STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **x**, Month 202x, pp 0–23.
Published online in International Academic Press (www.IAPress.org)

A Comparative Study of Monte Carlo Methods and Numerical Schemes for SDE Mixed-Effects Models

Gabriel Barragán-Ramírez [1,*], Saba Infante [1,2], Inti Becerra [1], Aracelis Hernández [1,2]

[1] *School of Mathematical and Computational Sciences, Yachay Tech University, Ecuador*
[2] *Facultad de Ciencias y Tecnología, Universidad de Carabobo, Venezuela*

Abstract Stochastic differential equation mixed-effects models (SDEMEMs) provide a flexible framework for studying time-evolving processes governed by stochastic dynamics. Analytical inference is often intractable due to incomplete observations, inter-individual variability, repeated measurements, and measurement error. This work focuses on Bayesian inference for latent states and model parameters in SDEMEMs using numerical discretization and Monte Carlo methodologies. We implement three discretization schemes—Euler–Maruyama (EM), the modified diffusion bridge (MDB), and the residual diffusion bridge (RDB)—combined with Markov Chain Monte Carlo (MCMC), pseudo-marginal, and Sequential Monte Carlo (SMC) approaches. In particular, we investigate Individual Augmentation (IA), the Metropolis-adjusted Langevin algorithm (MALA), and Hamiltonian Monte Carlo (HMC). The proposed methods are evaluated using both simulated and real datasets, including a bivariate Ornstein–Uhlenbeck (OU) process and orange tree growth data. The results indicate that correlated pseudo-marginal methods improve sampling efficiency by increasing the effective sample size (ESS). Gradient-based methods, particularly HMC, generally achieve higher acceptance rates and larger ESS values than IA-based approaches, although at a higher computational cost due to repeated gradient evaluations. Among the discretization schemes, MDB and RDB improve proposal conditioning and acceptance rates relative to EM, whereas EM and MDB provide lower computational cost than RDB. These results highlight practical trade-offs between computational efficiency, sampling performance, and implementation complexity in Bayesian inference for SDEMEMs.

Keywords Stochastic Differential Equation Mixed-Effects Models, Markov Chain Monte Carlo, Particle Filter, Numerical Methods

AMS 2010 subject classifications 62L10, 60H10, 62L20.

DOI: 10.19139/soic-2310-5070-2740

1. Introduction

Many physical and biological processes are modeled using diffusion processes governed by stochastic differential equations (SDEs). SDEs provide a flexible framework for describing dynamic systems with inherent uncertainty, making them well-suited for modeling random phenomena. Proper modeling requires carefully identifying relevant system components and accurately estimating their latent states and parameters.

In many applied fields, repeated measurements are collected from populations of objects, animals, or individuals. There is growing interest in analyzing such stochastic systems, with applications in biomedical experiments [44], tumor growth studies [43], pharmacokinetics/pharmacodynamics [17], population dynamics [30], growth curve analysis [16], intracellular processes [55], insect population modeling [56], Ornstein-Uhlenbeck processes [21],

*Correspondence to: Saba Infante (Email: sinfante@yachaytech.edu.ec). School of Mathematical and Computational Sciences, Yachay Tech University, Ecuador.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

and finance [53], among others.

The theory of mixed-effects models is well established in frequentist statistics, with a vast literature on their applications [32, 7, 34, 10, 46]. More recently, there has been growing interest in extending these models to SDEMEMs [39, 44, 13, 43, 41, 52, 51, 6, 21, 57].

An SDEMEM describes a temporal process whose dynamics follow an SDE, meaning they are influenced by stochastic noise. The model includes fixed effects (shared across individuals) and random effects (individual-specific), which may enter either the latent process dynamics or the observation model. SDEMEMs extend traditional regression models for longitudinal data by incorporating random effects to account for both within- and between-individual variability [49]. These models are especially valuable for analyzing repeated measurements taken at discrete time points in the presence of observational error across multiple experimental units.

Given a set of stochastic processes corresponding to K individuals, an SDEMEM consists of K SDEs whose drift and diffusion terms may depend linearly or non-linearly on the latent states. The parameters include fixed effects, assumed to be common across all individuals, and random effects that vary across individuals [12, 44]. A major challenge in SDEMEMs is the estimation of parameters and latent states due to the lack of closed-form solutions.

In certain cases, exact simulation-based methods such as the algorithm of [4] can be employed, avoiding discretization error through retrospective rejection sampling under specific structural assumptions on the drift and diffusion. Alternatively, parameter inference can be achieved using variational methods, such as the black-box variational inference approach developed by [48], which uses a standard mean-field variational approximation of the parameter posterior, and introduce a recurrent neural network to approximate the posterior for the diffusion paths conditional on the parameters. Others approaches are numerical approximation techniques [38], MCMC methods [35, 29, 37], and SMC methods [33, 18].
Recent developments in Bayesian inference for stochastic differential equation mixed-effects models (SDEMEMs) include simulation-based inference methods for nonlinear stochastic mixed-effects systems, manifold MCMC approaches for diffusion models, filtering-based Bayesian estimation strategies, efficient diffusion bridge constructions, and gradient-based algorithms such as MH, MALA, HMC, and Lip-MALA for large-scale inverse problems and stochastic dynamical systems [28, 27, 9, 8, 40].
This work contributes to the Bayesian inference literature for stochastic differential equation mixed-effects models (SDEMEMs) through a unified comparative study of MCMC, particle MCMC, and Sequential Monte Carlo (SMC) methodologies for latent diffusion inference. In particular, we compare augmentation-based, gradient-based, and pseudo-marginal sampling strategies within a common inferential framework.
First, we investigate a one-dimensional SDEMEM for orange tree growth using real longitudinal data, where the latent diffusion trajectories are approximated using the Euler–Maruyama (EM), modified diffusion bridge (MDB), and residual diffusion bridge (RDB) schemes. Second, we evaluate Individual Augmentation (IA), the Metropolis-adjusted Langevin algorithm (MALA), and Hamiltonian Monte Carlo (HMC) for Bayesian parameter estimation in nonlinear latent diffusion models. Third, we analyze correlated and uncorrelated pseudo-marginal approaches, including MH-Gibbs, MALA-Gibbs, and HMC-Gibbs algorithms, using simulated data from a bivariate Ornstein–Uhlenbeck process.
The proposed comparison highlights the trade-offs between computational cost, acceptance probability, effective sample size (ESS), numerical approximation accuracy, and implementation complexity across the different inference methodologies.
The remainder of this paper is organized as follows. Section 2 introduces the formulation of stochastic differential equation mixed-effects models (SDEMEMs). Section 3 presents the numerical discretization schemes used to approximate the latent diffusion processes. Section 4 describes the Bayesian inference methodologies and Monte Carlo algorithms considered in this work. Section 5 reports the numerical experiments and applications to both

real and simulated datasets. Finally, Section 6 summarizes the main findings and discusses directions for future research.

2. Formulation of SDEMEM

Given an Itô process $\{\mathbf{X}_t\}_{t\geq 0}$, the general form of a one-dimensional continuous stochastic differential equation (SDE) is

$$d\mathbf{X}_t = \mu\left(\mathbf{X}_t, \phi_X, t\right) dt + \sqrt{\nu\left(\mathbf{X}_t, \phi_X, t\right)} d\mathbf{W}_t, \quad X_0 = \mathbf{X}_0(\phi_X) \tag{1}$$

where $\mu(\cdot)$ is the drift function, $\sqrt{\nu(\cdot)}$ is the diffusion coefficient, ϕ_X are fixed effect model parameters, and $\{\mathbf{W}_t\}_{t\geq 0}$ is a standard Brownian motion process. This formulation can be extended to define a stochastic differential equation mixed-effects model (SDEMEM) by incorporating individual-specific random effects $\mathbf{b}^{(m)}$, where some parameters can vary between the $m = 1, \ldots, K$ individuals. Consider the case where we have K experimental units randomly chosen from a theoretical population. Our goal is to perform inference based on simultaneously fitting all data from the K units. Consider an experiment that involves observing a process that evolves stochastically, such that each m-unit interval corresponds to a continuous-time Itô process $\{\mathbf{X}_t\}_{t\geq 0}$ governed by the stochastic differential equation

$$\begin{aligned} d\mathbf{X}_t^{(m)} &= \mu\left(\mathbf{X}_t^{(m)}, \phi_X, \mathbf{b}^{(m)}, t\right) dt + \sqrt{\nu\left(\mathbf{X}_t^{(m)}, \phi_X, \mathbf{b}^{(m)}, t\right)} d\mathbf{W}_t^{(m)} \\ \mathbf{X}_0^{(m)} &= \mathbf{x}_0^{(m)}\left(\phi_X, \mathbf{b}^{(m)}\right), \quad m = 1, \ldots, K \end{aligned} \tag{2}$$

where $\mathbf{b}^{(m)} = (b_1^{(m)}, \ldots, b_q^{(m)})' \in \mathbb{R}^q$, with distribution $\mathbf{b}^{(m)} \sim p(b^{(m)} \mid \psi)$, where $\psi \in \mathbb{R}^r$ denotes the hyperparameter vector governing the random-effects distribution. The drift and diffusion functions are assumed to satisfy standard regularity conditions ensuring existence and uniqueness of a weak solution [38]. The model in (2) assumes that all individuals follow the same underlying functional form, while inter-individual variability is introduced through both the random effects $\mathbf{b}^{(m)}$ and the independent Brownian motion paths $\{\mathbf{W}_t^{(m)}\}_{t\geq 0}$. In general, the transition density associated with (2) is unavailable in closed form, requiring numerical approximations or Monte Carlo methods for statistical inference. The latent diffusion process $\{\mathbf{X}_t^{(m)}\}_{t\geq 0}$is not directly observed. Instead, inference is performed using discretized latent trajectories $\mathbf{x}_t^{(m)} \in \mathbf{X}_t^{(m)}\}_{t\geq 0}$. The observations $\{\mathbf{Y}_t^{(m)}\}_{t\geq 0}$ are assumed conditionally independent (given the latent process) and we link them to the latent process via

$$\mathbf{Y}_{t_i}^{(m)} = h\left(\mathbf{X}_{t_i}^{(m)}\right) + \epsilon_{t_i}^{(m)}, \quad \epsilon_{t_i}^{(m)}, \quad \overset{\text{i.i.d}}{\sim} \mathcal{N}(0, \xi^2), \qquad m = 1, \ldots, K \tag{3}$$

where $h(\cdot)$ is a possibly nonlinear observation function. A common special case is the linear observation model $h(\mathbf{X}_t^{(m)}) = H'\mathbf{X}_t^{(m)}$, where H is a constant matrix. The coupled system defined by (2) and (3) constitutes a state-space representation of a SDEMEM:

$$\begin{aligned} d\mathbf{X}_t^{(m)} &= \mu\left(\mathbf{X}_t^{(m)}, \phi_X, \mathbf{b}^{(m)}, t\right) dt + \sqrt{\nu\left(\mathbf{X}_t^{(m)}, \phi_X, \mathbf{b}^{(m)}, t\right)} d\mathbf{W}_t^{(m)} \\ \mathbf{Y}_t^{(m)} &= h\left(\mathbf{X}_t^{(m)}\right) + \epsilon_t^{(m)}, \quad \epsilon_t^{(m)} \sim \mathcal{N}(0, \xi^2), \qquad m = 1, \ldots, K, \end{aligned}$$

Equation (2) captures both intrinsic stochasticity through the diffusion variance $\nu(\cdot)$ and inter-individual variability through the random effects $\mathbf{b}^{(m)}$, whereas (3) accounts for measurement error through the observation noise parameter ξ.

The main objective is to infer the latent diffusion trajectories $\{\mathbf{X}_t^{(m)}\}_{t\geq 0}$ defined by (2) together with the unknown model parameters $\boldsymbol{\theta} = (\phi_X, \mathbf{b}^{(m)}, \psi, \xi)$. The joint posterior distribution of the latent trajectories $\mathbf{x} = (x^{(1)}, \ldots, x^{(K)})$, the random effects $\mathbf{b}^{(m)} = (b_1, \ldots, b_q)$, and the model parameters $\boldsymbol{\theta}$, given the observed data

$\mathbf{y} = (y^{(1)}, \ldots, y^{(K)})$, can be expressed as

$$p(\boldsymbol{\theta}, \mathbf{x} \mid \mathbf{y}) \propto p(\phi_X)p(\psi)p(\xi) \prod_{m=1}^{K} p(y^{(m)} \mid \mathbf{x}^{(m)}, \xi)\, p(x^{(m)} \mid \phi_X, \mathbf{b}^{(m)})\, p(\mathbf{b}^{(m)} \mid \psi). \tag{4}$$

In general, the likelihood associated with SDEMEMs is analytically intractable because the transition densities of the latent diffusion process are unavailable in closed form. Consequently, both numerical approximation schemes and Monte Carlo methods are required for statistical inference. In this work, numerical discretization methods are employed to approximate the latent diffusion trajectories, while Bayesian inference techniques are used to characterize the posterior distribution of the latent states and model parameters.

3. Numerical methods for SDEs

In this section, we apply a discretization of (2) between observation times using a data augmentation approach. Let $[t_{j-1}, t_j]$ denote the interval between two consecutive observation times. We introduce a partition:

$$t_{j-1} = \tau_{j,0} < \tau_{j,1} < \ldots < \tau_{j,P-1} < \tau_{j,P} = t_j, \qquad \Delta\tau = \frac{t_j - t_{j-1}}{P} \tag{5}$$

where P is the discretization level. The complete trajectory from t_0 to t_n for unit m, (for simplicity, all individuals are assumed to be observed at the common observation times $t_1, \ldots, t_n$). Let $t_0 < t_1 < \cdots < t_n$ denote the common observation times for all individuals. For each interval $[t_{j-1}, t_j]$, we introduce a discretization grid $\tau_0 = t_{j-1} < \tau_1 < \cdots < \tau_P = t_j$, where the intermediate points are used to augment the latent diffusion trajectories. The complete discretized latent trajectory for the m-th individual is represented as

$$\mathbf{X}^{(m)}_{[t_0,t_n]} = \left((\mathbf{X}^{(m)}_{[t_0,t_1]})', (\mathbf{X}^{(m)}_{(t_1,t_2]})', \ldots, (\mathbf{X}^{(m)}_{(t_{n-1},t_n]})' \right),$$

where

$$\mathbf{X}^{(m)}_{[t_{j-1},t_j]} = \left(\mathbf{X}^{(m)}_{j,\tau_0}, \mathbf{X}^{(m)}_{j,\tau_1}, \ldots, \mathbf{X}^{(m)}_{j,\tau_P} \right)'$$

contains the latent states associated with the discretization points over the interval $[t_{j-1}, t_j]$, including $P-1$ imputed intermediate states.

3.1. Euler-Maruyama

The Euler–Maruyama (EM) discretization is a standard numerical scheme used to simulate approximate trajectories of stochastic differential equations (SDEs). The method assumes that the drift and diffusion coefficients remain locally constant over sufficiently small time intervals. For notational simplicity, we define

$$\mu_{\tau_i} = \mu \left(\mathbf{X}^{(m)}_{j,\tau_{i-1}}, \phi_X, \mathbf{b}^{(m)} \right), \qquad \nu_{\tau_i} = \nu \left(\mathbf{X}^{(m)}_{j,\tau_{i-1}}, \phi_X, \mathbf{b}^{(m)} \right)$$

The discretized SDE is then given by

$$x^{(m)}_{j,\tau_i} = x^{(m)}_{j,\tau_{i-1}} + \mu_{\tau_i}\Delta\tau + \sqrt{\nu_{\tau_i}}\,\Delta\mathbf{W}^{(m)}_{j,\tau_{i-1}}, \qquad x^{(m)}_{j,\tau_0} = x^{(m)}_{j,\tau_0}$$

Consequently, the EM transition proposal is

$$x^{(m)}_{j,\tau_i} \mid x^{(m)}_{j,\tau_{i-1}} \sim N\left(x^{(m)}_{j,\tau_{i-1}} + \mu_{\tau_i}\Delta\tau,\, \nu_{\tau_i}\Delta\tau \right)$$

which approximates the transition density $p\left(x^{(m)}_{j,\tau_i} \mid x^{(m)}_{j,\tau_{i-1}}, \phi_X, b^{(m)}\right)$. When the drift and diffusion coefficients are constant, the Euler–Maruyama approximation coincides with the exact Gaussian transition density.

3.2. *Diffusion Bridge*

Diffusion bridge methods generate trajectories that are more efficiently guided toward future observations. A diffusion bridge is a trajectory of a diffusion process that satisfies an SDE, conditioned on its value at a fixed future time. These methods are particularly useful for partially and noisily observed systems with nonlinear dynamics [56].

The modified diffusion bridge (MDB) constructs proposal trajectories by conditioning the latent diffusion process on the future observation at time t_j. Following [20] and [25], the proposal is obtained through a locally Gaussian approximation of the conditional transition density, where the drift term is adjusted to guide the latent trajectory toward the observation. Consequently, the proposal combines the original diffusion dynamics with a correction term that depends on the discrepancy between the predicted latent state and the future observation. The modified drift term is given by

$$\tilde{\mu}_{\tau_i} = \mu_{\tau_i} + \frac{\nu_{\tau_i}\left[y_{t_j}^{(m)} - \left(x_{\tau_i}^{(m)} + \mu_{\tau_i}\Delta_i\right)\right]}{\nu_{\tau_i}\Delta_i + \xi^2},$$

while the modified diffusion coefficient is $\tilde{\nu}_{\tau_i} = \nu_{\tau_i} - \frac{\nu_{\tau_i}^2}{\nu_{\tau_i}\Delta_i + \xi^2}$. Under a locally Gaussian approximation, the proposal transition density is

$$x_{\tau_{i+1}}^{(m)} \mid x_{\tau_i}^{(m)}, y_{t_j}^{(m)} \sim \mathcal{N}\left(x_{\tau_i}^{(m)} + \tilde{\mu}_{\tau_i}\Delta\tau,\ \tilde{\nu}_{\tau_i}\Delta\tau\right),$$

where $\Delta\tau = \tau_{i+1} - \tau_i$, and $\Delta_i = t_j - \tau_i$.

3.3. *Residual Diffusion Bridge*

The residual diffusion bridge (RDB), proposed by [56], decomposes the latent process $\mathbf{X}_t^{(m)}$ into a deterministic component $D_t^{(m)}$ and a stochastic residual $\rho_t^{(m)}$, such that $\mathbf{X}_t^{(m)} = D_t^{(m)} + \rho_t^{(m)}$ for $t \in [t_{j-1}, t_j]$. The MDB is then applied to the residual process. The deterministic path $D_t^{(m)}$ solves the ordinary differential equation (ODE) $dD_t^{(m)} = \mu(D_t^{(m)}, \cdot)dt$ with initial condition $D_0^{(m)} = \mathbf{X}_{t_{j-1}}^{(m)}$. The residual process is then defined by

$$\rho_t^{(m)} = \mathbf{X}_t^{(m)} - D_t^{(m)},$$

and evolves according to:

$$d\rho_t^{(m)} = \left\{\mu\left(\mathbf{X}_t^{(m)}, \cdot\right) - \mu\left(D_t^{(m)}, \cdot\right)\right\} dt + \sqrt{\nu\left(X_t^{(m)}, \cdot\right)}\, dW_t^{(m)}, \qquad \rho_{t_{j-1}}^{(m)} = 0.$$

The corresponding adjusted drift term is given by:

$$\mu(\mathbf{X}_{j,\tau_i}^{(m)}, \phi_X, \mathbf{b}^{(m)}) = \mu_{\tau_i} + \frac{\nu_{\tau_i}\left(\mathbf{Y}_{t_j}^{(m)} - \left(D_{t_j}^{(m)} + \rho_{\tau_i}^{(m)} + (\mu_{\tau_i} - \delta_i^D)\Delta_i\right)\right)}{\nu_{\tau_i}\Delta_i + \xi^2}$$

The adjusted diffusion variance is

$$\nu(\mathbf{X}_{j,\tau_i}^{(m)}, \phi_X, \mathbf{b}^{(m)}) = \nu_{\tau_i} - \frac{\nu_{\tau_i}^2\Delta\tau}{\nu_{\tau_i}\Delta_i + \xi^2}$$

where $\delta_i^D = \frac{D_{\tau_{i+1}}^{(m)} - D_{\tau_i}^{(m)}}{\Delta\tau}$, $\Delta\tau = \tau_{i+1} - \tau_i$ and $\Delta_i = t_j - \tau_i$. Under the RDB approximation, the proposal transition density is

$$x_{\tau_{i+1}}^{(m)} \mid x_{\tau_i}^{(m)}, y_{t_j}^{(m)} \sim \mathcal{N}\left(D_{\tau_{i+1}}^{(m)} + \rho_{\tau_i}^{(m)} + \tilde{\mu}_{\tau_i}^{\rho}\Delta\tau,\ \tilde{\nu}_{\tau_i}^{\rho}\Delta\tau\right),$$

where $\rho_{\tau_i}^{(m)} = x_{\tau_i}^{(m)} - D_{\tau_i}^{(m)}$. Therefore, the RDB proposal density can be written as

$$q_{\text{RDB}}\left(\mathbf{X}_{\tau_{i+1}}^{(m)} \mid \mathbf{X}_{\tau_i}^{(m)}, \mathbf{Y}_{t_j}^{(m)}\right) = \phi\left(\mathbf{X}_{\tau_{i+1}}^{(m)};\, D_{\tau_{i+1}}^{(m)} + \rho_{\tau_i}^{(m)} + \tilde{\mu}_{\tau_i}^{\rho}\Delta\tau,\, \tilde{\nu}_{\tau_i}^{\rho}\Delta\tau\right),$$

where $\phi(\cdot;\alpha,\Sigma)$ denotes the Gaussian density with mean α and variance Σ.
The residual diffusion bridge (RDB) further improves proposal conditioning by decomposing the latent process into a deterministic component and a stochastic residual process, following [56]. The deterministic trajectory captures the dominant nonlinear dynamics through an auxiliary ordinary differential equation, while the residual bridge accounts for the remaining stochastic variability. MDB-type corrections are then applied to the residual process rather than directly to the original diffusion trajectory.

4. Bayesian inference

Denote by $\mathbf{x}^{(m)} = \left(x_0^{(m)}, x_1^{(m)}, \ldots, x_T^{(m)}\right)$ the vector of latent states corresponding to the m-th diffusion process $\mathbf{X}_t^{(m)}$ observed at discrete time points $t = 1,\ldots,T,\ m = 1,\ldots,K$ and define $\mathbf{x} = \left(x^{(1)},\ldots,x^{(K)}\right)$ as the collection of all latent trajectories. Similarly, denote the data $\mathbf{y}^{(m)} = \left(y_1^{(m)},\ldots,y_T^{(m)}\right)$, $\mathbf{y} = \left(y^{(1)},\ldots,y^{(K)}\right)$, and fixed/random effects $\mathbf{b}^{(m)} = \left(b_1^{(m)},\ldots,b_q^{(m)}\right)$. Our goal is to infer both the collection of all trajectories $\mathbf{x}^{(m)}$ and the parameters $\boldsymbol{\theta}$. For convenience, we define the full set of unknowns parameters as $\tilde{\boldsymbol{\theta}} = (\mathbf{x}^{(m)}, \boldsymbol{\theta})$.
By Bayes' theorem, the unnormalized posterior distribution of $\tilde{\boldsymbol{\theta}}$ given the data $\mathbf{y}^{(m)}$ is

$$p\left(\tilde{\boldsymbol{\theta}} \mid \mathbf{y}^{(m)}\right) \propto p\left(\mathbf{y}^{(m)} \mid \tilde{\boldsymbol{\theta}}\right) p\left(\tilde{\boldsymbol{\theta}}\right)$$

where $p\left(\mathbf{y}^{(m)} \mid \tilde{\boldsymbol{\theta}}\right)$ is the likelihood and $p\left(\tilde{\boldsymbol{\theta}}\right)$ is the prior distribution over the collection of all latent trajectories and parameters. We note that the likelihood $p\left(\mathbf{y}^{(m)} \mid \tilde{\boldsymbol{\theta}}\right)$ simplifies to $p\left(\mathbf{y}^{(m)} | \mathbf{x}^{(m)}, \xi\right)$, as the observations depend only on the states and the measurement error parameter. Assuming that ϕ_X, ψ and ξ are a prior independent, the joint prior distribution $p\left(\tilde{\boldsymbol{\theta}}\right)$ can be factorized as

$$p\left(\tilde{\boldsymbol{\theta}}\right) = p\left(\mathbf{x}^{(m)} \mid \phi_X, \mathbf{b}^{(m)}\right) p\left(\mathbf{b}^{(m)} \mid \psi\right) p\left(\phi_X\right) p\left(\psi\right) p\left(\xi\right)$$

In addition, we have that $p\left(\mathbf{b}^{(m)} \mid \psi\right) = \prod_{t=1}^{q} p\left(b_t^{(m)} \mid \psi\right)$ and

$$p\left(\mathbf{x}^{(m)} \mid \phi_X, b^{(m)}\right) = \prod_{m=1}^{K} p\left(x_0^{(m)}\right) \prod_{t=1}^{T} p\left(x_t^{(m)} \mid x_{t-1}^{(m)}, \phi_X, b^{(m)}\right) \tag{6}$$

$$p\left(\mathbf{y}^{(m)} \mid \mathbf{x}^{(m)}, \xi\right) = \prod_{m=1}^{K} \prod_{t=1}^{T} p(y_t^{(m)} \mid x_t^{(m)}, \xi) \tag{7}$$

Note that the transition distribution $p(x_t^{(m)} \mid x_{t-1}^{(m)}, \phi_X, b^{(m)})$ in (6) may be intractable for certain models. In such cases, we employ numerical approximation methods such as EM, MDB and RDB (see Section 3). In addition, the individual likelihood $p(y_t^{(m)} \mid x_t^{(m)}, \xi)$ in (7) is typically intractable in SDEMEMs due to the complexity of the latent diffusion process. To address this issue, we employ a particle filter—a method from the Sequential Monte Carlo (SMC) family—which provides an unbiased approximation of the likelihood. In terms of efficiency, there may exist strong correlation between the latent states $\mathbf{x}^{(m)}$ and the model parameters $\boldsymbol{\theta}$.

The joint posterior distribution of the latent trajectories, random effects, and model parameters is given by

$$p\left(\mathbf{x}^{(m)}, \mathbf{b}^{(m)}, \phi_X, \psi, \xi \mid \mathbf{y}^{(m)}\right) \propto \left[\prod_{m=1}^{K}\prod_{t=1}^{T} p(y_t^{(m)} \mid x_t^{(m)}, \xi)\right]\left[\prod_{m=1}^{K} p(x_0^{(m)}) \prod_{t=1}^{T} p(x_t^{(m)} \mid x_{t-1}^{(m)}, \phi_X, b^{(m)})\right] \times \left[\prod_{m=1}^{K} p(b^{(m)} \mid \psi)\right] p(\phi_X)\, p(\psi)\, p(\xi) \tag{8}$$

To perform Bayesian inference for the parameters $\boldsymbol{\theta}$ in the context of SDEMEMs, we rely on Markov Chain Monte Carlo (MCMC) methods to sample from the posterior distribution $p(\boldsymbol{\theta} \mid \mathbf{y}^{(m)})$. However, the likelihood function $p(\mathbf{y}^{(m)} \mid \boldsymbol{\theta})$ involves integration over the latent paths $\mathbf{x}^{(m)}$, which is analytically intractable due to the nonlinearity and stochasticity of the underlying diffusion processes.

4.1. Gibbs sampler

The Gibbs sampler is a Markov Chain Monte Carlo (MCMC) method used to generate samples from complex posterior distributions by iteratively sampling from lower-dimensional conditional distributions. At each iteration, the parameters are updated sequentially according to their full conditional posterior distributions while conditioning on the current values of the remaining variables. Gibbs sampling is particularly attractive in hierarchical Bayesian models and SDEMEMs when some conditional distributions admit closed-form expressions or can be efficiently approximated [24]. Instead of augmenting the MCMC state space with the latent variables $\mathbf{x}^{(m)}$, we consider the marginal posterior distribution obtained by integrating them out $p(\boldsymbol{\theta} \mid \mathbf{y}^{(m)}) = \int p(\boldsymbol{\theta}, \mathbf{x}^{(m)} \mid \mathbf{y}^{(m)})\, d\mathbf{x}^{(m)}$. We recall that $\boldsymbol{\theta} = (\phi_X, \mathbf{b}^{(m)}, \psi, \xi)$, and the posterior distribution factorizes as

$$p\left(\phi_X, \mathbf{b}^{(m)}, \psi, \xi \mid \mathbf{y}^{(m)}\right) \propto p\left(\mathbf{y}^{(m)} \mid \phi_X, \mathbf{b}^{(m)}, \xi\right)\, p\left(\mathbf{b}^{(m)} \mid \psi\right)\, p(\phi_X)\, p(\psi)\, p(\xi)$$

and $p\left(\mathbf{y}^{(m)} \mid \phi_X, \mathbf{b}^{(m)}, \xi\right) = \prod_{m=1}^{K} p\left(\mathbf{y}^{(m)} \mid \phi_X, \mathbf{b}^{(m)}, \xi\right)$. This factorization suggests a Gibbs sampler that iteratively samples each parameter from its full conditional distribution, given the current values of the others:

$$\begin{aligned}
p\left(\phi_X \mid \mathbf{b}^{(m)}, \psi, \xi, \mathbf{y}^{(m)}\right) &\propto p\left(\mathbf{y}^{(m)} \mid \phi_X, \mathbf{b}^{(m)}, \xi\right)\, p(\phi_X) \\
p\left(\psi \mid \phi_X, \mathbf{b}^{(m)}, \xi, \mathbf{y}^{(m)}\right) &\propto p(\mathbf{b}^{(m)} \mid \psi)\, p(\psi) \\
p\left(\mathbf{b}^{(m)} \mid \phi_X, \psi, \xi, \mathbf{y}^{(m)}\right) &\propto p\left(\mathbf{y}^{(m)} \mid \phi_X, \mathbf{b}^{(m)}, \xi\right)\, p(\mathbf{b}^{(m)} \mid \psi) \\
p\left(\xi \mid \phi_X, \mathbf{b}^{(m)}, \psi, \mathbf{y}^{(m)}\right) &\propto p\left(\mathbf{y}^{(m)} \mid \phi_X, \mathbf{b}^{(m)}, \xi\right)\, p(\xi)
\end{aligned}$$

Several comprehensive reviews of the Gibbs sampler and its variants can be found in the literature; see, for instance, [35, 3]. When conjugate priors are available, the full conditional distributions admit closed-form expressions and can be sampled directly. Otherwise, the corresponding parameters are updated using Metropolis-Hastings steps within the Gibbs sampling framework (MH-within-Gibbs).

4.2. Metropolis-Hastings algorithm

The Metropolis-Hastings (MH) algorithm is a Markov Chain Monte Carlo (MCMC) method for generating samples from a posterior distribution $p\left(\boldsymbol{\theta} \mid \mathbf{y}^{(m)}\right)$ when direct sampling is not feasible. It constructs an ergodic Markov chain whose stationary distribution is the target posterior [35]. At each iteration i, given the current state $\boldsymbol{\theta}^{(i)}$, the algorithm proceeds as follows:

1. Propose a new value $\boldsymbol{\theta}^{\text{cand}} \sim q(\boldsymbol{\theta}^{\text{cand}} \mid \boldsymbol{\theta}^{(i)})$, where q is a proposal distribution.
2. Compute the acceptance probability: $\alpha = \min\left\{1, \frac{p\left(\mathbf{y}^{(m)} \mid \boldsymbol{\theta}^{\text{cand}}\right) p\left(\boldsymbol{\theta}^{\text{cand}}\right) q\left(\boldsymbol{\theta}^{(i)} \mid \boldsymbol{\theta}^{\text{cand}}\right)}{p\left(y^{(m)} \mid \boldsymbol{\theta}^{(i)}\right) p\left(\boldsymbol{\theta}^{(i)}\right) q\left(\boldsymbol{\theta}^{\text{cand}} \mid \boldsymbol{\theta}^{(i)}\right)}\right\}$.

3. With probability α, accept the proposal by setting $\boldsymbol{\theta}^{(i+1)} = \boldsymbol{\theta}^{\text{cand}}$; otherwise, set $\boldsymbol{\theta}^{(i+1)} = \boldsymbol{\theta}^{(i)}$.

The resulting sequence $\{\boldsymbol{\theta}^{(i)}\}$ defines an ergodic Markov chain that converges to the target posterior distribution.

4.3. Particle filter

The particle filter (PF) is a sequential Monte Carlo method used to estimate the latent states of a state-space model when the system is observed with noise and evolves over time. In models where the latent process is non-linear or non-Gaussian—as is common in SDE's and SDEMEM's—analytical solutions are not available. The particle filter provides an approximation to the filtering distribution using a set of weighted samples, or particles, denoted by $\{x_t^{(m,n)}, w_t^{(m,n)}\}_{n=1}^N$ [11]. The algorithm proceeds in three main steps:

1. Prediction (sampling): particles are sampled from a proposal distribution.
2. Update (weighting): weights are assigned based on the likelihood of the new observation.
3. Resampling: particles are resampled according to their weights to avoid degeneracy.

A key advantage of the particle filter is its ability to produce a non-negative, unbiased estimate of the marginal likelihood $p\left(\mathbf{y}^{(m)} \mid \boldsymbol{\theta}\right)$, which is essential for pseudo-marginal MCMC methods [11, 45]. The bootstrap particle filter is a special case of this algorithm in which the proposal distribution is the state transition $p\left(x_t^{(m)} \mid x_{t-1}^{(m)}, \boldsymbol{\theta}\right)$, which is numerically approximated using methods such as Euler-Maruyama, the modified diffusion bridge (MDB), or the residual diffusion bridge (RDB). The importance weights are then given by the observation likelihood $p(\mathbf{y}_t^{(m)} \mid \mathbf{x}_t^{(m)}, \boldsymbol{\theta})$ [26]. The complete procedure for estimating the likelihood using a bootstrap particle filter is presented in Algorithm 1.

Algorithm 1 Bootstrap particle filter to estimate likelihood $p\left(\mathbf{y}^{(m)} \mid \boldsymbol{\theta}\right)$

Require: Data $\mathbf{y}^{(m)} = \{y_1^{(m)}, \ldots, y_T^{(m)}\}$, parameters $\boldsymbol{\theta}$, number of particles N, $\mathbf{u}$ are auxiliary variables used for prediction and resampling steps.
Ensure: Individual likelihood estimate $\widehat{p}(y^{(m)} \mid \boldsymbol{\theta}, \mathbf{u})$
Initialize: Sample $x_0^{(m,n)} \sim \mu(x_0^{(m,n)} \mid \boldsymbol{\theta})$ for $n = 1, \ldots, N$; set $\widehat{p} = 1$
for $t = 1$ to T **do**
    for $n = 1$ to N **do**
        Prediction: $x_t^{(m,n)} \sim p(x_t^{(m,n)} \mid x_{t-1}^{(m,n)}, \boldsymbol{\theta})$ using a numerical discretization scheme such as EM, MDB, or RDB. In this step, the auxiliary variables $\mathbf{u}$ are used to simulate increments of the Brownian motion paths in the numerical scheme.
        Weighting: $w_t^{(m,n)} = p(y_t^{(m)} \mid x_t^{(m,n)}, \boldsymbol{\theta})$ using equation 3
    end for
    Normalize weights: $\widetilde{w}_t^{(m,n)} = w_t^{(m,n)} / \sum_{i=1}^N w_t^{(m,i)}$
    Estimate likelihood: $\widehat{p} = \widehat{p} \cdot \left(\frac{1}{N} \sum_{n=1}^N w_t^{(m,n)}\right)$
    Resample $\{x_t^{(m,n)}\}_{n=1}^N$ with replacement according to $\{\widetilde{w}_t^{(m,n)}\}_{n=1}^N$. The random variables $\mathbf{u}$ are used in the resampling step
end for
Individual likelihood estimate: $\widehat{p}\left(\mathbf{y}^{(m)} \mid \boldsymbol{\theta}, \mathbf{u}\right) = \widehat{p}$
The non-negative unbiased estimate of $p\left(\mathbf{y}^{(m)} \mid \boldsymbol{\theta}\right)$ is given by

$$\widehat{p}(\mathbf{y}^{(m)} \mid \boldsymbol{\theta}, \mathbf{u}) = \prod_{m=1}^{K} \widehat{p}(y^{(m)} \mid \boldsymbol{\theta}, \mathbf{u}) \tag{9}$$

4.4. Pseudo-Marginal MCMC approach

The pseudo-marginal MCMC approach is a Bayesian inference methodology designed for models with intractable likelihood functions. Instead of evaluating the exact likelihood $p(\mathbf{y} \mid \boldsymbol{\theta})$, the method replaces it with a non-negative unbiased estimator $\widehat{p}(\mathbf{y} \mid \boldsymbol{\theta})$, typically obtained through Monte Carlo techniques such as particle filtering. The resulting Markov chain targets the exact posterior distribution despite using stochastic likelihood estimates [2]. Pseudo-marginal methods are particularly useful in SDEMEMs, where the latent diffusion processes make direct likelihood evaluation analytically infeasible.
When the likelihood $p\left(\mathbf{y}^{(m)} \mid \boldsymbol{\theta}\right)$ is intractable—as occurs in SDEMEMs,where the individual likelihood contributions

$$p\left(\mathbf{y}^{(m)} \mid \boldsymbol{\theta}\right) = p\left(\mathbf{y}^{(m)} \mid \phi_X, \mathbf{b}^{(m)}, \xi\right)$$

cannot be evaluated analytically—it can be replaced by a non-negative unbiased estimator $\widehat{p}(y \mid \boldsymbol{\theta}, \mathbf{u})$, where $\mathbf{u} \sim p(\mathbf{u})$ denotes a collection of auxiliary random variables introduced to construct the likelihood estimator. The auxiliary variables $\mathbf{u}$ represent the complete set of random draws used within the particle filter, including the Gaussian random variables employed to simulate Brownian motion increments in the discretized diffusion trajectories, together with the random variables used during the resampling steps. Within the correlated pseudo-marginal framework, correlation is introduced directly on these auxiliary variables in order to reduce the variance of successive likelihood estimates. This strategy is known as the pseudo-marginal MCMC approach [2]. At iteration i of the Metropolis–Hastings (MH) algorithm, given the current state $(\boldsymbol{\theta}^{(i)}, \mathbf{u}^{(i)})$ a candidate pair$(\boldsymbol{\theta}^{\text{cand}}, \mathbf{u}^{\text{cand}})$, is proposed according to

$$\boldsymbol{\theta}^{\text{cand}} \sim q(\boldsymbol{\theta}^{\text{cand}} \mid \boldsymbol{\theta}^{(i)}), \qquad \mathbf{u}^{\text{cand}} \sim p(\mathbf{u})$$

The proposed values are accepted with probability

$$\alpha = \min\left\{1, \frac{\widehat{p}\left(y^{(m)} \mid \boldsymbol{\theta}^{\text{cand}}, \mathbf{u}^{\text{cand}}\right) p(\boldsymbol{\theta}^{\text{cand}}) q(\boldsymbol{\theta}^{(i)} \mid \boldsymbol{\theta}^{\text{cand}})}{\widehat{p}\left(y^{(m)} \mid \boldsymbol{\theta}^{(i)}, \mathbf{u}^{(i)}\right) p\left(\boldsymbol{\theta}^{(i)}\right) q\left(\boldsymbol{\theta}^{\text{cand}} \mid \boldsymbol{\theta}^{(i)}\right)}\right\}. \tag{10}$$

4.5. Particle Marginal Metropolis-Hastings algorithm

The Particle Marginal Metropolis-Hastings (PMMH) algorithm is a Pseudo-Marginal MCMC method that replaces the intractable likelihood with a non-negative, unbiased estimate obtained from a particle filter [1]. In this context, the auxiliary random variables $\mathbf{u}$ correspond to the random numbers used to propagate and resample the particles within the filtering algorithm. A known drawback of the PMMH algorithm is the difficulty in designing efficient proposal distributions. Moreover, the Markov chain may get stuck when the likelihood is significantly overestimated at a given parameter value $\boldsymbol{\theta}$, since this leads to a low acceptance probability for new proposals unless their likelihoods are similarly overestimated. This issue is linked to the variability of the log-likelihood ratio

$$R = \log\left(\frac{\widehat{p}\left(\mathbf{y}^{(m)} \mid \boldsymbol{\theta}^{\text{cand}}, \mathbf{u}^{\text{cand}}\right)}{\widehat{p}\left(\mathbf{y}^{(m)} \mid \boldsymbol{\theta}, \mathbf{u}\right)}\right)$$

. whose variance should remain sufficiently small to ensure satisfactory mixing of the Markov chain. A common strategy to reduce this variance is to increase the number of particles N used in the particle filter [6]. Although this increases the computational cost per iteration, it generally improves the accuracy of the likelihood estimator. Empirical and theoretical studies suggest that efficient pseudo-marginal performance is often achieved when the variance of the log-likelihood ratio remains close to one [50].

4.6. Correlated Particle Marginal Metropolis-Hastings (PMMH)

The Correlated PMMH algorithm is a variant of the Particle Marginal Metropolis-Hastings method [1], designed to improve sampling efficiency by reducing the variance of the estimated likelihood ratio. It achieves this by

Algorithm 2 Particle marginal Metropolis-Hastings algorithm

Require: Data $\mathbf{y}^{(m)}$, initial values $(\boldsymbol{\theta}^{(1)}, \mathbf{u}^{(1)})$, N number of particles for particle filter algorithm, number of iterations I, burn-in B

Ensure: Posterior samples $\boldsymbol{\theta}^{(B+1)}, \ldots, \theta^{(I)}$

1: Sample $\mathbf{u}^{(1)} \sim p(u)$
2: Compute the likelihood estimate $\widehat{p}(\mathbf{Y} \mid \boldsymbol{\theta}^{(1)}, \mathbf{u}^{(1)})$ using Algorithm 1
3: **for** $i = 1$ to I **do**
4: Sample $\boldsymbol{\theta}^{\text{cand}} \sim q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(i)})$ and $\mathbf{u}^{\text{cand}} \sim p(\mathbf{u})$
5: Compute the likelihood estimate $\widehat{p}\left(\mathbf{y}^{(m)} \mid \boldsymbol{\theta}^{\text{cand}}, \mathbf{u}^{\text{cand}}\right)$ using Algorithm 1
6: Calculate the acceptance probability

$$\alpha = \min\left\{1, \frac{\widehat{p}(\mathbf{y}^{(m)} \mid \boldsymbol{\theta}^{\text{cand}}, \mathbf{u}^{\text{cand}})\, p(\boldsymbol{\theta}^{\text{cand}})\, q(\boldsymbol{\theta}^{(i)} \mid \boldsymbol{\theta}^{\text{cand}})}{\widehat{p}(\mathbf{y}^{(m)} \mid \boldsymbol{\theta}^{(i)}, \mathbf{u}^{(i)})\, p(\boldsymbol{\theta}^{(i)})\, q(\boldsymbol{\theta}^{\text{cand}} \mid \boldsymbol{\theta}^{(i)})}\right\}$$

7: Draw $\zeta \sim \mathcal{U}(0, 1)$
8: **if** $\zeta \leq \alpha$ **then**
9: Accept the proposal: $(\boldsymbol{\theta}^{(i+1)}, \mathbf{u}^{(i+1)}) = (\boldsymbol{\theta}^{\text{cand}}, \mathbf{u}^{\text{cand}})$
10: **else**
11: Set $(\boldsymbol{\theta}^{(i+1)}, \mathbf{u}^{(i+1)}) = (\boldsymbol{\theta}^{(i)}, \mathbf{u}^{(i)})$
12: **end if**
13: **end for**
14: Discard the first B iterations and return $\{\boldsymbol{\theta}(B+1), ..., \boldsymbol{\theta}^{(I)}\}$

correlating the auxiliary random variables u used to compute the likelihood estimator at successive MCMC iterations [14]. At iteration i, given the current state $(\boldsymbol{\theta}, \mathbf{u})$, the algorithm proceeds as follows:

1. Propose a new value $\boldsymbol{\theta}^{\text{cand}} \sim q_\theta(\cdot \mid \boldsymbol{\theta}^{(i)})$.
2. Correlated auxiliary variables: Sample $\mathbf{u}^{\text{cand}} \sim \mathcal{N}(\rho \mathbf{u}^{(i)}, (1-\rho^2)\mathbf{I})$, where $\rho \in [0, 1)$ controls the level of correlation and $\mathbf{I}$ is the identity matrix.
3. Compute the likelihood estimate: $\widehat{p}(\mathbf{y}^{(m)} \mid \theta^{\text{cand}}, \mathbf{u}^{\text{cand}})$ using a particle filter (e.g., bootstrap filter). Note that $\widehat{p}(\mathbf{y}^{(m)} \mid \boldsymbol{\theta}^{(i)}, \mathbf{u}^{(i)})$ is already available from previous iteration.
4. With probability α given by equation (10), accept the proposal by setting $(\boldsymbol{\theta}^{(i+1)}, u^{(i+1)}) = (\boldsymbol{\theta}^{\text{cand}}, \mathbf{u}^{\text{cand}})$; otherwise, set $(\boldsymbol{\theta}^{(i+1)}, \mathbf{u}^{(i+1)}) = (\boldsymbol{\theta}^{(i)}, \mathbf{u}^{(i)})$.

This approach leverages correlation in the auxiliary space to stabilize the Metropolis-Hastings ratio and increase acceptance rates, particularly when the number of particles is small [57].

4.7. Metropolis-adjusted Langevin algorithm (MALA)

The Metropolis-adjusted Langevin algorithm (MALA) is a MCMC method that i ncorporates gradient information to guide the proposals more effectively. MALA uses a first-order Langevin diffusion process to propose moves in directions of increasing posterior density [47, 31, 37]. The proposal is defined as:

$$\boldsymbol{\theta}^{\text{cand}} \sim q(\boldsymbol{\theta}^{\text{cand}}|\boldsymbol{\theta}) = \mathcal{N}\left(\boldsymbol{\theta} + \frac{\epsilon^2}{2}\mathbf{M}\,\nabla_{\boldsymbol{\theta}} \log p\left(\boldsymbol{\theta} \mid \mathbf{y}^{(m)}\right),\, \epsilon^2 \mathbf{M}\right)$$

where ϵ is a step size parameter, $\mathbf{M}$ is a preconditioning matrix, $\nabla_{\boldsymbol{\theta}} \log p\left(\boldsymbol{\theta} \mid \mathbf{y}^{(m)}\right)$ is the gradient of the posterior distribution and the acceptance probability follows the standard MH ratio adjusted for the asymmetric proposal, i.e., $q(\boldsymbol{\theta}|\boldsymbol{\theta}^{\text{cand}}) \neq q(\boldsymbol{\theta}^{\text{cand}}|\boldsymbol{\theta})$. MALA offers improved convergence properties compared to random walk MH, especially in high-dimensional spaces or when the posterior is strongly correlated. In practice, the gradient must be computed or approximated, and the step size ϵ must be carefully tuned to balance efficiency and acceptance rate. Optimal

tuning theory and empirical studies suggest that choosing ϵ so that the average acceptance probability is around 0·574 leads to good performance in many settings [45]. In many practical scenarios, the preconditioning matrix is unknown beforehand. A common adaptive strategy is to initially set $\mathbf{M}$ as the identity matrix and then estimate the covariance matrix $\hat{\Sigma}$ of the target distribution from the MCMC samples obtained during initial pilot runs. The preconditioning matrix $\mathbf{M}$ is subsequently set to $\hat{\Sigma}$. Algorithm 3 outlines the steps to sample from the posterior distribution using MALA.

Algorithm 3 Metropolis-adjusted Langevin algorithm (MALA)

Require: Data $\mathbf{y}^{(m)}$, initial value $\boldsymbol{\theta}^{(1)}$, step size ϵ, preconditioning matrix $\mathbf{M}$, number of iterations I, burn-in B
Ensure: Samples $\{\boldsymbol{\theta}^{(B+1)}, \ldots, \boldsymbol{\theta}^{(I)}\}$

for $i = 1$ to I **do**
  Draw $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  Propose candidate:

$$\boldsymbol{\theta}^{\text{cand}} = \boldsymbol{\theta}^{(i)} + \frac{\epsilon^2}{2}\mathbf{M}\nabla_\theta \log p(\boldsymbol{\theta}^{(i)}|\mathbf{y}^{(m)}) + \epsilon\sqrt{\mathbf{M}}\mathbf{Z}$$

  Compute acceptance probability:

$$\alpha = \min\left\{1, \frac{p(\mathbf{y}^{(m)}|\boldsymbol{\theta}^{\text{cand}})\, p(\boldsymbol{\theta}^{\text{cand}})\, q(\boldsymbol{\theta}^{(i)}|\boldsymbol{\theta}^{\text{cand}})}{p(\mathbf{y}^{(m)}|\boldsymbol{\theta}^{(i)})\, p(\boldsymbol{\theta}^{(i)})\, q(\boldsymbol{\theta}^{\text{cand}}|\boldsymbol{\theta}^{(i)})}\right\}$$

  Draw $\zeta \sim \mathcal{U}(0, 1)$
  if $\zeta \leq \alpha$ **then**
    Accept the proposal $\boldsymbol{\theta}^{(i+1)} = \boldsymbol{\theta}^{\text{cand}}$
  else
    Set $\boldsymbol{\theta}^{(i+1)} = \boldsymbol{\theta}^{(i)}$
  end if
end for
Discard the first B iterations and return $\{\boldsymbol{\theta}^{(B+1)}, \ldots, \boldsymbol{\theta}^{(I)}\}$

4.8. Hamiltonian Monte Carlo (HMC)

Hamiltonian Monte Carlo (HMC) is an MCMC method originally developed for molecular dynamics [19] and later adapted for Bayesian inference in high-dimensional spaces [36, 5]. HMC improves sampling efficiency by introducing auxiliary momentum variables and simulating Hamiltonian dynamics to explore the posterior distribution more effectively. Let $\boldsymbol{\theta} \in \mathbb{R}^d$ be the parameter vector, $p\left(\boldsymbol{\theta} \mid \mathbf{y}^{(m)}\right)$ denote the posterior density of $\boldsymbol{\theta}$, and $\boldsymbol{p} \in \mathbb{R}^d$ an auxiliary momentum variable drawn from a Gaussian distribution $\mathcal{N}(0, \mathbf{M})$, where $\boldsymbol{\theta}$ is is interpreted as the position of a particle and $-\log p\left(\boldsymbol{\theta} \mid \mathbf{y}^{(m)}\right)$ escribes its potential energy while $\boldsymbol{p}$ is the momentum with kinetic energy $\frac{1}{2}\boldsymbol{p}'\mathbf{M}^{-1}\boldsymbol{p}$ then the total energy of a closed system is the Hamiltonian function,

$$H(\boldsymbol{\theta}, \boldsymbol{p}) = U(\boldsymbol{\theta}) + K(\boldsymbol{p}) = -\log p(\boldsymbol{\theta} \mid \mathbf{y}^{(m)}) + \frac{1}{2}\boldsymbol{p}'\mathbf{M}^{-1}\boldsymbol{p}$$

where $U(\boldsymbol{\theta}) = -\log p(\boldsymbol{\theta} \mid \mathbf{y}^{(m)})$, $K(\boldsymbol{p}) = \frac{1}{2}\boldsymbol{p}'\mathbf{M}^{-1}\boldsymbol{p}$ and let $\mathcal{L}(\theta) = \log p(\boldsymbol{\theta} \mid \mathbf{y}^{(m)})$. The (unormalized) joint density of $(\boldsymbol{\theta}, \boldsymbol{p})$ is then given by,

$$f(\boldsymbol{\theta}, \boldsymbol{p}) \propto p(\boldsymbol{\theta} \mid \mathbf{y}^{(m)}) \exp\left(-\frac{1}{2}\boldsymbol{p}'\mathbf{M}^{-1}\boldsymbol{p}\right) \propto \exp\left(-H(\boldsymbol{\theta}, \boldsymbol{p})\right)$$

For continuous time t, the system evolves according to Hamilton's equations:

$$\frac{\partial \boldsymbol{\theta}}{\partial t} = \frac{\partial H(\boldsymbol{\theta}, \boldsymbol{p})}{\partial \boldsymbol{p}} = \mathbf{M}^{-1}\boldsymbol{p},$$
$$\frac{\partial \boldsymbol{p}}{\partial t} = -\frac{\partial H(\boldsymbol{\theta}, \boldsymbol{p})}{\partial \boldsymbol{\theta}} = -\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta} \mid \mathbf{y}^{(m)})$$

where $\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta} \mid \mathbf{y}^{(m)})$ is the gradient of $\mathcal{L}(\theta)$ with respect to $\boldsymbol{\theta}$. Since these equations cannot be solved analytically in most applications, a symplectic integrator such as the leapfrog method is used [36, 23], which discretizes the Hamiltonian dynamics as the following steps:

$$\boldsymbol{p}^{(\tau+\frac{\epsilon}{2})} = \boldsymbol{p}^{(\tau)} + \frac{\epsilon}{2}\nabla_{\boldsymbol{\theta}}\mathcal{L}\left(\boldsymbol{\theta}^{(\tau)}\right)$$
$$\boldsymbol{\theta}^{(\tau+\epsilon)} = \boldsymbol{\theta}^{(\tau)} + \epsilon\mathbf{M}^{-1}\boldsymbol{p}^{(\tau+\frac{\epsilon}{2})}$$
$$\boldsymbol{p}^{(\tau+\epsilon)} = \boldsymbol{p}^{(\tau+\frac{\epsilon}{2})} + \frac{\epsilon}{2}\nabla_{\boldsymbol{\theta}}\mathcal{L}\left(\boldsymbol{\theta}^{(\tau+\epsilon)}\right)$$

for some user specified small step-size $\epsilon > 0$.

HMC alternates between sampling a new momentum and proposing a new $(\boldsymbol{\theta}, \boldsymbol{p})$ pair by simulating Hamiltonian dynamics. The candidate is accepted or rejected based on the change in Hamiltonian energy. HMC can yield higher effective sample sizes per iteration than standard random walk proposals, especially in complex posteriors. In summary, the steps for sampling model parameters using HMC are presented in Algorithm 4.

The performance of gradient-based samplers such as MALA and HMC depends strongly on the tuning parameters, particularly the step size ϵ, the number of leapfrog steps L, and the mass matrix $\mathbf{M}$. Poor tuning may lead to unstable Hamiltonian trajectories, low acceptance probabilities, or inefficient random-walk behavior, especially in high-dimensional SDEMEMs. In this work, the gradients required for MALA and HMC were computed from the approximate log-posterior distributions induced by the Euler–Maruyama (EM), modified diffusion bridge (MDB), and residual diffusion bridge (RDB) discretization schemes. By discretizing the latent diffusion process, the corresponding transition densities admit tractable Gaussian approximations, allowing stable evaluation of posterior gradients with respect to the model parameters. Importantly, the particle-filter likelihood estimator was not directly differentiated. Instead, particle filtering was used exclusively within the pseudo-marginal PMMH framework to construct unbiased likelihood estimators. The MALA and HMC algorithms were therefore applied to approximate posterior distributions conditioned on augmented latent trajectories, avoiding the noisy gradient estimates typically associated with particle-filter differentiation. Consequently, the gradients used during the MALA and HMC updates remained deterministic conditional on the latent augmented states, improving numerical stability during proposal generation. The implementation of MALA and HMC requires repeated evaluations of $\nabla_{\boldsymbol{\theta}} \log p\left(\boldsymbol{\theta} \mid \mathbf{y}^{(m)}\right)$.

4.9. Individual-Augmentation pseudo-marginal (IA)

The Individual-Augmentation (IA) algorithm proposed by [6] estimates the marginal likelihood of each individual by integrating over their random effects via importance sampling:

$$\widehat{p}\left(\mathbf{y}^{(m)} \mid \boldsymbol{\theta}\right) \propto \frac{1}{L_{\text{re}}} \sum_{l=1}^{L_{\text{re}}} \frac{\widehat{p}\left(\mathbf{y}^{(m)} \mid \phi_X, \mathbf{b}_l^{(m)}, \xi\right) p\left(\mathbf{b}_l^{(m)} | \psi\right)}{g\left(\mathbf{b}_l^{(m)} | \psi\right)}, \quad \textit{where} \quad \mathbf{b}_l^{(m)} \sim g\left(\mathbf{b}^{(m)} | \psi\right).$$

Here, $g(\cdot|\psi)$ is an importance proposal, and the weights adjust for the discrepancy between the proposal and prior. The Laplace approximation is employed for $p(\mathbf{b}^{(m)}|\psi)$ using an approximate latent trajectory $\mathbf{x}^{(m)}$, computed from one of three numerical discretization schemes [6].

The Individual-Augmentation (IA) pseudo-marginal algorithm combines importance sampling over the random effects with particle filtering for latent state estimation. At each MCMC iteration, the likelihood is approximated through Monte Carlo integration using the procedure summarized in Algorithm 6. The resulting unbiased likelihood estimator is then embedded within a pseudo-marginal Metropolis–Hastings framework, as described in Algorithm 5. Algorithm 5 follows the standard pseudo-marginal PMMH framework described in [1].

Algorithm 4 Hamiltonian Monte Carlo (HMC)

Require: Data $\mathbf{y}^{(m)}$, initial value $\boldsymbol{\theta}^{(1)}$, step size ϵ, number of leapfrog steps L, mass matrix $\mathbf{M}$, number of iterations I, burn-in B

Ensure: Posterior samples $\{\boldsymbol{\theta}^{(B+1)}, \ldots, \boldsymbol{\theta}^{(I)}\}$

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

  Sample momentum: $\boldsymbol{p}^{(i)} \sim \mathcal{N}(0, \mathbf{M}^{-1})$

  Set $\boldsymbol{\theta}^* = \boldsymbol{\theta}^{(i)}$ and $\mathbf{p}^* = \mathbf{p}^{(i)}$

  Make a half-step update on the momentum:

$$\mathbf{p}^* = \mathbf{p}^* + \frac{\epsilon}{2} \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}^* \mid \mathbf{y}^{(m)})$$

  Simulate leapfrog dynamics: For $\ell = 1$ to L

  Update position:

$$\boldsymbol{\theta}^* = \boldsymbol{\theta}^* + \epsilon \mathbf{M}^{-1} \mathbf{p}^*$$

  if $\ell < L$ **then**

    Update momentum:

$$\mathbf{p}^* = \mathbf{p}^* + \epsilon \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}^* \mid \mathbf{y}^{(m)})$$

  end if

end for

Make a final half-step update on the momentum:

$$\mathbf{p}^* = \mathbf{p}^* + \frac{\epsilon}{2} \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}^* \mid \mathbf{y}^{(m)})$$

Negate momentum: $\mathbf{p}^* = -\mathbf{p}^*$

Compute acceptance probability:

$$\alpha = \min \left\{ 1, \exp \left[-H(\boldsymbol{\theta}^*, \mathbf{p}^*) + H(\boldsymbol{\theta}^{(i)}, \mathbf{p}^{(i)}) \right] \right\}$$

Draw $\zeta \sim \mathcal{U}(0, 1)$

if $\zeta \leq \alpha$ **then** Accept the proposal: $\boldsymbol{\theta}^{(i+1)} = \boldsymbol{\theta}^*$

else Reject the proposal: $\boldsymbol{\theta}^{(i+1)} = \boldsymbol{\theta}^{(i)}$

end if

Discard the first B iterations and return $\{\boldsymbol{\theta}^{(B+1)}, \ldots, \boldsymbol{\theta}^{(I)}\}$

5. Applications

In this section, we illustrate and compare the proposed algorithms using two case studies: a real dataset on orange tree growth and a simulated dataset from a bivariate Ornstein–Uhlenbeck (OU) process. Correlated versions of the MCMC methods follow the approach of [57].

5.1. Orange tree growth model

We analyze the orange tree growth data from [46], which contains seven circumference measurements (in mm) for each of five orange trees, recorded at known time points $t_j^{(m)}$ (in days), with $j = 1, \ldots, 7$ and $m = 1, \ldots, 5$.

Algorithm 5 Individual-Augmentation Pseudo-Marginal (IA)

Require: Data $\mathbf{y}^{(m)}$, initial values $(\boldsymbol{\theta}^{(1)}, \mathbf{u}^{(1)})$, number of iterations I, burn-in B
Ensure: Posterior samples $\{\boldsymbol{\theta}^{(B+1)}, \ldots, \boldsymbol{\theta}^{(n_{\text{iter}})}\}$
1: Draw auxiliary variables $\mathbf{u}^{\text{cand}} \sim p(\mathbf{u}^{(1)})$.
2: Compute the likelihood estimate $\widehat{p}\left(\mathbf{y}^{(m)}|\boldsymbol{\theta}^{(1)}, \mathbf{u}^{(1)}\right)$ using Algorithm 6
3: **for** $i = 1$ to I **do**
4: Propose $\boldsymbol{\theta}^{\text{cand}} \sim q(\boldsymbol{\theta}^{\text{cand}}|\boldsymbol{\theta}^{(i-1)})$ and $\mathbf{u}^{\text{cand}} \sim p(\mathbf{u}^{(i)})$
5: Compute the estimate $\widehat{p}\left(\mathbf{y}^{(m)} \mid \boldsymbol{\theta}^{\text{cand}}, \mathbf{u}^{\text{cand}}\right)$ using the likelihood estimation procedure described in Algorithm 6.
6: Compute acceptance probability:

$$\alpha = \min\left\{1, \frac{\widehat{p}\left(\mathbf{y}^{(m)} \mid \boldsymbol{\theta}^{\text{cand}}, \mathbf{u}^{\text{cand}}\right) p\left(\boldsymbol{\theta}^{\text{cand}}\right) q\left(\boldsymbol{\theta}^{(i)}|\boldsymbol{\theta}^{\text{cand}}\right)}{\widehat{p}\left(\mathbf{y}^{(m)} \mid \boldsymbol{\theta}^{(i)}, \mathbf{u}^{(i)}\right) p\left(\boldsymbol{\theta}^{(i)}\right) q\left(\boldsymbol{\theta}^{\text{cand}}|\boldsymbol{\theta}^{(i)}\right)}\right\}$$

7: Draw $\zeta \sim \mathcal{U}(0,1)$
8: **if** $\zeta \leq \alpha$ **then**
9: Accept the proposal $(\boldsymbol{\theta}^{(i+1)}, \mathbf{u}^{(i+1)}) = (\boldsymbol{\theta}^{\text{cand}}, \mathbf{u}^{\text{cand}})$
10: **else**
11: Set $(\boldsymbol{\theta}^{(i+1)}, \mathbf{u}^{(i+1)}) = (\boldsymbol{\theta}^{(i)}, \mathbf{u}^{(i)})$
12: **end if**
13: **end for**
14: Discard first B iterations and return $\{\boldsymbol{\theta}^{(B+1)}, \ldots, \boldsymbol{\theta}^{(I)}\}$

Algorithm 6 Estimating the likelihood in IA

Require: Data $\mathbf{y}^{(m)}$, parameter vector $\boldsymbol{\theta}$, number of random effects draws L_{re}, number of particles N, auxiliary variables $\mathbf{u}$
Ensure: Likelihood estimate $\widehat{p}(\mathbf{y}^{(m)} \mid \boldsymbol{\theta}, \mathbf{u})$
1: **for** $m = 1$ to K **do**
2: **for** $l = 1$ to L_{re} **do**
3: Draw $\mathbf{b}_l^{(m)} \sim g\left(\mathbf{b}^{(m)}|\psi\right)$
4: Run the particle filter with N particles to obtain $\widehat{p}\left(y^{(m)} \mid \mathbf{b}_l^{(m)}, \phi_X, \xi, \mathbf{u}\right)$
5: Compute importance weight:

$$Z_l^{(m)} = \frac{\widehat{p}\left(y^{(m)} \mid \phi_X, \mathbf{b}_l^{(m)}, \xi, \mathbf{u}\right) \cdot p\left(\mathbf{b}_l^{(m)}|\psi\right)}{g\left(\mathbf{b}_l^{(m)}|\psi\right)}$$

6: **end for**
7: Compute: $\widehat{p}\left(y^{(m)}|\boldsymbol{\theta}, \mathbf{u}\right) = \frac{1}{L_{\text{re}}} \sum_{l=1}^{L_{\text{re}}} Z_l^{(m)}$
8: **end for**
9: Compute full data likelihood: $\widehat{p}\left(\mathbf{y}^{(m)}|\boldsymbol{\theta}, \mathbf{u}\right) = \prod_{m=1}^{K} \widehat{p}\left(y^{(m)}|\boldsymbol{\theta}, \mathbf{u}\right)$

5.1.1. Model specification. The orange tree growth dynamics are modeled through the following stochastic differential equation mixed-effects model (SDEMEM):

$$dX_t^{(m)} = \frac{1}{\phi_1^{(m)}\phi_2^{(m)}} X_t^{(m)} \left(\phi_1^{(m)} - X_t^{(m)}\right) dt + \sigma\sqrt{X_t^{(m)}}\, dW_t^{(m)}, \qquad X_0^{(m)} = x_0^{(m)}, \tag{11}$$

$$\phi_1^{(m)} \sim \mathcal{N}(\mu_{\phi_1}, \sigma_{\phi_1}^2), \qquad \phi_2^{(m)} \sim \mathcal{N}(\mu_{\phi_2}, \sigma_{\phi_2}^2). \tag{12}$$

The parameter $\phi_1^{(m)}$ (measured in mm) represents the asymptotic circumference of the m-th tree, whereas $\phi_2^{(m)}$ (measured in days) controls the growth rate. The drift component therefore corresponds to a stochastic logistic growth model with individual-specific random effects. Observed measurements are modeled through

$$Y_t^{(m)} = X_t^{(m)} + \epsilon_t^{(m)}, \qquad \epsilon_t^{(m)} \overset{\text{iid}}{\sim} \mathcal{N}(0, \xi^2) \tag{13}$$

Weakly informative prior distributions were assigned to the diffusion coefficient σ, the measurement error parameter ξ, and the population-level parameters

$$\psi = (\mu_{\phi_1}, \tau_{\phi_1}, \mu_{\phi_2}, \tau_{\phi_2})',$$

where

$$\tau_{\phi_1} = \sigma_{\phi_1}^{-2}, \qquad \tau_{\phi_2} = \sigma_{\phi_2}^{-2}$$

denote the corresponding precision parameters.
The joint prior distribution was specified as

$$\begin{aligned} p(\boldsymbol{\theta}) = \text{Ga}(\sigma \mid 1, 0.01) \cdot \text{Ga}(\xi \mid 1, 0.01) \cdot \mathcal{N}(\mu_{\phi_1} \mid 0, 100^2) \cdot \text{Ga}(\tau_{\phi_1} \mid 1, 0.01) \\ \cdot \mathcal{N}(\mu_{\phi_2} \mid 0, 100^2) \cdot \text{Ga}(\tau_{\phi_2} \mid 1, 0.01), \end{aligned}$$

where $\text{Ga}(\cdot \mid a, b)$ denotes a Gamma distribution with shape parameter, a and rate parameter b.

5.1.2. Computational setup. We implemented IA, MALA, and HMC under three discretization schemes: Euler-Maruyama (EM), modified diffusion bridge (MDB), and residual diffusion bridge (RDB). Settings: $P = 50$ steps in discretization schemes, $N = 20$ particles, $L_{\text{re}} = 20$ random effects per individual, $I = 2000$ iterations, $B = 200$ iterations were discarded (burn-in period). MALA step sizes: $\epsilon = 0.20$ (EM), 0.75 (MDB/RDB). HMC: $\epsilon = 1/15$, $L = 15$. These tuning parameters, including the preconditioning matrix $\mathbf{M}$, were selected based on preliminary runs to improve acceptance rates and effective sample sizes (ESS). Matrix $\mathbf{M}$ was chosen as a diagonal matrix based on preliminary estimates of posterior scale obtained from pilot runs.
The tuning parameters were selected empirically through preliminary pilot simulations aimed at achieving stable acceptance probabilities and satisfactory effective sample size (ESS) values. For MALA and HMC, the step size ϵ was adjusted to balance posterior exploration and proposal stability. Excessively small values of ϵ produced slow random-walk behavior, whereas excessively large values reduced acceptance rates and occasionally generated unstable proposals. In HMC, the number of leapfrog steps L and the mass matrix $\mathbf{M}$ were chosen to maintain efficient exploration while avoiding unstable Hamiltonian trajectories. For pseudo-marginal implementations, the number of particles N used in the particle filter was selected to provide sufficiently stable likelihood estimates without excessive computational cost. In correlated pseudo-marginal methods, the correlation coefficient ρ was chosen close to one in order to reduce the variance of successive likelihood ratios while preserving adequate posterior exploration. Consequently, the selected tuning parameters should be interpreted as problem-dependent empirical choices rather than universally optimal values.

5.1.3. Results and interpretation. In Table 1, we report the performance of various MCMC methods (IA, MALA, and HMC) combined with the discretization schemes Euler–Maruyama (EM), modified diffusion bridge (MDB),

and residual diffusion bridge (RDB) for the orange tree growth model. The results indicate that the HMC-based methods generally achieved higher effective sample sizes (ESS) and acceptance rates than the competing approaches, although at increased computational cost due to repeated gradient evaluations.
Among the methods considered, HMC-MDB achieved the largest ESS/sec value (0.78 ESS/sec) together with a relatively high acceptance rate (51.93%), suggesting a favorable empirical trade-off between sampling efficiency and computational cost for this application. HMC-EM yielded the shortest total runtime (134.86 seconds), whereas HMC-RDB produced the highest acceptance rate (56.34%). Overall, the HMC-based methods exhibited competitive performance relative to the IA and MALA approaches in terms of ESS, ESS/sec, and acceptance probability.
From Table 1, we additionally observe that MALA-EM, HMC-EM, and HMC-MDB generated less correlated posterior samples, as reflected by their larger ESS values and ESS/sec rates. Figure 1 provides a visual comparison of these methods through posterior trace plots, histograms, and autocorrelation functions (ACF). The trace plots suggest satisfactory mixing behavior for MALA-EM and HMC-MDB, while HMC-EM exhibited slightly slower exploration of the posterior distribution. The corresponding ACF plots indicate lower autocorrelation levels for MALA-EM and HMC-MDB relative to the remaining methods considered.

Method	Time (sec)	ESS	ESS/sec	Acceptance Rate (%)
IA-EM	335.52	24.28	0.07	7.40
IA-MDB	576.93	70.96	0.12	18.21
IA-RDB	1401.30	62.11	0.04	15.41
MALA-EM	141.75	42.40	0.30	43.22
MALA-MDB	159.70	81.51	0.51	55.31
MALA-RDB	486.77	71.58	0.15	46.99
HMC-EM	134.86	75.48	0.56	38.19
HMC-MDB	161.58	126.75	0.78	51.93
HMC-RDB	524.77	122.25	0.23	56.34

Table 1. Orange tree growth model: runtime, effective sample size (ESS), ESS/sec, and acceptance rate for each method and discretization scheme.

5.2. Ornstein–Uhlenbeck (OU)

We simulate data from a bivariate OU process, commonly used in pharmacokinetics, neuroscience, and finance [22, 15, 42].

5.2.1. Model specification. The process is modeled as:

$$dX_t^{(m)} = (\beta * b^{(m)})(\alpha - X_t^{(m)})dt + \sigma dW_t^{(m)}, \quad X_0^{(m)} = x_0^{(m)} \tag{14}$$

$$Y_t^{(m)} = X_t^{(m)} + \epsilon_t^{(m)}, \quad \epsilon_t^{(m)} \overset{\text{iid}}{\sim} \mathcal{N}(0, \Sigma) \tag{15}$$

where $*$ denotes elementwise multiplication. Matrices:

$$\alpha = \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix}, \quad X_t^{(m)} = \begin{bmatrix} X_t^{(1,m)} \\ X_t^{(2,m)} \end{bmatrix}, \quad \beta = \begin{bmatrix} \beta_{11} & \beta_{12} \\ \beta_{21} & \beta_{22} \end{bmatrix},$$

$$b^{(m)} = \begin{bmatrix} b_{11}^{(m)} & b_{12}^{(m)} \\ b_{21}^{(m)} & b_{22}^{(m)} \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \xi_1^2 & 0 \\ 0 & \xi_2^2 \end{bmatrix}.$$

5.2.2. Computational setup. We simulate $M = 5$ individuals with $j = 20$ time points, $\Delta t = 0.1$, and parameters:

$$\boldsymbol{\theta} = (3, 2.5, 1.8, 2, 1, 1.5, 0.3, 0.5, 0.1, 0.5), \quad \boldsymbol{\psi} = (45, 100, 100, 25)$$

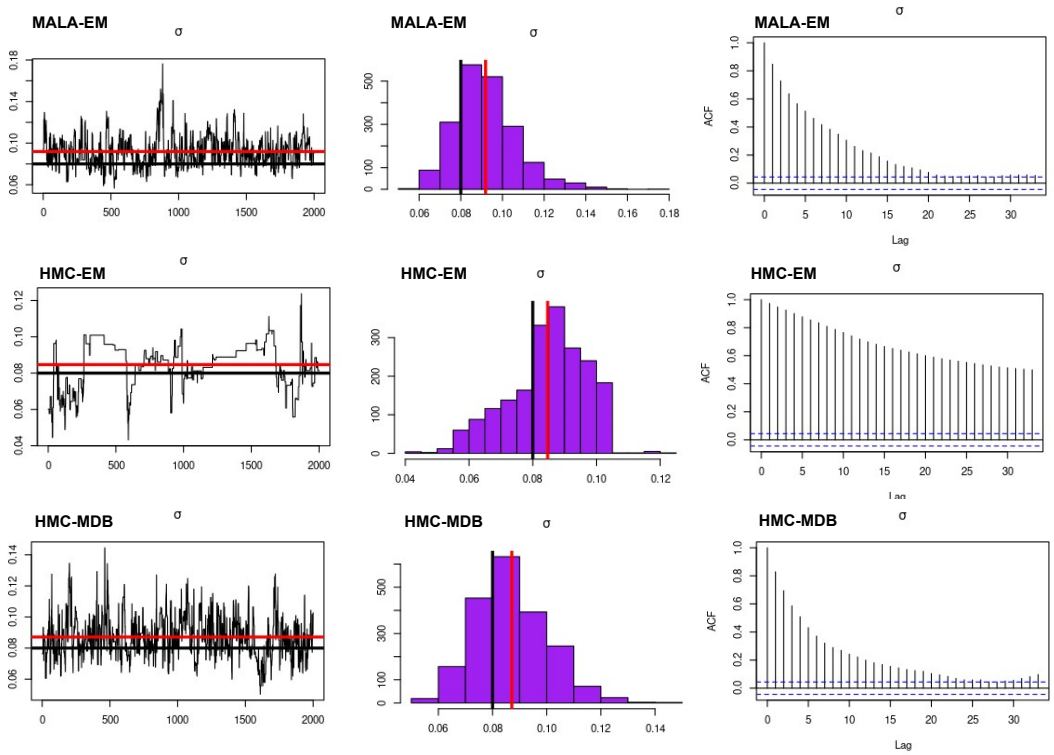

Figure 1. Trace plots, histograms, and ACFs of parameter σ using MALA-EM, HMC-EM, and HMC-MDB. Red lines indicate posterior mean; black lines denote initial values.

We apply MH, MALA, and HMC (with/without correlation) using three update blocks (random effects, parameters, hyperparameters), with $N = 50$ particles, $I = 5000$ iterations, $B = 2500$ iterations were discarded (burn-in period), $\epsilon = 1/10$, and $L = 10$ for HMC. These tuning parameters, including the preconditioning matrix $\mathbf{M}$, were selected based on preliminary runs to improve acceptance rates and effective sample sizes (ESS). In the correlated versions we use $\rho = 0.999$.

5.2.3. Results and interpretation. Table 2 summarizes the performance of different MCMC methods and their correlated versions applied to the bivariate Ornstein–Uhlenbeck (OU) process. The metrics reported include total runtime (in seconds), effective sample size (ESS), ESS per second (ESS/sec), and acceptance rate. Among the methods evaluated in this experiment, the correlated MALA-Gibbs method achieved the highest ESS/sec (14.18) and the highest ESS/sec (0.0037), indicating superior sampling efficiency. Although this method does not have the highest acceptance rate, its ability to generate less correlated and more informative samples within a shorter runtime (3806.18 seconds) makes it the most efficient overall. The standard MALA-Gibbs method shows the lowest acceptance rate (0.40%), which likely hinders its performance despite having a slightly higher ESS than MH-Gibbs. The standard MH-Gibbs and its correlated version perform poorly in terms of ESS and ESS/sec, though the correlated version slightly improves both metrics and shows a substantially higher acceptance rate (18.29% vs. 2.40%). HMC-Gibbs and its correlated version show balanced performance with moderate ESS and ESS/sec values, and relatively high acceptance rates (14.73% and 25.57%, respectively). Notably, the correlated HMC-Gibbs achieves comparable efficiency to its standard counterpart but with slightly reduced runtime and improved acceptance rate. Overall, the results obtained in these experiments suggest that incorporating correlation into MCMC updates may improve sampling efficiency. Figure 2 presents the trace plots of posterior samples obtained from the MCMC methods (MH-Gibbs, MALA, HMC) and their correlated variants. We observe satisfactory convergence for both the standard and correlated HMC methods. The trace plots illustrate the behavior of the

posterior samples for the parameter α_2 under different sampling schemes. The correlated variants generally exhibit improved mixing compared to their standard counterparts, as evidenced by reduced autocorrelation and smoother trajectories. In particular, the correlated MALA and correlated HMC methods show better exploration of the parameter space, leading to higher effective sample sizes (ESS) as indicated in Table 2. The Gibbs-based methods, especially MH-Gibbs, display higher autocorrelation and slower convergence, as seen in the fluctuating trace plots. Overall, the results suggest that incorporating correlation into the proposals enhances sampling efficiency, particularly for gradient-based methods like MALA and HMC.

Method	Time (sec)	ESS	ESS/sec	Acceptance Rate (%)
MH-Gibbs	4191.00	5.10	0.0012	2.40
Corr. MH-Gibbs	3901.66	4.92	0.0013	18.29
MALA-Gibbs	4271.93	5.58	0.0013	0.40
Corr. MALA-Gibbs	3806.18	14.18	0.0037	16.52
HMC-Gibbs	4131.88	10.70	0.0026	14.73
Corr. HMC-Gibbs	3934.39	10.19	0.0026	25.57

Table 2. Bivariate OU process: runtime, effective sample size (ESS), ESS/sec, and acceptance rate.

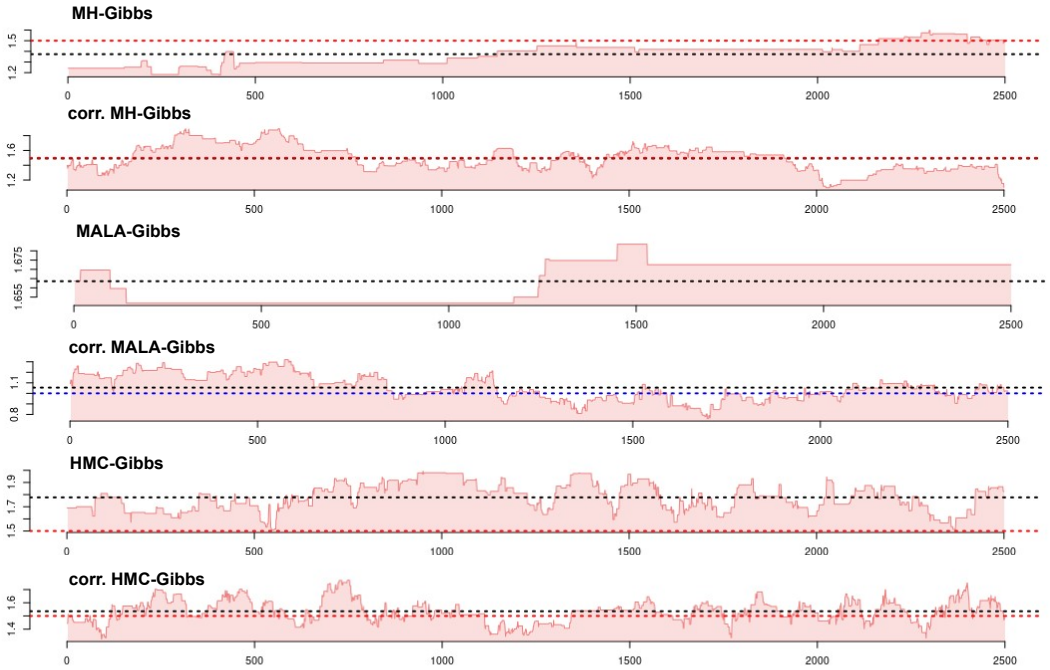

Figure 2. Trace plots of parameter α_2 for all MCMC methods (with and without correlation) under the bivariate OU model. Blue lines show initial values; red lines denote true values and posterior means.

Due to computational constraints, convergence assessment in this work relied primarily on trace plots and ESS diagnostics. Although multiple independent chains and Gelman–Rubin diagnostics would provide additional validation, they were not systematically implemented in the present study and remain an important direction for future work.

5.3. Practical considerations, limitations, and tuning issues

The numerical experiments suggest that no single method is uniformly optimal for all SDEMEM settings. The relative performance of the algorithms depends on the dimensionality of the latent process, the availability of

gradient information, the variance of the likelihood estimator, and the available computational resources. While gradient-based methods such as MALA and HMC often improve sampling efficiency, they also require careful tuning and additional computational effort. Similarly, advanced bridge constructions such as MDB and RDB may improve proposal quality at the expense of increased implementation complexity and runtime.

The comparison between MDB and RDB highlights an important practical trade-off between proposal quality

Table 3. Practical comparison of the MCMC and bridge methods considered in this work.

Method	**Advantages**	**Limitations**	**Recommended use**
IA-EM	Simple implementation	Low ESS and poor mixing	Small problems or baseline comparisons
MALA-EM	Faster exploration than MH	Sensitive to tuning parameters	Moderate-dimensional posteriors with available gradients
HMC-MDB	High ESS/sec in our experiments	Expensive gradient evaluations	Complex correlated posteriors when computational resources are available
HMC-RDB	High acceptance rates	High runtime and implementation complexity	Highly nonlinear diffusion paths
Correlated PMMH	Stabilizes noisy likelihood ratios	Requires careful particle-filter tuning	Intractable likelihoods without gradient access
MDB	Good balance between efficiency and cost	Less accurate conditioning than RDB	General-purpose diffusion bridge
RDB	Improved conditioned proposals	Expensive ODE corrections	Difficult nonlinear latent dynamics

and computational complexity. Although RDB generally produces better conditioned proposals by incorporating deterministic residual corrections, the empirical gains in ESS and acceptance rate were often moderate relative to the substantial increase in computational cost. In the orange tree application, for instance, HMC-RDB achieved only a marginal improvement in acceptance rate compared with HMC-MDB, while requiring more than three times the runtime. These results suggest that, for moderately nonlinear SDEMEMs, MDB may offer a more favorable balance between implementation simplicity, computational efficiency, and sampling performance. Nevertheless, RDB may remain advantageous in highly nonlinear or sparsely observed systems where accurate bridge conditioning becomes critical for avoiding trajectory mismatch.

The performance of MALA and HMC strongly depends on the tuning parameters, particularly the step size ϵ, the number of leapfrog steps L, and the choice of the preconditioning or mass matrix. Poor tuning may lead to random-walk behavior, low acceptance rates, or unstable Hamiltonian trajectories. In high-dimensional SDEMEMs, these effects become more pronounced due to the strong posterior correlations induced by latent diffusion paths. Therefore, the tuning parameters used in this work should be interpreted as problem-dependent choices obtained through preliminary pilot runs, rather than universally optimal values.

The efficiency of MALA and HMC strongly depends on tuning parameters such as the step size ϵ, the number of leapfrog steps L, and the preconditioning matrix Σ. Preliminary pilot runs suggested that:

- Decreasing ϵ increases acceptance rates but substantially lowers ESS due to smaller exploration steps;
- Increasing ϵ reduces acceptance rates and may generate unstable proposals;
- HMC is generally more robust than MALA to moderate perturbations of ϵ;
- MDB-based methods exhibit greater stability than EM-based proposals under tuning perturbations.

Finally, the scalability of the proposed methods poses important computational challenges. The computational cost generally increases with the number of individuals, observation times (T), and discretization levels. In pseudo-marginal approaches, larger values of T typically increase the variance of the estimated likelihood, requiring a larger number of particles (N) to maintain adequate mixing performance, which further increases the computational burden and may exacerbate particle degeneracy. Consequently, while HMC remains attractive for moderately high-dimensional problems due to its use of gradient information, simpler MH-based approaches may remain computationally cheaper per iteration in large-scale settings. Future research will focus on adaptive HMC strategies, guided proposals, and scalable particle filtering methods for high-dimensional SDEMEMs.
In our implementation, the gradients required by MALA and HMC were computed from the discretized approximate posterior distributions induced by the EM, MDB, and RDB schemes. Since the particle-filter likelihood estimator was not directly differentiated, the resulting gradients remained deterministic conditional on the augmented latent trajectories, which improved numerical stability relative to noisy particle-gradient approaches.

5.4. Reproducibility

All numerical experiments were implemented in `R`. All computations were performed on an Intel Core i7-12700H CPU with 32 GB RAM, running Windows 11. To facilitate reproducibility, the source code and scripts used in this work are available at: https://github.com/Gabriel-Barragan/SDEMEM

6. Conclusions and Discussions

In this work, Markov Chain Monte Carlo (MCMC), Sequential Monte Carlo (SMC), and numerical discretization schemes were combined to perform Bayesian inference in stochastic differential equation mixed-effects models (SDEMEMs). For parameter estimation, Metropolis–Hastings (MH), MALA, and Hamiltonian Monte Carlo (HMC) methods were considered, while particle filtering techniques were used for latent state estimation. For the orange tree growth model, the IA, MALA, and HMC approaches were implemented together with the Euler–Maruyama (EM), modified diffusion bridge (MDB), and residual diffusion bridge (RDB) discretization schemes. The empirical results indicated that the MALA and HMC methods combined with the EM and MDB schemes generally achieved larger effective sample sizes (ESS), lower autocorrelation, and higher acceptance probabilities than the remaining approaches considered in this study. In particular, HMC-MDB provided a favorable empirical trade-off between sampling efficiency and computational cost for the examples analyzed here.
For the bivariate Ornstein–Uhlenbeck process, correlated and uncorrelated versions of MH-Gibbs, MALA-Gibbs, and HMC-Gibbs algorithms were investigated under the EM discretization scheme. For the fixed number of MCMC iterations considered in our experiments, the correlated methods did not necessarily reduce the per-iteration computational cost. Rather, their main advantage was improved sampling efficiency through variance reduction in successive likelihood-ratio estimates, which frequently increased ESS/sec values and improved chain mixing.
The numerical experiments also highlighted important trade-offs between proposal quality and computational complexity. Although the RDB scheme often improved acceptance probabilities by generating trajectories more consistent with future observations, it also introduced substantially higher computational cost due to the auxiliary deterministic trajectory and repeated residual corrections required at each discretization interval. In contrast, the MDB scheme provided competitive performance with considerably lower runtime, suggesting a more favorable balance between implementation complexity and computational efficiency for moderately nonlinear SDEMEMs.
Overall, the comparisons presented in this study should be interpreted as empirical evidence for the specific models, tuning choices, and computational settings considered here, rather than as universal rankings of the algorithms. Future work will focus on scalable particle MCMC methods, adaptive HMC strategies, and guided proposal mechanisms for high-dimensional diffusion processes [56, 54].

REFERENCES

[1] Christophe Andrieu, Arnaud Doucet, and Roman Holenstein. Particle markov chain monte carlo methods. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 72(3):269–342, 2010.

[2] Christophe Andrieu and Gareth O. Roberts. The pseudo-marginal approach for efficient monte carlo computations. *The Annals of Statistics*, 37(2):697–725, 2009.

[3] Julian Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(2):192–225, 1974.

[4] A. Beskos, O. Papaspiliopoulos, G. O. Roberts, and P. Fearnhead. Exact and computationally efficient likelihood-based estimation for discretely observed diffusion processes (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):333–382, 2006.

[5] Michael Betancourt. A conceptual introduction to hamiltonian monte carlo. *arXiv preprint arXiv:1701.02434*, 2017.

[6] Imke Botha, Robert Kohn, and Christopher Drovandi. Particle methods for stochastic differential equation mixed effects models. *Bayesian Analysis*, 16(2):575–609, 2021.

[7] Norman E Breslow and David G Clayton. Approximate inference in generalized linear mixed models. *Journal of the American statistical Association*, 88(421):9–25, 1993.

[8] H Chau, Justin Lars Kirkby, DH Nguyen, D Nguyen, N Nguyen, and T Nguyen. An efficient method to simulate diffusion bridges. *Statistics and Computing*, 34(4):131, 2024.

[9] Meng Chen, Sy-Miin Chow, Zita Oravecz, and Emilio Ferrer. Fitting bayesian stochastic differential equation models with mixed effects through a filtering approach. *Multivariate behavioral research*, 58(5):1014–1038, 2023.

[10] M. Davidian and D Giltinan. *Nonlinear models for repeated measurement data*. Chapman & Hall, London., 2001.

[11] Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential monte carlo samplers. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 68(3):411–436, 05 2006.

[12] Maud Delattre. A review on asymptotic inference in stochastic differential equations with mixed effects. *Japanese Journal of Statistics and Data Science*, 4:543–575, 2020.

[13] Maud Delattre and Marc Lavielle. Coupling the saem algorithm and the extended kalman filter for maximum likelihood estimation in mixed-effects diffusion models. *Statistics and Its Interface*, 6:519–532, 2013.

[14] George Deligiannidis, Arnaud Doucet, and Michael K Pitt. The correlated pseudomarginal method. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 80(5):839–870, 2018.

[15] Susanne Ditlevsen and Petr Lansky. Estimation of the input parameters in the feller neuronal model. *Physical Review E*, 73(6):061910, 2006.

[16] Sophie Donnet, Jean Louis Foulley, and Adeline Samson. Bayesian analysis of growth curves using mixed models defined by stochastic differential equations. *Biometrics*, 66(3):733–741, 2010.

[17] Sophie Donnet and Adeline Samson. A review on estimation of stochastic differential equations for pharmacokinetic/pharmacodynamic models. *Advanced Drug Delivery Reviews*, 65(7):929–939, 2013.

[18] Arnaud Doucet, Nando de Freitas, and Neil Gordon. *An Introduction to Sequential Monte Carlo Methods*, pages 3–14. Springer New York, New York, NY, 2001.

[19] Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid monte carlo. *Physics Letters B*, 195(2):216–222, 1987.

[20] Garland B Durham and A Ronald Gallant. Numerical techniques for maximum likelihood estimation of continuous-time diffusion processes. *Journal of Business & Economic Statistics*, 20(3):297–338, 2002.

[21] Bakrim Fadwa, Hamid El Maroufy, and Hassan Ait Mousse. Simulation and parametric inference of a mixed effects model with stochastic differential equations using the fokker-planck equation solution. In Dragan M. Cvetković and Gunvant A. Birajdar, editors, *Numerical Modeling and Computer Simulation*, chapter 6. IntechOpen, Rijeka, 2020.

[22] Benjamin Favetto and Adeline Samson. Parameter estimation for a bidimensional partially observed ornstein–uhlenbeck process with biological application. *Scandinavian Journal of Statistics*, 37(2):200–220, 2010.

[23] Andreas Fichtner and Andrea Zunino. Hamiltonian nullspace shuttles. *Geophysical research letters*, 46(2):644–651, 2019.

[24] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

[25] Andrew Golightly and Darren J Wilkinson. Bayesian inference for nonlinear multivariate diffusion models observed with error. *Computational Statistics & Data Analysis*, 52(3):1674–1693, 2008.

[26] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings F (Radar and Signal Processing)*, 140:107–113, 1993.

[27] Matthew M Graham, Alexandre H Thiery, and Alexandros Beskos. Manifold markov chain monte carlo methods for bayesian inference in diffusion models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(4):1229–1256, 2022.

[28] Henrik Häggström, Sebastian Persson, Marija Cvijovic, and Umberto Picchini. Simulation-based inference for stochastic nonlinear mixed-effects models with applications in systems biology. *Statistics and Computing*, 36(3):99, 2026.

[29] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 04 1970.

[30] Saba Infante, Aracelis Hernández, et al. Stochastic models to estimate population dynamics. *Statistics, Optimization & Information Computing*, 7(2):311–328, 2019.

[31] Muhammad Izzatullah, Tristan van Leeuwen, and Daniel Peter. Langevin dynamics markov chain monte carlo solution for seismic inversion. In *82nd EAGE Annual Conference & Exhibition*, volume 2021, pages 1–5. European Association of Geoscientists & Engineers, 2021.

[32] Mary J. Lindstrom and Douglas M. Bates. Nonlinear mixed effects models for repeated measures data. *Biometrics*, 46(3):673–687, 1990.

[33] David Luengo García, Luca Martino, Mónica Bugallo, Víctor Elvira Arregui, and Simo Särkkä. A survey of monte carlo methods for parameter estimation. *EURASIP Journal on Advances in Signal Processing*, (25), 2020.

[34] C. E. McCulloch and S. R Searle. *Generalized linear and mixed models*. Wiley, New York, 2001.

[35] N Metropolis, A Rosenbluth, M Rosenbluth, A Teller, and E Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[36] Radford M Neal et al. Mcmc using hamiltonian dynamics. *Handbook of Markov chain Monte Carlo*, 2(11):2, 2011.

[37] Christopher Nemeth and Paul Fearnhead. Stochastic gradient markov chain monte carlo. *Journal of the American Statistical Association*, 116(533):433–450, 2021.

[38] Bernt Oksendal. *Stochastic differential equations (3rd ed.): an introduction with applications*. Springer-Verlag, Berlin, Heidelberg, 1992.

[39] Rune V. Overgaard, Niclas Jonsson, Christoffer W. Tornøe, and Henrik Madsen. Non-linear mixed-effects models with stochastic differential equations: implementation of an estimation algorithm. *Journal of Pharmacokinetics and Pharmacodynamics*, 32(1):85–107, 2005.

[40] Richard Perez-Roa, Saba Infante, Gabriel Barragan, and Raul Manzanilla. Bayesian inference based on algorithms: Mh, hmc, mala and lip-mala for prestack seismic inversion. *Nonlinear Processes in Geophysics*, 33(2):173–195, 2026.

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

[42] Umberto Picchini, Susanne Ditlevsen, Andrea De Gaetano, and Petr Lansky. Parameters of the diffusion leaky integrate-and-fire neuronal model for a slowly fluctuating signal. *Neural computation*, 20(11):2696–2714, 2008.

[43] Umberto Picchini and Julie Lyng Forman. Bayesian inference for stochastic differential equation mixed effects models of a tumour xenography study. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 68(4):887–913, 2019.

[44] Umberto Picchini, Andrea Gaetano, and Susanne Ditlevsen. Stochastic differential mixed-effects models. *Scandinavian Journal of statistics*, 37(1):67–90, 2010.

[45] Natesh S. Pillai, Andrew M. Stuart, and Alexandre H. Thiéry. Optimal scaling and diffusion limits for the Langevin algorithm in high dimensions. *The Annals of Applied Probability*, 22(6):2320 – 2356, 2012.

[46] José Pinheiro and Douglas Bates. *Mixed-effects models in S and S-PLUS*. Springer Science & Business Media, 2006.

[47] Gareth O Roberts and Osnat Stramer. Langevin diffusions and metropolis-hastings algorithms. *Methodology and computing in applied probability*, 4:337–357, 2002.

[48] Tom Ryder, Andrew Golightly, A. Stephen McGough, and Dennis Prangle. Black-box variational inference for stochastic differential equations. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4423–4432. PMLR, 10–15 Jul 2018.

[49] Henry Scheffe. Alternative models for the analysis of variance. *The Annals of Mathematical Statistics*, 27(2):251–271, 1956.

[50] Chris Sherlock, Alexandre H. Thiery, Gareth O. Roberts, and Jeffrey S. Rosenthal. On the efficiency of pseudo-marginal random walk metropolis algorithms. *The Annals of Statistics*, 43(1):238–275, 2015.

[51] Jose Soto and Saba Infante. Ensemble kalman filter and extended kalman filter for state-parameter dual estimation in mixed effects models defined by a stochastic differential equation. In Andrea Basantes-Andrade, Miguel Naranjo-Toro, Marcelo Zambrano Vizuete, and Miguel Botto-Tobar, editors, *Technology, Sustainability and Educational Innovation (TSIE)*, pages 285–300, Cham, 2020. Springer International Publishing.

[52] José Soto, Saba Infante, Franklin Camacho, and Isidro R Amaro. Estimulación de un modelo de efectos mixtos usando un proceso de difusión parcialmente observado. *Revista de Matemática: Teoría y Aplicaciones*, 26(1):83–98, 2019.

[53] J Michael Steele. *Stochastic calculus and financial applications*. Springer Science & Business Media, 2012.

[54] Frank van der Meulen and Moritz Schauer. Bayesian estimation of discretely observed multi-dimensional diffusion processes using guided proposals. *Electronic Journal of Statistics*, 11(1):2358–2396, 2017.

[55] Gavin A Whitaker, Andrew Golightly, Richard J Boys, and Chris Sherlock. Bayesian inference for diffusion-driven mixed-effects models. *Bayesian Analysis*, 12(2):435–463, 2017.

[56] Gavin A Whitaker, Andrew Golightly, Richard J Boys, and Chris Sherlock. Improved bridge constructs for stochastic differential equations. *Statistics and Computing*, 27:885–900, 2017.

[57] Samuel Wiqvist, Andrew Golightly, Ashleigh T McLean, and Umberto Picchini. Efficient inference for stochastic differential equation mixed-effects models using correlated particle pseudo-marginal algorithms. *Computational Statistics & Data Analysis*, 157:107–151, 2021.