Coupling stochastic EM and Approximate Bayesian computation for parameter inference in state-space models

Umberto Picchini, Adeline Samson

Noname manuscript No.
(will be inserted by the editor)

Coupling stochastic EM and Approximate Bayesian Computation for parameter inference in state-space models

Umberto Picchini · Adeline Samson

Abstract We study the class of state-space models (or hidden Markov models) and perform maximum likelihood inference on the model parameters. We consider a stochastic approximation expectation-maximization (SAEM) algorithm to maximize the likelihood function with the novelty of using approximate Bayesian computation (ABC) within SAEM. The task is to provide each iteration of SAEM with a filtered state of the system and this is achieved using ABC-SMC, that is we used an approximate sequential Monte Carlo (SMC) sampler for the hidden state. Three simulation studies are presented, first a nonlinear Gaussian state-space model then a state-space model having dynamics expressed by a stochastic differential equation, finally a stochastic volatility model. In our examples, ten iterations of our SAEM-ABC-SMC strategy were enough to return sensible parameter estimates. Comparisons with results using SAEM coupled with a standard, non-ABC, SMC sampler show that the ABC algorithm can be calibrated to return accurate solutions.

Keywords hidden Markov model · maximum likelihood · SAEM · sequential Monte Carlo · stochastic differential equation

1 Introduction

State-space models, also known as Hidden Markov models, see Cappé et al. [2005], are widely used in many fields, such as biology, chemistry, ecology, etc. Let us now introduce some notation. Consider an observable, discrete-time stochastic process $\{\mathbf{Y}_t\}_{t\geq t_0}$, $\mathbf{Y}_t \in \mathsf{Y} \subseteq \mathbb{R}^{d_y}$ and a latent and unobserved continuous-time stochastic process $\{\mathbf{X}_t\}_{t\geq t_0}$, $\mathbf{X}_t \in \mathsf{X} \subseteq \mathbb{R}^{d_x}$. Process

U. Picchini
Centre for Mathematical Sciences, Lund University, Lund, Sweden. Phone: +46 (0)46 222 9270; Fax: +46 (0)46 222 4623. E-mail: umberto@maths.lth.se

A. Samson
Universite Grenoble Alpes, LJK, F-38000 Grenoble, France; CNRS, LJK, F-38000 Grenoble, France. E-mail: adeline.leclercq-samson@imag.fr

$\mathbf{X}_t \sim p(x_t|x_{t-1}, \boldsymbol{\theta}_x)$ is assumed Markov with transition densities $p(\cdot)$. Both processes $\{\mathbf{X}_t\}$ and $\{\mathbf{Y}_t\}$ depend on their own (assumed unknown) vector-parameters $\boldsymbol{\theta}_x$ and $\boldsymbol{\theta}_y$ respectively. We think at $\{\mathbf{Y}_t\}$ as a measurement-error-corrupted version of $\{\mathbf{X}_t\}$ and assume that observations for $\{\mathbf{Y}_t\}$ are conditionally independent given $\{\mathbf{X}_t\}$. Our state-space model can be summarised as

$$\begin{cases} \mathbf{Y}_t \sim f(\mathbf{y}_t|\mathbf{X}_t, \boldsymbol{\theta}_y), & t \geq t_0 \\ \mathbf{X}_t \sim p(\mathbf{x}_t|\mathbf{x}_{t-1}, \boldsymbol{\theta}_x). \end{cases} \tag{1}$$

Typically $f(\cdot)$ is considered a known density (or probability mass) function set by the modeller, however this is not always the case, as discussed later on. Regarding $p(\cdot)$, this is typically unknown except for very simple toy models.

Goal of our work is to estimate the parameters $(\boldsymbol{\theta}_x, \boldsymbol{\theta}_y)$ using observations $\mathbf{Y}_{1:n} = (\mathbf{Y}_1, ..., \mathbf{Y}_n)$ from $\{\mathbf{Y}_t\}_{t \geq t_0}$ collected at discrete times $\{t_1, ..., t_n\}$. For ease of notation we refer to the vector $\boldsymbol{\theta} := (\boldsymbol{\theta}_x, \boldsymbol{\theta}_y)$ as the object of our inference.

Bayesian estimation for state-space models has been widely developed. There the goal is to derive analytically the posterior distribution $\pi(\boldsymbol{\theta}|\mathbf{Y}_{1:n})$ or, most frequently, implement an algorithm for sampling draws from the posterior. Sampling procedures are often carried out using Markov chain Monte Carlo (MCMC) or Sequential Monte Carlo (SMC) embedded in MCMC procedures [Andrieu et al., 2010].

In this work we instead aim at maximum likelihood estimation of θ. Several methods have been proposed in the literature, usually based on the well-known EM algorithm [Dempster et al., 1977]. The EM algorithm computes the conditional expectation of the complete-likelihood for the pair $(\{\mathbf{Y}_t\}, \{\mathbf{X}_t\})$ and then produces a (local) maximizer for the likelihood function based on the actual observations $\mathbf{Y}_{1:n}$. One of the problems is to compute the conditional expectation of the state $\{\mathbf{X}_t\}$ given the observations $\mathbf{Y}_{1:n}$. This conditional expectation can be computed exactly with the Kalman filter when the state-space is linear and Gaussian [Cappé et al., 2005]. Otherwise, it has to be approximated. In this work we focus on a stochastic approximation that leads to a stochastic version of the EM algorithm, namely the Stochastic Approximation EM (SAEM) [Delyon et al., 1999]. The problem is to generate, conditionally on the current value of $\boldsymbol{\theta}$ during the EM maximization, an appropriate "proposal" for the state $\{\mathbf{X}_t\}$. Sequential Monte Carlo (SMC) algorithms [Doucet et al., 2001] have already been coupled to stochastic EM algorithms (see e.g. Huys et al. [2006], Huys and Paninski [2009], Lindsten [2013], Ditlevsen and Samson [2014] and references therein). The simplest and most popular SMC algorithm, the bootstrap filter, is easy to implement though much literature has been devoted to propose improvements over such basic filter (Cappe et al. [2007], Jacob [2015]). However here we would like to consider options which are not overspecialised and not requiring expert tuning.

In order to select a path $\{\mathbf{X}_t\}$ to feed SAEM with, we follow a SMC approach based on approximate Bayesian computation (ABC) and specifically we use the ABC-SMC method for state-estimation proposed in Jasra et al.

[2012]. We illustrate our SAEM-ABC-SMC approach for (approximate) maximum likelihood estimation of $\boldsymbol{\theta}$ using three case studies, a nonlinear Gaussian state-space model, a more complex state-space model based on stochastic differential equations and a stochastic volatility model. As a side-product of the method introduced, we also retrieve the filtered state $\{\mathbf{X}_t\}$ at the parameters maximum likelihood estimate. In our examples, when starting from unlikely parameter values SAEM-ABC-SMC shows rapid convergence to the true parameter values after a few iterations (about ten), hence it can be considered as a viable alternative to pure SMC techniques or expensive algorithms for full Bayesian inference.

The paper is structured as follows: in section 2 we introduce the standard SAEM algorithm and basic notions of ABC. In section 3 we propose a new method for maximum likelihood estimation by integrating an ABC-SMC algorithm within SAEM. Section 4 shows simulation results and section 5 summarize conclusions.

2 The complete likelihood and stochastic approximation EM

Recall that $\mathbf{Y}_{1:n} = (\mathbf{Y}_1, ..., \mathbf{Y}_n)$ denotes the available data collected at times $(t_1, ..., t_n)$ and denote with $\mathbf{X}_{1:n} = (\mathbf{X}_1, ..., \mathbf{X}_n)$ the corresponding unobserved states. We additionally set $\mathbf{X}_{0:n} = (\mathbf{X}_0, \mathbf{X}_{0:n})$ for the vector including an initial (fixed or random) state $\mathbf{X}_0$, that is $\mathbf{X}_1$ is generated as $\mathbf{X}_1 \sim p(\mathbf{x}_1|\mathbf{x}_0)$. When the transition densities $p(\mathbf{x}_j|\mathbf{x}_{j-1})$ are available in closed form $(j = 1, ..., n)$, the likelihood function for θ can be written as (here we have assumed a random initial state with density $p(\mathbf{X}_0)$)

$$p(\mathbf{Y}_{1:n}; \boldsymbol{\theta}) = \int p_{\mathbf{Y},\mathbf{X}}(\mathbf{Y}_{1:n}, \mathbf{X}_{0:n}\,; \boldsymbol{\theta})\, d\mathbf{X}_{0:n} = \int p_{\mathbf{Y}|\mathbf{X}}(\mathbf{Y}_{1:n}|\mathbf{X}_{0:n}\,; \boldsymbol{\theta}) p_{\mathbf{X}}(\mathbf{X}_{0:n}; \boldsymbol{\theta})\, d\mathbf{X}_{0:n}$$
$$= \int p(\mathbf{X}_0) \Big\{ \prod_{j=1}^{n} p(\mathbf{Y}_j|\mathbf{X}_j; \boldsymbol{\theta}) p(\mathbf{X}_j|\mathbf{X}_{j-1}; \boldsymbol{\theta}) \Big\} d\mathbf{X}_0 \cdots d\mathbf{X}_n \qquad (2)$$

where $p_{\mathbf{Y},\mathbf{X}}$ is the "complete data likelihood", $p(\mathbf{Y}_j|\mathbf{X}_j)$ the conditional density of $\mathbf{Y}_j$ and $p_{\mathbf{X}}(\mathbf{X}_{0:n}; \boldsymbol{\theta})$ the joint density of $\mathbf{X}_{0:n}$. The last equality in (2) exploits the notion of conditional independence of observations given latent states and the Markovian property of $\{\mathbf{X}_t\}$. In general the likelihood (2) is not explicitly known either because the integral is multidimensional and because typically expressions for transition densities are not available.

In addition, when an exact simulator for the solution of the dynamical process associated with the Markov process $\{\mathbf{X}_t\}$ is unavailable, hence it is not possible to sample from $p(\mathbf{X}_i|\mathbf{X}_{i-1}; \boldsymbol{\theta})$, numerical discretization methods are required. Without loss of generality, say that we have equispaced sampling times such that $t_j = t_{j-1} + \Delta$, with $\Delta > 0$. Now introduce a discretization for the interval $[t_1, t_n]$ given by $\{\tau_1, \tau_h, ..., \tau_{Mh}, ..., \tau_{nMh}\}$ where $h = \Delta/R$ and $R \geq 1$. We take $\tau_1 = t_1$, $\tau_{nMh} = t_n$ and therefore $\tau_i \in \{t_1, ..., t_n\}$ for $i = 1, Mh, 2Mh, ..., nMh$. We denote with N the number

of elements in the discretisation $\{\tau_1, \tau_h, ..., \tau_{Mh}, ..., \tau_{nMh}\}$ and with $\mathbf{X}_{1:N} = (\mathbf{X}_{\tau_1}, \ldots, \mathbf{X}_{\tau_N})$ the corresponding values of $\{\mathbf{X}_t\}$ obtained when using a given numerical/approximated method of choice. Then the likelihood function becomes

$$p(\mathbf{Y}_{0:n};\boldsymbol{\theta}) = \int p_{\mathbf{Y},\mathbf{X}}(\mathbf{Y}_{0:n}, \mathbf{X}_{0:N}\,;\boldsymbol{\theta})\, d\mathbf{X}_{0:N} = \int p_{\mathbf{Y}|\mathbf{X}}(\mathbf{Y}_{0:n}|\mathbf{X}_{0:N}\,;\boldsymbol{\theta}) p_{\mathbf{X}}(\mathbf{X}_{0:N};\boldsymbol{\theta})\, d\mathbf{X}_{0:N}$$
$$= \int \left\{ \prod_{j=0}^{n} p(\mathbf{Y}_j|\mathbf{X}_j;\boldsymbol{\theta}) \right\} p(\mathbf{X}_0) \prod_{i=1}^{N} p(\mathbf{X}_i|\mathbf{X}_{i-1};\boldsymbol{\theta}) d\mathbf{X}_0 \cdots d\mathbf{X}_N,$$

where the product in j is over the $\mathbf{X}_{t_j}$ and the product in i is over the $\mathbf{X}_{\tau_i}$.

2.1 The standard SAEM algorithm

The EM algorithm introduced by Dempster et al. [1977] is a classical approach to estimate parameters of models with non-observed or incomplete data. Let us briefly cover the EM principle. The complete data of the model is $(\mathbf{Y}_{0:n}, \mathbf{X}_{0:N})$, where $\mathbf{X}_{0:N} \equiv \mathbf{X}_{0:n}$ if numerical discretization is not required, and for ease of writing we denote this as $(\mathbf{Y}, \mathbf{X})$ in the remaining of this section. The EM algorithm maximizes the function $Q(\boldsymbol{\theta}|\boldsymbol{\theta}') = \mathbb{E}(L_c(\mathbf{Y}, \mathbf{X}; \boldsymbol{\theta})|\mathbf{Y}; \boldsymbol{\theta}')$ in two steps, where $L_c(\mathbf{Y}, \mathbf{X}; \boldsymbol{\theta}) := \log p_{\mathbf{Y},\mathbf{X}}$ is the log-likelihood of the complete data and $\mathbb{E}$ is the conditional expectation under the conditional distribution $p_{\mathbf{X}|\mathbf{Y}}(\cdot; \boldsymbol{\theta}')$. At the k-th iteration, the E-step is the evaluation of $Q_k(\boldsymbol{\theta}) = Q(\boldsymbol{\theta}\,|\,\widehat{\boldsymbol{\theta}}^{(k-1)})$, whereas the M-step updates $\widehat{\boldsymbol{\theta}}^{(k-1)}$ by maximizing $Q_k(\boldsymbol{\theta})$. For cases in which the E-step has no analytic form, Delyon et al. [1999] introduce a stochastic version (SAEM) of the EM algorithm which evaluates the integral $Q_k(\boldsymbol{\theta})$ by a stochastic approximation procedure. The authors prove the convergence of this algorithm under general conditions if $L_c(\mathbf{Y}, \mathbf{X}; \boldsymbol{\theta})$ belongs to the regular exponential family

$$L_c(\mathbf{Y}, \mathbf{X}; \boldsymbol{\theta}) = -\Lambda(\boldsymbol{\theta}) + \langle \mathbf{S}_c(\mathbf{Y}, \mathbf{X}), \Gamma(\boldsymbol{\theta}) \rangle,$$

where $\langle ., . \rangle$ is the scalar product, Λ and Γ are two functions of $\boldsymbol{\theta}$ and $\mathbf{S}_c(\mathbf{Y}, \mathbf{X})$ is the minimal sufficient statistic of the complete model. The E-step is then divided into a simulation step (S-step) of the missing data $\mathbf{X}^{(k)}$ under the conditional distribution $p_{\mathbf{X}|\mathbf{Y}}(\cdot; \widehat{\boldsymbol{\theta}}^{(k-1)})$ and a stochastic approximation step (SA-step) of the conditional expectation, using $(\gamma_k)_{k\geq 1}$ a sequence of real numbers in $[0, 1]$, such that $\sum_{k=1}^{\infty} \gamma_k = \infty$ and $\sum_{k=1}^{\infty} \gamma_k^2 < \infty$. This SA-step approximates $\mathbb{E}\left[\mathbf{S}_c(\mathbf{Y}, \mathbf{X})|\widehat{\boldsymbol{\theta}}^{(k-1)}\right]$ at each iteration by the value $\mathbf{s}_k$ defined recursively as follows

$$\mathbf{s}_k = \mathbf{s}_{k-1} + \gamma_k(\mathbf{S}_c(\mathbf{Y}, \mathbf{X}^{(k)}) - \mathbf{s}_{k-1}).$$

The M-step is thus the update of the estimates $\widehat{\boldsymbol{\theta}}^{(k-1)}$

$$\widehat{\boldsymbol{\theta}}^{(k)} = \arg\max_{\boldsymbol{\theta}\in\Theta} \left(-\Lambda(\boldsymbol{\theta}) + \langle \mathbf{s}_k, \Gamma(\boldsymbol{\theta})\rangle\right).$$

The starting $\mathbf{s}_0$ can be set to be a vector of zeros.

Usually, the simulation step of the hidden trajectory $\mathbf{X}^{(k)}$ conditionally to the observations $\mathbf{Y}$ cannot be directly performed. We propose to resort to Approximate Bayesian Computation (ABC) for this simulation step.

2.2 The SAEM algorithm coupled to an ABC simulation step

Approximate Bayesian Computation (ABC, Pritchard et al. [1999], Tavaré et al. [1997], Marjoram et al. [2003]) is a class of probabilistic algorithms allowing sampling from an approximation of a posterior distribution. The most typical usage of ABC is when posterior inference on $\boldsymbol{\theta}$ is the goal of the analysis and the purpose is to sample draws from the approximate posterior $\pi_\delta(\boldsymbol{\theta}|\mathbf{Y})$, see Marin et al. [2012] for a review. Here and in the following $\mathbf{Y} \equiv \mathbf{Y}_{1:n}$. The parameter $\delta > 0$ is a "threshold" influencing the quality of the inference, the smaller the δ the more accurate the inference, and $\pi_\delta(\boldsymbol{\theta}|\mathbf{Y}) \equiv \pi(\boldsymbol{\theta}|\mathbf{Y})$ when $\delta = 0$. However in our study we are not interested in conducting Bayesian inference on $\boldsymbol{\theta}$. We will use ABC to sample from an approximation to the posterior distribution $\pi(\mathbf{X}_{0:N}|\mathbf{Y};\boldsymbol{\theta}) \equiv p(\mathbf{X}_{0:N}|\mathbf{Y};\boldsymbol{\theta})$, that is for a fixed value of $\boldsymbol{\theta}$, we wish to sample from $\pi_\delta(\mathbf{X}_{0:N}|\mathbf{Y};\boldsymbol{\theta})$ (recall from section 2.1 that when feasible we can take $N \equiv n$). For simplicity of notation, in the following we avoid specifying the dependence on the current value of $\boldsymbol{\theta}$, which has to be assumed as a deterministic unknown. There are several ways to generate a "candidate" $\mathbf{X}^*_{0:N}$: for example we might consider "blind" forward simulation, meaning that $\mathbf{X}^*_{0:N}$ is simulated from $p_{\mathbf{X}}(\mathbf{X}_{0:N})$ and therefore unconditionally to data (i.e. the simulator is blind with respect to data). Then $\mathbf{X}^*_{0:N}$ is accepted if the corresponding $\mathbf{Y}^*$ simulated from $f(\cdot|\mathbf{X}^*_{1:n})$ is "close" to $\mathbf{Y}$, according to a threshold $\delta > 0$, where $\mathbf{X}^*_{1:n}$ contains the interpolated values of $\mathbf{X}^*_{0:N}$ at sampling times $\{t_1, ..., t_n\}$ and $\mathbf{Y}^* \equiv \mathbf{Y}^*_{1:n}$. Notice that the appeal of the methodology is that knowledge of the probabilistic features of the data generating model is not necessary, meaning that even if the transition densities $p(\mathbf{X}_i|\mathbf{X}_{i-1})$ are not known (hence $p_{\mathbf{X}}$ is unknown) it is enough to be able to simulate from the model (using a numerical scheme if necessary) hence draws $\mathbf{X}^*_{0:N}$ are produced by forward-simulation regardless the explicit knowledge of the underlying densities.

If we consider for a moment $\mathbf{X}^* \equiv \mathbf{X}^*_{0:N}$ as a generic unknown it is easy to realise that we wish to sample from

$$\pi_\delta(\mathbf{X}^*|\mathbf{Y}) \propto J_\delta(\mathbf{Y}, \mathbf{Y}^*) \underbrace{p(\mathbf{Y}^*|\mathbf{X}^*)\pi(\mathbf{X}^*)}_{\propto \pi(\mathbf{X}^*|\mathbf{Y}^*)} \tag{3}$$

where $p(\mathbf{Y}^*|\mathbf{X}^*)$ is the law of the data-generating model in dependence of $\mathbf{X}^*$, the latter having "prior" $\pi(\mathbf{X}^*)$. Here $J_\delta(\cdot)$ is some function that depends on δ

and weights the intractable posterior based on simulated data $\pi(\mathbf{X}^*|\mathbf{Y}^*)$ with high values in regions where $\mathbf{Y}$ and $\mathbf{Y}^*$ are similar; therefore we would like (i) $J_\delta(\cdot)$ to give higher rewards to proposals $\mathbf{X}^*$ corresponding to $\mathbf{Y}^*$ having values close to $\mathbf{Y}$. In addition (ii) $J_\delta(\mathbf{Y},\mathbf{Y}^*)$ is assumed to be a constant when $\mathbf{Y}^*=\mathbf{Y}$ (i.e. when $\delta=0$) so that J_δ is absorbed into the proportionality constant and the *exact* marginal posterior $\pi(\mathbf{X}^*|\mathbf{Y})$ is recovered. Basically the use of (3) means to simulate $\mathbf{X}^*$ from its prior (the product of transition densities), then plug such draw into $f(\cdot|\mathbf{X}^*)$ to simulate $\mathbf{Y}^*$, so that $\mathbf{X}^*$ will be weighted by $J_\delta(\mathbf{Y},\mathbf{Y}^*)$. A common choice for $J_\delta(\cdot)$ is the uniform kernel

$$J_\delta(\mathbf{Y},\mathbf{Y}^*) \propto \mathbb{I}_{\{\rho(\mathbf{Y}^*,\mathbf{Y})\leq\delta\}}$$

where $\rho(\mathbf{Y},\mathbf{Y}^*)$ is some measure of closeness between $\mathbf{Y}^*$ and $\mathbf{Y}$ and $\mathbb{I}$ is the indicator function; see section 4.1 for a further option. However, one of the difficulties is that, in practice, δ has to be set as a compromise between statistical accuracy (with a small positive δ) and computational feasibility (δ not too small). Notice that a proposal's acceptance can be significantly enhanced when the posterior (3) is conditional on summary statistics of data $\eta(\mathbf{Y})$, rather than $\mathbf{Y}$ itself, and in such case we would consider $\rho(\eta(\mathbf{Y}),\eta(\mathbf{Y}^*))$. In the most favourable case, if sufficient statistics $\eta(\cdot)$ for $\boldsymbol{\theta}$ are available then inference based on $\pi_\delta(\boldsymbol{\theta}|\eta(\mathbf{Y}))$ is equivalent to inference based on $\pi_\delta(\boldsymbol{\theta}|\mathbf{Y})$, and of course if in addition $\delta=0$ this results in exact posterior inference. However, in practice for dynamical models it is difficult to identify "informative enough" (if not sufficient) summary statistics $\eta(\cdot)$, but see Martin et al. [2014] and Picchini and Forman [2015]. Another important problem with the strategy outlined above is that "blind simulation" for the generation of $\mathbf{X}^*$ is often poor. In fact, even when the current value of $\boldsymbol{\theta}$ is close to its true value, proposed trajectories rarely follow measurements when (a) the dataset is a long series (see an instance in the example presented in section 4.1) and/or (b) the model is highly erratic, for example when latent dynamics are expressed by a stochastic differential equation (section 4.2). Hence, proposing trajectories conditional to data is a more sensible strategy, and for this purpose sequential Monte Carlo (SMC) [Cappe et al., 2007] methods have emerged as the most successful solution for filtering in non-linear non-Gaussian state-space models.

However, should a simple accept-reject procedure with blind simulations be feasible, algorithm 1 presents a way to perform maximum likelihood estimation via SAEM and ABC acceptance-rejection, and we keep it here also for ease of introduction to more advanced methods that we are about to present. Algorithm 1 illustrates a generic iteration k of a SAEM-ABC method, where the current value of the parameters is $\hat{\boldsymbol{\theta}}^{(k-1)}$ and an updated value of the estimates is produced as $\hat{\boldsymbol{\theta}}^{(k)}$. By iterating the procedure many times, the resulting $\widehat{\theta}^{(k)}$ is an approximate maximum likelihood estimate. The "repeat loop" can be considerably expensive using a distance $\rho(\mathbf{Y}^*,\mathbf{Y})\leq\delta$ as acceptance of $\mathbf{Y}^*$ (hence acceptance of $\mathbf{X}^*$) is a rare event for δ reasonably small. If appropriate statistics $\eta(\cdot)$ are available, it is recommended to consider $\rho(\eta(\mathbf{Y}^*),\eta(\mathbf{Y}))$ instead.

Algorithm 1 A generic iteration of SAEM-ABC using acceptance-rejection

Simulation step: here we update $\mathbf{X}^{(k)}$ using an ABC procedure sampling from $\pi_\delta(\mathbf{X}|\mathbf{Y};\hat{\boldsymbol{\theta}}^{(k-1)})$:

Repeat

- Generate a candidate $\mathbf{X}^*$ from the latent model dynamics conditionally on $\hat{\boldsymbol{\theta}}^{(k-1)}$, either by numerical methods or using the transition density (if available) i.e. by generating using the exact law $p_{\mathbf{X}}(\cdot;\hat{\boldsymbol{\theta}}^{(k-1)})$
- Generate $\mathbf{Y}^*$ from the error model $f(\mathbf{Y}^*|\mathbf{X}^*)$

Until $\rho(\mathbf{Y}^*,\mathbf{Y}) \leq \delta$

Set $\mathbf{X}^{(k)} = \mathbf{X}^*$

Stochastic Approximation step : update of the sufficient statistics

$$\mathbf{s}_k = \mathbf{s}_{k-1} + \gamma_k \left(\mathbf{S}_c(\mathbf{Y},\mathbf{X}^{(k)}) - \mathbf{s}_{k-1}\right)$$

Maximisation step: update $\boldsymbol{\theta}$

$$\widehat{\boldsymbol{\theta}}^{(k)} = \arg\max_{\theta\in\Theta} \left(-\Lambda(\boldsymbol{\theta}) + \langle \mathbf{s}_k, \Gamma(\boldsymbol{\theta})\rangle\right)$$

Below we consider the ABC-SMC methodology from Jasra et al. [2012], which proves much more effective for state-space models.

3 SAEM coupled with an ABC-SMC algorithm for filtering

3.1 The ABC-SMC filter

Here we consider a strategy for filtering that is based on an ABC version of sequential Monte Carlo sampling, as presented in Jasra et al. [2012], with some minor modifications. The advantage of the methodology is that the generation of proposed trajectories is not blind to data, and the ABC distance that is evaluated is "local", i.e. what is evaluated is the proximity of trajectories/particles to each data point $\mathbf{Y}_j$, and "bad trajectories" are killed thus preventing the propagation of unlikely states to the next observation $\mathbf{Y}_{j+1}$ and so on. For simplicity we consider the case $N \equiv n$, $h \equiv \Delta$. The algorithm samples from the following target density at time $t_{n'}$ ($n' \leq n$):

$$\pi^{\delta}_{n'}(\mathbf{X}_{1:n'}, \mathbf{Y}^*_{1:n'}|\mathbf{Y}_{1:n'}) \propto \prod_{j=1}^{n'} J_{j,\delta}(\mathbf{Y}_j,\mathbf{Y}^*_j) f(\mathbf{Y}^*_j|\mathbf{X}_j) p(\mathbf{X}_{j-1}|\mathbf{X}_j)$$

and for example we could take $J_{j,\delta}(\mathbf{y}_j,\mathbf{y}^*_j) = \mathbb{I}_{A_{\delta,\mathbf{y}_j}}(\mathbf{y}^*_j)$ with $A_{\delta,\mathbf{y}_j} = \{\mathbf{y}^*_j;\ \rho(\eta(\mathbf{y}^*_j),\eta(\mathbf{y}_j)) < \delta\}$ as in Jasra et al. [2012] (see section 4.1 for a Gaussian kernel).

The ABC-SMC procedure is set in algorithm 2 with the purpose to propagate forward M simulated states ("particles"). After algorithm 2 is executed, we select a single trajectory by retrospectively looking at the genealogy of the generated particles, as explained below. The quantity ESS is the effective sam-

Algorithm 2 ABC-SMC for filtering

Step 0. Set $j = 1$: for $m = 1, ..., M$ sample $\mathbf{X}_1^{(m)} \sim p(\mathbf{X}_0)$, $\mathbf{Y}_1^{*(m)} \sim f(\cdot|\mathbf{X}_1^{(m)})$, compute weights $W_1^{(m)} = J_{1,\delta}(\mathbf{Y}_1, \mathbf{Y}_1^{*(m)})$ and normalize weights $w_1^{(m)} := W_1^{(m)} / \sum_{m=1}^M W_1^{(m)}$.
Step 1.
if $ESS(\{w_j^{(m)}\}) < \bar{M}$ **then**
resample M particles $\{\mathbf{X}_j^{(m)}, w_j^{(m)}\}$ and set $W_j^{(m)} = 1/M$.
end if
Set $j := j + 1$ and if $j = n + 1$, stop.
Step 2. For $m = 1, ..., M$ sample $\mathbf{X}_j^{(m)} \sim p(\cdot|\mathbf{X}_{j-1}^{(m)})$ and $\mathbf{Y}_j^{*(m)} \sim f(\cdot|\mathbf{X}_j^{(m)})$. Compute

$$W_j^{(m)} := w_{j-1}^{(m)} J_{j,\delta}(\mathbf{Y}_j, \mathbf{Y}_j^{*(m)})$$

normalize weights $w_j^{(m)} := W_j^{(m)} / \sum_{m=1}^M W_j^{(m)}$ and go to step 1.

ple size (e.g. Liu [2008]) often estimated as $ESS(\{w_j^{(m)}\}) = 1/\sum_{m=1}^M (w_j^{(m)})^2$ and taking values between 1 and M: when considering an indicator function for $J_{j,\delta}$ it coincides with the number of particles having positive weight [Jasra et al., 2012]. Under such choice the integer $\bar{M} \leq M$ is a lower bound (threshold set by the experimenter) on the number of particles with non-zero weight. In our experiments we used "stratified resampling" [Kitagawa, 1996] in step 1 of algorithm 2.

In addition to the procedure outlined in algorithm 2, once the set of weights $\{w_n^{(1)}, ..., w_n^{(m)}\}$ is available at the end of ABC-SMC we propose to follow Andrieu et al. [2010] (see their PMMH algorithm) and sample a single index from the set $\{1, ..., M\}$ having associated probabilities $\{w_n^{(1)}, ..., w_n^{(m)}\}$. Denote with m' such index and with a_j^m the "ancestor" of the generic mth particle sampled at time t_{j+1}, with $1 \leq a_j^m \leq M$ ($m = 1, ..., M$, $j = 1, ..., n$). Then we have that particle m' has ancestor $a_{n-1}^{m'}$ and in general particle m'' at time t_{j+1} has ancestor $b_j^{m''} := a_j^{b_{j+1}^{m''}}$, with $b_n^{m'} := m'$. Hence, at the end of algorithm 2 we can sample m' and construct its genealogy: the sequence of states $\{X_t\}$ resulting from the genealogy of m' is the chosen path that will be passed to SAEM, see algorithm 3. Notice that the ABC threshold δ does not need to be fixed by the user but can be set adaptively: when $j = 1$ Jasra et al. [2012] set δ to be the largest difference between $\mathbf{Y}_1$ and the $\mathbf{Y}_1^{*(m)}$ values, while we obtained better results by setting δ to be the αth percentile of $\{\rho(\eta(\mathbf{Y}_1^{*(m)}), \eta(\mathbf{Y}_1)); 1 \leq m \leq M\}$. When $j > 1$ we take δ to be the αth percentile of

$$\{\rho(\eta(\mathbf{Y}_j^{*(m)}), \eta(\mathbf{Y}_j)); 1 \leq m \leq M, W_{j-1}^{(m)} > 0\}.$$

Notice in particular that the latter is computed on distances generated by forward simulation from particles with starting points having strictly positive weights at time t_{j-1}.

3.2 SAEM-ABC using SMC

We propose to integrate ABC-SMC with SAEM, resulting in algorithm 3. SAEM allows also to compute standard errors of the estimators, through the approximation of the Fisher Information matrix. This is detailed below.

Algorithm 3 SAEM-ABC using SMC

Step 0. Set parameters starting values $\hat{\boldsymbol{\theta}}^{(0)}$, then set M, $\bar{M}$ and $k := 1$.
Step 1. For fixed $\hat{\boldsymbol{\theta}}^{(k-1)}$ apply the ABC-SMC algorithm 2 with M particles and particles threshold $\bar{M}$.
2 Sample an index m' from the probability distribution $\{w_n^{(1)}, ..., w_n^{(m)}\}$ on $\{1, ..., M\}$ and form the path $\mathbf{X}^{(k)}$ resulting from the genealogy of m'.
Step 3. **Stochastic Approximation step :** update of the sufficient statistics

$$\mathbf{s}_k = \mathbf{s}_{k-1} + \gamma_k \left(\mathbf{S}_c(\mathbf{Y}, \mathbf{X}^{(k)}) - \mathbf{s}_{k-1} \right)$$

Step 4. **Maximisation step:** update $\boldsymbol{\theta}$

$$\widehat{\boldsymbol{\theta}}^{(k)} = \arg\max_{\boldsymbol{\theta} \in \Theta} \left(-\Lambda(\boldsymbol{\theta}) + \langle \mathbf{s}_k, \Gamma(\boldsymbol{\theta}) \rangle \right)$$

Set $k := k + 1$ and go to step 1.

Notice that for models where the evaluation of the likelihood is expensive, using the ABC-SMC filter has evident computational benefits. Indeed, when forward simulation from the likelihood is faster than evaluating the likelihood, this strategy can be significantly faster than a standard SMC algorithm targeting the exact posterior $\pi_{n'}(\mathbf{X}_{1:n'}|\mathbf{Y}_{1:n'})$. Furthermore, should filtering be the only goal of the analysis (this is not our case) or should estimation of parameters $\boldsymbol{\theta}_x$ be the only goal (assuming $\boldsymbol{\theta}_y$ known), then with ABC-SMC it is possible to treat cases where the observation density $f(\mathbf{y}_t|\cdot)$ is not analytically available, see e.g. the examples in Calvet and Czellar [2014] such as α-stable stochastic volatility processes.

Among other advantages, let us emphasize that the ABC-SMC filter can be useful to alleviate particle degeneracy issues, i.e. the phenomenon where most of the particle weights result equal to zero, either because of model misspecification or because of the occasional outlier in the data or floating-point underflow in the numerical computation of weights. The ability to modify the ABC threshold δ adaptively, as previously discussed, can alleviate this problem: though this will definitely affect the accuracy of the filter, it does not necessarily have to result in a major impact for the resulting parameter estimate. Also, as shown in section 4.1, SAEM-ABC-SMC can have advantages over exact strategies for small datasets.

Fisher Information matrix The standard errors of the parameter estimates can be calculated from the diagonal elements of the inverse of the Fisher information matrix. Its direct evaluation is difficult because it has no explicit

analytic form, however an estimate of the Fisher information matrix can be easily implemented within SAEM as proposed by Delyon et al. [1999] using the Louis' missing information principle [Louis, 1982].

The Hessian of the log-likelihood $\ell(\boldsymbol{\theta}) = L(\mathbf{Y};\boldsymbol{\theta})$ can be expressed as:

$$\begin{aligned}\partial^2_\theta \ell(\boldsymbol{\theta}) &= \mathbb{E}\left[\partial^2_\theta L_c(\mathbf{Y},\mathbf{X};\boldsymbol{\theta})|\mathbf{Y},\boldsymbol{\theta}\right] \\ &+ \mathbb{E}\left[\partial_\theta L_c(\mathbf{Y},\mathbf{X};\boldsymbol{\theta})\ (\partial_\theta L_c(\mathbf{Y},\mathbf{X};\boldsymbol{\theta}))'|\mathbf{Y},\boldsymbol{\theta}\right] \\ &- \mathbb{E}\left[\partial_\theta L_c(\mathbf{Y},\mathbf{X};\boldsymbol{\theta})|\mathbf{Y},\boldsymbol{\theta}\right]\ \mathbb{E}\left[\partial_\theta L_c(\mathbf{Y},\mathbf{X};\boldsymbol{\theta})|\mathbf{Y},\boldsymbol{\theta}\right]'\end{aligned}$$

where $'$ denotes transposition. An on-line estimation of the Hessian is obtained using the stochastic approximation procedure of the SAEM algorithm as follows (see Lavielle [2014] for an off-line approach). At the $(k+1)$th iteration of the algorithm, we evaluate the three following quantities:

$$\begin{aligned}\mathbf{G}_{k+1} &= \mathbf{G}_k + \gamma_k\left[\partial_\theta L_c(\mathbf{Y},\mathbf{X}^{(k)},\boldsymbol{\theta}) - \mathbf{G}_k\right] \\ \mathbf{H}_{k+1} &= \mathbf{H}_k + \gamma_k\Big[\partial^2_\theta L_c(\mathbf{Y},\mathbf{X}^{(k)},\boldsymbol{\theta}) \\ &\qquad + \partial_\theta L_c(\mathbf{Y},\mathbf{X}^{(k)},\boldsymbol{\theta})\ (\partial_\theta L_c(\mathbf{Y},\mathbf{X}^{(k)},\boldsymbol{\theta}))' - \mathbf{H}_k\Big] \\ \mathbf{F}_{k+1} &= \mathbf{H}_{k+1} - \mathbf{G}_{k+1}\ (\mathbf{G}_{k+1})'.\end{aligned}$$

As the sequence $(\hat{\boldsymbol{\theta}}^{(k)})_k$ converges to the maximum of the likelihood, the sequence $(\mathbf{F}_k)_k$ converges to the Fisher information matrix. It is possible to initialize $\mathbf{G}_0$ and $\mathbf{H}_0$ to be a vector and a matrix of zeros respectively.

4 Simulation studies

Simulations were coded in MATLAB and executed on a Intel Core i7-2600 CPU 3.40 GhZ. For all examples we consider $\eta(\cdot)$ to be the identity function and $\rho(\cdot)$ the $\mathbb{L}_1$ norm when using the indicator function for $J_{j,\delta}$. In SAEM we always set $\gamma_k = 1$ for the first $K_1 < K$ iterations and $\gamma_k = (K-K_1)^{-1}$ for the remaining iterations as in Lavielle [2014]. All results involving ABC are produced using algorithm 3 i.e. using trajectories selected via ABC-SMC.

4.1 Non-linear Gaussian state-space model

Here we study a simple non-linear model, using a setup similar to Jasra et al. [2012]. We have

$$\begin{cases} Y_j = X_j + \sigma_y \nu_j \\ X_j = 2\sin(e^{X_{j-1}}) + \sigma_x \tau_j, \quad j \geq 1 \end{cases} \tag{4}$$

with $\nu_j, \tau_j \sim N(0,1)$ i.i.d. and $X_0 = 0$. We wish to estimate $\sigma_x, \sigma_y > 0$. We generate $n = 200$ observations for $\{Y_j\}$ with $\sigma^2_x = \sigma^2_y = 5$ and conduct inference for $\boldsymbol{\theta} = (\sigma^2_x, \sigma^2_y)$.

We first construct the set of sufficient statistics corresponding to the complete log-likelihood $L_c(\mathbf{Y}, \mathbf{X})$. This is a very simple task since $Y_j|X_j \sim N(X_j, \sigma_y^2)$ and $X_j|X_{j-1} \sim N(2\sin(e^{X_{j-1}}), \sigma_x^2)$ and therefore it is easy to show that $S_{\sigma_x^2} = \sum_{j=1}^{n}(X_j - 2\sin(e^{X_{j-1}}))^2$ and $S_{\sigma_y^2} = \sum_{j=1}^{n}(Y_j - X_j)^2$ are sufficient for σ_x^2 and σ_y^2 respectively. By plugging these statistics into $L_c(\mathbf{Y}, \mathbf{X})$ and equating to zero the gradient of L_c with respect to (σ_x^2, σ_y^2), we find that the M-step of SAEM results in updated values for σ_x^2 and σ_y^2 given by $S_{\sigma_x^2}/n$ and $S_{\sigma_y^2}/n$ respectively. Derivation of the second and mixed derivatives, useful to obtain the Fisher information as in 3.2, is also trivial and expressions are not reported for brevity.

In the following, we refer to Algorithm 3 as the SAEM-ABC algorithm and to SAEM-SMC when a non-ABC sampler is used. We executed $K = 1,000$ iterations of SAEM-ABC and used an indicator function for $J_{j,\delta}$ as defined in section 3.1. We used $M = 5,000$ particles, $\bar{M} = 50$ and $\alpha = 10$ to determine the ABC threshold δ. This induced a resampling step every third observation corresponding to an ESS equal to 50 at the last time point. By denoting with $K_1 = 300$ the number of warmup iterations (i.e. when $k \leq K_1$ we have $\gamma_k = 1$), we used $\gamma_k = (k - K_1)^{-1}$ for $k > K_1$. SAEM-ABC took about 140 seconds of computation. The evolution of the optimization is in Figure 1 where we set on purpose exaggeratedly large starting values for the parameters to show how rapidly the algorithm approaches convergence. If we force the algorithm to resample at each observation, we obtain much worse results. Basically the additional variance introduced by frequent resampling steps is detrimental here, perhaps because the importance weights are already nearly equal and resampling is actually reducing the number of distinct particles (see section 7.3.2 in Cappé et al. [2005]). In Figure 2 we compare data with the selected $\mathbf{X}^{(k)}$ corresponding to the last iteration of SAEM-ABC. By looking at Figure 1 we can appreciate that it took about ten iterations of SAEM-ABC to approach desirable values, that is only 1.5 seconds of computation were required for the initial ten iterations. However, unfortunately it seemed necessary to let the algorithm run for many more iterations in order to obtain stable values for the standard errors, meaning that stopping the computations when convergence is visually apparent might produce a not-positive defined Fisher information. In practice we started computing the quantities $(\mathbf{G}_k, \mathbf{H}_k, \mathbf{F}_k)$ from section 3.2 for $k > 50$ and a "long enough" warmup period K_1 was needed to update those values (which are initialised to contain zeros when the algorithm starts), but then it was required a further stretch in the execution before it returned satisfactory standard errors.

We now compare our results with maximum likelihood inference obtained by combining SAEM with non-ABC sequential Monte Carlo, that is with the bootstrap filter with trajectory selection as in the PMMH algorithm by Andrieu et al. [2010]. Basically the strategy is the same as in algorithm 3 except that we use an SMC step where in algorithm 2 we have $f(Y_j|X_j^{(m)})$ in place of $J_{j,\delta}(Y_j, Y_j^{*(m)})$ and there is no need to simulate the $Y_j^{*(m)}$. We refer to this algorithm as SAEM-SMC.

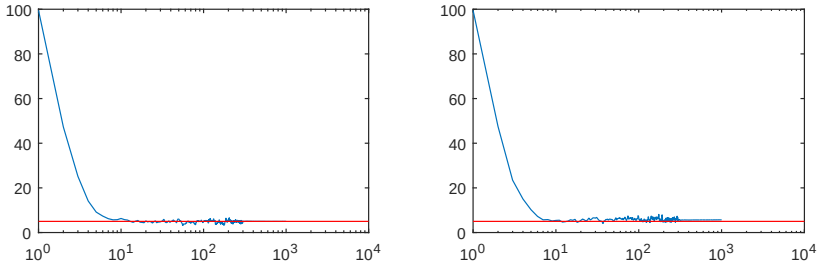

Fig. 1: Non-linear Gaussian state-space model: evolution of the estimators for σ_x^2 (left) and σ_y^2 (right) along $K = 1,000$ iterations of SAEM-ABC. Horizontal lines are the true parameter values $\sigma_x^2 = \sigma_y^2 = 5$. Abscissa values are on log-scale.

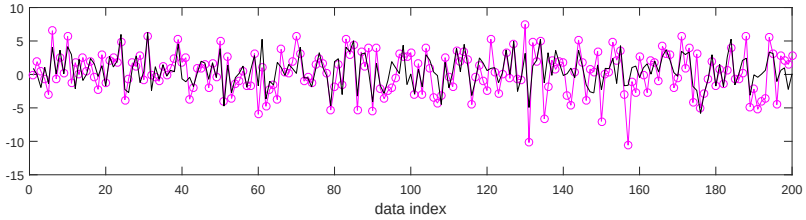

Fig. 2: Non-linear Gaussian state-space model: true hidden trajectory X (circles connected with lines) and the selected trajectory corresponding to the last iteration of SAEM-ABC.

To compare the two algorithms, we consider a parametric bootstrap experiment of size 100, that is we simulate independently 100 datasets with parameters $\sigma_x^2 = \sigma_y^2 = 5$, and for each dataset we apply SAEM-ABC and SAEM-SMC. To study the influence of the kernel choice $J_{j,\delta}$, this time SAEM-ABC is implemented with a Gaussian kernel

$$J_{j,\delta}(Y_j, Y_j^{*(m)}) \propto \frac{1}{\delta} e^{-(Y_j^{*(m)} - Y_j)^2/(2\delta^2)} \tag{5}$$

so that weights $W_j^{(m)}$ are larger for particles having $Y_j^{*(m)} \approx Y_j$.

The bootstrap experiment is repeated for different sample sizes $n = 20$, 50 and 200. Using bootstrap to estimate variances for the sampling distributions implies that we can avoid the previous problems with a not-positive defined Fisher matrix and thus perform shorter simulations, using $K = 200$ and $K_1 = 100$. Starting values for the optimization are always $\sigma_x^2 = \sigma_y^2 = 100$. For ease of comparison, we report results for (σ_x, σ_y) instead of (σ_x^2, σ_y^2). Same as before, for SAEM-ABC and for a given value of j we take α to be a percentile for the distances $\{|Y_j^{*(m)} - Y_j|, m = 1, ..., M\}$: we set $\alpha = 20$ to select the initial threshold δ (at $j = 1$) and then $\alpha = 3$ for the remaining observations ($j > 1$). Means and standard errors for the resulting sampling distributions are in Table 1, but see the discussion below and Figure 4. We notice that bootstrap means for SAEM-ABC are very stable, with a minimal increasing in variability

n	20	50	200
σ_x (true value 2.23)			
SAEM-ABC	2.11 (0.37)	2.12 (0.27)	2.13 (0.12)
SAEM-SMC	1.92 (1.32)	2.61 (0.57)	2.61 (0.29)
σ_y (true value 2.23)			
SAEM-ABC	2.60 (0.63)	2.62 (0.45)	2.69 (0.20)
SAEM-SMC	1.90 (1.26)	1.70 (0.70)	1.92 (0.39)

Table 1: Non-linear Gaussian model: means and standard errors for 100 bootstrap replications using SAEM-ABC and SAEM-SMC using different sample sizes n.

for decreasing n. For SAEM-SMC we instead observe a marked increase in variability for decreasing n. What we noticed during our simulations was a much smaller ESS for SAEM-ABC than for SAEM-SMC, which is expected as in general the ABC approach relies on setting a small threshold $\delta > 0$, which necessarily "kills" particles. With SAEM-ABC particles will have weights appreciably larger than zero only when very close to observations, because of the small δ (e.g. ESS is about 10 at t_n when $n = 20$). When $n = 20$ SAEM-SMC produces an ESS of about 1,000 which gives a larger freedom to select a trajectory which might wander far away from the (few) data points, this producing several biased estimates in the bootstrap sample, see Figure 4. These results can have interesting implications for data-poor scenarios, e.g. in pharmacokinetics/pharmacodynamics studies (see Lavielle [2014]). However in the next example the variance of the residual error is much smaller and we do not observe a marked difference between SAEM-ABC and SAEM-SMC.

4.2 A pharmacokinetics model

Here we consider a model for pharmacokinetics dynamics. For example we may imagine to formulate a model to study the Theophylline drug pharmacokinetics. This example has often been described in literature devoted to longitudinal data modelling with random parameters (mixed–effects models), see Pinheiro and Bates [1995] and Donnet and Samson [2008]. Same as in Picchini [2014] here we do not consider a mixed–effects model. We denote with X_t the level of Theophylline drug concentration in blood at time t (hrs). Consider the following non-authonomous stochastic differential equation:

$$dX_t = \left(\frac{Dose \cdot K_a \cdot K_e}{Cl} e^{-K_a t} - K_e X_t \right) dt + \sigma \sqrt{X_t} dW_t, \qquad t \geq t_0 \qquad (6)$$

where $Dose$ is the known drug oral dose received by a subject, K_e is the elimination rate constant, K_a the absorption rate constant, Cl the clearance of the drug and σ the intensity of intrinsic stochastic noise. We simulate data measured at $n = 30$ equispaced sampling times $\{t_1, t_\Delta, ..., t_{30\Delta}\} = \{1, 2, ..., 30\}$ where $\Delta = t_j - t_{j-1} = 1$. The drug oral dose is chosen to be 4 mg. After the drug is administered, we consider as $t_0 = 0$ the time when the concentration first reaches $X_{t_0} = X_0 = 8$. The error model is assumed to be linear, $Y_j =$

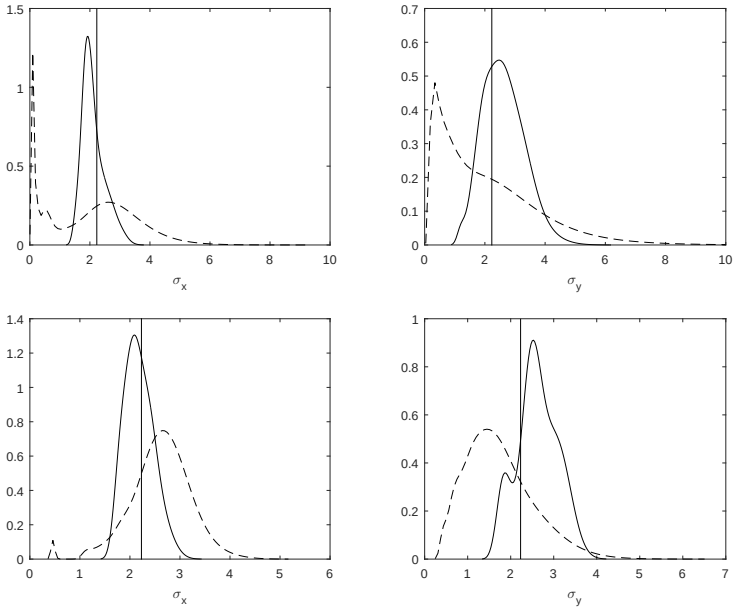

Fig. 4: Gaussian model: (top) sampling distributions (by kernel smoothing) from parametric bootstrap when $n = 20$ for σ_x (left) and σ_y (right). Distributions using SAEM-ABC have solid lines, while SAEM-SMC have dashed lines. Vertical lines are true parameter values. (bottom) Same as above, with $n = 50$.

$X_j + \varepsilon_j$ where the $\varepsilon_j \sim N(0, \sigma_\varepsilon^2)$ are i.i.d., $j = 1, ..., 30$. Inference is based on data $\{Y_1, ..., Y_{30}\}$ collected at corresponding sampling times. Parameter K_a is assumed known, hence parameters of interest are $\boldsymbol{\theta} = (K_e, Cl, \sigma^2, \sigma_\varepsilon^2)$ as X_0 is also assumed known.

Equation (6) has no available closed-form solution, hence simulated data are created in the following way. We first simulate numerically a solution to (6) using the Euler–Maruyama discretization with stepsize $h = 0.05$ on the time interval $[t_0, 30]$ and

$$X_{t+h} = X_t + \left(\frac{Dose \cdot K_a \cdot K_e}{Cl} e^{-K_a t} - K_e X_t \right) h + (\sigma \sqrt{h \cdot X_t}) Z_{t+h}$$

where the $\{Z_t\}$ are i.i.d. $N(0, h)$ distributed. The grid of generated values $\mathbf{X}_{0:N}$ is then linearly interpolated at sampling times $\{t_1, ..., t_{30}\}$ to give $\mathbf{X}_{1:n}$, and finally residual error is added to $\mathbf{X}_{1:n}$ according to the error model $Y_j = X_j + \varepsilon_j$ as explained above. Data $\{Y_j\}$ are conditionally independent given the latent process $\{X_t\}$ and are generated with $(K_e, K_a, Cl, \sigma^2, \sigma_\varepsilon^2) = (0.05, 1.492, 0.04, 0.01, 0.102)$.

Sufficient statistics for SAEM The complete likelihood is given by

$$p(\mathbf{Y}, \mathbf{X}_{0:N}; \boldsymbol{\theta}) = p(\mathbf{Y}|\mathbf{X}_{0:N}; \boldsymbol{\theta}) p(\mathbf{X}_{0:N}; \boldsymbol{\theta}) = \prod_{j=1}^{n} p(Y_j|X_j; \boldsymbol{\theta}) \prod_{i=1}^{N} p(X_i|X_{i-1}; \boldsymbol{\theta})$$

where the unconditional density $p(x_0)$ is disregarded in the last product since we assume X_0 deterministic. Hence the complete-data loglikelihood is

$$L_c(\mathbf{Y}, \mathbf{X}_{0:N}; \boldsymbol{\theta}) = \sum_{j=1}^{n} \log p(Y_j|X_j; \boldsymbol{\theta}) + \sum_{i=1}^{N} \log p(X_i|X_{i-1}; \boldsymbol{\theta}).$$

Here $p(y_j|x_j; \boldsymbol{\theta})$ is a Gaussian with mean x_j and variance σ_ε^2. The transition density $p(x_i|x_{i-1}; \theta)$ is not known for this problem, hence we approximate it with the Gaussian density induced by the Euler-Maruyama scheme, that is

$$p(x_i|x_{i-1}; \boldsymbol{\theta}) \approx \frac{1}{\sigma\sqrt{2\pi x_{i-1} h}} \exp\left\{ - \frac{\left[x_i - x_{i-1} - (\frac{Dose \cdot K_a \cdot K_e}{Cl} e^{-K_a \tau_{i-1}} - K_e x_{i-1})h\right]^2}{2\sigma^2 x_{i-1} h} \right\}.$$

We now wish to derive sufficient summary statistics for the parameters of interest, based on the complete loglikelihood. Regarding σ_ε^2 this is trivial as we only have to consider $\sum_{j=1}^{n} \log p(y_j|x_j; \theta)$ to find that a sufficient statistic is $S_{\sigma_\varepsilon^2} = \sum_{j=1}^{n} (y_j - x_j)^2$. Regarding the remaining parameters we have to consider $\sum_{i=1}^{N} \log p(x_i|x_{i-1}; \boldsymbol{\theta})$. For σ^2 it is clear that a sufficient statistic is

$$S_{\sigma^2} = \sum_{i=1}^{N} \left(\frac{\left[x_i - x_{i-1} - (\frac{Dose \cdot K_a \cdot K_e}{Cl} e^{-K_a \tau_{i-1}} - K_e x_{i-1})h\right]^2}{x_{i-1} h} \right).$$

Regarding K_e and Cl things are a bit more complicated: we can write

$$\begin{aligned} \sum_{i=1}^{N} \log p(x_i|x_{i-1}; \boldsymbol{\theta}) &\propto \sum_{i=1}^{N} \frac{\left[x_i - x_{i-1} - (\frac{Dose \cdot K_a \cdot K_e}{Cl} e^{-K_a \tau_{i-1}} - K_e x_{i-1})h\right]^2}{x_{i-1}} \\ &= \sum_{i=1}^{N} \left[\frac{x_i - x_{i-1}}{\sqrt{x_{i-1}}} - \left(\frac{Dose \cdot K_a \cdot K_e}{Cl\sqrt{x_{i-1}}} e^{-K_a \tau_{i-1}} - \frac{K_e x_{i-1}}{\sqrt{x_{i-1}}} \right) h \right]^2. \end{aligned}$$

The last equality suggests a linear regression approach $E(V) = \beta_1 C_1 + \beta_2 C_2$ for "responses" $V_i = (x_i - x_{i-1})/\sqrt{x_{i-1}}$ and "covariates"

$$\begin{aligned} C_{i1} &= \frac{Dose \cdot K_a e^{-K_a \tau_{i-1}} h}{\sqrt{x_{i-1}}} \\ C_{i2} &= -\frac{x_{i-1}}{\sqrt{x_{i-1}}} h = -\sqrt{x_{i-1}} h \end{aligned}$$

and $\beta_1 = K_e/Cl$, $\beta_2 = K_e$. By considering the design matrix $\mathbf{C}$ with columns $\mathbf{C}_1$ and $\mathbf{C}_2$, that is $\mathbf{C} = [\mathbf{C}_1, \mathbf{C}_2]$, from standard regression theory we have that $\hat{\boldsymbol{\beta}} = (\mathbf{C}'\mathbf{C})^{-1}\mathbf{C}'\mathbf{V}$ is a sufficient statistic for $\boldsymbol{\beta} = (\beta_1, \beta_2)$, where $'$ denotes transposition. We take $S_{K_e} := \hat{\beta}_2$ also to be used as the updated value of K_e in the maximisations step of SAEM. Then we have that $\hat{\beta}_1$ is sufficient for the ratio K_e/Cl and use $\hat{\beta}_2/\hat{\beta}_1$ as the update of Cl in the M-step of SAEM. The updated values of σ and σ_ε are given by $\sqrt{S_{\sigma^2}/N}$ and $\sqrt{S_{\sigma_\varepsilon^2}/n}$ respectively.

Fisher Information matrix To compute the Fisher Information matrix as suggested in section 3.2 we need to differentiate the complete data log-likelihood with respect to the four parameters $\boldsymbol{\theta} = (K_e, Cl, \sigma^2, \sigma_\varepsilon^2)$. We differentiate w.r.t. $(\sigma^2, \sigma_\varepsilon^2)$ instead of $(\sigma, \sigma_\varepsilon)$ because the complete log-likelihood is expressed as a function of sufficient statistics for $(\sigma^2, \sigma_\varepsilon^2)$.

In the following we set formulas for the computation of gradient and the Hessian matrix. Set, for $i = 1, \ldots, N$,

$$z_i(\theta) = x_i - x_{i-1} - h(Dose \cdot K_a \cdot \frac{K_e}{Cl} \cdot e^{-K_a \tau_{i-1}} - K_e \cdot x_{i-1}).$$

The four coordinates of the gradient are:

$$\begin{aligned}
\frac{\partial}{\partial K_e} L_c(\mathbf{Y}, \mathbf{X}; \boldsymbol{\theta}) &= -\frac{1}{\sigma^2} \sum_{i=1}^{N} \frac{z_i(\boldsymbol{\theta})}{x_{i-1}} \left(x_{i-1} - \frac{Dose \cdot K_a}{Cl} e^{-K_a \tau_{i-1}} \right) \\
\frac{\partial}{\partial Cl} L_c(\mathbf{Y}, \mathbf{X}; \boldsymbol{\theta}) &= -\frac{1}{\sigma^2} \sum_{i=1}^{N} \frac{z_i(\boldsymbol{\theta})}{x_{i-1}} \left(\frac{Dose \cdot K_a \cdot K_e}{Cl^2} e^{-K_a \tau_{i-1}} \right) \\
\frac{\partial}{\partial \sigma^2} L_c(\mathbf{Y}, \mathbf{X}; \boldsymbol{\theta}) &= -\frac{N}{2\sigma^2} + \frac{1}{2h\sigma^4} \sum_{i=1}^{N} \frac{z_i(\theta)^2}{x_{i-1}} \\
\frac{\partial}{\partial \sigma_\varepsilon^2} L_c(\mathbf{Y}, \mathbf{X}; \theta) &= -\frac{n}{2\sigma_\varepsilon^2} + \frac{1}{2\sigma_\varepsilon^4} \sum_{j=1}^{n} (y_j - x_j)^2.
\end{aligned}$$

Entries for the Hessian matrix are (recall that the Hessian is a symmetric matrix, therefore redundant terms are not reported. Further missing entries consist of zeros):

$$\begin{aligned}
\frac{\partial^2}{\partial^2 K_e} L_c(\mathbf{Y}, \mathbf{X}; \boldsymbol{\theta}) &= -\frac{h}{\sigma^2} \sum_{i=1}^{N} (x_{i-1} - Dose \cdot \frac{K_a}{Cl} \cdot e^{-K_a \tau_{i-1}})^2 \frac{1}{x_{i-1}} \\
\frac{\partial^2}{\partial^2 Cl} L_c(\mathbf{Y}, \mathbf{X}; \boldsymbol{\theta}) &= -\frac{1}{\sigma^2} \sum_{i=1}^{N} \left\{ \frac{1}{x_{i-1}} \left[\frac{Dose \cdot K_a \cdot K_e}{Cl^2} e^{-K_a \tau_{i-1}} \left(h - \frac{2 z_i(\boldsymbol{\theta})}{Cl} \right) \right] \right\} \\
\frac{\partial^2}{\partial^2 \sigma^2} L_c(\mathbf{Y}, \mathbf{X}; \boldsymbol{\theta}) &= \frac{N}{2\sigma^4} - \frac{1}{h\sigma^6} \sum_{i=1}^{N} \frac{z_i(\boldsymbol{\theta})^2}{x_{i-1}} \\
\frac{\partial^2}{\partial^2 \sigma_\varepsilon^2} L_c(\mathbf{Y}, \mathbf{X}; \boldsymbol{\theta}) &= \frac{n}{2\sigma_\varepsilon^4} - \frac{1}{\sigma_\varepsilon^6} \sum_{j=1}^{n} (y_j - x_j)^2
\end{aligned}$$

	K_e	Cl	σ	σ_ε
true values	0.050	0.040	0.100	0.319
SAEM-ABC	0.053 (0.007)	0.073 (0.011)	0.153 (0.030)	0.329 (0.118)
SAEM-SMC	0.054 (0.008)	0.070 (0.012)	0.118 (0.024)	0.400 (0.110)

Table 2: Theophylline: means and standard errors for 100 bootstrap replications using SAEM-ABC and SAEM-SMC.

$$\frac{\partial^2}{\partial K_e \partial Cl} L_c(\mathbf{Y}, \mathbf{X}; \boldsymbol{\theta}) = -\frac{1}{\sigma^2} \sum_{i=1}^{N} \frac{1}{x_{i-1}} \left\{ \frac{Dose \cdot K_a}{Cl^2} e^{-K_a \tau_{i-1}} \left[h \cdot K_e \left(x_{i-1} - \frac{Dose \cdot K_a}{Cl} e^{-K_a \tau_{i-1}} \right) + z_i(\boldsymbol{\theta}) \right] \right\}$$

$$\frac{\partial^2}{\partial \sigma^2 \partial K_e} L_c(\mathbf{Y}, \mathbf{X}; \boldsymbol{\theta}) = \frac{1}{\sigma^4} \sum_{i=1}^{N} \frac{z_i(\boldsymbol{\theta})}{x_{i-1}} \left(x_{i-1} - \frac{Dose \cdot K_a}{Cl} e^{-K_a \tau_{i-1}} \right)$$

$$\frac{\partial^2}{\partial \sigma^2 \partial Cl} L_c(\mathbf{Y}, \mathbf{X}; \boldsymbol{\theta}) = \frac{1}{\sigma^4} \sum_{i=1}^{N} \frac{z_i(\boldsymbol{\theta})}{x_{i-1}} \left(\frac{Dose \cdot K_a \cdot K_e}{Cl^2} e^{-K_a \tau_{i-1}} \right).$$

Results Recall the setup given at the beginning of section 4.2, and in particular the parameter values used to generate data $(K_e, K_a, Cl, \sigma, \sigma_\varepsilon) = (0.05, 1.492, 0.04, 0.1, 0.319)$. In this section, all results pertaining SAEM-ABC use the Gaussian kernel (5). For illustration purposes, we initially consider a single long simulation, using $M = 10,000$ particles with $K = 1,500$ and warmup $K_1 = 300$, see Figure 5, which is completed in 270 seconds. It requires about ten iterations to approach the true parameter values, even though we let K_e and Cl start from very unlikely values. Data and the selected trajectory $\{X_t\}$ from the last iteration of SAEM-ABC are in Figure 6. Then, same as in section 4.1, for both SAEM-ABC and SAEM-SMC we run a bootstrap simulation of size 100 where each data set is produced using the true parameter values. For SAEM-ABC we determined the threshold δ using $\alpha = 2.5$ for the first sampling time t_1 and $\alpha = 1$ afterwards, with the additional condition of forcing $\delta := \tilde{\delta}$ when the δ determined by the percentile α was larger than $\tilde{\delta} = 0.003$. Therefore, by comparing such $\tilde{\delta}$ with the typical scale of simulated data (of which Figure 6 is an example) we can tell that we are forcing selected trajectories to lie very close to data. We report the values for the SAEM-ABC estimates of $(\sigma, \sigma_\varepsilon)$ instead of $(\sigma^2, \sigma_\varepsilon^2)$ not to unnecessarily magnify discrepancies with the true values. Since standard errors are computed from the sampling distribution of the 100 estimates, we do not need to run long SAEM simulations and instead set $K = 80$ and $K_1 = 50$, see Table 2 and Figure 7 for the results. In summary, the results are qualitatively similar and the approximation error induced by ABC is negligible.

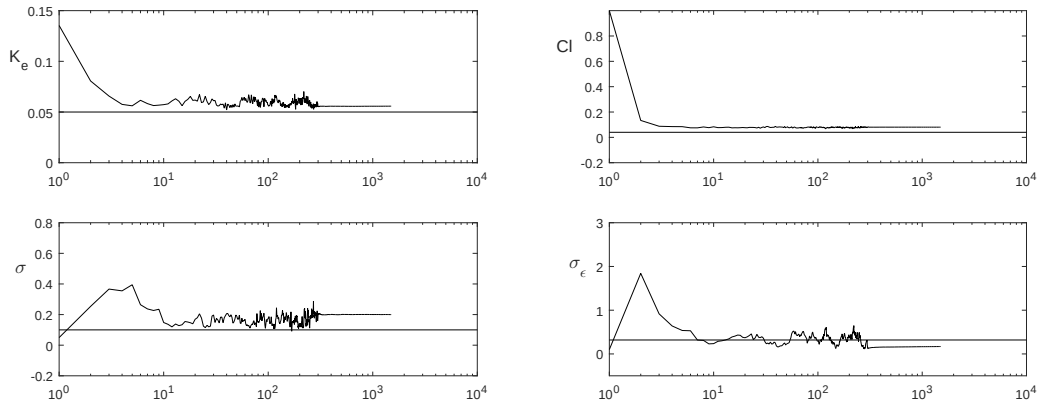

Fig. 5: Theophylline model: $K = 1,500$ iterations of SAEM-ABC. Top: K_e (left) and Cl (right). Bottom: σ (left) and σ_ϵ (right). Horizontal lines are the true parameter values. Abscissa values are on log-scale.

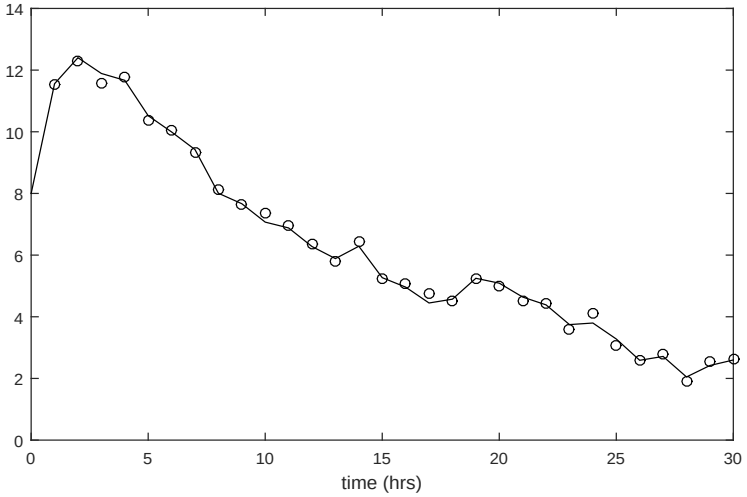

Fig. 6: Theophylline model: data (circles) and the selected trajectory $\{X_t\}$ corresponding to the last iteration of SAEM-ABC.

4.3 Stochastic volatility model

Here we consider the following stochastic volatility model

$$\begin{cases} Z_j = \beta \exp(X_j/2)\nu_j \\ X_j = \alpha X_{j-1} + \sigma\tau_j, \quad j \geq 1 \end{cases} \tag{7}$$

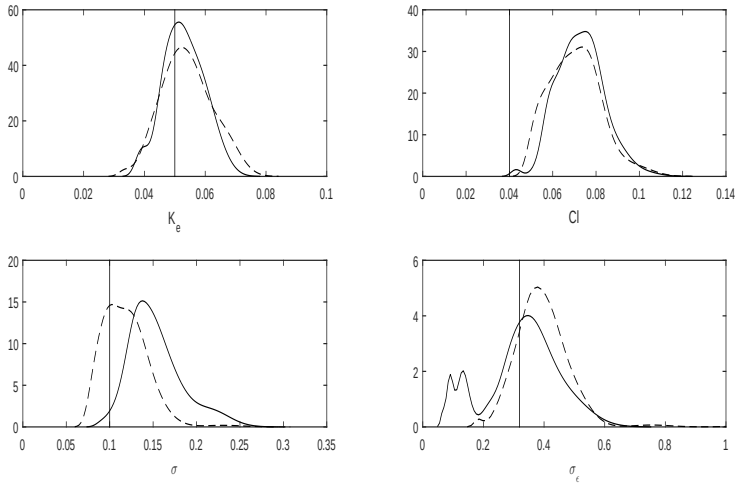

Fig. 7: Theophylline: sampling distributions (by kernel smoothing) from parametric bootstrap using SAEM-ABC (solid lines) and SAEM-SMC (dashed lines). Vertical lines are true parameter values.

with $\nu_j \sim N(0,1)$ i.i.d. and $\tau_j \sim N(0,1)$ i.i.d. where the Z_j can be observed while the X_j are latent (volatility). Here $\boldsymbol{\theta} = (\alpha, \beta, \sigma)$. This type of model, and its generalisations, have been very widely used in various areas of economics and mathematical finance: inferring and predicting underlying volatility from observed price or rate data is an important problem. For such a model $p(Z_j|X_j)$ is $N(0, \beta^2 \exp(X_j))$ and $p(X_j|X_{j-1})$ is $N(\alpha X_{j-1}, \sigma^2)$ hence it is easy to write the complete likelihood, and is also easy to determine the sufficient statistics. However the model is challenging, as unlike in the previously discussed examples where the variance of $Z_j|X_j$ is constant (albeit unknown), here the conditional variance of the observations $\beta^2 \exp(X_j)$ is dependent on the process itself. Therefore a less than optimal filtered value for the state X_j will result in a drastic change in the estimated variability of $Z_j|X_j$, meaning that in ABC simulated values Z_j^* might lie very far from actual observations. Section 1.3.5 in Cappé et al. [2005] suggests that working with the following, equivalent, representation can be numerically convenient

$$\begin{cases} Y_j = \log \beta^2 + X_j + \epsilon_j \\ X_j = \alpha X_{j-1} + \sigma \tau_j \end{cases} \tag{8}$$

where $Y_j := \log Z_j^2$ and the ϵ_j are i.i.d. realizations from a $\log \chi_1^2$ distribution. We employ model (8) in our experiments. Notice that generating a draw from a $\log \chi_1^2$ distribution is trivial, by simply generating from the $N(0,1)$ distribution then squaring the result and finally taking the natural logarithm. We have that

$$p(y_j|x_j) = \frac{1}{2^{1/2}\Gamma(\frac{1}{2})} \cdot \frac{1}{\beta} \exp\left(\frac{y_j - x_j}{2}\right) e^{-\frac{\exp(y_j - x_j)}{2\beta^2}}$$

and we can easily deduce sufficient statistics for the parameters as shown in the previous examples. Regarding α a sufficient statistic is given by the least squares estimate of the "regression through the origin" model $E(X_j) = \alpha X_{j-1}$ (here E denotes expectation), hence $S_\alpha = \sum_j (X_{j-1}X_j)/\sum_j X_{j-1}^2$ which also results in the updated value of α in the M-step. Then we have $S_{\beta^2} = \sum_j \exp(Y_j - X_j)$ and $S_{\sigma^2} = \sum_j (X_j - \alpha X_{j-1})^2$ and the M-step results in updated values $\sqrt{S_{\beta^2}/n}$ and $\sqrt{S_{\sigma^2}/n}$ respectively for β and σ, where n is the sample size of the measurements.

Stochastic volatility models (SVM) are often used in finance, where abundance of data is usually not an issue, at least compared to other areas, that is inference studies often consider results based on, say, $n = 500$ measurements (Doucet and Johansen [2011], Calvet and Czellar [2014]). However there is no reason why a SVM should not be considered in other application scenarios, with a smaller n. As we remarked above, inference here is challenging and in Figure 9 we show a comparison between SAEM-ABC and SAEM-SMC based on one hundred simulations for three different sample sizes $n = 50$, 100 and 200. All simulations assume a deterministic initial condition $X_0 = 0$ and true parameter values $(\alpha, \beta, \sigma) = (0.9, 1, 1)$. All optimizations started at parameter values (0.7,0.7,0.3). For each value of n, Figure 9 shows root mean squared errors (RMSE) computed over the 100 repetitions. Each simulation was executed with $M = 10,000$ particles and SAEM iterations $K = 140$ and $K_1 = 120$. For SAEM-ABC we determined the threshold δ by considering the 0.01th percentile of distances together with kernel (5). The RMSE shows how challenging the model is and also that overall the ABC approach is to be preferred in this case: by setting a very small δ we force the ABC-SMC filter to return a trajectory corresponding to particles producing Y_j^* values very close to data. However the figure also shows an increasing RMSE for increasing n, especially for SAEM-ABC: we can conjecture that using such a small δ can deteriorate the quality of the inference by assigning negligible weights to most particles as time increases (particle degeneracy). Perhaps we should set a larger δ as n increases. However, we also argue that since we are not using particle weights to approximate the likelihood function, in principle for both SAEM-ABC and SAEM-SMC we should not be particularly concerned over issues of particle degeneracy as we only require a single plausible path to be returned, so that SAEM can compute the required sufficient statistics and update parameters. This aspect should be investigated further, however we note that when using a large number of particles it can happen that, even for a "wrong" value of the current θ, a plausible trajectory can still be produced when, for example, an unlikely realization (outlier) for ϵ_j or τ_j is generated, such that Y_j^* is close to Y_j. Of course this is not reassuring as we are interested in inference for θ, however it highlights the challenge posed by this specific model.

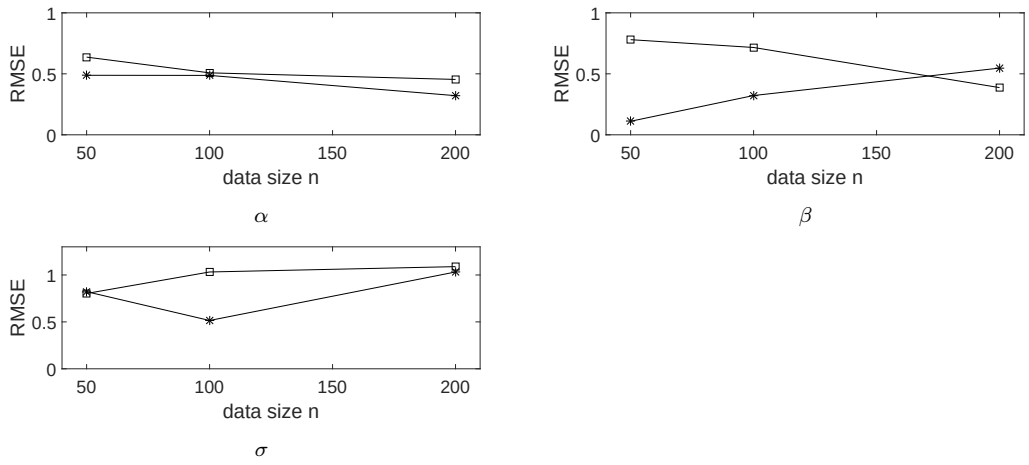

Fig. 9: Stochastic volatility model: RMSE using SAEM-ABC (asterisks) and SAEMC-SMC (squares) for sample sizes $n = 50$, 100 and 200. Each RMSE is computed over one hundred simulations.

5 Summary

We have introduced a new method for approximate maximum likelihood estimation in state-space models. The method uses a stochastic approximation of the EM algorithm, named SAEM (Delyon et al. [1999]), to maximize the likelihood function and here we use an approximate Bayesian computation (ABC) strategy to simulate proposals for the latent states. To the best of our knowledge this is the first combination of ABC with SAEM. SAEM requires model-specific preliminary analytic computations, at the very least the derivation of sufficient statistics for the complete log-likelihood. We used the ABC-SMC algorithm by Jasra et al. [2012] within SAEM: an advantage of using this filter is its flexibility, as it is possible to modify its setup to influence the weighting of the particles. In other words we automatically determine a positive tolerance δ favouring only those particles very close to observations. This of course introduces some bias in the parameter estimation, but it didn't prevent SAEM-ABC to approach reasonable parameter values in about ten iterations for two of the considered examples, even when starting from very unlikely values (Figure 1 and 5). Furthermore, in section 4.1 it was shown a very good behaviour of SAEM-ABC for small datasets ($n = 20$ and 50) as opposed to a non-approximative strategy, the latter showing limitations in identifying the parameters in presence of large residual variability. Good behaviour for a small n ($n = 50$, 100) was also shown for the more challenging example in section 4.3. Moreover, when an indicator function is used for the distance $\rho(y^*, y)$ the values of the unnormalised particle weights are either 0 or 1, hence we won't have problems of numerical underflow/overflow which are typical in sequential Monte Carlo experiments.

Given the investment in terms of preparatory calculations for SAEM-ABC to be performed, we believe the best use of this methodology is to be found when standard models are to be fitted routinely, that is models from the typical toolbox in a given application area. Should instead the research work at hand require exploration/construction of several candidate models, plug-and-play Bayesian methods such as pseudo-marginal methods (Andrieu and Roberts [2009], Andrieu et al. [2010]) or ABC embedded within pseudo-marginal algorithms (see Jasra [2015] for a review) could offer a more immediate support, though these do not result in maximum likelihood inference.

Acknowledgements Umberto Picchini's research was partly funded by the Swedish Research Council (VR grant 2013-5167).

References

C. Andrieu and G. Roberts. The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics*, pages 697–725, 2009.

C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo methods (with discussion). *Journal of the Royal Statistical Society: Series B*, 72(3):269–342, 2010.

L. Calvet and V. Czellar. Accurate methods for approximate Bayesian computation filtering. *Journal of Financial Econometrics*, 2014. doi: 10.1093/jjfinec/nbu019.

O. Cappé, E. Moulines, and T. Rydén. *Inference in hidden Markov models*. Springer, 2005.

O Cappe, SJ Godsill, and E Moulines. An overview of existing methods and recent advances in sequential monte carlo. *Proceedings of the IEEE*, 95: 899–924, 2007.

B. Delyon, M. Lavielle, and E. Moulines. Convergence of a stochastic approximation version of the EM algorithm. *Annals of Statistics*, pages 94–128, 1999.

A.P. Dempster, N. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

S. Ditlevsen and A. Samson. Estimation in the partially observed stochastic morris-lecar neuronal model with particle filter and stochastic approximation methods. *Annals of Applied Statistics*, 2:674–702, 2014.

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

A. Doucet and A. Johansen. *Oxford Handbook of Nonlinear Filtering*, chapter A Tutorial on Particle Filtering and Smoothing: Fifteen years later. Oxford University Press, 2011.

A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo methods in practice*. Springer Science & Business Media, 2001.

Q. Huys, M.B. Ahrens, and L. Paninski. Efficient estimation of detailed single-neuron models. *J Neurophysiol*, 96(2):872–890, 2006.

Quentin J. M. Huys and Liam Paninski. Smoothing of, and Parameter Estimation from, Noisy Biophysical Recordings. *PLOS Computational Biology*, 5(5), 2009.

P. Jacob. Sequential Bayesian inference for implicit hidden Markov models and current limitations. *ESAIM: Proceedings and Surveys*, 51, 2015.

A. Jasra. Approximate Bayesian computation for a class of time series models. *International Statistical Review*, 2015. doi: 10.1111/insr.12089.

A. Jasra, S. Singh, J. Martin, and E. McCoy. Filtering via approximate Bayesian computation. *Statistics and Computing*, 22(6):1223–1237, 2012.

G. Kitagawa. Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical statistics*, 5 (1):1–25, 1996.

M. Lavielle. *Mixed effects models for the population approach: models, tasks, methods and tools*. CRC Press, 2014.

F Lindsten. An efficient stochastic approximation EM algorithm using conditional particle filters. *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 6274 – 6278, 2013.

J. Liu. *Monte Carlo strategies in scientific computing*. Springer Science & Business Media, 2008.

T. Louis. Finding the observed information matrix when using the EM algorithm. *Journal of the Royal Statistical Society. Series B*, pages 226–233, 1982.

J. M. Marin, P. Pudlo, C. P. Robert, and R. Ryder. Approximate Bayesian computational methods. *Statistics and Computing*, 22(6):1167–1180, 2012.

P. Marjoram, J. Molitor, V. Plagnol, and S. Tavaré. Markov chain Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 100 (26):15324–15328, 2003.

G. Martin, B. McCabe, W. Maneesoonthorn, and C. Robert. Approximate Bayesian computation in state space models. 2014. `arXiv:1409.8363`.

U. Picchini. Inference for SDE models via approximate Bayesian computation. *Journal of Computational and Graphical Statistics*, 23(4):1080–1100, 2014.

U. Picchini and J. Forman. Accelerating inference for diffusions observed with measurement error and large sample sizes using approximate Bayesian computation. *Journal of Statistical Computation and Simulation*, 2015. doi: 10.1080/00949655.2014.1002101.

J. Pinheiro and D. Bates. Approximations to the log-likelihood function in the nonlinear mixed-effects model. *Journal of computational and Graphical Statistics*, 4(1):12–35, 1995.

J.K. Pritchard, M.T. Seielstad, A. Perez-Lezaun, and M.W. Feldman. Population growth of human Y chromosomes: a study of Y chromosome microsatellites. *Molecular Biology and Evolution*, 16(12):1791–1798, 1999.

S. Tavaré, D.J. Balding, R.C. Griffiths, and P. Donnelly. Inferring coalescence times from dna sequence data. *Genetics*, 145(2):505–518, 1997.