Journal de la Société Française de Statistique
Vol. 155 No. 1 (2014)

Using PMCMC in EM algorithm for stochastic mixed models: theoretical and practical issues

Titre: Utilisation de PMCMC dans l'algorithme EM pour des modeles mixtes stochastiques : enjeux théoriques et pratiques

Sophie Donnet[1] and Adeline Samson [2,3]

Abstract: Biological processes measured repeatedly among a series of individuals are standardly analyzed by mixed models. Recently, stochastic processes have been introduced to model the variability along time for each subject. Although the likelihood of these stochastic mixed models is intractable, various estimation methods have been proposed when the latent stochastic process is a discrete time finite state Markov chain. This is not the case when the hidden stochastic process is a continuous time process with non finite state space. This paper focuses on mixed models defined by parametric Stochastic Differential Equations (SDEs). We propose to use Particle MCMC algorithm for the maximum likelihood estimation of mixed SDE models, by combining it with SAEM algorithm. Theoretical and numerical convergence properties are discussed. Two simulated examples, an Ornstein-Uhlenbeck process and a time-inhomogeneous SDE with stochastic volatility, illustrate this estimator convergence, including the volatility parameter which is known to be hard to estimate.

Résumé : Des données longitudinales mesurées chez plusieurs individus d'un processus biologique sont classiquement analysées par modeles mixtes. Récemment, des version stochastiques de ces modeles ont été proposées pour tenir compte de la variabilité dans le temps. Meme si la vraisemblance de ces modeles mixtes stochastiques n'est pas explicite, de nombreuses méthodes d'estimation ont été proposées dans le cas où le processus stochastique est une chaine de Markov discrète à espace d'état fini. Quand le processus stochastique caché est un processus à temps continu sur un espace d'état non fini, il existe peu de références. Nous nous interessons à ce cadre. Nous proposons de combiner un algorithme MCMC particulaire à un algorithme SAEM pour calculer l'estimateur du maximum de vraisemblance du modele. Les propriétés théoriques et numériques de l'algorithme sont discutées. A partir de deux exemples, un processus d'Ornstein-Uhlenbeck et un processus non homogène en temps et à volatilité stochastique, nous illustrons la convergence de l'algorithme.

Keywords: Mixed models, Stochastic Differential Equations, SAEM algorithm, Particle Filter, PMCMC
Mots-clés : modeles mixtes, equations différentielles stochastiques, algorithme SAEM, filtre particulaire, PMCMC
AMS 2000 subject classifications: 35L05, 35L70

1. Introduction

Studies where a response is measured along time on a group of individuals arise in many fields of research (clinical trials, studies of growth in agriculture, etc). Such longitudinal data can be inferred using the following model: let y_{ij} be the j-th observation of unit i ($i = 1 \dots n$) at time t_{ij}

[1] UMR MIA 518 AgroParisTech/INRA, Paris, France
E-mail: sophie.donnet@agroparistech.fr
[2] Laboratoire MAP5, UMR CNRS 8145, Universite Paris Descartes, France
[3] Laboratoire Jean Kuntzmann, UMR CNRS 5224, Univ Grenoble-Alpes, France
E-mail: adeline.leclercq-samson@imag.fr

$(j = 0, \ldots, J_i)$, we assume that y_{ij} is the noisy observation of a latent time-dependent process X_{ij}:

$$y_{ij} = g(X_{ij}, \varepsilon_{ij}), \quad \varepsilon_{ij} \sim_{i.i.d.} \mathcal{N}(0, \sigma^2), \tag{1}$$

where g is the known error model function, and ε_{ij} is the measurement noise.
When X_{ij} is a deterministic process depending on time, one can write

$$X_{ij} = f(t_{ij}, \phi_i) \tag{2}$$

where f is a known function and $(\phi_i)_{i=1\ldots n}$ are random individual parameters, distributed with a density $p(\,\cdot\,; \beta)$ depending on the population parameter β:

$$\phi_i \sim_{i.i.d} p(\cdot|\beta). \tag{3}$$

The parameter vector is $\theta = (\beta, \sigma)$. Model (1-2-3) is a mixed-effects model (Pinheiro and Bates, 2000), which has proved its ability to discriminate between the inter-subjects variability (the variance of the individual parameters ϕ_i) and the residual variability (measurement noise, σ^2).

Stochastic mixed models Recently –in order to capture the complexity of the biological process of interest– mixed effects models have been extended to cases where the latent process X_{ij} is not deterministic but is the realization of a temporal stochastic process. A natural property of temporal stochastic process is the Markov property, and we focus on this class of stochastic processes. More precisely, for each subject i and each time t_{j-1}, t_j, we have:

$$q(X_{ij}|X_{ij-1}, X_{ij-2}, \ldots; \phi_i, \theta) = q(X_{ij}|X_{ij-1}; \phi_i, \theta) \quad a.s. \tag{4}$$

Two classes of mixed models with latent markovian stochastic processes have been studied in the literature.

• In its simplest version, $(X_{ij})_{j\geq 0}$ is a discrete-time Markov chain with finite state space:

$$\forall i = 1\ldots n, \quad X_{ij} \in \{1, \ldots, S\} \text{ with transition probabilities } P(X_{ij} = s'|X_{ij-1} = s; \phi_i, \theta). \tag{5}$$

The model defined by equations (1-5-3) results into an extension of what is usually called Hidden Markov Models to the longitudinal data setting (Altman, 2007; Maruotti, 2011; Delattre and Lavielle, 2012).

• The second class of stochastic mixed models is based on time-continuous stochastic processes with continuous state space. Then for each subject i, X_{ij} is the solution at discrete time t_{ij} $(X_{ij} = X_{it_{ij}})$ of a stochastic differential equation (SDE):

$$dX_{it} = a(X_{it}, t, \phi_i)dt + b(X_{it}, t, \phi_i, \gamma)dB_{it}, \quad X_{i0}|\phi_i \sim \pi_0(\cdot|\phi_i), \tag{6}$$

with $(B_{it})_{i=1\ldots n}$ a collection of independent Brownian motions, a and b the drift and volatility functions –which may depend on a new parameter γ– and $\pi_0(\cdot|\phi_i)$ the initial distribution. Note that SDEs have emerged as a powerful tool to model complex biological processes, when deterministic models (defined for instance by ordinary differential equations ODE) do not capture the exact process of interest, some individuals displaying for instance local "random" fluctuations (Overgaard et al., 2005; Tornøe et al., 2005; Picchini et al., 2010). Especially, they appeared in pharmacokinetics/pharmacodynamics (see Donnet and Samson, 2013, for a review) and in neurosciences (Picchini et al., 2008).

In this paper we focus on mixed effects models defined by SDE, summarized below, in model (7).

$$\left\{\begin{array}{lll} y_{ij} & = & g(X_{ij},\varepsilon_{ij}), \quad \varepsilon_{ij} \sim_{i.i.d.} \mathscr{N}(0,\sigma^2), \\ dX_{it} & = & a(X_{it},t,\phi_i)dt + b(X_{it},t,\phi_i,\gamma)dB_{it}, \quad X_{it_{i0}}|\phi_i \sim \pi_0(\cdot|\phi_i) \\ \phi_i & \sim_{i.i.d.} & p(\phi|\beta) \end{array}\right. \tag{7}$$

with the parameter of interest $\theta = (\beta,\gamma,\sigma)$. Mixed SDEs can be viewed as an extension of mixed Markov chain with finite state space, to continuous space and time. They cover a larger class of problems, especially with irregular observation times and different time scales. Even though of high interest because more general, these models are really difficult to study and to handle, and statistical inference becomes a challenging issue.

***Remark* 1.** *In the following, we use the notations:* $\mathbf{y}_i = (y_{ij})_{j=0\ldots,J_i}$, $\underline{\mathbf{y}} = (\mathbf{y}_i)_{i=1\ldots n}$, $\mathbf{X}_i = (X_{ij})_{j=0\ldots,J_i}$, $\underline{\mathbf{X}} = (\mathbf{X}_i)_{i=1\ldots n}$ *and* $\Phi = (\phi_i)_{i=1\ldots n}$.

Statistical inference Inference for mixed effects models with deterministic latent process $X_{ij} = f(t_{ij},\phi_i)$ has been widely studied in the literature. As soon as f is non-linear with respect to ϕ_i, the likelihood function

$$\mathscr{L}(\underline{\mathbf{y}};\theta) = \prod_{i=1}^{n}\int_{\phi_i} p(\mathbf{y}_i,\phi_i,\theta)d\phi_i = \prod_{i=1}^{n}\int_{\phi_i} p(\mathbf{y}_i|\phi_i;\theta)p(\phi_i|\theta)d\phi_i \tag{8}$$

has no closed form and advanced estimation strategies have to be used (Davidian and Giltinan, 1995; Wolfinger, 1993). Pinheiro and Bates (2000) propose methods relying on a linearisation of the likelihood (FOCE) whereas Kuhn and Lavielle (2005) develop a stochastic version of the Expectation-Maximisation algorithm (SAEM-MCMC).

The difficulty of the task is enhanced when the latent process is stochastic. Indeed, expression (8) remains valid but the quantity $p(\mathbf{y}_i|\phi_i,\theta)$ has itself an integral form

$$p(\mathbf{y}_i|\phi_i;\theta) = \int_{\mathbf{X}_i} p(\mathbf{y}_i|\mathbf{X}_i,\theta)p(\mathbf{X}_i|\phi_i,\theta)d\mathbf{X}_i, \tag{9}$$

Using the Markov property of the latent process, expression (9) can be decomposed into:

$$p(\mathbf{y}_i|\phi_i,\theta) = \int_{\mathbf{X}_i}\prod_{j=0}^{J_i} p(y_{ij}|X_{ij},\phi_i,\theta)p(X_{i0}|\phi_i)\prod_{j=0}^{J_i-1} p(X_{ij+1}|X_{ij},\phi_i;\theta)d\mathbf{X}_i. \tag{10}$$

• If the hidden Markov process evolves in a finite state space (model (1-5-3)), integral (10) is in fact a finite sum over the latent space

$$p(\mathbf{y}_i|\phi_i,\theta) = \sum_{X_{i0},\ldots,X_{iJ_i}\in\{1,\ldots,S\}^{J_i+1}}\prod_{j=0}^{J_i} p(y_{ij}|X_{ij}\phi_i,\theta)p(X_{i0}|\phi_i;\theta)\prod_{j=0}^{J_i-1} p(X_{ij+1}|X_{ij},\phi_i;\theta). \tag{11}$$

In that case, provided the use of a specific filtering methodology, this quantity could be computed explicitly and the problem reduces to a problem similar to the standard mixed effects models. For instance, Delattre and Lavielle (2012) combine the famous forward-backward Kalman filter to SAEM. See also Maruotti (2011) for a recent review on these models. Note that if the number of terms involved in the sum is very large, computational problems may also arrive in that case.

• If the Hidden Markov process is issued from a SDE (model (7)), integral (9) can not be computed explicitly and the difficulty of the problem is automatically enhanced. One strategy is to approximate the quantity $p(\mathbf{y}_i|\phi_i,\theta)$ using the extended Kalman Filter. Naturally, this approximation is then embedded in algorithms developed for standard mixed effects models. This strategy has been adopted by several authors: Overgaard et al. (2005) combine the extended Kalman Filter to the FOCE algorithm and Delattre and Lavielle (2013) include it in SAEM. However, even if numerically efficient, no theoretical convergence result can be proved for these two algorithms due to the intrinsic properties of the extended Kalman filter.

In this work, we aim at proposing a new estimation algorithm for mixed effects models defined by SDEs (model (7)) which is numerically efficient and whose convergence properties can be studied. Our approach combines the SAEM algorithm for which convergence has been widely discussed in theory and in practice, with new sophisticated Markov Chain Monte Carlo (MCMC) algorithm, which are especially adapted to Hidden Markov Models with infinite state space and unknown parameters. These algorithms proposed by Andrieu et al. (2010) (namely the PMCMCs) combine particle filtering with standard MCMC algorithms and have well established theoretical convergence properties. Originally designed for a Bayesian estimation of State Space Models, we propose to use PMCMC for the maximum likelihood estimation of mixed stochastic models by combining it with the maximum likelihood SAEM. Especially, different variants of PMCMC have been considered: the PMMH (Particle Marginal Metropolis-Hastings) and the PGibbs (Particle Gibbs sampler). Both versions have advantages and drawbacks, especially when combined with the SAEM algorithm. Both theoretical and practical aspects of the combination of SAEM with PMCMC are thus studied thereafter.

The paper is organized as follows. In Section 2, we specify basic assumptions and describe the stochastic EM algorithm. In section 2.3, we develop the different strategies of imputation of the latent variables, especially the main advantages of the Particle version of the MCMC. In Section 3, the theoretical convergence of the algorithm SAEM-PMCMC is studied. The numerical efficiency of the algorithm is illustrated on a large simulated study in Section 4. In Section 5 we conclude and discuss the advantages and limitations of our approach.

2. Maximum likelihood estimation method: a stochastic EM algorithm

2.1. Technical assumptions

Consider a mixed SDE model for n subjects, with dynamics described by model (7). We make the following assumptions:

(M0-1): Functions a and b in SDE (6) are assumed to be sufficiently regular (with linear growth) to ensure a unique strong solution of the SDE with random effects (Oksendal, 2007; Delattre et al., 2013).
(M0-2): Moreover, we assume that SDE (6) has an explicit transition density $p(x_t|t-s,s,x_s,\phi;\gamma)$.

***Remark* 2.** *(**M0-2**) is a strong assumption which is not automatically implied by assumption (**M0-1**). If (**M0-2**) is not fulfilled, the transition has to be approximated by a Euler-Maruyama scheme for instance, see Donnet and Samson (2008).*

As said in the introduction, we aim at maximizing the likelihood $\mathscr{L}(\underline{\mathbf{y}};\theta)$ –with respect to θ– which can be written using an integral:

$$\mathscr{L}(\underline{\mathbf{y}};\theta) = \int_{\Phi} p(\underline{\mathbf{y}},\Phi;\theta)d\Phi = \int_{\underline{\mathbf{X}},\Phi} p(\underline{\mathbf{y}},\underline{\mathbf{X}},\Phi;\theta)d\underline{\mathbf{X}}d\Phi.$$

Although $\mathscr{L}(\underline{\mathbf{y}};\theta)$ has no explicit expression, under assumption **(M0-2)** the so-called *complete likelihood* $p(\underline{\mathbf{y}},\underline{\mathbf{X}},\Phi;\theta)$ can be written in a closed form (which is the key point for using EM, as see hereafter). The complete likelihood can be decomposed into

$$p(\underline{\mathbf{y}},\underline{\mathbf{X}},\Phi;\theta) = p(\underline{\mathbf{y}}|\underline{\mathbf{X}},\Phi;\theta)p(\underline{\mathbf{X}}|\Phi;\theta)p(\Phi|\theta) = p(\underline{\mathbf{y}}|\underline{\mathbf{X}};\sigma)p(\underline{\mathbf{X}}|\Phi;\gamma)p(\Phi|\beta).$$

We restrict our study to exponential complete likelihood, i.e. we assume that:

(M1): g, a, b and the individual parameters distribution $p(\phi;\beta)$ are such that there exist functions ψ_1,ν_1 of σ, functions ψ_2,ν_2 of γ and functions ψ_3,ν_3 of β verifying

$$\begin{aligned}\log p(\underline{\mathbf{y}},\underline{\mathbf{X}},\Phi;\theta) &= -\psi_1(\sigma)-\psi_2(\gamma)-\psi_3(\beta)\\ &\quad +\big\langle S_1(\underline{\mathbf{y}},\underline{\mathbf{X}}),\nu_1(\sigma)\big\rangle+\langle S_2(\underline{\mathbf{X}},\Phi),\nu_2(\gamma)\rangle+\langle S_3(\Phi),\nu_3(\beta)\rangle\\ &= -\psi(\theta)+\big\langle S(\underline{\mathbf{y}},\underline{\mathbf{X}},\Phi),(\nu_1(\sigma),\nu_2(\gamma),\nu_3(\beta))\big\rangle\\ &= -\psi(\theta)+\big\langle S(\underline{\mathbf{y}},\underline{\mathbf{X}},\Phi),\nu(\theta)\big\rangle\end{aligned}$$

where $\psi(\theta)=\psi_1(\sigma)+\psi_2(\gamma)+\psi_3(\beta)$, $S(\underline{\mathbf{y}},\underline{\mathbf{X}},\Phi)=(S_1(\underline{\mathbf{y}},\underline{\mathbf{X}}),S_2(\underline{\mathbf{X}},\Phi),S_3(\Phi))$ is a minimal sufficient statistic of the complete model, taking its value in a subset $\mathscr{S}$ and $\langle\cdot,\cdot\rangle$ is the scalar product.

***Remark* 3.** *Assumption **(M1)** is imposed for convergence reasons. However, it is not a restrictive assumption since it is verified by standard choices for g and $p(\phi;\beta)$. For example, g can be $g(x,\varepsilon)=x+\varepsilon$, resulting into the additive error model, or $g(x,\varepsilon)=x(1+\varepsilon)$, $g(x,\varepsilon)=h(x)\exp(\varepsilon)$ with h known, resulting into a stochastic volatility model. The typical Gaussian or log Gaussian distributions on the individual parameters also fulfill condition **(M1)**.*

2.2. *Stochastic version of the EM algorithm for mixed SDE models*

We now describe the stochastic version of the Expectation-Maximization (EM) algorithm used to maximize the likelihood. The EM algorithm (Dempster et al., 1977) is useful in situations where the observations $\underline{\mathbf{y}}$ can be enhanced by latent variables and when the direct maximization of the marginal likelihood $\mathscr{L}(\underline{\mathbf{y}};\theta)$ is more complex than the maximization of the conditional expectation of the complete likelihood.

The EM algorithm is an iterative procedure: at the m-th iteration, given the current parameter value $\widehat{\theta}_{m-1}$, the E-step is the evaluation of $Q_m(\theta)=Q(\theta\,|\,\widehat{\theta}_{m-1})=E\left[\log p(\underline{\mathbf{y}},\underline{\mathbf{X}},\Phi;\theta)|\underline{\mathbf{y}};\widehat{\theta}_{m-1}\right]$ while the M-step updates $\widehat{\theta}_{m-1}$ by maximizing $Q_m(\theta)$. Under assumption **(M1)**, the E-step reduces to the computation of $E\left[S(\underline{\mathbf{y}},\underline{\mathbf{X}},\Phi)|\underline{\mathbf{y}};\widehat{\theta}_{m-1}\right]$. When this expectation has no closed form, Delyon et al. (1999) propose the Stochastic Approximation EM algorithm (SAEM) and replace the E-step by a stochastic approximation of $Q_m(\theta)$. The E-step is then divided into a simulation step (S-step) of the non-observed data $(\underline{\mathbf{X}}^{(m)},\Phi^{(m)})$ under the conditional distribution $p(\underline{\mathbf{X}},\Phi\,|\underline{\mathbf{y}};\widehat{\theta}_{m-1})$ and a stochastic approximation step (SA-step) of $E\left[S(\underline{\mathbf{y}},\underline{\mathbf{X}},\Phi)|\underline{\mathbf{y}};\widehat{\theta}_{m-1}\right]$ with a sequence $(\alpha_m)_{m\in\mathbb{N}}$ of positive numbers decreasing to zero:

$$s_m = s_{m-1} + \alpha_m\left[S(\underline{\mathbf{y}},\underline{\mathbf{X}}^{(m)},\Phi^{(m)}) - s_{m-1}\right]$$

where s_m stands for the approximation of this expectation.

However, in many models, simulating exactly the latent quantities $(\underline{\mathbf{X}},\Phi)$ given the observations $\underline{\mathbf{y}}$ is impossible. In such cases, Kuhn and Lavielle (2005) prove that the exact simulation can be replaced by the use of a MCMC algorithm whose stationary distribution is the distribution of interest $p(\underline{\mathbf{X}},\Phi\,|\underline{\mathbf{y}};\widehat{\theta}_{m-1})$. The convergence of this SAEM-MCMC algorithm towards a maximum of the likelihood is proved by Kuhn and Lavielle (2005). Obviously, the practical and theoretical convergences of such an algorithm depend on the convergence properties of the MCMC algorithm used to simulate the latent quantities.

In model (7), the natural set of latent data is $(\underline{\mathbf{X}},\Phi)$. However, $(\underline{\mathbf{X}},\Phi)$ involves a huge amount of latent data set, which can lead to bad convergence properties of the MCMC algorithm. Therefore solutions suggested in the literature tend to reduce the space of simulation to Φ, integrating out the latent process $\underline{\mathbf{X}}$. This reduction of the latent dataset is for instance natural in cases where γ and σ are known parameters or considered as individual random parameters (included in the vector ϕ_i). Indeed, in this case, θ reduces to β and the maximisation step of the EM algorithm can be written as (using the notations of assumption **(M1)**):

$$\widehat{\beta}_m = \arg\max_{\theta}\left(-\psi_3(\beta) + \langle s_m, \nu_3(\beta)\rangle\right) \quad \text{where} \quad s_m = s_{m-1} + \alpha_m\left[S_3(\Phi^{(m)}) - s_{m-1}\right]$$

and only requires simulations of Φ.
However, even in the previous case, we highlight in Section 2.3.1 that integrating out $\underline{\mathbf{X}}$ is not an easy task and that the use of standard methods (such as extended Kalman filter) can lead to algorithms without established theoretical convergence properties. In Section 2.3.2, we describe the PMCMC algorithms which allow to tackle this point and guaranty a practical and theoretical convergence result of the estimation method.

2.3. Simulation of the latent variables

As underlined before, simulating the natural set of latent variable $p(\underline{\mathbf{X}},\Phi|\underline{\mathbf{y}};\theta)$ is a tough algorithmic task. In the following, we present a MCMC algorithm on Φ only ($\underline{\mathbf{X}}$ being integrated out). We prove that it supplies a satisfactory answer for mixed SDE only in a few cases such as in discrete time finite state Markov chain (5) or in the case of a linear SDE with Gaussian additive observation noise. In any other cases, we show that approximations are required which can lead to a lack of theoretical properties. Then, we present the alternative simulating both $(\underline{\mathbf{X}}, \Phi)$.

Let us remark that the individual trajectories are independent, and then sampling $p(\Phi\,|\underline{\mathbf{y}};\theta)$ (or $p(\underline{\mathbf{X}}, \Phi\,|\underline{\mathbf{y}};\theta)$, respectively) by a MCMC is equivalent to using n independent MCMC, each of them sampling $p(\phi_i\,|\mathbf{y}_i;\theta)$ (or $p(\mathbf{X}_i, \phi_i\,|\mathbf{y}_i;\theta)$, respectively). As a consequence, for the ease of reading, *we omit the index i in the following and consider the simulation of* $p(\mathbf{X}, \phi|\mathbf{y})$.

2.3.1. Simulation of the latent variables ϕ given $\mathbf{y}$

Simulation under the marginalized conditional distribution $p(\phi|\mathbf{y};\theta)$ has the great advantage to reduce the dimension of the latent variables set, compared to the second strategy of latent variables $\mathbf{X}, \phi$. This could be perceived as an important advantage. However, the conditional simulation under $p(\Phi|\mathbf{y};\theta)$ is not direct because of the non-linearity of the model, and we expose thereafter that standard MCMC could lead to biased strategy. We detail the case of Metropolis-Hastings (MH) algorithm, but the same comments hold for a Gibbs sampler.

A MH algorithm would consist in simulating a new candidate ϕ^c with a proposal distribution $q(\phi^c|\phi;\theta)$ and accepting the candidate with probability:

$$\rho(\phi^c|\phi) = \min\left\{1, \frac{q(\phi|\phi^c)}{q(\phi^c|\phi)}\frac{p(\mathbf{y}|\phi^c;\gamma,\sigma)p(\phi^c;\beta)}{p(\mathbf{y}|\phi;\gamma,\sigma)p(\phi;\beta)}\right\} \tag{12}$$

Therefore one has to compute analytically $p(\mathbf{y}|\phi^c;\gamma,\sigma)$.

If the Hidden Markov model has a finite state space, exact filters have been proposed. Similarly, if the SDE is linear and the error model additive, the exact Kalman filter provides an exact computation of $p(\mathbf{y}|\phi^c;\gamma,\sigma)$. This leads to exact MCMC, with good theoretical convergence properties.

But as soon as the SDE is non-linear in X or the error model is not additive, the exact Kalman filter can not be applied. Delattre and Lavielle (2013) consider the extended Kalman filter (EKF) to approximate $p(\mathbf{y}|\phi^c;\gamma,\sigma)$ within a SAEM-MCMC algorithm. EKF is very computationally efficient (no simulation, only Gaussian approximation calculus) but without any theoretical convergence properties. Especially, it is known that if the SDE is highly non-linear, EKF provides an estimation of $p(\mathbf{y}|\phi;\gamma,\sigma)$ with error. Then the stationary distribution of the corresponding MCMC could not be the expected one, as well as the coupled SAEM could not converge towards the maximum likelihood estimator.

***Remark* 4.** *In case where γ and σ are known or considered as individual parameters, we already mention that the M-Step of the SAEM algorithm relies only on the distribution $p(\phi;\beta)$ and that only simulations of the individual parameters Φ are needed. But as explained above, no direct strategy of simulating only the Φ leads to a MCMC with theoretical convergence. Therefore, even in that case, this strategy should not be applied. See Remark 5 for the solution we propose in that case.*

2.3.2. *Simulation of the latent variables* $(\mathbf{X}, \phi)$ *given* $\mathbf{y}$

As explained above, the first strategy of simulating only the Φ leads to approximated approaches, when the acceptance probability of the MH algorithm is approximated by EKF. If one wants to keep the same idea than those presented in Section 2.3.1, namely use an approximation to compute the probability acceptance, the only convergence result on that topic has been proved by Andrieu and Roberts (2009) when $p(\mathbf{y}|\phi^c;\gamma,\sigma)$ is approximated by an *unbiased estimator*. This argument is exploited by Andrieu et al. (2010) to propose a convergent algorithm, based on particle filtering technics, as described below. As particle filters naturally simulate the hidden trajectory $\mathbf{X}$, PMCMC enters the context of simulating the complete latent variables $\mathbf{X},\phi|\mathbf{y}$. Again, even if this problem seems more difficult given the greater dimension of the hidden state spaces, the family of particles MCMC is the only known technic at that time to provide an unbiased MCMC exploiting the Markovian structure of the hidden variables.

Two algorithms proposed by Andrieu et al. (2010) and targeting $p(\mathbf{X},\phi|\mathbf{y},\theta)$ are presented here, namely the Particle Marginal Metropolis Hastings (PMMH) and the Particle Gibbs sampler.

An Ideal Metropolis-Hastings algorithm. Assume that one wants to design a Metropolis-Hastings algorithm whose stationary distribution is $p(\mathbf{X},\phi|\mathbf{y};\theta)$. Then at iteration $\ell+1$, we would propose a candidate $(\mathbf{X}^c,\phi^c)$ with a proposal distribution $q(\mathbf{X}^c,\phi^c|\mathbf{X}^{(\ell)},\phi^{(\ell)})$ and accept it with probability

$$\rho(\mathbf{X}^c,\phi^c|\mathbf{X}^{(\ell)},\phi^{(\ell)}) = \min\left\{1, \frac{q(\mathbf{X}^{(\ell)},\phi^{(\ell)}|\mathbf{X}^c,\phi^c)}{q(\mathbf{X}^c,\phi^c|\mathbf{X}^{(\ell)},\phi^{(\ell)})}\frac{p(\mathbf{y}|\mathbf{X}^c;\sigma)p(\mathbf{X}^c|\phi^c;\gamma)p(\phi^c;\beta)}{p(\mathbf{y}|\mathbf{X}^{(\ell)};\sigma)p(\mathbf{X}^{(\ell)}|\phi^{(\ell)};\gamma)p(\phi^c;\beta)}\right\}$$

Assume that we can use the following ideal proposal distribution:

$$q(\mathbf{X}^c,\phi^c|\mathbf{X}^{(\ell)},\phi^{(\ell)}) = q(\phi^c|\phi^{(\ell)})p(\mathbf{X}^c|\mathbf{y},\phi^c;\theta)$$

then the acceptance probability would simplify into:

$$\rho(\mathbf{X}^c,\phi^c|\mathbf{X}^{(\ell)},\phi^{(\ell)}) = \min\left\{1, \frac{q(\phi^{(\ell)}|\phi^c)}{q(\phi^c|\phi)}\frac{p(\mathbf{y}|\phi^c;\gamma,\sigma)p(\phi^c;\beta)}{p(\mathbf{y}|\phi^{(\ell)};\gamma,\sigma)p(\phi;\beta)}\right\} \tag{13}$$

Note that the ideal acceptance probability (13) is exactly equal to (12), the one obtained if the Metropolis-Hastings algorithm is only performed on the individual parameters ϕ – instead of $(\mathbf{X},\phi)$.

However, two difficulties arise here. First, the conditional distribution $p(\mathbf{X}|\mathbf{y},\phi;\theta)$ is the filtering distribution. Secondly, the acceptance probability requires the computation of the marginal quantity $p(\mathbf{y}|\phi;\gamma,\sigma)$. In general these two quantities are unknown.

The PMMH algorithm. The PMMH algorithm is nothing but the previously presented ideal Metropolis-Hastings algorithm where the two problematic tasks are “solved" by the use of a Particle filtering sampler, also called Sequential Monte Carlo (SMC) algorithm. We first quickly recall the principle of the SMC algorithm and then write the pseudo-code of the PMMH algorithm.

The SMC produces a set of K particles $(\mathbf{X}^{(k)})_{k=1\dots K}$ and respective weights $(W_J^{(k)})_{k=1\dots K}$ approximating the conditional distribution $p(\mathbf{X}|\mathbf{y},\phi;\,\gamma,\sigma)$ by an empirical measure $\Psi_J^K=\sum_{k=1}^K W_J^{(k)}\mathbb{1}_{\mathbf{X}^{(k)}}$. We introduce the following notations: $\forall j=1\dots J$, $y_{0:j}=(y_0,\dots y_j)$ and $X_{0:j}=(X_0,\dots X_j)$. The SMC works as follows:

***Algorithm* 1 (SMC algorithm).**

- At time $j=0$
  - *sample* $X_0^{(k)}\sim\pi_0(\cdot|\phi)\ \forall k=1,\dots,K$ *and*
  - *Compute and normalize the weights:*

$$w_0^{(k)}=p(y_0,X_0^{(k)}|\phi;\gamma,\sigma),\quad W_0(k)=\frac{w_0^{(k)}}{\sum_{k=1}^K w_0^{(k)}}$$

- Time $j=1,\dots,J$:
  - *Sample K iid variables* $X_{0:j-1}^{'(k)}$ *according to the distribution* $\Psi_{j-1}^K=\sum_{k=1}^K W_{j-1}^{(k)}\mathbb{1}_{X_{0:j-1}^{(k)}}$.
  - *Then, for each particle* $k=1,\dots,K$, *sample* $X_j^{(k)}\sim q_{SMC}(\cdot|y_j,X_{0:j-1}^{'(k)},\phi;\gamma,\sigma)$ *(i.e. propagate the particle) and set* $X_{0:j}^{(k)}=(X_{0:j-1}^{'(k)},X_j^{(k)})$.
  - *Finally compute and normalize the weights:*

$$w_j^{(k)}=\frac{p(y_j|X_j^{(k)};\sigma)p(X_j^{(k)}|X_{j-1}^{'(k)},\phi;\gamma)}{q_{SMC}\left(X_j^{(k)}|y_j,X_{0:j-1}^{'(k)},\phi;\gamma,\sigma\right)},\quad W_j^{(k)}=\frac{w_j^{(k)}}{\sum_{k=1}^K w_j^{(k)}}$$

The simulation of one trajectory $\mathbf{X}=(X_1,\dots,X_J)$ (called a "run of SMC algorithm") under the approximation of $p(\mathbf{X}|\mathbf{y},\phi;\gamma,\sigma)$ is directly achieved by randomly choosing one particle $\mathbf{X}^{(k)}$ among the K particles with weights $(W_J^{(k)})_{k=1\dots K}$. Besides, the marginal distribution $p(\mathbf{y}|\phi;\gamma,\sigma)$ can be estimated through the weights

$$\widehat{p}^K(\mathbf{y}|\phi;\gamma,\sigma)=\prod_{j=0}^J\left(\frac{1}{K}\sum_{k=1}^K w_j^{(k)}\right). \tag{14}$$

Then, the Particle marginal Metropolis-Hastings sampler (Andrieu et al., 2010) is defined as:

***Algorithm* 2 (PMMH algorithm).**

- Initialization*: starting from* $\phi^{(0)}$, *generate* $\mathbf{X}^{(0)}$ *by a run of SMC algorithm –with K particles– targeting* $p(\mathbf{X}|\mathbf{y},\phi^{(0)};\gamma,\sigma)$ *and estimate* $p(\mathbf{y}|\phi^{(0)};\gamma,\sigma)$ *by* $\widehat{p}^K(\mathbf{y}|\phi^{(0)};\gamma,\sigma)$
- At iteration $\ell+1\geq 1$

  1. (a) *Sample a candidate* $\phi^c\sim q(\cdot|\phi^{(\ell)})$

     (b) *By a run of SMC algorithm with K particles, generate* $\mathbf{X}^c$ *targeting* $p(\cdot|\mathbf{y},\phi^c;\gamma,\sigma)$ *and compute* $\widehat{p}^K(\mathbf{y}|\phi^c;\gamma,\sigma)$ *estimating* $p(\mathbf{y}|\phi^c;\gamma,\sigma)$
  2. *Set* $\left(\mathbf{X}^{(\ell+1)},\phi^{(\ell+1)}\right)=(\mathbf{X}^c,\phi^c)$ *and* $\widehat{p}^K(\mathbf{y}|\phi^{(\ell+1)};\gamma,\sigma)=\widehat{p}^K(\mathbf{y}|\phi^c;\gamma,\sigma)$ *with probability*

$$\widehat{\rho}^K(X_{0:J}^c,\phi^c|X_{0:J}^{(\ell)},\phi^{(\ell)})=\min\left\{1,\frac{q(\phi^{(\ell)}|\phi^c)}{q(\phi^c|\phi^{(\ell)})}\frac{\widehat{p}^K(\mathbf{y}|\phi^c;\gamma,\sigma)p(\phi^c;\beta)}{\widehat{p}^K(\mathbf{y}|\phi^{(\ell)};\gamma,\sigma)p(\phi^{(\ell)};\beta)}\right\}$$

If the candidate is not accepted, then set $\left(\mathbf{X}^{(\ell+1)}, \phi^{(\ell+1)}\right) = \left(\mathbf{X}^{(\ell)}, \phi^{(\ell)}\right)$ *and* $\widehat{p}^K(\mathbf{y}|\phi^{(\ell+1)};\gamma,\sigma) = \widehat{p}^K(\mathbf{y}|\phi^{(\ell)};\gamma,\sigma)$

Note that, as announced, Step 1.b corresponds to an approximation of the ideal filtering distribution $p(\mathbf{X}^c|\mathbf{y},\phi^c;\theta)$ by the particle filter SMC whereas, in Step 2, the acceptance probability has been approximated using the by-product of the SMC, i.e. the estimator of $p(\mathbf{y}|\phi;\gamma,\sigma)$.

Under weak assumptions, PMMH admits the exact conditional distribution $p(X,\phi|y_{0:J};\theta)$ as invariant distribution for any number of particles K. This is a remarkable property that is very convenient when coupling to a maximum likelihood estimation method. Details are given in Section 3.
The practical convergence of this algorithm is conditioned by the proposal distributions $q(\phi^c|\phi^{(\ell)})$ and $q_{SMC}(X_j|y_j,X_{j-1},\phi^c;\gamma,\sigma)$ used in the SMC algorithm, which plays a crucial role to ensure good mixing properties. They are discussed with more details for the two simulated examples (Section 4).

***Remark* 5.** *As already said in Remark 4, when* γ *and* σ *are known or considered as individual parameters, the M-Step relies only on* $p(\phi;\beta)$*. Let us emphasize that this PMMH can be used to perform this simulation task and the corresponding maximization step. The main point is that this strategy has a strong theoretical convergence, contrary to the solution proposed by Delattre and Lavielle (2013) which is only based on the simulation of the* Φ*.*

The Particle Gibbs sampler. The PMMH presented before updates simultaneously ϕ and $\mathbf{X}$. A popular alternative to MH algorithm consists of using Gibbs sampler, to update sequentially $\mathbf{X}$ and ϕ, especially when the dimension of the latent variables is large (which is the case for the SDE mixed models). By alternately simulating $\mathbf{X}$ under the distribution $p(\mathbf{X}|\mathbf{y},\phi;\gamma,\sigma)$ and ϕ under the distribution $p(\phi|\mathbf{y},\mathbf{X};\gamma,\sigma)$, one expects to increase the acceptance rate of the MCMC scheme and its practical convergence.

If the conditional distributions $p(\mathbf{X}|\mathbf{y},\phi;\gamma,\sigma)$ and $p(\phi|\mathbf{y},\mathbf{X};\beta,\gamma,\sigma)$ cannot be simulated easily, we can theoretically resort to Metropolis Hastings algorithms for each component of ϕ and each time component of $\mathbf{X}$. This was proposed in (Donnet and Samson, 2008). However, such standard MCMC algorithms have reached their limits in high dimensional context: by not exploiting the Markovian structure of the latent trajectories $\mathbf{X}$, they can prove slow mixing properties.

An alternative is a Gibbs algorithm, where the latent trajectory is not simulated exactly by using the particle filter SMC. This is exactly the aim of the Particle Gibbs (PGibbs) proposed by Andrieu et al. (2010). The invariance of $p(\mathbf{X},\phi|\mathbf{y};\gamma,\sigma)$ as the exact target distribution is ensured provided the use of a conditional SMC. Conditional SMC comes after a first run of a standard SMC. This run provides a particle $X_{0:J}$; its corresponding ancestral lineage B is also stored. Then conditionally to this fixed particle and lineage, conditional SMC simulates all the remaining particles (see details in Andrieu et al., 2010). Then Particle Gibbs sampler is defined as:

***Algorithm* 3 (Particle Gibbs Sampler).**
• Initialization*: set randomly* $\phi^{(0)}$*, generate* $\mathbf{X}^{(0)}$ *by a run of SMC algorithm targeting* $p(\mathbf{X}|\mathbf{y},\phi^{(0)};\gamma,\sigma)$ *and store its ancestral lineage* $B^{(0)}$.
• At iteration $\ell = 1,\ldots,N$

1. *Sample* $\phi^{(\ell)} \sim p(\phi|\mathbf{X}^{(\ell-1)}, \mathbf{y})$
2. *Run a conditional SMC algorithm, consistent with* $\phi(\ell)$*, and the fixed particle* $\mathbf{X}(\ell-1), B(\ell-1)$*, sample* $\mathbf{X}(\ell)$ *from this conditional SMC and denote* $B(\ell)$ *its ancestral lineage.*

Under weak assumptions, the Particle Gibbs sampler (PGibbs) admits the exact conditional distribution $p(\mathbf{X}, \phi|\mathbf{y}; \theta)$ as invariant distribution for any number of particles K.

2.4. SAEM-PMCMC estimation algorithm

Finally, we propose the following algorithm of maximum likelihood estimation:

***Algorithm* 4** (SAEM-PMCMC algorithm)**.**

- Iteration 0: *initialization of* $\widehat{\theta}_0$ *and* $s_0 = E\left[S(\underline{\mathbf{y}}, \underline{\mathbf{X}}, \Phi)|\underline{\mathbf{y}}; \widehat{\theta}_0\right]$.
- At iteration $m \geq 1$:

  S-Step:
  - *For each individual* $i = 1 \ldots n$*, run* N *iterations of a PMCMC algorithm, targeting* $p(\mathbf{X}_i, \phi_i|\mathbf{y}_i, \widehat{\theta}_{m-1})$
  - *Set* $\underline{\mathbf{X}}^{(m)} = (\mathbf{X}_1^{(m)}, \ldots, \mathbf{X}_n^{(m)})$ *and* $\Phi^{(m)} = (\phi_1^{(m)}, \ldots, \phi_n^{(m)})$ *the simulated latent variables*

  SA-Step: *update of* s_{m-1} *using the stochastic approximation scheme:*

$$s_m = s_{m-1} + \alpha_m \left[S(\underline{\mathbf{y}}, \underline{\mathbf{X}}^{(m)}, \Phi^{(m)}) - s_{m-1}\right]. \qquad (15)$$

  M-Step: *update of* $\widehat{\theta}_{m-1}$ *by* $\widehat{\theta}_m = \arg\max_{\theta} \left(-\psi(\theta) + \langle s_m, \nu(\theta)\rangle\right)$.

3. Theoretical and practical convergence of the SAEM-PMCMC

In this section, we study the convergence of the SAEM-PMCMC. This convergence depends on regularity assumptions of the model and on the convergence of the PMCMC algorithm. Thus we first recall the convergence properties, both theoretical and numerical, of the different versions of PMCMC. Then we prove a convergence result for the SAEM-PMCMC algorithm, and discuss the assumptions on which it relies.

3.1. Convergence of the PMCMC algorithms

As already said, the most remarkable property of both the PMMH and PGibbs algorithms is that the distribution of interest $p(\mathbf{X}, \phi|\mathbf{y}; \gamma, \sigma)$ is left invariant by the transition kernel, *whatever the number of particles K is*, the ergodicity being reached under weak assumptions. The proof of these results relies on the introduction of an extended space, taking into account all the auxiliary variables generated by the SMC algorithm (set of generated trajectories, the resampling indices...).

In the following, we need a kind of geometric ergodicity of the chain to prove the convergence of the SAEM-PMCMC algorithm (see assumption **(SAEM5')**). Evidently, if the state space is finite, the geometric ergodicity will be ensured for both PMMH and PGibbs, as soon as the chain is irreducible and aperiodic. But this is not true when the state space is not finite. This is again a major difference between finite state space stochastic models (5) and SDE mixed models (7). Under some "wrong" choices of proposal distributions, it can even be shown that some PMCMC are not geometrically ergodic (Andrieu and Roberts, 2009; Pitt et al., 2012). Anyway, Andrieu and Roberts (2009) give conditions on the proposal which ensure the uniform geometric ergodicity of the PMMH. More precisely, they prove that if the noise density $p(y|X;\sigma)$ is bounded above

$$\sup_{y,X} p(y|X;\sigma) < M_\sigma, \tag{16}$$

PMMH inherits the convergence properties of the corresponding ideal MCMC algorithm. For instance, if $q(\phi^c|\phi) = p(\phi^c;\beta)$ then the kernel of the ideal MCMC $q(X^c_{0:J},\phi^c) = p(\phi^c;\beta)p(X^c_{0:J}|y_{0:J},\phi^c;\gamma,\sigma)$ is independent. For this kernel, the ratio $\frac{p(X_{0:J},\phi|y_{0:J})}{q(X_{0:J},\phi)}$ is bounded if (16) holds, ensuring the uniform ergodicity (Tierney, 1994). This is not so simple for the Particle Gibbs algorithm. It has not been shown that the standard PGibbs that is presented above is geometrically ergodic. But Chopin and Singh (2013) propose a version of Particle Gibbs with uniform geometric ergodicity.

From a numerical point of view, when the dimension of the hidden state space is large, which is obviously the case for model (7), it is well known that Gibbs samplers have better mixing properties and larger acceptance rates. We implement both within the SAEM algorithm.

3.2. *Convergence of the SAEM-PMCMC algorithms*

As the PMCMC algorithm can be viewed as a standard MCMC algorithm, the convergence of SAEM-PMCMC can be proved using Kuhn and Lavielle (2005) result. We recall the assumptions of Kuhn and Lavielle (2005), with two weakened assumptions on the compacity of the Markov chain and on the uniform geometric ergodicity.

(M2) The functions $\psi(\theta)$ and $\nu(\theta)$ are twice continuously differentiable on Θ.

(M3) The function $\bar{s}:\Theta \longrightarrow \mathscr{S}$ defined as $\bar{s}(\theta) = \int S(\underline{\mathbf{y}},\underline{\mathbf{X}},\Phi)p(\underline{\mathbf{X}},\Phi|\underline{\mathbf{y}};\theta)d\underline{\mathbf{X}}d\Phi$ is continuously differentiable on Θ.

(M4) The function $\ell(\theta) = \log p(\mathbf{y},\theta)$ is continuously differentiable on Θ and

$$\partial_\theta \int p(\mathbf{y},\underline{\mathbf{X}},\Phi;\theta)d\underline{\mathbf{X}}d\Phi = \int \partial_\theta p(\underline{\mathbf{y}},\underline{\mathbf{X}},\Phi;\theta)d\underline{\mathbf{X}}d\Phi.$$

(M5) Define $L:\mathscr{S}\times\Theta \longrightarrow \mathbb{R}$ as $L(s,\theta) = -\psi(\theta)+\langle s,\nu(\theta)\rangle$. There exists a function $\hat{\theta}:\mathscr{S}\longrightarrow\Theta$ such that

$$\forall\theta\in\Theta,\quad \forall s\in\mathscr{S},\quad L(s,\hat{\theta}(s)) \geq L(s,\theta).$$

(SAEM1) The positive decreasing sequence of the stochastic approximation $(\alpha_m)_{m\geq 0}$ is such that $\sum_m \alpha_m = \infty$ and $\sum_m \alpha_m^2 < \infty$.

(SAEM2) $\ell:\Theta\to\mathbb{R}$ and $\hat{\theta}:\mathscr{S}\to\Theta$ are d times differentiable, where d is the dimension of $S(\underline{\mathbf{y}},\underline{\mathbf{X}},\Phi)$.

(SAEM3) For all $\theta\in\Theta$, $\int ||S(\underline{\mathbf{y}},\underline{\mathbf{X}},\Phi)||^2 p(\underline{\mathbf{X}},\Phi|\underline{\mathbf{y}};\theta)d\underline{\mathbf{X}}d\Phi < \infty$ and the function $\Gamma(\theta) = Cov_\theta(S(\underline{\mathbf{X}},\Phi))$ is continuous.

(SAEM4) S is a bounded function.

(SAEM5') Let us denote Π_θ the transition kernel of the PMCMC algorithm and $\pi(\underline{\mathbf{X}},\Phi) = p(\underline{\mathbf{X}},\Phi|\underline{\mathbf{y}};\theta)$ its stationary distribution. We assume that Π_θ is Lipschitz in θ and generates a ergodic chain such that for any starting point (x,φ)

$$\sum_{m\geq 0} \|\mathbb{1}_{x,\varphi}\Pi_\theta^m - \pi\|_{TV} < \infty$$

where $\|\cdot\|_{TV}$ is the total variation of probability measures. This property is also called ergodicity of degree 2.

Then, we are able to provide the convergence of the SAEM-PMCMC algorithm. Note that despite the SMC approximation in the PMCMC algorithm, the fact that the marginal stationary distribution of PMCMC is the exact conditional distribution $p(X,\Phi|\underline{\mathbf{y}};\theta)$ is sufficient to prove the convergence of SAEM-PMCMC to the exact maximum likelihood. This is a powerful result when the diffusion is not directly observed. The convergence result is stated as follows:

Theorem 1. *Assume that (M1-5), (SAEM1-SAEM5') hold and that Θ is an open set. The sequence $\widehat{\theta}_m$ supplied by the SAEM-PMCMC algorithm converges a.s. towards a (local) maximum of the log-likelihood $\ell(\theta) = \log p(\underline{\mathbf{y}};\theta)$.*

The proof is given in Appendix.

Now, we comment the different assumptions for the SAEM-PMCMC algorithm. Assumptions **(M1-SAEM4)** refer to the regularity of the model. Assumptions **(M1-M5)**, **(SAEM1-SAEM3)** are standard and not restrictive. Assumption **(SAEM4)** is restrictive and not really realistic. It could be relaxed using a principle of random boundaries presented in Allassonnière et al. (2010).

Assumption **(SAEM5')** refers to the Markov Chain generated by the PMCMC algorithm. It requires two conditions. First, the Lipschitz condition on the Markov kernel Π_θ. This property holds if the complete likelihood is continuously derivable which is the case under (M2-M3) and if θ remains in a compact set. Second, the ergodicity of order 2 of the chain. It is obviously implied if geometric ergodicity of the chain can be ensured. But we emphasize that it is a weaker condition that is needed here. Geometric ergodicity has been proved for some of the PMCMC algorithms, as discussed in Section 3.1. In the following, we implement SAEM with PMMH, for which geometric ergodicity has been proved under some conditions of the proposal distributions q, which are verified with our practical choices (see Section 4). Even if the geometric ergodicity has not been proved for the standard PGibbs sampler, as other (more complex) versions have been proposed which ensured the geometric ergodicity, and as again, we need less than geometric ergodicity in **(SAEM5')**, we also implement SAEM with the standard PGibbs.

Note that we have been able to weaken the assumption that the generated Markov chain takes its values in a compact subset. This is a strong point, as $\underline{\mathbf{X}}$ is the solution of a diffusion and is generally not bounded.

4. Simulation study

The respective performances of the SAEM-PMCMC (both PMMH and PGibbs) and the SAEM implemented with a standard MCMC are compared on two models of various complexity: the Ornstein-Uhlenbeck process with additive error and the time-inhomogeneous Gompertz process with stochastic volatility error. The linearized estimation method proposed by Overgaard et al. (2005) which combines the extended Kalman Filter to the FOCE algorithm and is implemented in the PSM R package (Klim et al., 2009), is also compared to the two SAEM algorithms but only on the Ornstein-Uhlenbeck example, the use of a stochastic volatility error model being not allowed by the PSM package. We compare the results by computing the mean and standard deviation of all the estimators obtained on 100 datasets generated for different design and parameters value.

The SAEM algorithm requires initial value θ_0 and the choice of the sequence $(\alpha_m)_{m\geq 0}$. The initial values are chosen arbitrarily as the convergence of the SAEM algorithm is not sensitive to the initialization. The step of the stochastic approximation scheme is chosen as recommended by Kuhn and Lavielle (2005): $\alpha_m = 1$ during the first iterations $1 \leq m \leq M_1$, and $\alpha_m = \frac{1}{(m-M_1)^{0.8}}$ during the subsequent ones. Indeed, the initial guess θ_0 might be far from the maximum likelihood value and the first iterations with $\alpha_m = 1$ allow the sequence of estimates to converge to a neighborhood of the maximum likelihood estimate. Subsequently, smaller step sizes during $M - M_1$ additional iterations ensure the almost sure convergence of the algorithm to the maximum likelihood estimate. We implement the SAEM algorithm with $M_1 = 50$ and $M = 100$ iterations.

The `PSM.estimate` function of the PSM package requires initial values, which are similar to those of the SAEM algorithm. We were not able to avoid numerical problems when using the `PSM.estimate` function with unconstrained optimization. Therefore, `PSM.estimate` function is implemented with lower and upper bounds for parameters.

4.1. Example 1: Ornstein-Uhlenbeck process

4.1.1. Ornstein-Uhlenbeck mixed model

The Ornstein-Uhlenbeck process has been widely used in neuronal modeling, biology, and finance (see *e.g.* Kloeden and Platen, 1992). Consider an SDE mixed model driven by the Ornstein-Uhlenbeck process and an additive error model

$$\begin{aligned} y_{ij} &= X_{ij} + \varepsilon_{ij}, \;\; \varepsilon_{ij} \sim \mathcal{N}(0, \sigma^2), \\ dX_{it} &= -\left(\frac{X_{it}}{\tau_i} - \kappa_i\right) dt + \gamma dB_{it}, \quad X_0 = 0 \end{aligned}$$

where $\kappa_i \in \mathbb{R}$, $\tau_i > 0$. We set $\phi_i = (\log(\tau_i), \kappa_i)$ the vector of individual random parameters. We assume that $\log(\tau_i) \sim_{i.i.d.} \mathcal{N}(\log(\tau), \omega_\tau^2)$, $\kappa_i \sim_{i.i.d.} \mathcal{N}(\kappa, \omega_\kappa^2)$. The parameter vector is $\theta = (\log\tau, \kappa, \omega_\tau, \omega_\kappa, \gamma, \sigma)$. This model can be discretized at times t_{ij}, resulting in a state-space model with random parameters and non-finite state space:

$$X_{ij} = X^i_{ij-1} e^{-\Delta_{ij}\tau_i} - \kappa_i \tau_i (1 - e^{-\Delta_{ij}\tau_i}) + \eta_{ij}, \quad \eta_{ij} \sim \mathcal{N}\left(0, \gamma^2 \tau_i (1 - e^{-\Delta_{ij}\tau_i})\right),$$

where $\Delta_{ij} = t_{ij} - t_{ij-1}$ The vector $X_i = (x_{i1}, \ldots, x_{iJ})$ conditional on ϕ_i is Gaussian with mean vector m_{iX} and covariance matrix G_{iX} equal to

$$\begin{aligned} m_{iX} &= \left(\tau_i \kappa_i \left(1 - e^{-\frac{t_{i1}}{\tau_i}}\right), \ldots, \tau_i \kappa_i \left(1 - e^{-\frac{t_{iJ}}{\tau_i}}\right)\right)', \\ G_{iX} &= \left(\frac{\tau_i \gamma^2}{2}\left(1 - e^{-\frac{2\min(t_{ij}, t_{ik})}{\tau_i}}\right) e^{-\frac{|t_{ij} - t_{ik}|}{\tau_i}}\right)_{1 \leq j,k \leq J}, \end{aligned} \tag{17}$$

where $'$ is the transposed vector. Altough this SDE is linear, the Gaussian transition density $p(X_{ij}, \Delta_{ij}, t_{ij-1}|X_{ij-1}, \phi_i; \theta)$ is a nonlinear function of ϕ_i. Thus, the likelihood has no closed form and the exact estimator of θ is unavailable.

This example is a toy example. First, we compare the performances of the SAEM-PMCMC algorithm to those of the SAEM algorithm implemented with a simple MCMC and the PSM function. Next, we compare the influence of the number of particles K and the choice of the proposal distribution $q(X_{t_{ij}}|\ X_{ij-1}, y_{ij}, \phi_i; \theta)$ in the SAEM-PMCMC.

4.1.2. *Simulation step of the SAEM algorithm*

Whereas the conditional distribution $p(X_{i,1:J}|, y_{i,0:J}, \phi_i; \theta)$ is Gaussian, the joint distribution $p(X_{i,1:J}, \phi_i|, y_{i,0:J}; \theta)$ is not explicitly and we have to resort to a MCMC algorithm at the S-step.

A first solution is to implement a standard MCMC algorithm, alternatively simulating under the distributions $p(X_{i,1:J}|, y_{i,0:J}, \phi_i; \theta)$ and $p(\phi_i|, y_{i,0:J},\ X_{i,1:J}; \theta)$ for each subject. The posterior distribution $p(X_{i,1:J}|, y_{i,0:J},\ \phi_i; \theta)$ is Gaussian with easily computable mean vector and variance matrix derived from (17). Similarly, the posterior distribution of κ_i is Gaussian with explicit mean and variance. On the other hand, the posterior distribution of τ_i is not explicit and we use a Metropolis-Hastings step with a random walk proposal. This MCMC is the ideal one since the filtering distribution has an explicit expression. However, we want to use this toy example to test the performances of the SAEM-PMCMC algorithm. As a consequence, we also implement the SAEM coupled with the PMMH algorithm.

If we consider implementing a PMMH at the S-step, we have to choose two proposals $q_{SMC}(X_{t_{ij}}|\ X_{ij-1}, y_{ij}, \phi_i; \theta)$ and $q(\cdot|\phi_i)$. As said before, in this particular linear SDE, the proposal $q(X_{ij}|\ X_{ij-1}, y_{ij}, \phi_i; \theta)$ can be the optimal proposal, the exact posterior density $p(X_{ij}|X_{ij-1},\ y_{ij}, \phi_i; \theta)$, which minimizes the variance of the particle weights. Indeed, this distribution is explicit for the Ornstein-Uhlenbeck, Gaussian with conditional mean and variance easily computable. As an alternative proposal, we also consider the transition density $q_{SMC}(X_{t_{ij}}|\ X_{ij-1}, y_{ij}, \phi_i; \theta) = p(X_{ij}|X_{ij-1}, \phi_i; \theta)$. The proposal $q(\cdot|\phi)$ for the individual parameters ϕ within the PMCMC algorithm is a classical random walk on each component of vector ϕ.

4.1.3. *Maximization step of the SAEM algorithm*

The maximization step is based on the sufficient statistics. The statistics for the parameters $\mu = (\log \tau, \kappa)$, $\Omega = diag(\omega_\tau^2, \omega_\kappa^2)$ and σ are the three classic ones for mixed models (see *e.g.* Samson et al., 2007): $S_1(\mathbf{y}, X, \Phi) = \sum_{i=1}^n \phi_i$, $S_2(\mathbf{y}, X, \Phi) = \sum_{i=1}^n \phi_i \phi_i'$, and $S_3(\mathbf{y}, X, \Phi) = \sum_{i=1}^n (y_i - X_i)'(y_i - X_i)$. Let s_{1m}, s_{2m}, s_{3m} denote the corresponding stochastic approximated conditional expectations at iteration m of SAEM. The M step for these parameters reduces to

$$\widehat{\mu}^{(m)} = \frac{1}{n} s_{1m} \quad \widehat{\Omega}^{(m)} = \frac{1}{n} s_{2m} - \frac{1}{n^2} s_{1m} s_{1m}' \quad \widehat{\sigma}^{(m)} = \sqrt{\frac{1}{n} s_{3m}}.$$

The statistic corresponding to γ depends on the SDE. For the Ornstein-Uhlenbeck, we have

$$S_4(\mathbf{y}, X, \Phi) = \sum_{i=1}^n \sum_{j=1}^J \left(X_{it_{ij}}^i - X_{it_{ij-1}}^i e^{-\Delta_{ij}\tau_i} - \kappa_i \tau_i (1 - e^{-\Delta_{ij}\tau_i}) \right)^2.$$

The M step is thus $\widehat{\gamma}^{(m)} = \sqrt{\frac{1}{nJ} s_{4m}}$.

4.1.4. Simulation design and results

Three different designs are used for the simulations with equally spaced observation times: $n = 20$, $J = 40$, $\Delta = 0.5$; $n = 40$, $J = 20$, $\Delta = 0.5$ and $n = 20$, $J = 40$, $\Delta = 2.5$. Two sets of parameter values are used. The first set is $\log(\tau) = 0.6$, $\kappa = 1$, $\omega_\tau = 0.1$, $\omega_\kappa = 0.1$, $\gamma = 0.05$, $\sigma = 0.05$. The second set uses greater variances and is $\log(\tau) = \log(10)$, $\kappa = 1$, $\omega_\tau = 0.5$, $\omega_\kappa = 0.5$, $\gamma = 0.5$, $\sigma = 0.5$. For each design and each set of parameter values, one hundred datasets are simulated.

The three algorithms are initialized with the following values. For the first set of parameters, we set $\widehat{\log(\tau)}_0 = 1.1$, $\widehat{\kappa}_0 = 1.5$, $\widehat{\omega_{\tau 0}} = 0.5$, $\widehat{\omega_{\kappa 0}} = 0.5$, $\widehat{\gamma}_0 = 0.25$ and $\widehat{\sigma}_0 = 0.25$, i.e. the initial standard deviations are 5 times greater than the true standard deviations. For the second set of parameters, we set $\widehat{\log(\tau)}_0 = \log(10) + 0.5$, $\widehat{\kappa}_0 = 1.5$, $\widehat{\omega_{\tau 0}} = 0.5$, $\widehat{\omega_{\kappa 0}} = 0.5$, $\widehat{\gamma}_0 = 5$ and $\widehat{\sigma}_0 = 5$.

PSM function has numerical convergence problem for some datasets (9 among 100 when $n = 20$ and 46 among 100 when $n = 40$), which are excluded from the PSM results. The SAEM-MCMC algorithm is implemented with $N = 100$ MCMC iterations. Several values of N and K are used in the SAEM-PMCMC algorithm.

Figure 1 presents the convergence of the SAEM-PMCMC algorithm for one dataset simulated with the second set of parameters, $n = 40$ and $J = 20$. This illustrates the low dependence of the initialization of SAEM and the quick convergence in a small neighborhood of the maximum likelihood.

Table 1 presents the results of the SAEM-PMCMC, SAEM-MCMC algorithms and PSM function obtained for the three designs and two sets of parameters. The results are almost identical for the two SAEM algorithms and very satisfactory. PSM has greater bias than the two SAEM algorithms, especially for $\log(\tau)$, ω_τ, ω_κ, γ and σ. Besides, note that the design (n, J) of the study affects very little the two SAEM algorithms. On the contrary and surprisingly, PSM results are deteriorated when n increases. Computational time of PSM algorithm is two times larger than the one of SAEM-PMCMC algorithm (40 minutes instead of 20 minutes with $n = 20$).

In this particular linear model, the standard MCMC on $(\underline{\mathbf{X}}, \Phi)$ is "ideal" as the distribution $p(X_{i,0:J} | y_{i,0:J}, \phi_i)$ can be sampled exactly. As a consequence, the SAEM-PMCMC can not be hopped to perform better. However, results show that the use of a particle approximation of $p(X_{i,0:J} | y_{i,0:J}, \phi_i)$ does not deteriorate the quality of estimation.

In a second part, we study the influence of the particles number K in the SMC algorithm. Table 2 presents the results obtained on the 100 datasets simulated with $n = 40, J = 20$ and different implementations of the SAEM-PMCMC algorithm. We successively use $K = 25$, $K = 50$ and $K = 100$ particles and either $N = 50$, or $N = 150$ PMCMC iterations, using the exact posterior distribution $q(X_{ij} | X_{ij-1}, y_{ij}, \phi_i; \theta) = p(X_{ij} | X_{ij-1}, y_{ij}, \phi_i; \theta)$. The results are almost identical. On the contrary, the choice of the proposal $q(X_{ij} | y_{ij}, X_{ij-1}, \phi; \gamma, \sigma)$ in the SMC algorithm may affect the result. The algorithm fails to converge due to the degeneracy of the particles on 90% of the simulated datasets (results not shown), even when increasing significantly the number of particles ($K = 1000$). In conclusion, although the proposal q seems to be crucial in these simulations, the number of particles K has less influence on the convergence of the estimation algorithm. This is in concordance with the theoretical results proved by Andrieu et al. (2010).

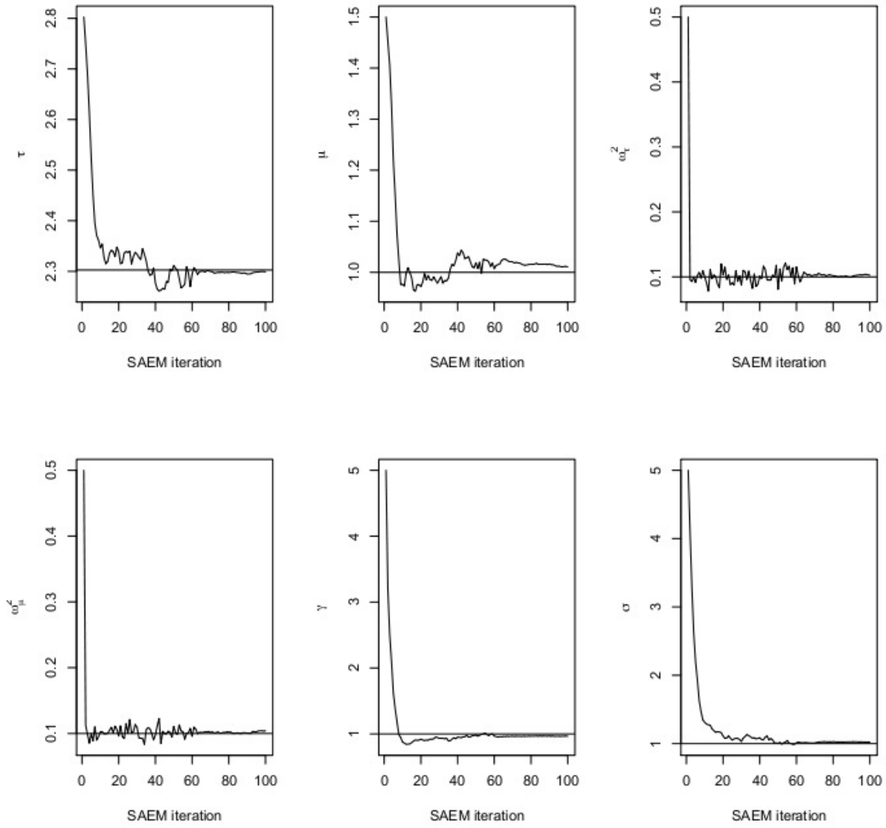

FIGURE 1. *Convergence of the SAEM-PMCMC estimates* $\widehat{\log(\tau)}_m$, $\widehat{\kappa}_m$, $\widehat{\omega_\tau}_m$, $\widehat{\omega_\kappa}_m$, $\widehat{\gamma}_m$ *and* $\widehat{\sigma}_m$ *along the SAEM iterations for one dataset simulated with the Ornstein-Uhlenbeck mixed model, with the second set of parameters,* $n = 40$ *and* $J = 20$*. Horizontal lines represent the true values.*

TABLE 1. *Ornstein-Uhlenbeck mixed model: mean and standard deviations for $\widehat{\theta}$ obtained by the SAEM-PMCMC, SAEM-MCMC algorithms and the PSM function on 100 simulated datasets with two designs ($n = 20$, $J = 40$ and $n = 40$, $J = 20$) and two sets of parameters. PMCMC is implemented with $K = 50$ particles and the exact posterior proposal $q(X_{t_{ij}}|X_{t_{ij-1}}, y_{ij}, \phi_i;\theta) = p(X_{t_{ij}}|X_{t_{ij-1}}, y_{ij}, \phi_i;\theta)$.*

Parameters		$\log(\tau)$	κ	ω_τ	ω_κ	γ	σ
True value		0.600	1.000	0.100	0.100	0.050	0.050
$n = 20$, $J = 40$, $\Delta = 0.5$							
SAEM-PMCMC	Mean	0.600	1.004	0.085	0.094	0.049	0.049
	SD	0.024	0.025	0.032	0.031	0.005	0.003
SAEM-MCMC	Mean	0.602	1.002	0.084	0.089	0.050	0.049
	SD	0.023	0.021	0.022	0.022	0.005	0.002
PSM	Mean	0.616	1.001	0.049	0.050	0.102	0.095
	SD	0.071	0.030	0.005	0.002	0.056	0.022
$n = 40$, $J = 20$, $\Delta = 0.5$							
SAEM-PMCMC	Mean	0.600	1.000	0.096	0.094	0.050	0.050
	SD	0.027	0.025	0.026	0.020	0.004	0.002
SAEM-MCMC	Mean	0.599	1.000	0.096	0.092	0.051	0.050
	SD	0.027	0.025	0.024	0.022	0.004	0.002
PSM	Mean	0.620	0.979	0.049	0.050	0.095	0.105
	SD	0.020	0.137	0.005	0.003	0.016	0.086
True value		2.30	1.00	0.50	0.50	0.50	0.50
$n = 20$, $J = 40$, $\Delta = 2.5$							
SAEM-PMCMC	Mean	2.315	1.001	0.473	0.448	0.500	0.500
	SD	0.145	0.136	0.115	0.094	0.031	0.041
SAEM-MCMC	Mean	2.319	0.999	0.470	0.446	0.499	0.495
	SD	0.150	0.141	0.117	0.088	0.031	0.040
PSM	Mean	2.386	1.101	0.489	0.541	0.531	0.502
	Sd	0.242	0.227	0.029	0.088	0.129	0.132

However, as emphasized by Andrieu et al. (2010), increasing K can make arbitrarily small the probability of visiting unfavorable states by the Markov chain, which correspond to large values of the ratio target density to proposal density. But our simulations show that when choosing the "optimal" proposal density q, namely the exact posterior distribution, the ratio of the target density to the proposal density seems to have no large values and thus the number of particles K has a very low influence. This low influence of K can also be explained by the fact that the SAEM only requires few iterations of PMCMC, without convergence of the Markov chain to the stationary distribution. The convergence is made over the iterations of SAEM.

4.2. *Example 2: stochastic volatility Gompertz model*

4.2.1. *Stochastic volatility Gompertz mixed model*

The Gompertz model is a well-known growth model (see *e.g.* Jaffrézic and Foulley, 2006). Recently, Donnet et al. (2010) proposed a stochastic version of this model to take into account random fluctuations in growth process. The stochastic volatility Gompertz mixed model is:

$$\begin{aligned} y_{ij} &= X_{ij}(1+\varepsilon_{ij}),\ \varepsilon_{ij} \sim \mathcal{N}(0,\sigma^2), \\ dX_{it} &= B_i C_i e^{-C_i t} X_{it}dt + \gamma X_{it} dB_{it}, \quad X_{i0} = A_i e^{-B_i} \end{aligned} \tag{18}$$

TABLE 2. *Ornstein-Uhlenbeck mixed model, $n = 40$, $J = 20$: mean and standard deviations for $\widehat{\theta}$ obtained by the SAEM-PMCMC algorithm on 100 simulated datasets. The PMCMC algorithm is successively implemented with posterior proposal $q(X_{t_{ij}}|X_{t_{ij-1}}, y_{ij}, \phi_i;\theta) = p(X_{t_{ij}}|X_{t_{ij-1}}, y_{ij}, \phi_i;\theta)$ and $K = 25$, $K = 50$ or $K = 100$ particles, $N = 10$ or $N = 150$ PMCMC iterations.*

Parameters	$\log(\tau)$	κ	ω_τ	ω_κ	γ	σ
True value	0.600	1.000	0.100	0.100	0.050	0.050
SAEM-PMCMC with posterior proposal, $K = 25$, $N = 10$						
Mean	0.600	1.000	0.097	0.094	0.051	0.050
SD	0.029	0.028	0.023	0.020	0.005	0.003
SAEM-PMCMC with posterior proposal, $K = 50$, $N = 10$						
Mean	0.600	1.001	0.098	0.094	0.051	0.050
SD	0.027	0.027	0.023	0.019	0.005	0.003
SAEM-PMCMC with posterior proposal, $K = 100$, $N = 10$						
Mean	0.600	1.000	0.098	0.091	0.051	0.050
SD	0.029	0.028	0.023	0.021	0.004	0.002
SAEM-PMCMC with posterior proposal, $K = 50$, $N = 150$						
Mean	0.600	1.000	0.096	0.094	0.050	0.050
SD	0.027	0.025	0.026	0.020	0.004	0.002

where $A_i > 0, B_i > 0, C_i > 0$. We set $\phi_i = (\log A_i, \log B_i, \log C_i)$ the vector of individual random parameters. The initial condition is a function of the random individual parameters ϕ_i. We assume that $\log A_i \sim_{i.i.d.} \mathcal{N}(\log A, \omega_A^2)$, $\log B_i \sim_{i.i.d.} \mathcal{N}(\log B, \omega_B)$, $\log C_i \sim_{i.i.d.} \mathcal{N}(\log C, \omega_C^2)$ and the parameter of interest is $\theta = (\log A, \log B, \log C, \omega_A, \omega_B, \omega_C, \gamma, \sigma)$. By the Itô formula, the conditional expectation and variance of $\log(X_{it})|\phi_i$, for $t \geq 0$, are

$$\mathbb{E}(\log X_{it}|\phi_i) = \log A_i - B_i e^{-C_i t} - \frac{1}{2}\gamma^2 t, \quad \mathrm{Var}(\log X_{it}|\phi_i) = \gamma^2 t.$$

The transition density of $(\log(X_t))$ is explicit, Gaussian and equals to

$$\log p(\log X_{ij}, \Delta_{ij}, t_{ij-1}|\log X_{ij-1}, \phi_i;\theta) = -\tfrac{1}{2}\log(2\pi\gamma^2\Delta_{ij})$$
$$-\frac{1}{2}\frac{\left(\log X_{ij} - \log X_{ij-1} + B_i e^{-Ct_{ij-1}}(e^{-C_i\Delta_{ij}} - 1) - \frac{1}{2}\gamma^2\Delta_{ij}\right)^2}{\gamma^2\Delta_{ij}}.$$

which is a nonlinear function of ϕ_i. As a consequence, the likelihood has no closed form and the exact estimator of θ is unavailable.

4.2.2. *Simulation step of the SAEM algorithm*

First, the S step of the SAEM algorithm is tackled with a standard MCMC algorithm. Due to the multiplicative structure of the observation noise, the distribution $p(X_{i,0:J}|y_{i,0:J}, \phi_i, \theta)$ is no more explicit and we have to resort to a random walk proposal to simulate the trajectories $X_{i,0:J}$ conditionally to the observations $y_{i,0:J}$. The components of $X_{i,0:J}$ are updated time by time using a random walk proposal. A random walk proposal is also used to simulate the individual parameters ϕ_i from the conditional distribution.

Secondly, we implement the SAEM-PMCMC algorithm. A first attempt proved that jointly updating the latent process X and ϕ with the PMMH algorithm implied bad practical convergence properties. As a consequence, in this particular example we use the Particle Gibbs sampler. Although the theoretical convergence of the SAEM coupled with the Particle Gibbs sampler can not be proved, its practical convergence is illustrated here.

In order to implement the PGibbs sampler, the proposal $q_{SMC}(X_{ij}|X_{ij-1}, y_{ij},\ \phi_i;\theta)$ has to be chosen. We propose to approximate the ideal proposal $p(X_{ij}|X_{ij-1}, y_{ij}, \phi_i;\theta)$ by a Gaussian distribution with mean and variance deduced from the true ones. More precisely, we consider the following proposal on $\log X_{t_{ij}}$

$$q_{SMC}(\log X_{t_{ij}}|\log X_{ij-1}, y_{ij}, \phi_i;\theta) = \mathscr{N}(m_{X_{ij},post}, \Gamma_{X_{ij},post}),$$

with

$$\begin{aligned}
\Gamma_{X_{ij},post} &= \left(\sigma^{-2} + (\gamma^2\Delta_{ij})^{-1}\right)^{-1}, \\
m_{X_{ij},post} &= \Gamma_{X_{ij},post}\, \mu_{X_{ij},post}, \\
\mu_{X_{ij},post} &= \left(\frac{\log y_{ij}}{\sigma^2} + \frac{1}{\gamma^2\Delta_{ij}}\left(\log X_{ij-1} - B_i e^{-C_i t_{ij-1}}(e^{-C_i\Delta_{ij}} - 1) - \frac{\gamma^2\Delta_{ij}}{2}\right)\right).
\end{aligned}$$

and then take the exponential to obtain a candidate for $X_{t_{ij}}$. The proposal $q(\cdot|\phi)$ for the individual parameters ϕ is a classical random walk on each component of the vector ϕ.

4.2.3. *Maximization step of the SAEM algorithm*

The sufficient statistics and the M step are the same as in Example 1 for parameters $\mu = (\log A, \log B, \log C)$, $\Omega = diag(\omega_A^2, \omega_B^2, \omega_C^2)$ and σ. As the parameter γ appears both in the expectation and the variance of $\log(X_t)|\phi$, its estimator is not the same as in Example 1. The sufficient statistic corresponding to γ is

$$S_4(\mathbf{y}, \log X, \Phi) = \sum_{i=1}^{n}\sum_{i=1}^{J} \Delta_{ij}\left(\log X_{it_{ij}} - \log X_{it_{ij-1}} + B_i e^{-C_i t_{ij-1}}(e^{-C_i\Delta_{ij}} - 1)\right)^2.$$

Let s_{4m} denote the stochastic approximation of this sufficient statistic at iteration m of the SAEM algorithm. For the sake of simplicity, we assume that the step size Δ_{ij} is a constant Δ. The estimator $\widehat{\gamma}_m$ at iteration m is deduced by maximizing the complete likelihood:

$$\widehat{\gamma}_m = \sqrt{\frac{2}{\Delta}\left(-1 + \sqrt{1 + \frac{s_{4m}}{nJ}}\right)}$$

When Δ_{ij} is not a constant, the estimator is more complex but also explicit.

4.2.4. *Simulation results*

We simulate two experimental designs with the same observation times for all individuals: $n = 20$, $J = 50$, $\Delta = 0.02$ and $n = 50$, $J = 50$, $\Delta = 0.02$ ($t_0 = 0$), respectively. As in Donnet et al. (2010), we use the following population parameters: $\log A = \log(3000)$, $\log B = \log(5)$, $\log C = \log(14)$, $\omega_A = 0.1$, $\omega_B = 0.1$, $\omega_C = 0.1$. The values of γ and σ are 0.5 and 0.1, respectively. Figure 2

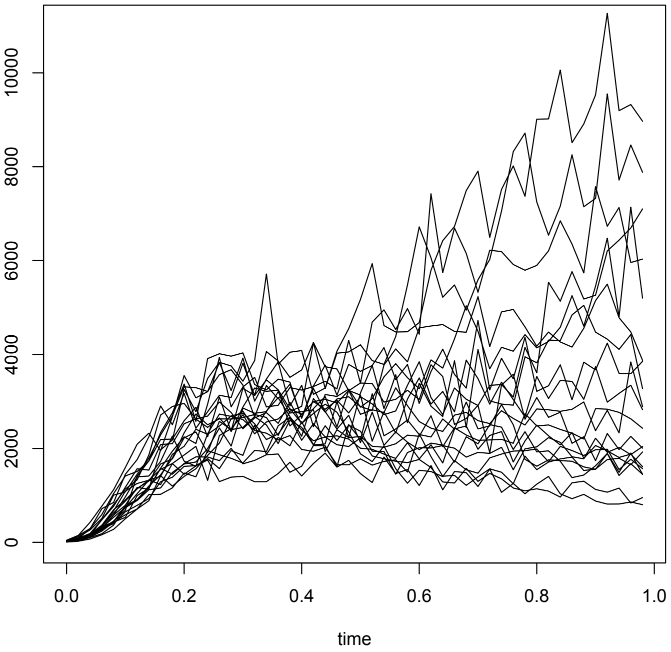

FIGURE 2. *Simulated dataset of the Gompertz mixed model with* $n = 20$ *subjects,* $\gamma = 0.5$ *and* $\sigma = 0.1$

illustrates the variability in the data for one simulated dataset with $n = 20$. For each design, 100 datasets are simulated and θ is estimated with the two algorithms (SAEM-PMCMC and SAEM-MCMC). The SAEM algorithm is initialized with $\widehat{\log A}_0 = 8.21$, $\widehat{\log B}_0 = 1.81$, $\widehat{\log C}_0 = 2.84$, $\widehat{\omega_{A}}_0 = 0.5$, $\widehat{\omega_{B}}_0 = 0.5$, $\widehat{\omega_{C}}_0 = 0.5$, $\widehat{\gamma}_0 = 1.2$, $\widehat{\sigma}_0 = 0.66$. The PMCMC is implemented with $N = 100$ iterations and $K = 50$ particles. The MCMC is implemented with $N = 100$ iterations.

Results are presented in Table 3. Both algorithms give satisfactory results overall given the complexity of the problem. Nevertheless, we clearly see that the SAEM-PMCMC algorithm performs better than the SAEM-MCMC algorithm. Indeed, for the population means $(\log A, \log B, \log C)$, the population standard deviations $(\omega_A, \omega_B, \omega_C)$ and γ, the bias and variance are smaller for SAEM-PMCMC. The difference is obvious for γ, which is estimated with a large bias by SAEM-MCMC while this is not the case for SAEM-PMCMC.

The improved quality of estimation has a computational cost since it takes less than 8 minutes for the SAEM-MCMC and nearly 23 minutes for the SAEM-PMCMC algorithm to supply results. However, estimating precisely the population parameters is a key issue in population studies (such a genetic specification) and a bias can have real consequences on the conclusions.

5. Discussion

The stochastic differential mixed-effects models are quite widespread in the applied statistics field. However, in the absence of efficient and computationally reasonable estimation methods,

TABLE 3. *Stochastic volatility Gompertz mixed model: mean and standard deviations of $\widehat{\theta}$ obtained by the SAEM-PMCMC and SAEM-MCMC algorithms, on the 100 datasets with designs $n = 20$, $J = 50$ and $n = 50$, $J = 50$.*

Parameters		$\log A$	$\log B$	$\log C$	ω_A	ω_B	ω_C	γ	σ
True value		8.006	1.609	2.639	0.100	0.100	0.100	0.500	0.100
$n = 20, J = 50$									
SAEM-PMCMC	Mean	8.363	1.674	2.568	0.096	0.092	0.129	0.605	0.098
	SD	0.348	0.064	0.086	0.159	0.021	0.259	0.261	0.006
SAEM-MCMC	Mean	8.810	1.752	2.457	0.050	0.083	0.051	0.771	0.097
	SD	0.440	0.081	0.087	0.054	0.018	0.039	0.184	0.010
$n = 50, J = 50$									
SAEM-PMCMC	Mean	8.092	1.623	2.624	0.136	0.097	0.119	0.522	0.100
	SD	0.105	0.024	0.031	0.142	0.013	0.073	0.149	0.004
SAEM-MCMC	Mean	8.522	1.704	2.525	0.073	0.090	0.077	0.746	0.098
	SD	0.252	0.045	0.059	0.028	0.012	0.025	0.118	0.004

a simplifying assumption is often made: either the observation noise or the volatility term are standardly neglected.

In this paper we present an EM algorithm combined with a Particular Monte Carlo Markov Chain method to estimate parameters in stochastic differential mixed-effects models including observation noise. We prove the convergence of the algorithm towards the maximum likelihood estimator when the transition density is explicit. This proof is classical as the PMCMC acts as an exact marginal MCMC.

The suggested method supplies accurate parameter estimation in a really moderate computational time on practical examples. Moreover, it is not restricted to homogeneous time SDE. On various simulated datasets, we illustrate the superiority of this method over the SAEM algorithm combined with a standard MCMC algorithm. This efficiency is due to the fact that the PMCMC algorithm takes advantage of the Markovian properties of the non-observed process. Simulations with less individuals should also be tested.

A major advantage of this methodology is its automatic implementation. Indeed the generation of the non-observed process does involve less tuning parameters than a standard MCMC, such as the size of the random move in the random walk Metropolis-Hastings algorithm. For example, the number of particles is not crucial, as illustrated in Example 1. In this paper we present a PMCMC algorithm where all the particles are re-sampled at each iteration. Many other resampling distributions have been proposed in the literature, trying to achieve an optimal procedure. West (1993) developed an effective method of adaptive importance sampling to address this issue. The procedure developed by Pitt and Shephard (2001) is similar in spirit and has real computational advantages. A stratified resampling is proposed in Kitawaga (1996). Each resampling distribution implies a specific update of the weights.

6. Appendix

Sketch of the proof of Theorem 1 The proof is essentially the same as those proposed by Kuhn and Lavielle (2005) except that two assumptions are weakened. The arguments that justify these two points rest upon Benveniste et al. (1990)'s original proof of stochastic approximation scheme with Markov dependency. Benveniste et al. (1990) assume that the Markov kernel is Lipschitz with respect to both θ and (x,φ). Consequently, to ensure the existence of the function

$$\nu_s(x,\varphi) = \sum_{m\geq 0} (\Pi^m_{\hat{\theta}(s)} S(x,\varphi) - \pi S),$$

which is a key element in the proof, the assumption that the Markov chain (X,Φ) stays in a compact set is required to obtain the existence of moments of every order of the chain (assumption **(A5)** of Benveniste et al. (1990)). But we only assume a Lipschitz condition with respect to θ for the Markov kernel Π **(SAEM5')**. So we do not need the compactness assumption for the Markov chain. An additional argument is needed to ensure the existence of the function $\nu_s(x,\varphi)$. Kuhn and Lavielle (2005) introduce the uniform geometric ergodicity of the chain. We claim that we can weaken this assumption, as soon as the sufficient statistics are bounded **(SAEM4)**. Indeed, the ergodicity of order 2 as stated in **(SAEM5')** is then sufficient to prove the existence of $\nu_s(z)$.

The rest of the proof of the convergence of SAEM remains the same and we refer the reader to Kuhn and Lavielle (2005). □

Acknowledgments

Sophie Donnet has been financially supported by the french program ANR BigMC.

References

Allassonnière, S., Kuhn, E., and Trouvé, A. (2010). Construction of bayesian deformable models via stochastic approximation algorithm: A convergence study. *Bernoulli*, 641-678:605–908.

Altman, R. (2007). Mixed hidden markov models: An extension of the hidden markov model to the longitudinal data setting. *J. Amer. Statist. Assoc*, 102 (407):201–210.

Andrieu, C., Doucet, A., and Holenstein, R. (2010). Particle Markov chain Monte Carlo methods. *J. R. Statist. Soc. B*, 72(3)(72):1–33.

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

Benveniste, A., Métivier, M., and Priouret, P. (1990). *Adaptive algorithms and stochastic approximations*, volume 22 of *Applications of Mathematics (New York)*. Springer-Verlag, Berlin.

Chopin, N. and Singh, S. S. (2013). On the particle Gibbs sampler. *ArXiv e-prints*.

Davidian, M. and Giltinan, D. (1995). *Nonlinear models to repeated measurement data*. Chapman and Hall.

Delattre, M., Genon-Catalot, V., and Samson, A. (2013). Maximum likelihood estimation for stochastic differential equations with random effects. *Scand. J. Statistics*, 40(2):322–343.

Delattre, M. and Lavielle, M. (2012). Maximum Likelihood Estimation in Discrete Mixed Hidden Markov Models using the SAEM algorithm. *Comput. Statist. Data Anal.*, 56(6):2073–2085.

Delattre, M. and Lavielle, M. (2013). Coupling the saem algorithm and the extended kalman filter for maximum likelihood estimation in mixed-effects diffusion models. *Stat. Interface*.

Delyon, B., Lavielle, M., and Moulines, E. (1999). Convergence of a stochastic approximation version of the EM algorithm. *Ann. Statist.*, 27:94–128.

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. B*, 39:1–38.

Donnet, S., Foulley, J., and Samson, A. (2010). Bayesian analysis of growth curves using mixed models defined by stochastic differential equations. *Biometrics*, 66:733–741.

Donnet, S. and Samson, A. (2008). Parametric inference for mixed models defined by stochastic differential equations. *ESAIM P&S*, 12:196–218.

Donnet, S. and Samson, A. (2013). A review on estimation of stochastic differential equations for pharmacokinetic - pharmacodynamic models. *Adv Drug DELIVER REV*, 65(7):929–939.

Jaffrézic, F. and Foulley, J. (2006). Modelling variances with random effects in non linear mixed models with an example in growth curve analysis. *XXIII International Biometric Conference, Montreal, 16-21 juillet*.

Kitawaga, G. (1996). Monte carlo filter and smoother for non-gaussian nonlinear state space models. *J. Comp. Graph. Statist.*, 5(1)(5):1–25.

Klim, S., Mortensen, S., Kristensen, N., Overgaard, R., and Madsen, H. (2009). Population stochastic modelling (PSM) - an R package for mixed-effects models based on stochastic differential equations. *Comput Meth Prog Bio*, 94:279–289.

Kloeden, P. E. and Platen, E. (1992). *Numerical solution of stochastic differential equations*, volume 23 of *Applications of Mathematics (New York)*. Springer-Verlag, Berlin.

Kuhn, E. and Lavielle, M. (2005). Maximum likelihood estimation in nonlinear mixed effects models. *Comput. Statist. Data Anal.*, 49:1020–1038.

Maruotti, A. (2011). Mixed hidden markov models for longitudinal data: An overview. *International Statistical Review*, 79 (3):427–454.

Oksendal, B. (2007). *Stochastic differential equations: an introduction with applications*. Springer-Verlag, Berlin-Heidelberg.

Overgaard, R., Jonsson, N., Tornøe, C., and Madsen, H. (2005). Non-linear mixed-effects models with stochastic differential equations: Implementation of an estimation algorithm. *J Pharmacokinet. Pharmacodyn.*, 32(1):85–107.

Picchini, U., De Gaetano, A., and Ditlevsen, S. (2010). Stochastic differential mixed-effects models. *Scand. J. Statist.*, 37:67–90.

Picchini, U., Ditlevsen, S., De Gaetano, A., and Lansky, P. (2008). Parameters of the diffusion leaky integrate-and-fire neuronal model for a slowly fluctuating signal. *Neural computation*, 20(11):2696–2714.

Pinheiro, J. and Bates, D. (2000). *Mixed-effect models in S and Splus*. Springer-Verlag.

Pitt, M. and Shephard, N. (2001). Auxiliary variable based particle filters. In *Sequential Monte Carlo methods in practice*, Stat. Eng. Inf. Sci., pages 273–293. Springer, New York.

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. *J Econometrics*, 171(2):134 – 151.

Samson, A., Lavielle, M., and Mentré, F. (2007). The SAEM algorithm for group comparison tests in longitudinal data analysis based on non-linear mixed-effects model. *Stat. Med.*, 26(27):4860–4875.

Tierney, L. (1994). Markov chains for exploring posterior distributions. *Ann. Statist.*, 22(4):1701–1762.

Tornøe, C., Overgaard, R., Agersø, H., Nielsen, H., Madsen, H., and Jonsson, E. (2005). Stochastic differential equations in NONMEM: implementation, application, and comparison with ordinary differential equations. *Pharm. Res.*, 22(8):1247–58.

West, M. (1993). Approximating posterior distributions by mixtures. *J. R. Statist. Soc. B*, 55(2):409–422.

Wolfinger, R. (1993). Laplace's approximation for nonlinear mixed models. *Biometrika*, 80(4):791–795.