JYX JYVÄSKYLÄN YLIOPISTO UNIVERSITY OF JYVÄSKYLÄ

This is a self-archived version of an original article. This version may differ from the original in pagination and typographic details.

Author(s): Karppinen, Santeri; Vihola, Matti

Title: Conditional particle filters with diffuse initial distributions

Year: 2021

Version: Published version

Copyright: © The Author(s) 2021

Rights: CC BY 4.0

Rights url: https://creativecommons.org/licenses/by/4.0/

Please cite the original version:

Karppinen, S., & Vihola, M. (2021). Conditional particle filters with diffuse initial distributions. Statistics and Computing, 31(3), Article 24. https://doi.org/10.1007/s11222-020-09975-1

Statistics and Computing (2021) 31:24
https://doi.org/10.1007/s11222-020-09975-1

Conditional particle filters with diffuse initial distributions

Santeri Karppinen[1] · **Matti Vihola**[1]

Received: 30 June 2020 / Accepted: 4 December 2020
© The Author(s) 2021

Abstract
Conditional particle filters (CPFs) are powerful smoothing algorithms for general nonlinear/non-Gaussian hidden Markov models. However, CPFs can be inefficient or difficult to apply with diffuse initial distributions, which are common in statistical applications. We propose a simple but generally applicable auxiliary variable method, which can be used together with the CPF in order to perform efficient inference with diffuse initial distributions. The method only requires simulatable Markov transitions that are reversible with respect to the initial distribution, which can be improper. We focus in particular on random walk type transitions which are reversible with respect to a uniform initial distribution (on some domain), and autoregressive kernels for Gaussian initial distributions. We propose to use online adaptations within the methods. In the case of random walk transition, our adaptations use the estimated covariance and acceptance rate adaptation, and we detail their theoretical validity. We tested our methods with a linear Gaussian random walk model, a stochastic volatility model, and a stochastic epidemic compartment model with time-varying transmission rate. The experimental findings demonstrate that our method works reliably with little user specification and can be substantially better mixing than a direct particle Gibbs algorithm that treats initial states as parameters.

Keywords Adaptive Markov chain Monte Carlo · Bayesian inference · Compartment model · Conditional particle filter · Diffuse initialisation · Hidden Markov model · Smoothing · State space model

1 Introduction

In statistical applications of general state space hidden Markov models (HMMs), commonly known also as state space models, it is often desirable to initialise the latent state of the model with a diffuse (uninformative) initial distribution (cf. Durbin and Koopman 2012). We mean by 'diffuse' the general scenario, where the first marginal of the smoothing distribution is highly concentrated relative to the prior of the latent Markov chain, which may also be improper.

The conditional particle filter (CPF) (Andrieu et al. 2010), and in particular its backward sampling variants (Whiteley 2010; Lindsten et al. 2014), has been found to provide efficient smoothing even with long data records, both empirically (e.g. Fearnhead and Künsch 2018) and theoretically (Lee et al. 2020). However, a direct application of the CPF to a model with a diffuse initial distribution will lead to poor performance, because most of the initial particles will ultimately be redundant, as they become drawn from highly unlikely regions of the state space.

There are a number of existing methods which can be used to mitigate this inefficiency. For simpler settings, it is often relatively straightforward to design proposal distributions that lead to an equivalent model, which no longer has a diffuse initial distribution. Indeed, if the first filtering distribution is already informative, its analytical approximation may be used directly as the first proposal distribution. The iteratively refined look-ahead approach suggested by Guarniero et al. (2017) extends to more complicated settings, but can require careful tuning for each class of problems.

We aim here for a general approach, which does not rely on any problem-specific constructions. Such a general approach which allows for diffuse initial conditions with particle Markov chain Monte Carlo (MCMC) is to include the initial latent state of the HMM as a 'parameter'. This was suggested by Murray et al. (2013) with the particle marginal Metropolis–Hastings (PMMH). The same

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s11222-020-09975-1.

✉ Santeri Karppinen
santeri.j.karppinen@jyu.fi

[1] Department of Mathematics and Statistics, University of Jyväskylä, 40014 Jyväskylä, Finland

Published online: 03 March 2021

approach is directly applicable also with the CPF (using particle Gibbs); see Fearnhead and Meligkotsidou (2016), who discuss general approaches based on augmentation schemes.

Our approach may be seen as an instance of the general 'pseudo-observation' framework of Fearnhead and Meligkotsidou (2016), but we are unaware of earlier works about the specific class of methods we focus on here. Indeed, instead of building the auxiliary variable from the conjugacy perspective as Fearnhead and Meligkotsidou (2016), our approach is based on Markov transitions that are reversible with respect to the initial measure of the HMM. This approach may be simpler to understand and implement in practice, and is very generally applicable. We focus here on two concrete cases: the 'diffuse Gaussian' case, where the initial distribution is Gaussian with a relatively uninformative covariance matrix, and the 'fully diffuse' case, where the initial distribution is uniform. We suggest online adaptation mechanisms for the parameters, which make the methods easy to apply in practice.

We start in Sect. 2 by describing the family of models we are concerned with, and the general auxiliary variable initialisation CPF that underlies all of our developments. We present the practical methods in Sect. 3. Section 4 reports experiments of the methods with three academic models and concludes with a realistic inference task related to modelling the COVID-19 epidemic in Finland. We conclude with a discussion in Sect. 5.

2 The model and auxiliary variables

Our main interest is with HMMs having a joint smoothing distribution π of the following form:

$$\pi(x_{1:T}) \propto p(x_1)p(y_1 \mid x_1)\prod_{k=2}^{T} p(x_k \mid x_{k-1})p(y_k \mid x_k), \quad (1)$$

where $\ell{:}u$ denotes the sequence of integers from ℓ to u (inclusive), $x_{1:T}$ denotes the latent state variables, and $y_{1:T}$ the observations. Additionally, π may depend on (hyper)parameters θ, the dependence on which we omit for now, but return to later, in Sect. 3.4.

For the convenience of notation, and to allow for some generalisations, we focus on the Feynman–Kac form of the HMM smoothing problem (cf. Del Moral 2004), where the distribution of interest π is represented in terms of a σ-finite measure $M_1(\mathrm{d}x_1)$ on the state space X, Markov transitions $M_2, \ldots, M_T$ on X and potential functions $G_k : \mathsf{X}^k \to [0, \infty)$ so that

$$\pi(\mathrm{d}x_{1:T}) \propto M_1(\mathrm{d}x_1)G_1(x_1)\prod_{k=2}^{T} M_k(x_{k-1}, \mathrm{d}x_k)G_k(x_{1:k}). \quad (2)$$

The classical choice, the so-called 'bootstrap filter' (Gordon et al. 1993), corresponds to $M_1(\mathrm{d}x_1) = p(x_1)\mathrm{d}x_1$ and $M_k(x_{k-1}, \mathrm{d}x_k) = p(x_k \mid x_{k-1})\mathrm{d}x_k$, where '$\mathrm{d}x$' stands for the Lebesgue measure on $\mathsf{X} = \mathbb{R}^d$, and $G_k(x_{1:k}) = p(y_k \mid x_k)$, but other choices with other 'proposal distributions' M_k are also possible. Our main focus is when M_1 is diffuse with respect to the first marginal of π. We stress that our method accomodates also improper M_1, such as the uniform distribution on $\mathbb{R}^d$, as long as (2) defines a probability.

The key ingredient of our method is an auxiliary Markov transition, Q, which we can simulate from, and which satisfies the following:

Assumption 1 (M_1-reversibility) The Markov transition probability Q is reversible with respect to the σ-finite measure M_1, or M_1-reversible, if

$$\int M_1(\mathrm{d}x_0)Q(x_0, \mathrm{d}x_1)\mathbf{1}(x_0 \in A, x_1 \in B) = \int M_1(\mathrm{d}x_1)Q(x_1, \mathrm{d}x_0)\mathbf{1}(x_0 \in A, x_1 \in B), \quad (3)$$

for all measurable $A, B \subset \mathsf{X}$.

We discuss practical ways to choose Q in Sect. 3. Assuming an M_1-reversible Q, we define an augmented target distribution, involving a new 'pseudo-state' x_0 which is connected to x_1 by Q:

$$\tilde{\pi}(\mathrm{d}x_{0:T}) = \pi(\mathrm{d}x_{1:T})Q(x_1, \mathrm{d}x_0) \propto M_1(\mathrm{d}x_0)Q(x_0, \mathrm{d}x_1)G_1(x_1)\prod_{k=2}^{T} M_k(x_{k-1}, \mathrm{d}x_k)G_k(x_{1:k}).$$

It is clear by construction that $\tilde{\pi}$ admits π as its marginal, and therefore, if we can sample $x_{0:T}$ from $\tilde{\pi}$, then $x_{1:T} \sim \pi$.

Our method may be viewed as a particle Gibbs (Andrieu et al. 2010) which targets $\tilde{\pi}$, regarding x_0 as the 'parameter', and $x_{1:T}$ the 'latent state', which are updated using the CPF. Algorithm 1 summarises the method, which we call the 'auxiliary initialisation' CPF (AI-CPF). Algorithm 1 determines a π-invariant Markov transition $\dot{x}_{1:T} \to \tilde{X}_{1:T}^{(B_{1:T})}$; the latter output of the algorithm will be relevant later, when we discuss adaptation.

Algorithm 1 AI-CPF($\dot{x}_{1:T}$; Q, $M_{2:T}$, $G_{1:T}$, N)

1: Simulate $X_0 \sim Q(\dot{x}_1, \cdot)$.
2: Simulate $\tilde{X}_1^{(2:N)} \sim Q(X_0, \cdot)$ and set $\tilde{X}_1^{(1)} = \dot{x}_1$.
3: $(\tilde{X}_{1:T}^{(1:N)}, W_{1:T}^{(1:N)}, A_{1:T-1}^{(1:N)}) \leftarrow$ F-CPF($\dot{x}_{2:T}, \tilde{X}_1^{(1:N)}$; $M_{2:T}, G_{1:T}, N$).
4: $(B_{1:T}, V^{(1:N)}) \leftarrow$ PickPath-x($\tilde{X}_{1:T}^{(1:N)}, W_{1:T}^{(1:N)}, A_{1:T-1}^{(1:N)}, M_{2:T}, G_{2:T}$).
5: Set $\tilde{x}_{1:T} = \big(\tilde{X}_1^{(B_1)}, \tilde{X}_2^{(B_2)}, \ldots, \tilde{X}_T^{(B_T)}\big)$.
6: **output** $\big(\tilde{x}_{1:T}, (B_1, V^{(1:N)}, \tilde{X}_1^{(1:N)})\big)$.

Line 1 of Algorithm 1 implements a Gibbs step sampling X_0 conditional on $X_{1:T} = \dot{x}_{1:T}$, and lines 2–4 implement together a CPF targeting the conditional of $X_{1:T}$ given X_0. Line 3 runs what we call a 'forward' CPF, which is just a standard CPF conditional on the first state particles $X_1^{(1:N)}$, detailed in Algorithm 2. Line 4 refers to a call of PICKPATH- AT (Algorithm 3) for ancestor tracing as in the original work of Andrieu et al. (2010), or PICKPATH- BS (Algorithm 4) for backward sampling (Whiteley 2010). $\text{Categ}(w^{(1:N)})$ stands for the categorical distribution, that is, $A \sim \text{Categ}(w^{(1:N)})$ if $\Pr(A = i) = w^{(i)}$.

Algorithm 2 F-CPF($\dot{x}_{2:T}, X_1^{(1:N)}; M_{2:T}, G_{1:T}, N$)

1: Set $\mathbf{X}_1^{(1:N)} \leftarrow X_1^{(1:N)}$.
2: **for** $k = 1, \ldots, T-1$ **do**
3: $\tilde{W}_k^{(i)} \leftarrow G_k(\mathbf{X}_k^{(i)})$ and $W_k^{(i)} \leftarrow \tilde{W}_k^{(i)} / \sum_{j=1}^N \tilde{W}_k^{(j)}$ for $i \in \{1{:}N\}$.
4: $A_k^{(2:N)} \sim \text{Categ}\big(W_k^{(1:N)}\big)$ and set $A_k^{(1)} \leftarrow 1$.
5: Draw $X_{k+1}^{(i)} \sim M_{k+1}(\cdot \mid X_k^{(A_k^{(i)})})$ for $i \in \{2{:}N\}$.
6: Set $X_{k+1}^{(1)} = \dot{x}_{k+1}$.
7: Set $\mathbf{X}_{k+1}^{(i)} = (\mathbf{X}_k^{(A_k^{(i)})}, X_{k+1}^{(i)})$ for $i \in \{1{:}N\}$.
8: **end for**
9: $\tilde{W}_T^{(1:N)} \leftarrow G_T(\mathbf{X}_T^{(1:N)})$ and $W_T^{(i)} \leftarrow \tilde{W}_T^{(i)} / \sum_{j=1}^N \tilde{W}_T^{(j)}$ for $i = \{1{:}N\}$.
10: **output** $(X_{1:T}^{(1:N)}, W_{1:T}^{(1:N)}, A_{1:T-1}^{(1:N)})$.

The ancestor tracing variant can be used when the transition densities are unavailable. However, our main interest here is with backward sampling, summarised in Algorithm 4 in the common case where the potentials only depend on two consecutive states, that is, $G_k(x_{1:k}) = G_k(x_{k-1:k})$, and the transitions admit densities $M_k(x_{k-1}, \mathrm{d}x_k) = M_k(x_{k-1}, x_k)\,\mathrm{d}x_k$ with respect to some dominating σ-finite measure '$\mathrm{d}x_k$'.

Algorithm 3 PICKPATH- AT($\tilde{X}_{1:T}^{(1:N)}, W_{1:T}^{(1:N)}, A_{1:T-1}^{(1:N)}, M_{2:T}, G_{2:T}$)

1: Draw $B_K \sim \text{Categ}\big(W_T^{(1:N)}\big)$.
2: **output** $(B_{1:T}, W_1^{(1:N)})$ where $B_k = A_k^{(B_{k+1})}$ for $k = T-1, \ldots, 1$.

Algorithm 4 PICKPATH- BS($\tilde{X}_{1:T}^{(1:N)}, W_{1:T}^{(1:N)}, A_{1:T-1}^{(1:N)}, M_{2:T}, G_{2:T}$)

1: Draw $B_K \sim \text{Categ}\big(W_T^{(1:N)}\big)$.
2: **for** $k = T-1, \ldots, 1$ **do**
3: $\tilde{V}_k^{(i)} \leftarrow W_k^{(i)} M_{k+1}(\tilde{X}_k^{(i)}, \tilde{X}_{k+1}^{(B_{k+1})}) G_{k+1}(\tilde{X}_k^{(i)}, \tilde{X}_{k+1}^{(B_{k+1})})$ for $i \in \{1{:}N\}$.
4: Simulate $B_k \sim \text{Categ}(V_k^{(1:N)})$, where $V_k^{(i)} = \tilde{V}_k^{(i)} / \sum_{j=1}^N \tilde{V}_k^{(j)}$.
5: **end for**
6: **output** $(B_{1:T}, V_1^{(1:N)})$.

We conclude with a brief discussion on the general method of Algorithm 1.

(i) We recognise that Algorithm 1 is not new per se, in that it may be viewed just as a particle Gibbs applied for a specific auxiliary variable model. However, we are unaware of Algorithm 1 being presented with the present focus: with an M_1-reversible Q, and allowing for an improper M_1.
(ii) Algorithm 1 may be viewed as a generalisation of the standard CPF. Indeed, taking $Q(x_0, \mathrm{d}x_1) = M_1(\mathrm{d}x_1)$ in Algorithm 1 leads to the standard CPF. Note that Line 1 is redundant in this case, but is necessary in the general case.
(iii) In the case $T = 1$, Line 3 of Algorithm 1 is redundant, and the algorithm resembles certain multiple-try Metropolis methods (cf. Martino 2018) and has been suggested earlier by Mendes et al. (2015).
(iv) Algorithm 2 is formulated using multinomial resampling, for simplicity. We note that any other unbiased resampling may be used, as long as the conditional resampling is designed appropriately; see Chopin and Singh (2015).

The 'CPF generalisation' perspective of Algorithm 1 may lead to other useful developments; for instance, one could imagine the approach to be useful with the CPF applied for static (non-HMM) targets, as in sequential Monte Carlo samplers (Del Moral et al. 2006). The aim of the present paper is, however, to use Algorithm 1 with diffuse initial distributions.

3 Methods for diffuse initialisation of conditional particle filters

To illustrate the typical problem that arises with a diffuse initial distribution M_1, we examine a simple noisy AR(1) model:

$$\begin{aligned} x_{k+1} &= \rho x_k + \eta_k,\ \eta_k \sim N(0, \sigma_x^2) \\ y_k &= x_k + \epsilon_k,\ \epsilon_k \sim N(0, \sigma_y^2), \end{aligned} \tag{4}$$

for $k \geq 1$, $x_1 \sim N(0, \sigma_1^2)$, $M_1(\mathrm{d}x_1) = p(x_1)\mathrm{d}x_1$, $M_k(x_{k-1}, \mathrm{d}x_k) = p(x_k \mid x_{k-1})\mathrm{d}x_k$ and $G_k(x_{1:k}) = p(y_k \mid x_k)$.

We simulated a dataset of length $T = 50$ from this model with $x_1 = 0$, $\rho = 0.8$ and $\sigma_x = \sigma_y = 0.5$. We then ran 6000 iterations of the CPF with backward sampling (CPF-BS) with $\sigma_1 \in \{10, 100, 1000\}$; that is, Algorithm 1 with $Q(x_0, \cdot) = M_1(\cdot)$ together with Algorithm 4, and discarded the first 1000 iterations as burn-in. For each value of σ_1, we monitored the efficiency of sampling x_1. Figure 1 displays the resulting traceplots. The estimated integrated autocorrelation times (IACT) were approximately 3.75, 28.92 and 136.64, leading to effective sample sizes ($\mathrm{n_{eff}}$) of 1600, 207 and 44,

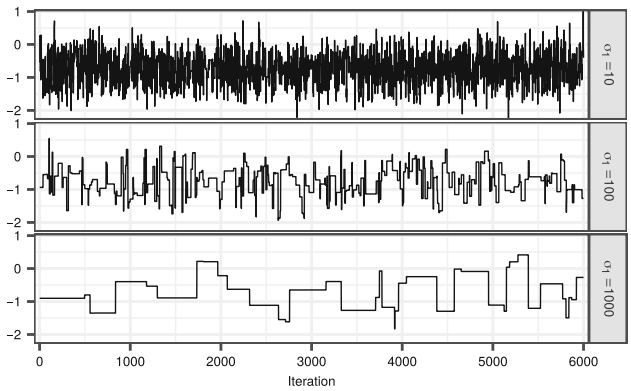

Fig. 1 Traceplot of the initial state of the noisy AR(1) model, using the CPF with 16 particles and backward sampling with $\sigma_1 = 10$ (top), 100 (middle) and 1000 (bottom)

respectively. This demonstrates how the performance of the CPF-BS deteriorates as the initial distribution of the latent state becomes more diffuse.

3.1 Diffuse Gaussian initialisation

In the case that M_1 in (2) is Gaussian with mean μ and covariance Σ, we can construct a Markov transition function that satisfies (3) using an autoregressive proposal similar to 'preconditioning' in the Crank-Nicolson algorithm (cf. Cotter et al. 2013). This proposal comes with a parameter $\beta \in (0, 1]$, so we denote this kernel by Q_β^{AR}. A variate $Z \sim Q_\beta^{\mathrm{AR}}(x, \cdot)$ can be drawn simply by setting

$$Z = \sqrt{1-\beta^2}(x-\mu) + \beta W + \mu, \tag{5}$$

where $W \sim N(0, \Sigma)$. We refer to Algorithm 1 with $Q = Q_\beta^{\mathrm{AR}}$ as the diffuse Gaussian initialisation CPF (DGI-CPF). In the special case $\beta = 1$, we have $Q_1^{\mathrm{AR}} = M_1$, and so the DGI-CPF is equivalent with the standard CPF.

3.2 Fully diffuse initialisation

Suppose that $M_1(\mathrm{d}x) = M_1(x)\mathrm{d}x$ where $M_1(x) \equiv 1$ is a uniform density on $\mathsf{X} = \mathbb{R}^d$. Then, any symmetric transition Q satisfies M_1-reversibility. In this case, we suggest to use $Q_C^{\mathrm{RW}}(x, \mathrm{d}y) = q_C^{\mathrm{RW}}(x, y)\mathrm{d}y$ with a multivariate normal density $q_C^{\mathrm{RW}}(x, y) = N(y; x, C)$, with covariance $C \in \mathbb{R}^{d\times d}$. In case of constraints, that is, a non-trivial domain $D \subset \mathbb{R}^d$, we have $M_1 = 1(x \in D)$. Then, we suggest to use a Metropolis–Hastings type transition probability:

$$Q_C^{\mathrm{RW}}(x, \mathrm{d}y) = q_C^{\mathrm{RW}}(x, y) \min\left\{1, \frac{M_1(y)}{M_1(x)}\right\}\mathrm{d}y + \delta_x(\mathrm{d}y) r(x),$$

where $r(x) \in [0, 1]$ is the rejection probability. This method works, of course, with arbitrary M_1, but our focus is with a diffuse case, where the domain D is regular and large enough, so that rejections are rare. We stress that also in this case, $M_1(x) = 1(x \in D)$ may be improper. We refer to Algorithm 1 with Q_C^{RW} as the 'fully diffuse initialisation' CPF (FDI-CPF).

We note that whenever M_1 can be evaluated pointwise, the FDI-CPF can always be applied, by considering the modified Feynman–Kac model $\tilde{M}_1 \equiv 1$ and $\tilde{G}_1(x) = M_1(x)G_1(x)$. However, when M_1 is Gaussian, the DGI-CPF can often lead to a more efficient method. As with standard random walk Metropolis algorithms, choosing the covariance $C \in \mathbb{R}^{d\times d}$ is important for the efficiency of the FDI-CPF.

3.3 Adaptive proposals

Finding a good autoregressive parameter of Q_β^{AR} or the covariance parameter of Q_C^{RW} may be time-consuming in practice. Inspired by the recent advances in adaptive MCMC (cf. Andrieu and Thoms 2008; Vihola 2020), it is natural to apply adaptation also with the (iterated) AI-CPF. Algorithm 5 summarises a generic adaptive AI-CPF (AAI-CPF) using a parameterised family $\{Q_\zeta\}_{\zeta\in\mathsf{Z}}$ of M_1-reversible proposals, with parameter ζ.

Algorithm 5 AAI-CPF($\dot{x}_{1:T}^{(0)}$; $Q_{\zeta^{(0)}}, M_{2:T}, G_{1:T}, N$)

1: **for** $j = 1, \ldots, n$ **do**
2: $\quad(\dot{x}_{1:T}^{(j)}, \xi^{(j)}) \leftarrow$ AI-CPF($\dot{x}_{1:T}^{(j-1)}$; $Q_{\zeta^{(j-1)}}, M_{2:T}, G_{1:T}, N$).
3: $\quad\zeta^{(j)} \leftarrow$ ADAPT($\zeta^{(j-1)}, \xi^{(j)}, j$).
4: **end for**
5: **output** $(\dot{x}_{1:T}^{(1)}, \ldots, \dot{x}_{1:T}^{(n)})$.

The function ADAPT implements the adaptation, which typically leads to $\zeta^{(j)} \to \zeta^*$, corresponding to a well-mixing configuration. We refer to the instances of the AAI-CPF with the AI-CPF step corresponding to the DGI-CPF and the FDI-CPF as the adaptive DGI-CPF and FDI-CPF, respectively.

We next focus on concrete adaptations which may be used within our framework. In the case of the FDI-CPF, Algorithm 6 implements a stochastic approximation variant (Andrieu and Moulines 2006) of the adaptive Metropolis covariance adaptation of Haario et al. (2001).

Here, η_j are step sizes that decay to zero, $\zeta_j = (\mu_j, \Sigma_j)$ the estimated mean and covariance of the smoothing distribution, respectively, and $Q_\zeta = Q_{c\Sigma}^{\mathrm{RW}}$ where $c > 0$ is a scaling factor of the covariance Σ. In the case of random walk

Algorithm 6 $\textsc{Adapt}_{FDI,AM}\big((\mu, \Sigma), (B_1, W_1^{(1:N)}, X_1^{(1:N)}), j\big)$

1: $\mu_* \leftarrow (1-\eta_j)\mu + \eta_j X_1^{(B_1)}$.
2: $\Sigma_* \leftarrow (1-\eta_j)\Sigma + \eta_j (X_1^{(B_1)} - \mu)(X_1^{(B_1)} - \mu)^{\mathrm{T}}$.
3: **output** (μ_*, Σ_*).

Metropolis, this scaling factor is usually taken as $2.38^2/d$ (Gelman et al. 1996), where d is the state dimension of the model. In the present context, however, the optimal value of $c > 0$ appears to depend on the model and on the number of particles N. This adaptation mechanism can be used both with PickPath- AT and with PickPath- BS, but may require some manual tuning to find a suitable $c > 0$.

Algorithm 7 details another adaptation for the FDI-CPF, which is intended to be used together with PickPath- BS only. Here, $\zeta_j = (\mu_j, \Sigma_j, \delta_j)$ contains the estimated mean, covariance and the scaling factor, and $Q_\zeta = Q^{\mathrm{RW}}_{C(\zeta)}$, where $C(\zeta) = e^\delta \Sigma$.

Algorithm 7 $\textsc{Adapt}_{FDI,ASWAM}\big((\mu, \Sigma, \delta), (B_1, W_1^{(1:N)}, X_1^{(1:N)}), j\big)$

1: $\mu_* \leftarrow (1-\eta_j)\mu + \eta_j \sum_{i=1}^N W_1^{(i)} X_1^{(i)}$.
2: $\Sigma_* \leftarrow (1-\eta_j)\Sigma + \eta_j \sum_{i=1}^N W_1^{(i)} (X_1^{(i)} - \mu)(X_1^{(i)} - \mu)^{\mathrm{T}}$.
3: $\delta_* \leftarrow \delta + \eta_j(\alpha - \alpha_*)$ where $\alpha = 1 - W_1^{(1)}$.
4: **output** $(\mu_*, \Sigma_*, \delta_*)$.

Algorithm 8 $\textsc{Adapt}_{DGI,AS}\big(\zeta, (B_1, W_1^{(1:N)}, X_1^{(1:N)}), j\big)$

1: $\zeta_* \leftarrow \zeta + \eta_j(\alpha - \alpha_*)$ where $\alpha = 1 - W_1^{(1)}$.
2: **output** ζ_*.

This algorithm is inspired by a Rao–Blackwellised variant of the adaptive Metropolis within adaptive scaling method (cf. Andrieu and Thoms 2008), which is applied with standard random walk Metropolis. We use all particles with their backward sampling weights to update the mean μ and covariance Σ, and an 'acceptance rate' α, that is, the probability that the first coordinate of the reference trajectory is not chosen. Recall that after the AI–CPF in Algorithm 5 has been run, the first coordinate of the reference trajectory and its associated weight reside in the first index of the particle and weight vectors contained in $\xi^{(j)}$.

The optimal value of the acceptance rate parameter α_* is typically close to one, in contrast with random walk Metropolis, where $\alpha_* \in [0.234, 0.44]$ are common (Gelman et al. 1996). Even though the optimal value appears to be problem-dependent, we have found empirically that $0.7 \le \alpha_* \le 0.9$ often leads to reasonable mixing. We will show empirical evidence for this finding in Sect. 4.

Algorithm 8 describes a similar adaptive scaling type mechanism for tuning $\beta = \mathrm{logit}^{-1}(\zeta)$ in the DGI-CPF, with $Q_\zeta = Q^{\mathrm{AR}}_\beta$. The algorithm is most practical with PickPath-BS.

We conclude this section with a consistency result for Algorithm 5, using the adaptation mechanisms in Algorithms 6 and 7. In Theorem 1, we denote $(\mu_j, \Sigma_j) = \zeta_j$ in the case of Algorithm 6, and $(\mu_j, \Sigma_j, \delta_j) = \zeta_j$ with Algorithm 7.

Theorem 1 *Suppose D is a compact set, a uniform mixing condition (Assumption 2 in Appendix A) holds, and there exists an $\epsilon > 0$ such that for all $j \ge 1$, the smallest eigenvalue $\lambda_{\min}(\Sigma_j) \ge \epsilon$, and with Algorithm 7 also $\delta_j \in [\epsilon, \epsilon^{-1}]$. Then, for any bounded function $f : \mathsf{X} \to \infty$,*

$$\frac{1}{n}\sum_{k=1}^{n} f(\dot{x}_{1:T}^{(k)}) \xrightarrow{n\to\infty} \pi(f). \quad \textit{almost surely.}$$

The proof of Theorem 1 is given in Appendix A. The proof is slightly more general, and accomodates for instance t-distributed instead of Gaussian proposals for the FDI-CPF. We note that the latter stability condition, that is, existence of the constant $\epsilon > 0$, may be enforced by introducing a 'rejection' mechanism in the adaptation; see the end of Appendix A. However, we have found empirically that the adaptation is stable also without such a stabilisation mechanism.

3.4 Use within particle Gibbs

Typical application of HMMs in statistics involves not only smoothing, but also inference of a number of 'hyperparameters' θ, with prior density $\mathrm{pr}(\theta)$, and with

$$\begin{aligned}\gamma_\theta(x_{1:T}) &= p(y_{1:T}, x_{1:T} \mid \theta) \qquad (6)\\ &= M_1(x_1)G_1^{(\theta)}(x_1)\prod_{k=2}^{T} M_k^{(\theta)}(x_{k-1}, x_k)G_k^{(\theta)}(x_{k-1}, x_k).\end{aligned}$$

The full posterior, $\check{\pi}(\theta, x_{1:T}) \propto \mathrm{pr}(\theta)\gamma_\theta(x_{1:T})$ may be inferred with the particle Gibbs (PG) algorithm of Andrieu et al. (2010). (We assume here that M_1 is diffuse, and thereby independent of θ.)

The PG alternates between (Metropolis-within-)Gibbs updates for θ conditional on $x_{1:T}$, and CPF updates for $x_{1:T}$ conditional on θ. The (A)AI-CPF applied with $M_{2:T}^{(\theta)}$ and $G_{1:T}^{(\theta)}$ may be used as a replacement of the CPF steps in a PG. Another adaptation, independent of the AAI-CPF, may be used for the hyperparameter updates (cf. Vihola 2020).

Algorithm 9 summarises a generic adaptive PG with the AAI-CPF. Line 2 involves an update of $\theta^{(j-1)}$ to $\theta^{(j)}$ using transition probabilities $K_{\zeta_\theta}(\cdot, \cdot \mid x_{1:T})$ which leave

$\check{\pi}(\theta \mid x_{1:T})$ invariant, and Line 3 is (optional) adaptation. This could, for instance, correspond to the robust adaptive Metropolis algorithm (RAM) (Vihola 2012). Lines 4 and 5 implement the AAI-CPF. Note that without Lines 3 and 5, Algorithm 9 determines a $\check{\pi}$-invariant transition rule.

Algorithm 9 AAI-PG$(\theta^{(0)}, \dot{x}_{1:T}^{(0)}; Q_{C(\zeta^{(0)})}, M_{2:T}, G_{1:T}, N)$

1: **for** $j = 1, \ldots, n$ **do**
2: $\quad (\theta^{(j)}, \xi_\theta^{(j)}) \sim K_{\zeta_\theta^{(j-1)}}(\theta^{(j-1)}, \cdot \mid \dot{x}_{1:T}^{(j-1)})$.
3: $\quad \zeta_\theta^{(j)} \leftarrow \textsc{Adapt}_\theta(\zeta_\theta^{(j-1)}, \theta^{(j)}, \xi_\theta^{(j)})$.
4: $\quad (\dot{x}_{1:T}^{(j)}, \xi^{(j)}) \leftarrow$ AI-CPF$(\dot{x}_{1:T}^{(j-1)}; Q_{\zeta^{(j-1)}}, M_{2:T}^{(\theta^{(j)})}, G_{1:T}^{(\theta^{(j)})}, N)$.
5: $\quad \zeta^{(j)} \leftarrow \textsc{Adapt}(\zeta^{(j-1)}, \xi^{(j)}, j)$.
6: **end for**
7: **output** $\big((\theta^{(1)}, \dot{x}_{1:T}^{(1)}), \ldots, (\theta^{(n)}, \dot{x}_{1:T}^{(n)})\big)$.

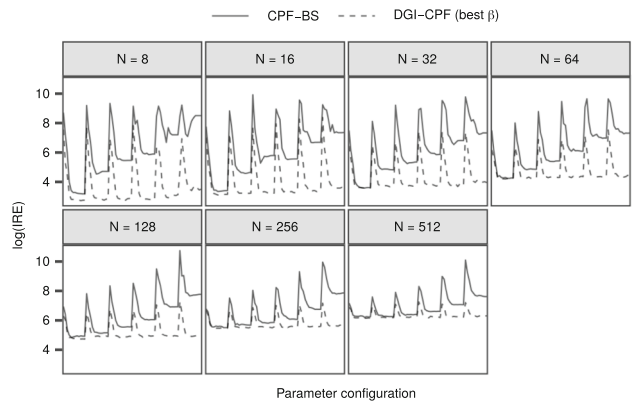

Fig. 2 The log (IRE) resulting from the application of the CPF-BS and the best case DGI-CPF to the RW model. The horizontal axis depicts different configurations of σ_1 and σ_x, and in each panel N varies

4 Experiments

In this section, we study the application of the methods presented in Sect. 3 in practice. Our focus will be on the case of the bootstrap filter, that is, $M_1(\mathrm{d}x_1) = p(x_1)\mathrm{d}x_1$, $M_k(x_{k-1}, \mathrm{d}x_k) = p(x_k \mid x_{k-1})\mathrm{d}x_k$ and $G_k(x_{1:k}) = p(y_k \mid x_k)$.

We start by investigating two simple HMMs: the noisy random walk model (RW), that is, (4) with $\rho = 1$, and the following stochastic volatility (SV) model:

$$\begin{aligned} x_{k+1} &= x_k + \eta_k, \\ y_k &= e^{x_k}\epsilon_k, \end{aligned} \tag{7}$$

with $x_1 \sim N(0, \sigma_1^2)$, $\eta_k \sim N(0, \sigma_x^2)$ and $\epsilon_k \sim N(0, \sigma_y^2)$. In Sect. 4.3, we study the dependence of the method with varying dimension, with a static multivariate normal model. We conclude in Sect. 4.4 by applying our methods in a realistic inference problem related to modelling the COVID-19 epidemic in Finland.

4.1 Comparing DGI-CPF and CPF-BS

We first studied how the DGI-CPF performs in comparison to the CPF-BS when the initial distributions of the RW and SV model are diffuse. Since the efficiency of sampling is affected by both the values of the model parameters (cf. Fig. 1) and the number of particles N, we experimented with a range of values $N \in \{8, 16, 32, 64, 128, 256, 512\}$ for which we applied both methods with $n = 10000$ iterations plus 500 burn-in. We simulated data from both the RW and SV models with $T = 50$, $x_1 = 0$, $\sigma_y = 1$ and varying $\sigma_x \in \{0.01, 0.05, 0.1, 0.5, 1, 2, 5, 10, 20, 50, 100, 200\}$. We then applied both methods for each dataset with the corresponding σ_x, but with varying $\sigma_1 \in \{10, 50, 100, 200, 500, 1000\}$, to study the sampling efficiency under different parameter configurations (σ_x and σ_1). For the DGI-CPF, we varied the parameter $\beta \in \{0.01, 0.02, \ldots, 0.99\}$. We computed the estimated integrated autocorrelation time (IACT) of the simulated values of x_1 and scaled this by the number of particles N. The resulting quantity, the inverse relative efficiency (IRE), measures the asymptotic efficiencies of estimators with varying computational costs (Glynn and Whitt 1992).

Figure 2 shows the comparison of the CPF-BS with the best DGI-CPF, that is, the DGI-CPF with the β that resulted in the lowest IACT for each parameter configuration and N.

The results indicate that with N fixed, a successful tuning of β can result in greatly improved mixing in comparison with the CPF-BS. While the performance of the CPF-BS approaches that of the best DGI-CPF with increasing N, the difference in performance remains substantial with parameter configurations that are challenging for the CPF-BS.

The optimal N which minimizes the IRE depends on the parameter configuration. For 'easy' configurations (where IRE is small), even $N = 8$ can be enough, but more 'difficult' configurations (where IRE is large), higher values of N can be optimal. Similar results for the SV model are shown in Online Resource 1 (Fig. 1), and lead to similar conclusions.

The varying 'difficulty' of the parameter configurations is further illustrated in Fig. 3, which shows the log (IACT) for the SV model with $N = 256$ particles. The CPF-BS performed the worst when the initial distribution was very diffuse with respect to the state noise σ_x, as expected. In contrast, the well-tuned DGI-CPF appears rather robust with respect to changing parameter configuration. The observations were similar with other N, and for the RW model; see Online Resource 1 (Fig. 2).

The results in Figs. 2 and 3 illustrate the potential of the DGI-CPF, but are overly optimistic because in practice, the

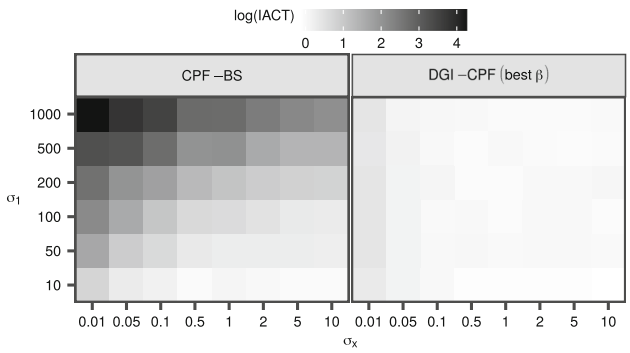

Fig. 3 The log (IACT) of the CPF-BS (left) and the best case DGI-CPF (right) with respect to σ_1 and σ_x in the case of the SV model and $N = 256$

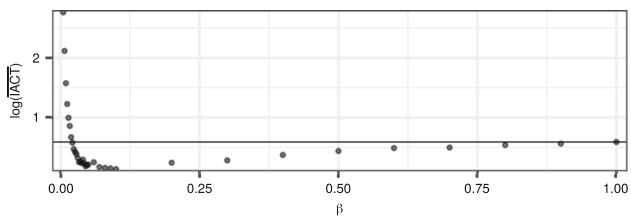

Fig. 4 The logarithm of the mean IACT over 5 replicate runs of the DGI-CPF with respect to varying β. The dataset was simulated from the SV model with parameters $\sigma_x = 1$ and $\sigma_1 = 50$ and fixed in each replicate run of the algorithm. N was set to 128. The horizontal line depicts the performance of the CPF-BS

β parameter of the DGI-CPF cannot be chosen optimally. Indeed, the choice of β can have a substantial effect on the mixing. Figure 4 illustrates this in the case of the SV model by showing the logarithm of the mean IACT over replicate runs of the DGI-CPF, for a range of β. Here, a β of approximately 0.125 seems to yield close to optimal performance, but if the β is chosen too low, the sampling efficiency is greatly reduced, rendering the CPF-BS more effective.

This highlights the importance of choosing an appropriate value for β, and motivates our adaptive DGI-CPF, that is, Algorithm 5 together with Algorithm 8. We explored the effect of the target acceptance rate $\alpha_* \in \{0.01, 0.02, \ldots, 1\}$, with the same datasets and parameter configurations as before. Figure 5 summarises the results for both the SV and RW models, in comparison with the CPF-BS. The results indicate that with a wide range of target acceptance rates, the adaptive DGI-CPF exhibits improved mixing over the CPF-BS. When N increases, the optimal values for α_* appear to tend to one. However, in practice, we are interested in a moderate N, for which the results suggest that the best candidates for values of α_* might often be found in the range from 0.7 to 0.9.

For the CPF-BS, the mean IRE is approximately constant, which might suggest that the optimal number of particles is more than 512. In contrast, for an appropriately tuned

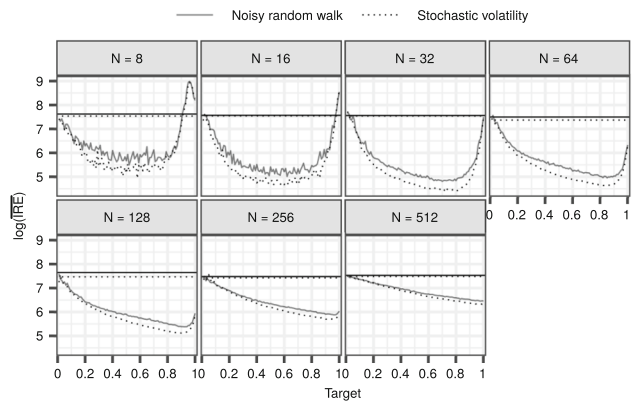

Fig. 5 The logarithm of the mean IRE over the parameter configurations with the adaptive DGI-CPF and varying target acceptance rates. The horizontal lines depict the mean performance of the CPF-BS

DGI-CPF, the mean IRE is optimised by $N = 32$ in this experiment.

4.2 Comparing FDI-CPF and particle Gibbs

Next, we turn to study a fully diffuse initialisation. In this case, M_1 is improper, and we cannot use the CPF directly. Instead, we compare the performance of the adaptive FDI-CPF with what we call the diffuse particle Gibbs (DPG-BS) algorithm. The DPG-BS is a standard particle Gibbs algorithm, where the first latent state x_1 is regarded as a 'parameter', that is, the algorithm alternates between the update of x_1 conditional on $x_{2:T}$ using a random walk Metropolis-within-Gibbs step, and the update of the latent state variables $x_{2:T}$ conditional on x_1 using the CPF-BS. We also adapt the Metropolis-within-Gibbs proposal distribution Q_{DPG} of the DPG-BS, using the RAM algorithm (cf. Vihola 2020). For further details regarding our implementation of the DPG-BS, see Appendix B.

We used a similar simulation experiment as with the adaptive DGI-CPF in Sect. 4.1, but excluding σ_1, since the initial distribution was now fully diffuse. The target acceptance rates in the FDI-CPF with the ASWAM adaptation were again varied in $\alpha_* \in \{0.01, 0.02, \ldots, 1\}$ and the scaling factor in the AM adaptation was set to $c = 2.38^2$. In the DPG-BS, the target acceptance rate for updates of the initial state using the RAM algorithm was fixed to 0.441 following Gelman et al. (1996).

Figure 6 shows results with the RW model for the DPG-BS, the FDI-CPF with the AM adaptation, and the FDI-CPF with the ASWAM adaptation using the best value for α_*. The FDI-CPF variants appear to perform better and improve upon the performance of the DPG-BS especially with small σ_x. Similar to Figs. 2 and 3, the optimal N minimizing the IRE depends on the value of σ_x: smaller values of σ_x call for higher number of particles.

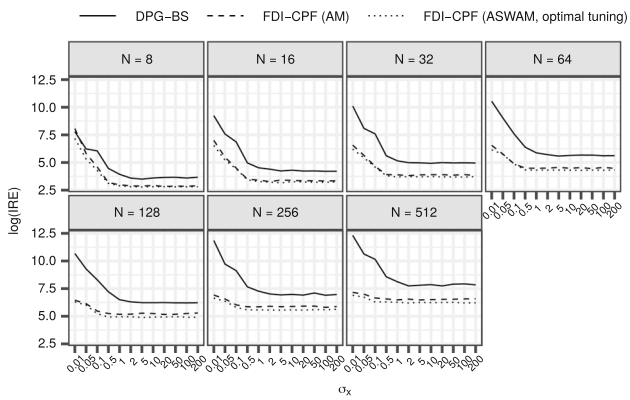

Fig. 6 The log (IRE) for the DPG-BS, FDI-CPF with the AM adaptation and the best case FDI-CPF with the ASWAM adaptation to the datasets generated with varying σ_x from the RW model

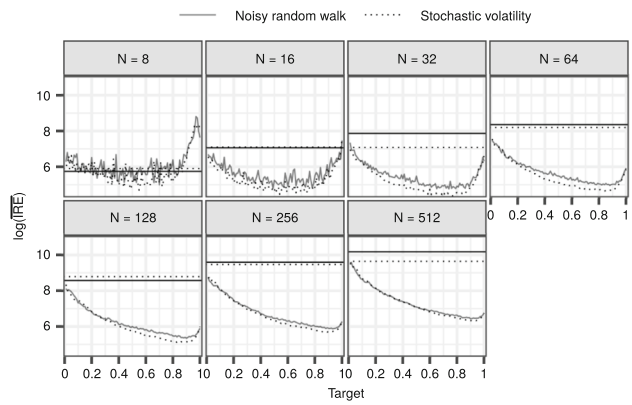

Fig. 7 A comparison of the FDI-CPF with the ASWAM adaptation against the DPG-BS. The horizontal axis shows the target acceptance rate α_* used in the adaptive FDI-CPF. The logarithm of the mean IRE on the vertical axis is computed over the different σ_x values. The black horizontal lines show the performance with the DPG-BS

The performance of the adaptive FDI-CPF appears similar regardless of the adaptation used, because the chosen scaling factor $c = 2.38^2$ for a univariate model was close to the optimal value found by the ASWAM variant in this example. We experimented also with $c = 1$, which led to less efficient AM, in the middle ground between the ASWAM and the DPG-BS.

The IACT for the DPG-BS stays approximately constant with increasing N, which results in a log (IRE) that increases roughly by a constant as N increases. This is understandable, because in the limit as $N \to \infty$, the CPF-BS (within the DPG-BS) will correspond to a Gibbs step, that is, a perfect sample of $x_{2:T}$ conditional on x_1. Because of the strong correlation between x_1 and x_2, even an 'ideal' Gibbs sampler remains inefficient, and the small variation seen in the panels for the DPG-BS is due to sampling variability. The results for the SV model, with similar findings, are shown in Online Resource 1 (Fig. 3).

Figure 7 shows the logarithm of the mean IRE of the FDI-CPF with the ASWAM adaptation with respect to varying target acceptance rate α_*. The results are reminiscent of Fig. 5 and show that with a moderate fixed N, the FDI-CPF with the ASWAM adaptation outperforms the DPG-BS with a wide range of values for α_*. The optimal value of α_* seems to tend to one as N increases, but again, we are mostly concerned with moderate N. For a well-tuned FDI-CPF the minimum mean IRE is found when N is roughly between 32 and 64.

4.3 The relationship between state dimension, number of particles and optimal target acceptance rate

A well chosen value for the target acceptance rate α_* appears to be key for obtaining good performance with the adaptive DGI-CPF and the FDI-CPF with the ASWAM adaptation. In Sects. 4.1–4.2, we observed a relationship between N and the optimal target acceptance rate, denoted here by α_{opt}, with two univariate HMMs. It is expected that α_{opt} is generally somewhat model-dependent, but in particular, we suspected that the methods might behave differently with models of different state dimension d.

In order to study the relationship between N, d and α_{opt} in more detail, we considered a simple multivariate normal model with $T = 1$, $M_1(x) \propto 1$, and $G_1(x_1) = N(x_1; 0, \sigma I_d)$, the density of d independent normals. We conducted a simulation experiment with 6000 iterations plus 500 burn-in. We applied the FDI-CPF with the ASWAM adaptation with all combinations of $N \in \{2^4, 2^5, \ldots, 2^{11}\}$, $\alpha_* \in \{0.01, 0.02, \ldots, 1\}$, $\sigma \in \{1, 5, 10, 50, 100\}$, and with dimension $d \in \{1, 2, \ldots, 10\}$. Unlike before, we monitor the IACT over the samples of x_1 as an efficiency measure.

Figure 8 summarises the results of this experiment. With a fixed state dimension, α_{opt} tended towards 1 with increasing numbers of particles N, as observed with the RW and SV models above. With a fixed number of particles N, α_{opt} appears to get smaller with increasing state dimension d, but the change rate appears slower with higher d. Again, with moderate values for N and d, the values in the range 0.7–0.9 seem to yield good performance.

Figure 9 shows a different view of the same data: $\text{logit}(\alpha_{\text{opt}})$ is plotted with respect to $\log(N)$ and d. Here, we computed α_{opt} by taking the target acceptance rate that produced the lowest IACT in the simulation experiment, for each value of σ, N and d. At least with moderate α_{opt} and N, there appears to be a roughly linear relationship between $\text{logit}(\alpha_{\text{opt}})$ and $\log(N)$, when d is fixed. However, because of the lack of theoretical backing, we do not suggest to use such a simple model for choosing α_{opt} in practice.

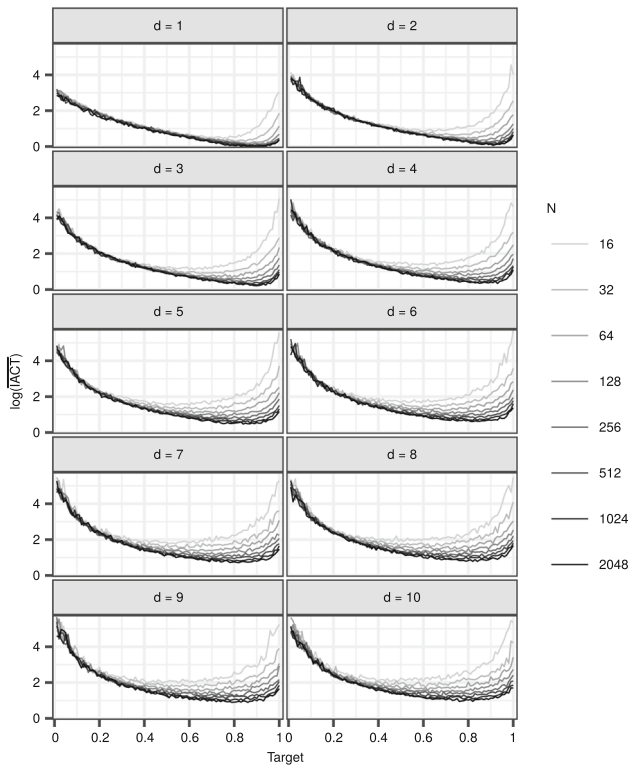

Fig. 8 The effect of state dimension d, number of particles N and target acceptance rate α_* on the logarithm of the mean IACT in the multivariate normal model. The means are computed over the different σ in the simulation experiment

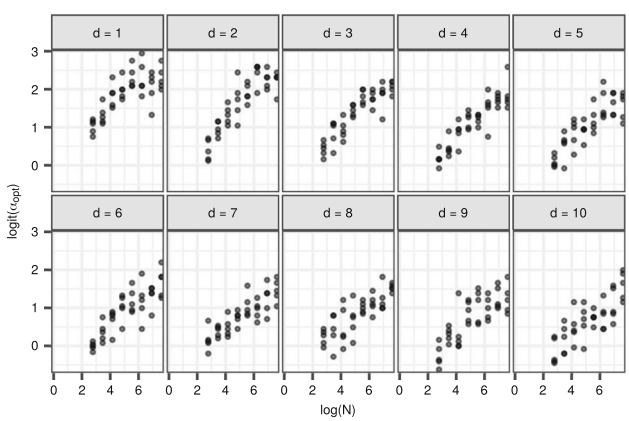

Fig. 9 The best target acceptance rate α_{opt} with respect to the number of particles N and state dimension d on the multivariate normal model

4.4 Modelling the COVID-19 epidemic in Finland

Our final experiment is a realistic inference problem arising from the modelling of the progress of the COVID-19 epidemic in Uusimaa, the capital region of Finland. Our main interest is in estimating the time-varying transmission rate, or the basic reproduction number $\mathscr{R}_0$, which is expected to change over time, because of a number of mitigation actions and social distancing. The model consists of a discrete-time 'SEIR' stochastic compartment model, and a dynamic model for $\mathscr{R}_0$; such epidemic models have been used earlier in different contexts (e.g. Shubin et al. 2016).

We use a simple SEIR without age/regional stratification. That is, we divide the whole population N_{pop} to four separate states: susceptible (S), exposed (E), infected (I) and removed (R), so that $N_{\text{pop}} = S + E + I + R$, and assume that N_{pop} is constant. We model the transformed $\mathscr{R}_0$, denoted by ρ, such that $\mathscr{R}_0 = \mathscr{R}_0{}^{\max}\text{logit}^{-1}(\rho)$, where $\mathscr{R}_0{}^{\max}$ is the maximal value for $\mathscr{R}_0$. The state vector of the model at time k is, therefore, $X_k = (S_k, E_k, I_k, R_k, \rho_k)$. One step of the SEIR is:

$$
\begin{aligned}
S_{k+1} &= S_k - \Delta E_{k+1},\\
E_{k+1} &= E_k + \Delta E_{k+1} - \Delta I_{k+1},\\
I_{k+1} &= I_k + \Delta I_{k+1} - \Delta R_{k+1},\\
R_{k+1} &= R_k + \Delta R_{k+1},\\
\rho_{k+1} &= \rho_k + \Delta\rho_{k+1},
\end{aligned}
$$

where the increments are as distributed as follows:

$$
\begin{aligned}
\Delta E_{k+1} &\sim \text{Binomial}(S_k, p_\beta),\ \ p_\beta = 1 - \exp\left(-\beta_k(I_k/N_{\text{pop}})\right),\\
\Delta I_{k+1} &\sim \text{Binomial}(E_k, p_a),\ \ p_a = 1 - \exp\left(-a\right),\\
\Delta R_{k+1} &\sim \text{Binomial}(I_k, p_\gamma),\ \ p_\gamma = 1 - \exp\left(-\gamma\right),\\
\Delta\rho_{k+1} &\sim \text{Normal}(0, \sigma^2).
\end{aligned}
$$

Here, $\beta_k = \mathscr{R}_0{}^{\max}\text{logit}^{-1}(\rho_k)p_\gamma$ is the time-varying infection rate, and a^{-1} and γ^{-1} are the mean incubation period and recovery time, respectively. Finally, the random walk parameter σ controls how fast $(\rho_k)_{k\geq 2}$ can change.

The data we use in the modelling consist of the daily number of individuals tested positive for COVID-19 in Uusimaa (Finnish Institute for Health and Welfare 2020). We model the counts with a negative binomial distribution dependent on the number of infected individuals:

$$
Y_k \sim \text{NegativeBinomial}\left(ep_\gamma \frac{p}{1-p} I_k, p\right). \tag{8}
$$

Here, the parameter e denotes sampling effort, that is, the average proportion of infected individuals that are observed, and p is the failure probability of the negative binomial distribution, which controls the variability of the distribution.

In the beginning of the epidemic, there is little information available regarding the initial states, rendering the diffuse initialisation a convenient strategy. We set

$$
\begin{aligned}
&M_1(S_1, E_1, I_1, R_1, \rho_1)\\
&\quad = 1(S_1 + E_1 + I_1 = N_{\text{pop}})1(S_1, E_1, I_1 \geq 0)1(R_1 = 0), \quad (9)
\end{aligned}
$$

where the number of removed $R_1 = 0$ is justified because we assume all were susceptible to COVID-19, and that the epidemic has started very recently.

In addition to the state estimation, we are interested in estimating the parameters σ and p. We assign the prior $N(-2.0, (0.3)^2)$ to $\log(\sigma)$ to promote gradual changes in $\mathscr{R}_0$, and an uninformative prior, $N(0, 10^2)$, for $\text{logit}(p)$. The remaining parameters are fixed to $N_{\text{pop}} = 1638469$, ${\mathscr{R}_0}^{\max} = 10$, $a = 1/3$, $\gamma = 1/7$ and $e = 0.15$, which are in part inspired by the values reported by the Finnish Institute for Health and Welfare.

We used the AAI-PG (Algorithm 9) with the FDI-CPF with the ASWAM adaptation, and a RAM adaptation (Vihola 2012) for σ and p, (i.e. in the Lines 2–3 of Algorithm 9). The form of (9) leads to the version of the FDI-CPF discussed in Sect. 3.2 where the initial distribution is uniform with constraints. We use a random walk proposal to generate proposals $(\rho_1, E_1, I_1) \rightarrow (\rho_1^*, E_1^*, I_1^*)$, round E_1^* and I_1^* to the nearest integer, and then set $R_1^* = 0$ and $S_1^* = N_{\text{pop}} - E_1^* - I_1^* - R_1^*$. We refer to this variant of the AAI-PG as the FDI-PG algorithm. Motivated by our findings in Sects. 4.1–4.3, we set the target acceptance rate α_* in the FDI-CPF (within the FDI-PG) to 0.8.

As an alternative to the FDI-PG we also used a particle Gibbs algorithm that treats σ, p as well as the initial states E_1, I_1 and ρ_1 as parameters, using the RAM to adapt the random walk proposal (Vihola 2012). This algorithm is the DPG-BS detailed in Appendix B with the difference that the parameters σ and p are updated together with the initial state, and p^{DPG} additionally contains all terms of (6) which depend on σ and p.

We ran both the FDI-PG and the DPG-BS with $N = 64$ a total of $n = 500,000$ iterations plus $10,000$ burn-in, and thinning of 10. Figures 10 and 11 show the first 50 autocorrelations and traceplots of E_1, I_1, $(\mathscr{R}_0)_1$, σ and p, for both methods, respectively. The corresponding IACT and n_{eff} as well as credible intervals for the means of these variables are shown in Table 1. The FDI-PG outperformed the DPG-BS with each variable. However, as is seen from Online Resource 1 (Fig. 4), the difference is most notable with the initial states, and the relative performance of the DPG-BS approaches that of the FDI-PG with increasing state index. The slow improvement in the mixing of the state variable R occurs because of the cumulative nature of the variable in the model, and the slow mixing of early values of I. We note that even though the mixing with the DPG-BS was worse, the inference with $500,000$ iterations leads in practice to similar findings. However, the FDI-PG could provide reliable inference with much less iterations than the DPG-BS. The marginal density estimates of the initial states and parameters are shown in Online Resource 1 (Fig. 5). The slight discrepancies in the density estimates of E_1 and I_1 between

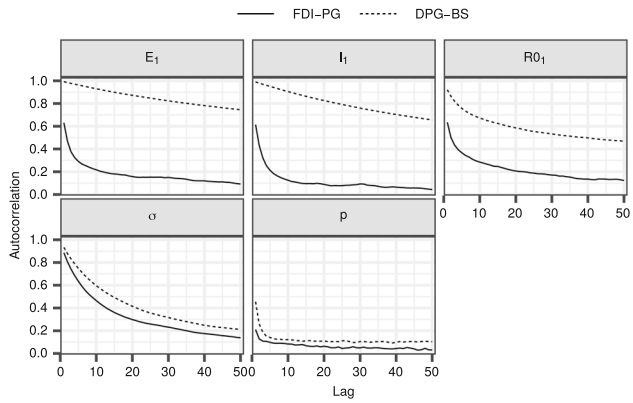

Fig. 10 The first 50 autocorrelations for the model parameters and initial states with the FDI-PG and the DPG-BS computed after thinning the total 500000 samples to every 10th sample

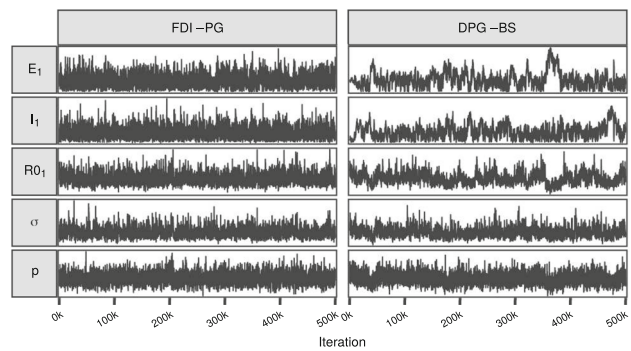

Fig. 11 Traceplots for the initial states and model parameters for the SEIR model with the FDI-PG and the DPG-BS. The 5000 samples shown per method and parameter correspond to every 100th sample of the total 500000 samples simulated

the methods are likely because of the poor mixing of these variables with the DPG-BS.

We conclude with a few words about our findings regarding the changing transmission rate, which may be of some independent interest. Figure 12 displays the data and a posterior predictive simulation, and the estimated distribution of $\mathscr{R}_0$ computed by the FDI-PG with respect to time, with annotations about events that may have had an effect on the spread of the epidemic, and/or the data. The initial $\mathscr{R}_0$ is likely somewhat overestimated, because of the influx of infections from abroad, which were not explicitly modelled. There is an overall decreasing trend since the beginning of 'lockdown', that is, when the government introduced the first mitigation actions, including school closures. Changes in the testing criteria likely cause some bias soon after the change, but no single action or event stands out.

Interestingly, if we look at our analysis, but restrict our focus up to the end of April, we might be tempted to quantify how much certain mitigation actions contribute to the suppression of the transmission rate in order to build projections using scenario models (cf. Anderson et al. 2020).

Table 1 The integrated autocorrelation time, effective sample size and credible intervals of the mean for the initial states and parameters in the SEIR model

Variable	IACT		n_{eff}		95% mean CI	
	FDI-PG	DPG-BS	FDI-PG	DPG-BS	FDI-PG	DPG-BS
E_1	30.087	882.583	1661.838	56.652	(353.888, 374.054)	(301.379, 423.106)
I_1	14.296	626.963	3497.603	79.75	(165.697, 172.388)	(155.374, 203.458)
$(\mathscr{R}_0)_1$	32.168	436.755	1554.331	114.481	(3.41, 3.513)	(3.266, 3.636)
σ	41.261	114.919	1211.796	435.088	(0.15, 0.154)	(0.147, 0.154)
p	5.18	38.178	9652.794	1309.647	(0.134, 0.135)	(0.133, 0.135)

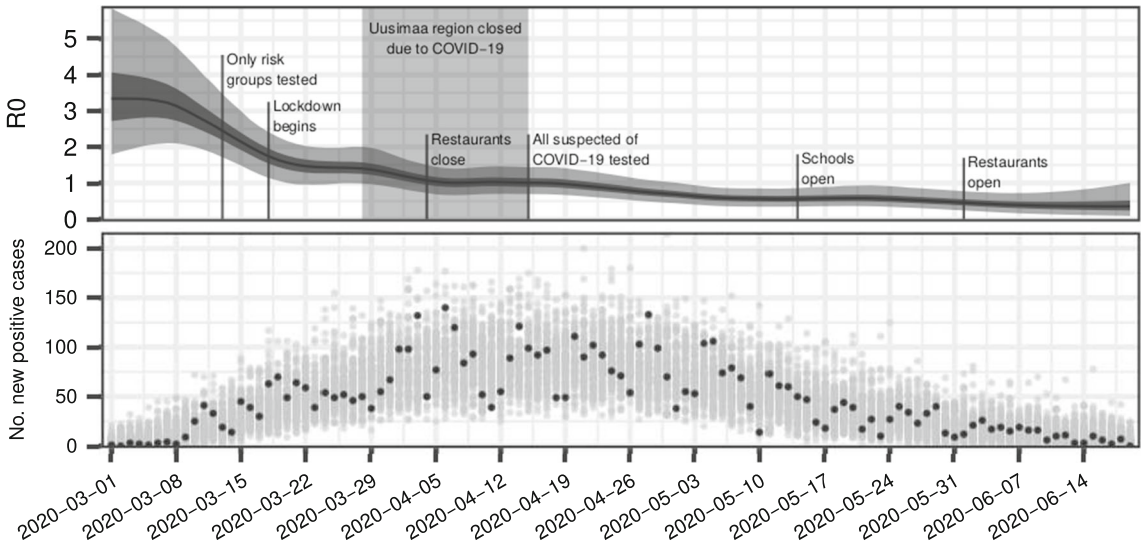

Fig. 12 The distribution of the basic reproduction number $\mathscr{R}_0$ (top) and a posterior predictive simulation (bottom) based on the posterior distribution computed with the FDI-PG. The plot for $\mathscr{R}_0$ shows the median in black and probability intervals (75% and 95%) in shades of grey. The black points in the bottom plot represent the data used. The grey points represent observations simulated conditional on the posterior distribution of the model parameters and states

However, when the mitigation measures have been gradually lifted by opening the schools and restaurants, the openings do not appear to have had notable consequences, at least until now. It is possible that at this point, the number of infections was already so low, that it has been possible to test all suspected cases and trace contacts so efficiently, and that nearly all transmission chains have been contained. Also, the public may have changed their behaviour, and are now following the hygiene and social distancing recommendations voluntarily. Such a behaviour is, however, subject to change over time.

5 Discussion

We presented a simple general auxiliary variable method for the CPF for HMMs with diffuse initial distributions and focused on two concrete instances of it: the FDI-CPF for a uniform initial density M_1 and the DGI-CPF for a Gaussian M_1. We introduced two mechanisms to adapt the FDI-CPF automatically: the adaptive Metropolis (AM) of Haario et al. (2001) and a method similar to a Rao–Blackwellised adaptive scaling within adaptive Metropolis (ASWAM) (cf. Andrieu and Thoms 2008), and provided a proof of their consistency. We also suggested an adaptation for the DGI-CPF, based on an acceptance rate optimisation. The FDI-CPF or the DGI-CPF, including their adaptive variants, may be used directly within a particle Gibbs as a replacement for the standard CPF.

Our experiments with a noisy random walk model and a stochastic volatility model demonstrated that the DGI-CPF and the FDI-CPF can provide orders of magnitude speed-ups relative to a direct application of the CPF and to diffuse initialisation using particle Gibbs, respectively. Improvement was substantial also in our motivating practical example, where we applied the adaptive FDI-CPF (within particle Gibbs) in the analysis of the COVID-19 epidemic in Finland, using a stochastic 'SEIR' compartment model with changing transmission rate. Latent compartment models are, more generally, a good example where our approach can be useful: there is substantial uncertainty in the initial states, and it

is difficult to design directly a modified model that leads to efficient inference.

Our adaptation schemes are based on the estimated covariance matrix and a scaling factor which can be adapted using acceptance rate optimisation. For the latter, we found empirically that with a moderate number of particles, good performance was often reached with a target acceptance rate ranging in 0.7–0.9. We emphasise that even though we found this '0.8 rule' to work well in practice, it is only a heuristic, and the optimal target acceptance rate may depend on the model of interest. Related to this, we investigated how the optimal target acceptance rate varied as a function of the number of particles and state dimension in a multivariate normal model, but did not find a clear pattern. Theoretical verification of the acceptance rate heuristic, and/or development of more refined adaptation rules, is left for future research. We note that while the AM adaptation performed well in our limited experiments, the ASWAM may be more appropriate when used within particle Gibbs (cf. Vihola 2020). The scaling of the AM remains similarly challenging, due to the lack of theory for tuning.

Acknowledgements We wish to acknowledge CSC, IT Center for Science, Finland, for computational resources, and thank Arto Luoma for inspiring discussions that led to the COVID-19 example.

Funding This work was supported by Academy of Finland Grant 315619.

Data Availability Statement All data analysed in this work are either freely available or available at https://nextcloud.jyu.fi/index.php/s/zjeiwDoxaegGcRe.

Compliance with ethical standards

Conflicts of interest The authors declare that they have no conflict of interest.

Code availability The code used to produce the results in this work is freely available at https://github.com/skarppinen/cpf-diff-init.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

Appendix

A Proof of Theorem 1

For a finite signed measure ξ, the total variation of ξ is defined as $\|\xi\|_{\mathrm{tv}} = \sup_{\|f\|_\infty \le 1} \xi(f)$, where $\|f\|_\infty = \sup_x |f(x)|$, and the supremum is over measurable real-valued functions f, and $\xi(f) = \int f \mathrm{d}\xi$. For Markov transitions P and P', define $d(P, P') = \sup_x \|P(x, \cdot) - P'(x, \cdot)\|_{\mathrm{tv}}$.

In what follows, we adopt the following definitions:

Definition 1 Consider Lines 3 and 4 of Algorithm 1 with $\tilde{X}_1^{(1:N)} = \tilde{x}_1^{(1:N)}$ and $\dot{x}_{2:T}$, and define:

(i) $P_{\mathrm{CPF}}(\tilde{x}_1^{(1:N)}, \dot{x}_{2:T}; \cdot)$ as the law of $\tilde{X}_{1:T}^{(B_{1:T})}$, and
(ii) (In case PICKPATH- BS is used:) $\tilde{P}_{\mathrm{CPF}}(\tilde{x}_1^{(1:N)}, \dot{x}_{2:T}; \cdot)$ as the law of $\big(\tilde{X}_{1:T}^{(B_{1:T})}, (B_1, V^{(1:N)}, \tilde{X}_1^{(1:N)})\big)$.

Consider then Algorithm 1 with parameterised $Q = Q_\zeta$, and define, analogously:

(iii) P_ζ is the Markov transition from $\dot{x}_{1:T}$ to $\tilde{X}_{1:T}^{(B_{1:T})}$.
(iv) $\tilde{P}_\zeta$ is the Markov transition from from $(\dot{x}_{1:T}, \cdot)$ to $\big(\tilde{X}_{1:T}^{(B_{1:T})}, (B_1, V^{(1:N)}, \tilde{X}_1^{(1:N)})\big)$.

Lemma 1 *We have* $d(P_\zeta, P_{\zeta'}) \le N d(Q_\zeta, Q_{\zeta'})$ *and* $d(\tilde{P}_\zeta, \tilde{P}_{\zeta'}) \le N d(Q_\zeta, Q_{\zeta'})$.

Proof Let $(\hat{P}_{\mathrm{CPF}}, \hat{P}_\zeta) \in \{(P_{\mathrm{CPF}}, P_\zeta), (\tilde{P}_{\mathrm{CPF}}, \tilde{P}_\zeta)\}$ and take measurable real-valued function f on the state space of $\hat{P}_\zeta$ with $\|f\|_\infty = 1$.

We may write

$$\begin{aligned}
&\hat{P}_\zeta(\dot{x}_{1:T}, f) \\
&= \int Q_\zeta(\dot{x}_1, \mathrm{d}x_0)\Big[\int \delta_{\dot{x}_1}(\mathrm{d}\tilde{x}_1^{(1)}) \\
&\quad \prod_{k=2}^{N} Q_\zeta(x_0, \mathrm{d}\tilde{x}_1^{(k)}) \hat{P}_{\mathrm{CPF}}(\tilde{x}_1^{(1:N)}, \dot{x}_{2:T}, f)\Big],
\end{aligned} \tag{10}$$

and therefore, upper bound

$$\begin{aligned}
&|\hat{P}_\zeta(\dot{x}_{1:T}, f) - \hat{P}_{\zeta'}(\dot{x}_{1:T}, f)| \\
&\quad \le |Q_\zeta(\dot{x}_1, g_0^{(\dot{x}_{1:T})}) - Q_{\zeta'}(\dot{x}_1, g_0^{(\dot{x}_{1:T})})| \\
&\qquad + \sum_{i=2}^{N} \int Q_{\zeta'}(\dot{x}_1, \mathrm{d}x_0) |Q_\zeta(x_0, g_i^{(\dot{x}_{1:T}, x_0)}) \\
&\qquad - Q_{\zeta'}(x_0, g_i^{(\dot{x}_{1:T}, x_0)})|
\end{aligned}$$

with functions defined below, which satisfy $\|g_0^{(\dot{x}_{1:T})}\|_\infty \leq 1$ and $\|g_i^{(\dot{x}_{1:T},x_0)}\|_\infty \leq 1$:

$$g_0^{(\dot{x}_{1:T})}(x_0) = \int \delta_{\dot{x}_1}(\mathrm{d}\tilde{x}_1^{(1)}) \prod_{k=2}^{N} Q_\zeta(x_0, \mathrm{d}\tilde{x}_1^{(k)}) P_{\mathrm{CPF}}(\tilde{x}_1^{(1:N)}, \dot{x}_{2:T}, f),$$

$$g_i^{(\dot{x}_{1:T},x_0)}(\tilde{x}_1^{(i)}) = \delta_{\dot{x}_1}(\mathrm{d}\tilde{x}_1^{(1)}) \prod_{k=2}^{i-1} Q_{\zeta'}(x_0, \mathrm{d}\tilde{x}_1^{(k)}) \prod_{k=i+1}^{N} Q_\zeta(x_0, \mathrm{d}\tilde{x}_1^{(k)}) P_{\mathrm{CPF}}(\tilde{x}_1^{(1:N)}, \dot{x}_{2:T}, f).$$

□

The following result is direct:

Lemma 2 *Let Q_Σ stand for the random walk Metropolis type kernel with increment proposal distribution q_Σ, and with target function $M_1 \geq 0$, that is, a transition probability of the form:*

$$Q_\Sigma(x, A) = \int_A q_\Sigma(\mathrm{d}z) \min\left\{1, \frac{M_1(x+z)}{M_1(x)}\right\} + 1(x \in A)\left(1 - \int q_\Sigma(\mathrm{d}z) \min\left\{1, \frac{M_1(x+z)}{M_1(x)}\right\}\right).$$

Then, $\|Q_\Sigma(x, \cdot) - Q_{\Sigma'}(x, \cdot)\|_{\mathrm{tv}} \leq 2\|q_\Sigma - q_{\Sigma'}\|_{\mathrm{tv}}$.

The following result is from (Vihola 2011, proof of Proposition 26):

Lemma 3 *Let $q_\Sigma(x, \mathrm{d}y)$ stand for the centred Gaussian distribution with covariance Σ, or the centred multivariate t-distribution with shape Σ and some constant degrees of freedom $\nu > 0$. Then, for any $0 < b_\ell < b_u < \infty$ there exists a constant $c = c(b_\ell, b_u) < \infty$ such that for all Σ, Σ' with all eigenvalues within $[b_\ell, b_u]$,*

$$\|q_\Sigma - q_{\Sigma'}\|_{\mathrm{tv}} \leq c\|\Sigma - \Sigma'\|,$$

where the latter stands for the Frobenius norm in $\mathbb{R}^d$.

Assumption 2 (Mixing) The potentials are bounded:

(i) $\|G_k\|_\infty < \infty$ for all $k = 1, \ldots, T$.

Furthermore, there exists $\epsilon > 0$ and probability measures ν_ζ such that for all $\zeta \in \mathsf{Z}$:

(ii) $Q_\zeta(x_0, A) \geq \epsilon \nu_\zeta(A)$ for all $x_0 \in \mathsf{X}$ and measurable $A \subset \mathsf{X}$.
(iii) $\int \nu_\zeta(\mathrm{d}x_0) Q_\zeta(x_0, \mathrm{d}x_1) G_1(x_1) \prod_{k=2}^{T} M_k(x_{k-1}, \mathrm{d}x_k) G_k(x_{k-1}, x_k) \mathrm{d}x_{1:T} \geq \epsilon$.

Lemma 4 *Suppose that Assumption 2 holds, then the kernels P_ζ and $\tilde{P}_\zeta$ satisfy simultaneous minorisation conditions, that is, there exists $\delta > 0$ and probability measures $\nu_\zeta, \tilde{\nu}_\zeta$, such that*

$$P_\zeta^k(x_{1:T}, \cdot) \geq \delta \nu_\zeta(\cdot) \quad \textit{and} \quad \tilde{P}_\zeta^k(\tilde{x}_{1:T}, \cdot) \geq \delta \tilde{\nu}_\zeta(\cdot),$$

for all $x_{1:T} \in \mathsf{X}$, $\tilde{x}_1^{(1:N)} \in \mathsf{X}^N$, and $\zeta \in \mathsf{Z}$.

Proof For $\hat{P}_\zeta \in \{P_\zeta, \tilde{P}_\zeta\}$, we may write as in the proof of Lemma1

$$\hat{P}_\zeta(x_{1:T}, \cdot) = \int Q_\zeta(x_1, \mathrm{d}x_0) \hat{P}^*_{\mathrm{CPF},\zeta,x_0}(x_{1:T}, \cdot),$$

where the latter term refers to the term in brackets in (10) — the transition probability of a conditional particle filter, with reference $x_{1:T}$, and the Feynman–Kac model $\check{M}_1^{(\zeta,x_0)}(\mathrm{d}x_1) = Q_\zeta(x_0, \mathrm{d}x_1)$, $M_{2:T}$ and $G_{1:T}$, whose normalised probability we call π^*_{ζ,x_0}. Assumption 2, 2 and 2 guarantee that $P^*_{\mathrm{CPF},\zeta,x_0}(x_{1:T}, \mathrm{d}x'_{1:T}) \geq \varepsilon \pi^*_{\zeta,x_0}(\mathrm{d}x'_{1:T})$, where $\hat{\epsilon} > 0$ is independent of x_0 and ζ (Andrieu et al. 2018, Corollary 12). Note that the same conclusion holds also with backward sampling, because it is only a further Gibbs step to the standard CPF. Likewise, in case of $\tilde{P}_\zeta$, the result holds because we may regard $\tilde{P}_\zeta$ as an augmented version of P_ζ (e.g. Franks and Vihola 2020). We conclude that

$$\hat{P}_\zeta(x_{1:T}, \cdot) \geq \epsilon\hat{\epsilon} \int \nu_\zeta(\mathrm{d}x_0) \pi^*_{\zeta,x_0}(\cdot),$$

where the integral defines a probability measure independent of $x_{1:T}$. □

We may write the k:th step of Algorithm 5 as:

(i) $(X_k, \xi_k) \sim \tilde{P}_{\zeta_{k-1}}(X_{k-1}, \cdot)$,
(ii) $\zeta_k^* = \zeta_{k-1} + \eta_k H(\zeta_{k-1}, X_k, \xi_k)$,

where H correspond to Algorithm 6 or 7, respectively. The stability may be enforced by introducing the following optional step:

(iii) $\zeta_k = \zeta_k^* 1(\zeta_k \in \mathsf{Z}) + \zeta_{k-1} 1(\zeta_k^* \notin \mathsf{Z})$,

which ensures that $\zeta \in \mathsf{Z}$, the feasible set for adaptation.

Proof (Proof of Theorem 1) The result follows by (Saksman and Vihola 2010, Theorem 2), as (A1) is direct, Lemma 4 implies (A2) with $V \equiv 1$, $\lambda_n = 0$, $b_n = 1$, $\delta_n = \delta$ and $\epsilon = 0$, Lemmas 2 and 3 imply (A3), and (A4) holds trivially, as $\|H(\cdot)\|_\infty < \infty$, thanks to the compactness of D. □

B Details of the DPG-BS algorithm

The diffuse particle Gibbs algorithm targets (2) by alternating the sampling of $x_{2:T}$ given x_1, and x_1 given $x_{2:T}$. Hence, the algorithm is simply particle Gibbs where the initial state is treated as a parameter. Define

$$p^{\mathrm{DPG}}(x_1 \mid x_{2:T}) \propto M_1(x_1)G_1(x_1)M_2(x_2 \mid x_1)G_2(x_1, x_2).$$

With this definition, the DPG-BS algorithm can be written as in Algorithm 10. Lines 3–5 constitute a CPF-BS update for $x_{2:T}$, and Line 6 updates x_1. A version of the RAM algorithm (Vihola 2012) (Algorithm 11) is used for adapting the normal proposal used in sampling x_1 from p^{DPG}.

Algorithm 10 DPG-BS$(X_1^{(0)}, \dot{x}_{2:T}^{(0)}; \pi, N)$

1: Set $S_0 = I, \alpha_* = 0.441, \eta^{\max} = 0.5, \gamma = 0.66$.
2: **for** j in $1, \ldots, n$ **do**
3: Simulate $\tilde{X}_2^{(2:N)} \sim M_2(\cdot \mid X_1^{(j-1)})$ and set $\tilde{X}_2^{(1)} = \dot{x}_2^{(0)}$.
4: $(\tilde{X}_{2:T}^{(1:N)}, W_{2:T}^{(1:N)}, A_{2:T-1}^{(1:N)}) \leftarrow$ F-CPF$(\dot{x}_{3:T}, \tilde{X}_2^{(1:N)}; M_{3:T}, G_{2:T}, N)$.
5: $(B_{2:T}, \xi) \leftarrow$ PickPath-BS$(\tilde{X}_{2:T}^{(1:N)}, W_{2:T}^{(1:N)}, A_{2:T-1}^{(1:N)}, M_{3:T}, G_{3:T})$
6: $(X_1^{(j)}, S_j) \leftarrow \mathrm{RAM}(p^{\mathrm{DPG}}(\cdot \mid \tilde{X}_{2:T}^{(B_{2:T})}), X_1^{(j-1)}, S_{j-1}, \alpha_*, \eta^{\max}, \gamma)$.
7: Set $\mathbf{X}^{(j)} = (X_1^{(j)}, \tilde{X}_2^{(B_2)}, \tilde{X}_3^{(B_3)}, \ldots, \tilde{X}_T^{(B_T)})$.
8: **end for**
9: **output** $\mathbf{X}^{(1:n)}$

Algorithm 11 RAM$(p, \theta^{(n-1)}, S_{n-1}, \alpha_*, \eta^{\max}, \gamma)$ (iteration n)

1: Simulate $U_n \sim N(0, I_d)$.
2: Propose $\theta^* = \theta^{(n-1)} + S_{n-1}U_n$.
3: Compute $\alpha_n = \min\left\{1, \dfrac{p(\theta^*)}{p(\theta^{(n-1)})}\right\}$.
4: With probability α_n, set $\theta^{(n)} = \theta^*$; otherwise set $\theta^{(n)} = \theta^{(n-1)}$.
5: Set $\eta_n = \min\{\eta^{\max}, dn^{-\gamma}\}$.
6: Compute S_n such that $S_n S_n' = S_{n-1}\left(I + \eta_n(\alpha_n - \alpha_*)\dfrac{U_n U_n'}{\|U_n\|^2}\right)S_{n-1}'$.
7: **output** $\theta^{(n)}, S_n$.

References

Anderson, R.M., Heesterbeek, H., Klinkenberg, D., Hollingsworth, T.D.: How will country-based mitigation measures influence the course of the COVID-19 epidemic? Lancet **395**(10228), 931–934 (2020)

Andrieu, C., Moulines, É.: On the ergodicity properties of some adaptive MCMC algorithms. Ann. Appl. Probab. **16**(3), 1462–1505 (2006)

Andrieu, C., Thoms, J.: A tutorial on adaptive MCMC. Statist. Comput. **18**(4), 343–373 (2008)

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)

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

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

Cotter, S.L., Roberts, G.O., Stuart, A.M., White, D.: MCMC methods for functions: modifying old algorithms to make them faster. Statist. Sci. **28**(3), 424–446 (2013)

Del Moral, P.: Feynman-Kac Formulae. Springer, New York (2004)

Del Moral, P., Doucet, A., Jasra, A.: Sequential Monte Carlo samplers. J. R. Stat. Soc. Ser. B Stat. Methodol. **68**(3), 411–436 (2006)

Durbin, J., Koopman, S.J.: Time Series Analysis by State Space Methods, 2nd edn. Oxford University Press, New York (2012)

Fearnhead, P., Künsch, H.R.: Particle filters and data assimilation. Ann. Rev. Stat. Its Appl. **5**, 421–449 (2018)

Fearnhead, P., Meligkotsidou, L.: Augmentation schemes for particle MCMC. Stat. Comput. **26**(6), 1293–1306 (2016)

Finnish Institute for Health and Welfare (2020) Confirmed corona cases in Finland (COVID-19). https://thl.fi/en/web/thlfi-en/statistics/statistical-databases/open-data/confirmed-corona-cases-in-finland-covid-19-, accessed on 2020-06-22

Franks, J., Vihola, M.: Importance sampling correction versus standard averages of reversible MCMCs in terms of the asymptotic variance. Stochastic Process Appl. **130**(10), 6157–6183 (2020)

Gelman, A., Roberts, G.O., Gilks, W.R.: Efficient metropolis jumping rules. Bayesian Stat. **5**, 599–607 (1996)

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

Gordon, N.J., Salmond, D.J., Smith, A.F.M.: Novel approach to nonlinear/non-Gaussian Bayesian state estimation. IEE Proceedings-F **140**(2), 107–113 (1993)

Guarniero, P., Johansen, A.M., Lee, A.: The iterated auxiliary particle filter. J. Am. Stat. Assoc. **112**(520), 1636–1647 (2017)

Haario, H., Saksman, E., Tamminen, J.: An adaptive Metropolis algorithm. Bernoulli **7**(2), 223–242 (2001)

Lee, A., Singh, S.S., Vihola, M.: Coupled conditional backward sampling particle filter. Ann. Stat. **48**(5), 3066–3089 (2020)

Lindsten, F., Jordan, M.I., Schön, T.B.: Particle Gibbs with ancestor sampling. J. Mach. Learn. Res. **15**(1), 2145–2184 (2014)

Martino, L.: A review of multiple try MCMC algorithms for signal processing. Digit. Signal Proc. **75**, 134–152 (2018)

Mendes, EF., Scharth, M., Kohn, R.: Markov interacting importance samplers. (2015) Preprint arXiv:1502.07039

Murray, L.M., Jones, E.M., Parslow, J.: On disturbance state-space models and the particle marginal Metropolis-Hastings sampler. SIAM/ASA J. Uncertain. Quantification **1**(1), 494–521 (2013)

Saksman, E., Vihola, M.: On the ergodicity of the adaptive Metropolis algorithm on unbounded domains. Ann. Appl. Probab. **20**(6), 2178–2203 (2010)

Shubin, M., Lebedev, A., Lyytikäinen, O., Auranen, K.: Revealing the true incidence of pandemic A(H1N1) pdm09 influenza in Finland during the first two seasons - an analysis based on a dynamic transmission model. PLoS Comput. Biol. **12**(3), 1–19 (2016). https://doi.org/10.1371/journal.pcbi.1004803

Vihola, M.: On the stability and ergodicity of adaptive scaling Metropolis algorithms. Stochastic Process. Appl. **121**(12), 2839–2860 (2011)

Vihola, M.: Robust adaptive Metropolis algorithm with coerced acceptance rate. Stat. Comput. **22**(5), 997–1008 (2012)

Vihola, M.: Ergonomic and reliable bayesian inference with adaptive markov chain monte carlo. In: Balakrishnan, N., Colton, T., Everitt, B., Piegorsch, W., Ruggeri, F., Teugels, J. L. (eds.) Wiley statsRef : statistics reference online, pp. 1–12. Wiley (2020). https://doi.org/10.1002/9781118445112.stat08286

Whiteley, N.: Discussion on "Particle Markov chain Monte Carlo methods". J. R. Stat. Soc. Ser. B Stat. Methodol. **72**(3), 306–307 (2010)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.