SAEM-MCMC: some criteria

Mylène Duval, Christèle Robert-Granié

SAEM-MCMC: some criteria

Mylène Duval, Christèle Robert-Granié

INRA SAGA UR 631 - Chemin de Borde Rouge- BP 52627 - 31 326 Castanet-Tolosan Cedex, France
email: Mylene.Duval@toulouse.inra.fr

Tel: 05-61-28-51-94, Fax: 05-61-28-53-53

Abstract

The SAEM-MCMC is a powerful algorithm used to estimate maximum likelihood in the wide class of exponential non-linear mixed effects models. The main problem of this method is that several parameters of simulation need to be calibrated. In this paper we propose some criteria to fix these parameters and we show on a real data set and by simulations that we need to run long markov chains in the Metropolis-Hastings algorithm to obtain an accurate estimator, which is relatively time consuming. In a second part, we applie our method to a model that does not belong to the exponential class, and we show on a simulated data set that we obtain the same results as the exact SAS NLMIXED procedure based on Gaussian quadrature. Our method seems to be appropriate for estimation in this class of non-linear models also.

Key words: non-linear mixed models, maximum likelihood, stochastic algorithm, Metropolis-Hastings, SAEM-MCMC.

1 Introduction

The main interest of this paper is to obtain good parameter estimates using maximum likelihood estimation in non-linear mixed effects models.
Let us recall some information about the linear mixed models. The Expectation-Maximization algorithm (Dempster et al, 1977) is a very famous tool used for parameter estimation in this class of models. Let y denote the vector of observed data, ϕ the vector of unobserved (or missing) data, θ the vector of parameters, $p(y;\theta)$ the incomplete likelihood and $p(y,\phi;\theta)$ the complete likelihood. In the general case, since $p(y;\theta)$ is not in a closed form or is hard to compute, it seems to be difficult to maximize $p(y;\theta)$. In this sense Dempster et al (1977) present the EM algorithm, which tends to maximize $\mathbb{E}\big[\log p(y,\phi;\theta)|y,\theta\big]$.
At iteration k of the algorithm, there are two steps: the E-step computes the conditional expectation of the complete log-likelihood, noted $Q_k(\theta,\theta^{(k-1)})$ equal to
$\mathbb{E}\big[\log p(y,\phi;\theta)|y,\theta^{(k-1)}\big]$ and the M-step determines $\theta^{(k)}$ as maximizing $Q_k(\theta,\theta^{(k-1)})$.
Dempster et al. (1977) and Wu (1983) proved the convergence of the sequence $(\theta^{(k)})_k$ towards a stationary point of the observed likelihood under general regularity conditions.
In the case where we study non-linear mixed models, the E-step leads to an integral that has no closed-form solution. Several approximations of the incomplete log-likelihood have been proposed: the linearization procedure (Sheiner and Beal, 1980), the LME approximation (Lindstrom and Bates, 1990), the Laplace approximation (Wolfinger, 1993; Vonesh, 1996). Some of these methods are available on statistical software like the NLME procedure in S-PLUS software (Pinheiro and Bates, 1995a) or the NLMIXED procedure of SAS. Since errors can be large in the approximation of the observed log-likelihood (Davidian and Giltinian, 1995; Pinheiro and Bates, 1995b; Lindstrom and Bates, 1990), some exact methods based on Monte Carlo methods have appeared. Wei and Tanner (1990) proposed the MCEM algorithm in which the expectation of the E-step is estimated with a mean of some simulated samples from the exact conditionnal distribution of $\phi|y,\theta$. The MCEM algorithm requires an increase of the number of simulated data in order to have some accuracy, and so it is highly time consuming. For instance, Booth and Hobert (1999) report some results from a study on a real data set: they simulated around 60,000 samples for the final iteration. There exists a variant of this method based on the stochastic approximation method of Robbins and Monroe (1951) which promises convergence with fewer simulations: the SAEM algorithm (Delyon et al, 1999). When the conditionnal distribution of the missing effects given the observations, that is to say $p(\phi|y,\theta)$ is unknown, Walker (1996) proposes an EM algorithm with importance sampling to estimate the conditional expectations. In his paper, Walker (1996) tells that importance sampling is more

efficient than other algorithms that use dependent Markov chains to evaluate an expectation under an unknown distribution. In another way, Kuhn and Lavielle (2004, 2005) presented a similar method based on the SAEM algorithm: the SAEM-MCMC algorithm, available on the MONOLIX group website (http://www.monolix.org/). In this method, the E-step of EM is replaced by a stochastic approximation and the simulated sample (which is under the unknown conditionnal distribution of $\phi|y,\theta$) used for the stochastic approximation is simulated with a Metropolis-Hastings algorithm (Robert and Casella, 2004). In this method, several parameters need to be calibrated in order to better estimate the vector of parameters. For example, the Metropolis-Hastings algorithm based on the Monte Carlo and Markov Chains methods, simulates a markov chain using an instrumental distribution. We need to choose the instrumental distribution and the length of the markov chain.

The aim of this study was to present some criteria which determine the parameters we need to fix before running the SAEM-MCMC algorithm, and we discuss if this choice of parameters is relevant or not. The paper is organized as follows: In section 2 we introduce the model and the SAEM-MCMC algorithm in its general version. In section 3 we propose some criteria to determine the parameters to run the algorithm. In section 4 we compare our SAEM-MCMC algorithm with the SAEM-MCMC algorithm in its general version applied with different sets of parameters on the well known orange tree data set and on a simulated data set. Section 5 is devoted to the development of the method when the complete data likelihood $p(y,\phi;\theta)$ does not belong to the curved exponentiel family. Finally a validation of this method by simulation is presented.

2 The nonlinear mixed effects model

2.1 The model

We consider the following model:

$$y_{ij} = f(z_{ij},\phi_i,\beta) + g(z_{ij},\phi_i,\beta,\alpha)\,\varepsilon_{ij},\ \forall i \in \{1,...,N\}\ \forall j \in \{1,..,n_i\},$$

where y_{ij} is the jth observation of subject i, N is the number of subjects, n_i the number of observations of subject i and $(z_{ij})_{ij}$ are known covariates. The ε_{ij}'s are supposed to be independent identically distributed centered Gaussian random variables, independent of the ϕ_i, with variance σ^2.

We note ϕ_i $(k \times 1)$ the vector of individual random parameters of function f and g. It is modeled by:

$$\phi_i = A_i\mu + \eta_i$$

where the $\eta_i \sim \mathcal{N}(0, \Gamma)$ are independent, μ $(c \times 1)$ is an unknown vector of fixed effets, the individual matrix A_i contains covariates and is assumed to be known, and Γ is the covariance matrix of η_i.
β $(p \times 1)$ corresponds to the vector of the unknown other fixed parameters. The variance function g is dependent on f and a parameter vector α. In general $g = f^\alpha$, reflecting the possible character of intra individual variability.

The aim of this study was to estimate the complete vector of unknown parameters $\theta = (\beta, \mu, \Gamma, \sigma^2, \alpha)$ by maximum likelihood estimation. In the case of a linear model, that is to say when f and g are linear in ϕ, the estimation of θ can be treated with the analytic EM algorithm (Dempster et al., 1977). However, a non-linear function is often more suitable for modeling the physical problems but requires a specific approach for estimating the parameters because the Expectation step of the EM algorithm can not be in closed form. Kuhn and Lavielle (2004) propose a method based on the SAEM algorithm coupled with a Monte Carlo Markov Chains method, especially the Metropolis-Hastings algorithm. In this method, some parameters need to be calibrated and we propose some criteria in this sense.

2.2 The SAEM algorithm

The Stochastic Approximation version of the EM algorithm was proposed by Delyon et al. (1999). It consists in replacing the E-step of the EM algorithm by two steps: a simulation step of the missing data under the conditional distribution of $\phi|y, \theta$ and a Stochastic Approximation step.

Given $\theta^{(k-1)}$, estimated parameter value of θ at iteration $k-1$

- Simulation step: Draw $\phi^{(k)}$ under the conditional distribution $p(.|y, \theta^{(k-1)})$

- Stochastic Approximation step: update $Q_k(\theta)$ according to

$$Q_k(\theta) = Q_{k-1}(\theta) + \gamma_k\big[log\ p(\phi^{(k)}|y, \theta^{(k-1)}) - Q_{k-1}(\theta)\big]$$

  where $(\gamma_k)_k$ is a decreasing sequence of positive numbers

- the M-step is the same as the EM algorithm: $\theta^{(k)} = arg\max_\theta Q_k(\theta)$

If we assume that the complete data likelihood $p(y, \phi; \theta)$ belongs to the curved exponentiel family, then we can write it by:

$$p(y, \phi; \theta) = \exp\Big\{ -\Phi(\theta) + \langle S(y, \phi), \psi(\theta) \rangle \Big\}$$

where $\langle ., . \rangle$ denotes the scalar product and $S(y, \phi)$ is the minimal sufficient statistic of the complete model. Then the Stochastic Approximation step of the SAEM algorithm is reduced to compute:

$$s_k = s_{k-1} + \gamma_k \big[S(y, \phi^{(k-1)}) - s_{k-1} \big]$$

The maximization step of the SAEM algorithm consists in computing

$$\theta^{(k)} = Arg \max_{\theta} \big\{ -\Phi(\theta) + \langle s_k, \psi(\theta) \rangle \big\}$$

Properties about the convergence of the sequence $(\theta^{(k)})_k$ towards the maximum likelihood under mild conditions are presented in Delyon et al (1999) in the case of the curved exponentiel family.

Remark In order to have the convergence of the sequence $(\theta^{(k)})_k$, the serie $(\gamma_k)_k$ has to be chosen such that each γ_k must belong to [0,1] and the series $\sum \gamma_k$ must diverge, $\sum \gamma_k^2$ must converge. Kuhn and Lavielle (2004) propose to take the sequence $(\gamma_k)_k$ such that $\gamma_k = 1$ for $1 \leq k \leq K$ and $\gamma_k = (k - K)^{-1}$ else, where K is an integer that can be fixed between 50 and 100. In practice since the sequence $(\gamma_k)_k$ is decreasing quickly, the choice of K seems to be very important and so $\theta^{(K)}$ must be close to the maximum likelihood of θ to be sure that the sequence converges towards the maximum likelihood. In section 3 we present a criterion to fix K.

2.3 The SAEM-MCMC algorithm

Since $(\phi_i | (y_{ij})_j, \theta)_i$ are independent random variables, we can generate them independently. In the general case, the distribution of $(\phi_i | (y_{ij})_j, \theta)_i$ is not in a closed form, so the Simulation step of the SAEM algorithm cannot be directly performed. In this sense Kuhn and Lavielle (2004) proposed to combine the SAEM algorithm with a MCMC procedure: the Metropolis-Hastings algorithm (Robert and Casella, 2004). The results of convergence of this method of estimation, called SAEM-MCMC, are proposed in Kuhn and Lavielle (2004).

This algorithm produces an ergodic Markov chain with stationnary distribution $p(\phi_i|(y_{ij})_j, \theta)$. At this stage some parameters need to be fixed before running this algorithm: the length of the chain (noted $itMC$) and the instrumental distribution. In this paper, we studied another instrumental distribution than the one proposed by Kuhn and Lavielle (2005) and the one proposed in the Monolix software. The algorithm proposed is presented in detail in Section 3.

As suggested in the user guide of Monolix software, we can improve the convergence of the SAEM-MCMC algorithm by running L independent Markov chains and then by doing the Stochastic Approximation on the mean of the $(S(y, \phi_i^{(k,l,itMC)}))_{l=1,...,L}$, where $\phi_i^{(k,l,t)}$ corresponds to the tth iteration of the lth chain at the kth iteration of the SAEM-MCMC algorithm. In fact if L is large enough, $\frac{1}{L}\sum_{l=1}^{L} S(y, \phi_i^{(k,l,itMC)})$ is a good estimator of $E\big[S(y, \phi_i^{(k,l,itMC)})|y, \theta\big]$. To obtain a better approximation, we prefer doing moreover the mean on several iterations intra chains. However the first iterations obtained in the simulated markov chains with the Metropolis-Hastings algorithm are not under the stationary distribution, so we define a parameter (noted $burn$), which corresponds to the length of the burn-in period in the Metropolis-Hastings algorithm.

Then the Stochastic Approximation step becomes:

$$s_k = s_{k-1} + \gamma_k \left(\frac{1}{L(itMC - burn)} \sum_{l=1}^{L} \sum_{t=burn+1}^{itMC} S(y, \phi^{(k,l,t)}) - s_{k-1} \right)$$

where L is the number of independent chains, $itMC$ the length of the chains and $burn$ the length of the burn-in period within chain in the Metropolis-Hastings algorithm.

2.4 Estimations of standard errors and log-likelihood

Let $\hat{\theta}$ be the estimator of θ at convergence, and we denote ∂_θ (∂^2_θ) the differential (the hessian) with respect to θ. Using the Louis formula (Louis, 1982) and following Delyon et al (1999), we approximate the observed information matrix of $\hat{\theta}$ in the following way:

At the Stochastic Approximation step, we calculate

$$\Delta_k = \Delta_{k-1} + \gamma_k[\partial_\theta \log p(y, \phi^{(k)}; \theta^{(k-1)}) - \Delta_{k-1}]$$

$$G_k = G_{k-1} + \gamma_k \Big[\partial^2_\theta \log p(y, \phi^{(k)}; \theta^{(k-1)}) + \big(\partial_\theta \log p(y, \phi^{(k)}; \theta^{(k-1)})\big)^{\,t} \big(\partial_\theta \log p(y, \phi^{(k)}; \theta^{(k-1)})\big) - G_{k-1} \Big]$$

$$H_k = G_k - \Delta_k \,{}^t\Delta_k$$

Then we note $H(\hat{\theta})$ the matrix H_k at the convergence, and we approximate the observed information matrix of $\hat{\theta}$ by the inverse of $-H(\hat{\theta})$.

Concerning the log-likelihood of the observations, we follow the user guide of the Monolix software,
- we draw $\phi_1, ..., \phi_s$ under the conditional distribution $p(\phi|y, \hat{\theta})$ using a Metropolis-Hastings algorithm
- we estimate the log-likelihood of the observations by $\log(l_s)$ where

$$l_s = \frac{1}{s}\sum_{j=1}^{s} p(y|\phi_j, \hat{\theta})\frac{p(\phi_j; \hat{\theta})}{p(\phi_j|y, \hat{\theta})}$$

and $p(\phi; \theta)$ is the distribution of the missing data ϕ. Since $p(\phi|y, \hat{\theta})$ is not known in the general case, we estimate this distribution with a Gaussian distribution, with mean $\mathbb{E}(\phi_s|y, \hat{\theta})$ and variance $Var(\phi_s|y, \hat{\theta})$ estimated using the simulated ϕ_j's.

3 The SAEM-MCMC algorithm with criteria

3.1 The Metropolis-Hastings algorithm

At the kth step of the SAEM-MCMC algorithm and at the $(t+1)$th step of the lth chain simulated by the Metropolis-Hastings algorithm. We note $m_1 \in \mathbb{N}$, $\rho_1 \in \mathbb{R}^+$ and $\rho_2 \in \mathbb{R}^+$.

Given $\phi_i^{(k,l,t)} = x^{(t)}$:

- If $t < m_1$, then generate $W_t \sim \mathcal{N}(A_i\mu^{(k-1)}, \Gamma^{(k-1)}) \rightarrow$ the value obtained is noted w_t.
  Let the acceptance rate: $\rho(x^{(t)}, w_t) = \min\left(\frac{p((y_{ij})_j|\theta^{(k-1)},\phi_i)_{|\phi_i=w_t}}{p((y_{ij})_j|\theta^{(k-1)},\phi_i)_{|\phi_i=x^{(t)}}}, 1\right)$
- If $t \geq m_1$, then generate $\rho \sim \mathcal{U}_{[\rho_1,\rho_2]}$ and $W_t \sim \mathcal{N}(x^{(t)}, \rho\Gamma^{(k-1)})$
  the value obtained is noted w_t.
  The acceptance rate is: $\rho(x^{(t)}, w_t) = \min\left(\frac{p((y_{ij})_j|\theta^{(k-1)},\phi_i)_{|\phi_i=w_t} \times p(\phi_i|\theta^{(k-1)})_{|\phi_i=w_t}}{p((y_{ij})_j|\theta^{(k-1)},\phi_i)_{|\phi_i=x^{(t)}} \times p(\phi_i|\theta^{(k-1)})_{|\phi_i=x^{(t)}}}, 1\right)$
- $\phi_i^{(k,l,t+1)} = \begin{cases} w_t & \text{with probability } \rho(x^{(t)}, w_t) \\ x^{(t)} & \text{with probability } 1 - \rho(x^{(t)}, w_t) \end{cases}$

At this stage some parameters need to be calibrated:
- the value of m_1 (iteration at which we change instrumental distribution),
- the value of $itMC$ (the length of the Markov chains),
- the value of $burn$ (the length of the burn-in period within chain),

- the parameters of the second instrumental distribution ρ_1 and ρ_2,
- the value of L (the number of independent chains).

3.2 Parameters of Metropolis-Hastings

These parameters are fixed as follows: we run a Markov chain with the Metropolis-Hastings algorithm for one individual taking the initial value θ_0 for the vector of parameter θ and we calibrate the parameters with the following methods. Then these parameters are used to run the Metropolis-Hastings algorithm for all individuals.

• **The second instrumental distribution**:
We decided to cut the markov chain into two parts. In the first part, we simulate W_t with the prior distribution of the random effects. In the second part of the chain, W_t is simulated with a Gaussian random walk centered on the precedent iteration of the chain. According to Robert and Casella (2004), the acceptance rate must not be greater than 50%. So we fix ρ_1 and ρ_2 such that the acceptance rate is between 30% and 40 %.

• $m1$: The simulated markov chain does not have the same behavior in the first and in the second part of the chain. In general, in the first part, since the new simulation W_t can be very far from $x^{(t)}$, W_t is often rejected and in this case the chain stays constant. In the second part, the chain varies more. So we choose to fix $m_1 = itMC/10$.

• *burn*: many ways to choose the *burn* parameter are presented in detail in Robert and Casella (2004). We fixed it at $itMC/2$.

• $itMC$: is it necessary that the chain reaches the stationary distribution to better estimate the conditional expectations? Indeed in the MCEM algorithm (Wei and Tanner, 1990), the longer the chains are, the more precise are the estimations of the conditional expectations. Nevertheless in the SAEM-MCMC algorithm the Stochastic Approximation step may be a way to exempt the convergence of the chain towards the stationnary distribution. In this study, we prefer to run enough iterations and test the convergence of the chains towards the stationnary distribution thanks to the Gelman and Rubin (1992) criterion. This criterion provides a diagnosis of the Markov chain convergence by comparing within-chain and between-chain variances.

• L: We know that the more chains are run, the better is the estimation of $E\big[S(y, \phi_i^{(k,l,itMC)})|y, \theta\big]$.

In practice we can run several chains and compare the behavior of each of them. If their properties (mean, variance) are very different, we may choose $L = 5$ or $L = 10$ chains, else, the $L = 1$ chain can be enough.

3.3 Convergence and "smoothing" criteria

3.3.1 A convergence criteria for the SAEM algorithm

Booth and Hobert (1999) propose to use in the MCEM algorithm the same standard stopping rule as in the deterministic EM algorithm:

$$e(k) = \max_j \left(\frac{|\theta_j^{(k)} - \theta_j^{(k-1)}|}{|\theta_j^{(k)}| + \delta_1} \right) < \delta_2 \tag{1}$$

where δ_1 and δ_2 have small values and can be $\delta_1 = 0.001$ and $\delta_2 = 0.0001$ (Searle, 1992 p.296). However, Booth and Hobert (1999) realized that the algorithm can satisfy the criterion thanks to random chance, and so they recommend to stop the algorithm when the criterion has satisfied three consecutive iterations.
On the contrary, Jank (2006) has shown that criterion (1) is not adapted if the sequence $(\gamma_k)_k$ is such that $\gamma_k \propto (1/k)^\alpha$ with $\alpha \approx 1$. Indeed in this case the slow convergence of the algorithm implies that the criterion may lead to a bad estimation of the parameters. In this sense he proposes a stopping rule based on an adaptation of the well known property of increase of the sequence $(Q_k(\theta, \theta^{(k-1)}))_k$ in the EM algorithm.
In our method, we begin the Approximation Step at the Kth iteration and the choice of K depends on the variation of the parameters (see Section 3.3.2). Moreover we run enough iterations in the Metropolis-Hastings algorithm to obtain a good estimation of the conditional expectations. In these conditions we think that the algorithm has almost converged at iteration K and finally we just smooth the estimations thanks to the Approximation Step. Finally we propose to use criterion (1) with $\delta_1 = 0$, $\delta_2 = 0.0001$, and we note $e_l(i)$ the lth component of $e(i)$.

3.3.2 A "smoothing" criterion

In order to determine the parameter K, the iteration at which the sequence $(\gamma_k)_k$ is decreasing in the SAEM-MCMC algorithm, we studied a "smoothing" criterion that is based on the variation of the $e(i)$'s.

Heuristic Figure 1 represents an illustration of the evolution of e_1 (for β a fixed parameter of the model) during the iterations of the SAEM-MCMC algorithm. K was fixed

at 43 iterations. The slope of the curve is high at the beginning and then after iteration around 20, e_1 varies slightly around a small positive value. At this moment, the curve can be smoothed.

At each iteration t of the SAEM algorithm, we fit a linear regression on the ten last points $(e(s))_{s=t-9,\ldots,t}$ and when the slope of the linear regression is not increasing, we add 15 iterations to ensure that we are really in the neighborhood of the maximum likelihood estimator. We define:

$$slope(l,k) = \frac{\sum_{m=1}^{10}\left(m - \frac{1}{10}\sum_{s=1}^{10} s\right)\left(e_l(k-m+1) - \frac{1}{10}\sum_{s=0}^{9} e_l(k-s)\right)}{\sum_{m=1}^{10}(m - \frac{1}{10}\sum_{s=1}^{10} s)^2}$$

$$K = 15 + \min\left\{k,\, \forall l\ slope(l,k) - slope(l,k-1) < 0\right\}$$

3.4 Differences between our SAEM-MCMC and the classical SAEM-MCMC

The two algorithms are based on the same theory presented in Kuhn and Lavielle (2004). Nevertheless many parameters need to be calibrated by the user in the classical SAEM-MCMC algorithm implemented for example in Monolix software (Lavielle 2005, Monolix user guide manuel at http://www.monolix.org/). Kuhn and Lavielle (2005) proposed to do the Approximation Stochastic step on $L = 10$ chains maximum with length $itMC = 10$ iterations. In general it is well known that MCMC methods need Markov chains with millions of iterations until they converge towards the stationary distribution (Robert and Casella, 2004). About the parameters proposed in Kuhn and Lavielle (2005), probably 10 iterations within chain are not enough to reach the stationary distribution. So we want to analyse if the difference of iterations can deteriorate the estimation of θ or if the Stochastic Approximation step is a way for the algorithm to converge towards the maximum likelihood estimator of θ despite short Markov chains.

4 Application and simulation

The SAEM-MCMC algorithm proposed with some criteria was applied to the orange tree data set and to a simulated data set in order to study its properties.

4.1 The Orange tree data set

4.1.1 The model

We applied our SAEM-MCMC algorithm and the classical SAEM-MCMC algorithm with several sets of parameters to the well known orange tree data, studied by Pinheiro and Bates (1995a, 1995b). This data set is available for example on S-Plus and R software and consists of seven measurements of the trunk circumference of each of five orange trees. Following Pinheiro and Bates (1995a, 1995b) and Kuhn and Lavielle (2005), we suggest using the following non-linear mixed model:

$$y_{ij} = \frac{\phi_i}{1 + \exp\big(-\frac{t_{ij}-\beta_2}{\beta_3}\big)} + \varepsilon_{ij},\ \forall i \in \{1, ..., 5\}\ \forall j \in \{1, .., 7\}, \tag{2}$$

where y_{ij} is the jth measurement at age t_{ij} (in days) of the ith tree, the $(\varepsilon_{ij})_{ij}$'s are assumed independent identically distributed with distribution $\mathcal{N}(0, \sigma^2)$, and the ϕ_i's are assumed independent identically distributed with distribution $\mathcal{N}(\beta_1, \tau^2)$. The ϕ_i's are independent from the $(\varepsilon_{ij})_{ij}$'s. Let $\theta = (\beta_1, \beta_2, \beta_3, \tau^2, \sigma^2)$ be the vector of parameters.
The parameters ϕ_i, β_2, β_3 have a physical interpretation: ϕ_i corresponds to the asymptotic trunk circumference, β_2 represents the age at which the tree attains half of its asymptotic trunk circumference, and β_3 is the growth scale. The parameters β_2, β_3 are treated as fixed effects.
The aim of this study was to obtain a good estimator of θ by maximum likelihood estimation. Since the model is linear on ϕ, we can compare our results with the exact EM algorithm results.
Details about the minimal sufficient statistic functions used to estimate θ can be obtained in Kuhn and Lavielle (2005). We compare our SAEM-MCMC algorithm with the classical SAEM-MCMC algorithm using six different sets of parameters (Table 1). Time of running of the SAEM-MCMC algorithm is noted *time* in seconds and let $itSAEM$ the number of iterations in the SAEM-MCMC algorithm, fixed with the stopping rule (1).

For example, set 1. is the set of parameters given in Kuhn and Lavielle (2005): 10 iterations with the first instrumental distribution, 0 with the second one, i.e. $itMC = 10$ and $m_1 = 11$, $L = 10$ independent chains, K fixed at 100 iterations, the algorithm stopped at iteration 303 (3 s).
Parameters ($itMC$, m_1, $burn$, L, K, $itSAEM$) in sets 2 to 6 have been chosen to study the behavior of the estimator of θ. In sets 1 and 2, only the first instrumental distribution is used, with longer Markov chains in set 2 than in set 1. In sets 3 and 4, we used the two

instrumental distributions, with longer chains in set 4 than in set 3. Set 5 is composed of the same parameters of set 4 but with a burn-in period of 50 iterations for each Markov chain. Set 6 is composed of the same parameters of set 4 with the value of K (iteration at which the sequence $(\gamma_k)_k$ is decreasing) fixed at 50 iterations.

4.1.2 Results and discussion

Our criteria determined the following parameters for the SAEM-MCMC algorithm:
$itMC = 500$, $m1 = 50$, $burn = 250$, $L = 10$, $K = 46$ and the convergence criterion stopped the SAEM-MCMC algorithm at the 61st iteration (19 s).

For each set (set 1. to set 6. and our set of parameters), the following initial values of θ were used: $\beta_1^{(0)}$=100, $\beta_2^{(0)}$=650, $\beta_3^{(0)}$=250, $\tau^{2(0)}$=500, $\sigma^{2(0)}$=10. Since the model is linear in ϕ, we can apply the EM algorithm and obtain the exact maximum likelihood estimator of θ.
Table 2 presents the estimation of θ and the value of the known log-likelihood of the observations at these points for the classical SAEM-MCMC applied with the six sets of parameters presented in Table 1, our SAEM-MCMC algorithm, and the EM algorithm. The standard error (noted $\hat{\sigma}(\hat{\theta})$) of the estimator of θ were calculated for the EM algorithm and our method.
Except for sets 1 and 3, we obtain good estimators of θ.

Comparison between the results of sets 1. and 2.
If we use only the first instrumental distribution (sets 1 and 2) to simulate new iterations in short chains in the Metropolis-Hastings algorithm then the sequence $(\theta^{(k)})_k$ does not converge towards the maximum likelihood estimator (set 1.). We obtain this result because the markov chains do not vary enough, and so the estimation of τ^2 is too small. If the chains are longer (set 2), they are more variable and so the variances are estimated better.

Comparison between the results of sets 1. and 3., sets 2. and 4.
Adding a second instrumental distribution in the Metropolis-Hastings algorithm (sets 3 and 4), the estimations for all parameters are better but the estimators for the variances are not exactly the same as those obtained with the EM algorithm.

Comparison between the results of sets 1. and 2., sets 3. and 4.
The longer are the Markov chains in the Metropolis-Hastings algorithm, the more precise

are the estimators.

Comparison between the results of sets 4. and 5.
When we do not run the Stochastic Approximation step with all the iterations of the chains simulated in the Metropolis-Hastings algorithm, that is to say when $burn > 0$, the estimators are closer to the EM estimates.

Comparison between the results of sets 4. and 6.
The choice of the parameter K does not seem to be important in this study, we obtained almost the same results with the two sets of parameters.

The results obtained with our method indicate that it is not necessary to run many iterations with the first instrumental distribution, but Gelman and Rubin's criterion implies to run more iterations with the second one. Our "smoothing" criterion fixed K to 46 iterations. Finally we obtained results close to the EM results.

The standard error of the estimator of θ obtained with our method was close to the EM standard error. In this real data set we only have a few individuals, that's why the standard errors of the estimators are very large, in particular: $\hat{\sigma}(\hat{\tau}^2)$=650.3 for our method. Moreover the log-likelihood of the observations remains stable around the estimator of the maximum likelihood so we cannot have precise estimators. Nevertheless we have seen that set 5. and our method give the best values for the estimators.
In order to validate our method we simulated a data set on the same model (2) with more observations.

4.2 A simulated data set

4.2.1 The model

The model used for the simulated data set is similar to (2).

$$y_{ij} = \frac{\phi_i}{1 + \exp\left(-\frac{t_{ij}-\beta_2}{\beta_3}\right)} + \varepsilon_{ij},\ \forall i \in \{1, ..., 100\}\ \forall j \in \{1, .., 15\},$$

The parameters for generating the 100 simulated data sets are the following:
$\theta = (\beta_1 = 20, \beta_2 = 70, \beta_3 = 30, \tau^2 = 10, \sigma^2 = 0.5)$, the 15 points of observations t_{ij} are the follow: 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 120, 140, 160, 180, 200.

In this section we carry out 100 simulated data sets to compare performances of our SAEM-MCMC algorithm, the classical SAEM-MCMC algorithms previously defined by 6 sets of parameters (set 1. to set 6. presented in Table 1) and the EM algorithm. The choice of the parameters of our method is described in the following section.

4.2.2 Choice of the parameters of SAEM-MCMC

All the parameters were fixed with the criteria presented in Section 3 on the first simulated data set. Then the same parameters were used to apply the SAEM-MCMC algorithm to the other data sets.

• The choice of m_1:
Let us remind the two different instrumental distributions: the first one is the Gaussian distribution with mean β_1 and variance τ^2. The second instrumental distribution is a Gaussian random walk with mean equal to the value of the precedent iteration and with variance equal to $\rho\tau^2$ where $\rho \sim \mathcal{U}[\rho_1, \rho_2]$. When the initial values of the parameters are quite far from the maximum likelihood estimator, the two instrumental distributions do not provide the same behavior to the markov chain simulated in the Metropolis-Hastings algorithm. To analyse that, we simulated a long chain (100,000 iterations) for the individual 1, with parameters $\beta_1 = 10$, β_2=60, β_3=20, σ^2=1, τ^2=3, and $m_1 = 50,000$. The choice of ρ_1 et ρ_2 did not have an incidence on the difference of behavior between the two parts of the chain.

Figure 2 shows that in the first 50,000 iterations, with the first instrumental distribution, the markov chain stays during many iterations at the same place on the contrary to the second part of the markov chain (from iteration 50,000 to 100,000) where it moves around a certain mean value. This fact can be explained by the following: the first instrumental distribution simulates larger variables than the second instrumental distribution and so these variables are more rejected in the first case than in the second case.
Since the model is linear in ϕ_i, the stationnary distribution of the markov chain is known: $p(\phi_1|Y_1, \theta) = \mathcal{N}(u_1, V)$ where u_1 and V are given in Kuhn and Lavielle (2005). Figure 2 clearly shows that the distribution of the markov chain from iterations 40,000 to 50,000 is not Gaussian and so the chain has not reached the stationnary distribution before iteration $m1$.
On the contrary, the histogram of the iterations from 50,000 to 60,000 of the markov

chain and the graph of the density of the distribution $\mathcal{N}(u_1, V)$ is presented in Figure 3. It clearly shows that the chain has yet reached the stationnary distribution at the 60,000 th iteration.

The lack of variability of the chain with the first instrumental distribution can lead us to use only the second instrumental distribution. Nevertheless the second instrumental distribution imposes to the chain to stay in a neighboorhood of the element obtained at iteration m_1. So we prefer to fix a small value for m_1, in particular: $m_1 = itMC/10$.

• The choice of ρ_1 and ρ_2:
Several values of ρ_1 and ρ_2 were tested and to obtain an average acceptance rate close to 35 %, the values retained are $\rho_1 = 1/4$, $\rho_2 = 1/2$.

• The choice of $itMC$ and $burn$:
As suggested in section 4.1., the burn-in period was fixed at $burn = itMC/2$ and we fixed itMC at 300 iterations. With these values, we obtained a Gelman and Rubin coefficient equal to 1.01.

• The choice of L:
Several independent chains were compared and their behavior was similar, so we decided to fix L at 1.

The other parameters of the SAEM-MCMC algorithm were fixed during the run of the algorithm.

4.2.3 Results and discussion

The same six sets of parameters presented in Section 4.1 were considered to compare our SAEM-MCMC algorithm with the classical SAEM-MCMC algorithm.

Table 3 presents the mean value of the estimators, the mean computing time to run the SAEM-MCMC algorithm in seconds (noted $time$), the mean of the number of iterations for the estimation of θ (noted itm) and the mean value of K (noted Km), based on the 100 simulated data sets. Table 4 presents the bias and the Mean Squared Error (MSE) of the estimator of θ obtained with the seven SAEM-MCMC algorithms and the EM algorithm.

The trajectories of the SAEM-MCMC algorithm strongly depend on the value of the initial parameter θ_0, and particularly when $itMC$ is very small. Indeed in this case, if θ_0 is far from θ, the algorithm may not converge towards the maximum likelihood estimator. Since the aim of the paper was to study the bias of the estimator with different sets of parameters for the SAEM-MCMC algorithm and not to study the divergence or the convergence of this algorithm, θ_0 was drawn with a Gaussian distribution of mean θ equal to $(\beta 1 = 20, \beta_2 = 70, \beta_3 = 30, \tau^2 = 10, \sigma^2 = 0.5)$ and standard deviation 0.1θ for each of the 100 simulated data sets.

Concerning the criteria chosen to compare the eight methods, the estimation of the variances of estimators obtained by the classical SAEM-MCMC algorithm may not be positive because of a bad estimation of the conditional expectations (in particular for set 1. and set 2.). In this case, the observed information matrix is not well estimated and the estimation of the matrix is not always definite positive. Consequently we could not compare the different methods with the MQE (Mean Quadratic Error) criterion. Nevertheless the variances of estimates may be a good indicator to analyse and to see if the curve of the complete likelihood is flat close to the maximum likelihood. With our method, the variances of estimators are the following: $\hat{\sigma}(\hat{\theta}) = (0.32, 0.27, 0.23, 1.43, 0.02)$, indicating that the complete likelihood is not flat around the maximum likelihood and we can hope to have precise estimators of the maximum likelihood.

Concerning the mean value of the estimators of the fixed effects, we obtained good results for all the methods. Since the markov chains do not vary enough during the first part of the Metropolis-Hastings algorithm (with the first instrumental distribution), the variance τ^2 estimated as the variability within chains, is underestimated in sets 1. and 2. In the same way results from sets 3. and 4. show that it is necessary to run long chains to have good estimators. The comparison between the results of sets 1. and 3. indicates that adding a second instrumental distribution is relevant for the estimation of the variances τ^2 and σ^2. The mean value of τ^2 (equal to 8.8) and σ^2 (equal to 0.8) for set 2. is better than the ones for set 3. ($\hat{\tau}^2 = 7.4$ and $\hat{\sigma}^2 = 1.3$), which indicates that we must use several instrumental distributions and also run long chains in the Metropolis-Hastings algorithm. Concerning the choice of the parameter $burn$, the results from sets 4. and 6. show that we obtain a light improvement with the burn-in period in the estimation of τ^2 and σ^2. As in the orange tree data set, the results from sets 4. and 6. indicate that the value of K does not have a large incidence since we obtain the same estimators. This may be

because we simulate long chains and so the conditional expectations are good estimates as the first iterations of the SAEM-MCMC algorithm. If we had varied the parameter $itMC$ along the SAEM-MCMC algorithm, that is to say for example short chains at the beginning and long chains at the end, we think that the parameter K would have had more impact on the results.
Our method provides the same results as the EM algorithm, which gives the exact value of the maximum likelihood estimator. Set 5. also provides good estimators close to the EM algorithm estimators but when compared to our method, no criteria are used to choose the parameters of simulation. In terms of computing time, our method is the best one.

Concerning the biases and the MSE, set 5. and our method provide the best results close to the ones of the EM algorithm. In particular, our SAEM-MCMC algorithm provides the best results on biases, especially for the variance τ^2, and the results are quite similar for MSE. Running long chains with two instrumental distributions reduce the MSE and the bias of the estimators for all the parameters.

To conclude, we need to run long chains and use several instrumental distributions in the Metropolis-Hastings algorithm to have a good estimator for θ, and in particular for the variances. The choice of the parameter $burn$ is relevant and the burn-in period is necessary. Concerning the parameter K, we think that it depends on the length of the chain. The longer the chains are, the smaller is the impact of parameter K on the estimation of θ. One advantage of our method is to propose criteria to fix all these parameters, even if some preliminary statistical study must be done before. However some of these criteria may be stringent. For example, the Gelman and Rubin criterion that diagnose the convergence of Markov chains, imposes generally to run long chains.
In the next section we present the case where the model does not belong to the exponential family class, for which Kuhn and Lavielle (2004) did not prove the convergence of the SAEM-MCMC algorithm. Wang (2007) presents a method of estimation based on the EM algorithm and applied the SAEM-MCMC algorithm on several non-exponential models.

5 The SAEM-MCMC algorithm in a wider class of models

Wang (2007) used the SAEM-MCMC algorithm to estimate parameters of Model (2) with the two parameters ϕ and β_2 considered as random effects. In this case the model does

not belong to the exponential family. Although we cannot use the minimal sufficient statistic to estimate the maximum likelihood estimator of θ, the same strategy as the EM algorithm was adopted to estimate the conditional expectations with the Stochastic Approximation step of the SAEM-MCMC algorithm. In this section, our procedure and the exact SAS NLMIXED procedure based on the quadrature method, were compared on the orange trees data set and on a simulated data set.

5.1 The Model

The model is defined by:

$$y_{ij} = \frac{\phi_{1i}}{1 + \exp\left(-\frac{t_{ij} - \phi_{2i}}{\beta_3}\right)} + \varepsilon_{ij},\ \forall i \in \{1, ..., N\}\ \forall j \in \{1, .., N_i\}, \tag{3}$$

where y_{ij} is the jth measurement of the ith tree, the $(\varepsilon_{ij})_{ij}$'s are assumed independent identically distributed as $\mathcal{N}(0, \sigma^2)$, and the $\phi_i = (\phi_{1i}, \phi_{i2})$'s are random variables assuming independent identically distributed with distribution $\mathcal{N}(\beta, D)$ where $\beta = (\beta_1, \beta_2)$ and $D = \begin{pmatrix} \tau_a^2 & \tau_{ab} \\ \tau_{ab} & \tau_b^2 \end{pmatrix}$ the covariance matrix of the ϕ_i's. Let $\theta = (\beta_1, \beta_2, \beta_3, D, \sigma^2)$ the vector of parameters.

5.2 The estimation of the parameters

The heuristic of this method of estimation is the following: for the two random parameters ϕ_1 and ϕ_2, the mean and the variance are estimated from the simulations of the missing data, and more precisely from the chains of the metropolis-Hastings algorithm. Concerning the other parameters, in the general case, we need to evaluate the gradient and the hessien of the complete log-likelihood and we apply Newton's method to obtain the estimation of the parameters. The method used to estimate the vector of parameters θ is presented in Wang (2007).

Since $\log\ p(y, \phi; \theta) = \log\ p(y|\phi; \theta) + \log\ p(\phi; \theta)$ where $p(\phi; \theta)$ is the log-likelihood of the missing data, the EM algorithm at iteration t leads to these estimations:

We note $\omega = (\beta_3, \sigma^2)$.

$\hat{\beta}_1^{(t)} = \sum_{i=1}^{N} \frac{\mathbb{E}\left[\phi_{1i}|y, \theta^{(t-1)}\right]}{N}$

$$\hat{\tau}_a^{2(t)} = \sum_{i=1}^{N} \frac{\mathbb{E}\left[(\phi_{1i}-\hat{\beta}_1^{(t)})^2|y,\theta^{(t-1)}\right]}{N}$$

$$\hat{\beta}_2^{(t)} = \sum_{i=1}^{N} \frac{\mathbb{E}\left[\phi_{2i}|y,\theta^{(t-1)}\right]}{N}$$

$$\hat{\tau}_b^{2(t)} = \sum_{i=1}^{N} \frac{\mathbb{E}\left[(\phi_{2i}-\hat{\beta}_2^{(t)})^2|y,\theta^{(t-1)}\right]}{N}$$

$$\hat{\tau}_{ab}^{(t)} = \sum_{i=1}^{N} \frac{\mathbb{E}\left[(\phi_{1i}-\hat{\beta}_1^{(t)})(\phi_{2i}-\hat{\beta}_2^{(t)})|y,\theta^{(t-1)}\right]}{N}$$

$\frac{\partial \log p(y,\phi;\theta^{(t-1)})}{\partial \omega} = \frac{\partial \log p(y|\phi,\theta^{(t-1)})}{\partial \omega}$ can not be solved in closed form. To obtain an estimation of $\omega^{(t)}$, we consider a single iteration of Newton's method:

$$\omega^{(t)} = \omega^{(t-1)} - H(\phi,\omega^{(t-1)})^{-1} P(\phi,\omega^{(t-1)})$$

where $H(\phi,\omega^{(t-1)}) = \frac{\partial^2 \log p(y|\phi,\theta^{(t-1)})}{\partial \omega^2}$ and $P(\phi,\omega^{(t-1)}) = \frac{\partial \log p(y|\phi,\theta^{(t-1)})}{\partial \omega}$.

When H is not definite positive, that is to say when the estimator is not in the neighborhood of the maximum likelihood, we can use a Marquardt algorithm (Marquardt, 1963): we replace $H(\phi,\omega^{(t-1)})$ by $(H(\phi,\omega^{(t-1)}) + \lambda_t \text{diag}(H(\phi,\omega^{(t-1)})))$, where λ_t is a positive real. In this sense we take λ_t large and then
$(H(\phi,\omega^{(t-1)}) + \lambda_t \text{diag}(H(\phi,\omega^{(t-1)})))^{-1} P(\phi,\omega^{(t-1)}) \sim \frac{1}{\lambda_t} P(\phi,\omega^{(t-1)})$: the algorithm is a descent of gradient.

5.3 The NLMIXED procedure (Pinheiro and Bates, 1995b)

The NLMIXED procedure of SAS software estimates parameters in the nonlinear mixed models by maximizing an approximation of the likelihood of the observations. This one can be written under an integral form over the random effects. Different integral approximations are available, the two principal ones are adaptive Gaussian quadrature (exact approximation) and a first-order Taylor series approximation (approximation by linearization). In this paper we used the adaptive Gaussian quadrature approximation, because it seems to be the best method in terms of estimation of the value of the parameters (Pinheiro and Bates, 1995b). It is used to approximate integrals thanks to a given kernel by

a weighted average of the integrand evaluated at pre-determined abscissas. In this sense, this method can be viewed as a deterministic version of importance sampling method in which the sample and the weights are fixed beforehand. One disadvantage of this method is that it gives accurate results for a large number of abscissas, and so the computing time can be very large.

5.4 Results on the Orange tree data set

In this model seven measurements (Ni=7) of the trunk circumference of each of five orange trees (N=5) were made. The following initial values were taken: $\beta_1^{(0)}$=150, $\beta_2^{(0)}$=600, $\beta_3^{(0)}$=200, $\tau_a^{2(0)}$=500, $\tau_b^{2(0)}$=200, $\tau_{ab}(0)$=0 and $\sigma^{2(0)}$=10. Our criteria fixed the parameters of the SAEM-MCMC algorithm for this real data set and we obtained: $itMC = 300$, $m_1 = 30$, $burn = 150$, with a Gelman and Rubin criterion at 1.06, $\rho_1 = 0$, $\rho_2 = 0.4$ with an average acceptation rate at 25%. It was difficult to increase this rate because either we would have reduced the interval $[\rho_1; \rho_2]$, which may affect the behavior of the markov chains, either we may have reduced the length of the chain, which would have generated a problem to satisfy the Gelman and Rubin criterion. Then we fixed $L = 5$. The other criteria of the algorithm that are setting up during the algorithm runnig fixed $K = 37$, and the stopping rules stopped the algorithm at 172 iterations, the computing time was equal to 28 seconds. The results are presented in Table 5.

Concerning the estimators and the standard errors for the fixed effects, and the variances τ_a^2 and σ^2, we obtained similar results with the two methods. However, the estimators of τ_b^2 and τ_{ab} are very different. We noted that in this study, the NLMIXED procedure was unstable and the estimators strongly depended on the initial values of the parameters. In terms of value of the log-likelihood of the observations, the estimator obtained with our method was the best.

Since the estimated value for the variances are very different between the two methods, we study in the next paragraph a simulated data set based on Model(3).

5.5 Results on a simulated data set

We simulated 100 data sets based on Model (3) with the following parameters: $N = 100$, $Ni = 15$, $\beta_1 = 20$, $\beta_2 = 70$, $\beta_3 = 30$, t_{ij} defined in Section 4.2.1, $D = \begin{pmatrix} 10 & -1 \\ -1 & 40 \end{pmatrix}$ and $\sigma^2 = 0.5$.

After applying our criteria to fix the parameters of simulations of the SAEM-MCMC algorithm for the first simulated data set, we obtained: $itMC = 800$, $m1 = 80$, $burn = 400$, with Gelman and Rubin criterion at 1.04, $L = 1$, $\rho_1 = 0, \rho_2 = 0.4$ with an average rate of acceptation at 30%. The mean computing time was 360 seconds, the mean number of iterations was 143 and the mean K was 38.

Results

The values of the estimators obtained with the two methods were similar for all the parameters. These values were close to the exact parameters, except for τ_{ab}, which indicates that the two methods give coherent results. Concerning biases, MSE, $\hat{\sigma}(\hat{\theta})$ and MQE the results were also similar.

On the contrary to the study of the Orange tree data in the same model, here the NLMIXED method was stable for all initial values. Since we obtained the same results as the exact NLMIXED method, our method seems to be adapted and efficient in maximum likelihood estimation in the general class of the non-linear mixed models.

6 Conclusion

In summary, we propose some criteria to fix the different parameters of the SAEM-MCMC algorithm presented by Kuhn and Lavielle (2004, 2005) in maximum likelihood estimation. We show on the orange tree data and on a simulated data set that we need to run long chains in the Metropolis-Hastings algorithm to obtain precise estimates. These chains must be simulated using several instrumental distributions and taking a burn-in period improves the estimator's value.

In this study our method provides estimates similar to the EM algorithm estimates and the computing time was comparable to that of other software used in maximum likelihood estimation.

Moreover we test our algorithm on a wider class of non-linear models, according to the method of estimation presented by Wang (2007). We obtain the same estimates as the NLMIXED method and close to the exact parameters on a simulated data set. Our algorithm seems to be adapted to the study of several kinds of non-linear models.

In quantitative genetics and animal breeding, heteroscedasticity has generated much interest. In fact the assumption of homogeneous variances may not always be appropriate. In linear mixed models, there is now a large amount of experimental evidence of hetero-

geneous variances for most important livestock production traits (Garrick et al., 1989; Visscher et al., 1991; Visscher et Hill, 1992; Robert-Granié et al., 1999). Major theoretical and applied work has been carried out for estimating and testing the source of heterogeneous variances arising in univariate linear mixed models (Foulley et al., 1990; Gianola et al., 1992; San Cristobal et al., 1993; Foulley and Quaas, 1995; Foulley et al., 1998; Robert et al., 1995a; Robert et al., 1995b). Foulley et al. (1990) formalized the heterogeneous variances using a mixed model approach on log variances in linear mixed models. According to this modelisation, it would be interesting to adapt our algorithm to the study of heterogeneous variances in non-linear mixed models.

7 Acknowledgment

The authors are grateful to Jean-Louis Foulley, Jean-Michel Marin, Didier Concordet, Djalil Chafaï, Julie Antic and Béatrice Laurent for interesting discussions and useful comments.

8 References

Booth, G.J., Hobert, P.J., 1999. Maximizing generalized linear mixed models likelihoods with an automated Monte Carlo EM algorithm. J. Roy. Statist. Soc. B, 61,265-285.

Davidian, M., Giltinan, D.M., 1995. Nonlinear Models for repeated Measures Data. Chapman & Hall, New York.

Delyon, B., Lavielle, M., and Moulines, E., 1999. Convergence of a stochastic approximation version of the EM algorithm. Ann. Statist., 27(1), 94-128.

Dempster, A.P., Laird, N.M., and Rubin, D.B., 1977. Maximum likelihood from incomplete data via the EM algorithm. J. Roy. Stat. Soc. Ser. B, 39, 1-38.

Foulley, J.L., Gianola, D., San Cristobal, M., and Im, S., 1990. A method for assessing extent and sources of heterogeneity of residual vriances in mixed linear models. J. Dairy Sci., 73, 1612-1624.

Foulley, J.L., Quaas, R.L., 1995. Heterogeneous variances in Gaussian linear mixed models. Genet. Sel. Evol, 27, 211-228.

Foulley, J.L., Quaas, R.L., and Thaon d'Arnoldi, C., 1998. A link function approach to heterogeneous variance components. Genet. Sel. Evol., 30, 27-43.

Garrick, D.J., Pollack, E.J., Quaas, R.L., and Van Vleck, L.D., 1989. Variance heterogeneity in direct and maternal weight by sex and percent purebred for Simmental-sired calves. J. Anim. Sci., 67, 2513-2528.

Gelman, A., Rubin, D.B., 1992. Inference from iterative simulation using multiple sequences. Statistical Science, 7, 457-511.

Gianola, D., Foulley, J.L., Fernando, R.L., Henderson, C.R., and Weigel, K.A., 1992. Estimation of heterogeneous variances using empirical Bayes methods: theoretical considerations. J. Dairy Sci., 75, 2805-2823.

Jank, W., 2006. Implementing and diagnosing the stochastic approximation EM algorithm. J. Comput. Graph. Statist., 15, 803-829.

Kuhn, E., Lavielle, M., 2004. Coupling a stochastic approximation version of EM with a MCMC procedure. ESAIM Probab. Stat., 8, 115-131.

Kuhn, E., Lavielle, M., 2005. Maximum Likelihood estimation in non linear mixed effects models. Comput. Statist. Data Anal., 49, 1020-1038.

Lindstrom, M., Bates, D., 1990. Nonlinear mixed effects models for repeated measures data. Biometrics,

44, 673-687.

Louis, T.A., 1982. Finding the observed information matrix when using the EM algorithm. J. Roy. Stat. Soc. Ser. B, 44 (2), 226-233.

Marquardt, D., 1963. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. SIAM J. Appl. Math., 11, 431-441.

Pinheiro, J.C., Bates, D.M., 1995a. Mixed-effects Models in S and S-PLUS. Springer, New York.

Pinheiro, J.C., Bates, D.M., 1995b. Approximations to the log-likelihood function in the nonlinear mixed effects model. J. Comput. Graph. Statist., 4, 12-35.

Robbins, H., Monroe, S., 1951. A stochastic Approximation Method, Ann. Math. Stat., 22, 400-407.

Robert, C.P., Casella, G., 2004. Monte Carlo Statistical methods. Springer, New York.

Robert, C., Foulley, J.L., and Ducrocq V., 1995a. Genetic variation of traits measured in several environments. I. Estimation and testing of homogeneous genetic and intra-class correlations. Genet. Sel. Evol., 27, 111-123.

Robert, C., Foulley, J.L., and Ducrocq V., 1995b. Genetic variation of traits measured in several environments. II. Inference on between environment homogeneity of intra-class correlations using heteroscedastic models. Genet. Sel. Evol., 27, 125-134.

Robert-Granié, C., Banaïti, B., Boichard, D., and Barbat, A., 1999. Accounting for variance heterogeneity in french dairy cattle genetic evaluation. Livest. Prod. Sci., 60, 343-357.

San Cristobal, M., Foulley, J.L., and Manfredi, E., 1993. Inference about multiplicative heteroscedastic components of variance in a mixed linear Gaussian model with an application to beef cattle breeding. Genet. Sel. Evol., 25, 3-30

Sheiner, L.B., Beal, S.L., 1980. Evaluation of methods for estimating population pharmacokinetic parameters. I. Michaelis-Menten model: routine clinical pharmacokinetic data. J. Pharm. Biopharm., 8, 553-571.

Searle, S.R., Casella, G. and McCulloch, C.E., 1992. Variance components. New York: Wiley.

Vonesh, E.F., 1996. A note on Laplacians approximation in nonlinear mixed effects models. Biometrika, 83, 447-452.

Visscher, P.M., Thompson, R., and Hill, W.G., 1991. Estimation of genetic and environmental variances for fat yield in individual herds and an investigation into heterogeneity of variance between herds. Livest. Prod. Sci., 28, 273-290.

Visscher, P.M., Hill, W.G., 1992. Heterogeneity of variance and dairy cattle breeding. Anim. Prod., 55, 321-329.

Walker, S., 1996. An EM algorithm for nonlinear random effects models. Biometrics, 52(3), 934-944.

Wang, J., 2007. EM algorithms for nonlinear mixed effects models. Comput. Statist. Data Anal., 51, 3244-3256.

Wei, G.C.G., Tanner, M.A., 1990. A Monte Carlo implementation of the EM algorithm and the Poor's

Man's data augmentation algorithms. J. Amer. Statist. Assoc., 85 (411), 699-704.

Wolfinger, R.D., 1993. Laplacian's approximation for nonlinear mixed models. Biometrika, 80, 791-795.

Wu, C.-F.J., 1983. On the convergence properties of the EM algorithm. Ann. Stat., 11(1), 95-103.

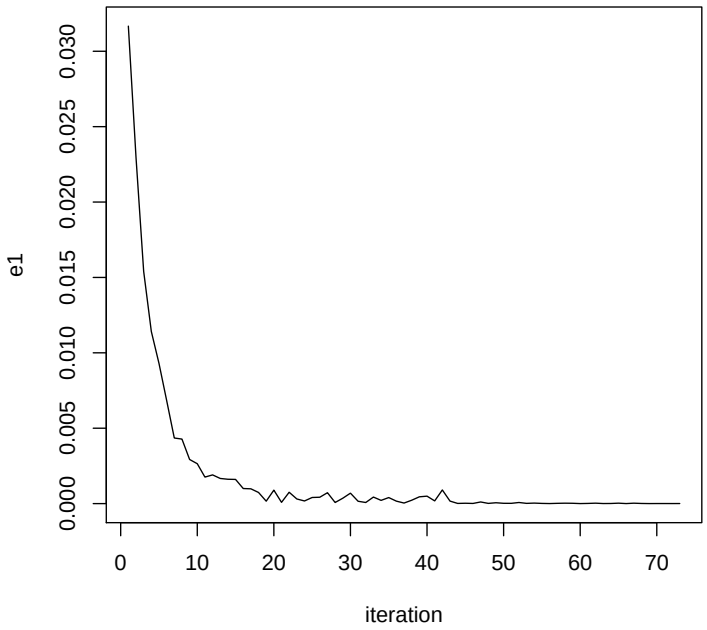

Figure 1: Illustration of the evolution of e_1 during the iterations of the SAEM-MCMC algorithm.

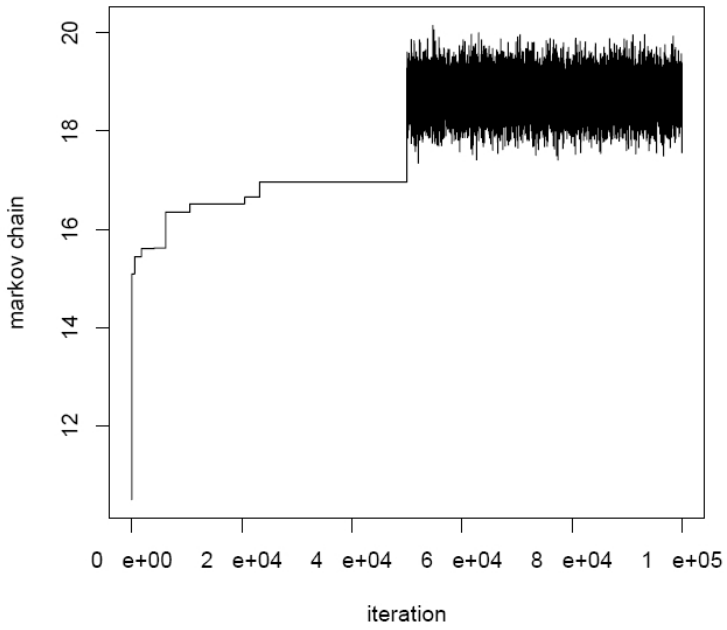

Figure 2: Simulation of a markov chain of the Metropolis-Hastings algorithm.

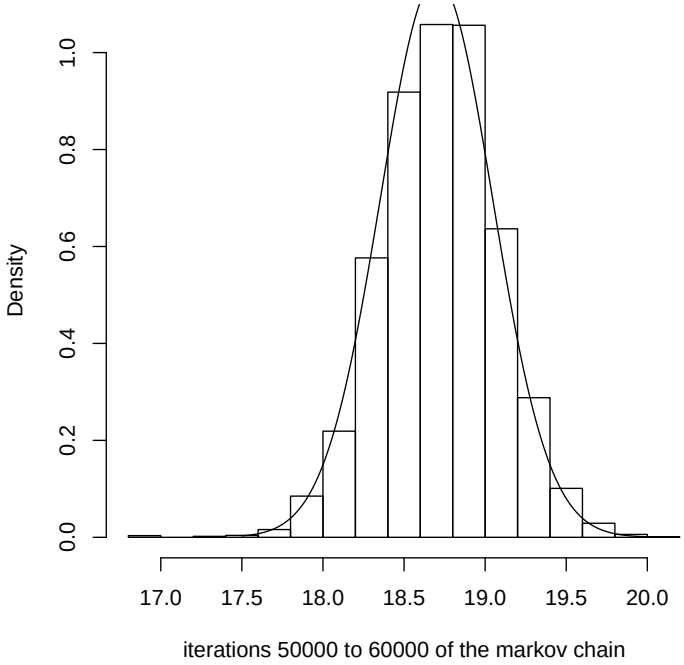

Figure 3: Histogram of the 50,000 to 60,000 iterations of the markov chain and the graph of the density of the distribution $\mathcal{N}(u_1, V)$.

Set number	$itMC$	m_1	$burn$	L	K	$itSAEM$	time
set 1.	10	11	0	10	100	303	3
set 2.	100	101	0	10	100	195	18
set 3.	20	11	0	10	100	222	5
set 4.	200	101	0	10	100	120	22
set 5.	200	101	50	10	100	137	20
set 6.	200	101	0	10	50	102	18
SAEM-MCMC with criteria	500	51	250	10	46	61	19

Table 1: Presentation of the six sets of parameters used in the SAEM-MCMC algorithm, and the set of parameters fixed thanks to the criteria.

θ	β_1	β_2	β_3	τ^2	σ^2	logL
initial value θ_0	100	650	250	500	10	-
EM	192	728	348	1001	62	-131.6
$\hat{\sigma}(\hat{\theta})$	15.7	35.2	27.1	649.5	15.9	-
set 1.	186	697	326	727	169	-137.4
set 2.	191	720	343	957	75	-131.9
set 3.	188	710	335	849	121	-134.4
set 4.	192	728	348	990	68	-131.6
set 5.	192	728	348	1001	61	-131.6
set 6.	192	725	346	984	68	-131.6
SAEM-MCMC with criteria	192	726	347	998	61	-131.6
$\hat{\sigma}(\hat{\theta})$	15.4	32.6	25.3	650.3	16.0	-

$LogL = \log p(y|\hat{\theta})$

Table 2: Values of $\hat{\theta}$ for the EM algorithm, the classical SAEM-MCMC algorithm with six sets of parameters and our SAEM-MCMC algorithm on the Orange tree data.

θ	β_1	β_2	β_3	τ^2	σ^2	logL	t	itm	Km
true value	20	70	30	10	0.5	-	-	-	-
estimate									
EM	19.4	71.5	28.7	10.0	0.6	-1965.50	-	-	-
set 1.	19.3	70.7	28.2	4.3	2.1	-2382.40	41	156	100
set 2.	19.3	71.2	28.5	8.8	0.8	-2002.63	480	131	100
set 3.	19.3	71.2	28.5	7.4	1.3	-2154.08	90	143	100
set 4.	19.4	71.4	28.6	9.5	0.7	-1975.98	600	120	100
set 5.	19.4	71.4	28.6	9.9	0.6	-1965.82	502	114	100
set 6.	19.4	71.4	28.6	9.5	0.7	-1975.96	390	72	50
SAEM-MCMC with criteria	19.4	71.4	28.7	10.0	0.59	-1965.50	28	49	33

$LogL = \log p(y|\hat{\theta})$

The mean estimates are based on 100 simulations.

Table 3: Estimation of the mean estimates for the estimators on the simulation data set using Model (2).

θ	β_1	β_2	β_3	τ^2	σ^2
Bias					
EM	-0.61	1.46	-1.31	0.03	0.09
set 1.	-0.75	0.74	-1.84	-5.67	1.64
set 2.	-0.66	1.22	-1.48	-1.23	0.32
set 3.	-0.66	1.18	-1.51	-2.60	0.85
set 4.	-0.63	1.35	-1.39	-0.54	0.20
set 5.	-0.62	1.40	-1.35	-0.14	0.10
set 6.	-0.63	1.35	-1.39	-0.54	0.20
SAEM-MCMC with criteria	-0.62	1.44	-1.33	0.03	0.09
MSE					
EM	0.46	2.21	1.77	2.35	0.01
set 1.	0.75	1.95	4.00	32.84	2.78
set 2.	0.53	1.75	2.32	3.33	0.11
set 3.	0.55	1.81	2.48	8.04	0.74
set 4.	0.49	1.96	1.99	2.36	0.04
set 5.	0.48	2.08	1.88	2.28	0.01
set 6.	0.49	1.96	1.99	2.36	0.04
SAEM-MCMC with criteria	0.47	2.17	1.80	2.35	0.01

The biases and MSE of the estimates, based on 100 simulations.

Table 4: Estimation of the MSE and MQE of the estimators on the simulation data set using Model (2).

θ	β_1	β_2	β_3	τ_a^2	τ_b^2	τ_{ab}	σ^2	$\log L(\hat{\theta})$
initial value θ_0	150	600	200	500	200	0	10	-
SAEM-MCMC with criteria	191	714	344	1169	984	877	57	-130.89
$\hat{\sigma}(\hat{\theta})$	16.2	31.3	23.3	761.7	1895	951	16	-
NLMIXED	192	725	348	1176	193	313	59	-131.2
$\hat{\sigma}(\hat{\theta})$	16.7	37.4	26.7	905.3	2180	838.3	18.9	-

$LogL(\hat{\theta})$ is an estimation of the log-likelihood of the observations to the point $\hat{\theta}$

Table 5: Value of $\hat{\theta}$ and the standard error of $\hat{\theta}$ for our SAEM-MCMC algorithm and the NLMIXED procedure on the Orange tree data.

θ	β_1	β_2	β_3	τ_a^2	τ_b^2	τ_{ab}	σ^2	$\log L(\hat{\theta})$
exact	20	70	30	10	40	-1	0.5	-
initial values θ_0	10	35	15	5	30	0	0.1	-
SAEM-MCMC with criteria	19.41	71.51	28.67	9.77	40.17	-2.09	0.59	-2071.25
biais	-0.59	1.51	-1.33	-0.23	0.17	-1.09	0.09	-
MSE	0.43	2.66	1.81	2.48	36.94	5.06	0.01	-
$\hat{\sigma}(\hat{\theta})$	0.32	0.69	0.23	1.41	6.52	2.16	0.02	-
MQE	0.53	3.15	1.86	4.50	80.29	9.80	0.01	-
NLMIXED	19.41	71.49	28.68	9.81	40.15	-2.12	0.59	-2071.00
biais	-0.59	1.49	-1.32	-0.19	0.15	-1.12	0.09	-
MSE	0.43	2.59	1.80	2.36	35.77	5.20	0.01	-
$\hat{\sigma}(\hat{\theta})$	0.32	0.69	0.23	1.41	6.52	2.16	0.02	-
MQE	0.53	3.08	1.86	4.40	78.97	9.94	0.01	-

$LogL(\hat{\theta})$ is an estimation of the log-likelihooh of the observations to the point $\hat{\theta}$

Table 6: Value of estimates, biases, MSE and MQE of $\hat{\theta}$ for our SAEM-MCMC procedure and the NLMIXED procedure on a simulated data.