Bayesian Analysis (2022) **17**, Number 1, pp. 223–260

Ensemble MCMC: Accelerating Pseudo-Marginal MCMC for State Space Models using the Ensemble Kalman Filter

Christopher Drovandi[*,†,‡,**], Richard G. Everitt[§], Andrew Golightly[¶], and Dennis Prangle[‖]

Abstract. Particle Markov chain Monte Carlo (pMCMC) is now a popular method for performing Bayesian statistical inference on challenging state space models (SSMs) with unknown static parameters. It uses a particle filter (PF) at each iteration of an MCMC algorithm to unbiasedly estimate the likelihood for a given static parameter value. However, pMCMC can be computationally intensive when a large number of particles in the PF is required, such as when the data are highly informative, the model is misspecified and/or the time series is long. In this paper we exploit the ensemble Kalman filter (EnKF) developed in the data assimilation literature to speed up pMCMC. We replace the unbiased PF likelihood with the biased EnKF likelihood estimate within MCMC to sample over the space of the static parameter. On a wide class of different non-linear SSM models, we demonstrate that our extended ensemble MCMC (eMCMC) methods can significantly reduce the computational cost whilst maintaining reasonable accuracy. We also propose several extensions of the vanilla eMCMC algorithm to further improve computational efficiency. Computer code to implement our methods on all the examples can be downloaded from `https://github.com/cdrovandi/Ensemble-MCMC`.

Keywords: data assimilation, ensemble Kalman filter, particle filter, particle MCMC, pseudo-marginal MCMC, state space models.

1 Introduction

Particle Markov chain Monte Carlo (pMCMC, Andrieu et al., 2010) is now a popular method for performing Bayesian statistical inference on challenging state space models (SSMs) with unknown static parameters. The appeal of pMCMC is that it is a pseudo-marginal method (Andrieu and Roberts, 2009), which attempts to mimic the ideal sampler that proposes directly over the space of the static parameters and integrates

[*]School of Mathematical Sciences, Queensland University of Technology, Australia, c.drovandi@qut.edu.au
[†]ARC Centre of Excellence for Mathematical & Statistical Frontiers (ACEMS)
[‡]QUT Centre for Data Science
[§]Department of Statistics, University of Warwick, UK, richard.everitt@warwick.ac.uk
[¶]School of Mathematics, Statistics and Physics, Newcastle University, UK, Andrew.Golightly@newcastle.ac.uk
[‖]School of Mathematics, Statistics and Physics, Newcastle University, UK, Dennis.Prangle@newcastle.ac.uk
[**]Ordering of authors is alphabetical.

 https://doi.org/10.1214/20-BA1251

out the hidden states. Furthermore it is an *exact approximation*, exactly targeting the true posterior distribution.

Each static parameter proposal in pMCMC is evaluated using a particle filter (Gordon et al., 1993). Particle filters were originally proposed to solve the state space *filtering* problem: inferring the state parameters at a given time under known static parameters. To do so they propagate a set of particles through the state space model, and use a weighting and resampling process to concentrate on the particles with significant posterior weights. Using particle filters in pMCMC is costly. Firstly, each particle filter involves processing the entire dataset. Secondly, a particle filter can require a large number of particles, especially when the data are highly informative and/or the model is misspecified. This is because there must be enough particles to randomly propagate forwards to produce good matches to unlikely data. Thus, despite the popularity of pMCMC, it is generally a highly computationally intensive method.

Data assimilation (DA) is a field of research originating in the geosciences, initially based on the problem of numerical weather prediction. The task most commonly addressed in this field is the estimation of the state of a dynamical system, based on a dynamic model (usually a system of partial differential equations) and noisy and/or indirect measurements of this state. In this paper we take inspiration from the DA literature to propose a new approach to estimating the posterior distribution of static parameters in SSMs.

The field of DA has evolved in parallel to other fields in which SSMs play an important role, such as target tracking, economics and statistical ecology. The distinguishing feature of problems in DA is the large dimension of the state space. For example, in numerical weather prediction the state space consists of a representation of the state of the atmosphere across the globe, which for modern applications can have dimension d_x of the order of 10^9 (van Leeuwen, 2015). The traditional approach to estimating the dynamic state in DA is to use approaches that solely estimate the mode of the state posterior (e.g. 4DVar) or Kalman filters that make use of approximations so as to avoid storing the full state covariance, whose size scales quadratically in the state dimension. Such methods have huge practical importance and are still deployed in DA applications, but more recent research has focussed on methods that improve the accuracy of state estimation when using nonlinear dynamics. As in other fields where this the case, particle filters are an important methodology.

Particle filters are not the usual method of choice in DA. The reason is their degeneracy when used on states of high dimension (Snyder et al., 2008). This degeneracy arises due to the limitations of importance sampling in high dimensions: the variance of importance sampling estimators depends on the distance between the target and proposal distributions, and this distance grows with dimension such that the variance is only controlled by using a number of importance samples that is exponential in the dimension (see Agapiou et al., 2017 for a review). To combat this degeneracy, the approach usually taken in the particle filtering literature is to introduce diversity into the sample through MCMC updates of the state (Beskos et al., 2014). However, in many problems in DA, MCMC updates are often not available due to the use of an intractable dynamic model, and where available may have a low acceptance rate. In this case of an

intractable dynamic model, most advanced particle filtering techniques are not available, and hence the class of models to which pMCMC is practicable for inferring the posterior distribution of static parameters is relatively limited: roughly speaking the dimension of both the state space and the static parameter must be less than 10, the length of the observed data must be less than 100,000, and simulations from the model cannot be radically different from the observed data. In such situations approximate Bayesian computation (Sisson et al., 2018) or Bayesian synthetic likelihood (Price et al., 2018) can be used as an alternative, but this approach necessarily conditions the inference on summary statistics of the data which can result in posterior distributions that are very different to the truth (see Fasiolo et al., 2016 for an introduction to these alternative methods and a comparison of their performance with pMCMC). The focus in the current paper is on estimating the posterior distribution of static parameters in situations where pMCMC, for the reasons outlined above, becomes too computationally demanding when implemented on a desktop or laptop computer, and when we wish to avoid reducing the full dataset to a set of summary statistics.

An alternative means of maintaining diversity to using MCMC updates in a particle filter is given by the ensemble Kalman filter (EnKF) (Evensen, 1994; see Katzfuss et al., 2016 for a tutorial). This approach propagates a set of particles (often referred to as "ensemble members") through the dynamic model in the same way as the bootstrap particle filter, but instead uses these particles to approximate a Gaussian representation of the state distribution. This method approximates the Kalman filter and provides a means to avoid storing and manipulating the state covariance matrix. Further, it has also been shown to perform well when applied to nonlinear dynamic models. Its performance is often superior to the particle filter in cases where the particle filter suffers from degeneracy, including but not limited to the case of a state space of high dimension.

Methods from DA have also been applied to the situation of inferring an unknown static parameter simultaneously with the state. The standard approach is to augment the state vector with the static parameter, then to apply one of the previously mentioned filters to this augmented state (see, for example, Evensen, 2007). In this case, the EnKF assumes that both parameters and states follow a linear Gaussian state space model. When this assumption is unreasonable, another approach is to combine the EnKF likelihood with a particle representation of the static parameter (Stroud et al., 2018; Katzfuss et al., 2019). To mitigate degeneracy, the static parameter is allowed to dynamically vary, by adding Gaussian noise to each parameter particle. This step can be further refined via the kernel resampling strategy of Liu and West (2001). However, additional tuning parameters must be specified (e.g. to control the smoothness of the kernel) and the particle approximation can be sensitive to these choices, and in particular, the number of particles used (Vieira and Wilkinson, 2016). We note that maximum likelihood methods are also possible (see e.g. Mitchell and Houtekamer, 2000; Stroud et al., 2010; Carrassi et al., 2017).

Katzfuss et al. (2019) propose the use of the EnKF as a substitute for the particle filter in pMCMC, where the filter in each case is run with fixed static parameters to produce a likelihood estimate, which is used within a Metropolis-Hastings algorithm

(Minvielle et al., 2014 developed a similar approach in the Physics literature). By analogy with pMCMC we refer to this approach as "ensemble MCMC" (eMCMC). The contributions of this paper are: several extensions of eMCMC to further improve computational efficiency and reduce the bias in the EnKF estimate of the likelihood; and a more extensive empirical study than that of Katzfuss et al. (2019) where the focus is mainly on other EnKF-based approaches. This demonstrates that the method can be successful on a wider variety of more challenging applications. Moreover, we compare pMCMC with eMCMC, investigating the reduction in the number of particles/members required and the ability of eMCMC to cope better with informative or surprising data. We find that eMCMC exhibits a reduced computational cost relative to pMCMC and also improves the accuracy relative to a state augmentation approach using the EnKF where static parameters are treated as time-varying.

The rest of this article is structured as follows. In Section 2 we provide the necessary background on state space models, pMCMC and EnKF to understand our method. The eMCMC approach together with extensions is described in Section 3. Section 4 shows the results of our approach on a wide class of different models. We discuss limitations, further extensions and possible future research in Section 5.

2 Background

This section describes relevant existing work. Section 2.1 defines state space models. Section 2.2 describes pseudo-marginal Metropolis Hastings and the bootstrap particle filter, which can be used to perform inference for these models. Section 2.3 introduces the EnKF.

2.1 State Space Models

A state space model is a model for sequential data. It introduces a *Markov chain* of latent *states* $x_1, \ldots, x_T$. Independent noisy observations y_t are available that depend on the state x_t. Let $x = (x_1, \ldots, x_T)$ denote the collection of all latent states and $y = (y_1, \ldots, y_T)$ the collection of all observations. The model can be defined using an *evolution* distribution for $x_{t+1}|x_t, \theta$ and an *observation* distribution $y_t|x_t, \theta$. Here θ is a vector of parameters controlling the model's behaviour. We also specify a distribution for an initial state x_0. For more background on state space models see for example Särkkä (2013).

Throughout the paper we will make some standard assumptions about state space models. We will assume that each x_t and y_t are random vectors with support $\mathbb{R}^{d_x}$ and $\mathbb{R}^{d_y}$ respectively. In this section we assume the distributions above – evolution, observation and initial state – have densities $p(x_{t+1}|x_t, \theta), p(y_t|x_t, \theta)$ and $p(x_0)$. The material immediately generalises to the case where some or all of these distributions have probability mass functions instead (by interpreting these as densities with respect to the counting measure). This is required in several of our examples. A point mass can be used for the initial state distribution if the initial state is known.

As we shall see, the EnKF is restricted to certain observation models. Hence in this paper we focus on one particular case,

$$y_t|x_t,\theta \sim \mathcal{N}(Px_t, S) \tag{1}$$

where P is a $d_y \times d_x$ matrix and S is a variance matrix, possibly a function of θ. (We assume conditional independence of the y_t's given x and θ.) The EnKF can also be used where P is replaced by a time dependent matrix P_t. The case where $P = I$ gives a *complete observation* regime, in the sense that all components of x_t have a corresponding noisy observation. In contrast, a *partial observation* regime only allows observation of a subset of the components e.g. by taking P to be a projection matrix.

In practice we may wish to model states at a finer time discretisation than that at which the observation data are available. For example, consider the case where we only have observations y_t at $t = k, 2k, \ldots, kL$. This can easily be converted into the framework described above, by defining a state space model with $x^*_\tau = x_{\tau k}$ and $y^*_\tau = y_{\tau k}$ for $\tau = 1, 2, \ldots, L$.

The joint density of the latent states and observations in a state space model is:

$$p(x, y|\theta) = p(x_0) \prod_{t=1}^{T} \big[p(x_t|x_{t-1},\theta)p(y_t|x_t,\theta)\big]. \tag{2}$$

The likelihood can be found by marginalisation i.e. integrating out the latent states x,

$$L(\theta) = \int p(x_0) \prod_{t=1}^{T} \big[p(x_t|x_{t-1},\theta)p(y_t|x_t,\theta)\big]\,dx. \tag{3}$$

(If there is an observation y_0, a factor $p(y_0|x_0,\theta)$ can easily be included in (2) and (3).)

Bayesian inference assigns a prior $p(\theta)$ to the parameters and targets the posterior $p(\theta|y) \propto p(\theta)L(\theta)$. The likelihood $L(\theta)$ typically cannot be evaluated as it is a high dimensional integral. One strategy to perform inference is to instead consider an augmented target density (often of interest in its own right), the joint posterior $p(\theta, x|y) \propto p(\theta)p(x, y|\theta)$. The posterior for θ can then be obtained by marginalisation.

2.2 Pseudo-marginal MCMC, Particle Filters, and Particle MCMC

Monte Carlo algorithms are designed to sample from a target distribution, often a Bayesian posterior distribution. Markov chain Monte Carlo (MCMC) does so using a Markov chain which converges to the target distribution in the long run. Performing each update in MCMC typically requires likelihood calculations, which are not possible for models with intractable likelihoods. However it is often possible to produce unbiased likelihood estimates. Algorithm 1, *pseudo-marginal Metropolis Hastings* (PMMH) (Andrieu and Roberts, 2009), makes use of these to perform parameter inference. Unbiased likelihood estimates for state space models can be produced by *particle filter*

algorithms. Algorithm 2 presents the basic *bootstrap particle filter* (BPF) used in this paper, but there are many variations. For more details see for example Doucet and Johansen (2011), Särkkä (2013) and Fearnhead and Künsch (2018). For proof that the particle filter likelihood estimate is indeed unbiased see Del Moral (2004) and Pitt et al. (2010).

Combining the PMMH algorithm with a particle filter can target $p(\theta|y)$ for state space models. Andrieu et al. (2010) extend this approach to give *particle MCMC* (pMCMC), which targets the joint posterior $p(\theta, x|y)$; we refer the reader to this paper for a full description of pMCMC.

Algorithm 1 Pseudo-marginal Metropolis Hastings.

Input: initial state θ_0 and likelihood estimate $\hat{L}_0$, proposal density $q(\theta^*|\theta)$
for $i = 1, 2, \ldots$ **do**
1. Sample proposal θ^* from $q(\theta^*|\theta_{i-1})$.
2. Calculate $\hat{L}^*$, an estimate of $L(\theta^*)$.
3. Accept proposal with probability $\min(1, r)$ where

$$r = \frac{\hat{L}^* \pi(\theta^*) q(\theta_{i-1}|\theta^*)}{\hat{L}_{i-1} \pi(\theta_{i-1}) q(\theta^*|\theta_{i-1})}.$$

Upon acceptance let $\theta_i = \theta^*$ and $\hat{L}_i = \hat{L}^*$. Otherwise let $\theta_i = \theta_{i-1}$ and $\hat{L}_i = \hat{L}_{i-1}$.
end for
Output: $\theta_1, \theta_2, \ldots$

Algorithm 2 Bootstrap particle filter. (This algorithm drops θ from the conditioning for notational simplicity.)

Input: number of particles N
Initialise. For $i = 1, 2, \ldots, N$ sample particle $x_0^{(i)}$ from the initial state distribution and assign weight $w_0^{(i)} = 1/N$ (or, if a y_0 observation is available, compute weights as in step 3.).
for $t = 1, 2, \ldots, T$ **do**
1. Resample. For $i = 1, 2, \ldots, N$ sample $\tilde{x}_t^{(i)}$ from the $x_{t-1}^{(j)}$ particles with probabilities $w_{t-1}^{(j)}$. (This step can be omitted for $t = 1$ if there is no y_0 observation.)
2. Propagate. For $i = 1, 2, \ldots, N$ sample $x_t^{(i)}$ from $p(\cdot|\tilde{x}_t^{(i)})$.
3. Weight. For $i = 1, 2, \ldots, N$ compute weight $\tilde{w}_t^{(i)} = p(y_t|x_t^{(i)})$ and normalised weight $w_t^{(i)} = \tilde{w}_t^{(i)}/S_t$ where $S_t = \sum_{j=1}^{N} \tilde{w}_t^{(j)}$.
end for
Output: likelihood estimate $\hat{L} = \prod_{t=1}^{T} \frac{S_t}{N}$ (or, if a y_0 observation is available, take the product from $t = 0$.).

PMMH Tuning

PMMH using BPF likelihood estimates has several tuning choices. This section sets out the approach we use to make these choices in this paper. Our choices are consistent with the theoretical analyses of Sherlock et al. (2015) and Doucet et al. (2015), who derive tuning recommendations under two different sets of simplifying assumptions.

We select the number of particles N for the BPF prior to running Algorithm 1 so that the estimated log-likelihood at a representative parameter value has a standard deviation of roughly 1.5. The parameter value used should have good support under the posterior; we typically use marginal posterior medians from exploratory analyses.

We use a normal random walk proposal distribution: $\theta^* \sim \mathcal{N}(\theta_{i-1}, \Sigma_{\text{RW}})$. We take Σ_{RW} to be an estimate of the posterior variance, again taken from exploratory analyses. Sherlock et al. (2015) and Doucet et al. (2015) provide guidance for scaling the variance matrix by a scalar to improve performance, finding that this was helpful for high dimensional target distributions for instance. We did not find this necessary for our analyses of low dimensional targets, but make use of this approach in Section 4.6 where an 11-dimensional target is considered.

2.3 Ensemble Kalman Filter

Here, we give a brief overview of the ensemble Kalman filter (Evensen, 1994) and refer the reader to Katzfuss et al. (2016) and the references therein for further details.

Consider the task of generating samples according to the filtering density $p(x_t|y_{1:t})$ where $y_{1:t} = (y_1, \ldots, y_t)$. (We omit explicit conditioning on the parameter vector θ throughout this section.) The EnKF generates approximate draws from $p(x_t|y_{1:t})$ via a sequence of forecasting and updating steps. Suppose that a sample $\{x_{t-1}^{(1)}, \ldots, x_{t-1}^{(N)}\}$ (known as the *filtering ensemble*) is available at time $t-1$ from $p(x_{t-1}|y_{1:t-1})$. The *forecast ensemble* $\{\tilde{x}_t^{(1)}, \ldots, \tilde{x}_t^{(N)}\}$ is obtained by drawing $\tilde{x}_t^{(i)} \sim p(\cdot|x_{t-1}^{(i)}), i = 1, \ldots, N$. The forecast density $p(x_t|y_{1:t-1})$ is then approximated by

$$p_{\text{enkf}}(x_t|y_{1:t-1}) = \mathcal{N}(x_t\,;\,\hat{\mu}_{t|t-1}\,,\,\hat{\Sigma}_{t|t-1})$$

where $\mathcal{N}(\cdot;\mu,\Sigma)$ denotes the multivariate Gaussian density with mean μ and variance matrix Σ. The quantities $\hat{\mu}_{t|t-1}$ and $\hat{\Sigma}_{t|t-1}$ are typically taken to be the sample mean and variance computed from the forecast ensemble (some extensions of the EnKF use alternative estimates; see Katzfuss et al., 2016 for some common approaches). Now, given the linear Gaussian form of (1), the joint distribution of X_t and Y_t (given $y_{1:t-1}$) can be obtained approximately as

$$\begin{pmatrix} X_t \\ Y_t \end{pmatrix} \sim N\left\{ \begin{pmatrix} \hat{\mu}_{t|t-1} \\ P\hat{\mu}_{t|t-1} \end{pmatrix}, \begin{pmatrix} \hat{\Sigma}_{t|t-1} & \hat{\Sigma}_{t|t-1}P' \\ P\hat{\Sigma}_{t|t-1} & P\hat{\Sigma}_{t|t-1}P' + S \end{pmatrix} \right\}. \tag{4}$$

Hence, conditioning on $Y_t = y_t$ gives

$$p_{\text{enkf}}(x_t|y_{1:t}) = \mathcal{N}(x_t\,;\,\hat{\mu}_{t|t}\,,\,\hat{\Sigma}_{t|t}) \tag{5}$$

where $\hat{\mu}_{t|t} = \hat{\mu}_{t|t-1} + \hat{K}_t(y_t - P\hat{\mu}_{t|t-1})$, $\hat{\Sigma}_{t|t} = (I_{d_x} - \hat{K}_t P)\hat{\Sigma}_{t|t-1}$ and $\hat{K}_t$ is an estimate of the Kalman gain, that is

$$\hat{K}_t = \hat{\Sigma}_{t|t-1}P'(P\hat{\Sigma}_{t|t-1}P' + S)^{-1}. \tag{6}$$

It is then straightforward to generate samples from (5) to be used as the filtering ensemble at the next time point. However, rather than explicitly calculate the filtering density in (5), the standard implementation of the EnKF (see e.g. Katzfuss et al., 2016) performs a *shifting* step, which is equivalent under the Gaussianity assumption (4) (and a Gaussian prior for x_0). For each particle (known in this context as an *ensemble member*), we compute $x_t^{(i)} = \tilde{x}_t^{(i)} + \hat{K}_t(y_t - \tilde{y}_t^{(i)})$, where $\tilde{y}_t^{(i)} \sim \mathcal{N}(P\tilde{x}_t^{(i)}, S)$ is a pseudo-observation. Note that the shifting step only requires a draw from a d_y variate Gaussian distribution per particle, rather than a draw from a d_x variate Gaussian if (5) is sampled directly. Moreover, the shifting approach does not make the strong assumption that the forecast ensemble is Gaussian distributed. We note that there are other schemes for performing the shifting, but we do not consider these here.

Given a sample $\{x_0^{(1)}, \ldots, x_0^{(N)}\}$ from the state prior, the EnKF recursively alternates between computing the forecast ensemble and shifting each ensemble member, giving approximate draws from the filtering density $p(x_t|y_{1:t})$ for $t = 1, \ldots, T$. We state a version of the EnKF based on the shifting step described above as Algorithm 3.

The EnKF is most easily understood in the context of a linear Gaussian state space model. In this special case, the filtering distribution $p_{\text{enkf}}(x_t|y_{1:t-1})$ converges to the true filtering distribution as the number of ensemble members $N \to \infty$. Essentially, the EnKF converges to the Kalman filter. For finite N and a linear state space model, the EnKF approximates the Kalman filter by replacing the mean and variance of the forecast distribution with their sample equivalents. The resulting dimension reduction (that only requires storing and manipulating d_x-vectors) avoids the potentially expensive calculation and storage of the forecast variance matrix. Moreover, several studies (e.g. Lei et al., 2010; Houtekamer et al., 2014; Katzfuss et al., 2019) have found that the EnKF shifting step works well for non-Gaussian evolution densities. We therefore consider the use of the EnKF likelihood inside a Metropolis-Hastings scheme. We provide a motivation and give details of the proposed approach in the next section.

3 Ensemble MCMC

It is well known that as the variance of the likelihood estimator increases, the acceptance probability of the pseudo-marginal MH scheme rapidly decreases to 0 (Pitt et al., 2012), resulting in slow mixing behaviour of the parameter chains. As discussed in Section 2.2, a value of N (the number of particles) can be chosen to balance mixing performance and computational cost. Nevertheless, in scenarios where the stochasticity inherent in the state process dominates the observation variance, the number of particles required to maintain a reasonable likelihood variance is likely to render BPF-driven PMMH computationally infeasible. Methods that aim to alleviate this problem include the use of an auxiliary particle filter (see e.g. Golightly and Wilkinson, 2015), which requires

careful exploitation of the model structure in order to propagate particles conditional on the observations. By comparison, the algorithm studied here is simple to implement and, for the simplest implementation, does not require the specification of any additional tuning parameters.

Here we outline the *ensemble MCMC* (eMCMC) algorithm proposed in Katzfuss et al. (2019). In essence, this is PMMH using the EnKF as a fast replacement for the BPF to estimate the likelihood $L(\theta)$. In the following subsections we discuss some extensions to improve its efficiency.

First we derive a likelihood estimate based on EnKF calculations. Recall that the (marginal) likelihood can be factorised as

$$L(\theta) = p(y_1|\theta) \prod_{t=2}^{T} p(y_t|y_{1:t-1}, \theta). \tag{7}$$

From (4) it follows that an EnKF approximation of $p(y_t|y_{1:t-1}, \theta)$ is

$$p^N_{\text{enkf}}(y_t|y_{1:t-1}, \theta) = \mathcal{N}(y_t\,;\, P\hat{\mu}_{t|t-1}\,,\, P\hat{\Sigma}_{t|t-1}P' + S)$$

which can easily be computed for each $t = 1, \ldots, T$, with the notational convention that $p(y_1|\theta) = p(y_1|y_{1:0}, \theta)$. The need for the explicit dependence on N for this likelihood estimate will become clearer later in this section. The overall approximation to the likelihood is given by

$$\hat{L}^N_{\text{enkf}}(\theta) = \prod_{t=1}^{T} p^N_{\text{enkf}}(y_t|y_{1:t-1}, \theta). \tag{8}$$

The EnKF including likelihood estimation is given by Algorithm 3. The eMCMC scheme is then implemented by running Algorithm 1 with $\hat{L}$ replaced by $\hat{L}^N_{\text{enkf}}$. One issue in implementing eMCMC is how to perform tuning. Due to the absence of specialised theory, we use the same tuning guidance as for PMMH with BPF likelihood estimates, described above in Section 2.2.

It is worth emphasising that, unlike pMCMC, the eMCMC posterior

$$p^N_{\text{enkf}}(\theta|y) \propto \hat{L}^N_{\text{enkf}}(\theta)p(\theta),$$

does not in general equal the posterior $\pi(\theta|y)$ exactly. The reason is that, unlike the BPF, the EnKF gives a biased estimator of $L(\theta)$, precluding its use for exact approximate inference. Nevertheless, as noted by Stroud et al. (2010), Stroud et al. (2018) and Katzfuss et al. (2019) among others, the variance of the likelihood estimator under the EnKF can be relatively small, suggesting that use of EnKF inside a Metropolis-Hastings scheme is likely to be of practical use, particularly in scenarios when the BPF is computationally prohibitive.

In fact, even when the forecast ensemble is exactly Gaussian distributed for all t, the EnKF posterior still does not target the exact posterior, since $\mathcal{N}(y_t; P\hat{\mu}_{t|t-1}, P\hat{\Sigma}_{t|t-1}P' + S)$ is a biased estimate of the *idealised* normal density should we be able to take $N \to \infty$.

Algorithm 3 Ensemble Kalman filter.

Input: number of ensemble members N

Initialise. For $i = 1, 2, \ldots, N$ sample $x_0^{(i)}$ from the initial state distribution.

for $t = 1, 2, \ldots, T$ **do**

1. Forecast ensemble. For $i = 1, 2, \ldots, N$ sample $\tilde{x}_t^{(i)} \sim p(\cdot|x_{t-1}^{(i)})$.

2. Likelihood calculation. Compute estimates of the forecast mean and variance: $\hat{\mu}_{t|t-1}$ and $\hat{\Sigma}_{t|t-1}$. Compute likelihood component $\hat{L}^N_{\text{enkf},t} = \mathcal{N}(y_t\,;\, P\hat{\mu}_{t|t-1}\,,\, P\hat{\Sigma}_{t|t-1}P' + S)$.

3. Shift ensemble. Compute the approximate Kalman gain, $\hat{K}_t$, given by (6). For $i = 1, 2, \ldots, N$, set $x_t^{(i)} = \tilde{x}_t^{(i)} + \hat{K}_t(y_t - \tilde{y}_t^{(i)})$, where $\tilde{y}_t^{(i)} \sim \mathcal{N}(P\tilde{x}_t^{(i)}, S)$ is a pseudo-observation.

end for

Output: likelihood estimate $\hat{L}^N_{\text{enkf}} = \prod_{t=1}^T \hat{L}^N_{\text{enkf},t}$.

Thus, for finite N, the eMCMC target is not the idealised eMCMC target, $p^{\infty}_{\text{enkf}}(\theta|y)$. However, we find empirically that our method appears to be weakly dependent on N. Given this, we suggest to choose N to maximise the computational efficiency by borrowing similar advice from the pseudo-marginal literature (as described in Section 2.2). Interestingly, there is an exactly unbiased estimator of a normal density given a sample from it, and we exploit this in Section 3.3. We discuss the unbiased version and other extensions below.

3.1 Randomised Quasi-Monte Carlo

For this subsection, all quantities are conditioned on θ so we drop it for notational convenience. At iteration t of the EnKF we estimate $\mu_{t|t-1}$ and $\Sigma_{t|t-1}$, so that we can approximate the conditional likelihood $p(y_t|y_{1:t-1})$ with a Gaussian density. These moments are estimated via firstly performing the shifting step at $t-1$ and conditional on the result simulating from the forward evolution density. This is effectively an approximate sample from the joint distribution $p(x_t, x_{t-1}|y_{1:t-1}) = p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})$. Then $\mu_{t|t-1}$ and $\Sigma_{t|t-1}$ are estimated from the N ensemble members.

Often it is possible to write the simulation from a standard statistical distribution as a function of a uniform random number. For example, to simulate a random draw y from a normal distribution, $\mathcal{N}(\mu, \sigma^2)$, we can compute the following, $y = \mu + \sigma \cdot \Phi^{-1}(u)$ where $u \sim \mathcal{U}(0, 1)$ and $\Phi^{-1}(u)$ is the quantile function of the standard normal density. Assume that we can write the evolution density as a function of m uniform random variates. Then, we require $d_y + m$ uniform random numbers to approximately simulate from $x_t, x_{t-1}|y_{1:t-1}$ (d_y for the shifting step and m for simulating the evolution density). Given N particles, we use $N \times (d_y + m)$ uniform random numbers for estimating $\mu_{t|t-1}$ and $\Sigma_{t|t-1}$. The naive approach is to draw these via pseudo-random numbers. However, significant variance reduction could be achieved by simulating from the $(d_y + m)$-dimensional object N times using *randomised quasi-Monte Carlo* (RQMC, e.g. L'Ecuyer and Lemieux, 2005). QMC is well known to generate a sequence of numbers that have

superior space filling properties in the unit hypercube compared to pseudo-random numbers. The randomised component ensures that expectations can be estimated unbiasedly. Random numbers from the joint distribution of interest, $x_t, x_{t-1}|y_{1:t-1}$, can be achieved via transforming the RQMC numbers as recently discussed. We use this approach to bring down the variance of the estimators of $\mu_{t|t-1}$ and $\Sigma_{t|t-1}$, which hopefully reduces the variance of the estimator for $p(y_t|y_{1:t-1})$. For generating the RQMC numbers in this paper, we use the scrambled Sobol's net, i.e. the scrambled (t, m, s)-net in base $b = 2$.

RQMC has recently received increasing attention in the statistics community. Tran et al. (2017) document a faster convergence in their variational Bayes updating procedure when the noisy gradient is computed using RQMC, Drovandi and Tran (2018) use it to reduce the variance of expected utility estimation within Bayesian optimal design and Gerber and Chopin (2015) show the efficiency of RQMC in particle filtering. However their application to particle filtering requires considerable ingenuity (using a Hilbert curve method to perform resampling). It is interesting to note the ease with which RQMC can be exploited in the EnKF in comparison.

3.2 Correlated eMCMC

As mentioned above, the EnKF requires generating random numbers for the shifting step and simulating the evolution density. The former can be generated by standard normal random variates and the Cholesky factorisation of the covariance matrix. We assume in this section that the evolution density can be simulated either directly or indirectly via a suitable transformation with standard normal random numbers. Denote the collection of these random numbers required in the EnKF as u.

Deligiannidis et al. (2018) and Dahlin et al. (2015) develop the *correlated pseudo-marginal MCMC* method where they consider the joint target density $p(\theta, u|y)$ where u are random numbers required to estimate the likelihood unbiasedly, $p(y|\theta, u)$. It is easy to show that the θ-marginal of the joint distribution is the posterior of interest, $p(\theta|y)$. Assume that u are independent standard normal random variates. The idea of the correlated pseudo-marginal method is to induce correlation in successive likelihood estimates in MCMC by correlating the u random numbers. This can have the effect of mitigating "sticky" behaviour often seen in pseudo-marginal chains since, in the correlated scheme, if the likelihood is overestimated at the current iteration, it is also likely to be overestimated at the next. The joint proposal distribution of the correlated pseudo-marginal method is given by $q(\theta^*, u^*|\theta, u) = q(\theta^*|\theta)\mathcal{N}(u^*; \sqrt{1-\sigma_u^2}u, \sigma_u^2\mathcal{I})$, where $\mathcal{I}$ is the identity matrix. The proposal for u is the *Crank-Nicolson proposal* and it is invariant with respect to the marginal distribution of u. σ_u^2 is an additional tuning parameter that is typically set to be small so that u^* is highly correlated with u.

Here we consider applying this correlated pseudo-marginal approach to our eMCMC method, with the motivation that a smaller ensemble size N can be used, reducing computational cost. Note that BPF driven pMCMC requires additional modification to accommodate this approach, as it did for RQMC. Essentially, the resampling step has the effect of breaking down correlation between successive likelihood estimates. To

alleviate this problem, the particles can be sorted before propagation e.g. using a Hilbert sorting procedure (Deligiannidis et al., 2018) or simple Euclidean sorting (Choppala et al., 2016). The random numbers used in the resampling step itself should also be updated using the Crank-Nicolson proposal. Since the eMCMC scheme does not use resampling, incorporating correlation is straightforward.

3.3 Unbiased Ensemble Kalman Filter Likelihood

As mentioned earlier, even if the sample from the forecast distribution was exactly Gaussian for some t, the corresponding EnKF likelihood estimate for y_t would not be unbiased. In general, for some data y and a sample of size N from a Gaussian distribution, $x = x_1, \ldots, x_N \sim \mathcal{N}(\mu, \Sigma)$, the density estimator $\mathcal{N}(y; \mu_N, \Sigma_N)$ is not an unbiased estimator of $\mathcal{N}(y; \mu, \Sigma)$ where μ_N and Σ_N are the sample mean and covariance computed from the sample x. Given the bias present in the EnKF likelihood estimate, even when the Gaussian assumption is correct, the EnKF posterior, unlike standard pseudo-marginal, theoretically depends on N.

Even though we demonstrate empirically in Section 4 that the eMCMC posterior seems to be only weakly dependent on N, we present a new approach now that will likely be less sensitive to N. Interestingly, there does exist an unbiased estimator of a Gaussian density given only an iid sample from the same Gaussian density. Using the notation of Ghurye and Olkin (1969), let

$$c(k, v) = \frac{2^{-kv/2}\pi^{-k(k-1)/4}}{\prod_{i=1}^{k} \Gamma\left(\frac{1}{2}(v - i + 1)\right)},$$

and for a square matrix A write $\psi(A) = |A|$ if $A > 0$ and $\psi(A) = 0$ otherwise, where $|A|$ is the determinant of A and $A > 0$ means that A is positive definite. The result of Ghurye and Olkin (1969) shows that an exactly unbiased estimator of $\mathcal{N}(y; \mu, \Sigma)$ is (in the case where y is Gaussian and $N > d + 3$ where d is the dimension of y)

$$\widehat{\mathcal{N}}(y; \mu, \Sigma) = (2\pi)^{-d/2} \frac{c(d, N-2)}{c(d, N-1)(1-1/N)^{d/2}} |M_N|^{-(N-d-2)/2} \psi\left(M_N - (y - \mu_N)(y - \mu_N)^\top/(1 - 1/N)\right)^{(N-d-3)/2},$$

where $M_N = (N-1)\Sigma_N$. We propose to replace the standard Gaussian density estimator in the EnKF likelihood estimator with this alternative estimator. Note that this estimator has also been used in Price et al. (2018) for approximating intractable likelihoods in simulation-based likelihood-free estimation problems.

We refer to the method when we use the unbiased Gaussian density estimator in the EnKF likelihood estimator as ueMCMC. We stress that this approach still does not target the true posterior, but at least it will not depend on the number of ensemble members N when the Gaussian assumption is correct, i.e. the target is exactly the idealised approximation, $p^{\infty}_{\text{enkf}}(\theta|y)$. Even though the forecast density is unlikely to be exactly Gaussian in practice, we do expect ueMCMC to be less sensitive to N compared

with eMCMC. We note that this method might be particularly useful when combined with the correlated approach in Section 3.2, since it might be sufficient to use a very small N to achieve reasonable computational efficiency, but the small N may produce bias in the eMCMC posterior compared to the idealised eMCMC posterior.

3.4 Early Rejection

Prangle et al. (2018) apply pMCMC in the setting of approximate Bayesian computation (ABC). They outline a method for rejecting proposed values of θ that have a small estimated likelihood without running the whole particle filter. In Everitt and Sibly (2019) it is shown that this approach can be extended to pMCMC when using the BPF. A similar approach may be used in eMCMC. Suppose that the likelihood estimate from the EnKF is implemented sequentially, as the EnKF is running. Recall from (8) that the EnKF likelihood estimate is a product, $\hat{L}^N_{\text{enkf}}(\theta) = \prod_{t=1}^T \alpha_t$. Here $\alpha_t = \mathcal{N}\left(y_t; P\hat{\mu}_{t|t-1}, P\Sigma_{t|t-1}P' + S\right)$ is calculated in iteration t of the EnKF. We are guaranteed that an upper bound on α_t is given by $B(\theta^*) := \mathcal{N}(0; 0, S(\theta^*))$, where 0 represents a vector of zeros of appropriate dimension. (See Algorithm 4 for more details on notation.) Thus $\alpha_t/B \leq 1$. This fact ensures that $\hat{r}^{(\tau)}_{\text{enkf}} := \prod_{t=1}^{\tau} \alpha_t/B$ is an upper bound on $\hat{L}^N_{\text{enkf}}(\theta)/B^T$ which can be calculated at iteration τ of the EnKF.

We use this property to propose an "early rejection" algorithm. The idea is that during an EnKF run, as soon as $\hat{r}_{\text{enkf}}$ drops below a certain threshold, we are sure that the MCMC proposal θ^* will not be accepted. Hence we can save time by immediately terminating the EnKF run. No time is saved for an accepted proposal; only for those that are rejected. Therefore the computational savings are largest in cases where the acceptance probability is low, as we would find, for example: in a higher dimensional parameter space when the proposal variance is not reduced accordingly; or when the likelihood estimate has a high variance. Algorithm 4 describes a single iteration of the resultant MCMC algorithm, which involves reorganising the order of calculation of the acceptance probability and likelihood estimate from our standard eMCMC algorithm. This early rejection approach is employed in Section 4.6, where a computationally expensive model is studied.

4 Results

Here we demonstrate the potential of our method on several examples with different kinds of complexity. We select the number of particles N and MCMC proposal variance as described in Section 2.2. Given that the different methods have different target distributions, we tune the random walk covariance matrix individually for each method.

In terms of accuracy we compare the approximate eMCMC and the 'exact' pMCMC approach visually. We note that in many applications it might not be critical to obtain samples from the exact posterior given the potential for model misspecification and/or high accuracy not being important for the analysis aims. When we deem the eMCMC approximation to be reasonable enough, we compare the statistical efficiency of the

Algorithm 4 An iteration of early-rejection eMCMC.

Input: θ, the current value of the parameter and $\hat{L}^N_{\text{enkf}}(\theta)$, the estimate of the likelihood for this parameter.
Simulate $\theta^* \sim q(\cdot \mid \theta)$, and let $S(\theta^*)$ be the measurement noise matrix for this proposed parameter.
Let $B(\theta^*) = \mathcal{N}(0; 0, S(\theta^*))$.
Simulate $u \sim \mathcal{U}(0,1)$.
Initial EnKF step: for $i = 1 \dots N$, simulate $x_0^{(i)}$ from the initial state distribution.
Initialise estimate $\hat{r}_{\text{enkf}} = 1$, then perform first **early rejection** step:
if $\hat{r}_{\text{enkf}} < u \frac{p(\theta)\hat{L}^N_{\text{enkf}}(\theta)}{p(\theta^*)} \frac{q(\theta^*|\theta)}{q(\theta|\theta^*)} \frac{1}{B^T(\theta^*)}$ **then**
  reject θ^* and break.
end if
for $t = 1 \dots T$ **do**
  1. Forecast ensemble. For $i = 1, \dots, N$ sample $\tilde{x}_t^{(i)} \sim p(\cdot|x_{t-1}^{(i)})$.
  2. Likelihood update. Compute estimates of the forecast mean and variance: $\hat{\mu}_{t|t-1}$ and $\hat{\Sigma}_{t|t-1}$. Set $\hat{r}_{\text{enkf}} := \hat{r}_{\text{enkf}} \times \mathcal{N}\left(y_t\,;\, P\hat{\mu}_{t|t-1}\,,\, P\hat{\Sigma}_{t|t-1}P' + S(\theta^*)\right)/B(\theta^*)$.
  3. Early rejection.
  if $\hat{r}_{\text{enkf}} < u \frac{p(\theta)\hat{L}^N_{\text{enkf}}(\theta)}{p(\theta^*)} \frac{q(\theta^*|\theta)}{q(\theta|\theta^*)} \frac{1}{B^T(\theta^*)}$ **then**
    reject θ^* and break.
  end if
  4. Shift ensemble. Compute the approximate Kalman gain, $\hat{K}_t$, given by (6). For $i = 1, \dots, N$, set $x_t^{(i)} = \tilde{x}_t^{(i)} + \hat{K}_t(y_t - \tilde{y}_t^{(i)})$, where $\tilde{y}_t^{(i)} \sim \mathcal{N}\left(P\tilde{x}_t^{(i)}, S(\theta^*)\right)$ is a pseudo-observation.
end for
Accept θ^* and let $\hat{L}^N_{\text{enkf}}(\theta^*) = \hat{r}_{\text{enkf}} B^T(\theta^*)$.

two methods using effective sample size (ESS). ESS is the number of independent and identically distributed samples from the target that would produce an estimate with the same variance as the autocorrelated MCMC output. We generally use the multivariate ESS estimate of Vats et al. (2019) which takes posterior dependence into account (in one case where this performs poorly we instead use the univariate ESS of Plummer et al. 2006). We also consider overall efficiency – ESS per second – which incorporates both statistical efficiency and computing time.

In Section 5 we provide suggestions on how 'exact' posterior sampling can be achieved whilst still using the EnKF. However, the statistical efficiency gains of these approaches will be reduced compared to eMCMC.

Unless otherwise stated, we use the standard normal density estimator in eMCMC as opposed to the unbiased version in Section 3.3. We find empirically they produce similar results and there is a small amount of overhead and implementation effort with the unbiased normal density estimator. However, practitioners may decide to use the unbiased version given the theoretical benefit it brings.

4.1 Population Ecology Example

Model and Inference Task

Peters et al. (2010) consider a set of competing nonlinear state space population models in ecology and apply them to several datasets. Denoting the observation at time t as y_t and the corresponding hidden state as n_t, the four models we consider are defined below:

1. Ricker model: $\log n_{t+1} = \log n_t + \beta_0 + \beta_1 n_t + \epsilon_t$.
2. Theta-logistic model: $\log n_{t+1} = \log n_t + \beta_0 + \beta_2 n_t^{\beta_3} + \epsilon_t$.
3. Mate-limited model: $\log n_{t+1} = 2\log n_t + \beta_0 + \beta_1 n_t - \log(\beta_4 + n_t) + \epsilon_t$.
4. Flexible-Allee model: $\log n_{t+1} = \log n_t + \beta_0 + \beta_1 n_t + \beta_5 n_t^2 + \epsilon_t$.

Here $\epsilon_t \sim \mathcal{N}(0, \sigma_w^2)$. The observation process is assumed to be Gaussian, $y_t | n_t \sim \mathcal{N}(\log n_t, \sigma_e^2)$. See Peters et al. (2010) for a justification and some qualitative analyses of these models. The parameters are assumed independent *a priori* and have the following specifications: $\beta_0, \beta_1, \beta_3, \beta_5 \sim \mathcal{N}(0, 1)$, $\beta_4, \sigma_w, \sigma_e \sim \mathcal{E}xp(1)$ and $\log n_0$ has an improper uniform prior over the real line.

Here we re-analyse the nutria dataset, a time series of female nutria abundance in East Anglia at monthly intervals, considered in Peters et al. (2010) and some references therein. The data is shown in Figure 1.

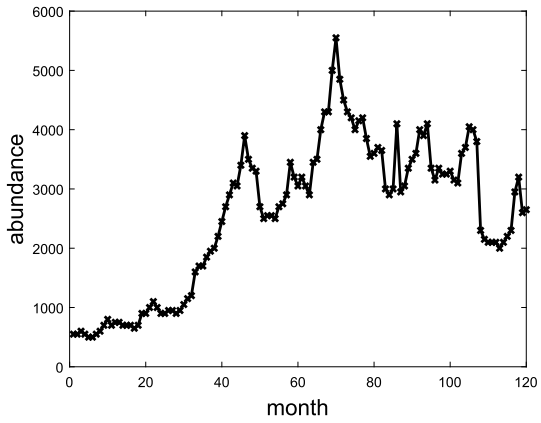

Figure 1: The nutria dataset. The observations are shown as crosses and the solid line is a linear interpolation between observations.

It is useful to consider this class of models since various quantities of the model are tractable to compute, permitting more advanced particle filters to compare with. For example, the fully adapted auxiliary particle filter (see e.g. Pitt et al., 2010) pre-weights

the particles by (using generic notation) $p(y_t|x_{t-1},\theta) = \int_{x_t} p(y_t|x_t,\theta)p(x_t|x_{t-1},\theta)dx_t$ and propagates resampled particles by $p(x_t|y_t,x_{t-1},\theta)$. This particle filter results in particle weights that remain uniform throughout the algorithm. It is the optimal one-step look-ahead particle filter. The fully adapted particle filter works extremely well in this application. It requires only $N = 5$ particles for superior performance compared to the bootstrap filter using $N = 50{,}000$ particles.

Although the quantities required for the fully adapted auxiliary particle filter can be obtained for this example, they cannot for a wide class of complex models. For many state space models, the transition density $p(x_t|x_{t-1},\theta)$ cannot be evaluated. In these cases, it is convenient to propagate particles according to the transition law so that intractable terms cancel in the importance weights. Thus we consider another particle filter for comparison, referred to here as the partially adapted particle filter, where $p(y_t|x_{t-1},\theta)$ is used to pre-weight particles and the transition density is used to propagate the resampled particles. However, in many models evaluating $p(y_t|x_{t-1},\theta)$ is also intractable. Often it is approximated with $p(y_t|\mu_{t|t-1},\theta)$ where $\mu_{t|t-1}$ is given by some location measure of the $x_t|x_{t-1},\theta$ (e.g. mean). We find that the partially adapted approach produces similar results to the standard bootstrap filter.

For the results presented below, we assume that it is only feasible to simulate the transition density, and thus compare with pMCMC using the bootstrap filter, given that the partially adapted filter produces similar results.

Inference

It is likely that all the considered models are misspecified but we would like a robust method for fitting them in order to compare the models and investigate possibilities for extending the models. We find that all models have particular difficulty in capturing the sudden drop in abundance between months 107 and 108. Further, there appears to be only small observation error. The consequence for the bootstrap filter is a very small ESS and high variance estimates of the likelihood unless a very large number of particles is used.

For eMCMC, we only require $N = 250$ (Ricker, Flexible-Allee, theta-logistic) and $N = 200$ (mate-limited) particles. In contrast, we use $N = 50{,}000$ for pMCMC. For some of the models, the standard deviation of the estimated log likelihood is still larger than 1.5 even with this large number of particles. However, we find that when these occur the distribution of the log-likelihood estimator with the BPF has a skew-left distribution, which is less problematic for pMCMC getting stuck at overestimated log likelihood values. We find that the pMCMC acceptance rates remain reasonable with $N = 50{,}000$ particles.

The MCMC acceptance rates for the four models are 15%, 4%, 11% and 10% (eMCMC), and 8%, 3%, 6% and 5% (pMCMC), respectively. The acceptance rates are lower for the theta-logistic model as the posterior distribution is far more irregular compared to the other three models (see Figure 3).

Based on Figures 2, 4 and 5, eMCMC obtains estimated univariate posterior distributions that are remarkably similar to pMCMC. There is more difference for the

theta-logistic model (Figure 3) but they remain broadly similar. Further, the Monte Carlo error is greater for this model, potentially exaggerating the differences.

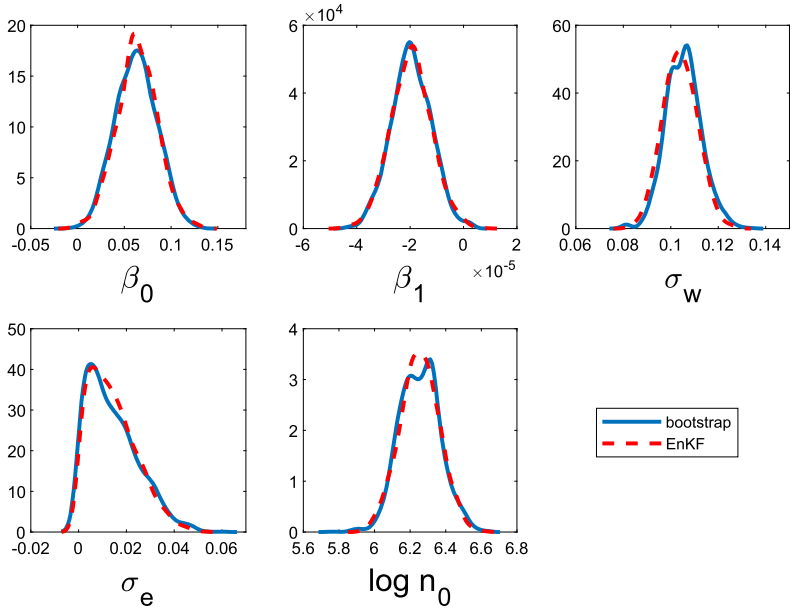

Figure 2: Estimated univariate posterior distributions for the parameters of the Ricker model based on pMCMC (blue solid) and eMCMC (red dash).

The two methods are compared in terms of computational efficiency on the four models in Table 1. It is evident that the eMCMC approach produces a two order of magnitude improvement in terms of computational efficiency and still produces reasonable approximations of the posterior.

We test eMCMC for a range of N values in 100–1000 and find the univariate posteriors to show little sensitivity to N (results not shown). For $N = 100$, the MCMC acceptance rate drops substantially and reducing N further is likely to significantly reduce the statistical efficiency of MCMC due to the high-variance likelihood estimates. Therefore, it is difficult to test the sensitivity of the results to small N.

However, using the correlated extension (with $\sigma_u = 0.1$) allows us to use small N and maintain statistically efficient results. Similar MCMC acceptance rates as eMCMC with 250 particles can be achieved using only $N = 25$ particles. The efficiency results can be seen in Table 1. It is evident that the correlation further improves the computational efficiency in this example. The resulting approximate marginal posteriors compared to eMCMC with $N = 1000$ are shown for the four models in the figures of Appendix A of the Supplementary Materials (Drovandi et al., 2020). It is clear that similar approximate posteriors are obtained even with vastly different N values. However, for all models there is a noticeable bias in the approximate posterior for σ_w. We also run the unbiased version

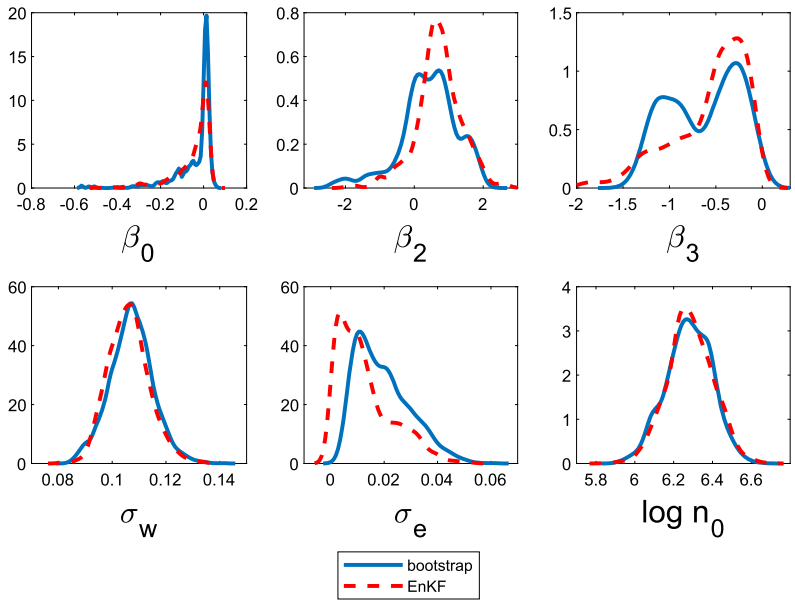

Figure 3: Estimated univariate posterior distributions for the parameters of the theta-logistic model based on pMCMC (blue solid) and eMCMC (red dash).

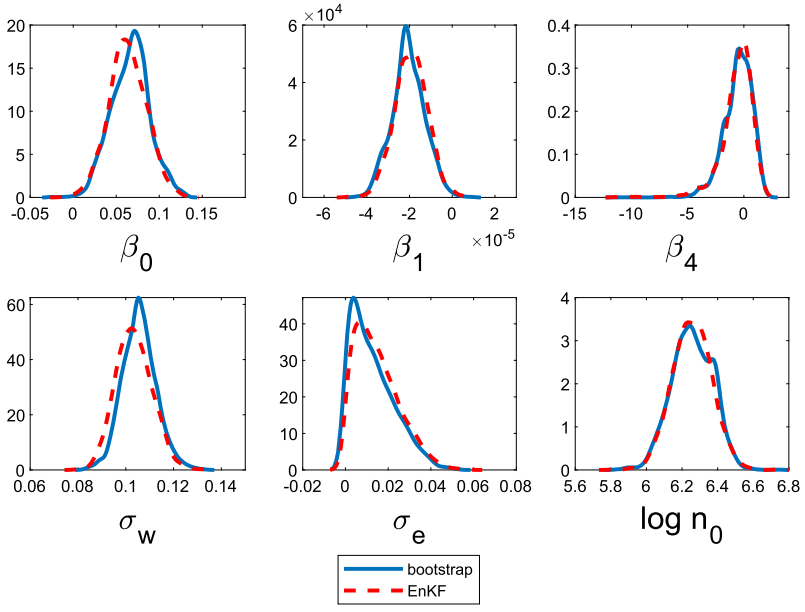

Figure 4: Estimated univariate posterior distributions for the parameters of the mate-limited model based on pMCMC (blue solid) and eMCMC (red dash).

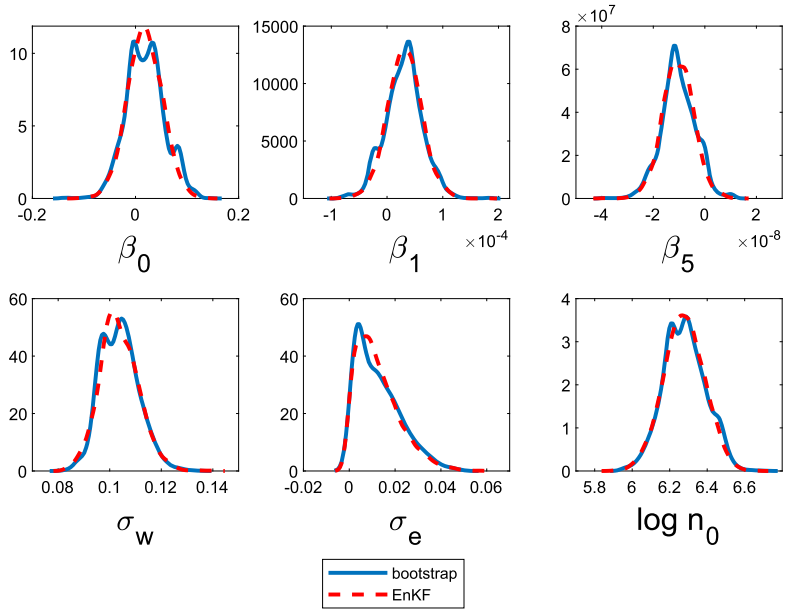

Figure 5: Estimated univariate posterior distributions for the parameters of the flexible-allee model based on pMCMC (blue solid) and eMCMC (red dash).

Model	filter	N	ESS	Time (h)	ESS/Time
Ricker	BPF	50000	920	36.8	25
Ricker	EnKF	250	2400	0.14	17000
Ricker	EnKF + correlation	25	2100	0.07	30000
theta-logistic	BPF	50000	410	40.8	10
theta-logistic	EnKF	250	500	0.48	1040
theta-logistic	EnKF + correlation	25	540	0.06	9000
mate-limited	BPF	50000	770	37.6	20
mated-limited	EnKF	200	1460	0.35	4200
mated-limited	EnKF + correlation	25	1800	0.07	25700
flexible-allee	BPF	50000	750	37.0	20
flexible-allee	EnKF	250	1750	0.26	6700
flexible-allee	EnKF + correlation	25	1600	0.08	20000

Table 1: Efficiency comparisons for the four non-linear population ecology models.

of Section 3.3 with the correlated extension, again for $N = 25$. The same figures in the appendix demonstrate that ueMCMC is able to reduce the bias in the approximate posterior for σ_w. The largest difference between the results for $N = 1000$ and $N = 25$ occurs for the theta-logistic model. For $N = 25$, the unbiased version seems to offer some correction for θ_w and θ_e but produces similar results to the biased version for the other parameters. We find that with the unbiased version the ESS remains similar, but the

overall efficiency is slightly reduced. The reduction in computational efficiency mainly comes here from the extra time to compute the unbiased multivariate normal density estimator (the ESS is roughly the same). We note that for applications where simulating the transition density consumes the majority of the computation, the additional time associated with computing the unbiased estimator will be significantly less noticeable.

Finally, we investigate improvements that can be obtained in this example when using the RQMC extension. Here we use $N = 50$ ensemble members for each model. It is evident that the RQMC extension produces similar marginal posteriors compared to eMCMC with $N = 1000$ particles (see Appendix B of the Supplementary Materials). The ESS values for the four models are roughly 2500, 450, 1700, and 2100, which are competitive with standard eMCMC using a significantly larger, $N = 200$–250, number of particles (see Table 1). However, the ESS/Time scores for the four models are only roughly 900, 130, 500 and 620 with the RQMC extension. Given that simulation of the transition density is trivial in this example, the cost associated with generating the RQMC samples is significant and consequently the ESS/Time score with the RQMC extension is substantially reduced. However, in complex examples where simulating the transition density is expensive, the cost associated with RQMC will be far less noticeable.

4.2 Lorenz 63 Example

Model and Inference Task

The Lorenz 63 dynamical system (Lorenz, 1963) is a classic low dimensional example of chaotic behaviour. An Itô stochastic differential equation (SDE) version from Vrettas et al. (2015) is

$$\begin{aligned} dX_t &= \alpha(X_t, \theta)dt + \Sigma^{1/2}dW_t, \\ \alpha(X_t, \theta) &= \begin{pmatrix} \theta_1(X_{2,t} - X_{1,t}) \\ \theta_2 X_{1,t} - X_{2,t} - X_{1,t}X_{3,t} \\ X_{1,t}X_{2,t} - \theta_3 X_{3,t} \end{pmatrix}, \\ \Sigma &= \begin{pmatrix} \sigma_1^2 & 0 & 0 \\ 0 & \sigma_2^2 & 0 \\ 0 & 0 & \sigma_3^2 \end{pmatrix}. \end{aligned} \tag{9}$$

Here X_t is a vector of the random variables $X_{1,t}, X_{2,t}, X_{3,t}$, and W_t is a vector of standard uncorrelated Brownian motion processes of the same length as X_t. Note that $\Sigma^{1/2}$ is interpreted as a matrix square root. We assume each $X_{i,t}$ at a grid of prespecified t values has a corresponding observation $Y_{i,t} \sim \mathcal{N}(X_{i,t}, \sigma^2_{\text{obs}})$, and that these are independent.

Exact simulation of SDEs is extremely challenging, so it is common to work with an Euler-Maruyama discretisation (see e.g. Wilkinson, 2018). For the Lorenz model above this gives,

$$x_{i+1} = x_i + \alpha(x_i, \theta)\Delta t + \Sigma^{1/2}\sqrt{\Delta t}z_{i+1}, \tag{10}$$

where each z_{i+1} is an independent $\mathcal{N}(0, I_3)$ realisation. Then x_i is an approximation to X_t for $t = i\Delta t$.

Following Vrettas et al. (2015) we simulate data from this discretised model under $\theta = (10, 28, 8/3)$, $\sigma_i^2 = 10$ for $i = 1, 2, 3$, $\sigma_{\text{obs}}^2 = 2$ and $\Delta t = 0.01$. The initial conditions are specified by x_0, which we take to be a vector of zeros. We make observations at $i = 20, 40, \ldots, 600$, corresponding to $t = 0.2, 0.4, \ldots, 6$. Figure 6 shows our data.

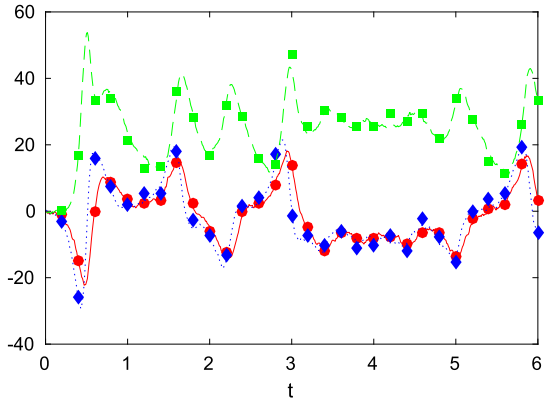

Figure 6: Simulated Lorenz 63 data. The lines show simulated x_i values from the discretised SDE, and the points noisy y_i observations. Component $x_{1,i}$ is represented by red circles, $x_{2,i}$ by blue diamonds and $x_{3,i}$ by green squares.

Log Likelihoods

First we compare log likelihood estimates produced by the EnKF and BPF. We run each method 5 times for $\theta_1 = 1, 2, \ldots, 20$. The other parameters are held constant at their true values. We use 100 particles for both filtering methods. The average run-times were roughly half as long for EnKF – 0.019 s – compared to BPF 0.046 s. Appendix C of the Supplementary Materials shows the results. For any θ_1 value, the log likelihood estimates are more variable under BPF than EnKF. Variability becomes particularly large under BPF when θ_1 is far from its true value. Such high variance is problematic in PMMH, as it is likely to cause chains to become stuck. The figure suggests that the EnKF and BPF produce similar expected likelihood estimates when θ_1 is close to its true value. It is hard to draw any conclusions for other θ_1 values, as the BPF expected likelihood will be strongly driven by the upper tail of its log-likelihood estimates, and this would take a very large number of simulations to estimate well.

Inference

Here we assume σ_{obs} is known, and attempt to infer θ_i and σ_i for $i = 1, 2, 3$. We assume these parameters have independent exponential prior distributions with rate 0.1. We

ran the EnKF and BPF at the true parameter 30 times for each of various choices of N and calculated empirical log-likelihood variances. Based on these values we select $N = 500$ for eMCMC and $N = 2500$ for pMCMC.

We ran our algorithms targeting the log transformed parameters. Both pMCMC and eMCMC achieve acceptance rates in the range 10% to 20% indicating reasonable mixing. Trace plots also suggested good mixing, with no evidence of chains becoming stuck in the same state for a large number of iterations. The ESS values for the MCMC outputs were 390 (eMCMC) and 197 (pMCMC). Run times were 689 s and 10,992 s for eMCMC and pMCMC respectively. Interestingly, the pMCMC run time is roughly 15 times that of eMCMC despite using only 5 times as many particles.

Figure 7 shows the resulting marginal posterior estimates. The eMCMC posterior approximation is similar to the gold standard pMCMC results, but there are some noticeable differences for some parameters e.g. the θ_1 and θ_3 posterior marginals are shifted downwards. Posterior correlations were small for both MCMC methods (all below 0.35 in magnitude).

To validate our ensemble size, we ran eMCMC again using $N = 25, 50, \ldots, 600$, with other tuning choices unchanged. ESS per second was maximised by $N = 125$ and was 62% larger than using $N = 500$ as above. This shows that our tuning diagnostic was reasonably accurate. (We found multivariate ESS behaved poorly for MCMC output with low acceptance rates e.g. when $N = 25$. So here we used the univariate ESS of Plummer et al. 2006 instead, averaged over parameters.)

We also ran RQMC and correlated variants of eMCMC (using $\sigma_u = 0.1$ for the latter). For RQMC, initial tuning based on variance of the log likelihood selected $N = 500$, as for eMCMC. For correlated eMCMC we used a reduced number of particles, $N = 100$. Posterior marginals are shown in Figure 7 and are extremely similar to eMCMC results. RQMC eMCMC produced a similar acceptance rate and ESS value (305) to eMCMC, but the cost of QMC sampling increased the run time to 3,073 s (roughly a 5 times increase). Correlated eMCMC increased the acceptance rate (to 26%) and ESS (to 417) while also reducing the run time (to 195 s).

Finally, we implemented the particle EnKF (pEnKF) method of Katzfuss et al. (2019) (see their Algorithm 4). This algorithm avoids the need for MCMC. Instead, it evolves a population of static parameters (particles) over time, where each particle has an associated ensemble for the hidden state. The approach uses the EnKF approximation of the likelihood to re-weight the particles. The ensemble of latent states is propagated by the transition density and the particles are propagated via resampling and a jittering step. Our implementation used 10,000 particles, each with 100 ensemble members, which we found gave reasonable performance. However a difficulty of this algorithm is that it is unclear how to make these tuning choices optimally. Our run time was 2,529 s, slower than eMCMC. Figure 7 shows that the resulting posterior marginal approximations are significantly less accurate than eMCMC.

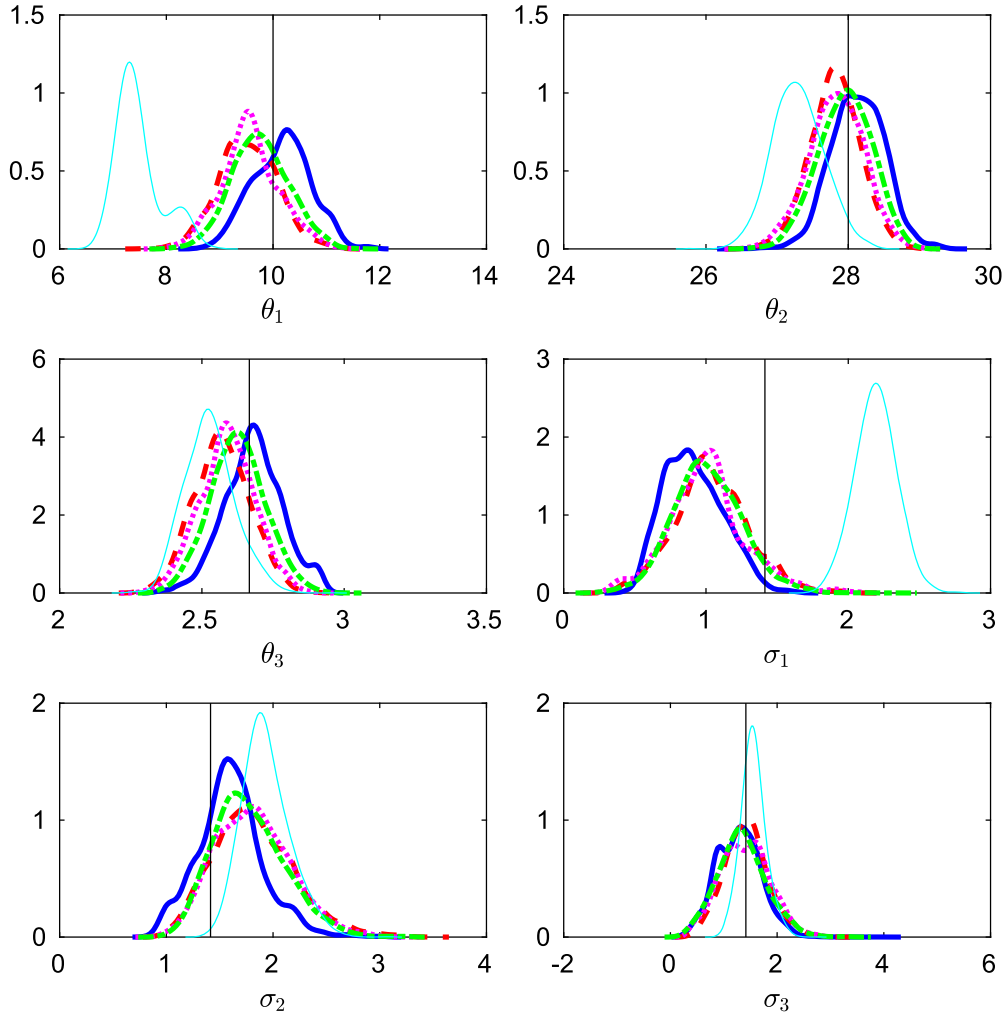

Figure 7: Estimated Lorenz 63 marginal parameter posteriors using pMCMC (blue solid), eMCMC (red dashed), RQMC eMCMC (magenta dotted), correlated eMCMC (green dot-dash), and pEnKF (cyan thin solid). The lines are kernel density estimates based on Monte Carlo samples. True parameter values are shown as black vertical lines.

4.3 Lorenz 96 Example

The Lorenz 96 dynamical system (Lorenz, 1996) is often used to test the performance of data assimilation methods in high dimensions, and we use it for this purpose here to test the eMCMC algorithm. We introduce an SDE version. This again follows the general SDE (9), but now the ith entry of $\alpha(X_t, \theta)$ is defined as

$$\alpha_i(X_t, \theta) = \theta_1(X_{i+1,t} - X_{i-2,t})X_{i-1,t} - \theta_2 X_{i,t} + \theta_3,$$

for $i = 1, 2, \ldots, d_x$, and $\Sigma = \sigma^2 I$. Here X_t is a vector of the d_x random variables $X_{1,t}, X_{2,t}, \ldots, X_{d_x,t}$. Addition and subtraction of indices above are interpreted modulo d_x so that, for example, when $i = 1$ then $X_{i-1,t}$ is $X_{d_x,t}$. We work with the discretised SDE (10). We simulated a dataset with $d_x = 50$, $\theta = (1, 1, 8)$, $\sigma^2 = 10$, $\sigma^2_{\text{obs}} = 25$ and performed inference on θ and σ. Other details are as in the Section 4.2.

Our tuning diagnostic for the number of particles suggested using $N = 5{,}000$ for EnKF. We ran eMCMC for 10,000 iterations. We got an acceptance rate of 18% indicating good mixing. The ESS was 321 and the compute time was 79,097 s. We could not compare with gold standard pMCMC results here, as this algorithm was not computationally feasible for this high dimensional example (as even with 10,000 particles, log likelihood estimates had standard deviations far above our diagnostic target of 1.5, while MCMC runtimes were in the order of days). However, Appendix D of the Supplementary Materials shows that the approximate posterior marginals from eMCMC are consistent with the true parameter values. This demonstrates that eMCMC can produce sensible results for problems where the state has (moderately) high dimension.

4.4 Lotka Volterra Example

Model and Inference Task

The Lotka-Volterra predator-prey model (e.g. Boys et al., 2008) describes the continuous time evolution of a non-negative integer-valued process $X_t = (X_{1,t}, X_{2,t})'$ where $X_{1,t}$ denotes prey and $X_{2,t}$ denotes predator. Starting from an initial value, X_t evolves according to a Markov jump process (MJP) parameterised by rate constants $c = (c_1, c_2, c_3)'$ and characterised by the instantaneous rate or hazard function $h(x_t, c) = (h_1(x_t, c_1), h_2(x_t, c_2), h_3(x_t, c_3))'$. Transitions over $(t, t + dt]$ take the form of one of three types (prey reproduction, prey death / predator reproduction, predator death) with associated probabilities given by

$$\begin{aligned}
\Pr\{X_{1,t+dt} = x_{1,t} + 1, X_{2,t+dt} = x_{2,t} | x_t\} &= h_1(x_t, c_1)dt + o(dt), \\
\Pr\{X_{1,t+dt} = x_{1,t} - 1, X_{2,t+dt} = x_{2,t} + 1 | x_t\} &= h_2(x_t, c_2)dt + o(dt), \\
\Pr\{X_{1,t+dt} = x_{1,t}, X_{2,t+dt} = x_{2,t} - 1 | x_t\} &= h_3(x_t, c_3)dt + o(dt).
\end{aligned}$$

The hazard function for this system is

$$h(X_t, c) = (c_1 x_{1,t}, c_2 x_{1,t} x_{2,t}, c_3 x_{2,t})'.$$

It is then relatively simple to generate realisations of this process via Gillespie's direct method (Gillespie, 1977), where at time t, the dwell time between transition events is drawn from an exponential distribution with rate $h_0(x_t, c) = \sum_{i=1}^{3} h_i(x_t, c_i)$ and the transition is type i with probability proportional to $h_i(x_t, c_i)$.

We assume that the MJP is observed with Gaussian error so that

$$Y_t | X_t \sim \mathcal{N}\left\{ \begin{pmatrix} x_{1,t} \\ x_{2,t} \end{pmatrix}, \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix} \right\}.$$

As all parameters of interest must be strictly positive, we consider inference for

$$\theta = (\log c_1, \log c_2, \log c_3, \log \sigma_1, \log \sigma_2)'.$$

We consider two synthetic data sets ($\mathcal{D}_1$ and $\mathcal{D}_2$) simulated with rate parameters $c = (0.5, 0.0025, 0.3)'$ and initial condition $x_0 = (71, 79)'$. We further assume $\sigma_1 = \sigma_2 = 1$ to be unknown. To allow the analysis of two data-poor scenarios, dataset $\mathcal{D}_1$ has 51 equally spaced observations on $[0, 50]$ and dataset $\mathcal{D}_2$ is constructed by thinning $\mathcal{D}_1$ to give 26 equally spaced observations on $[0, 25]$.

Inference

We compare the performance of EnKF to the gold standard auxiliary particle filter (APF) driven pMCMC scheme described in Golightly and Wilkinson (2015). In brief, state particles are propagated using Gillespie's direct method, with the hazard function replaced by an approximate conditioned hazard, derived from a linear Gaussian approximation to the MJP. Full details of this approach, including the calculation of the particle filter weights can be found in Golightly and Wilkinson (2015).

We follow the practical advice given in Section 2.2 to choose the number of particles / ensemble members N and the scaling of the innovation variance in the random walk proposal distribution. We assumed independent uniform $U(-8, 8)$ priors for the components of θ and ran both eMCMC and pMCMC for 10^5 iterations. Since the EnKF treats the state as continuous, eMCMC used a reflecting barrier at 0 to avoid the state of the system going negative.

The results are summarised by Table 2 and Figure 8, with additional results shown in Appendix E of the Supplementary Materials. We see that for both data sets, the output of eMCMC is consistent with the true values that produced the data and, more importantly, the ground truth posterior based on the output of pMCMC. For dataset $\mathcal{D}_1$, eMCMC required more particles than pMCMC but gives better overall efficiency (as measured by the ESS per second) since sampling from the propagation construct in the auxiliary particle filter is relatively expensive. We see an increase of about a factor of 3. For dataset $\mathcal{D}_2$, the number of particles required by pMCMC must be increased, since the propagation construct is based on a linear Gaussian approximation of the true (but unknown) hazard function of the conditioned MJP. The construct breaks down as observations are made sparsely in time (and the dynamics of the conditioned process are nonlinear between observations). Ensemble MCMC on the other hand seems to work well, requiring even fewer particles than for $\mathcal{D}_1$. We see an increase in overall efficiency (compared to pMCMC) of a factor of around 55.

4.5 Autoregulatory Network Example

Model and Inference Task

A commonly used mechanism for auto-regulation in prokaryotes which has been well-studied and modelled is a negative feedback mechanism whereby dimers of a protein

Filter	N	τ	Acc. rate	ESS	Time (s)	ESS/Time
	$\mathcal{D}_1$ (51 obs. every 1 time unit)					
APF	55	1.4	0.11	1117	26298	0.042
EnKF	150	1.4	0.11	1762	12299	0.143
	$\mathcal{D}_2$ (26 obs. every 2 time units)					
APF	350	1.4	0.08	1156	165015	0.007
EnKF	65	1.4	0.11	2054	5282	0.389

Table 2: Summaries for the Lotka Volterra application: number of particles N, standard deviation of the noise in the log-posterior (τ) at the posterior median, acceptance rate, multivariate ESS, wall clock time in seconds and ESS per second.

repress its own transcription (e.g. Arkin et al., 1998). A simplified model for such a prokaryotic auto-regulation, based on this mechanism of dimers of a protein coded for by a gene repressing its own transcription into RNA, can be found in Golightly and Wilkinson (2005) (see also Golightly and Wilkinson, 2011).

Let $X_t = (X_{1,t}, X_{2,t}, X_{3,t}, X_{4,t}, X_{5,t})'$ denote the number of copies of the unbound gene $X_{1,t}$, bound gene $X_{2,t}$, RNA $X_{3,t}$, protein $X_{4,t}$ and dimers of the protein $X_{5,t}$. We assume that X_t evolves according to a Markov jump process. The possible transitions can be succinctly described by the pseudo-reaction list

$$\begin{aligned}
&\mathcal{R}_1: \quad X_1 + X_5 \longrightarrow X_2 && \mathcal{R}_2: \quad X_2 \longrightarrow X_1 + X_5\\
&\mathcal{R}_3: \quad X_1 \longrightarrow X_1 + X_3 && \mathcal{R}_4: \quad X_3 \longrightarrow X_3 + X_4\\
&\mathcal{R}_5: \quad 2X_4 \longrightarrow X_5 && \mathcal{R}_6: \quad X_5 \longrightarrow 2X_4\\
&\mathcal{R}_7: \quad X_3 \longrightarrow \emptyset && \mathcal{R}_8: \quad X_4 \longrightarrow \emptyset
\end{aligned}$$

where, for example, occurrence of $\mathcal{R}_1$ at time t reduces $X_{1,t}$ and $X_{5,t}$ by 1, increases $X_{2,t}$ by 1, and leaves the remaining components unchanged. The associated hazard function is

$$h(x_t, c) = (c_1 x_{1,t} x_{5,t}, c_2 x_{2,t}, c_3 x_{1,t}, c_4 x_{3,t}, c_5 x_{4,t}(x_{4,t} - 1)/2, c_6 x_{5,t}, c_7 x_{3,t}, c_8 x_{4,t})'.$$

We consider here two challenging synthetic datasets, each consisting of 101 observations at integer times on $X_{3,t}$ (RNA) and total protein counts, $X_{4,5} + 2X_{5,t}$ so that $X_{1,t}$, $X_{2,t}$, $X_{4,t}$ and $X_{5,t}$ are not observed exactly. Moreover, as in Section 4.4, we corrupt the observations by adding independent, Gaussian $\mathcal{N}\{0, \text{diag}(\sigma_1^2, \sigma_2^2)\}$ innovations to each count. We fix $\sigma_1 = \sigma_2 = 1$ for dataset $\mathcal{D}_1$ and $\sigma_1 = \sigma_2 = 0$ for dataset $\mathcal{D}_2$. These variance components are assumed known. Following Golightly and Wilkinson (2005), we use the rate constants

$$c = (0.1, 0.7, 0.35, 0.2, 0.1, 0.9, 0.3, 0.1)'.$$

We assume that the initial condition $x_0 = (5, 5, 8, 8, 8)'$, the measurement error variances and the rate constants of the reversible dimerisation reactions (c_5 and c_6) are known leaving $\theta_i = \log c_i$, $i = 1, 2, 3, 4, 7, 8$ as the object of inference.

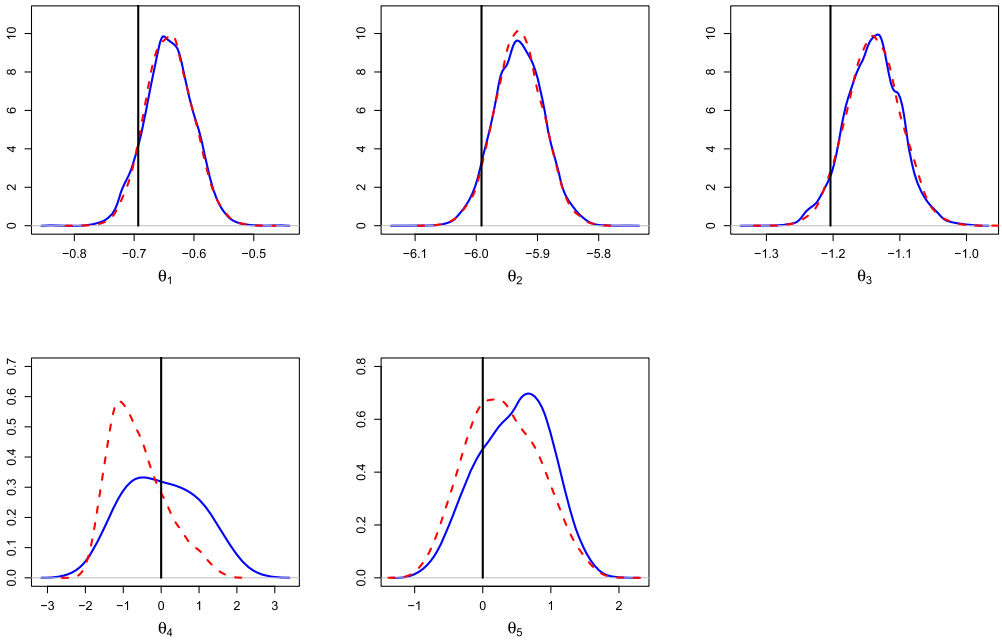

Figure 8: Lotka Volterra dataset $\mathcal{D}_1$. Marginal posterior densities based on the output of pMCMC (solid) and eMCMC (dashed).

Inference

We again compare the performance of eMCMC to the gold standard auxiliary particle filter driven pMCMC scheme described in Golightly and Wilkinson (2015). The number of particles / ensemble members N was chosen as in Section 2.2. We assigned independent Gamma $Ga(1, 0.5)$ priors to each unknown rate constant and ran eMCMC and pMCMC for 2×10^5 iterations. Note that when running eMCMC for dataset $\mathcal{D}_2$, the (assumed known) values of σ_1 and σ_2 result in no shifting of the ensemble members, rendering this step ineffectual. We therefore ran eMCMC for this scenario by setting the measurement error variance to be "small" throughout the algorithm's execution. Specifically, we found that setting $\sigma_1^2 = \sigma_2^2 = 0.01$ gave reasonable mixing, at the expense of introducing additional bias into the eMCMC posterior.

Table 3 and Figure 9 summarise the results. Appendix F of the Supplementary Materials shows posterior estimates for dataset $\mathcal{D}_2$. It is clear that eMCMC gives output that is consistent with the true values that produced the data and output from pMCMC, which exactly targets the posterior of interest. We therefore compare overall efficiency of eMCMC and pMCMC in terms of ESS per second, as reported in Table 3. For dataset $\mathcal{D}_1$, eMCMC requires half the number of particles of pMCMC and gives a comparable ESS value. In terms of overall efficiency, eMCMC outperforms pMCMC by around a factor of 4. For dataset $\mathcal{D}_2$, pMCMC requires around 2000 particles, due to the strict requirement of particle trajectories having to "hit" the observations to receive a non

Filter	N	τ	Acc. rate	ESS	Time (s)	ESS/Time
			$\mathcal{D}_1$ $(\sigma_1 = \sigma_2 = 1)$			
APF	400	1.3	0.15	3348	72081	0.046
EnKF	200	1.5	0.13	2972	15862	0.187
			$\mathcal{D}_2$ $(\sigma_1 = \sigma_2 = 0)$			
APF	2000	1.4	0.11	3314	403456	0.0082
EnKF	370	1.4	0.11	3176	34112	0.0931

Table 3: Summaries for the autoregulatory example: number of particles N, standard deviation of the noise in the log-posterior (τ) at the posterior median, acceptance rate, multivariate ESS, wall clock time in seconds and ESS per second.

zero weight. Ensemble MCMC on the other hand is able to give a comparable ESS value with just 370 particles. Consequently, for this example, eMCMC outperforms pMCMC by around a factor of 11.

4.6 Neuroscience Example

Model

We investigate the following realistic Neural Population Model (NPM) for brain activity. This model (see e.g. Bojak and Liley, 2005) is known as the Liley model, and Bayesian inference for the parameters of this model has previously been described by Maybank et al. (2017). Here a high-level description of the model is presented; more detail can be found in this latter paper. The model is of two neural populations: one population that has an excitatory effect on the activity of connected neurons, the other that has an inhibitory effect. The model consists of the following differential equations, where $k = e, i$ for excitatory and inhibitory contributions:

$$\left(\frac{d}{dt} + \gamma_{ek}\right)\left(\frac{d}{dt} + \bar{\gamma}_{ek}\right) I_{ek}(t) = \exp\left(\gamma_{ek} d_{ek}\right) \Gamma_{ek} \tilde{\gamma}_{ek} \times \left[N_{ek}^{\beta} S_e\left(h_e(t)\right) + \Phi_{ek}(t) + \bar{p}_{ek} + \delta_{ek} p(t)\right], \quad (11)$$

$$\left(\frac{d}{dt} + \gamma_{ik}\right)\left(\frac{d}{dt} + \bar{\gamma}_{ik}\right) I_{ik}(t) = \exp\left(\gamma_{ik} d_{ik}\right) \Gamma_{ik} \tilde{\gamma}_{ik} \left[N_{ik}^{\beta} S_i\left(h_i(t)\right)\right], \quad (12)$$

$$\left(\frac{d}{dt} + v\Lambda\right)^2 \Phi_{ek}(t) = v^2 \Lambda^2 N_{ek}^{\alpha} S_e\left(h_e(t)\right), \quad (13)$$

$$\Phi_{ek}(t) = 0, \quad (14)$$

where the Kronecker delta δ_{ek} admits only excitatory noise input p (white noise with zero mean and fixed standard deviation) to this stochastic differential equation system, and where S is a sigmoidal activation function. The 14 state variables are time-evolving properties of the two populations: h_e, h_i (the mean soma membrane potentials); $I_{ee}, I_{ei}, I_{ii}, I_{ii}$ (local reaction to synaptic inputs) and their differentials; and Φ_{ee}, Φ_{ei} (long-range propagation of activity) and their differentials. The parameters

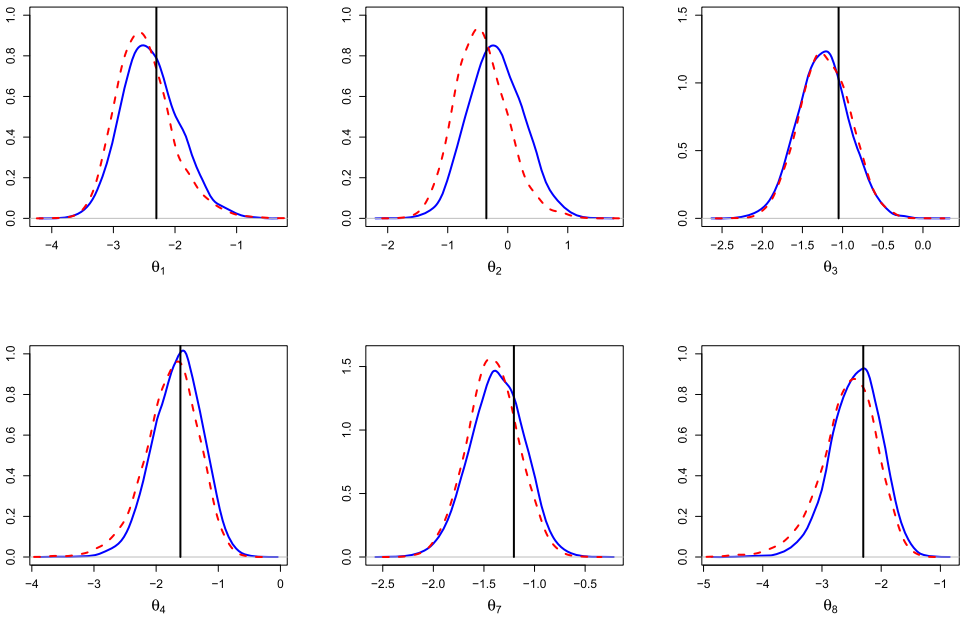

Figure 9: Autoregulatory dataset $\mathcal{D}_1$. Marginal posterior densities based on the output of pMCMC (blue solid) and eMCMC (red dash).

are $\Gamma_{ee}, \Gamma_{ei}, \Gamma_{ie}, \Gamma_{ii}$ (the synaptic input peak amplitude) $\gamma_{ee}, \gamma_{ei}, \gamma_{ie}, \gamma_{ii}$ (shape parameters of the post synaptic potentials) and $\bar{p}_{ee} = 6603.4, \bar{p}_{ei} = 2625.7, \sigma = 0.01$ (the mean rate of the extracortical inputs). The parameters Λ (decay scale of long-range connectivity) v (axonal conduction velocity) and $d_{ee}, d_{ei}, d_{ie}, d_{ii}$ (rise times to peak of the post synaptic potentials) are fixed.

We model an electroencephalogram (EEG) time-series as noisy observations of the h_e variable of this NPM, assuming that the EEG observations are linearly proportional to h_e with some added observational noise:

$$y_i = h_e(i \cdot \Delta t) + z_i \tag{15}$$

where Δt is some constant time-step and the z_i are iid normal random variables $z_i \sim \mathcal{N}\left(0, \sigma^2\right)$ for $i = 0, \ldots, n-1$.

In this paper an input noise of variance 10^8 was used to simulate data, and the dynamics were simulated using the Euler-Maruyama method with step size 2.5×10^{-3}. Appendix G of the Supplementary Materials gives the prior distributions for the parameters that were treated as unknown, giving the uniform priors that restrict the parameters to ranges found to be plausible in Bojak and Liley (2005); other parameters were fixed to values chosen from the ranges given by Bojak and Liley (2005).

Results

We compared the performance of eMCMC and pMCMC on data simulated from the Liley model for parameters that result in quasi-linear dynamics about a stable fixed point. The work in Maybank et al. (2017) suggests that the accuracy of the parameter posterior is likely to be improved by using a method that is suitable for non-linear systems (such as pMCMC) compared to using a linearised approach such as the extended Kalman filter or the approach introduced in Maybank et al. (2017). Both MCMC approaches are forms of Metropolis-Hastings, with a truncated multivariate normal proposal for the 11 parameters and covariance chosen to be $2.562^2/11$ times the estimated posterior covariance from pilot runs (this scaling being recommended by Sherlock et al., 2015). We considered a situation that is challenging for a particle filter, with a relatively small measurement noise of $\sigma = 0.01$.

We study a simulated data set, generated from the model using Euler-Maruyama approximation. The data, shown in Figure 10 has a length of 4 s and a sampling frequency of 50 Hz, and was generated for the parameters ($\Gamma_{ee} = 0.10631, \Gamma_{ei} = 0.64105, \Gamma_{ie} = 0.46477, \Gamma_{ii} = 0.28663, \gamma_{ee} = 291.5, \gamma_{ei} = 697.76, \gamma_{ie} = 458.67, \gamma_{ii} = 82.33, \bar{p}_{ee} = 6603.4, \bar{p}_{ei} = 2625.7, \sigma = 0.01$).

We ran 40 chains of 1000 iterations of pMCMC and eMCMC on this data. All were initialised from the parameters at which the data was generated and used a burn in of 500 iterations. Based on the scheme described in Section 2.2, we chose 1000 particles for the BPF in pMCMC, and 100 ensemble members for the EnKF in eMCMC. Both algorithms were implemented with early rejection schemes, as detailed in Section 3.4. In both cases the early rejection results in a reduction in computational cost of approximately a factor of two; with this scheme each iteration of pMCMC took an average of 1415 s, compared to the average of 91 s for eMCMC. The mean acceptance rate for pMCMC was 0.33%, compared to 0.93% for eMCMC, indicating that eMCMC is (by this measure) is approximately three times as efficient whilst being more than 15 times faster. Pilot runs on longer simulated time series suggest that the efficiency of eMCMC (relative to pMCMC) improves as the length of the time series increases, but in these cases the computational cost of pMCMC was too large to permit a rigorous comparison. Kernel density estimates of the marginal posterior of each parameter are shown in Figure 10: we observe that the posteriors obtained by both methods are similar.

5 Discussion

In this paper we replace the BPF with the EnKF within a pMCMC algorithm. We have demonstrated on a variety of examples that significant computational gains can be achieved without sacrificing much on posterior accuracy.

We expect eMCMC to work relatively well on applications with characteristics that may reduce the performance of particle filters such as: intractable transition densities for complex models, sparse observations, small observation variance, surprising observations and/or model misspecification. Our methods may perform comparatively less well for models that have some level of tractability that permit more advanced particle

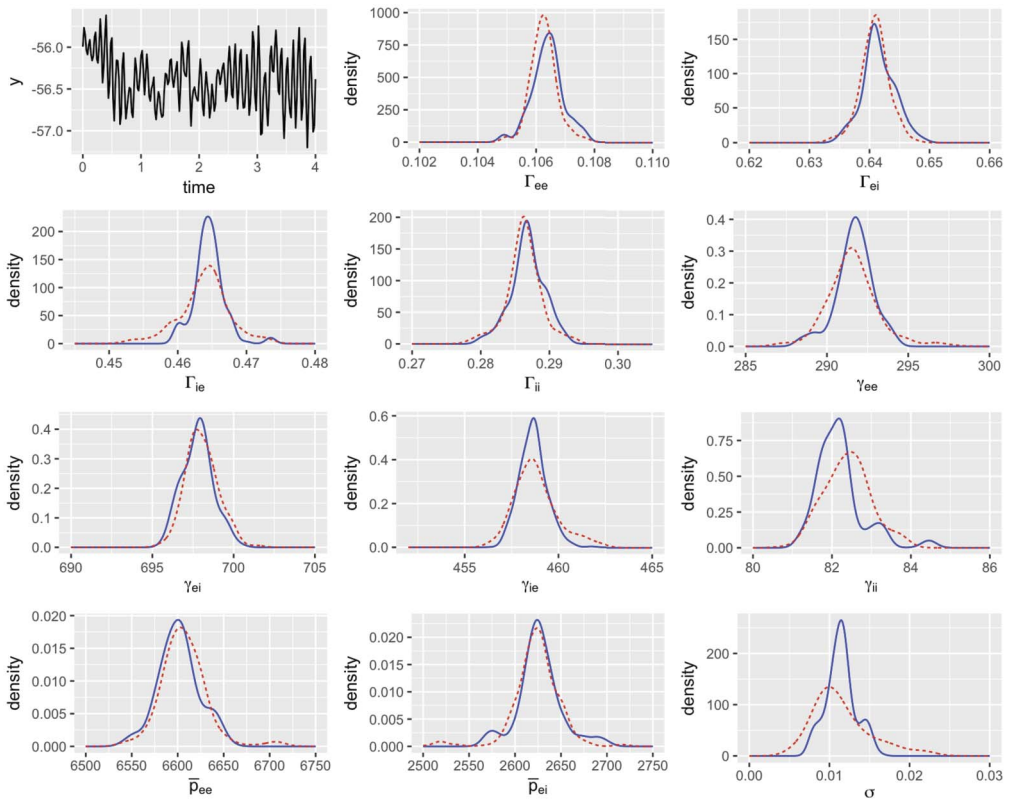

Figure 10: Top left: simulated data from the Liley model with added measurement noise. Other plots: estimated marginal posterior distributions for the parameters of the Liley model based on pMCMC (blue solid) and eMCMC (red dash).

filters. However, for the Markov jump process applications we obtained competitive results even when particle filters exploited cleverly-designed guided proposals and where eMCMC simply simulated directly according to the evolution density. In some sense, eMCMC is a middle ground between exact-approximate particle MCMC and pure data assimilation methods; we attempt to gain speed improvements over pMCMC via some data assimilation ideas and obtain more accurate posterior approximations compared to pure data assimilation by using a similar algorithm structure to pMCMC.

If exact posterior inferences are essential, there are likely to be ways to exploit our EnKF approach to improve computational performance. For example, it could be used as the cheap approximate likelihood within a delayed-acceptance MCMC algorithm (e.g. Sherlock et al., 2017 and Golightly et al., 2015) or importance sampling scheme (Franks and Vihola, 2017). Alternatively, we might bridge our approximate posterior with the true posterior using sequential Monte Carlo. Further, our approach could be used in pilot MCMC runs to more quickly identify the regions of the parameter space

with non-negligible posterior support and assist MCMC tuning generally. Particularly in the posterior tails we find that the EnKF likelihood estimator has significantly lower variance than the BPF likelihood estimator.

It is important to note that there will likely be many applications where the EnKF approximation may not be appropriate. The approach relies on being able to approximate the filtering distribution reasonably well with a Gaussian density. However, our paper illustrates that there are a wide class of models where our approach can provide reasonable accuracy. Further, Katzfuss et al. (2019) present a hierarchical approach for allowing non-Gaussian observation densities with EnKF methods, which would also be applicable to our approach.

We chose the EnKF to approximate the likelihood in this paper since it respects non-linear evolution models. However, we note that other approximate KF methods could be adopted. For example, the extended KF and its variations/extensions (see, for example, Law et al., 2015, chap. 4 and Asch et al., 2016, chap. 6). Further, for problems with higher state space dimensions, innovations in the EnKF literature such as localisation and inflation could be considered (see, for example, Evensen, 2009, chap. 15). Investigating these different approximate filters is an avenue for future research.

An alternative to MCMC is state augmentation: processing the data once to infer both θ and x. In Section 4.2 we implemented one such method, particle EnKF, and found that eMCMC performed better in terms of speed and accuracy. Nonetheless, this or other versions of state augmentation may well perform better in other settings. Further, our method is strictly an offline method whereas particle EnKF can also be applied in an online setting (see Table 1 of Katzfuss et al. (2019)). Alternatively, we suggest that our approach could be incorporated into the SMC^2 algorithm of Chopin et al. (2013), which uses an MCMC kernel for jittering particles and thus preserves the current target. We leave that for further research.

We did not consider posterior inference for the hidden states in this paper. It is possible to combine our method with the ensemble Kalman smoother of van Leeuwen and Evensen (1996), as proposed in Katzfuss et al. (2019). The framework described in Katzfuss et al. (2019) may also be used to extend the methods in this paper to nonlinear non-Gaussian measurement models through the use of a transformation.

In this paper we compared the most commonly used particle filter (the BPF, except in the Markov jump process examples) and EnKF (the stochastic EnKF) within MCMC algorithms. In future work it would be interesting to compare extensions to both approaches. Extensions to the BPF are familiar to many in computational statistics (e.g. adaptive resampling, MCMC rejuvenation moves, the auxiliary PF) and the improvements they can bring to particle MCMC algorithms are relatively well understood. In the paper we have seen how ideas previously used in the pMCMC context (i.e. using Quasi Monte Carlo, and the correlated approach) can also be exploited in the EnKF case. Other extensions and alternatives to the stochastic EnKF from the DA literature are also possible and have the potential to provide further improvements in the pMCMC setting. Examples are: the deterministic EnKF (Tippett et al., 2003), which uses a deterministic rather than a stochastic transformation in the shift step, which may

further reduce the variance of the likelihood estimates (but which may introduce further bias for nonlinear models); or the equivalent weights particle filter (van Leeuwen, 2010), which uses a deterministic transform in each step of a PF to avoid degeneracy (but which may, again, introduce bias).

Supplementary Material

Ensemble MCMC: Accelerating Pseudo-Marginal MCMC for State Space Models using the Ensemble Kalman Filter (DOI: 10.1214/20-BA1251SUPP; .pdf).

References

Agapiou, S., Papaspiliopoulos, O., Sanz-Alonso, D., and Stuart, A. M. (2017). "Importance sampling: computational complexity and intrinsic dimension." *Statistical Science*, 32(3): 405–431. MR3696003. doi: https://doi.org/10.1214/17-STS611. 224

Andrieu, C., Doucet, A., and Holenstein, R. (2010). "Particle Markov chain Monte Carlo methods." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3): 269–342. MR2758115. doi: https://doi.org/10.1111/j.1467-9868.2009.00736.x. 223, 228

Andrieu, C. and Roberts, G. O. (2009). "The pseudo-marginal approach for efficient Monte Carlo computations." *The Annals of Statistics*, 37(2): 697–725. MR2502648. doi: https://doi.org/10.1214/07-AOS574. 223, 227

Arkin, A., Ross, J., and McAdams, H. H. (1998). "Stochastic kinetic analysis of developmental pathway bifurcation in phage λ-infected *Escherichia coli* cells." *Genetics*, 149: 1633–1648. 248

Asch, M., Bocquet, M., and Nodet, M. (2016). *Data assimilation: methods, algorithms, and applications*, volume 11. SIAM. MR3602006. doi: https://doi.org/10.1137/1.9781611974546.pt1. 254

Beskos, A., Crisan, D., and Jasra, A. (2014). "On the stability of sequential Monte Carlo methods in high dimensions." *The Annals of Applied Probability*, 24(4): 1396–1445. MR3211000. doi: https://doi.org/10.1214/13-AAP951. 224

Bojak, I. and Liley, D. T. J. (2005). "Modeling the effects of anesthesia on the electroencephalogram." *Physical Review E*, 71(4): 041902. 250, 251

Boys, R. J., Wilkinson, D. J., and Kirkwood, T. B. L. (2008). "Bayesian inference for a discretely observed stochastic kinetic model." *Statistics and Computing*, 18: 125–135. MR2390814. doi: https://doi.org/10.1007/s11222-007-9043-x. 246

Carrassi, A., Bocquet, M., Hannart, A., and Ghil, M. (2017). "Estimating model evidence using data assimilation." *Quaterly Journal of the Royal Meteorological Society*, 143: 866–880. MR3602006. doi: https://doi.org/10.1137/1.9781611974546.pt1. 225

Chopin, N., Jacob, P. E., and Papaspiliopoulos, O. (2013). "SMC^2: an efficient algorithm for sequential analysis of state space models." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75(3): 397–426. MR3065473. doi: https://doi.org/10.1111/j.1467-9868.2012.01046.x. 254

Choppala, P., Gunawan, D., Chen, J., Tran, M.-N., and Kohn, R. (2016). "Bayesian inference for state space models using block and correlated pseudo marginal methods." Available from arXiv:1612.07072. 234

Dahlin, J., Lindsten, F., Kronander, J., and Schön, T. B. (2015). "Accelerating pseudo-marginal Metropolis-Hastings by correlating auxiliary variables." *arXiv preprint arXiv:1511.05483*. 233

Del Moral, P. (2004). *Feynman-Kac formulae*. Springer. MR2044973. doi: https://doi.org/10.1007/978-1-4684-9393-1. 228

Deligiannidis, G., Doucet, A., and Pitt, M. K. (2018). "The correlated pseudo-marginal method." *Journal of the Royal Society: Series B (Statistical Methodology)*, 80(5): 839–870. MR3874301. doi: https://doi.org/10.1111/rssb.12280. 233, 234

Doucet, A. and Johansen, A. M. (2011). "A tutorial on particle filtering and smoothing: Fifteen years later." In Crisan, D. and Rozovskii, B. (eds.), *Handbook of nonlinear filtering*, 656–704. Oxford University Press. MR2884612. 228

Doucet, A., Pitt, M. K., Deligiannidis, G., and Kohn, R. (2015). "Efficient implementation of Markov chain Monte Carlo when using an unbiased likelihood estimator." *Biometrika*, 102(2): 295–313. MR3371005. doi: https://doi.org/10.1093/biomet/asu075. 229

Drovandi, C. C. and Tran, M.-N. (2018). "Improving the efficiency of fully Bayesian optimal design of experiments using randomised quasi-Monte Carlo." *Bayesian Analysis*, 13(1): 139–162. MR3737946. doi: https://doi.org/10.1214/16-BA1045. 233

Drovandi, C., Everitt, R. G., Golightly, A., and Prangle, D. (2020). "Supplementary Material of "Ensemble MCMC: Accelerating Pseudo-Marginal MCMC for State Space Models using the Ensemble Kalman Filter"." *Bayesian Analysis*. doi: https://doi.org/10.1214/20-BA1251SUPP. 239

Evensen, G. (1994). "Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics." *Journal of Geophysical Research*, 99: 10143–10162. 225, 229

Evensen, G. (2007). *The Ensemble Kalman filter*. Springer. 225

Evensen, G. (2009). *Data assimilation: the ensemble Kalman filter*. Springer Science & Business Media. MR2555209. doi: https://doi.org/10.1007/978-3-642-03711-5. 254

Everitt, R. G. and Sibly, R. M. (2019). "Comparing ABC and particle MCMC for inference of intractable temporal models." Technical report, University of Reading. 235

Fasiolo, M., Pya, N., and Wood, S. N. (2016). "A comparison of inferential methods for highly non-linear state space models in ecology and epidemiology." *Statistical Science*, 31(1): 96–118. MR3458595. doi: https://doi.org/10.1214/15-STS534. 225

Fearnhead, P. and Künsch, H. R. (2018). "Particle filters and data assimilation." *Annual Review of Statistics and Its Application*, 5: 421–449. MR3774754. doi: https://doi.org/10.1146/annurev-statistics-031017-100232. 228

Franks, J. and Vihola, M. (2017). "Importance sampling correction versus standard averages of reversible MCMCs in terms of the asymptotic variance." Available from arXiv:1706.09873. MR4140030. doi: https://doi.org/10.1016/j.spa.2020.05.006. 253

Gerber, M. and Chopin, N. (2015). "Sequential quasi Monte Carlo." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 77(3): 509–579. MR3351446. doi: https://doi.org/10.1111/rssb.12104. 233

Ghurye, S. G. and Olkin, I. (1969). "Unbiased estimation of some multivariate probability densities and related functions." *The Annals of Mathematical Statistics*, 40(4): 1261–1271. MR0245125. doi: https://doi.org/10.1214/aoms/1177697501. 234

Gillespie, D. T. (1977). "Exact stochastic simulation of coupled chemical reactions." *Journal of Physical Chemistry*, 81: 2340–2361. 246

Golightly, A., Henderson, D. A., and Sherlock, C. (2015). "Delayed acceptance particle MCMC for exact inference in stochastic kinetic models." *Statistics and Computing*, 25(5): 1039–1055. MR3375634. doi: https://doi.org/10.1007/s11222-014-9469-x. 253

Golightly, A. and Wilkinson, D. J. (2005). "Bayesian inference for stochastic kinetic models using a diffusion approximation." *Biometrics*, 61(3): 781–788. MR2196166. doi: https://doi.org/10.1111/j.1541-0420.2005.00345.x. 248

Golightly, A. and Wilkinson, D. J. (2011). "Bayesian parameter inference for stochastic biochemical network models using particle Markov chain Monte Carlo." *Interface Focus*, 1(6): 807–820. 248

Golightly, A. and Wilkinson, D. J. (2015). "Bayesian inference for Markov jump processes with informative observations." *Statistical Applications in Genetics and Molecular Biology*, 14(2): 169–188. MR3331772. doi: https://doi.org/10.1515/sagmb-2014-0070. 230, 247, 249

Gordon, N. J., Salmond, D. J., and Smith, A. F. M. (1993). "Novel approach to nonlinear/non-Gaussian Bayesian state estimation." *IEE Proceedings F (Radar and Signal Processing)*, 140(2): 107–113. 224

Houtekamer, P. L., Deng, X., Mitchell, H. L., Baek, S.-J., and Gagnon, N. (2014). "Higher resolution in an operational ensemble Kalman filter." *Monthly Weather Review*, 142: 1143–1162. 230

Katzfuss, M., Stroud, J. R., and Wikle, C. K. (2016). "Understanding the en-

semble Kalman filter." *The American Statistician*, 70(4): 350–357. MR3574787. doi: https://doi.org/10.1080/00031305.2016.1141709. 225, 229, 230

Katzfuss, M., Stroud, J. R., and Wikle, C. K. (2019). "Ensemble Kalman methods for high-dimensional hierarchical dynamic space-time models." *Journal of the American Statistical Association*, 115: 866–885. MR4107685. doi: https://doi.org/10.1080/01621459.2019.1592753. 225, 226, 230, 231, 244, 254

Law, K., Stuart, A., and Zygalakis, K. (2015). "Data assimilation." *Cham, Switzerland: Springer*. MR3363508. doi: https://doi.org/10.1007/978-3-319-20325-6. 254

Lei, J., Bickel, P. J., and Snyder, C. (2010). "Comparison of ensemble Kalman filters under non-Gaussianity." *Monthly Weather Review*, 138: 1293–1306. MR2942007. 230

Liu, J. and West, M. (2001). "Combined parameter and state estimation in simulation-based filtering." In Doucet, A., de Freitas, N., and Gordon, N. (eds.), *Sequential Monte Carlo Methods in Practice*. New York: Springer. MR1847784. doi: https://doi.org/10.1007/978-1-4757-3437-9_1. 225

Lorenz, E. N. (1963). "Deterministic nonperiodic flow." *Journal of the Atmospheric Sciences*, 20(2): 130–141. MR4021434. doi: https://doi.org/10.1175/1520-0469(1963)020<0130:DNF>2.0.CO;2. 242

Lorenz, E. N. (1996). "Predictability: A problem partly solved." In *Proc. Seminar on Predictability*. 245

L'Ecuyer, P. and Lemieux, C. (2005). "Recent advances in randomized quasi-Monte Carlo methods." In *Modeling Uncertainty*, 419–474. Springer. MR1893290. doi: https://doi.org/10.1007/0-306-48102-2_20. 232

Maybank, P., Bojak, I., and Everitt, R. G. (2017). "Fast approximate Bayesian inference for stable differential equation models." *arXiv preprint arXiv:1706.00689*. 250, 252

Minvielle, P., Todeschini, A., Caron, F., and Del Moral, P. (2014). "Particle MCMC for Bayesian microwave control." In *Journal of Physics: Conference Series*, volume 542, 012007. IOP Publishing. 226

Mitchell, H. L. and Houtekamer, P. L. (2000). "An adaptive ensemble Kalman filter." *Monthly Weather Review*, 128: 416–433. 225

Peters, G. W., Hosack, G. R., and Hayes, K. R. (2010). "Ecological non-linear state space model selection via adaptive particle Markov chain Monte Carlo (AdPMCMC)." *arXiv preprint arXiv:1005.2238*. 237

Pitt, M., Silva, R., Giordani, P., and Kohn, R. (2010). "Auxiliary particle filtering within adaptive Metropolis-Hastings sampling." *arXiv preprint arXiv:1006.1914*. 228, 237

Pitt, M. K., dos Santos Silva, R., Giordani, P., and Kohn, R. (2012). "On some properties of Markov chain Monte Carlo simulation methods based on the particle filter." *Journal of Econometrics*, 171(2): 134–151. MR2991856. doi: https://doi.org/10.1016/j.jeconom.2012.06.004. 230

Plummer, M., Best, N., Cowles, K., and Vines, K. (2006). "CODA: convergence diagnosis and output analysis for MCMC." *R News*, 6(1): 7–11. 236, 244

Prangle, D., Everitt, R. G., and Kypraios, T. (2018). "A rare event approach to high-dimensional approximate Bayesian computation." *Statistics and Computing*, 28(4): 819–834. MR3766045. doi: https://doi.org/10.1007/s11222-017-9764-4. 235

Price, L. F., Drovandi, C. C., Lee, A., and Nott, D. J. (2018). "Bayesian synthetic likelihood." *Journal of Computational and Graphical Statistics*, 27(1): 1–11. MR3788296. doi: https://doi.org/10.1080/10618600.2017.1302882. 225, 234

Särkkä, S. (2013). *Bayesian filtering and smoothing*. Cambridge University Press. MR3154309. doi: https://doi.org/10.1017/CBO9781139344203. 226, 228

Sherlock, C., Golightly, A., and Henderson, D. A. (2017). "Adaptive, delayed-acceptance MCMC for targets with expensive likelihoods." *Journal of Computational and Graphical Statistics*, 26(2): 434–444. MR3640199. doi: https://doi.org/10.1080/10618600.2016.1231064. 253

Sherlock, C., Thiery, A. H., Roberts, G. O., and Rosenthal, J. S. (2015). "On the efficiency of pseudo-marginal random walk Metropolis algorithms." *The Annals of Statistics*, 43(1): 238–275. MR3285606. doi: https://doi.org/10.1214/14-AOS1278. 229, 252

Sisson, S. A., Fan, Y., and Beaumont, M. (2018). *Handbook of approximate Bayesian computation*. Chapman and Hall/CRC. MR3889281. 225

Snyder, C., Bengtsson, T., Bickel, P. J., and Anderson, J. (2008). "Obstacles to high-dimensional particle filtering." *Monthly Weather Review*, 136: 4629–4640. 224

Stroud, J. R., Katzfuss, M., and Wikle, C. K. (2018). "A Bayesian adaptive ensemble Kalman filter for sequential state and parameter estimation." *Monthly Weather Review*, 146: 373–386. 225, 231

Stroud, J. R., Stein, M. L., Lesht, B. M., Schwab, D. J., and Beletsky, D. (2010). "An ensemble Kalman filter and smoother for satellite data assimilation." *Journal of the American Statistical Association*, 105: 978–990. MR2752594. doi: https://doi.org/10.1198/jasa.2010.ap07636. 225, 231

Tippett, M. K., Anderson, J. L., Bishop, C. H., Hamill, T. M., and Whitaker, J. S. (2003). "Ensemble square root filters." *Monthly Weather Review*, 131(7): 1485–1490. 254

Tran, M.-N., Nott, D., and Kohn, R. (2017). "Varitational Bayes with intractable likelihood." *Journal of Computational and Graphical Statistics*, 26(4): 873–882. arXiv:1503.08621. MR3765351. doi: https://doi.org/10.1080/10618600.2017.1330205. 233

van Leeuwen, P. J. (2010). "Nonlinear data assimilation in geosciences: an extremely efficient particle filter." *Quarterly Journal of the Royal Meteorological Society*, 136: 1991–1999. 255

van Leeuwen, P. J. (2015). *Nonlinear data assimilation*. Springer. MR3381734. doi: https://doi.org/10.1007/978-3-319-18347-3. 224

van Leeuwen, P. J. and Evensen, G. (1996). "Data assimilation and inverse methods in terms of a probabilistic formulation." *Monthly Weather Review*, 124(12): 2898–2913. 254

Vats, D., Flegal, J. M., and Jones, G. L. (2019). "Multivariate output analysis for Markov chain Monte Carlo." *Biometrika (online preview)*. MR3949306. doi: https://doi.org/10.1093/biomet/asz002. 236

Vieira, R. and Wilkinson, D. J. (2016). "Online state and parameter estimation in Dynamic Generalised Linear Models." Available from arXiv:1608.08666. 225

Vrettas, M. D., Opper, M., and Cornford, D. (2015). "Variational mean-field algorithm for efficient inference in large systems of stochastic differential equations." *Physical Review E*, 91. MR3416655. doi: https://doi.org/10.1103/PhysRevE.91.012148. 242, 243

Wilkinson, D. J. (2018). *Stochastic modelling for systems biology*. CRC Press. MR2222876. 242

Acknowledgments

CD and DP are grateful to RGE for providing support to present at a Bayesian workshop at the University of Reading, where this research project was instigated. CD was supported by an Australian Research Council Discovery Project (DP200102101). AG was supported by the UK EPSRC grant EP/N510129/1 via the Alan Turing Institute project "Streaming data modelling for real-time monitoring and forecasting". RGE was supported by EPSRC grant EP/N023927/1, and thanks Philip Maybank for the implementation of the solver for the Liley model, and Philip Maybank and Ingo Bojak for invaluable discussions about this model.