Particle Methods for Stochastic Differential Equation Mixed Effects Models

Imke Botha[⋆,‡], Robert Kohn[†,‡], and Christopher Drovandi[⋆,‡]

[⋆]School of Mathematical Sciences, Queensland University of Technology
[†]School of Economics, University of New South Wales
[‡]Australian Research Council Centre of Excellence for Mathematical & Statistical Frontiers (ACEMS)

July 25, 2019

Abstract

Parameter inference for stochastic differential equation mixed effects models (SDEMEMs) is a challenging problem. Analytical solutions for these models are rarely available, which means that the likelihood is also intractable. In this case, exact inference is possible using the pseudo-marginal approach of Andrieu and Roberts (2009), where the intractable likelihood is replaced with a nonnegative unbiased estimate. A useful application of this idea is particle MCMC, which uses a particle filter estimate of the likelihood. While the exact posterior is targeted by these methods, a naive implementation for SDEMEMs can be highly inefficient. We develop three extensions to the naive approach which exploits specific aspects of SDEMEMs and other advances such as correlated pseudo-marginal methods. We compare these methods on real and simulated data from a tumour xenography study on mice.

1 Introduction

Stochastic differential equations (SDEs) may be 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. Assuming the data cannot be measured directly gives a state-space 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), which renders the likelihood intractable. The inclusion of random effects adds further complexity.

Parameter inference for SDEMEMs has largely been focussed on maximum likelihood estimation, see Picchini et al. (2010); Picchini and Ditlevsen (2011); Delattre et al. (2013); Donnet and Samson (2013a,b). Bayesian inference methods are comparatively scarce. Donnet et al. (2010) propose a Gibbs sampler coupled with an Euler-Maruyama discretization of the intractable transition density. This approach targets an approximation to the posterior, whose error can be controlled for some models. 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.

Here we focus on significant extensions to the pseudo-marginal approach of Picchini and Forman (2019) for SDEMEMs. Pseudo-marginal methods have been shown to overcome some limitations of data augmentation approaches, especially when there is substantial correlation between latent variables and the parameter of interest (Stramer and Bognar, 2011; Gunawan et al., 2018b). Our article develops a suite of new and efficient Bayesian methods for SDEMEMs by taking advantage of advances in pseudo-marginal methods and exploiting specific aspects of SDEMEMs. As a by-product we compare the performance of a collection of pseudo-marginal methods for our models of interest. The results of this comparison will be of interest to the wider computational Bayesian community. 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.

The paper is organised as follows. Sections 2 and 3 provides the necessary background on state space models, stochastic differential equations, particle filters and pseudo-marginal methods. Section 4 proposes three potential pseudo-marginal methods for SDEMEMs. Section 5 compares our methods with the approach of Picchini and Forman (2019) on simulated and real data from a tumor xenography study on mice. Section 8 concludes with a discussion of the results and possible future work.

2 Stochastic Differential Equation Mixed Effects Models

We denote random variables by capital letters and their realisations by lowercase letters, and $\mathbb{N}$ is the set of positive integers.

2.1 State Space Models

State space models (SSMs) consist of two processes: a hidden Markov process $\{X_t\}_{t\geq 0} \subset \mathcal{X}^{\mathbb{N}}$ (often referred to as the latent or unobserved variables) and an observed process $\{Y_t\}_{t\geq 0} \subset \mathcal{Y}^{\mathbb{N}}$. If the SSM is parameterised by $\boldsymbol{\theta}$, then the latent variables have initial density $\mu(x_0|\boldsymbol{\theta})$ and transition density

$$f(x_{t+1} \mid \boldsymbol{x}_{1:t}, \boldsymbol{\theta}) = f(x_{t+1} \mid x_t, \boldsymbol{\theta}),$$

i.e. x_{t+1} is conditionally independent of $\boldsymbol{x}_{1:t-1}$ given x_t. Observations $y_t \in \{Y_t\}_{t\geq 0}$ are assumed conditionally independent given these states, so for $t = 0, \ldots, T-1$, the observation density has the form

$$g(y_t \mid \boldsymbol{x}_{0:T-1}, \boldsymbol{\theta}) = g(y_t \mid x_t, \boldsymbol{\theta}).$$

Since $\boldsymbol{\theta}$ is unknown, it is assigned a prior $\pi(\boldsymbol{\theta})$. The unnormalized posterior density of the latent states and model parameters is

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

where

$$P(\boldsymbol{y}_{0:T-1} \mid \boldsymbol{x}_{0:T-1}, \boldsymbol{\theta}) = \prod_{t=0}^{T-1} g(y_t \mid x_t, \boldsymbol{\theta})$$
$$P(\boldsymbol{x}_{0:T-1} \mid \boldsymbol{\theta}) = \mu(x_1 \mid \boldsymbol{\theta}) \prod_{t=1}^{T-1} f(x_t \mid x_{t-1}, \boldsymbol{\theta}).$$

If the aim is parameter inference for $\boldsymbol{\theta}$, then it may be preferable to sample from $P(\boldsymbol{\theta} \mid \boldsymbol{y}_{0:T-1})$, especially if high correlation exists between $\boldsymbol{\theta}$ and $\boldsymbol{x}_{0:T-1}$. The marginal posterior of $\boldsymbol{\theta}$ is

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

where

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

However, direct calculation of this integral is not always possible. For some models, inference may also be complicated by an intractable transition density, e.g. SDEs (see Section 2.2). While approximate methods can be used in this case, exact inference is still feasible if it is possible to simulate according to the transition density (see Section 3.2).

2.2 Stochastic Differential Equation Mixed Effects Models

It is possible to construct a stochastic differential equation (SDE) from an ordinary differential equation by adding noise or replacing one of the terms in the model by a stochastic process. For simplicity, we decribe a one-dimensional SDE, but our methods (introduced in Section 4) can be extended straightforwardly to higher dimensions. Given an Itô process $\{X_t\}_{t\geq 0}$, the general form for a 1-dimensional continuous SDE is

$$dX_t = \mu(X_t, \boldsymbol{\phi}_X, t)dt + \sigma(X_t, \boldsymbol{\phi}_X, t)dB_t, \quad X_0 = X_0(\boldsymbol{\phi}_X)$$

where $\mu(\cdot)$ is the drift, $\sigma(\cdot)$ the diffusion, $\boldsymbol{\phi}_X$ are the 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 $m = 1, \ldots, M$ individuals, such that $\{X_{m,t}\}_{t\geq 0}$. Denote the vector of fixed model parameters as $\boldsymbol{\phi}_X$ and the vector of subject specific parameters (random effects) as $\boldsymbol{\eta}_m$, where $\boldsymbol{\eta}_m \sim P(\boldsymbol{\phi}_\eta)$. Then a stochastic differential equation mixed effects model (SDEMEM) is given by,

$$dX_{m,t} = \mu(X_{m,t}, \boldsymbol{\phi}_X, \boldsymbol{\eta}_m)dt + \sigma(X_{m,t}, \boldsymbol{\phi}_X, \boldsymbol{\eta}_m)dB_{m,t}, \quad X_{m0} = X_{m0}(\boldsymbol{\phi}_X, \boldsymbol{\eta}_m). \tag{3}$$

Assuming the process $\{X_t\}_{t\geq 0}$ cannot be measured directly leads to the construction of a state-space model as defined in Section 2.1.

Let $y_{m,t} \in \{Y_{m,t}\}$ denote a noisy observation for individual $m = 1, \ldots, M$ at time $\tau_t, t = 0, \ldots, T_m - 1$, where T_m is the number of observations for individual m. It is assumed that observations are given by

$$Y_{m,t} \mid x_{m,t}, \sigma^2 \sim \mathcal{N}(Y_{m,t} \mid x_{m,t}, \sigma^2). \tag{4}$$

The transition density of the model is given by the solution of (3). If an analytical solution is not available, numerical methods can be used (some of these are discussed in Section 2.3). Let $\boldsymbol{\theta} = (\sigma, \boldsymbol{\phi}_X, \boldsymbol{\phi}_\eta)$, $\boldsymbol{Y}_m = Y_{m,0:T_m-1}$ and $\boldsymbol{X}_m = X_{m,0:T_m-1}$. The full posterior of the model can then be written as

$$P(\boldsymbol{\theta}, \boldsymbol{\eta}_{1:M}, \boldsymbol{X}_{1:M} \mid \boldsymbol{y}_{1:M}) = P(\boldsymbol{\theta}) \prod_{m=1}^{M} P(\boldsymbol{\eta}_m, \boldsymbol{X}_m \mid \boldsymbol{y}_m, \boldsymbol{\theta})$$

where

$$P(\boldsymbol{\eta}_m, \boldsymbol{X}_m \mid \boldsymbol{y}_m, \boldsymbol{\theta}) = f(\boldsymbol{Y}_m \mid \boldsymbol{x}_m, \boldsymbol{\theta}) P(\boldsymbol{X}_m \mid \boldsymbol{\theta}, \boldsymbol{\eta}_m) P(\boldsymbol{\eta}_m \mid \boldsymbol{\theta}).$$

2.3 SDE Simulation

Consider the SDEMEM for a single individual

$$dX_t = \mu(X_t, \boldsymbol{\phi}_X, \boldsymbol{\eta})dt + \sigma(X_t, \boldsymbol{\phi}_X, \boldsymbol{\eta})dB_t, \quad X_0 = X_0(\boldsymbol{\phi}_X, \boldsymbol{\eta}).$$

For notational simplicity, let $v(X_t, \boldsymbol{\phi}_X, \boldsymbol{\eta}) = \sigma^2(X_t, \boldsymbol{\phi}_X, \boldsymbol{\eta})$.

Since most SDEs cannot be solved exactly, approximate methods are needed. This section describes two common approaches for approximate simulation of SDEs.

2.3.1 Euler-Maruyama

The Euler-Maruyama (EM) discretization is the simplest method for approximately simulating from an SDE. Given a process $\{X_t\}_{t\geq 0}$, the time interval $[0, J]$ is split into D subintervals

$$0 = \tau_0 < \tau_1 < \cdots < \tau_k < \tau_{k+1} < \cdots < \tau_D = J, \quad \Delta\tau = \frac{J}{D}.$$

Assuming the drift and diffusion coefficients are locally constant,

$$\begin{aligned}\mu(X_{\tau_k}, \boldsymbol{\phi}_X, \boldsymbol{\eta}) &= \mu_k \\ \sigma(X_{\tau_k}, \boldsymbol{\phi}_X, \boldsymbol{\eta}) &= \sigma_k, \quad v_k = \sigma_k^2,\end{aligned}$$

the EM method simulates over each subinterval as follows

$$\begin{aligned}X_{\tau_{k+1}} &= X_{\tau_k} + \mu_k \Delta\tau + \sigma_k \Delta B_{\tau_k} \\ \Delta B_{\tau_k} &= \tau_{k+1} - B_{\tau_k}.\end{aligned}$$

Since $\Delta B_{\tau_k} \sim \mathcal{N}(0, \Delta\tau)$ by definition, the path is simulated through a recursive application of

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

Thus, the joint density of this approximate simulation can be written as

$$q(x_{\tau_1:J} \mid x_0, \phi_X, \eta) \propto \prod_{k=0}^{D-1} \mathcal{N}(x_{\tau_{k+1}}; x_{\tau_k} + \mu_k \Delta\tau, v_k \Delta\tau).$$

Note that for an SDE with constant drift and diffusion, the EM method gives the exact solution.

2.3.2 Diffusion Bridges

Simulating from the (approximate) transition density may not perform well in pseudo-marginal methods if any particular observations are highly informative or there is little observation noise. More efficient estimates of the likelihood of SSMs can be achieved if the proposal for x_t can be directed towards y_t. This can be done with a diffusion bridge. The modified diffusion bridge (MDB) of Durham and Gallant (2002) (see also Golightly and Wilkinson, 2008) is derived by approximating the joint distribution of $X_{\tau_{k+1}}, Y_J \mid x_{\tau_k}$ using multivariate Normal theory, and then conditioning on $Y_J = y_J$. The density $X_{\tau_{k+1}}, Y_J \mid x_{\tau_k}$ is obtained from the observation density (4) and an EM approximation of $X_{\tau_{k+1}} \mid x_{\tau_k}$. See Appendix 1 of Golightly and Wilkinson (2008) for a more detailed derivation. The MDB gives a bridge proposal of the form

$$X_{\tau_{k+1}} \mid x_{\tau_k}, y_J \sim \mathcal{N}\{x_{\tau_k} + \mu_{\text{MDB}}(x_{\tau_k}, y_J)\Delta\tau, \Psi_{\text{MDB}}(x_{\tau_k})\Delta\tau\}$$

where

$$\begin{aligned}\mu_{\text{MDB}}(x_{\tau_k}, y_J) &= \mu_k + \frac{v_k(y_J - (x_{\tau_k} + \mu_k \Delta_k))}{v_k \Delta_k + \sigma^2} \\ \Psi_{\text{MDB}}(x_{\tau_k}) &= v_k - \frac{v_k^2 \Delta\tau}{v_k \Delta_k + \sigma^2} \\ \Delta_k &= J - \tau_k.\end{aligned}$$

As noted by Whitaker et al. (2017b), the modified diffusion bridge can perform poorly when the drift coefficient is not constant within each interval. To overcome this problem, they propose to partition the SDE into a deterministic and residual stochastic process, such that the latter has constant drift. Rewriting the model in terms of a residual stochastic process gives

$$\begin{aligned} X_t &= \zeta_t + R_t, & \zeta_t, t \geq 0 \\ d\zeta_t &= f(\zeta_t)dt, & \zeta_0 = x_0 \\ dR_t &= \{\mu(X_{\tau_k}, \boldsymbol{\phi}_X, \boldsymbol{\eta}) - f(\zeta_t)\}dt + \sigma(X_{\tau_k}, \boldsymbol{\phi}_X, \boldsymbol{\eta})dB_t, & R_0 = 0. \end{aligned}$$

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

$$\begin{aligned} X_t &= \eta_t + R_t, & \eta_t, t \geq 0 \\ d\eta_t &= \mu(\eta_t, \boldsymbol{\phi}_X, \boldsymbol{\eta})dt, & \eta_0 = x_0 \\ dR_t &= \{\mu(X_{\tau_k}, \boldsymbol{\phi}_X, \boldsymbol{\eta}) - \mu(\eta_t, \boldsymbol{\phi}_X, \boldsymbol{\eta})\}dt + \sigma(X_{\tau_k}, \boldsymbol{\phi}_X, \boldsymbol{\eta})dB_t, & R_0 = 0. \end{aligned}$$

Note that $Y_J - \eta_J = R_J + \epsilon_J$. The residual bridge is obtained by constructing the MDB on the residual process $\{R_t\}$. This gives a proposal of the form

$$X_{\tau_{k+1}} \mid x_{\tau_k}, y_J \sim \mathcal{N}(x_{\tau_k} + \mu_{\text{RB}}(x_{\tau_k}, y_J)\Delta\tau, \Psi_{\text{RB}}(x_{\tau_k, y_J})\Delta\tau)$$

where

$$\begin{aligned} \Psi_{\text{RB}}(x_{\tau_k}, y_J) &= \Psi_{\text{MDB}}(x_{\tau_k}, y_J) \\ \mu_{\text{RB}}(x_{\tau_k}, y_J) &= \mu_k + \frac{v_k(y_J - (\eta_J + r_{\tau_k} + (\mu_k - \delta_k^\eta)\Delta k))}{v_k\Delta_k + \sigma^2} \\ \delta_k^\eta &= \frac{\eta_{\tau_{k+1}} - \eta_{\tau_k}}{\Delta\tau}. \end{aligned}$$

3 Pseudo-Marginal Methods

3.1 Particle Filters

Exact state estimation of SSMs using the Kalman filter is only possible for models with linear state transitions and a Gaussian observation density. In the case of non-linear, non-Gaussian SSMs, a particle filter can be used for simulation consistent of conditionally Gaussian 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 filtering distribution of the form

$$\pi_t(\boldsymbol{x}_{1:t} \mid \boldsymbol{y}_{1:t}, \boldsymbol{\theta}) \propto P(y_1|x_1, \boldsymbol{\theta})P(x_1 \mid \boldsymbol{\theta}) \prod_{j=2}^{t} P(y_j|x_j, \boldsymbol{\theta})P(x_j \mid x_{j-1}, \boldsymbol{\theta}), \quad t = 1, \ldots, T, \quad (5)$$

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,

$$w_t^n = W_{t-1}^n \frac{\pi_t(x_t \mid y_t, \boldsymbol{\theta})}{q(x_t \mid y_t, x_{t-1}, \boldsymbol{\theta})}, \quad W_1^n = \frac{1}{N},$$

where $W_t^n = \frac{w_t^n}{\sum_{i=1}^N w_t^i}$ are the normalised weights at iteration t. Particles are then resampled with probability $\boldsymbol{W}^{1:N}$ for the next iteration. This is done to avoid the particle impoverishment problem, 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, Srinivasan (Gerber et al., 2019).

A key advantage of particle filters is that an unbiased estimate of the likelihood may be obtained from the unnormalized weights,

$$\hat{P}(\boldsymbol{Y}_{1:T} \mid \boldsymbol{\theta}) = \prod_{t=1}^{T} \sum_{n=1}^{N} w_t^{(n)}.$$

A useful variation of the generic filter is the bootstrap particle filter (BPF) of Gordon et al. (1993), where one proposes from the transition density $q(x_t \mid y_t, x_{t-1}, \boldsymbol{\theta}) = p(x_t \mid x_{t-1}, \boldsymbol{\theta})$. The calculation of the weights then simplify to

$$w_t^n = W_{t-1}^n P(y_t \mid x_t, \boldsymbol{\theta}).$$

Input: data $\boldsymbol{y}_{1:T}$, the number of particles N, the static parameters $\boldsymbol{\theta}$ and the initial state x_0. We use the convention that index (n) means 'for all $n \in \{1, \ldots, N\}$'

Initialise $x_1^{(n)} = x_0,\ W_1^{(n)} = \frac{1}{N},\ w_1^{(n)} = W_1^{(n)} P(y_1 | x_1^{(n)}, \boldsymbol{\theta}),\ Z = \sum_{n=1}^N w_1^{(n)}$

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

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

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

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

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

  Update likelihood estimate $Z = Z \times \sum_{n=1}^N w_t^{(n)}$

end

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

3.2 Pseudo-Marginal MCMC

The pseudo-marginal approach of Andrieu and Roberts (2009) allows for exact inference for models with intractable likelihoods. Andrieu and Roberts (2009) proved that an MCMC algorithm will target the exact posterior if the intractable likelihood is replaced with an nonnegative unbiased estimate. Two pseudo-marginal algorithms for state-space models are the particle marginal Metropolis-Hastings (PMMH) and particle Gibbs (PG) algorithms (see Andrieu et al., 2010).

3.2.1 Particle Marginal Metropolis-Hastings

The PMMH method is a Metropolis-Hastings algorithm where the intractable likelihood is replaced with an unbiased particle filter estimate (see Section 3.1). 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 to estimate the likelihood. Unbiasedness implies the target of interest $P(\boldsymbol{\theta} \mid \boldsymbol{y}_{1:T})$ is obtained through marginalisation.

Initialise $\boldsymbol{\theta}^{(0)}$
Run Algorithm 1 to obtain an unbiased estimate of $\hat{P}(\boldsymbol{y}_{1:T} \mid \boldsymbol{\theta}^{(0)})$
for $i = 1$ to $I - 1$ **do**
  Sample $\boldsymbol{\theta}^* \sim q(\cdot \mid \boldsymbol{\theta}^{(i-1)})$
  Run Algorithm 1 to obtain an unbiased estimate of $\hat{P}(\boldsymbol{y}_{1:T} \mid \boldsymbol{\theta}^*)$
  Calculate the Metropolis-Hastings ratio

$$\text{MHR} = \frac{\hat{P}(\boldsymbol{y}_{1:T} \mid \boldsymbol{\theta}^*)P(\boldsymbol{\theta}^*)}{\hat{P}(\boldsymbol{y}_{1:T} \mid \boldsymbol{\theta}^{(i-1)})P(\boldsymbol{\theta}^{(i-1)})} \frac{q(\boldsymbol{\theta}^{(i-1)} \mid \boldsymbol{\theta}^*)}{q(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}^{(i-1)})}$$

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

Algorithm 2: Particle marginal Metropolis-Hastings.

The main drawback of the PMMH algorithm is the chain's tendency to get stuck when the likelihood is greatly overestimated for particular values of $\boldsymbol{\theta}$. This can be mitigated by increasing the precision of the likelihood estimate, i.e. by using more particles in the particle filter. Sherlock et al. (2015), Pitt et al. (2012) and Doucet et al. (2015) showed that optimal performance (for random walk proposals) is gained when the standard deviation of the estimated log-likelihood is between 1 and 2. An alternate approach is the correlated pseudo-marginal (CPM) method of Dahlin et al. (2015) and Deligiannidis et al. (2018). Tran et al. (2016) introduced a variation of the CPM method called the block pseudo-marginal (BPM) approach.

3.2.2 Correlated Pseudo-Marginal

Dahlin et al. (2015) and Deligiannidis et al. (2018) showed that the mixing of the chain can be improved by correlating successive likelihood estimates, i.e. if the likelihood is overestimated at the current iteration, it will also be overestimated at the next. Recall, the chain targets the density $p(\boldsymbol{\theta}, \boldsymbol{u}|\boldsymbol{y}_{1:T})$. At iteration i and $i+1$, the estimates returned are proportional to $\hat{P}(\boldsymbol{\theta}^{(i)}, \boldsymbol{u}^{(i)} \mid \boldsymbol{y}_{1:T})$ and $\hat{P}(\boldsymbol{\theta}^{(i+1)}, \boldsymbol{u}^{(i+1)} \mid \boldsymbol{y}_{1:T})$. The CPM approach correlates these estimates by inducing positive correlation between $\boldsymbol{u}^{(i)}$ and $\boldsymbol{u}^{(i+1)}$. Assuming the random numbers are normally distributed, both Dahlin et al. (2015) and Deligiannidis et al. (2018) use the Crank-Nicolson (CN) proposal to induce the correlation

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

If the particle filter depends on non-normal random numbers, transformations to normality can be applied.

In BPM, correlation is induced by updating $\boldsymbol{u}$ in blocks (Tran et al., 2016). In this approach, the vector of random numbers is divided into B blocks, and a single block is updated at each iteration while the remaining $B-1$ are held constant. No assumption about the form or distribution of $\boldsymbol{u}$ is required.

Relative to standard PMMH, both 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 $\boldsymbol{u}$.

The number of particles N needed for the correlated methods can be tuned using the log-likelihood ratio (Deligiannidis et al., 2018)

$$R = \log\left(\frac{\hat{P}(\boldsymbol{y}_{1:T} \mid \boldsymbol{\theta}^*)}{\hat{P}(\boldsymbol{y}_{1:T} \mid \boldsymbol{\theta})}\right). \tag{6}$$

To minimize the distance between successive log-likelihood estimates, N may be chosen such that the variance of R is around 1.

3.2.3 Conditional Particle Filter

The second algorithm of Andrieu et al. (2010), particle Gibbs (PG), requires a variation of the generic PF (Section 3.1), called the conditional particle filter (CPF). The CPF differs from the generic PF by holding a single path $\boldsymbol{X}_{1:T}^k$, invariant through the iterations. See Algorithm 3 for more detail. Once the particles have been sampled, the backwards sampling method of Whiteley (2010) and Lindsten and Schön (2012) may be used to draw a new invariant path, see Algorithm 4.

Input: data $\boldsymbol{y}_{1:T}$, the number of particles N, the initial state x_0, the static parameters $\boldsymbol{\theta}$, invariant path $\boldsymbol{x}_{1:T}^k$ and associated ancestral lineage $\boldsymbol{b}_{1:T}^k$. We use the convention that index $(n \neq k)$ means 'for all $n \in \{1, \ldots, k-1, k+1, \ldots, N\}$'

Initialise $x_1^{(n \neq B_1^k)} = x_0,\ W_1^{(n)} = \frac{1}{N}, w_1^{(n)} = W_1^{(n)} P(y_1 \mid x_1^{(n)}, \boldsymbol{\theta}), Z = \sum_{n=1}^N w_1^{(n)}$

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

  Sample $A_{t-1}^{(n \neq B_1^k)} \sim \mathcal{F}(\cdot \mid W_{t-1}^{(n)})$

  Sample $x_t^{(n \neq B_1^k)} \sim q(\cdot \mid x_{t-1}^{A_{t-1}^{B_1^n}}, \boldsymbol{\theta})$

  Calculate weights $w_t^{(n)} = W_{t-1}^{(n)} \frac{\pi_t(x_t^{(n)}, \boldsymbol{\theta})}{q(x_t^{(n)} \mid x_{t-1}^{(n)}, \boldsymbol{\theta})}$

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

end

Run Algorithm 4 to obtain new ancestral lineage $\boldsymbol{b}_{1:T}^{k^*}$

Use $\boldsymbol{b}_{1:T}^{k^*}$ to determine new path $\boldsymbol{x}_{1:T}^{k^*}$

Algorithm 3: The conditional particle filter. The matrix A_{t-1}^n gives the parent index of particle n at time $t-1$. The relationship between the ancestral lineage and the matrix of parent indices is $A_{t-1}^{B_t^k} = B_{t-1}^k$, where $B_T^k = k$.

Input : $\boldsymbol{w}_{1:T}^{(n)},\ \boldsymbol{W}_T^{(n)}$
Output: a new ancestral lineage $\boldsymbol{B}_{1:T}^{k^*}$

Draw $k^* \sim \mathcal{F}(\cdot \mid W_T)$
Set $B_T^{k^*} = k^*$
for $t = T-1$ to 1 **do**
  Sample $W_{(t|T)}^{(n)} = w_t^{(n)} \dfrac{f_{\boldsymbol{\theta}}\left(x_{t+1}^{B_{t+1}^{k^*}} | x_t^{(n)}\right)}{\sum_{i=1}^{N} w_t^{(i)} f_{\boldsymbol{\theta}}\left(x_{t+1}^{B_{t+1}^{k^*}} | x_t^{(i)}\right)}$
  Draw $B_t^{k^*} \sim \mathcal{F}(\cdot \mid W_{(t|T)})$
end

Algorithm 4: Backward Sampling.

3.2.4 Particle Gibbs

In PMMH, the particle filter returns an estimate of the likelihood (2). 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 static parameters $\boldsymbol{\theta}$ may be updated using Gibbs sampling if the full conditional posterior is available, or a Metropolis-Hastings step if it is not.

Initialise $\boldsymbol{\theta}^0, \boldsymbol{x}_{1:T}^{(0)}$ and associated ancestral lineage $\boldsymbol{b}_{1:T}^{(0)}$
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

Algorithm 5: The particle Gibbs algorithm.

Since a new path $\boldsymbol{X}_{1:T}$ is simulated at each iteration, the PG method does not suffer from the same mixing problem as PMMH. As such, 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 MALA or HMC. While this method has a number of advantages over PMMH it is not as generally applicable, as a closed form transition density is required to update $\boldsymbol{\theta}$.

4 Methods

We are interested in parameter inference for a state-space SDEMEM as described in Section 2.2. For a single individual m, with observations taken at $\tau_{m,t}, t = 0, \ldots, T_m - 1$, the sequence of distributions (5) traversed by the particle filter (see Section 3.1) is

$$\pi_t(\boldsymbol{x}_{m,0:t} \mid \boldsymbol{y}_{m,0:t}, \boldsymbol{\eta}_m, \sigma, \boldsymbol{\phi}_X) \propto P(y_{m0}|x_{m0}, \sigma) P(x_{m0} \mid \boldsymbol{\eta}_m, \boldsymbol{\phi}_X) \prod_{j=1}^{t} P(y_{m,t}|x_{m,t}, \sigma) P(x_{m,t} \mid x_{m,t-1}, \boldsymbol{\eta}_m, \boldsymbol{\phi}_X).$$

This particle filter returns an estimate of $P(\boldsymbol{y}_m \mid \boldsymbol{\eta}_m, \sigma, \boldsymbol{\phi}_X)$. The estimated likehood for all the data $\boldsymbol{y}_{1:M}$ is given by

$$\hat{P}(\boldsymbol{y}_{1:M} \mid \boldsymbol{\eta}_{1:M}, \sigma, \boldsymbol{\phi}_X) = \prod_{m=1}^{M} \hat{P}(\boldsymbol{y}_m \mid \boldsymbol{\eta}_m, \sigma, \boldsymbol{\phi}_X).$$

4.1 Individual-Augmentation Pseudo-Marginal

The first method that will be covered is Individual-Augmentation Pseudo-Marginal (IAPM), named for the additional auxiliary variables required to estimate the likelihood for each individual. Here, we use the likelihood estimate,

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

within a PMMH algorithm (Algorithm 2). To do this, we replace step 5 in Algorithm 2 with the importance sampling step outlined in Algorithm 7. See Algorithm 6 for more detail.

initialise $\boldsymbol{\theta}^{(0)}$
Run Algorithm 7 to obtain likelihood estimate $\hat{P}(\boldsymbol{y}_{1:M} \mid \boldsymbol{\theta}^{(0)})$
for $i = 1$ to I **do**
    Draw $\boldsymbol{\theta}^* \sim q(\cdot \mid \boldsymbol{\theta}^{(i-1)})$
    Run Algorithm 7 to obtain initial likelihood estimate $\hat{P}(\boldsymbol{y}_{1:M} \mid \boldsymbol{\theta}^*)$
    Accept $\boldsymbol{\theta}^*$ with probability

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

end

Algorithm 6: The individual-augmentation pseudo-marginal (IAPM) method.

The accuracy of $\hat{P}(\boldsymbol{y}_m \mid \boldsymbol{\theta})$ is controlled by the number of particles N, as well as the number of random effects draws L. The choice of importance distribution has a significant impact on both N and L. A naive choice is to set $g(\boldsymbol{\eta}_m \mid \boldsymbol{\theta}) = P(\boldsymbol{\eta}_m \mid \boldsymbol{\theta})$ (Picchini and Forman, 2019). While this simplifies the likelihood calculation, it can be very inefficient if $\hat{P}(\boldsymbol{\eta}_m \mid \boldsymbol{y}_m, \boldsymbol{\theta})$ and $P(\boldsymbol{\eta}_m \mid \boldsymbol{\theta})$ are not similar. We propose instead to use a Laplace approximation of a distribution over $\boldsymbol{\eta}_m$ proportional to

$$P(\boldsymbol{y}_m \mid \hat{\boldsymbol{x}}_m, \boldsymbol{\theta}) P(\boldsymbol{\eta}_m \mid \boldsymbol{\theta}),$$

where $\hat{\boldsymbol{x}}_m$ is an approximation of $\boldsymbol{x}_m$. We present two choices for $\hat{\boldsymbol{x}}_{\boldsymbol{m}}$. The first is to use the solution of the ODE given by the drift of the SDEMEM (3),

$$d\hat{X}_{m,t} = \mu(\hat{X}_{m,t}, \boldsymbol{\phi}_X, t) dt.$$

1 **for** $m = 1$ to M **do**
2 **for** $l = 1$ to L **do**
3 Draw $\boldsymbol{\eta}_m^{(l)} \sim g(\cdot \mid \boldsymbol{\theta})$
4 Run Algorithm 1 with $\boldsymbol{\eta}_m^{(l)}$ to obtain likehood estimate $Z_m^{(l)}$
5 Correct for importance distribution $Z_m^{(l)} = \frac{Z_m^{(l)}}{g(\boldsymbol{\eta}_m^{(l)} \mid \boldsymbol{\theta})}$
6 **end**
7 Calculate $\hat{P}(\boldsymbol{y}_m \mid \boldsymbol{\theta}) = \frac{1}{L} \sum_{i=1}^{L} Z_m^{(i)}$
8 **end**
9 Calculate $\hat{P}(\boldsymbol{y}_{1:M} \mid \boldsymbol{\theta}) = \prod_{m=1}^{M} \hat{P}(\boldsymbol{y}_m \mid \boldsymbol{\theta})$

Algorithm 7: Estimating the likelihood for the IAPM algorithm.

The second approximates $\boldsymbol{x}_m$ using the mean of the modified diffusion bridge (see Section 2.3.2), with $\Delta_{\tau_{m,t}} = \Delta_t = \tau_{m,t} - \tau_{m,t-1}$, such that

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

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

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

A correlated version of IAPM (cIAPM) can be implemented using block pseudo-marginal as described in Section 3.2.2. In this case, the vector of random numbers is given by $\boldsymbol{u} = (\boldsymbol{u}_{\text{RE}}, \boldsymbol{u}_{\text{PF}})$, where $\boldsymbol{u}_{\text{RE}}$ and $\boldsymbol{u}_{\text{PF}}$ 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 $1 - \frac{1}{M}$ between successive log-likelihood estimates (Tran et al., 2016). Since BPM makes no assumptions about the distribution of $\boldsymbol{u}$, RQMC can also be used within cIAPM.

4.2 Component-Wise Pseudo-Marginal

In this section, we define a component-wise pseudo-marginal (CWPM) method, where a Markov chain is constructed for the random effects $\boldsymbol{\eta}_{1:M}$, as well as the static parameters $\boldsymbol{\theta}$ (Algorithm 8). Recall, $\boldsymbol{\theta} = (\sigma, \boldsymbol{\phi}_X, \boldsymbol{\phi}_\eta)$. If we denote $\boldsymbol{\theta}_X = (\sigma, \boldsymbol{\phi}_X)$, then the joint posterior is of the form

$$P(\boldsymbol{\theta}_X, \boldsymbol{\phi}_\eta, \boldsymbol{\eta}_{1:M} \mid \boldsymbol{y}_{1:M}) \propto P(\boldsymbol{y}_{1:M} \mid \boldsymbol{\eta}_{1:M}, \boldsymbol{\theta}_X) P(\boldsymbol{\eta}_{1:M} \mid \boldsymbol{\phi}_\eta) P(\boldsymbol{\theta}_X) P(\boldsymbol{\phi}_\eta).$$

Similarly, the full conditional posteriors for each of these parameters are

$$\begin{aligned} P(\boldsymbol{\eta}_m \mid \boldsymbol{y}_{1:M}, \boldsymbol{\theta}_X, \phi_\eta) &\propto P(\boldsymbol{y}_{1:M} \mid \boldsymbol{\eta}_{1:M}, \boldsymbol{\theta}_X) P(\boldsymbol{\eta}_{1:M} \mid \boldsymbol{\phi}_\eta) \\ P(\boldsymbol{\theta}_X \mid \boldsymbol{y}_{1:M}, \boldsymbol{\eta}_{1:M}) &\propto P(\boldsymbol{y}_{1:M} \mid \boldsymbol{\eta}_{1:M}, \boldsymbol{\theta}_X) P(\boldsymbol{\theta}_X) \\ P(\boldsymbol{\phi}_\eta \mid \boldsymbol{\eta}_{1:M}) &\propto P(\boldsymbol{\eta}_{1:M} \mid \boldsymbol{\phi}_\eta) P(\boldsymbol{\phi}_\eta). \end{aligned} \tag{7}$$

The parameters are updated in the following blocks: $\boldsymbol{\eta}_{1:M}, \boldsymbol{\theta}_X$ and $\boldsymbol{\phi}_\eta$. When updating $\boldsymbol{\eta}_{1:M}$ and $\boldsymbol{\theta}_X$, a particle filter estimate of $P(\boldsymbol{y}_{1:M} \mid \boldsymbol{\eta}_{1:M}, \boldsymbol{\theta}_X)$ is used (Algorithm 1). Since

(7) is tractable, $\boldsymbol{\phi}_\eta$ can be sampled directly. This method will generally be faster than IAPM as the particle filter is called $2 \times M$ times per MCMC iteration (with the above configuration), instead of $L \times M$ as in Section 4.1. If there is high correlation between $\boldsymbol{\eta}_{1:M}$ and $\boldsymbol{\theta}$, however, the CWPM chain may exhibit poor mixing.

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.

initialise $\boldsymbol{\eta}_{1:M}^{(0)}, \boldsymbol{\theta}_X^{(0)}$ and $\boldsymbol{\phi}_\eta^{(0)}$
Run Algorithm 1 to obtain likelihood estimate $\hat{P}(\boldsymbol{y}_{1:M} \mid \boldsymbol{\eta}_{1:M}^{(0)}, \boldsymbol{\theta}_X^{(0)})$
for $i = 1$ to I **do**
  Draw $\boldsymbol{\eta}_{1:M}^* \sim q(\cdot \mid \boldsymbol{\eta}_{1:M}^{(i-1)})$
  Run Algorithm 1 to obtain likelihood estimate $\hat{P}(\boldsymbol{y}_{1:M} \mid \boldsymbol{\eta}_{1:M}^*, \boldsymbol{\theta}_X^{(i-1)})$
  Accept $\boldsymbol{\eta}_{1:M}^*$ with probability

$$\alpha = \min\left(1, \frac{\hat{P}\left(\boldsymbol{y}_{1:M} \mid \boldsymbol{\eta}_{1:M}^*, \boldsymbol{\theta}_X^{(i-1)}\right) P\left(\boldsymbol{\eta}_{1:M}^* \mid \boldsymbol{\phi}_\eta^{(i-1)}\right) q\left(\boldsymbol{\eta}_{1:M}^{(i-1)} \mid \boldsymbol{\eta}_{1:M}^*\right)}{\hat{P}\left(\boldsymbol{y}_{1:M} \mid \boldsymbol{\eta}_{1:M}^{(i-1)}, \boldsymbol{\theta}_X^{(i-1)}\right) P\left(\boldsymbol{\eta}_{1:M}^{(i-1)} \mid \boldsymbol{\phi}_\eta^{(i-1)}\right) q\left(\boldsymbol{\eta}_{1:M}^* \mid \boldsymbol{\eta}_{1:M}^{(i-1)}\right)}\right)$$

  Draw $\boldsymbol{\theta}_X^* \sim q(\cdot \mid \boldsymbol{\theta}_X^{(i-1)})$
  Run Algorithm 1 to obtain likelihood estimate $\hat{P}(\boldsymbol{y}_{1:M} \mid \boldsymbol{\eta}_{1:M}^{(i)}, \boldsymbol{\theta}_X^*)$
  Accept $\boldsymbol{\theta}_X^*$ with probability

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

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

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

end

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

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, depending on the

form of the full conditional distributions,

$$\begin{aligned} P(\boldsymbol{\eta}_{1:M} \mid \boldsymbol{y}_{1:M}, \sigma, \boldsymbol{\phi}_X, \boldsymbol{\phi}_\eta) &\propto \hat{P}(\boldsymbol{y}_{1:M} \mid \boldsymbol{\eta}_{1:M}, \sigma, \boldsymbol{\phi}_X) P(\boldsymbol{\eta}_{1:M} \mid \boldsymbol{\phi}_\eta) \quad (8) \\ P(\sigma \mid \boldsymbol{y}_{1:M}, \boldsymbol{x}_{1:M}) &\propto P(\boldsymbol{y}_{1:M} \mid \boldsymbol{x}_{1:M}, \sigma) P(\sigma) \\ P(\boldsymbol{\phi}_X \mid \boldsymbol{y}_{1:M}, \boldsymbol{\eta}_{1:M}, \sigma, \boldsymbol{\phi}_X) &\propto \hat{P}(\boldsymbol{y}_{1:M} \mid \boldsymbol{\eta}_{1:M}, \sigma, \boldsymbol{\phi}_X) P(\boldsymbol{\phi}_X) \quad (9) \\ P(\boldsymbol{\phi}_\eta \mid \boldsymbol{\eta}_{1:M}) &\propto P(\boldsymbol{\eta}_{1:M} \mid \boldsymbol{\phi}_\eta) P(\boldsymbol{\phi}_\eta). \end{aligned}$$

At each iteration, the invariant path $\boldsymbol{x}_{1:M}$ is updated using a conditional particle filter (Algorithm 3). Where the density $P(\boldsymbol{y}_{1:M} \mid \boldsymbol{\eta}_{1:M}, \sigma, \boldsymbol{\phi}_X)$ is required, i.e. (8) and (9), a particle filter estimate is used (PMMH step). Since the full conditionals for σ and $\boldsymbol{\phi}_\eta$ are tractable, these parameters can be sampled 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. 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 block PM, where $\boldsymbol{u}$ is divided into M blocks based on individuals $m = 1, \ldots, M$. The log-likelihood estimates used for (8) and (9) are correlated in exactly the same way as with CWPM.

4.4 Likelihood Estimation

We have introduced three pseudo-marginal methods for SDEMEMs: IAPM, CWPM and MPM. Each of these methods rely on a particle filter to calculate an unbiased estimate of the intractable likelihood. Tuning parameters for this calculation include the level of discretization (D), the number of particles (N) and, for IAPM, the number of random effects draws (L). We use the log-likelihood ratio R (6) as described in Section 3.2.2 to tune D, N and L. We denote the standard deviation of $|R|$ as σ_Δ and aim for $\sigma_\Delta \leq 1.05$.

It is also necessary to specify a propagation function for the particle filter and an importance density for IAPM. Section 2.3 describes three different ways to simulate from an SDE: the Euler-Maruyama (EM) discretization, 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 proposed the Laplace-ODE and Laplace-MDB importance densities for IAPM. Optimal choice of propagation function and importance density is problem specific and may have a large effect on the efficiency of the log-likelihood estimation.

5 Example

5.1 Data

We apply our methods to real data from a tumour xenography study on mice. This data was obtained from Picchini and Forman (2019). The study had 4 treatment groups and 1 control, 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 $1000mm^3$.

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

initialise $\boldsymbol{\eta}_{1:M}^{(0)}, \sigma^{(0)}, \boldsymbol{\phi}_X^{(0)}$ and $\boldsymbol{\phi}_\eta^{(0)}, \boldsymbol{x}_{1:M}^{(0)}$ and $\boldsymbol{b}_{1:M}^{(0)}$

Run Algorithm 1 to obtain likelihood estimate $\hat{P}(\boldsymbol{y}_{1:M} \mid \boldsymbol{\eta}_{1:M}^{(0)}, \sigma^{(0)}, \boldsymbol{\phi}_X^{(0)})$

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

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

Run Algorithm 1 with $\boldsymbol{u}^*$ to obtain likelihood estimate $\hat{P}(\boldsymbol{y}_{1:M} \mid \boldsymbol{\eta}_{1:M}^*, \sigma^{(i-1)}, \boldsymbol{\phi}_X^{(i-1)}, \boldsymbol{u}^*)$

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

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

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

Accept σ^* with probability

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

Run Algorithm 1 with $\boldsymbol{u}$ to update $\hat{P}(\boldsymbol{y}_{1:M} \mid \boldsymbol{\eta}_{1:M}^{(i)}, \sigma^{(i)}, \boldsymbol{\phi}_X^{(i-1)}, \boldsymbol{u})$

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

Run Algorithm 1 to obtain likelihood estimate $\hat{P}(\boldsymbol{y}_{1:M} \mid \boldsymbol{\eta}_{1:M}^{(i)}, \sigma^{(i)}, \boldsymbol{\phi}_X^*)$

Accept $\boldsymbol{\phi}_X^*$ with probability

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

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

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

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

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

end

Algorithm 9: Mixed particle method (MPM) algorithm. Random effects $\boldsymbol{\eta}_{1:M}$ and parameters ϕ_X are updated using PMMH. The latent states $\boldsymbol{X}_{1:M}$ are updated using PG and σ and ϕ_η are updated directly.

survived longer than 11 days, being euthanized on day 32 of the study. Figure 2 plots this data.

5.2 Model

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

$$dV_{m,t} = \left(\beta_m + \frac{\gamma^2}{2}\right) V_{m,t}dt + \gamma V_{m,t}^{\rho} dB_{m,t}, \quad V_{m0} = v_{m0}, \tag{10}$$

where $V_{m,t}$ is the volume of subject m at time τ_t. The random effects for this model are the parameters β_m and V_{m0} where

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

The observations are modelled by

$$Y_{m,t} = \log(V_{m,t}) + \epsilon_{m,t}, \quad \epsilon_{m,t} \sim \mathcal{N}(0, \sigma^2). \tag{11}$$

Since the data is observed on the log scale, the transformation $X_{m,t} = \log(V_{m,t})$ can be applied to (10) and (11) using Ito's lemma. Then, the full model is given by

$$\begin{cases} Y_{m,t} = X_{m,t} + \epsilon_{m,t}, \quad \epsilon_{m,t} \sim \mathcal{N}(0, \sigma^2) \\ dX_{m,t} = \left(\beta_m + \frac{\gamma^2}{2}(1 - e^{2(\rho-1)X_{m,t}})\right) dt + \gamma e^{(\rho-1)X_{m,t}} dB_{m,t} \\ X_{m0} \sim \mathcal{N}(\mu_{X0}, \sigma_{X0}^2) \\ \log(\beta_m) \sim \mathcal{N}(\mu_\beta, \sigma_\beta^2). \end{cases} \tag{12}$$

Since model (12) does not have a closed form solution for $X_{m,t}$, the likelihood is intractable. Priors were assigned to the static parameters $\theta = (\mu_{X0}, \sigma_{X0}, \mu_\beta, \sigma_\beta, \gamma, \sigma, \rho)^T$

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

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

Note that taking $\rho = 1$ gives a reduced model of the form

$$dX_{m,t} = \beta_m dt + \gamma dB_{m,t}, \tag{13}$$

which has the tractable transition density

$$X_{m,t} \sim \mathcal{N}(X_{m,t-1} + \beta_m, \gamma^2).$$

Model (13) is the original SDEMEM used by Picchini and Forman (2019). We add the parameter ρ which allow for both a more flexible variance and renders the transition

density intractable. We test this model on the dataset introduced in Section 5.1. Figure 1 plots this data. 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 (12) using $\theta = (\mu_{X0}, \sigma_{X0}, \mu_\beta, \sigma_\beta, \gamma, \sigma, \rho)^T = (3, 1, -1, 1, 1, 0.5, 1)^T$.

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 and vice versa, e.g. sim(, 1) represents sim(10, 1), sim(100, 1) and sim(1000, 1). The performance of our methods on these datasets will give an indication of their scalability with respect to the density of the time series and number of subjects. Figure 2 plots this data.

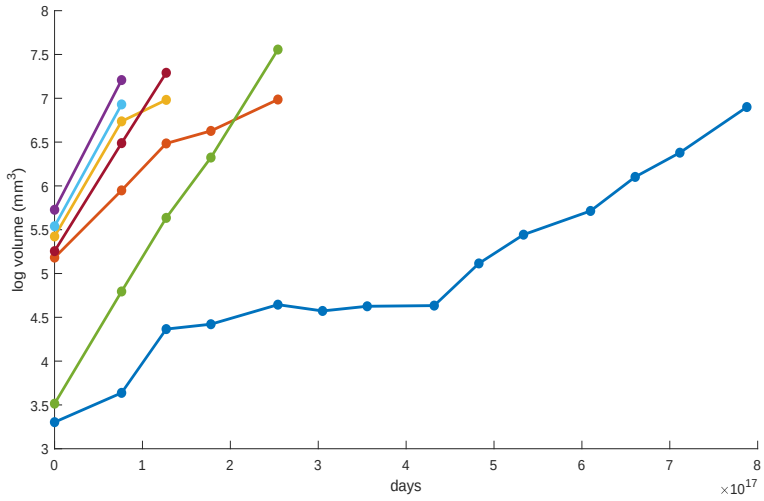

Figure 1: Plot of real tumour volume data.

6 Likelihood Estimation Results

All code was implemented in MATLAB. Vectorisation and parallelisation were applied where possible, e.g. we used vectorised code for the particle operations and parallelised over the subjects in the particle filter. For IAPM we also parallelised over the subjects when running the importance sampler. Our results were calculated using 8 cores.

We first considered the efficiency of the log-likelihood calculation. For each of the three methods, we tested all possible combinations of propagation function and importance density (IAPM). We define the naive method or combination as the IAPM algorithm with the prior as importance density and the Euler-Maruyama approximation as the propagation function in the particle filter.

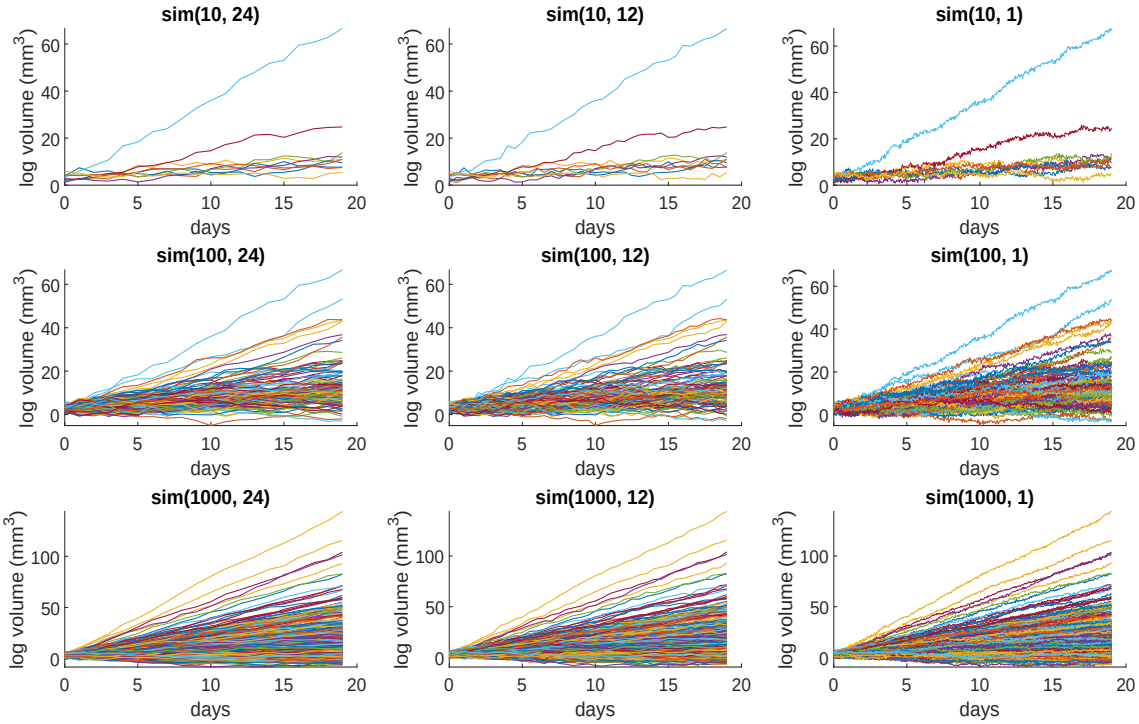

Figure 2: Plot of all simulated datasets. Sim(M, H) refers to the size of the subset, where M is the number of subjects and H is the number of hours between observations. The full dataset is denoted sim(1000, 1).

As outlined in Section 4.4, we set the tuning parameters such that $\sigma_\Delta \leq 1.05$. Measurements were 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 used $\boldsymbol{\theta} = (4, 1, 2, 1, 1.6, 0.05, 1)$, which was obtained from a few preliminary MCMC runs (low values of N, L and D was sufficient for this). For the simulated data, we used the true value $\boldsymbol{\theta} = (3, 1, -1, 1, 1, 0.5, 1)$. The random effects $\boldsymbol{\eta}_{1:M}$ were determined similarly, using preliminary runs for the real data and the true values for the synthetic data.

We define the level of discretization (D) as the number of intermediate timepoints between each observation. We found that the results are not particularly sensitive to this value, so we fixed D at 10 for all methods. Computation was stopped if the computation time for a single log-likelihood estimate exceeded 15 minutes or required more than 150gb of RAM.

In this section, we use the notation 'importance density + propagation function' to refer to a particular combination of the two, e.g. prior + RB. All combinations were tuned to roughly the same statistical efficiency (based on σ_Δ), so the most efficient method was taken as the one with the lowest computation time. Further mention of statistical efficiency refers to the value of the tuning parameters N and L.

6.1 IAPM

To tune the IAPM method, we made the simplifying assumption $N = L$. Tuning was done through trial and error. Of the three methods, IAPM was the most difficult and time-consuming to tune. The assumption $N = L$ made the tuning process a bit simpler, but it is not ideal. Depending on the implementation of the code, having a larger/smaller

N or L may have a significant impact on the computation time.

Once we started testing combinations, we found that the variance of the Laplace-ODE importance density tends to 0 for at least one of the random effects, such that the draws for that random effect were close to equal. We solved this by setting the covariance to a diagonal matrix of the prior variances scaled by 0.5. We denote this altered importance density as L-ODE.

Tables 1-4 give the log-likelihood results for all datasets. Dashed lines indicate that computation reached the time limit specified in Section 6. This limit was reached for all prior and L-ODE combinations on the sim(1000,) datasets. For the correlated versions of these, we found that the value of σ_Δ had a very high variance. All versions of IAPM exceeded the time limit on dataset sim(1000, 1).

For the synthetic data, the Laplace-MDB importance density outperformed the prior and L-ODE in terms of overall efficiency. Of the latter, the L-ODE showed the poorest performance. Results for the uncorrelated versions are only available for sim(10, 24) and sim(10, 12) and these were also the only datasets with L-ODE combinations that outperformed the prior. Based on these results, the ODE may not a good approximation of the underlying states. A large diffusion coefficient and/or measurement error could account for this.

The most efficient propagation function depended on the size of the dataset. In terms of statistical efficiency, the MDB and RB have nearly identical results across all datasets, and generally outperforms the EM. The RB takes slightly longer to run than the MDB however, and both are slower than the EM. While this had little effect on the smallest datasets, the time difference was significant on the larger ones. The EM approximation gave the best results on the sim(, 1) datasets.

Correlating the log-likelihoods generally increased the statistical efficiency. On the larger datasets, this increase was significant, as was the corresponding reduction in computation time. Interestingly, the number of particles needed for the correlated Laplace-MDB + MDB/RB remained almost constant as the number of subjects increased, e.g. 4 particles were needed for correlated Laplace-MDB + MDB on all sim(, 24) datasets. Also, for all sim(10,) datasets, the uncorrelated Laplace-MDB + EM was more statistically efficient than the correlated version. The same was also true for Laplace-MDB + MDB and RB on sim(10, 1).

For the real data, the best results were given by the Laplace-MDB in combination with the MDB or RB. A large gain in statistical efficiency was observed relative to the naive combination, i.e. prior + EM. In the uncorrelated case, the tuning parameters reduced from $L = N = 200$ to $L = N = 4$, and in the correlated from $L = N = 90$ to $L = N = 3$. A 5.5-fold decrease in time was observed from the uncorrelated naive to the best method.

		Prior			L-ODE			Lap-MDB		
PF	Cor.	L, N	σ_Δ	time (s)	L, N	σ_Δ	time (s)	L, N	σ_Δ	time (s)
EM	No	200	0.99	0.21	60	1.04	0.12	28	0.97	0.11
	Yes	90	1.00	0.11	30	1.02	0.10	19	0.99	0.09
MDB	No	180	1.02	0.36	35	0.87	0.11	4	1.02	0.05
	Yes	65	0.93	0.12	16	0.99	0.10	3	0.94	0.04
RB	No	180	1.02	0.38	35	0.85	0.11	4	1.02	0.05
	Yes	65	0.98	0.13	16	0.98	0.11	3	0.95	0.04

Table 1: Log-likelihood results for the IAPM method on the real dataset. The highlighted rows show the combinations which gave the best computation time.

			sim(10, 24)			sim(10, 12)			sim(10, 1)		
IS	PF	Cor.	L, N	σ_Δ	time (s)	L, N	σ_Δ	time (s)	L, N	σ_Δ	time (s)
Prior	EM	No	250	0.97	1.69	370	1.04	6.29	530	0.96	134.1
		Yes	115	0.99	0.54	130	1.00	1.13	335	0.99	52.58
	MDB	No	220	0.95	3.06	220	1.03	5.84	570	0.97	373.3
		Yes	95	0.91	0.86	100	0.93	1.63	250	1.03	80.02
	RB	No	220	0.96	3.07	220	1.03	6.14	570	0.98	385.9
		Yes	95	0.93	0.87	100	0.95	1.79	250	1.01	87.62
L-ODE	EM	No	220	1.04	1.31	950	1.02	35.9	-	-	-
		Yes	60	0.99	0.32	120	0.98	0.99	370	1.00	62.71
	MDB	No	145	1.03	1.57	800	1.02	55.75	-	-	-
		Yes	20	1.00	0.22	50	1.04	0.78	310	0.98	118.0
	RB	No	145	1.03	1.62	800	1.02	60.45	-	-	-
		Yes	20	1.00	0.21	50	1.04	0.81	310	0.95	126.7
Lap-MDB	EM	No	40	1.00	0.20	55	0.96	0.45	150	1.03	14.22
		Yes	45	1.00	0.25	75	1.02	0.57	290	0.96	38.61
	MDB	No	8	1.01	0.12	16	0.99	0.23	120	0.99	26.22
		Yes	4	0.88	0.10	10	0.98	0.21	190	1.00	52.18
	RB	No	8	1.01	0.12	16	0.97	0.25	120	0.97	29.70
		Yes	4	0.90	0.11	10	0.97	0.25	190	0.98	53.87

Table 2: Log-likelihood results for the IAPM method on the sim(10,) datasets. The highlighted rows show the combinations which gave the best computation time. Number of observations for each dataset (from left to right): 200, 390, 4,570.

6.2 CWPM

For CWPM, it was only necessary to select a propagation function and find a value for N. Again, this was done through experimentation. Tables 5-8 show results for all datasets. Dashed lines indicate that the memory limit specified in Section 6 was reached.

For the synthetic datasets, the correlated versions had the best results across all propagation functions. The number of particles needed for the uncorrelated versions grew quickly with the size of the dataset. Also, since the correlation induced is approximately $1 - 1/M$, the correlated versions showed greater improvement as the number of subjects increased.

As with IAPM, the most efficient propagation function depended on the size of the dataset. The best results were given by the MDB/RB, MDB and EM for the sim(, 24), sim(, 12) and sim(, 1) datasets respectively. For the sim(, 1) datasets, any benefit in statistical efficiency from the bridges was outweighed by the increase in computation time.

For the real data, we found that a single particle was sufficient to obtain $\sigma_\Delta \leq 1.05$ when using MDB or RB.

6.3 MPM

This method uses the same log-likelihood estimate as CWPM, so no extra tuning was required. When $N > 1$, we use the same number of particles for the conditional particle filter as for the standard. When $N = 1$, as is the case for the real data (see Section 6.2), we add an extra particle to account for the invariant path.

			sim(100, 24)			sim(100, 12)			sim(100, 1)		
IS	PF	Cor.	L, N	σ_Δ	time (s)	L, N	σ_Δ	time (s)	L, N	σ_Δ	time (s)
Prior	EM	No	500	1.02	52.78	500	1.01	105.7	-	-	-
		Yes	95	1.00	3.28	110	0.96	8.3985	300	1.01	419.8
	MDB	No	300	1.03	49.71	390	1.01	154.0	-	-	-
		Yes	45	0.97	2.88	45	1.00	6.16	200	1.03	556.8
	RB	No	320	0.97	60.60	390	1.00	174.0	-	-	-
		Yes	45	0.96	3.07	45	0.95	6.05	200	0.98	584.8
L-ODE	EM	No	-	-	-	-	-	-	-	-	-
		Yes	155	1.00	7.51	370	1.03	76.64	-	-	-
	MDB	No	-	-	-	-	-	-	-	-	-
		Yes	100	0.97	8.13	230	1.00	57.27	-	-	-
	RB	No	-	-	-	-	-	-	-	-	-
		Yes	100	1.00	8.75	230	1.02	65.51	-	-	-
Lap-MDB	EM	No	130	1.00	5.55	140	1.03	11.70	-	-	-
		Yes	70	1.01	2.76	85	1.00	6.20	300	0.97	417.7
	MDB	No	30	0.96	2.12	50	0.96	7.48	-	-	-
		Yes	4	0.91	0.63	10	0.98	1.53	200	0.99	532.1
	RB	No	30	0.99	2.53	50	0.98	7.92	-	-	-
		Yes	4	0.94	0.68	10	0.99	1.71	200	0.95	587.6

Table 3: Log-likelihood results for the IAPM method on the sim(100,) datasets. The highlighted rows show the combinations which gave the best computation time. Number of observations for each dataset (from left to right): 2,000, 3,900, 45,700.

			sim(1000, 24)			sim(1000, 12)			sim(1000, 1)		
IS	PF	Cor.	L, N	σ_Δ	time (s)	L, N	σ_Δ	time (s)	L, N	σ_Δ	time (s)
Prior	EM	No	-	-	-	-	-	-	-	-	-
		Yes	-	-	-	-	-	-	-	-	-
	MDB	No	-	-	-	-	-	-	-	-	-
		Yes	-	-	-	-	-	-	-	-	-
	RB	No	-	-	-	-	-	-	-	-	-
		Yes	-	-	-	-	-	-	-	-	-
L-ODE	EM	No	-	-	-	-	-	-	-	-	-
		Yes	-	-	-	-	-	-	-	-	-
	MDB	No	-	-	-	-	-	-	-	-	-
		Yes	-	-	-	-	-	-	-	-	-
	RB	No	-	-	-	-	-	-	-	-	-
		Yes	-	-	-	-	-	-	-	-	-
Lap-MDB	EM	No	350	1.04	285.5	400	1.05	701.2	-	-	-
		Yes	60	1.05	22.74	80	1.03	50.76	-	-	-
	MDB	No	90	1.04	71.46	145	1.02	292.2	-	-	-
		Yes	4	0.92	6.50	10	1.01	15.65	-	-	-
	RB	No	90	1.05	77.83	145	0.98	315.3	-	-	-
		Yes	4	0.94	7.20	10	1.00	16.63	-	-	-

Table 4: Log-likelihood results for the IAPM method on the sim(1000,) datasets. The highlighted rows show the combinations which gave the best computation time. Number of observations for each dataset (from left to right): 20,000, 39,000, 457,000.

PF	Cor.	L, N	σ_Δ	time (s)
EM	No	200	1.02	0.0044
	Yes	60	0.98	0.0030
MDB	No	1	0.34	0.0023
	Yes	1	0.16	0.0023
RB	No	1	0.33	0.0024
	Yes	1	0.17	0.0024

Table 5: Log-likelihood results for the CWPM method on the real data. The highlighted rows show the combinations which gave the best time.

		sim(10, 24)			sim(10, 12)			sim(10, 1)		
PF	Cor.	N	σ_Δ	time (s)	N	σ_Δ	time (s)	N	σ_Δ	time (s)
EM	No	450	1.02	0.0578	700	1.02	0.0915	3100	1.03	1.8672
	Yes	65	1.00	0.0559	85	1.02	0.0583	300	0.95	0.2773
MDB	No	30	1.00	0.0490	110	0.96	0.0728	2100	1.02	3.1266
	Yes	3	0.98	0.0470	10	1.04	0.0554	215	1.00	0.4922
RB	No	30	1.00	0.0503	110	0.96	0.0743	2100	1.02	3.3687
	Yes	3	0.97	0.0473	10	1.02	0.0578	210	0.99	0.54

Table 6: Log-likelihood results for the CWPM method on the sim(10,) datasets. The highlighted rows show the combinations which gave the best time.

		sim(100, 24)			sim(100, 12)			sim(100, 1)		
PF	Cor.	N	σ_Δ	time (s)	N	σ_Δ	time (s)	N	σ_Δ	time (s)
EM	No	6500	1.04	1.1254	9000	0.99	2.9294	-	-	-
	Yes	120	1.03	0.1107	120	1.05	0.1536	360	1.04	2.783
MDB	No	350	0.97	0.2385	1200	1.00	1.0330	-	-	-
	Yes	3	1.05	0.1038	11	1.03	0.1492	240	0.99	3.544
RB	No	350	0.99	0.2527	1200	0.99	1.13	-	-	-
	Yes	3	1.05	0.1035	11	1.01	0.1523	240	0.99	4.019

Table 7: Log-likelihood results for the CWPM method on the sim(100,) datasets. The highlighted rows show the combinations which gave the best time.

		sim(1000, 24)			sim(1000, 12)			sim(1000, 1)		
PF	Cor.	N	σ_Δ	time (s)	N	σ_Δ	time (s)	N	σ_Δ	time (s)
EM	No	-	-	-	-	-	-	-	-	-
	Yes	90	1.03	0.4702	110	1.03	0.9162	350	1.02	24.90
MDB	No	3500	0.98	12.76	11000	1.04	78.72	-	-	-
	Yes	3	1.05	0.4504	12	0.99	0.8920	230	1.00	34.42
RB	No	3500	1.01	13.91	11000	1.02	86.91	-	-	-
	Yes	3	1.04	0.4906	12	0.96	0.9433	230	0.99	38.19

Table 8: Log-likelihood results for the CWPM method on the sim(1000,) datasets. The highlighted rows show the combination which gave the best time.

7 MCMC Results

We used the time per log-likelihood estimate from Section 6 to determine which methods to run, i.e. ≤ 2 seconds for IAPM and ≤ 0.4 seconds for CWPM and MPM. Each of these were run for 100,000 iterations starting at the same values of $\boldsymbol{\theta}$ that were used in Section 6. The best propagation function and importance density (for IAPM) from Section 6 were used. Where the MDB and RB propagation function gave similar results, MDB was the preferred choice. Due to the time constraints, the naive method was only run on the real and sim(10, 24) datasets. None of the methods were run on the sim(1000,) datasets.

Since we used random walk proposals for all methods, we needed to tune the random walk covariance. This was done in two stages. Firstly, a good estimate of the posterior covariance for $\boldsymbol{\theta}$ was obtained. Then we calculated estimates of the posterior variances for the random effects. This was repeated for each dataset.

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 ($\frac{\text{multiESS}}{\text{time}}$). These results are shown in Table 9. A breakdown of the multiESS for each update block can be found in Table 10. Tables 11-13 show the acceptance rates (AR) for the three methods on all datasets and Figure 3 and Figures 4-5 show the marginal posteriors of $\boldsymbol{\theta}$ for the real and synthetic data respectively. As would be expected, the marginal posteriors become more precise as the size of the dataset grows (via more subjects and/or more densely observed time series).

In general, the largest score is given by CWPM. This is partly due to the $\boldsymbol{X}_0$ hyperparameters. From Table 10 it is clear that the multiESS for $(\mu_{X_0}, \sigma_{X_0})$ is always larger than the multiESS for any of the other parameter update blocks. Based on this, a more efficient algorithm for this particular example might be to use IAPM for $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$ and CWPM for $\boldsymbol{X}_0$.

On the sim(10,) datasets, the multiESS increased with the density of the time series - the highest multiESS was achieved on the sim(10, 1) dataset. On the sim(100,) datasets however, the multiESS is smaller for denser time series. It also tends to decrease as the number of subjects increase. This follows the acceptance rates for the CWPM and MPM methods. For both CWPM and MPM, the acceptance rate is highest for sim(10, 1) and lowest for sim(100, 12), which might suggest that the chain is not exploring the parameter space effectively for sim(100, 12).

8 Discussion

We have introduced three methods for simulation consistent parameter inference of state-space SDEMEMs. We have also outlined some strategies for improving the efficiency of the log-likelihood estimation for these methods, namely through choice of importance density and propagation function. The efficiency of the calculation can also be increased by correlating successive log-likelihood estimates.

The recent paper by Wiqvist et al. (July 23, 2019) 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

Data	Method	MultiESS	time (min)	MultiESS/time
Real	Naive	805	413	1.95
	IAPM	970	89	10.96
	CWPM	1505	13	120.78
	MPM	1535	34	45.57
sim(10, 24)	Naive	2286	2917	0.78
	IAPM	1257	155	8.10
	CWPM	2623.3	285	9.19
	MPM	2772	329	8.41
sim(10, 12)	Naive	-	-	-
	IAPM	1556	456	3.41
	CWPM	2764	309	8.94
	MPM	3491	415	8.41
sim(10, 1)	Naive	-	-	-
	IAPM	-	-	-
	CWPM	2960	2678	1.11
	MPM	3937	4471	0.88
sim(100, 24)	Naive	-	-	-
	IAPM	1337	1031	1.30
	CWPM	2572	640	4.02
	MPM	2422	754	3.21
sim(100, 12)	Naive	-	-	-
	IAPM	1277	2785	0.46
	CWPM	2104	974	2.16
	MPM	2087	1186	1.76

Table 9: 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.

Data	Method	$\boldsymbol{\theta}$	(γ, σ, ρ)	$(\mu_{X_0}, \sigma_{X_0})$	$(\mu_\beta, \sigma_\beta)$
Real	CWPM	1504.5	596.5	7154.4	870.4
	MPM	1535.2	673.4	6896.9	792.9
sim(10, 24)	CWPM	2623.3	1222.8	6297.6	2751.8
	MPM	2771.6	1354.3	6432.5	2891.1
sim(10, 12)	CWPM	2763.8	1488.8	7140.4	2388.2
	MPM	3490.6	1980.5	7303.1	3330.1
sim(10, 1)	CWPM	2960.4	2089.5	7369.4	1927.4
	MPM	3937.0	4107.3	7656.1	1749.1
sim(100, 24)	CWPM	2571.5	1085.7	6168.6	2819.9
	MPM	2421.7	900.3	6404.5	2913.2
sim(100, 12)	CWPM	2104.1	670.8	7275.3	2300.9
	MPM	2086.6	681.5	7287.7	2333.0

Table 10: MultiESS breakdown for each parameter blocks. The $\boldsymbol{\theta}$ column shows the multiESS for all parameters.

Data	$AR(\boldsymbol{\theta})$
Real	0.07
sim(10, 24)	0.11
sim(10, 12)	0.11
sim(10, 1)	-
sim(100, 24)	0.12
sim(100, 12)	0.11

Table 11: Acceptance rates for the IAPM method based on a chain of length 100,000.

Data	$AR(\gamma, \sigma, \rho)$	$AR(\mu_{X_0}, \sigma_{X_0})$	$AR(\mu_\beta, \sigma_\beta)$
Real	0.09	0.51	0.32
sim(10, 24)	0.12	0.47	0.43
sim(10, 12)	0.13	0.49	0.40
sim(10, 1)	0.16	0.51	0.36
sim(100, 24)	0.14	0.50	0.39
sim(100, 12)	0.06	0.50	0.39

Table 12: Acceptance rates for the CWPM method based on a chain of length 100,000.

Data	$AR(\gamma, \rho)$	$AR(\mu_{X_0}, \sigma_{X_0})$	$AR(\mu_\beta, \sigma_\beta)$	$AR(\sigma)$
Real	0.21	0.32	0.51	0.17
sim(10, 24)	0.16	0.43	0.48	0.42
sim(10, 12)	0.16	0.40	0.49	0.46
sim(10, 1)	0.27	0.51	0.36	0.63
sim(100, 24)	0.19	0.50	0.39	0.26
sim(100, 12)	0.08	0.39	0.50	0.40

Table 13: Acceptance rates for the MPM method based on a chain of length 100,000.

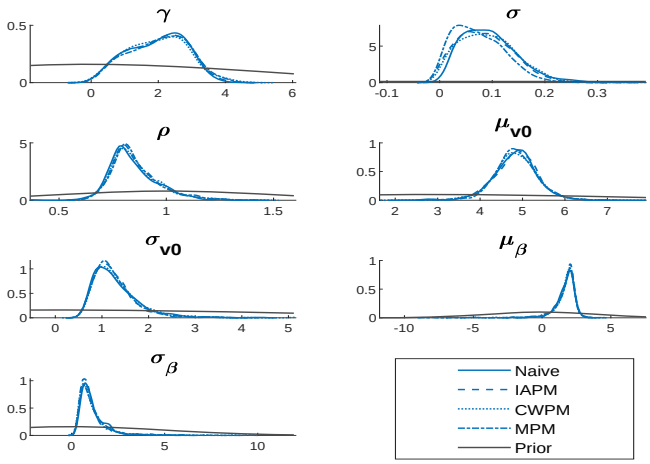

Figure 3: Univariate posterior density plots of the parameters for all methods for the real data.

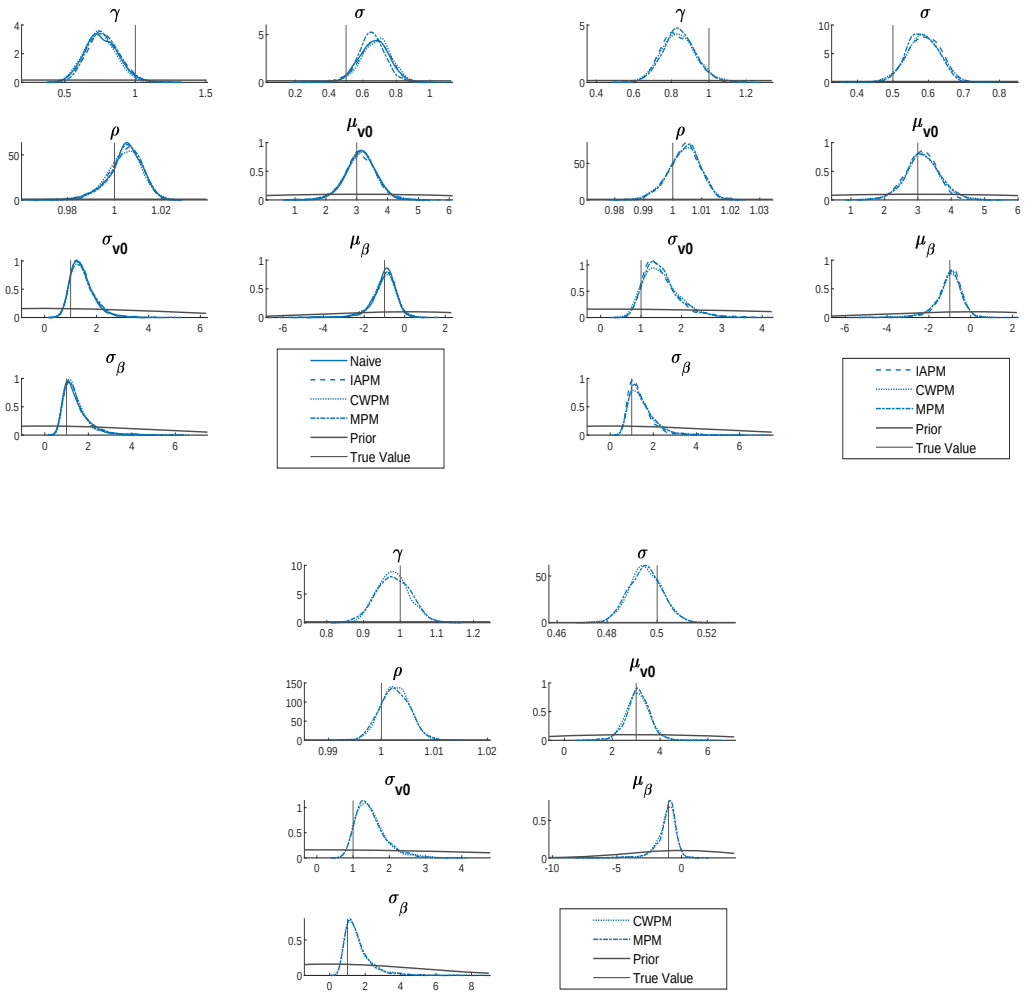

Figure 4: Univariate posterior density plots of the parameters for sim(10, 24), sim(10, 12) and sim(10, 1).

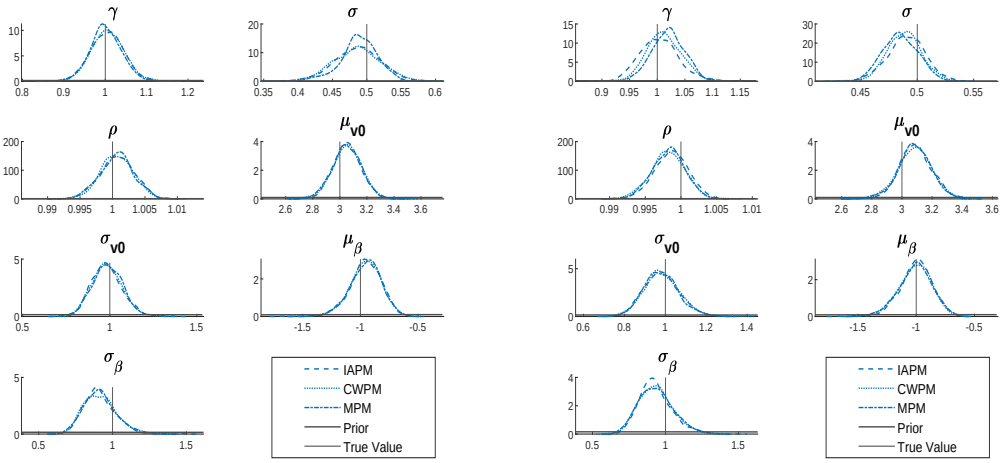

Figure 5: Univariate posterior density plots of the parameters for sim(100, 24) and sim(100, 12).

numbers u are updated whenever the likelihood is estimated. In blocked Gibbs, 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. Our approach differs in that we use the block pseudo-marginal method of Tran et al. (2016), which is natural to apply in the context of mixed effects models. To 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). With the inclusion of 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 could also be included in our methods.

With IAPM, CWPM and MPM, we were able to greatly improve upon the efficiency of the naive method, particularly in computational efficiency. For the majority of the simulated datasets, the naive approach was not computationally feasible at all. The statistical efficiency of a given method depends on the correlation between the model parameters, random effects and/or latent states. These methods are 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). For our particular example, CWPM gave the best results. 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.

A significant drawback of these methods is the amount of tuning required. For all methods (including the naive), there are at least two tuning parameters for the log-likelihood estimation. We are also missing a standard way to select the importance density and propagation function, as well as guidelines to indicate whether a correlated pseudo-marginal approach should be used. The last depends on the values of L and N, which in turn depends on the efficiency of the method and the dimension of the data. All methods require tuning the MCMC proposal densities. In order to reduce the tuning burden, we plan to

embed these methods into a sequential Monte Carlo sampler (Del Moral et al., 2006) in future research.

It may be possible to choose the importance density based on the propagation function, i.e. EM + Laplace-ODE (or L-ODE) and MDB/RB + Laplace-MDB. Recall also, the Laplace-ODE approximates the underlying states using the ODE specified by the drift of the SDEMEM. The feasibility of this importance density then relies on how quickly the solution of the ODE can be computed. Exploration of the model could potentially indicate a sensible choice of propagation function and level of discretization D. Investigating how well this approach works in practice and developing an adaptive method to tune L and N is an area of future research.

9 Acknowledgments

We would like to thank Umberto Picchini and the research team at the Centre for Nanomedicine and Theranostics (DTU Nanotech, Denmark) for providing the real data. 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's Discovery Early Career Researcher Award funding scheme (DE160100741). We gratefully acknowledge the computational resources provided by QUT's High Performance Computing and Research Support Group (HPC).

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.

Andrieu, C. and Roberts, G. O. (2009). The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics*, 37(2):697–725.

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

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*.

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.

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.

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.

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.

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.

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.

Doucet, A., Godsill, S., and Andrieu, C. (2000). On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing*, 10(3):197–208.

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.

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.

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.

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.

Gerber, M., Chopin, N., and Whiteley, N. (2019). Negative association, ordering and convergence of resampling methods. *The Annals of Statistics*, 47(4):2236–2260.

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.

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.

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

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*.

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.

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.

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)*.

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*, pages 3845–3848. IEEE.

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

Picchini, U., De Gaetano, A., and Ditlevsen, S. (2010). Stochastic differential mixed-effects models. *Scandinavian Journal of Statistics*, 37(1):67–90.

Picchini, U. and Ditlevsen, S. (2011). Practical estimation of high dimensional stochastic differential mixed-effects models. *Computational Statistics & Data Analysis*, 55(3):1426–1444.

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)*.

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.

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.

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.

Stramer, O. and Bognar, M. (2011). Bayesian inference for irreducible diffusion processes using the pseudo-marginal approach. *Bayesian Analysis*, 6(2):231–258.

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

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

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.

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.

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

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*.

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