.74 (10.96)	6.77 (3.47)	3.69 (1.94)	2.91 (0.87)
N = 2048	T = 800	13.58 (9.56)	6.34 (2.95)	3.54 (1.9)	2.95 (0.87)