Bayesian Analysis (2021) **16**, Number 2, pp. 575–609

Particle Methods for Stochastic Differential Equation Mixed Effects Models

Imke Botha[*,§,¶], Robert Kohn[†,§], and Christopher Drovandi[‡,§]

Abstract. Parameter inference for stochastic differential equation mixed effects models (SDEMEMs) is challenging. Analytical solutions for these models are rarely available, which means that the likelihood is also intractable. In this case, exact inference (up to the discretisation of the stochastic differential equation) is possible using particle MCMC methods. Although the exact posterior is targeted by these methods, a naive implementation for SDEMEMs can be highly inefficient. Our article develops three extensions to the naive approach which exploit specific aspects of SDEMEMs and other advances such as correlated pseudo-marginal methods. We compare these methods on simulated data and data from a tumour xenography study on mice.

Keywords: Bayesian inference, hierarchical models, MCMC, particle Gibbs, pseudo-marginal, random effects.

1 Introduction

Stochastic differential equations (SDEs) are defined as ordinary differential equations (ODEs) with one or more stochastic components. SDEs allow for random variations around the mean dynamics specified by the ODE. These models can be used to capture inherent randomness in the system of interest. For repeated measures data, random effects can be used to account for between-subject variability; this gives an SDE mixed effects model (SDEMEM).

SDEMEMs are emerging as a useful class of models for biomedical and pharmacokinetic/pharmacodynamic data (Donnet et al., 2010; Donnet and Samson, 2013a; Leander et al., 2015). They have also been applied in psychology (Oravecz et al., 2011) and spatio-temporal modelling (Duan et al., 2009). Statistical inference for these models is generally difficult however. In most cases, the SDE does not have an explicit or analytical solution (transition density), making the likelihood intractable. Including random effects adds further complexity.

Parameter inference for SDEMEMs has largely focussed on maximum likelihood estimation; e.g. Picchini et al. (2010), Picchini and Ditlevsen (2011), Delattre et al. (2013)

[*]School of Mathematical Sciences, Queensland University of Technology, Australia, imke.botha@hdr.qut.edu.au

[†]School of Economics, University of New South Wales, r.kohn@unsw.edu.au

[‡]School of Mathematical Sciences, Queensland University of Technology, Australia, c.drovandi@qut.edu.au

[§]ARC Centre of Excellence for Mathematical & Statistical Frontiers (ACEMS)

[¶]IB was supported by an Australian Reseach Training Program Stipend and an ACEMS Top-Up Scholarship.

 https://doi.org/10.1214/20-BA1216

and Donnet and Samson (2013a,b). There are few Bayesian inference methods; Donnet et al. (2010) propose a Gibbs sampler coupled with an Euler-Maruyama discretisation of the intractable transition density. Whitaker et al. (2017a) take a data augmentation approach based on a diffusion bridge, which allows for non-linear dynamics between observed time points. Picchini and Forman (2019) compare results from a particle MCMC algorithm (Andrieu and Roberts, 2009; Andrieu et al., 2010) and a Bayesian synthetic likelihood approach (Wood, 2010; Price et al., 2018). They apply both methods to an SDE with known solution, and suggest an Euler-Maruyama approximation if the solution is unavailable.

It is unlikely however that any one approach to estimating SDEMEMs is optimal for all applications. Performance will depend on the complexity of the underlying SDE, the number of parameters, the number of observations for each subject, as well as the complexity of the random effects. Inference results also depend on the properties of the ODE underlying the SDEMEM, and whether it reflects important features of the data, such as monotonicity or periodicity. It has been our experience that methods that work well on simple examples can often fail badly on more complex ones. This motivates our focus on significant extensions to the pseudo-marginal approach of Picchini and Forman (2019) for SDEMEMs. Pseudo-marginal methods can overcome some limitations of data augmentation approaches because they integrate out the latent states (Stramer and Bognar, 2011; Gunawan et al., 2018b). This is especially useful when there is substantial correlation between the latent variables and the parameters of interest. Our article develops a suite of new and efficient Bayesian methods for SDEMEMs that take advantage of advances in particle methods and exploit specific aspects of SDEMEMs. We compare these methods on a model adapted from one used by Picchini and Forman (2019) to model the growth of tumour volumes in mice. We believe that the results of this comparison are of interest to the wider Bayesian community.

The rest of the paper is organised as follows. Sections 2 and 3 provide the necessary background on state space models, stochastic differential equations, particle filters and particle MCMC methods. Section 4 proposes three potential particle methods for SDEMEMs. Sections 5–7 compare these methods with the Picchini and Forman (2019) approach on simulated and real data from a tumor xenography study on mice, modelled with a monotonic growth SDEMEM. Appendix B (Botha et al., 2020) gives a second example which shows the performance of our methods on a non-monotonic data example. Section 8 concludes with a discussion of the results and possible future work. Code for our methods is available at `https://github.com/imkebotha/particle-mcmc-sdemem`.

2 Stochastic Differential Equation Mixed Effects Models

We denote random variables by capital letters and their realisations by lowercase letters; $\mathbb{N}$ is the set of positive integers. The symbol $\sim$ denotes simulation according to a probability distribution, which refers to either the distribution or density depending on the context. For conciseness, we define $\boldsymbol{x}_{i:j} \coloneqq \{x_i, x_{i+1}, \ldots, x_j\}$ for $j > i$.

2.1 State Space Models

State space models (SSMs) consist of two processes: a Markov process $\{X_t\}_{t\geq 0} \subset \mathcal{X}$, where X_t is usually only partially observed and is often viewed as a latent process, and an observed process $\{Y_t\}_{t\geq 0} \subset \mathcal{Y}$. The $\mathcal{X}$ and $\mathcal{Y}$ spaces are typically subsets of Euclidean space. To obtain an SSM, we assume that $\{(x_t, y_t); t \geq 0\}$ is Markov with model parameters $\boldsymbol{\theta}$, so that

$$\begin{aligned} p(x_t, y_t \mid \boldsymbol{x}_{0:t-1}, \boldsymbol{y}_{0:t-1}, \boldsymbol{\theta}) &= p(x_t, y_t \mid x_{t-1}, y_{t-1}, \boldsymbol{\theta}) \\ &= g(y_t \mid x_t, x_{t-1}, y_{t-1}, \boldsymbol{\theta}) f(x_t \mid x_{t-1}, y_{t-1}, \boldsymbol{\theta}). \end{aligned}$$

For ease of exposition, $t = 0, \ldots, T-1$ are assumed to be the observation times. We simplify further and assume that

$$g(y_t \mid x_t, x_{t-1}, y_{t-1}, \boldsymbol{\theta}) = g(y_t \mid x_t, \boldsymbol{\theta}), \quad f(x_t \mid x_{t-1}, y_{t-1}, \boldsymbol{\theta}) = f(x_t \mid x_{t-1}, \boldsymbol{\theta}),$$

where $g(y_t \mid x_t, \boldsymbol{\theta})$ is the observation density and $f(x_t \mid x_{t-1}, \boldsymbol{\theta})$ is the state transition density; $\pi(\boldsymbol{\theta})$ is the prior for $\boldsymbol{\theta}$. The unnormalized posterior density of the latent states and model parameters is

$$p(\boldsymbol{x}_{0:T-1}, \boldsymbol{\theta} \mid \boldsymbol{y}_{0:T-1}) \propto p(\boldsymbol{y}_{0:T-1} \mid \boldsymbol{x}_{0:T-1}, \boldsymbol{\theta}) p(\boldsymbol{x}_{0:T-1} \mid \boldsymbol{\theta}) \pi(\boldsymbol{\theta}), \tag{1}$$

where

$$p(\boldsymbol{y}_{0:T-1} \mid \boldsymbol{x}_{0:T-1}, \boldsymbol{\theta}) = \prod_{t=0}^{T-1} g(y_t \mid x_t, \boldsymbol{\theta}), \text{ and}$$

$$p(\boldsymbol{x}_{0:T-1} \mid \boldsymbol{\theta}) = \mu(x_0 \mid \boldsymbol{\theta}) \prod_{t=1}^{T-1} f(x_t \mid x_{t-1}, \boldsymbol{\theta}).$$

The marginal posterior for $\boldsymbol{\theta}$ is,

$$p(\boldsymbol{\theta} \mid \boldsymbol{y}_{0:T-1}) \propto \pi(\boldsymbol{\theta}) p(\boldsymbol{y}_{0:T-1} \mid \boldsymbol{\theta}),$$

with likelihood

$$p(\boldsymbol{y}_{0:T-1} \mid \boldsymbol{\theta}) = \int p(\boldsymbol{y}_{0:T-1} \mid \boldsymbol{x}_{0:T-1}, \boldsymbol{\theta}) p(\boldsymbol{x}_{0:T-1} \mid \boldsymbol{\theta}) d\boldsymbol{x}_{0:T-1}. \tag{2}$$

The integral in (2) is usually intractable. For some models, inference is also complicated due to an intractable transition density, see e.g. the SDEs in Section 2.2. While approximate methods can be used in this case, Section 3.2 shows that exact inference is still feasible if it is possible to simulate from the transition density.

2.2 Stochastic Differential Equation Mixed Effects Models

A one-dimensional Itô process (Øksendal, 2013, p. 22) is a stochastic process $\{X_t\}_{t\geq 0}$ satisfying

$$X_t = X_0 + \int_0^t \mu(s, X_t) ds + \int_0^t \sqrt{v}(s, X_t) dB_s, \tag{3}$$

where $B_t = \int_0^t dB_s$ is standard one-dimensional Brownian motion. The differential form of (3) gives the stochastic differential equation (SDE) governing $\{X_t\}_{t\geq 0}$. For simplicity, we only consider one-dimensional SDEs, but it is straightforward to extend the methods introduced in Section 4 to higher dimensions.

Given an Itô process $\{X_t\}_{t\geq 0}$, the general form for a one-dimensional continuous SDE parameterised by $\boldsymbol{\phi_X}$ is

$$dX_t = \mu(X_t, \boldsymbol{\phi_X}, t)dt + \sqrt{v}(X_t, \boldsymbol{\phi_X}, t)dB_t, \quad X_0 = X_0(\boldsymbol{\phi_X}),$$

where $\mu(\cdot)$ is the drift, $\sqrt{v}(\cdot)$ is the diffusion, $\boldsymbol{\phi_X}$ are the fixed model parameters and $\{B_t\}_{t\geq 0}$ is a standard Brownian motion process. This model can be extended by allowing some of the parameters to vary between the $m = 1, \ldots, M$ individuals. In this more general setting, let $\boldsymbol{\phi_X}$ be the vector of fixed common parameters of the SDE, and $\boldsymbol{\eta}_m$ the vector of subject specific parameters (random effects), where $\boldsymbol{\eta}_m \sim p(\boldsymbol{\phi_\eta})$. The stochastic differential equation mixed effects model (SDEMEM) is then given by,

$$dX_{m,t} = \mu(X_{m,t}, \boldsymbol{\phi_X}, \boldsymbol{\eta}_m)dt + \sqrt{v}(X_{m,t}, \boldsymbol{\phi_X}, \boldsymbol{\eta}_m)dB_{m,t}, \quad X_{m0} = X_{m0}(\boldsymbol{\phi_X}, \boldsymbol{\eta}_m). \quad (4)$$

The solution to (4) gives the transition density of the states. If an analytical solution for the transition density is unavailable, numerical methods can be used; Section 2.3 discusses some of these.

This leads to a state-space model if the process shown in (4) is hidden. Let $y_{m,t} \in \{Y_{m,t}\}_{t\geq 0}$ denote a noisy observation for individual $m, m = 1, \ldots, M$ at time $\xi_{m,t}, t = 0, \ldots, T_m - 1$, where T_m is the number of observations for individual m. To simplify notation, we assume that observations are taken at the same time points for all individuals, i.e. $\xi_t, t = 0, \ldots, T-1$, but this restriction is unnecessary for our methods. Further dependence on ξ_t is denoted simply by t, e.g. $0 : T-1$ represents $\xi_0 : \xi_{T-1}$. We assume that the observation equations are given by

$$y_{m,t} \mid x_{m,t}, \sigma^2 \sim \mathcal{N}(y_{m,t}; x_{m,t}, \sigma^2). \quad (5)$$

Let $\boldsymbol{\theta} = (\sigma, \boldsymbol{\phi_X}, \boldsymbol{\phi_\eta})$ be the vector of all unknown parameters in the model, $\boldsymbol{y}_m = \boldsymbol{y}_{m,0:T-1}$ and $\boldsymbol{x}_m = \boldsymbol{x}_{m,0:T-1}$. The joint posterior of $\boldsymbol{\theta}$, $\boldsymbol{\eta}_{1:M}$ and $\boldsymbol{x}_{1:M}$ can be expressed as

$$p(\boldsymbol{\theta}, \boldsymbol{\eta}_{1:M}, \boldsymbol{x}_{1:M} \mid \boldsymbol{y}_{1:M}) \propto p(\boldsymbol{\theta}) \prod_{m=1}^{M} p(\boldsymbol{y}_m \mid \boldsymbol{x}_m, \boldsymbol{\theta}) p(\boldsymbol{x}_m \mid \boldsymbol{\eta}_m, \boldsymbol{\theta}) p(\boldsymbol{\eta}_m \mid \boldsymbol{\theta}).$$

The following running example is used throughout the paper to illustrate some of the concepts and methods.

Example (SDEMEM with constant drift and diffusion)**.** Consider the SDEMEM

$$\begin{aligned} &X_{m,t} = \beta_m dt + \gamma dB_{m,t}, \quad X_{m,0} = x_0, \\ &\log(\beta_m) \sim \mathcal{N}\left(\log(\beta_m); \mu_\beta, \sigma_\beta^2\right), \end{aligned} \quad (6)$$

with random effects $\eta_m = \log(\beta_m)$, unknown static model parameters $\boldsymbol{\phi_X} = \{\gamma, x_0\}$ and random effects hyperparameters $\boldsymbol{\phi_\eta} = \{\mu_\beta, \sigma_\beta\}$. The exact transition density of this model is obtained by solving (6),

$$f(x_{m,t} \mid x_{m,t-1}, \beta_m, \gamma, x_0) = \mathcal{N}(x_{m,t}; x_{m,t-1} + \beta_m, \gamma^2).$$

If a Gaussian observation density is assumed, the full model is given by

$$\begin{cases} g(y_{m,t} \mid x_{m,t}, \boldsymbol{\theta}) = \mathcal{N}(y_{m,t}; x_{m,t}, \sigma^2), \\ f(x_{m,t} \mid x_{m,t-1}, \eta_m, \boldsymbol{\theta}) = \mathcal{N}(x_{m,t}; x_{m,t-1} + \beta_m, \gamma^2), \\ p(\eta_m \mid \boldsymbol{\theta}) = \mathcal{N}(\eta_m; \mu_\beta, \sigma_\beta^2), \end{cases} \tag{7}$$

where $\boldsymbol{\theta} = \{\sigma, \gamma, x_0, \mu_\beta, \sigma_\beta\}$.

2.3 SDE Simulation

Consider the SDEMEM for a single individual,

$$dX_t = \mu(X_t, \boldsymbol{\phi_X}, \boldsymbol{\eta})dt + \sqrt{v}(X_t, \boldsymbol{\phi_X}, \boldsymbol{\eta})dB_t, \quad X_0 = X_0(\boldsymbol{\phi_X}, \boldsymbol{\eta}).$$

If the SDE cannot be solved analytically, then it is necessary to use approximate methods. This section describes two common approaches for approximate simulation of SDEs: the Euler-Maruyama discretisation (EMD) and the diffusion bridge approach. Both methods simulate the SDE between discrete time points (generally corresponding to the observed times) along the entire diffusion trajectory or path, i.e. from $t = 0$ to $T-1$. The error resulting from the discretisation is controlled using data augmentation, which introduces additional (unobserved) states between observation times.

Given a process $\{X_t\}_{t \geq 0}$, the time interval $[\xi_t, \xi_{t+1}]$ between two observations is split into D subintervals, where D denotes the level of discretisation,

$$\xi_t = \tau_0 < \tau_1 < \cdots < \tau_k < \tau_{k+1} < \cdots < \tau_D = \xi_{t+1}, \quad \Delta\tau = \frac{\xi_{t+1} - \xi_t}{D}.$$

The EMD and diffusion bridges simulate over each subinterval as follows

$$X_{\tau_{k+1}} = X_{\tau_k} + \mu_{\mathrm{DB}}(\cdot)\Delta\tau + \sqrt{\Psi_{\mathrm{DB}}(\cdot)}\Delta B_{\tau_k},$$

where $\mu_{\mathrm{DB}}(\cdot)$ and $\Psi_{\mathrm{DB}}(\cdot)$ are determined by the method used, and $\Delta B_{\tau_k} = B_{\tau_{k+1}} - B_{\tau_k}$. Since $\Delta B_{\tau_k} \sim \mathcal{N}(\Delta B_{\tau_k}; 0, \Delta\tau)$ by definition, the path is simulated by recursively applying

$$x_{\tau_{k+1}} \mid x_{\tau_k} \sim \mathcal{N}(x_{\tau_{k+1}}; x_{\tau_k} + \mu_{\mathrm{DB}}(\cdot)\Delta\tau, \Psi_{\mathrm{DB}}(\cdot)\Delta\tau). \tag{8}$$

The joint density of this approximation is

$$q(x_{\tau_1:\tau_D} \mid x_{\tau_0}, \phi_X, \eta) \propto \prod_{k=0}^{D-1} \mathcal{N}(x_{\tau_{k+1}}; x_{\tau_k} + \mu_{\mathrm{DB}}(\cdot)\Delta\tau, \Psi_{\mathrm{DB}}(\cdot)\Delta\tau).$$

Euler-Maruyama

The Euler-Maruyama discretisation (EMD) is the simplest method to simulate an approximate trajectory from an SDE. Assuming that the drift and diffusion coefficients are locally constant,

$$\mu(X_{\tau_k}, \boldsymbol{\phi}_{\boldsymbol{X}}, \boldsymbol{\eta}) = \mu_k, \quad \sqrt{v}(X_{\tau_k}, \boldsymbol{\phi}_{\boldsymbol{X}}, \boldsymbol{\eta}) = \sqrt{v_k},$$

the EMD uses the proposal

$$x_{\tau_{k+1}} \mid x_{\tau_k} \sim \mathcal{N}(x_{\tau_{k+1}}; x_{\tau_k} + \mu_k \Delta\tau, v_k \Delta\tau), \tag{9}$$

which approximates the transition density $f(x_{\tau_{k+1}} \mid x_{\tau_k}, \boldsymbol{\phi}_{\boldsymbol{X}}, \boldsymbol{\eta})$. If the SDE has constant drift and diffusion, then the EMD gives the exact solution, i.e. the transition density.

Example (SDEMEM with constant drift and diffusion)**.** The EMD for the SDEMEM in (6), with $\mu_k = \beta_m$ and $v_k = \gamma^2$, is

$$\begin{aligned} x_{m,\tau_{k+1}} \mid x_{m,\tau_k} &\sim \mathcal{N}(x_{m,\tau_{k+1}}; x_{m,\tau_k} + \beta_m \Delta\tau, \gamma^2 \Delta\tau), \\ x_{m,\tau_{k+1}} \mid x_{m,\tau_k} &\sim \mathcal{N}(x_{m,\tau_{k+1}}; x_{m,\tau_k} + \beta_m, \gamma^2), \quad \Delta\tau = 1, \end{aligned}$$

which is the exact transition density.

Diffusion Bridges

Simulating from the (approximate) transition density may be sub-optimal if some observations are highly informative or there is little observation noise. More effective trajectories can be obtained if the proposal for x_t can be directed towards y_t. This is possible using a diffusion bridge.

The modified diffusion bridge (MDB) of Durham and Gallant (2002) (see also Golightly and Wilkinson, 2008) directs x_t linearly towards y_t. The MDB is derived by approximating the joint distribution of $X_{\tau_{k+1}}, Y_{\xi_{t+1}} \mid x_{\tau_k}$ using a multivariate normal distribution, and then conditioning on $Y_{\xi_{t+1}} = y_{\xi_{t+1}}$. The distribution of $X_{\tau_{k+1}}, Y_{\xi_{t+1}} \mid x_{\tau_k}$ is obtained from the observation density (5) and the EMD (9) of $X_{\tau_{k+1}} \mid x_{\tau_k}$; see Appendix 1 of Golightly and Wilkinson (2008) for a more detailed derivation. The MDB is a bridge proposal of the form

$$x_{\tau_{k+1}} \mid x_{\tau_k}, y_{\xi_{t+1}} \sim \mathcal{N}\{x_{\tau_{k+1}}; x_{\tau_k} + \mu_{\mathrm{MDB}}(x_{\tau_k}, y_{\xi_{t+1}})\Delta\tau, \Psi_{\mathrm{MDB}}(x_{\tau_k})\Delta\tau\},$$

where

$$\begin{aligned} \mu_{\mathrm{MDB}}(x_{\tau_k}, y_{\xi_{t+1}}) &= \mu_k + \frac{v_k(y_{\xi_{t+1}} - (x_{\tau_k} + \mu_k \Delta_k))}{v_k \Delta_k + \sigma^2} &&= \frac{\mu_k \sigma^2 + v_k(y_{\xi_{t+1}} - x_{\tau_k})}{v_k \Delta_k + \sigma^2}, \\ \Psi_{\mathrm{MDB}}(x_{\tau_k}) &= v_k - \frac{v_k^2 \Delta\tau}{v_k \Delta_k + \sigma^2} &&= \frac{v_k \sigma^2 + v_k^2(\Delta_k - \Delta\tau)}{v_k \Delta_k + \sigma^2}, \end{aligned}$$

and $\Delta_k = \xi_{t+1} - \tau_k$.

Whitaker et al. (2017b) notes that the modified diffusion bridge can perform poorly when the drift coefficient ($\mu(X_{\tau_k}, \boldsymbol{\phi_X}, \boldsymbol{\eta}) = \mu_k$) is not approximately constant. To overcome this problem, they propose partitioning the SDE into a deterministic process and a residual stochastic process, such that the latter has constant drift. Rewriting the model in terms of these processes gives

$$\begin{aligned} X_t &= \zeta_t + R_t, & \zeta_t, t \geq 0, \\ d\zeta_t &= f(\zeta_t)dt, & \zeta_0 = x_0, \\ dR_t &= \{\mu(X_t, \boldsymbol{\phi_X}, \boldsymbol{\eta}) - f(\zeta_t)\}dt + \sqrt{v}(X_t, \boldsymbol{\phi_X}, \boldsymbol{\eta})dB_t, & R_0 = 0. \end{aligned} \tag{10}$$

The idea is to choose ζ_t and $f(\cdot)$ such that the drift of (10) is approximately constant. The simplest solution (Whitaker et al., 2017b) is to set $\zeta_t = \pi_t$ and $f(\cdot) = \mu(\cdot)$ as

$$\begin{aligned} X_t &= \pi_t + R_t, & \pi_t, t \geq 0, \\ d\pi_t &= \mu(\pi_t, \boldsymbol{\phi_X}, \boldsymbol{\eta})dt, & \pi_0 = x_0, \\ dR_t &= \{\mu(X_t, \boldsymbol{\phi_X}, \boldsymbol{\eta}) - \mu(\pi_t, \boldsymbol{\phi_X}, \boldsymbol{\eta})\}dt + \sqrt{v}(X_t, \boldsymbol{\phi_X}, \boldsymbol{\eta})dB_t, & R_0 = 0 \end{aligned}$$

noting that $Y_{\xi_{t+1}} - \pi_{\xi_{t+1}} = R_{\xi_{t+1}} + \epsilon_{\xi_{t+1}}$. The residual bridge is obtained by constructing the MDB on the residual process $\{R_t\}$. This bridge proposal is

$$x_{\tau_{k+1}} \mid x_{\tau_k}, y_{\xi_{t+1}} \sim \mathcal{N}(x_{\tau_{k+1}}; x_{\tau_k} + \mu_{\text{RB}}(x_{\tau_k}, y_{\xi_{t+1}})\Delta\tau, \Psi_{\text{RB}}(x_{\tau_k}, y_{\xi_{t+1}})\Delta\tau),$$

where

$$\begin{aligned} \Psi_{\text{RB}}(x_{\tau_k}, y_{\xi_{t+1}}) &= \Psi_{\text{MDB}}(x_{\tau_k}, y_{\xi_{t+1}}), \quad \delta_k^{\pi} = \frac{\pi_{\tau_{k+1}} - \pi_{\tau_k}}{\Delta\tau} \quad \text{and} \\ \mu_{\text{RB}}(x_{\tau_k}, y_{\xi_{t+1}}) &= \mu_k + \frac{v_k(y_{\xi_{t+1}} - (\pi_{\xi_{t+1}} + r_{\tau_k} + (\mu_k - \delta_k^{\pi})\Delta k))}{v_k\Delta_k + \sigma^2}. \end{aligned}$$

3 Particle MCMC

There are at least two possible Bayesian strategies to obtain parameter inference for $\boldsymbol{\theta}$ for state-space SDEMEMs. The first is to use a Metropolis-Hastings algorithm to sample from $\boldsymbol{\theta} \mid \boldsymbol{y}_{1:M}$. This requires the likelihood $p(\boldsymbol{\theta} \mid \boldsymbol{y}_{1:M})$ to be analytically tractable, which is generally not the case for SDEMEMs. The second strategy involves a component-wise sampling approach, which iteratively updates $\boldsymbol{\theta} \mid \boldsymbol{y}_{1:M}, \boldsymbol{\eta}_{1:M}, \boldsymbol{x}_{1:M}$, $\boldsymbol{\eta}_{1:M} \mid \boldsymbol{y}_{1:M}, \boldsymbol{\theta}, \boldsymbol{x}_{1:M}$ and $\boldsymbol{x}_{1:M} \mid \boldsymbol{y}_{1:M}, \boldsymbol{\theta}, \boldsymbol{\eta}_{1:M}$. The distribution of $\boldsymbol{x}_m \mid \boldsymbol{y}_{1:M}, \boldsymbol{\theta}, \boldsymbol{\eta}_{1:M}$ can be complicated however, especially if the exact transition density is unknown. Inefficiency can also be an issue if there is high correlation between $\boldsymbol{x}_{1:M}$ and either $\boldsymbol{\theta}$ or $\boldsymbol{\eta}_{1:M}$.

In both cases, a particle filter may be used to overcome these issues. The following sections describe particle filters, particle methods and their variants for state-space models. Section 4 covers the specific application of these methods to SDEMEMs.

Algorithm 1: The generic particle filter of Doucet and Johansen (2009).

Input : data $\boldsymbol{y}_{1:T}$, number of particles N, static parameters $\boldsymbol{\theta}$, initial states $\boldsymbol{x}_0^{1:N}$ and the vector of random numbers $\boldsymbol{u}^{\dagger}$.

Output : weighted sample $\{\boldsymbol{x}_{1:T}^{1:N}, \boldsymbol{W}_{1:T}^{1:N}\}$, likelihood estimate $\widehat{Z}$

Notation: We use the convention that index (n) means 'for all $n \in \{1, \ldots, N\}$'

1 Initialise $x_1^{(n)} = x_0^{(n)}$, $W_1^{(n)} = \frac{1}{N}$, $w_1^{(n)} = \frac{1}{N}\pi_1(x_1^{(n)}, y_1 \mid \boldsymbol{\theta})$, $\widehat{Z} = \sum_{i=1}^{N} w_1^i$

2 **for** $t = 2$ *to* T **do**

3 Resample (with replacement) N particles from $\boldsymbol{x}_{t-1}^{1:N}$ according to $\boldsymbol{W}_{t-1}^{1:N}$

4 Move the particles, $x_t^{(n)} \sim q(x_t^{(n)} \mid y_t, x_{t-1}^{(n)}, \boldsymbol{\theta})$

5 Calculate weights $w_t^{(n)} = \frac{\pi_t(x_t^{(n)}, y_t | x_{t-1}^{(n)} \boldsymbol{\theta})}{N \cdot q(x_t^{(n)} | y_t, x_{t-1}^{(n)}, \boldsymbol{\theta})}$

6 Normalize weights $W_t^{(n)} = \frac{w_t^{(n)}}{\sum_{i=1}^{N} w_t^i}$

7 Update likelihood estimate $\widehat{Z} = \widehat{Z} \times \sum_{i=1}^{N} w_t^i$

8 **end**

†Note that the random numbers in $\boldsymbol{u}$ are used implicitly when the particles are resampled and moved (steps 3 and 4).

3.1 Particle Filters

Exact state estimation of SSMs using the Kalman filter is only possible when they are Gaussian or conditionally Gaussian. For non-linear, non-Gaussian SSMs, a particle filter can be used for simulation consistent estimation (Gordon et al., 1993; Carpenter et al., 1999; Doucet et al., 2000; Del Moral et al., 2006; Doucet and Johansen, 2009).

Particle filters are used to traverse through a sequence of intermediary distributions towards some target distribution. We describe the generic particle filter of Doucet and Johansen (2009) (see Algorithm 1), with a filtering distribution of the form

$$p_t(\boldsymbol{x}_{1:t} \mid \boldsymbol{y}_{1:t}, \boldsymbol{\theta}) \propto \pi_1(x_1, y_1 \mid \boldsymbol{\theta}) \prod_{j=2}^{t} \pi_t(x_j, y_j \mid x_{j-1}, \boldsymbol{\theta}), \quad t = 1, \ldots, T$$

$$= g(y_1|x_1, \boldsymbol{\theta}) f(x_1 \mid \boldsymbol{\theta}) \prod_{j=2}^{t} g(y_j|x_j, \boldsymbol{\theta}) f(x_j \mid x_{j-1}, \boldsymbol{\theta}). \tag{11}$$

A combination of move, reweight and resample steps are used to transition through this sequence. The move step generates values for x_t from some proposal distribution $q(x_t \mid y_t, x_{t-1}, \boldsymbol{\theta})$. Once moved, the N particles are re-weighted according to,

$$w_t^n = W_{t-1}^n \frac{\pi_t(x_t, y_t \mid x_{t-1} \boldsymbol{\theta})}{q(x_t \mid y_t, x_{t-1}, \boldsymbol{\theta})}, \quad W_t^n = \frac{w_t^n}{\sum_{i=1}^{N} w_t^i}.$$

Particles are then resampled with probability $\boldsymbol{W}_t^{1:N}$ for the next iteration. This is done to avoid particle impoverishment, where most of the weight is given to few particles.

There are several resampling methods that can be used, including multinomial, stratified (Kitagawa, 1996), and more recently, the Srinivasan sampling process (Gerber et al., 2019).

The particle filter gives an unbiased estimate of the likelihood from the unnormalized weights,

$$\widehat{p}(\boldsymbol{Y}_{1:T} \mid \boldsymbol{\theta}) = \prod_{t=1}^{T} \sum_{n=1}^{N} w_t^{(n)}. \tag{12}$$

The bootstrap particle filter of Gordon et al. (1993) is a special case of the generic particle filter with $q(x_t \mid y_t, x_{t-1}, \boldsymbol{\theta}) = f(x_t \mid x_{t-1}, \boldsymbol{\theta})$. The calculation of the weights then simplifies to $w_t^n = W_{t-1}^n g(y_t \mid x_t, \boldsymbol{\theta})$.

3.2 Pseudo-Marginal MCMC

Markov chain Monte Carlo (MCMC) algorithms work by constructing an ergodic Markov chain with the posterior distribution $p(\boldsymbol{\theta} \mid \boldsymbol{y})$ as its stationary distribution. The Metropolis-Hastings (MH) algorithm is a standard MCMC method, which accepts candidate values $\boldsymbol{\theta}^* \sim q(\boldsymbol{\theta}^* \mid \boldsymbol{\theta})$ with probability

$$\alpha = \min\left(1, \frac{p(\boldsymbol{y} \mid \boldsymbol{\theta}^*)p(\boldsymbol{\theta}^*)}{p(\boldsymbol{y} \mid \boldsymbol{\theta})p(\boldsymbol{\theta})} \frac{q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^*)}{q(\boldsymbol{\theta}^* \mid \boldsymbol{\theta})}\right). \tag{13}$$

If the likelihood function $p(\boldsymbol{y} \mid \boldsymbol{\theta})$ is intractable then the MH algorithm cannot be implemented directly, however pseudo-marginal MCMC can be used.

The pseudo-marginal approach of Andrieu and Roberts (2009) allows for exact inference for models with intractable likelihoods. In this approach, the intractable likelihood $p(\boldsymbol{y}_{1:T} \mid \boldsymbol{\theta})$ is replaced with a non-negative unbiased estimate within an otherwise standard MH algorithm. The likelihood estimate is written interchangeably as $\widehat{p}(\boldsymbol{y}_{1:T} \mid \boldsymbol{\theta}) = p(\boldsymbol{y}_{1:T} \mid \boldsymbol{\theta}, \boldsymbol{u})$ where $\boldsymbol{u} \sim p(\boldsymbol{u})$ are the auxiliary variables used to construct the likelihood estimate. Unbiasedness means that

$$\mathbb{E}_{p(\boldsymbol{u})}(\widehat{p}(\boldsymbol{y}_{1:T} \mid \boldsymbol{\theta})) = \mathbb{E}_{p(\boldsymbol{u})}(p(\boldsymbol{y}_{1:T} \mid \boldsymbol{\theta}, \boldsymbol{u})) = \int p(\boldsymbol{y}_{1:T} \mid \boldsymbol{\theta}, \boldsymbol{u})p(\boldsymbol{u})d\boldsymbol{u} = p(\boldsymbol{y}_{1:T} \mid \boldsymbol{\theta}).$$

Pseudo-marginal MCMC can therefore be defined as a standard MH algorithm on an augmented space, i.e. the space of $\boldsymbol{\theta}$ augmented with the auxiliary variables $\boldsymbol{u}$. The MCMC chain targets $p(\boldsymbol{\theta}, \boldsymbol{u} \mid \boldsymbol{y}_{1:T})$ which has the posterior $p(\boldsymbol{\theta} \mid \boldsymbol{y}_{1:T})$ as its θ-marginal distribution, since

$$\begin{aligned}
\int p(\boldsymbol{\theta}, \boldsymbol{u} \mid \boldsymbol{y}_{1:T})d\boldsymbol{u} &= \int \frac{p(\boldsymbol{y}_{1:T} \mid \boldsymbol{\theta}, \boldsymbol{u})p(\boldsymbol{\theta})p(\boldsymbol{u})}{p(\boldsymbol{y}_{1:T})}d\boldsymbol{u} \\
&= \frac{p(\boldsymbol{\theta})}{p(\boldsymbol{y}_{1:T})} \int p(\boldsymbol{y}_{1:T} \mid \boldsymbol{\theta}, \boldsymbol{u})p(\boldsymbol{u})d\boldsymbol{u} \\
&= \frac{p(\boldsymbol{\theta})p(\boldsymbol{y}_{1:T} \mid \boldsymbol{\theta})}{p(\boldsymbol{y}_{1:T})} = p(\boldsymbol{\theta} \mid \boldsymbol{y}_{1:T}).
\end{aligned}$$

The next sections describe the particle marginal Metropolis-Hastings (PMMH) and particle Gibbs (PG) algorithms proposed by Andrieu et al. (2010).

Particle Marginal Metropolis-Hastings

The PMMH algorithm is a pseudo-marginal method where the intractable likelihood is replaced by an unbiased particle filter estimate (12). As in Section 3.2, the resulting chain targets the joint density $p(\boldsymbol{\theta}, \boldsymbol{u} \mid \boldsymbol{y}_{1:T})$, where $\boldsymbol{u}$ is the vector of random numbers used in the particle filter; $\boldsymbol{u}$ contains all the random numbers required to resample and move the particles (steps 3 and 4 of Algorithm 1) and its elements are usually standard uniform or standard normal.

Algorithm 2: Particle marginal Metropolis-Hastings.

Input : data $\boldsymbol{y}_{1:T}$, initial values $\boldsymbol{\theta}^0$ and number of iterations I
Output : posterior samples $\boldsymbol{\theta}^{1:I}$

Initialise $\boldsymbol{\theta}^1 = \boldsymbol{\theta}^0$
Draw $\boldsymbol{u} \sim p(\cdot)$
Run Algorithm 1 to obtain an unbiased estimate of $p(\boldsymbol{y}_{1:T} \mid \boldsymbol{\theta}^1, \boldsymbol{u})$
for $i = 2$ *to* I **do**
  Sample $\boldsymbol{\theta}^* \sim q(\cdot \mid \boldsymbol{\theta}^{i-1})$ and $\boldsymbol{u}^* \sim p(\cdot)$
  Run Algorithm 1 to obtain an unbiased estimate of $p(\boldsymbol{y}_{1:T} \mid \boldsymbol{\theta}^*, \boldsymbol{u}^*)$
  Calculate the acceptance probability

$$\alpha = \min\left(1, \frac{p\left(\boldsymbol{y}_{1:T} \mid \boldsymbol{\theta}^*, \boldsymbol{u}^*\right) p\left(\boldsymbol{\theta}^*\right)}{p\left(\boldsymbol{y}_{1:T} \mid \boldsymbol{\theta}^{i-1}, \boldsymbol{u}\right) p\left(\boldsymbol{\theta}^{i-1}\right)} \frac{q\left(\boldsymbol{\theta}^{i-1} \mid \boldsymbol{\theta}^*\right)}{q\left(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}^{i-1}\right)}\right)$$

  Draw $u \sim \mathcal{U}(0,1)$
  if $u < \alpha$ **then**
    Set $\boldsymbol{\theta}^i = \boldsymbol{\theta}^*$ and $\boldsymbol{u} = \boldsymbol{u}^*$
  else
    Set $\boldsymbol{\theta}^i = \boldsymbol{\theta}^{i-1}$
  end
end

A drawback of the PMMH algorithm is that it can be difficult to find good proposals. Another drawback is the chain's tendency to get stuck whenever the likelihood is greatly overestimated for a particular value of $\boldsymbol{\theta}$, i.e. if $\widehat{p}(\boldsymbol{y}_{1:T} \mid \boldsymbol{\theta})$ is greatly overestimated, then the acceptance probability for $\boldsymbol{\theta}^*$ will be small unless $\widehat{p}(\boldsymbol{y}_{1:T} \mid \boldsymbol{\theta}^*)$ is also overestimated. This issue is mitigated by decreasing the variance of the log of the ratio of the likelihood estimates

$$R = \log\left(\frac{p(\boldsymbol{y}_{1:T} \mid \boldsymbol{\theta}^*, \boldsymbol{u}^*)}{p(\boldsymbol{y}_{1:T} \mid \boldsymbol{\theta}, \boldsymbol{u})}\right). \tag{14}$$

A common strategy to do this is to increase the number of particles N used in the particle filter. Since the estimates are unbiased, this increases both the precision and accuracy of the individual likelihood estimates. Sherlock et al. (2015), Pitt et al. (2012) and Doucet et al. (2015) showed that optimal performance is gained when N is chosen

such that the standard deviation of the estimated log-likelihood is between 1 and 4. A superior alternative approach is the correlated pseudo-marginal (CPM) method of Deligiannidis et al. (2018) (see also Dahlin et al. (2015)). Tran et al. (2016) introduced a variation of the CPM method called the block pseudo-marginal (BPM) approach.

Correlated Pseudo-Marginal

At any given iteration of the PMMH algorithm (Algorithm 2), the likelihood ratio is $p(\boldsymbol{y}_{1:T} \mid \boldsymbol{\theta}^*, \boldsymbol{u}^*)/p(\boldsymbol{y}_{1:T} \mid \boldsymbol{\theta}, \boldsymbol{u})$, where $\boldsymbol{\theta}^*$ and $\boldsymbol{u}^*$ are the proposed values and $\boldsymbol{\theta}$ and $\boldsymbol{u}$ are the current values. Deligiannidis et al. (2018) show that the mixing of the chain is greatly improved by correlating $p(\boldsymbol{y}_{1:T} \mid \boldsymbol{\theta}^*, \boldsymbol{u}^*)$ and $p(\boldsymbol{y}_{1:T} \mid \boldsymbol{\theta}, \boldsymbol{u})$. This helps to vastly reduce the variance of the log of the ratio of the estimated likelihoods (14), without having to reduce the variance of each of the individual likelihood estimates.

The correlated pseudo-marginal (CPM) approach does this by making $\boldsymbol{u}$ and $\boldsymbol{u}^*$ highly correlated. Assuming the random numbers are normally distributed, Deligiannidis et al. (2018) use the Crank-Nicolson proposal to induce a correlation of ρ

$$\begin{aligned} q_{\boldsymbol{\theta},\boldsymbol{u}}(\{\boldsymbol{\theta}^*, \boldsymbol{u}^*\} \mid \{\boldsymbol{\theta}, \boldsymbol{u}\}) &= q_{\boldsymbol{\theta}}(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}) q_{\boldsymbol{u}}(\boldsymbol{u}^* \mid \boldsymbol{u}) \\ &= q_{\boldsymbol{\theta}}(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}) \mathcal{N}(\boldsymbol{u}^*; \rho u, \quad (1-\rho^2)\boldsymbol{I}_{\boldsymbol{N_u}}). \end{aligned}$$

If the particle filter depends on non-normal random numbers, transformation to normality is applied. Deligiannidis et al. (2018) derive some optimality results to tune the parameters of the CPM method. In particular, they use the variance of the estimated log-likelihood ratio around the mode of the posterior to tune ρ for a given number of particles N.

The block pseudo-marginal (BPM) approach induces correlation by updating $\boldsymbol{u}$ in blocks or subsets (Tran et al., 2016); the vector of random numbers $\boldsymbol{u}$ is divided into B blocks, and a single block is updated at each iteration while the remaining $B-1$ are held constant. The resulting correlation between the logs of the likelihood estimates is approximately $1-1/B$ and is induced much more directly than in CPM. No assumption about the form or distribution of $\boldsymbol{u}$ is required. Tran et al. (2016) uses the variance of the log-likelihood estimator to tune N for each group or block. Given that B is sufficiently large, they derive the optimal variance for both Monte Carlo and randomised quasi-Monte Carlo (RQMC) log-likelihood estimators.

Relative to standard PMMH, CPM and BPM are able to tolerate significantly more variance in the log-likelihood estimates, such that less particles are needed for the chain to mix well. The increase in computational efficiency gained from this typically outweighs the overhead associated with storing the vector of random numbers $\boldsymbol{u}$.

Conditional Particle Filter

The particle Gibbs (PG) algorithm of Andrieu et al. (2010), requires a variation of the generic particle filter (Section 3.1) called the conditional particle filter (CPF). The CPF differs from the generic particle filter by holding a single path $\boldsymbol{x}_{1:T}^k$ invariant throughout the iterations.

The ancestral lineage $\boldsymbol{B}^k_{1:T}$ of the invariant path contains the index of each state $x^k_t \in \boldsymbol{x}^k_{1:T}$ relative to all other states $\boldsymbol{x}^{1:N}_t$ for $t = 1, \ldots, T$. For example, if $N = 4$ and $B^k_{1:3} = \{4, 1, 2\}$, then

$$\begin{aligned}
\boldsymbol{x}^{1:N}_{t=1} &= \{x^1_1, x^2_1, x^3_1, x^k_1\}^\top, \\
\boldsymbol{x}^{1:N}_{t=2} &= \{x^k_2, x^2_2, x^3_2, x^4_2\}^\top, \\
\boldsymbol{x}^{1:N}_{t=3} &= \{x^1_3, x^k_3, x^3_3, x^4_3\}^\top.
\end{aligned}$$

Once a weighted sample is obtained, a new invariant path and associated ancestral lineage may be drawn using the backwards sampling method of Whiteley (2010) and Lindsten and Schön (2012). See Algorithms 3 and 4 for more details.

Algorithm 3: The conditional particle filter.

Input : data $\boldsymbol{y}_{1:T}$, number of particles N, initial states $x^{1:N}_0$, static parameters $\boldsymbol{\theta}$, invariant path $\boldsymbol{x}^k_{1:T}$ and associated ancestral lineage $\boldsymbol{B}^k_{1:T}$.

Output : new path $\boldsymbol{x}^k_{1:T}$ and associated ancestral lineage $\boldsymbol{B}^k_{1:T}$

Notation: We use the convention that index (n) means 'for all $n \in \{1, \ldots, N\}$' and index $(n \neq k)$ means 'for all $n \in \{1, \ldots, k-1, k+1, \ldots, N\}$'

Initialise $x^{(n \neq B^k_1)}_1 = x^{(n \neq B^k_1)}_0$, $W^{(n)}_1 = \frac{1}{N}, w^{(n)}_1 = \frac{1}{N}\pi_1(x^{(n)}_1, y_1 \mid \boldsymbol{\theta})$, $\widehat{Z} = \sum_{n=1}^N w^{(n)}_1$

Set $\boldsymbol{x}^{\boldsymbol{B}^k_{1:T}}_{1:T} = \boldsymbol{x}^k_{1:T}$

for $t = 2$ *to* T **do**

  Sample parent indices $A^{(n \neq B^k_t)}_{t-1} \sim \mathcal{F}(\cdot \mid W^{(n)}_{t-1})$ `/* resample */`

  Sample $x^{(n \neq B^k_t)}_t \sim q(\cdot \mid x^{A^{(n \neq B^k_t)}_{t-1}}_{t-1}, \boldsymbol{\theta})$ `/* move */`

  Calculate weights $w^{(n)}_t = \frac{\pi_t(x^{(n)}_t, y_t \mid x^{(n)}_{t-1}, \boldsymbol{\theta})}{N \cdot q(x^{(n)}_t \mid x^{(n)}_{t-1}, \boldsymbol{\theta})}$

  Normalize weights $W^{(n)}_t = \frac{w^{(n)}_t}{\sum_{i=1}^N w^i_t}$

end

Run Algorithm 4 to obtain new ancestral lineage $\boldsymbol{B}^k_{1:T}$

Use $\boldsymbol{B}^k_{1:T}$ to determine new path $\boldsymbol{x}^k_{1:T}$

The matrix A^n_{t-1} gives the parent indices of the particles at time $t-1$. The relationship between the ancestral lineage and the matrix of parent indices is $A^{B^k_t}_{t-1} = B^k_{t-1}$, where $B^k_T = k$.

Particle Gibbs

PMMH uses the unbiased estimate of the likelihood computed by the particle filter. In particle Gibbs, the latent states are updated using a conditional particle filter, i.e. $\boldsymbol{x}_{1:T}$ is approximately sampled from $p(\boldsymbol{x}^*{}_{1:T} \mid \boldsymbol{y}_{1:T}, \boldsymbol{x}_{1:T}, \boldsymbol{\theta})$ (see Algorithms 3 and 5). The parameters $\boldsymbol{\theta}$ are updated using Gibbs sampling if the full conditional posterior is available, or a Metropolis-Hastings step otherwise.

Algorithm 4: Backward Sampling.

Input : particles $\boldsymbol{x}_{1:T}^{1:N}$, particle weights $\boldsymbol{w}_{1:T}^{1:N}$ and normalised particle weights $\boldsymbol{W}_{T}^{1:N}$

Output : new ancestral lineage $\boldsymbol{B}_{1:T}$

Notation: We use the convention that index (n) means 'for all $n \in \{1, \ldots, N\}$'

Draw $\boldsymbol{k} \sim \mathcal{F}(\cdot \mid \boldsymbol{W}_{T}^{1:N})$

Set $\boldsymbol{B}_T = \boldsymbol{k}$

for $t = T-1$ *to* 1 **do**

  Sample $W_{(t|T)}^{(n)} = w_t^{(n)} \frac{f_{\boldsymbol{\theta}}\left(x_{t+1}^{B_{t+1}} | x_t^{(n)}\right)}{\sum_{i=1}^{N} w_t^i f_{\boldsymbol{\theta}}\left(x_{t+1}^{B_{t+1}} | x_t^i\right)}$

  Draw $\boldsymbol{B}_t \sim \mathcal{F}(\cdot \mid \boldsymbol{W}_{(t|T)}^{1:N})$

end

Since a new path $\boldsymbol{x}_{1:T}$ is simulated at each iteration, PG does not suffer from the same mixing problem as PMMH; it is significantly less sensitive to the number of particles used. PG also has the advantage that more efficient updating schemes for $\boldsymbol{\theta}$ can be used, such as the Metropolis-adjusted Langevin algorithm (MALA) or Hamiltonian Monte Carlo (HMC). While this method has a number of advantages over PMMH, it is not as general as it requires a closed form transition density to update $\boldsymbol{\theta}$.

Algorithm 5: The particle Gibbs algorithm.

Input : data $\boldsymbol{y}_{1:T}$, initial values $\boldsymbol{\theta}^0$, initial path $\boldsymbol{x}_{1:T}^0$ and associated ancestral lineage $\boldsymbol{B}_{1:T}^0$, and number of iterations I

Output : posterior samples $\boldsymbol{\theta}^{1:I}$ and $\boldsymbol{x}_{1:T}^{1:I}$ with associated ancestral lineage $\boldsymbol{B}_{1:T}^{1:I}$

Initialise $\boldsymbol{\theta}^1 = \boldsymbol{\theta}^0$ and $\boldsymbol{x}_{1:T}^1 = \boldsymbol{x}_{1:T}^0$ and $\boldsymbol{B}_{1:T}^1 = \boldsymbol{B}_{1:T}^0$

for $i = 1$ *to* $I-1$ **do**

  Update $\boldsymbol{\theta}^{i+1}$ conditional on $\boldsymbol{\theta}^i$ and $\boldsymbol{x}_{1:T}^i$

  Run Algorithm 3 to sample $\boldsymbol{x}_{1:T}^{i+1}$ and $\boldsymbol{B}_{1:T}^{i+1}$ conditional on $\boldsymbol{\theta}^{i+1}, \boldsymbol{x}_{1:T}^i$ and $\boldsymbol{B}_{1:T}^i$.

end

4 Methods

We are interested in parameter inference for the state-space SDEMEM described in Section 2.2. For a single individual m, with observations taken at $t = 0, \ldots, T-1$, the sequence of distributions (11) traversed by the particle filter (see Section 3.1) is

$$\pi_t(\boldsymbol{x}_{m,0:t} \mid \boldsymbol{y}_{m,0:t}, \boldsymbol{\eta}_m, \sigma, \boldsymbol{\phi}_{\boldsymbol{X}}) \propto g(y_{m,0}|x_{m,0}, \sigma) f(x_{m,0} \mid \boldsymbol{\eta}_m, \boldsymbol{\phi}_{\boldsymbol{X}}) \prod_{j=1}^{t} g(y_{m,j}|x_{m,j}, \sigma) f(x_{m,j} \mid x_{m,j-1}, \boldsymbol{\eta}_m, \boldsymbol{\phi}_{\boldsymbol{X}}).$$

This particle filter returns an estimate of $p(\boldsymbol{y}_m \mid \boldsymbol{\eta}_m, \sigma, \boldsymbol{\phi}_{\boldsymbol{X}})$, which can be used to estimate the conditional likelihood for all individuals,

$$\widehat{p}(\boldsymbol{y}_{1:M} \mid \boldsymbol{\eta}_{1:M}, \sigma, \boldsymbol{\phi}_{\boldsymbol{X}}) = \prod_{m=1}^{M} \widehat{p}(\boldsymbol{y}_m \mid \boldsymbol{\eta}_m, \sigma, \boldsymbol{\phi}_{\boldsymbol{X}}).$$

Since the likelihood estimator for each individual is unbiased and independent, the product of the estimators is also unbiased,

$$\mathbb{E}\left(\prod_{m=1}^{M} \widehat{p}(\boldsymbol{y}_m \mid \boldsymbol{\eta}_m, \sigma, \boldsymbol{\phi}_{\boldsymbol{X}})\right) = \prod_{m=1}^{M} \mathbb{E}\left(\widehat{p}(\boldsymbol{y}_m \mid \boldsymbol{\eta}_m, \sigma, \boldsymbol{\phi}_{\boldsymbol{X}})\right).$$

If the solution of the SDE is unavailable, the transition density is approximated using the Euler-Maruyama discretisation (EMD), so the target distribution is exact only up to discretisation error of the SDE. This error can be made arbitrarily small by increasing the level of discretisation at the expense of increased computation (see Section 2.3).

4.1 Individual-Augmentation Pseudo-Marginal

Our first method is Individual-Augmentation Pseudo-Marginal (IAPM), named for the additional auxiliary variables required to estimate the likelihood for each individual. Here, the likelihood estimate is,

$$\begin{aligned}
\widehat{p}(\boldsymbol{y}_m \mid \boldsymbol{\theta}) &= \int \widehat{p}(\boldsymbol{y}_m \mid \boldsymbol{\eta}_m, \sigma, \boldsymbol{\phi}_{\boldsymbol{X}}) p(\boldsymbol{\eta}_m \mid \boldsymbol{\phi}_{\boldsymbol{\eta}}) d\boldsymbol{\eta}_m, && \boldsymbol{\theta} = (\sigma, \boldsymbol{\phi}_{\boldsymbol{X}}, \boldsymbol{\phi}_{\boldsymbol{\eta}}), \\
&\approx \frac{1}{L} \sum_{l=1}^{L} \frac{\widehat{p}(\boldsymbol{y}_m \mid \boldsymbol{\eta}_m^{(l)}, \sigma, \boldsymbol{\phi}_{\boldsymbol{X}}) p(\boldsymbol{\eta}_m^{(l)} \mid \boldsymbol{\phi}_{\boldsymbol{\eta}})}{g(\boldsymbol{\eta}_m^{(l)} \mid \boldsymbol{\theta})}, && \boldsymbol{\eta}_m^{(l)} \sim g(\boldsymbol{\eta}_m \mid \boldsymbol{\theta})
\end{aligned}$$

using the importance distribution $g(\boldsymbol{\eta}_m \mid \boldsymbol{\theta})$ within a PMMH algorithm (Algorithm 2); see Algorithms 6 and 7 for more details.

The variability of $\widehat{p}(\boldsymbol{y}_m \mid \boldsymbol{\theta})$ for a given $g(\boldsymbol{\eta}_m \mid \boldsymbol{\theta})$ is controlled by the number of particles N, as well as the number of random effects draws L. The choice of importance distribution $g(\cdot \mid \boldsymbol{\theta})$ has an important impact on both of these quantities. A naive choice is $g(\boldsymbol{\eta}_m \mid \boldsymbol{\theta}) = p(\boldsymbol{\eta}_m \mid \boldsymbol{\theta})$. While this simplifies the likelihood calculation, it can be very inefficient if $\widehat{p}(\boldsymbol{\eta}_m \mid \boldsymbol{y}_m, \boldsymbol{\theta})$ and $p(\boldsymbol{\eta}_m \mid \boldsymbol{\theta})$ are not similar. We propose instead to use a Laplace approximation of a distribution over $\boldsymbol{\eta}_m$ that is proportional to $p(\boldsymbol{y}_m \mid \widehat{\boldsymbol{x}}_m, \boldsymbol{\theta}) p(\boldsymbol{\eta}_m \mid \boldsymbol{\theta})$, where $\widehat{\boldsymbol{x}}_m$ is an approximation of $\boldsymbol{x}_m$. We present two choices for $\widehat{\boldsymbol{x}}_m$. The first uses the solution of the ODE given by the drift of the SDEMEM (4), $d\widehat{X}_{m,t} = \mu(\widehat{X}_{m,t}, \boldsymbol{\phi}_{\boldsymbol{X}}, \boldsymbol{\eta}_m) dt$. The second approximates $\boldsymbol{x}_m$ with the mean of the modified diffusion bridge (see Section 2.3), with $\Delta_k = \Delta_t = \xi_{m,t+1} - \xi_{m,t}$, such that

$$\widehat{x}_{m,t+1} = \widehat{x}_{m,t} + \mu_{\text{MDB}}(\widehat{x}_{m,t})\Delta_t = \widehat{x}_{m,t} + \frac{\mu_t \sigma^2 + v_t(y_{m,t+1} - \widehat{x}_{m,t})}{v_t \Delta_t + \sigma^2} \Delta_t.$$

We refer to these importance distributions as Laplace-ODE and Laplace-MDB respectively.

Algorithm 6: The individual-augmentation pseudo-marginal method.

Input : data $\boldsymbol{y}_{1:M}$, initial values $\boldsymbol{\theta}^0$, and number of iterations I
Output : posterior samples $\boldsymbol{\theta}^{1:I}$

initialise $\boldsymbol{\theta}^1 = \boldsymbol{\theta}^0$
Draw $\boldsymbol{u} \sim p(\cdot)$
Run Algorithm 7 to obtain likelihood estimate $p(\boldsymbol{y}_{1:M} \mid \boldsymbol{\theta}^1, \boldsymbol{u})$
for $i = 2$ *to* I **do**
  Draw $\boldsymbol{\theta}^* \sim q(\cdot \mid \boldsymbol{\theta}^{i-1})$ and $\boldsymbol{u}^* \sim p(\cdot)$
  Run Algorithm 7 to obtain likelihood estimate $p(\boldsymbol{y}_{1:M} \mid \boldsymbol{\theta}^*, \boldsymbol{u}^*)$
  Calculate the acceptance probability

$$\alpha = \min\left(1, \frac{p(\boldsymbol{y}_{1:M} \mid \boldsymbol{\theta}^*, \boldsymbol{u}^*)p(\boldsymbol{\theta}^*)q(\boldsymbol{\theta}^{i-1} \mid \boldsymbol{\theta}^*)}{p(\boldsymbol{y}_{1:M} \mid \boldsymbol{\theta}^{i-1}, \boldsymbol{u})p(\boldsymbol{\theta}^{i-1})q(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}^{i-1})}\right)$$

  Draw $a \sim \mathcal{U}(0,1)$
  if $a < \alpha$ **then**
    Set $\boldsymbol{\theta}^i = \boldsymbol{\theta}^*$ and $\boldsymbol{u} = \boldsymbol{u}^*$
  else
    Set $\boldsymbol{\theta}^i = \boldsymbol{\theta}^{i-1}$
  end
end

Algorithm 7: Estimating the likelihood for the IAPM algorithm.

Input : data $\boldsymbol{y}_{1:M}$, parameter values $\boldsymbol{\theta}$, number of random effects draws L, number of particles N and vector of random numbers $\boldsymbol{u}^\dagger$
Output : likelihood estimate $\widehat{p}(\boldsymbol{y}_{1:M} \mid \boldsymbol{\theta})$

for $m = 1$ *to* M **do**
  for $l = 1$ *to* L **do**
    Draw $\boldsymbol{\eta}_m^l \sim g(\cdot \mid \boldsymbol{\theta})$
    Run Algorithm 1 with N particles with $\boldsymbol{\eta}_m^l$ to obtain the likelihood estimate Z_m^l
    Correct for the importance distribution $Z_m^l = \frac{Z_m^l}{g(\boldsymbol{\eta}_m^l \mid \boldsymbol{\theta})}$
  end
  Calculate $\widehat{p}(\boldsymbol{y}_m \mid \boldsymbol{\theta}) = \frac{1}{L}\sum_{i=1}^{L} Z_m^i$
end
Calculate $\widehat{p}(\boldsymbol{y}_{1:M} \mid \boldsymbol{\theta}) = \prod_{m=1}^{M} \widehat{p}(\boldsymbol{y}_m \mid \boldsymbol{\theta})$

†The random numbers in u are used implicitly in steps 3 and 4. For notational simplicity, indexing of and dependence on $\boldsymbol{u}$ is omitted.

Randomised quasi-Monte Carlo (RQMC) can be used to draw $\boldsymbol{\eta}_m^{(l)}$ (step 2 of Algorithm 7) as a variance reduction technique. See L'Ecuyer (2016) for an overview of RQMC.

Section 3.2 describes a correlated version of PMMH using block pseudo-marginal, which can also be applied to IAPM (cIAPM). We now briefly outline how to do this. Let $\boldsymbol{u} = (\boldsymbol{u}_{\mathbf{RE}}, \boldsymbol{u}_{\mathbf{PF}})$, where $\boldsymbol{u}_{\mathbf{RE}}$ and $\boldsymbol{u}_{\mathbf{PF}}$ are the random numbers used to draw the random effects and those used in the particle filter respectively. At each iteration of the chain, new random numbers for individual $m, 1 \leq m \leq M$ are proposed, while the rest are held constant. This induces a correlation of approximately $1 - 1/M$ between successive log-likelihood estimates (Tran et al., 2016). RQMC is straightforward to use within cIAPM as the random numbers are independent when using BPM; while correlated RQMC random numbers are possible, they are very difficult to implement effectively (see Gunawan et al., 2016).

Example (SDEMEM with constant drift and diffusion)**.** For the SDEMEM in (6), the IAPM approximation of $p(\boldsymbol{y}_m \mid \boldsymbol{\theta})$ with importance distribution $g(\boldsymbol{\eta}_m \mid \boldsymbol{\theta})$ is given by

$$\frac{1}{L}\sum_{l=1}^{L} \frac{\widehat{p}(\boldsymbol{y}_m \mid \boldsymbol{\beta}_m^{(l)}, \sigma, \gamma)\mathcal{N}(\boldsymbol{\beta}_m^{(l)}; \mu_\beta, \sigma_\beta^2)}{g(\boldsymbol{\beta}_m^{(l)} \mid \boldsymbol{\theta})}, \quad \boldsymbol{\beta}_m^{(l)} \sim g(\boldsymbol{\eta}_m \mid \boldsymbol{\theta}),$$

where $\widehat{p}(\boldsymbol{y}_m \mid \boldsymbol{\beta}_m^{(l)}, \sigma, \gamma)$ is the particle filter estimate of $p(\boldsymbol{y}_m \mid \boldsymbol{\beta}_m^{(l)}, \sigma, \gamma)$.

4.2 Component-Wise Pseudo-Marginal

This section defines a component-wise pseudo-marginal (CWPM) method, where the random effects $\boldsymbol{\eta}_{1:M}$ are updated along with $\boldsymbol{\theta}$ leading naturally to the parameter blocks $\boldsymbol{\eta}_{1:M}, \{\sigma, \boldsymbol{\phi}_{\boldsymbol{X}}\}$ and $\boldsymbol{\phi}_{\boldsymbol{\eta}}$. Denote $\boldsymbol{\theta}_{\boldsymbol{X}} = \{\sigma, \boldsymbol{\phi}_{\boldsymbol{X}}\}$; the joint posterior is of the form

$$p(\boldsymbol{\theta}_{\boldsymbol{X}}, \boldsymbol{\phi}_{\boldsymbol{\eta}}, \boldsymbol{\eta}_{1:M} \mid \boldsymbol{y}_{1:M}) \propto p(\boldsymbol{y}_{1:M} \mid \boldsymbol{\eta}_{1:M}, \boldsymbol{\theta}_{\boldsymbol{X}})p(\boldsymbol{\eta}_{1:M} \mid \boldsymbol{\phi}_{\boldsymbol{\eta}})p(\boldsymbol{\theta}_{\boldsymbol{X}})p(\boldsymbol{\phi}_{\boldsymbol{\eta}}),$$

and the full conditional posteriors for each of the parameter blocks are

$$\begin{aligned} p(\boldsymbol{\eta}_m \mid \boldsymbol{y}_{1:M}, \boldsymbol{\theta}_{\boldsymbol{X}}, \phi_\eta) &\propto p(\boldsymbol{y}_{1:M} \mid \boldsymbol{\eta}_{1:M}, \boldsymbol{\theta}_{\boldsymbol{X}})p(\boldsymbol{\eta}_{1:M} \mid \boldsymbol{\phi}_{\boldsymbol{\eta}}), \\ p(\boldsymbol{\theta}_{\boldsymbol{X}} \mid \boldsymbol{y}_{1:M}, \boldsymbol{\eta}_{1:M}) &\propto p(\boldsymbol{y}_{1:M} \mid \boldsymbol{\eta}_{1:M}, \boldsymbol{\theta}_{\boldsymbol{X}})p(\boldsymbol{\theta}_{\boldsymbol{X}}), \\ p(\boldsymbol{\phi}_{\boldsymbol{\eta}} \mid \boldsymbol{\eta}_{1:M}) &\propto p(\boldsymbol{\eta}_{1:M} \mid \boldsymbol{\phi}_{\boldsymbol{\eta}})p(\boldsymbol{\phi}_{\boldsymbol{\eta}}). \end{aligned} \tag{15}$$

A particle filter estimate of $p(\boldsymbol{y}_{1:M} \mid \boldsymbol{\eta}_{1:M}, \boldsymbol{\theta}_{\boldsymbol{X}})$ is used when updating $\boldsymbol{\eta}_{1:M}$ and $\boldsymbol{\theta}_{\boldsymbol{X}}$ (Algorithm 1). The parameter $\boldsymbol{\phi}_{\boldsymbol{\eta}}$ is updated directly however, since (15) is tractable. See Algorithm 8 for more details. This method is generally faster than IAPM as the particle filter is called $2M$ times per MCMC iteration (with the above configuration), instead of LM times as in IAPM. However, the CWPM chain may mix poorly if there is a high correlation between $\boldsymbol{\eta}_{1:M}$ and $\boldsymbol{\theta}$.

A correlated version of CWPM (cCWPM) may be implemented using BPM. Again, only the random numbers for a single individual are updated at each iteration while the rest are held constant.

Example (SDEMEM with constant drift and diffusion)**.** For the SDEMEM in (6), the parameters are updated in the following blocks, $\boldsymbol{\eta}_m = \{\beta_m\}$, $\boldsymbol{\theta_X} = \{\sigma, \gamma, x_0\}$ and $\boldsymbol{\phi_\eta} = \{\mu_\beta, \sigma_\beta\}$.

Algorithm 8: The component-wise pseudo-marginal (CWPM) method.

Input : data $\boldsymbol{y}_{1:M}$, initial values $\boldsymbol{\eta}^0_{1:M}$, ${\boldsymbol{\theta_X}}^0$ and ${\boldsymbol{\phi_\eta}}^0$, and number of iterations I

Output : posterior samples $\boldsymbol{\eta}^{1:I}_{1:M}$, ${\boldsymbol{\theta_X}}^{1:I}$ and ${\boldsymbol{\phi_\eta}}^{1:I}$

1 initialise $\boldsymbol{\eta}^1_{1:M} = \boldsymbol{\eta}^0_{1:M}$, ${\boldsymbol{\theta_X}}^1 = {\boldsymbol{\theta_X}}^0$ and ${\boldsymbol{\phi_\eta}}^1 = {\boldsymbol{\phi_\eta}}^0$

2 Draw $\boldsymbol{u} \sim p(\cdot)$

3 Run Algorithm 1 to obtain the likelihood estimate $p(\boldsymbol{y}_{1:M} \mid \boldsymbol{\eta}^1_{1:M}, {\boldsymbol{\theta_X}}^1, \boldsymbol{u})$

4 **for** $i = 2$ *to* I **do**

5 Draw ${\boldsymbol{\eta}^*}_{1:M} \sim q(\cdot \mid \boldsymbol{\eta}^{i-1}_{1:M})$ and $\boldsymbol{u}^* \sim p(\cdot)$

6 Run Algorithm 1 to obtain the likelihood estimate $p(\boldsymbol{y}_{1:M} \mid {\boldsymbol{\eta}^*}_{1:M}, {\boldsymbol{\theta_X}}^{i-1}, \boldsymbol{u}^*)$

7 Accept ${\boldsymbol{\eta}^*}_{1:M}$ and $\boldsymbol{u}^*$ with probability

$$\alpha = \min\left(1, \frac{p\left(\boldsymbol{y}_{1:M} \mid {\boldsymbol{\eta}^*}_{1:M}, {\boldsymbol{\theta_X}}^{i-1}, \boldsymbol{u}^*\right) p\left({\boldsymbol{\eta}^*}_{1:M} \mid {\boldsymbol{\phi_\eta}}^{i-1}\right) q\left(\boldsymbol{\eta}^{i-1}_{1:M} \mid {\boldsymbol{\eta}^*}_{1:M}\right)}{p\left(\boldsymbol{y}_{1:M} \mid \boldsymbol{\eta}^{i-1}_{1:M}, {\boldsymbol{\theta_X}}^{i-1}, \boldsymbol{u}\right) p\left(\boldsymbol{\eta}^{i-1}_{1:M} \mid {\boldsymbol{\phi_\eta}}^{i-1}\right) q\left({\boldsymbol{\eta}^*}_{1:M} \mid \boldsymbol{\eta}^{i-1}_{1:M}\right)}\right)$$

8 Draw $\boldsymbol{\theta}^*_{\boldsymbol{X}} \sim q(\cdot \mid {\boldsymbol{\theta_X}}^{i-1})$ and $\boldsymbol{u}^* \sim p(\cdot)$

9 Run Algorithm 1 to obtain the likelihood estimate $p(\boldsymbol{y}_{1:M} \mid \boldsymbol{\eta}^i_{1:M}, \boldsymbol{\theta}^*_{\boldsymbol{X}}, \boldsymbol{u}^*)$

10 Accept $\boldsymbol{\theta}^*_{\boldsymbol{X}}$ and $\boldsymbol{u}^*$ with probability

$$\alpha = \min\left(1, \frac{p\left(\boldsymbol{y}_{1:M} \mid \boldsymbol{\eta}^i_{1:M}, \boldsymbol{\theta}^*_{\boldsymbol{X}}, \boldsymbol{u}^*\right) p\left(\boldsymbol{\theta}^*_{\boldsymbol{X}}\right) q\left({\boldsymbol{\theta_X}}^{i-1} \mid \boldsymbol{\theta}^*_{\boldsymbol{X}}\right)}{p\left(\boldsymbol{y}_{1:M} \mid \boldsymbol{\eta}^i_{1:M}, {\boldsymbol{\theta_X}}^{i-1}, \boldsymbol{u}\right) p\left({\boldsymbol{\theta_X}}^{i-1}\right) q\left(\boldsymbol{\theta}^*_{\boldsymbol{X}} \mid {\boldsymbol{\theta_X}}^{i-1}\right)}\right)$$

11 Draw $\boldsymbol{\phi}^*_{\boldsymbol{\eta}} \sim q(\cdot \mid {\boldsymbol{\phi_\eta}}^{i-1})$

12 Accept $\boldsymbol{\phi}^*_{\boldsymbol{\eta}}$ with probability

$$\alpha = \min\left(1, \frac{p\left(\boldsymbol{\eta}^i_{1:M} \mid \boldsymbol{\phi}^*_{\boldsymbol{\eta}}\right) p\left(\boldsymbol{\phi}^*_{\boldsymbol{\eta}}\right) q\left({\boldsymbol{\phi_\eta}}^{i-1} \mid \boldsymbol{\phi}^*_{\boldsymbol{\eta}}\right)}{p\left(\boldsymbol{\eta}^i_{1:M} \mid {\boldsymbol{\phi_\eta}}^{i-1}\right) p\left({\boldsymbol{\phi_\eta}}^{i-1}\right) q\left(\boldsymbol{\phi}^*_{\boldsymbol{\eta}} \mid {\boldsymbol{\phi_\eta}}^{i-1}\right)}\right)$$

13 **end**

For the correlated version, new random numbers are only drawn for a single block in steps 4 and 7, not the whole vector.

4.3 Mixed Particle Method

Our final method is a variation of the PMMH + PG algorithm of Gunawan et al. (2018a). We use a combination of PMMH and PG to update the parameters $\boldsymbol{\eta}_{1:M}, \sigma, \boldsymbol{\phi_X}$ and

$\boldsymbol{\phi}_{\boldsymbol{\eta}}$, depending on the form of the full conditional distributions,

$$\begin{aligned} p(\boldsymbol{\eta}_{1:M} \mid \boldsymbol{y}_{1:M}, \sigma, \boldsymbol{\phi}_{\boldsymbol{X}}, \boldsymbol{\phi}_{\boldsymbol{\eta}}) &\propto p(\boldsymbol{y}_{1:M} \mid \boldsymbol{\eta}_{1:M}, \sigma, \boldsymbol{\phi}_{\boldsymbol{X}}) p(\boldsymbol{\eta}_{1:M} \mid \boldsymbol{\phi}_{\boldsymbol{\eta}}), \qquad (16) \\ p(\sigma \mid \boldsymbol{y}_{1:M}, \boldsymbol{x}_{1:M}) &\propto p(\boldsymbol{y}_{1:M} \mid \boldsymbol{x}_{1:M}, \sigma) p(\sigma), \\ p(\boldsymbol{\phi}_{\boldsymbol{X}} \mid \boldsymbol{y}_{1:M}, \boldsymbol{\eta}_{1:M}, \sigma, \boldsymbol{\phi}_{\boldsymbol{X}}) &\propto p(\boldsymbol{y}_{1:M} \mid \boldsymbol{\eta}_{1:M}, \sigma, \boldsymbol{\phi}_{\boldsymbol{X}}) p(\boldsymbol{\phi}_{\boldsymbol{X}}), \qquad (17) \\ p(\boldsymbol{\phi}_{\boldsymbol{\eta}} \mid \boldsymbol{\eta}_{1:M}) &\propto p(\boldsymbol{\eta}_{1:M} \mid \boldsymbol{\phi}_{\boldsymbol{\eta}}) p(\boldsymbol{\phi}_{\boldsymbol{\eta}}). \end{aligned}$$

At each iteration, the invariant path $\boldsymbol{x}_{1:M}$ is updated using a conditional particle filter (Algorithm 3). Where the density $p(\boldsymbol{y}_{1:M} \mid \boldsymbol{\eta}_{1:M}, \sigma, \boldsymbol{\phi}_{\boldsymbol{X}})$ is required, i.e. (16) and (17), a particle filter estimate is used (PMMH step). The full conditionals for σ and $\boldsymbol{\phi}_{\boldsymbol{\eta}}$ are tractable as they only depend on the observation density, and the priors for the random effects and the fixed common parameters ($\boldsymbol{\theta}$), all of which are known. Hence, σ and $\boldsymbol{\phi}_{\boldsymbol{\eta}}$ can be updated directly. It is important that the likelihood estimate is updated once a new value of σ is accepted; this must be done with the same $\boldsymbol{u}$ that was used to estimate the previous likelihood. See Algorithm 9 for more details. As with CWPM (Section 4.2), mixing of the Markov chain can be poor if high correlation exists between $\boldsymbol{\eta}_{1:M}$ and $\boldsymbol{\theta}$ and/or $\boldsymbol{x}_{1:M}$ and σ.

Similarly to IAPM and CWPM, a correlated version of MPM (cMPM) can be implemented using BPM, where $\boldsymbol{u}$ is divided into M blocks based on the individuals $m = 1, \ldots, M$.

Example (SDEMEM with constant drift and diffusion)**.** For the SDEMEM in (6), the parameters are updated in the following blocks, $\boldsymbol{\eta}_m = \{\beta_m\}$, $\boldsymbol{\phi}_{\boldsymbol{X}} = \{\gamma, x_0\}$, $\boldsymbol{\phi}_{\boldsymbol{\eta}} = \{\mu_\beta, \sigma_\beta\}$ and σ.

4.4 Likelihood Estimation

Three particle MCMC methods are introduced: IAPM, CWPM and MPM; each relies on a particle filter to calculate an unbiased estimate of the intractable likelihood. Tuning parameters for this calculation are the level of discretisation D (if the exact transition density is unknown), the number of particles N and, for IAPM, the number of random effects draws L.

The likelihood estimator in PMMH is typically tuned such that the variance of the log-likelihood ratio is between 1 and 4 (Sherlock et al., 2015; Pitt et al., 2012; Doucet et al., 2015). This optimizes the trade-off between statistical and computational efficiency, i.e. the number of particles versus the computation time. Tuning is usually done through experimentation at a central location of the posterior, which is often obtained using pilot runs. Deligiannidis et al. (2018) and Tran et al. (2016) also utilise this approach for their correlated likelihood estimators.

Deligiannidis et al. (2018) use trial-and-error to tune the correlation of the random numbers such that the standard deviation is approximately 1.4. Tran et al. (2016) derives the optimal standard deviation of the log-likelihood ratio when Monte Carlo and randomised quasi-Monte Carlo random numbers are used to estimate the likelihood. For the latter, they obtain $0.82/(1-\rho^2)^{1/2}$, where $\rho = 1 - 1/B$ is the approximate correlation between the log-likelihoods and B is the number of blocks. Thus, for a

Algorithm 9: Mixed particle method (MPM) algorithm.

Input : data $\boldsymbol{y}_{1:M}$, initial values $\boldsymbol{\eta}^0_{1:M}$, σ^0, ${\boldsymbol{\phi}_{\boldsymbol{X}}}^0$, ${\boldsymbol{\phi}_{\boldsymbol{\eta}}}^0$, and $\boldsymbol{x}^0_{1:M}$, initial path $\boldsymbol{x}^0_{1:M}$ and associated ancestral lineage $\boldsymbol{b}_{1:T}$, and the number of iterations I

Output : posterior samples $\boldsymbol{\eta}^{1:I}_{1:M}$, $\sigma^{1:I}$, ${\boldsymbol{\phi}_{\boldsymbol{X}}}^{1:I}$, ${\boldsymbol{\phi}_{\boldsymbol{\eta}}}^{1:I}$, and $\boldsymbol{x}^{1:I}_{1:M}$

1 Initialise $\boldsymbol{\eta}^1_{1:M} = \boldsymbol{\eta}^0_{1:M}$, $\sigma^1 = \sigma^0$, ${\boldsymbol{\phi}_{\boldsymbol{X}}}^1 = {\boldsymbol{\phi}_{\boldsymbol{X}}}^0$, ${\boldsymbol{\phi}_{\boldsymbol{\eta}}}^1 = {\boldsymbol{\phi}_{\boldsymbol{\eta}}}^0$, and $\boldsymbol{x}^1_{1:M} = \boldsymbol{x}^0_{1:M}$

2 Draw $\boldsymbol{u} \sim p(\cdot)$

3 Run Algorithm 1 to obtain the likelihood estimate $p(\boldsymbol{y}_{1:M} \mid \boldsymbol{\eta}^1_{1:M}, \sigma^1, {\boldsymbol{\phi}_{\boldsymbol{X}}}^1, \boldsymbol{u})$

4 **for** $i = 2$ *to* I **do**

5 Draw ${\boldsymbol{\eta}^*}_{1:M} \sim q(\cdot \mid \boldsymbol{\eta}^{i-1}_{1:M})$ and $\boldsymbol{u}^* \sim p(\cdot)$

6 Run Algorithm 1 to obtain the likelihood estimate $p(\boldsymbol{y}_{1:M} \mid {\boldsymbol{\eta}^*}_{1:M}, \sigma^{i-1}, {\boldsymbol{\phi}_{\boldsymbol{X}}}^{i-1}, \boldsymbol{u}^*)$

7 Accept ${\boldsymbol{\eta}^*}_{1:M}$ and $\boldsymbol{u}^*$ with probability

$$\alpha = \min\left(1, \frac{p(\boldsymbol{y}_{1:M} \mid {\boldsymbol{\eta}^*}_{1:M}, \sigma^{i-1}, {\boldsymbol{\phi}_{\boldsymbol{X}}}^{i-1}, \boldsymbol{u}^*) p({\boldsymbol{\eta}^*}_{1:M} \mid {\boldsymbol{\phi}_{\boldsymbol{\eta}}}^{i-1}) q(\boldsymbol{\eta}^{i-1}_{1:M} \mid {\boldsymbol{\eta}^*}_{1:M})}{p(\boldsymbol{y}_{1:M} \mid \boldsymbol{\eta}^{i-1}_{1:M}, \sigma^{i-1}, {\boldsymbol{\phi}_{\boldsymbol{X}}}^{i-1}, \boldsymbol{u}) p(\boldsymbol{\eta}^{i-1}_{1:M} \mid {\boldsymbol{\phi}_{\boldsymbol{\eta}}}^{i-1}) q({\boldsymbol{\eta}^*}_{1:M} \mid \boldsymbol{\eta}^{i-1}_{1:M})}\right)$$

8 Draw $\sigma^* \sim q(\cdot \mid \sigma^{i-1})$

9 Accept σ^* with probability

$$\alpha = \min\left(1, \frac{p(\boldsymbol{y}_{1:M} \mid \boldsymbol{x}^{i-1}_{1:M}, \sigma^*) p(\sigma^*) q(\sigma^{i-1} \mid \sigma^*)}{p(\boldsymbol{y}_{1:M} \mid \boldsymbol{x}^{i-1}_{1:M}, \sigma^{i-1}) p(\sigma^{i-1}) q(\sigma^* \mid \sigma^{i-1})}\right)$$

10 Run Algorithm 1 to update the likelihood estimate $p(\boldsymbol{y}_{1:M} \mid \boldsymbol{\eta}^i_{1:M}, \sigma^i, {\boldsymbol{\phi}_{\boldsymbol{X}}}^{i-1}, \boldsymbol{u})$

11 Draw $\boldsymbol{\phi}^*_{\boldsymbol{X}} \sim q(\cdot \mid {\boldsymbol{\phi}_{\boldsymbol{X}}}^{i-1})$ and $\boldsymbol{u}^* \sim p(\cdot)$

12 Run Algorithm 1 to obtain the likelihood estimate $p(\boldsymbol{y}_{1:M} \mid \boldsymbol{\eta}^i_{1:M}, \sigma^i, \boldsymbol{\phi}^*_{\boldsymbol{X}}, \boldsymbol{u}^*)$

13 Accept $\boldsymbol{\phi}^*_{\boldsymbol{X}}$ and $\boldsymbol{u}^*$ with probability

$$\alpha = \min\left(1, \frac{\widehat{p}(\boldsymbol{y}_{1:M} \mid \boldsymbol{\eta}^i_{1:M}, \sigma^i, \boldsymbol{\phi}^*_{\boldsymbol{X}}) p(\boldsymbol{\phi}^*_{\boldsymbol{X}}) q({\boldsymbol{\phi}_{\boldsymbol{X}}}^{i-1} \mid \boldsymbol{\phi}^*_{\boldsymbol{X}})}{\widehat{p}(\boldsymbol{y}_{1:M} \mid \boldsymbol{\eta}^i_{1:M}, \sigma^i, {\boldsymbol{\phi}_{\boldsymbol{X}}}^{i-1}) p({\boldsymbol{\phi}_{\boldsymbol{X}}}^{i-1}) q(\boldsymbol{\phi}^*_{\boldsymbol{X}} \mid {\boldsymbol{\phi}_{\boldsymbol{X}}}^{i-1})}\right)$$

14 Draw $\boldsymbol{\phi}^*_{\boldsymbol{\eta}} \sim q(\cdot \mid {\boldsymbol{\phi}_{\boldsymbol{\eta}}}^{i-1})$

15 Accept $\boldsymbol{\phi}^*_{\boldsymbol{\eta}}$ with probability

$$\alpha = \min\left(1, \frac{p(\boldsymbol{\eta}^i_{1:M} \mid \boldsymbol{\phi}^*_{\boldsymbol{\eta}}) p(\boldsymbol{\phi}^*_{\boldsymbol{\eta}}) q({\boldsymbol{\phi}_{\boldsymbol{\eta}}}^{i-1} \mid \boldsymbol{\phi}^*_{\boldsymbol{\eta}})}{p(\boldsymbol{\eta}^i_{1:M} \mid {\boldsymbol{\phi}_{\boldsymbol{\eta}}}^{i-1}) p({\boldsymbol{\phi}_{\boldsymbol{\eta}}}^{i-1}) q(\boldsymbol{\phi}^*_{\boldsymbol{\eta}} \mid {\boldsymbol{\phi}_{\boldsymbol{\eta}}}^{i-1})}\right)$$

16 Run Algorithm 3 with $\boldsymbol{x}^{i-1}_{1:M}$ and $\boldsymbol{b}^{i-1}_{1:M}$ to obtain a new path $\boldsymbol{x}^i_{1:M}$ and $\boldsymbol{b}^i_{1:M}$

17

The random effects $\boldsymbol{\eta}_{1:M}$ and parameters $\boldsymbol{\phi}_{\boldsymbol{X}}$ are updated using PMMH. The latent states $\boldsymbol{X}_{1:M}$ are updated using PG and σ and $\boldsymbol{\phi}_{\boldsymbol{\eta}}$ are updated directly. For the correlated version, new random numbers are only drawn for a single block in steps 4 and 10, not the whole vector.

Tuning Parameters	description	Simplifying assumption
D†	level of discretisation for the SDE	–
N	number of particles	–
L*	number of random effects draws	$L = N$

Table 1: Tuning parameters for IAPM, CWPM and MPM. †Unnecessary if the exact transition density is known. *Only applicable to IAPM.

correlation between 0.8–0.99 (5–100 blocks), the target or optimal standard deviation is between 1.37–5.81. Tran et al. (2016) also use a different number of particles to estimate the likelihood for each block.

Following this general approach, we use

$$\sigma_{\Delta} = \text{std}\left(\log \frac{p(\boldsymbol{y}_{1:M} \mid \bar{\boldsymbol{\theta}}, \boldsymbol{u}^{*})}{p(\boldsymbol{y}_{1:M} \mid \bar{\boldsymbol{\theta}}, \boldsymbol{u})}\right),$$

to tune N and L for IAPM and

$$\sigma_{\Delta} = \text{std}\left(\log \frac{p(\boldsymbol{y}_{1:M} \mid \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\eta}}_{1:M}, \boldsymbol{u}^{*})}{p(\boldsymbol{y}_{1:M} \mid \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\eta}}_{1:M}, \boldsymbol{u})}\right),$$

for CWPM and MPM, where $\bar{\boldsymbol{\theta}}$ and $\bar{\boldsymbol{\eta}}_{1:M}$ are central values of $\boldsymbol{\theta}$ and $\boldsymbol{\eta}_{1:M}$, $\boldsymbol{u}^{*}$ is the proposed set of random numbers and $\boldsymbol{u}$ is the current set of random numbers. We aim for $1.4 \leq \sigma_{\Delta} \leq 1.8$.

The simplifying assumption that $L = N$ is made for IAPM. The number of particles for each of our methods can then be tuned through experimentation, by selecting N such that $1.4 \leq \sigma_{\Delta} \leq 1.8$. A fixed value of the level of discretisation D is used throughout. In many cases, it is possible to select a reasonable value of D based only on the model. To simplify the tuning process, the same number of particles is used across all individuals; however efficiency gains are possible if this value is allowed to vary between subjects. Table 1 shows the tuning parameters for all methods. Assumptions to simplify the tuning process are provided where available.

It is necessary to also specify a proposal function for the particle filter and the importance density for IAPM. Section 2.3 describes three different ways to simulate from an SDE: the Euler-Maruyama discretisation (EMD), the modified diffusion bridge (MDB) and the residual bridge (RB). Any of these can be used to move particles within a particle filter. Section 4.1 also proposes the Laplace-ODE and Laplace-MDB importance densities for IAPM. The optimal choice of the proposal function and the importance density is problem specific and may have a large impact on the efficiency of the likelihood estimate. In general, it is possible to choose the importance density based on the proposal function, i.e. EMD + Laplace-ODE (or L-ODE) and MDB/RB + Laplace-MDB. Recall that the Laplace-ODE approximates the underlying states using the ODE specified by the drift of the SDEMEM; the feasibility of this importance density relies on how quickly the solution of the ODE is computed. As with D, exploration of the model may indicate a sensible choice of proposal function. Table 2 shows the implementation choices for IAPM, CWPM and MPM, as well as the recommended default choices.

Implementation choice	Options	Recommended default
correlate the log-likelihood estimates using BPM	1. use BPM 2. do not use BPM	1
proposal function	1. Euler-Maruyama discretisation 2. modified diffusion bridge 3. residual bridge	2
importance density*	1. prior 2. Laplace-ODE 3. Laplace-MDB	3

Table 2: Implementation choices for IAPM, CWPM and MPM. *Only applicable to IAPM.

For CWPM and MPM, we recommend using more efficient proposals for $\boldsymbol{\phi}_{\boldsymbol{\eta}}$ and σ (MPM) if possible, e.g. those based on MALA or HMC.

5 Tumor Xenography Study

5.1 Data

We apply our methods to real data from a tumour xenography study on mice obtained from Picchini and Forman (2019). The study had 4 treatment groups and 1 control group, and each group had 7–8 mice. Measurements were taken every Monday, Wednesday and Friday for six weeks; however, the majority of the mice were euthanized before the end of the study, once their tumour volumes exceeded 1000 cubic mm.

We focus specifically on group 5 (the control group). There are 7 mice in this group, with 2–14 observations per mouse and 34 observations in total. Only one mouse in this group survived longer than 11 days, being euthanized on day 32 of the study. Figure 1 plots this data.

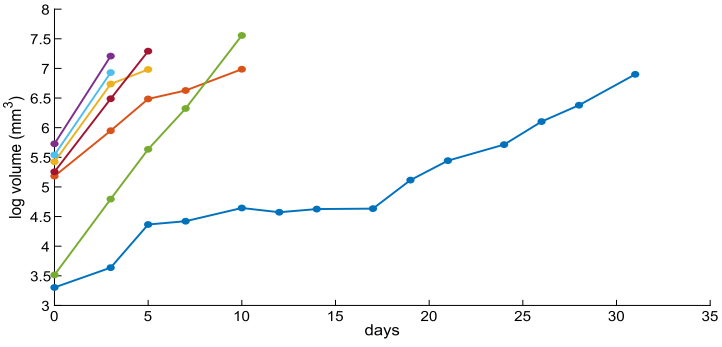

Figure 1: Plot of real tumour volume data.

5.2 Model

To fit the data, we consider an adaptation of the SDEMEM used by Picchini and Forman (2019) for unperturbed growth. There are $m = 1, \ldots, M$ subjects, with measurements taken at discrete times $\xi_t, t = 1, \ldots, T_m$, where T_m is the number of observations for subject m. The model is,

$$dV_{m,t} = \left(\beta_m + \frac{\gamma^2}{2}\right) V_{m,t} dt + \gamma V_{m,t}^{\rho} dB_{m,t}, \quad V_{m0} = v_{m0}, \tag{18}$$

where $V_{m,t}$ is the volume of subject m at time ξ_t. The underlying ODE model has solution $V_{m,t} = v_{m0} \exp(\beta_m t)$, which is the general exponential growth model. The random effects for this model are the parameters β_m and V_{m0}, which are assigned the prior distributions

$$\begin{aligned} \log(V_{m0}) &\sim \mathcal{N}(\log(V_{m0}); \mu_{V0}, \sigma_{V0}^2), \\ \log(\beta_m) &\sim \mathcal{N}(\log(\beta_m); \mu_\beta, \sigma_\beta^2). \end{aligned}$$

Since both β_m and V_{m0} are constrained to be positive, they are updated on the log scale. The observations are modelled as

$$Y_{m,t} = \log(V_{m,t}) + \epsilon_{m,t}, \quad \epsilon_{m,t} \sim \mathcal{N}(\epsilon_{m,t}; 0, \sigma^2). \tag{19}$$

Since the data is observed on the log scale, the transformation $X_{m,t} = \log(V_{m,t})$ can be applied to (18) and (19) using Itô's lemma. The full model is then given by

$$\begin{cases} Y_{m,t} = X_{m,t} + \epsilon_{m,t}, \quad \epsilon_{m,t} \sim \mathcal{N}(0, \sigma^2), \\ dX_{m,t} = \left(\beta_m + \frac{\gamma^2}{2}(1 - e^{2(\rho-1)X_{m,t}})\right) dt + \gamma e^{(\rho-1)X_{m,t}} dB_{m,t}, \\ X_{m0} \sim \mathcal{N}(X_{m0}; \mu_{X0}, \sigma_{X0}^2), \\ \log(\beta_m) \sim \mathcal{N}(\log(\beta_m); \mu_\beta, \sigma_\beta^2). \end{cases} \tag{20}$$

The likelihood is intractable since model (20) does not have a closed form solution for $X_{m,t}$. The following prior is assigned to $\theta = (\mu_{X0}, \sigma_{X0}, \mu_\beta, \sigma_\beta, \gamma, \sigma, \rho)^\top$

$$\begin{aligned} p(\boldsymbol{\theta}) = &\mathcal{N}(\mu_{X0}; 3, 4^2)\mathcal{HN}(\sigma_{X0}; 5^2)\mathcal{N}(\mu_\beta; 0, 4^2)\mathcal{HN}(\sigma_\beta; 5^2)\mathcal{HN}(\gamma; 5^2) \\ &\times \mathcal{HN}(\sigma; 5^2)\mathcal{N}(\rho; 1, 0.5^2), \end{aligned}$$

where $\mathcal{HN}(\sigma^2)$ refers to the half-normal distribution with mean zero and scale parameter σ.

Note that taking $\rho = 1$ gives model (7), which is the original SDEMEM used by Picchini and Forman (2019). We add the parameter ρ which allows for both a more flexible variance and renders the transition density intractable. We test this model on the dataset introduced in Section 5.1. To ensure numerical stability when simulating from the SDE, we scaled the observation times by the maximum time observed. In addition to the real data, we also apply our methods to synthetic data simulated from model (20) using $\boldsymbol{\theta} = (\mu_{X0}, \sigma_{X0}, \mu_\beta, \sigma_\beta, \gamma, \sigma, \rho)^\top = (3, 1, -1, 1, 1, 0.5, 1)^\top$.

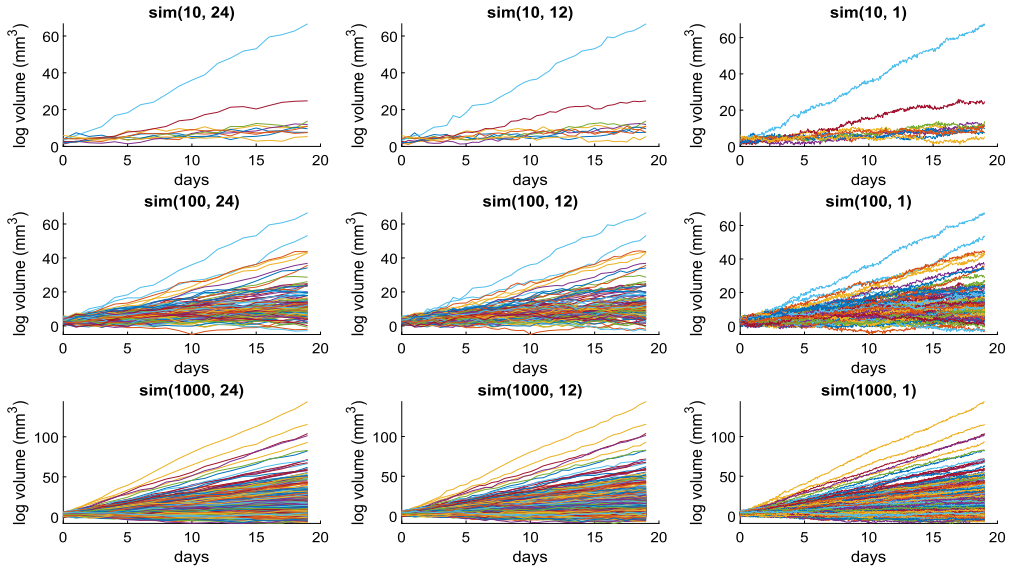

Figure 2: Plot of all simulated datasets. Sim(M, H) refers to the size of the subset, where M is the number of subjects and H is the number of hours between observations. The full dataset is denoted by sim(1000, 1).

For the synthetic data, we assumed 1000 mice with 457 observations each – this corresponds to a measurement every hour for 19 days following the initial measurement. We used 9 subsets of this dataset with all combinations of 10, 100 and 1000 subjects and an observation every 24 hours (20 observations), 12 hours (39 observations) and 1 hour (457 observations). We refer to these datasets as sim(M, H), where M is the number of subjects (10, 100, or 1000) and H is the number of hours between observations (24, 12 or 1). For example, the subset of 100 subjects with an observation every 12 hours is denoted sim(100, 12), while the full dataset is denoted sim(1000, 1). When M is left blank, we refer to all datasets with the specified value of H, e.g. sim(, 1) represents sim(10, 1), sim(100, 1) and sim(1000, 1). Similarly, when H is left blank, we refer to all datasets with the specified value of M, e.g. sim(1000,) represents sim(1000, 24), sim(1000, 12) and sim(1000, 1). The performance of our methods on these datasets indicates their scalability with respect to the density of the time series and number of subjects. Figure 2 plots this data.

6 Likelihood Estimation Results

All code is implemented in MATLAB. Vectorisation and parallelisation are applied where possible; e.g. in the particle filter we vectorise the particle operations and parallelise over the subjects. Parallelisation is only applied if the average number of observations per subject is greater than 10, and it is not used for CWPM and MPM on the

sim(10, 24) dataset as it increased the computation time. For IAPM we also parallelise over the random effects draws when running the importance sampler. Our results are produced using 8 cores. Resampling is done at every iteration in the conditional particle filter, but we use adaptive resampling when estimating the likelihood (resampling when ESS $< N/2$). For block pseudo-marginal, we use $B = \min(M, 100)$ blocks and update them systematically.

We first consider the efficiency of the likelihood estimation. For each of the three methods, all possible combinations of proposal function and importance density (IAPM) are tested. We define the naive combination as the IAPM algorithm with the prior as importance density and the Euler-Maruyama discretisation (EMD) as the proposal function in the particle filter. The naive method is the uncorrelated version of this combination.

As outlined in Section 4.4, the tuning parameters are set such that $1.4 \leq \sigma_\Delta \leq 1.8$. Measurements are calculated from a minimum of 1000 log-likelihood estimates at a fixed value of $\boldsymbol{\theta}$ and $\boldsymbol{\eta}_{1:M}$ (CWPM). For the real data, we use $\boldsymbol{\theta} = (4, 1, 2, 1, 1.6, 0.05, 1)$, which is obtained from a few preliminary MCMC runs (low values of N, L and D are sufficient for this). For the simulated data, we use the true value $\boldsymbol{\theta} = (3, 1, -1, 1, 1, 0.5, 1)$. The random effects $\boldsymbol{\eta}_{1:M}$ are determined similarly, using preliminary runs for the real data and the true values for the synthetic data.

We define the level of discretisation (D) as the number of intermediate timepoints between each observation. The results seem insensitive to this value, so D is fixed at 10 for all methods. Computation is stopped if the computation time for a single log-likelihood estimate exceeds 15 minutes or require more than 150 GB of RAM.

This section uses the notation 'importance density + proposal function' to refer to a particular combination of the two, e.g. prior + RB. All combinations are tuned to roughly the same statistical efficiency (based on σ_Δ), such that the most efficient method has the lowest computation time. Further mention of statistical efficiency refers to the value of the tuning parameters N and L.

6.1 IAPM

Of the three methods, IAPM is the most difficult and time-consuming to tune. Assuming $L = N$ simplifies the tuning process, but is sub-optimal. Depending on the implementation of the code, having a larger/smaller N or L can significantly improve the computation time.

Once we started testing combinations, we found that the variance of the Laplace-ODE importance density approaches 0 for at least one of the random effects, such that the draws for that random effect are approximately equal. This is solved by setting the covariance to a diagonal matrix of the prior variances scaled by 0.5; the altered importance density is denoted as L-ODE.

Tables 3–5 summarize the log-likelihood results for all datasets. Dashed lines indicate that the computation time exceeds the 15 minute time limit per likelihood estimate. All combinations exceed this limit on the sim(1000, 1) dataset. Likewise, for the sim(1000,

		Prior		L-ODE		Lap-MDB	
PF	Cor.	L, N	time (s)	L, N	time (s)	L, N	time (s)
EMD	No	200	0.21	60	0.12	28	0.11
	Yes	90	0.11	30	0.10	19	0.09
MDB	No	180	0.36	35	0.11	4	0.05
	Yes	65	0.12	16	0.10	3	0.04
RB	No	180	0.38	35	0.11	4	0.05
	Yes	65	0.13	16	0.11	3	0.04

Table 3: Log-likelihood results for the IAPM method on the real dataset. The highlighted rows show the combinations which gave the best computation time. Notation: PF = proposal function used in particle filter, Cor. = indicates whether likelihood estimates are correlated or not.

			sim(10, 24)		sim(10, 12)		sim(10, 1)		sim(100, 24)	
IS	PF	Cor.	L, N	time (s)	L, N	time (s)	L, N	time (s)	L, N	time (s)
Prior	EMD	No	250	1.69	370	6.29	530	134.1	500	52.78
		Yes	115	0.54	130	1.13	335	52.58	95	3.28
	MDB	No	220	3.06	220	5.84	570	373.3	300	49.71
		Yes	95	0.86	100	1.63	250	80.02	45	2.88
	RB	No	220	3.07	220	6.14	570	385.9	320	60.60
		Yes	95	0.87	100	1.79	250	87.62	45	3.07
L-ODE	EMD	No	220	1.31	950	35.9	–	–	–	–
		Yes	60	0.32	120	0.99	370	62.71	155	7.51
	MDB	No	145	1.57	800	55.75	–	–	–	–
		Yes	20	0.22	50	0.78	310	118.0	100	8.13
	RB	No	145	1.62	800	60.45	–	–	–	–
		Yes	20	0.21	50	0.81	310	126.7	100	8.75
Lap-MDB	EMD	No	40	0.20	55	0.45	150	14.22	130	5.55
		Yes	45	0.25	75	0.57	290	38.61	65	2.42
	MDB	No	8	0.12	16	0.23	120	26.22	30	2.12
		Yes	4	0.10	10	0.21	190	52.18	4	0.63
	RB	No	8	0.12	16	0.25	120	29.70	30	2.53
		Yes	4	0.11	10	0.25	190	53.87	4	0.68

Table 4: Log-likelihood results for the IAPM method on the sim(10,) and sim(100, 24) datasets. The highlighted rows show the combinations which give the best computation time. The number of observations for each dataset (from left to right): 200, 390, 4,570, 2,000. Notation: PF = proposal function used in particle filter, IS = importance density used for the importance sampling step, Cor. = indicates whether likelihood estimates are correlated or not.

24) and sim(1000, 12) datasets, results can only be obtained for the Laplace-MDB importance density within the time limit. On these datasets, we find that the variance of the estimated σ_Δ is very high for the correlated versions of the prior and L-ODE combinations.

IS	PF	Cor.	sim(100, 12) L, N	sim(100, 12) time (s)	sim(100, 1) L, N	sim(100, 1) time (s)	sim(1000, 24) L, N	sim(1000, 24) time (s)	sim(1000, 12) L, N	sim(1000, 12) time (s)
Prior	EMD	No	500	105.7	–	–	–	–	–	–
		Yes	110	8.3985	300	419.8	–	–	–	–
	MDB	No	390	154.0	–	–	–	–	–	–
		Yes	45	6.16	200	556.8	–	–	–	–
	RB	No	390	174.0	–	–	–	–	–	–
		Yes	45	6.05	200	584.8	–	–	–	–
L-ODE	EMD	No	–	–	–	–	–	–	–	–
		Yes	370	76.64	–	–	–	–	–	–
	MDB	No	–	–	–	–	–	–	–	–
		Yes	230	57.27	–	–	–	–	–	–
	RB	No	–	–	–	–	–	–	–	–
		Yes	230	65.51	–	–	–	–	–	–
Lap-MDB	EMD	No	140	11.70	–	–	350	285.5	400	701.2
		Yes	80	5.26	300	417.7	65	24.95	80	50.76
	MDB	No	50	7.48	–	–	90	71.46	145	292.2
		Yes	10	1.53	200	532.1	4	6.50	10	15.65
	RB	No	50	7.92	–	–	90	77.83	145	315.3
		Yes	10	1.71	200	587.6	4	7.20	10	16.63

Table 5: Log-likelihood results for the IAPM method on the sim(100,12), sim(100,1), sim(1000,24) and sim(1000,12) datasets. The highlighted rows show the combinations which give the best computation time. The number of observations for each dataset (from left to right): 3,900, 45,700, 20,000, 39,000. Notation: PF = proposal function used in particle filter, IS = importance density used for the importance sampling step, Cor. = indicates whether likelihood estimates are correlated or not.

Correlating the log-likelihoods generally increases the statistical efficiency. This increase is significant on the larger datasets, as is the corresponding reduction in computation time. Interestingly, for all sim(10,) datasets, the uncorrelated Laplace-MDB + EMD is more statistically efficient than the correlated version. This is also true for Laplace-MDB + MDB and RB on the sim(10, 1) dataset.

Across all datasets, the Laplace-MDB importance density outperforms the prior and L-ODE in terms of overall efficiency. Of the prior and L-ODE, the latter shows the poorest performance. Results for the uncorrelated versions are only available for the real, sim(10, 24) and sim(10, 12) datasets and these are also the only datasets with L-ODE combinations that outperform the prior. Based on these results, the drift ODE may not be a good approximation of the underlying states. A large diffusion coefficient and/or measurement error can account for this.

The most efficient proposal function depends on the size of the dataset. In terms of statistical efficiency, the modified diffusion bridge and residual bridge give nearly identical results and generally outperform the Euler-Maruyama discretisation. The latter is the fastest however, and the residual bridge is the slowest. While this has little effect on

		real		sim(10, 24)		sim(10, 12)		sim(10, 1)		sim(100, 24)	
PF	Cor.	N	time (s)	N	time (s)	N	time (s)	N	time (s)	N	time (s)
EMD	No	200	0.0044	450	0.0578	700	0.0915	3100	1.8672	6500	1.1254
	Yes	60	0.0030	65	0.0559	85	0.0583	300	0.2773	120	0.1107
MDB	No	1	0.0023	30	0.0490	110	0.0728	2100	3.1266	350	0.2385
	Yes	1	0.0023	3	0.0470	10	0.0554	215	0.4922	3	0.1038
RB	No	1	0.0024	30	0.0503	110	0.0743	2100	3.3687	350	0.2527
	Yes	1	0.0024	3	0.0473	10	0.0578	210	0.54	3	0.1035

Table 6: Log-likelihood results for the CWPM method on the real, sim(10,) and sim(100,24) datasets. The highlighted rows show the combinations which give the best time. Notation: PF = proposal function used in particle filter, Cor. = indicates whether likelihood estimates are correlated or not.

		sim(100, 12)		sim(100, 1)		sim(1000, 24)		sim(1000, 12)	
PF	Cor.	N	time (s)	N	time (s)	N	time (s)	N	time (s)
EMD	No	9000	2.9294	–	–	–	–	–	–
	Yes	120	0.1536	360	2.783	90	0.4702	110	0.9162
MDB	No	1200	1.0330	–	–	3500	12.76	11000	78.72
	Yes	11	0.1492	240	3.544	3	0.4504	12	0.8920
RB	No	1200	1.13	–	–	3500	13.91	11000	86.91
	Yes	11	0.1523	240	4.019	3	0.4906	12	0.9433

Table 7: Log-likelihood results for the CWPM method on the sim(100,12), sim(100,1), sim(1000,24) and sim(1000,12) datasets. The highlighted rows show the combinations which give the best time. Notation: PF = proposal function used in particle filter, Cor. = indicates whether likelihood estimates are correlated or not.

the smallest datasets, the time difference is significant on the larger ones. Correspondingly, the Euler-Maruyama discretisation gives the best results on the sim(, 1) datasets, while the diffusion bridges are more efficient for the rest.

On the datasets where results for the naive combination are available, a significant increase in relative efficiency (from the naive combination to the best one) is obtained. Interestingly, this improvement is less on the sim(10, 1) and sim(100, 1) datasets. For the latter, the value of N is the same for the prior and Laplace-MDB importance densities. On the real data, N reduces from 200 to 4 in the uncorrelated naive case, and from 90 to 3 in the correlated one. A corresponding 2.75-fold decrease in time is observed from the correlated naive to the best combination.

6.2 CWPM

For CWPM, it is only necessary to select a proposal function and find a value for N. Again, this is done through experimentation. Tables 6–7 show results for all datasets. Dashed lines indicate that the memory limit of 150 GB of RAM per likelihood is exceeded. Due to this limit, no results could be obtained for the sim(1000, 1) dataset.

For all datasets, the correlated versions give the best results. Since the correlation induced is approximately $1 - 1/M$, the relative gain in efficiency increases with the number of subjects. In contrast, the number of particles needed for the standard versions grows quickly with the size of the dataset.

As with IAPM, the most efficient proposal function depends on the number of observations per subject. MDB/RB, MDB and EMD gives the best results for the sim(, 24), sim(, 12) and sim(, 1) datasets respectively. For the latter, any benefit in statistical efficiency from the bridges is outweighed by the increase in computation time. The MDB and RB also give the best results on the real data.

6.3 MPM

This method uses the same log-likelihood estimate as CWPM, so no extra tuning is required. When $N > 1$, we use the same number of particles for the conditional particle filter as for the standard. When $N = 1$, as for the real data (see Table 6), we add an extra particle to account for the invariant path.

7 MCMC Results

We use the time per log-likelihood estimate from Section 6 to determine which methods to run, i.e. ≤ 2 seconds for IAPM, ≤ 1 second for CWPM and ≤ 0.5 second for MPM. The best proposal function and importance density (for IAPM) from Section 6 is also used. Where the MDB and RB proposal functions give similar results, the MDB is preferred. Due to the time constraints, the naive method (uncorrelated IAPM with prior + EMD) is only run on the real and sim(10, 24) datasets. No results are obtained for the sim(100,1) or sim(1000, 1) datasets.

Each of the methods are run for 100,000 iterations starting at the same value of $\boldsymbol{\theta}$ that was used in Section 6. We use random walk proposals for the parameters which cannot be updated directly, i.e. those updated with a PMMH step. In CWPM and MPM, we also use pre-conditioned MALA to update the random effects hyperparameters $\{\mu_{X0}, \sigma_{X0}, \mu_\beta, \sigma_\beta\}$, and in MPM, we use a slice sampler to update σ. Tuning parameters for these proposals include the random walk covariance (also used as the MALA pre-conditioning matrix), and the stepsize for MALA. These values are tuned through experimentation.

We compare the methods based on the multivariate effective sample size (multiESS) (Vats et al., 2015) of $\boldsymbol{\theta}$ and the computation time in minutes. A score for each method is calculated as the approximate rate of independent samples per minute (multiESS/time). Table 8 shows the score for each method. Table 1 in Appendix A shows the breakdown of the multiESS for each update block. Tables 2 and 3 in Appendix A show the acceptance rates (AR) for the three methods on all datasets and Figure 1 in Appendix A shows the marginal posteriors of $\boldsymbol{\theta}$ for all datasets. As expected, the marginal posteriors become more precise as the size of the dataset grows (via more subjects and/or more densely observed time series). The jagged marginal posteriors for the sim(1000, 12) dataset may be due to Monte Carlo error as the multiESS for $\{\gamma, \sigma, \rho\}$ is relatively small.

	real				sim(10,24)			
	Naive	IAPM	CWPM	MPM	Naive	IAPM	CWPM	MPM
MultiESS	802	733	1431	1719	2439	1289	3064	3371
time (min)	669	77	6	41	4802	142	180	448
MultiESS/time	1.20	9.48	242.90	41.82	0.51	9.11	17.00	7.53
	sim(10,12)				sim(10,1)			
	Naive	IAPM	CWPM	MPM	Naive	IAPM	CWPM	MPM
MultiESS	–	1504	3028	4503	–	–	3197	5607
time (min)	–	351	211	479	–	–	2127	4786
MultiESS/time	–	4.29	14.38	9.39	–	–	1.50	1.17
	sim(100,24)				sim(100,12)			
	Naive	IAPM	CWPM	MPM	Naive	IAPM	CWPM	MPM
MultiESS	–	1174	3012	3663	–	1181	2485	3541
time (min)	–	971	430	1088	–	2849	706	1634
MultiESS/time	–	1.21	7.01	3.37	–	0.41	3.52	2.17
	sim(1000,24)				sim(1000,12)			
	Naive	IAPM	CWPM	MPM	Naive	IAPM	CWPM	MPM
MultiESS	–	–	2742	3402	–	–	1875	–
time (min)	–	–	1609	4644	–	–	3158	–
MultiESS/time	–	–	1.70	0.73	–	–	0.59	–

Table 8: MCMC results for all methods on all datasets. Results are calculated from chains of length 100,000. Dashed lines indicate that the method was not computationally feasible on that particular dataset.

There is a large increase in multiESS between IAPM, and CWPM and MPM on all datasets. This is partly due to the higher multiESS for the $\boldsymbol{X_0}$ hyperparameters (see Table 1 in Appendix A) and the more efficient proposals used for $\boldsymbol{\phi_\eta}$ and σ (in MPM). For both the real and synthetic data, MPM gives the highest multiESS, followed by CWPM. Table 8 shows that CWPM has the largest score due to its relatively short computation time. In general, CWPM runs much faster than the other two methods.

8 Discussion

We introduced three methods for simulation consistent parameter inference of state-space SDEMEMs and outlined some strategies for improving the efficiency of the likelihood estimate for these methods through the choice of importance density and proposal function. The efficiency of the calculation is generally also increased by correlating successive log-likelihood estimates.

Wiqvist et al. (July 23, 2019) concurrently and independently introduced a method for SDEMEMs that is very similar to our CWPM method. They propose the same update blocks for the parameters as in CWPM and give three variations of this approach; namely naive Gibbs, blocked Gibbs and a correlated PMMH method. In the first, the random numbers $\boldsymbol{u}$ are updated whenever the likelihood is estimated. In blocked Gibbs,

$\boldsymbol{u}$ is updated with the random effects but kept fixed for the other parameter blocks. Lastly, their correlated PMMH method uses the approach of Deligiannidis et al. (2018) to correlate the likelihoods, i.e. by correlating the random numbers (see Section 3.2).

Our approach differs in that we use the block pseudo-marginal (BPM) method of Tran et al. (2016). For mixed effects models, BPM has a number of advantages over CPM: a) it is simple to implement; b) it induces more directly the correlation between the estimates of the log-likelihood; c) it is much more straightforward to use with RQMC; d) its efficient implementation only requires the random seed to be stored, which can greatly reduce the computational storage requirements. A drawback of BPM is that the correlation is limited by the number of subjects. If there are few subjects, then CPM may be more effective at inducing correlation. An attractive option in this case is to combine BPM with CPM, i.e. correlate the auxiliary variables in the current block, while keeping the rest fixed. The feasibility of this approach is an area of future research. Unlike Wiqvist et al. (2019), we have not explored different strategies to update the random numbers; the approach we use in our example in Sections 5–7 most closely follows their naive Gibbs approach.

To further improve efficiency, we exploit bridge proposals in the particle filter rather than proposing directly from the (approximate) transition density as in the standard bootstrap filter used by Wiqvist et al. (2019). By including the IAPM and MPM methods, our paper provides a more comprehensive suite of particle methods for application to general state-space SDEMEMs. Wiqvist et al. (2019) allow the number of particles to vary between individuals, which is also straightforward to implement in our methods; see also Tran et al. (2013).

The IAPM, CWPM and MPM methods are much more efficient than the naive method; for the majority of the simulated datasets, the naive approach is computationally infeasible. The best method to use greatly depends on the model and data; IAPM is a good choice when there are few parameters, while CWPM and MPM may be preferable when there are many parameters; see Gunawan et al. (2018a). These methods are also flexible in the sense that they can be tailored to a specific model and used in combination, e.g. by integrating over a subset of the random effects using IAPM, but updating the rest using CWPM or MPM steps. Note that if IAPM is combined with MPM, then the invariant path from the conditional particle filter may be used for $\widehat{\boldsymbol{x}}_m$ in the importance sampler. CWPM gives the best results for the example in Sections 5–7. In general, this method has the shortest computation time and is the easiest to tune; however, as noted before, care must be taken if high correlation exists between the random effects and model parameters.

Tables 1 and 2 in Section 4.4 summarize our recommendations on how to set the tuning parameters in the new sampling methods. The optimal selection of the tuning parameters is beyond the scope of our article, and is the subject of our ongoing research; we note as well that there are very few optimal results for tuning parameters in particle Metropolis within Gibbs MCMC sampling schemes.

Appendix B applies our methods to an SDEMEM based on an Ornstein-Uhlenbeck (OU) process and compares the exact posterior, i.e. one obtained without discretisation

(as the OU process has a computable transition density) with the posterior obtained by the Euler-Maruyama approximation using $D = 10$. Both versions give the same marginal posteriors for all methods; however, the discretised versions take longer than the exact ones. CWPM gives the best results here as well.

Lastly, zero-variance control variates (Mira et al., 2013; Friel et al., 2016; South et al., 2019) may be used to further reduce the variance of any expectation estimated from the chains, e.g. the expectation of the target with respect to the auxiliary variables. Efficiency of the methods may also be increased through non-centered parameterisations of the random effects $\boldsymbol{\eta}_{1:M}$ (Papaspiliopoulos et al., 2007).

The new methods can be applied to a large number of SDEMEMs. The example in Sections 5–7 applies the methods to monotonic data fitted using an SDEMEM based on exponential growth. The example in Appendix B applies our methods to an Ornstein-Uhlenbeck model on a simulated non-monotonic dataset. The choice of models in these examples is ad-hoc; however, in practice, the performance of each of the sampling methods depends on both the properties of the underlying ODE as well as the methods. The SDE can be viewed as a prior for the unknown signal; as such, the underlying ODE should reflect key characteristics of the data, i.e. whether it is monotonic or has some other features such as periodicity, e.g. see Ansley et al. (1993). While model selection is outside the scope of this paper, it is an interesting area of future research.

Supplementary Material

Supplementary Material for "Particle Methods for Stochastic Differential Equation Mixed Effects Models" (DOI: 10.1214/20-BA1216SUPP; .pdf). Appendix A contains extra results for the example in Sections 5–7 and Appendix B gives a second example of our methods applied to non-monotonic data.

References

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. 576, 583, 585

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. 576, 583

Ansley, C. F., Kohn, R., and Wong, C.-M. (1993). "Nonparametric spline regression with prior information." *Biometrika*, 80(1): 75–88. MR1225215. doi: https://doi.org/10.1093/biomet/80.1.75. 605

Botha, I., Kohn, R., and Drovandi, C. (2020). "Supplementary Material for "Particle Methods for Stochastic Differential Equation Mixed Effects Models"." *Bayesian Analysis*. doi: https://doi.org/10.1214/20-BA1216SUPP. 576

Carpenter, J., Clifford, P., and Fearnhead, P. (1999). "Improved particle filter for nonlinear problems." *IEE Proceedings – Radar, Sonar and Navigation*, 146(1): 2–7. 582

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. 585

Del Moral, P., Doucet, A., and Jasra, A. (2006). "Sequential Monte Carlo samplers." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3): 411–436. MR2278333. doi: https://doi.org/10.1111/j.1467-9868.2006.00553.x. 582

Delattre, M., Genon-Catalot, V., and Samson, A. (2013). "Maximum likelihood estimation for stochastic differential equations with random effects." *Scandinavian Journal of Statistics*, 40(2): 322–343. MR3066417. doi: https://doi.org/10.1111/j.1467-9469.2012.00813.x. 575

Deligiannidis, G., Doucet, A., and Pitt, M. K. (2018). "The correlated pseudomarginal method." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(5): 839–870. MR3874301. doi: https://doi.org/10.1111/rssb.12280. 585, 592, 604

Donnet, S., Foulley, J.-L., and Samson, A. (2010). "Bayesian analysis of growth curves using mixed models defined by stochastic differential equations." *Biometrics*, 66(3): 733–741. MR2758209. doi: https://doi.org/10.1111/j.1541-0420.2009.01342.x. 575, 576

Donnet, S. and Samson, A. (2013a). "A review on estimation of stochastic differential equations for pharmacokinetic/pharmacodynamic models." *Advanced Drug Delivery Reviews*, 65(7): 929–939. 575, 576

Donnet, S. and Samson, A. (2013b). "Using PMCMC in EM algorithm for stochastic mixed models: theoretical and practical issues." *Journal de la Société Française de Statistique*, 155(1): 49–72. MR3199550. doi: https://doi.org/10.1007/s10955-014-0946-6. 576

Doucet, A., Godsill, S., and Andrieu, C. (2000). "On sequential Monte Carlo sampling methods for Bayesian filtering." *Statistics and Computing*, 10(3): 197–208. doi: https://doi.org/10.1023/A:1008935410038. 582

Doucet, A. and Johansen, A. M. (2009). "A tutorial on particle filtering and smoothing: Fifteen years later." *Handbook of Nonlinear Filtering*, 12(656–704): 3. MR2884612. 582

Doucet, A., Pitt, M., 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. 584, 592

Duan, J. A., Gelfand, A. E., Sirmans, C., et al. (2009). "Modeling space-time data using stochastic differential equations." *Bayesian Analysis*, 4(4): 733–758. MR2570086. doi: https://doi.org/10.1214/09-BA427. 575

Durham, G. B. and Gallant, A. R. (2002). "Numerical techniques for maximum likelihood estimation of continuous-time diffusion processes." *Journal of Business & Economic Statistics*, 20(3): 297–338. MR1939904. doi: https://doi.org/10.1198/073500102288618397. 580

Friel, N., Mira, A., and Oates, C. J. (2016). "Exploiting multi-core architectures for reduced-variance estimation with intractable likelihoods." *Bayesian Analysis*, 11(1): 215–245. MR3447097. doi: https://doi.org/10.1214/15-BA948. 605

Gerber, M., Chopin, N., and Whiteley, N. (2019). "Negative association, ordering and convergence of resampling methods." *The Annals of Statistics*, 47(4): 2236–2260. MR3953450. doi: https://doi.org/10.1214/18-AOS1746. 583

Golightly, A. and Wilkinson, D. J. (2008). "Bayesian inference for nonlinear multivariate diffusion models observed with error." *Computational Statistics & Data Analysis*, 52(3): 1674–1693. MR2422763. doi: https://doi.org/10.1016/j.csda.2007.05.019. 580

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. 582, 583

Gunawan, D., Carter, C., and Kohn, R. (2018a). "Efficiently Combining Pseudo Marginal and Particle Gibbs Sampling." *arXiv preprint* arXiv:1804.04359. 591, 604

Gunawan, D., Tran, M. N., Suzuki, K., Dick, J., and Kohn, R. (2016). "Computationally Efficient Bayesian Estimation of High Dimensional Copulas with Discrete and Mixed Margins." *arXiv preprint* arXiv:1608.06174. MR3994610. doi: https://doi.org/10.1007/s11222-018-9846-y. 590

Gunawan, D., Tran, M.-N., Suzuki, K., Dick, J., and Kohn, R. (2018b). "Computationally efficient Bayesian estimation of high-dimensional Archimedean copulas with discrete and mixed margins." *Statistics and Computing*, 29(5): 933–946. MR3994610. doi: https://doi.org/10.1007/s11222-018-9846-y. 576

Kitagawa, G. (1996). "Monte Carlo filter and smoother for non-Gaussian nonlinear state space models." *Journal of Computational and Graphical Statistics*, 5(1): 1–25. MR1380850. doi: https://doi.org/10.2307/1390750. 583

Leander, J., Almquist, J., Ahlström, C., Gabrielsson, J., and Jirstrand, M. (2015). "Mixed effects modeling using stochastic differential equations: illustrated by pharmacokinetic data of nicotinic acid in obese Zucker rats." *The AAPS Journal*, 17(3): 586–596. 575

L'Ecuyer, P. (2016). "Randomized quasi-Monte Carlo: an introduction for practitioners." In *12th International conference on Monte Carlo and quasi-Monte Carlo methods in scientific computing (MCQMC 2016)*. MR3828133. doi: https://doi.org/10.1007/978-3-319-91436-7_2. 590

Lindsten, F. and Schön, T. B. (2012). "On the use of backward simulation in the particle Gibbs sampler." In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, 3845–3848. IEEE. 586

Mira, A., Solgi, R., and Imparato, D. (2013). "Zero variance Markov chain Monte Carlo for Bayesian estimators." *Statistics and Computing*, 23(5): 653–662. MR3094805. doi: https://doi.org/10.1007/s11222-012-9344-6. 605

Øksendal, B. (2013). *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media. MR1411679. doi: https://doi.org/10.1007/978-3-662-03185-8. 577

Oravecz, Z., Tuerlinckx, F., and Vandekerckhove, J. (2011). "A hierarchical latent stochastic differential equation model for affective dynamics." *Psychological Methods*, 16(4): 468. 575

Papaspiliopoulos, O., Roberts, G. O., and Sköld, M. (2007). "A General Framework for the Parametrization of Hierarchical Models." *Statistical Science*, 22(1): 59–73. URL http://www.jstor.org/stable/27645805. MR2408661. doi: https://doi.org/10.1214/088342307000000014. 605

Picchini, U., De Gaetano, A., and Ditlevsen, S. (2010). "Stochastic differential mixed-effects models." *Scandinavian Journal of Statistics*, 37(1): 67–90. MR2675940. doi: https://doi.org/10.1111/j.1467-9469.2009.00665.x. 575

Picchini, U. and Ditlevsen, S. (2011). "Practical estimation of high dimensional stochastic differential mixed-effects models." *Computational Statistics & Data Analysis*, 55(3): 1426–1444. MR2741425. doi: https://doi.org/10.1016/j.csda.2010.10.003. 575

Picchini, U. and Forman, J. L. (2019). "Bayesian inference for stochastic differential equation mixed effects models of a tumour xenography study." *Journal of the Royal Statistical Society: Series C (Applied Statistics)*. MR4002376. 576, 595, 596

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. 584, 592

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. 576

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. 584, 592

South, L., Oates, C. J., Mira, A., and Drovandi, C. C. (2019). "Regularised Zero-Variance Control Variates for High-Dimensional Variance Reduction." *arXiv preprint* arXiv:1811.05073. 605

Stramer, O. and Bognar, M. (2011). "Bayesian inference for irreducible diffusion processes using the pseudo-marginal approach." *Bayesian Analysis*, 6(2): 231–258. MR2806243. doi: https://doi.org/10.1214/11-BA608. 576

Tran, M.-N., Kohn, R., Quiroz, M., and Villani, M. (2016). "The block pseudo-marginal sampler." *arXiv preprint* arXiv:1603.02485. 585, 590, 592, 594, 604

Tran, M.-N., Scharth, M., Pitt, M. K., and Kohn, R. (2013). "Importance sampling squared for Bayesian inference in latent variable models." *arXiv preprint* arXiv:1309.3339. 604

Vats, D., Flegal, J. M., and Jones, G. L. (2015). "Multivariate output analysis for Markov chain Monte Carlo." *arXiv preprint* arXiv:1512.07713. MR3653667. 602

Whitaker, G. A., Golightly, A., Boys, R. J., and Sherlock, C. (2017a). "Bayesian inference for diffusion-driven mixed-effects models." *Bayesian Analysis*, 12(2): 435–463. MR3620740. doi: https://doi.org/10.1214/16-BA1009. 576

Whitaker, G. A., Golightly, A., Boys, R. J., and Sherlock, C. (2017b). "Improved bridge constructs for stochastic differential equations." *Statistics and Computing*, 27(4): 885–900. MR3627552. doi: https://doi.org/10.1007/s11222-016-9660-3. 580, 581

Whiteley, N. (2010). "Discussion on particle Markov chain Monte Carlo methods." *Journal of the Royal Statistical Society: Series B*, 72(3): 306–307. 586

Wiqvist, S., Golightly, A., Mclean, T. A., and Picchini, U. (2019). "Efficient inference for stochastic differential mixed-effects models using correlated particle pseudo-marginal algorithms." *arXiv preprint* arXiv:1907.09851. 603, 604

Wood, S. N. (2010). "Statistical inference for noisy nonlinear ecological dynamic systems." *Nature*, 466(7310): 1102. 576

Acknowledgments

We thank Umberto Picchini and the research team at the Centre for Nanomedicine and Theranostics (DTU Nanotech, Denmark) for providing the real data and Andrew Golightly for useful feedback on an earlier draft of this paper. IB was supported by an Australian Reseach Training Program Stipend and an ACEMS Top-Up Scholarship. IB would also like to thank ACEMS for funding a trip to visit RK at UNSW where some of this research took place. CD was supported by an Australian Research Council Discovery Project (DP200102101). The work by RK was partially supported by an ARC Center of Excellence grant (CE140100049). We gratefully acknowledge the computational resources provided by QUT's High Performance Computing and Research Support Group (HPC).