Population Monte Carlo

Author(s): O. Cappé, A. Guillin, J. M. Marin and C. P. Robert

Source: *Journal of Computational and Graphical Statistics*, Vol. 13, No. 4 (Dec., 2004), pp. 907-929
Published by: Taylor & Francis, Ltd. on behalf of the American Statistical Association, Institute of Mathematical Statistics, and Interface Foundation of America
Stable URL: http://www.jstor.org/stable/27594084
Accessed: 11-02-2017 20:11 UTC

Population Monte Carlo

O. CAPPÉ, A. GUILLIN, J. M. MARIN, and C. P. ROBERT

Importance sampling methods can be iterated like MCMC algorithms, while being more robust against dependence and starting values. The population Monte Carlo principle consists of iterated generations of importance samples, with importance functions depending on the previously generated importance samples. The advantage over MCMC algorithms is that the scheme is unbiased at any iteration and can thus be stopped at any time, while iterations improve the performances of the importance function, thus leading to an adaptive importance sampling. We illustrate this method on a mixture example with multiscale importance functions. A second example reanalyzes the ion channel model using an importance sampling scheme based on a hidden Markov representation, and compares population Monte Carlo with a corresponding MCMC algorithm.

Key Words: Adaptive algorithm; Hidden semi-Markov model; Importance sampling; Ion channel model; Multiple scales.

1. INTRODUCTION

When reviewing the MCMC literature, a striking feature is that its focus has been on producing *single* outputs from a given target distribution, π. This may sound a paradoxical statement when considering that one of the major applications of MCMC algorithms is the approximation of integrals

$$\mathfrak{I} = \int h(x)\pi(x)dx$$

by empirical sums

$$\hat{\mathfrak{I}} = \frac{1}{T}\sum_{t=1}^{T} h(x^{(t)}),$$

O. Cappé is Chargé de Recherche, CNRS and TSI, ENST, Telecom Paris, 75634 Paris, Cedex 13. A. Guillin is Maître de Conférence, CEREMADE, Université Paris Dauphine, 75775 Paris, Cedex 16. J. M. Marin is Maître de Conférence, CEREMADE, Université Paris Dauphine, 75775 Paris, Cedex 16. C. P. Robert is Professeur and Head, CREST and CEREMADE, Université Paris Dauphine, 75775 Paris, Cedex 16 (E-mail: xian@ceremade.dauphine.fr).

Journal of Computational and Graphical Statistics, Volume 13, Number 4, Pages 907–929
DOI: 10.1198/106186004X12803

where $(x^{(t)})$ is a Markov chain with stationary distribution π. But the main issue is that π is considered as the limiting distribution of x_t per se and that the Markov correlation between the x_t's is evacuated through the ergodic theorem (Meyn and Tweedie 1993). There exist only a few references to the use of MCMC algorithms for the production of samples of size n from π, including Warnes (2001) and Mengersen and Robert (2003), although the concept of simulation from a product distribution is not fundamentally different from the production of a single output from the target distribution.

Another striking feature in the MCMC literature is the early attempt to dissociate itself from pre-existing techniques such as *importance sampling*, although the latter shared with MCMC algorithms the property of simulating from the wrong distribution to produce approximate generation from the correct distribution (see Robert and Casella 1999, chap. 3). It is only lately that the realization that both approaches can be successfully coupled came upon the MCMC community, as shown for instance by MacEachern and Peruggia (2000), Liu (2001), or Liu, Liang, and Wong (2001). One clear example of this fruitful symbiosis is given by *iterated particle systems* (Doucet, deFreitas, and Gordon 2001). Originally, iterated particle systems were introduced to deal with dynamic target distributions, as for instance in radar tracking, where the imperatives of on-line processing of rapidly changing target distributions prohibited to resort to repeated MCMC sampling. Fundamentally, the basic idea, from a Monte Carlo point of view, consists in recycling previous weighted samples primarily through a modification of the weights (Gordon, Salmond, and Smith 1993), possibly enhanced by additional sampling steps (Berzuini, Best, Gilks, and Larizza 1997; Pitt and Shephard 1999; Gilks and Berzuini 2001). As pointed out by Chopin (2002), a particle system simplifies into a particular type of importance sampling scheme in a static—as opposed to dynamic—setup, where the target distribution π is fixed, which is the setting we consider here.

We introduce a method, called *population Monte Carlo* after Iba (2000), that aims at approximating the target distribution π and that tries to link the above "loose ends" into a coherent simulation principle: Population Monte Carlo borrows from MCMC algorithms for the construction of the proposal, from importance sampling for the construction of appropriate estimators, from SIR (Rubin 1987) for sample equalization, and from iterated particle systems for sample improvement. The population Monte Carlo (PMC) algorithm is in essence an iterated importance sampling scheme that simultaneously produces, at each iteration, a sample approximately simulated from a target distribution *and* (approximately) unbiased estimates $\hat{\mathfrak{I}}$ of integrals $\mathfrak{I}$ under that distribution. The sample is constructed using sample-dependent proposals for generation and importance sampling weights for pruning the sample.

Section 2 describes the PMC technique, which is used in Section 3 in a simple mixture example and in Section 4 for the more ambitious ion channel model. While reasonable in complexity, the mixture example still offers an interesting media to assess the adaptivity of the PMC sampler and the resistance to degeneracy. The ion channel model is more challenging in that there is no closed form representation of the observed likelihood, and the completion step is more delicate than in mixture settings. In particular, Section 4.6

explains why a Metropolis–Hastings algorithm based on the same proposal as PMC fails. Section 5 contains the overall conclusions of the article.

2. THE POPULATION MONTE CARLO APPROACH

As noted by Mengersen and Robert (2003), it is possible to construct an MCMC algorithm associated with the target distribution

$$\prod_{i=1}^{n} \pi(x_i)\,, \tag{2.1}$$

on the space $\mathcal{X}^n$, rather than with the distribution $\pi(x_1)$, on the space $\mathcal{X}$. All standard results and schemes for MCMC algorithms apply in this particular case, and irreducible Markov chains associated with such MCMC algorithms converge to the target (2.1) in distribution. Mengersen and Robert (2003) pointed out that additional sampling devices can be used to construct the proposal distributions, like Gibbs-type component-wise repulsive proposals that exclude immediate neighborhoods of the other points in the sample.

When considering, at MCMC iteration t, a sample $\mathbf{x}^{(t)} = (x_1^{(t)}, \ldots, x_n^{(t)})$, the components $x_i^{(t+1)}$ of $\mathbf{x}^{(t+1)}$ are for instance generated from proposals $q(x|x_i^{(t)})$. However, rather than accepting or rejecting each proposed $x_i^{(t+1)}$ individually, as in standard parallel MCMC sampling, or the whole $\mathbf{x}^{(t+1)}$ globally (which would suffer from the curse of dimensionality), we can remove the issue of convergence assessment by correcting *at each iteration* for the use of the wrong distribution by *importance weighting*.

Thus, instead of resorting to the asymptotic justification of MCMC schemes, we can invoke importance sampling arguments: if the sample $\mathbf{x}^{(t)}$ is produced by simulating the $x_i^{(t)}$'s from distributions q_{it}, independently of one another conditional on the past samples, and if we associate to each point $x_i^{(t)}$ of this sample the importance weight

$$\varrho_i^{(t)} = \pi\left(x_i^{(t)}\right) \Big/ q_{it}\left(x_i^{(t)}\right), \qquad i = 1, \ldots, n\,,$$

estimators of the form

$$\mathfrak{I}_t = \frac{1}{n} \sum_{i=1}^{n} \varrho_i^{(t)} h\left(x_i^{(t)}\right)$$

are *unbiased* for every integrable function h and at *every iteration* t.

This property is the starting point for *population Monte Carlo methods*, namely that extending regular importance sampling techniques to cases where the importance distribution for $x_i^{(t)}$ may depend on both the sample index i and the iteration index t, thus possibly on past samples, does not modify their validity. As already indicated by Robert and Casella (1999, Lemma 8.3.1) in a more restrictive setting, importance sampling estimators have the interesting property that the terms $\varrho_i^{(t)} h(x_i^{(t)})$ are uncorrelated, *even when the proposal* q_{it} *depends on the whole past of the experiment:* assuming that the variances var $\left(\varrho_i^{(t)} h(x_i^{(t)})\right)$

exist for every $1 \leq i \leq n$, which means that the proposals q_{it} should have heavier tails than π, we have

$$\text{var}(\mathfrak{I}_t) = \frac{1}{n^2} \sum_{i=1}^{n} \text{var}\left(\varrho_i^{(t)} h(x_i^{(t)})\right), \tag{2.2}$$

due to the canceling effect of the weights $\varrho_i^{(t)}$.

Because, in most settings, the target distribution is unscaled, we use instead

$$\varrho_i^{(t)} \propto \pi(x_i^{(t)})/q_{it}(x_i^{(t)}), \qquad i = 1, \ldots, n,$$

scaled so that the weights $\varrho_i^{(t)}$ sum up to 1. In this case, the above unbiasedness property and the variance decomposition are lost, although they hold asymptotically (in t). In fact, the estimation of the normalizing constant of π improves with each iteration t, since the overall average

$$\varpi_t = \frac{1}{tn} \sum_{\tau=1}^{t} \sum_{i=1}^{n} \pi\left(x_i^{(\tau)}\right)/q_{i\tau}\left(x_i^{(\tau)}\right) \tag{2.3}$$

is a converging estimator of the inverse of the normalizing constant. Therefore, as t increases, ϖ_t is contributing less and less to the variability of $\mathfrak{I}_t$ and the above properties can be considered to hold for t large enough. In addition, if the sum (2.3) is only based on the $(t-1)$ first iterations, that is, if

$$\varrho_i^{(t)} = \frac{\pi\left(x_i^{(t)}\right)}{\varpi_{t-1}\, q_{it}\left(x_i^{(t)}\right)},$$

the decomposition (2.2) can be recovered, via the same conditioning argument.

A major issue with this approach is the selection of the proposals q_{it}. First, the normalizing constants in these densities (or at least the part that depends on i) must be available in closed form. More generally, even though the unbiasedness property that sustains both the importance sampling technique and this generalization holds for *all* choices of q_{it}'s with support larger than the support of π, as shown in (2.4) below, the q_{it}'s must be such that they do not degenerate to thin-tailed distributions for the variance (2.2) to converge to 0 with n. A natural choice for q_{it} is to select a distribution indexed by the previous value $x_i^{(t-1)}$, say $g(\cdot|x_i^{(t-1)})$, but we will propose in Section 3 a more general scheme where the scale of g depends on the whole sample.

As pointed out by Rubin (1987) for regular importance sampling, it is preferable, rather than to update the weights $\varrho_i^{(t)}$ at each iteration t, to resample (with replacement) n values $y_i^{(t)}$ from $(x_1^{(t)}, \ldots, x_n^{(t)})$ using the weights $\varrho_i^{(t)}$ (including variance reduction by systematic sampling, as in Carpenter, Clifford, and Fernhead 1999). This partially avoids *degeneracy*, that is, the preservation of negligible weights and irrelevant points in the sample. The empirical distribution of the sample $(y_1^{(t)}, \ldots, y_n^{(t)})$ resulting from this sampling importance resampling (SIR) step is thus a convergent estimator of the target distribution.

A pseudo-code rendering of the PMC algorithm is as follows

PMCA: Population Monte Carlo Algorithm

For $t = 1, \ldots, T$

For $i = 1, \ldots, n$,

(a) Select the generating distribution $q_{it}(\cdot)$

(b) Generate $x_i^{(t)} \sim q_{it}(x)$ and compute $\varrho_i^{(t)} = \pi(x_i^{(t)})/q_{it}(x_i^{(t)})$

Normalize the $\varrho_i^{(t)}$'s to sum up to 1

Resample n values from the $x_i^{(t)}$'s with replacement, using the weights $\varrho_i^{(t)}$, to create the sample $(x_1^{(t)}, \ldots, x_n^{(t)})$

Step (a) above is the essential feature of the PMC algorithm: the proposals can be individualized at each step of the algorithm while preserving the validity of the method. They can therefore be picked according to the performances of the previous $q_{i(t-1)}$'s and, in particular, they can depend on the previous sample $(x_1^{(t-1)}, \ldots, x_n^{(t-1)})$ or even on all the previously simulated samples, if storage allows. For instance, the q_{it}'s are random walk proposals centered at the $x_i^{(t-1)}$'s, with various possible scales chosen from earlier performances. The q_{it}'s could also include large tails proposals as in *defensive sampling* of (Hesterberg 1998), to ensure finite variance, or use previous samples to build a kernel nonparametric approximation to π (Warnes 2001).

That the q_{it}'s can globally depend on past iterations with no modification of the $\varrho_i^{(t)}$'s is due to the feature that the unbiasedness equation

$$\begin{aligned} \mathbb{E}\left[\varrho_i^{(t)} h(x_i^{(t)})\right] &= \int\int \frac{\pi(x)}{q_{it}(x)}\, h(x) q_{it}(x)dx\, g(\zeta)d\zeta \\ &= \int\int h(x)\pi(x)dx\, g(\zeta)d\zeta = \mathbb{E}^{\pi}[h(X)]\,, \end{aligned} \tag{2.4}$$

where ζ denotes the vector of past random variates that contribute to q_{it}, does not depend on the density g of this random quantity. Moreover, the dependence on the past does not affect the correlation either, because

$$\begin{aligned} \mathbb{E}\left[\varrho_i^{(t)} h(x_i^{(t)}) \varrho_j^{(t)} h(x_j^{(t)})\right] &= \int\int \frac{\pi(x)}{q_{it}(x)}\, h(x) q_{it}(x)dx\, \frac{\pi(y)}{q_{it}(y)}\, h(y) q_{jt}(y)dy\, g(\zeta)d\zeta \\ &= \int\int h(x)\pi(x)dx \int h(y)\pi(y)dy\, g(\zeta)d\zeta = \mathbb{E}^{\pi}[h(X)]^2\,, \end{aligned} \tag{2.5}$$

where ζ now denotes the vector of past random variates that contribute to either q_{it} or q_{jt}, and g its density.

There are similarities between PMC and earlier proposals in the particle system literature, in particular with Berzuini et al. (1997) and Gilks and Berzuini (2001), since these authors also consider iterated samples with (SIR) resampling steps based on importance weights. A major difference though (besides their dynamic setting of moving target distributions) is that they remain within the MCMC realm by using the resample step *before* the proposal move, hence the need to use Markov transition kernels with a given stationary

distribution. There is also a similarity with Chopin (2002), who considered iterated importance sampling with changing proposals. His setting is a special case of PMC in a Bayesian framework, where the proposals q_{it} are the posterior distributions associated with a portion k_t of the observed dataset (and are thus independent of i and of the previous samples).

As noted earlier, a most noticeable property of the PMC method is that the generality in the choice of the proposal q_{it} is due to the relinquishment of the MCMC framework. Indeed, without the importance resampling correction, a regular Metropolis–Hastings acceptance step for each point of the n-dimensional sample produces a parallel MCMC sampler which simply converges to the target $\pi^{\otimes n}$ in distribution. Similarly, a regular Metropolis–Hastings acceptance step for the whole vector $\mathbf{x}^{(t)}$ converges to $\pi^{\otimes n}$; the advantage in producing an asymptotic approximation to an iid sample is balanced by the drawback that the acceptance probability decreases approximately as a power of n. Because, in PMC, we pick at each iteration the points in the sample according to their importance weight $\varrho_i^{(t)}$, we both remove the convergence issue and construct a selection mechanism over both the points of the previous sample and the proposal distributions. This is not solely a theoretical advantage: In the example of the ion channel in Section 4.6, it actually occurs that a Metropolis–Hastings scheme based on the same proposal does not work well, while a PMC algorithm produces correct answers.

The PMC framework thus allows for a much easier construction of *adaptive* schemes, that is, of proposals that improve themselves against past performances, compared with MCMC setups. Indeed, while adaptive importance sampling strategies have already been considered in the pre-MCMC area, as in, for example, Oh and Berger (1992), the MCMC environment is much harsher for adaptive algorithms, because adaptivity cancels the Markovian nature of the sequence and thus calls for more elaborate ergodicity results. See, for example, Andrieu and Robert (2001) and Haario, Saksman, and Tamminen (1999, 2001) for recent developments in this area. For PMC methods, ergodicity is not an issue because the validity is obtained via importance sampling justifications.

The samples produced by the PMC method can be exploited as regular importance sampling outputs at any iteration T, and thus do not require the construction of stopping rules as for MCMC samples (Robert and Casella 1999, chap. 8). Quite interestingly though, the whole sequence of samples can be exploited, both for adaptation of the proposals and for estimation purposes, as illustrated with the constant approximation (2.3). This does not require a static storage of all samples produced though, since approximations like (2.3) can be updated dynamically. In addition, this possibility to exploit the whole set of simulations implies that the sample size n is not necessarily very large, since the effective simulation size is $n \times T$. A last remark is that the number of points in the sample needs not be constant over iterations: like Chopin (2002), one may increase the number of sample points once the algorithm seems to stabilise in a stationary regime.

3. MIXTURE MODEL

Our first example is a Bayesian modeling of a mixture model, which is a problem simple enough to introduce but complex enough to lead to poor performances for badly

designed algorithms (Robert and Casella 1999, chap. 9). The mixture problem we consider is based on an iid sample $\mathbf{x} = (x_1, \ldots, x_n)$ from the distribution

$$\omega\,\mathcal{N}(\mu_1, \sigma^2) + (1-\omega)\,\mathcal{N}(\mu_2, \sigma^2),$$

where both $\omega \neq 1/2$ and $\sigma > 0$ are known. The prior associated with this model, π, is a normal $\mathcal{N}(\theta, \sigma^2/\lambda)$ prior on both μ_1 and μ_2. We thus aim at simulating from the posterior distribution

$$\pi(\mu_1, \mu_2|\mathbf{x}) \propto f(\mathbf{x}|\mu_1, \mu_2)\,\pi(\mu_1, \mu_2)\,.$$

Although the "standard" associated MCMC algorithm is a Gibbs sampler based on data augmentation, recent developments (Celeux, Hurn, and Robert 2000; Chopin 2002) have shown that the data augmentation step is not necessary to run an MCMC sampler. A PMC sampler can also be efficiently implemented without this augmentation step.

Our PMC algorithm is adaptive : The initialization step consists in choosing a set of initial values for μ_1 and μ_2 (e.g., a grid of points around the empirical mean of the x_i's). The proposals are then random walks, that is, random isotropic perturbations of the points of the current sample. As noted above, the PMC proposals may vary with i and t while preserving the validity of the method. At a first level, the q_{it}'s are all different, since these are densities of normal distributions centered at the previous means. At a second level, we also choose different variances for these normal distributions, using for instance a predetermined set of p scales v_i $(1 \leq i \leq p)$ ranging from 10^3 down to 10^{-3} say, and select these variances at each step of the PMC algorithm according to the performances of the scales on the previous iterations. In our implementation, we select a scale proportionally to its nondegeneracy rate on the previous iterations: when a given variance v_k leads to a high rejection rate of the μ_j's in the resampling step, it is less used in the next iteration, that is, its weight r_k is smaller for the next round. However, to avoid the complete removal of a given variance v_k, we keep a baseline number of points simulated from each variance level, namely 1% of the whole sample. At each iteration, the r_k points associated with a given variance v_k are also chosen at random, in order to avoid trapping configurations around modes associated with the smallest variances.

Note the formal similarity of this scheme to Stavropoulos and Titterington's (2001) *smooth bootstrap*, and Warnes' (2001) *kernel coupler*, when the kernel used in their mixture approximation of π is normal. The main difference is that we do not aim at a good approximation of π using standard kernel results like bandwidth selection, but rather keep the different scales v_i over the iterations. Our PMC algorithm thus looks as follows:

Mixture PMC

Step 0: Initialization

For $j = 1, \ldots, n = pm$, choose $(\mu_1)_j^{(0)}, (\mu_2)_j^{(0)}$
For $k = 1, \ldots, p$, set $r_k = m$

Step t: Update $(t = 1, \ldots, I)$

For $k = 1, \ldots, p$,

1. generate a sample of size r_k as

$$(\mu_1)_j^{(t)} \sim \mathcal{N}\left((\mu_1)_j^{(t-1)}, v_k\right) \quad \text{and} \quad (\mu_2)_j^{(t)} \sim \mathcal{N}\left((\mu_2)_j^{(t-1)}, v_k\right)$$

2. compute the weights

$$\varrho_j \propto \frac{f\left(\mathbf{x} \middle| (\mu_1)_j^{(t)}, (\mu_2)_j^{(t)}\right) \pi\left((\mu_1)_j^{(t)}, (\mu_2)_j^{(t)}\right)}{\varphi\left((\mu_1)_j^{(t)} \middle| (\mu_1)_j^{(t-1)}, v_k\right) \varphi\left((\mu_2)_j^{(t)} \middle| (\mu_2)_j^{(t-1)}, v_k\right)}$$

Resample the $\left((\mu_1)_j^{(t)}, (\mu_2)_j^{(t)}\right)_j$ using the weights ϱ_j,

For $k = 1, \ldots, p$,

update r_k as the number of elements generated with variance v_k which have been resampled.

where $\varphi(q|s, v)$ is the density of the normal distribution with mean s and variance v at the point q.

The performances of the above algorithm are illustrated on a simulated dataset of 1,000 observations from $.2\mathcal{N}(0, 1) + .8\mathcal{N}(2, 1)$, with $\theta = 1$ and $\lambda = .1$ as prior hyperparameters. Applying the above PMC algorithm to this sample, we see that it produces a nondegenerate sample, that is, not restricted to n replications of the same point. In addition, the adaptive feature for choosing among the v_k's is definitely helpful to explore the state space of the unknown means. In this case, $p = 5$ and the five variances are equal to $5, 2, .1, .05$, and $.01$. Moreover, at each Step i of the PMC algorithm, we generated $n = 1{,}050$ sample points. The two upper graphs of Figure 1 illustrate the degeneracy phenomenon associated with the PMC algorithm: they represent the sizes of the samples issued from the different proposals, that is, the number of different points resulting from the resampling step: the upper left graph exhibits a nearly cyclic behavior for the largest variances v_k, alternating from no point issued from these proposals to a large number of points. This behavior agrees with intuition: proposals that have too large a variance mostly produce points that are irrelevant for the distribution of interest, but once in a while they happen to generate points that are close to one of the modes of the distribution of interest. In the latter situation, the corresponding points are associated with large weights ϱ_j and are thus heavily resampled. The upper right graph shows that the other proposals are rather evenly considered along iterations. This is not surprising for the smaller variances, since they modify very little the current sample, but the cyclic predominance of the three possible variances is quite reassuring about the mixing abilities of the algorithm and thus about its exploration performances.

We can also study the influence of the variation in the proposals on the estimation of the means μ_1 and μ_2, as illustrated by the middle and lower panels of Figure 1. First, when considering the cumulative means of these estimations over iterations, they quickly stabilize. The corresponding variances are more unstable over iterations, but this is to

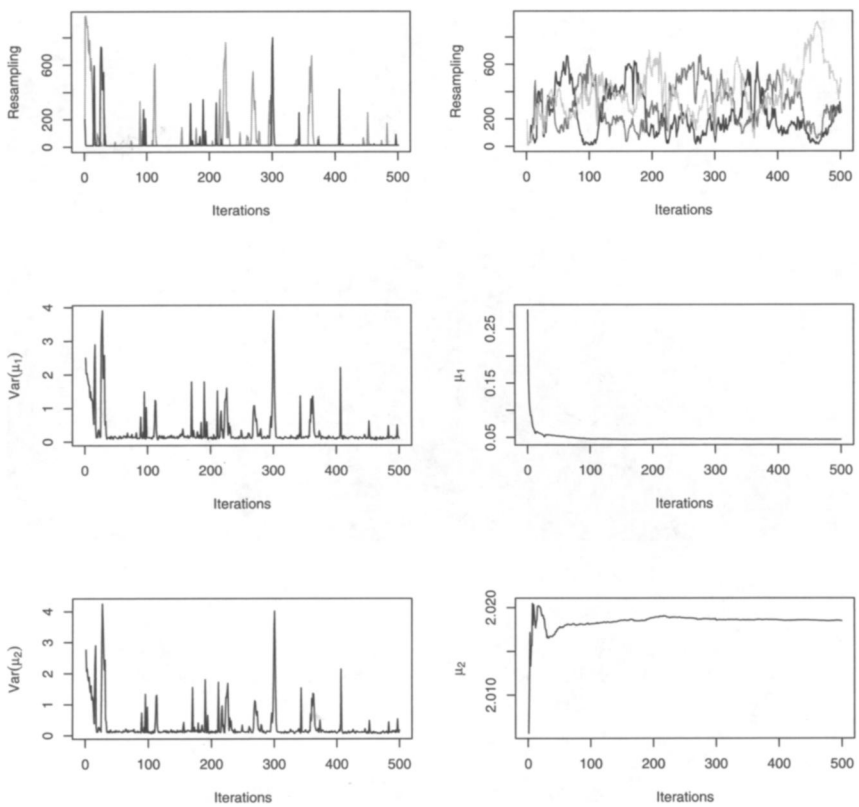

Figure 1. Performances of the mixture PMC algorithm. (upper left) Number of resampled points for the variances $v_1 = 5$ (darker) and $v_2 = 2$. (upper right) Number of resampled points for the other variances, $v_3 = .1$ is the darkest one. (center left) Variance of the simulated μ_1's along iterations. (center right) Complete average of the simulated μ_1's over iterations. (lower left) Variance of the simulated μ_2's along iterations. (lower right) Complete average of the simulated μ_2's over iterations.

be expected, given the regular reappearance of subsamples with large variances. Figure 2 provides an additional insight by representing a weighted sample of means with dots proportional to the weights. As should be obvious from this graph, there is no overwhelming point that concentrates most of the weight. On the opposite, the 1,050 points are rather evenly weighted, especially for those close to the posterior modes of the means.

Note that a better PMC scheme could be chosen. The approach selected for this section does not take advantage of the latent structure of the model, as in Chopin (2002). Indeed, after an initialization step, one could first simulate the latent indicator variable conditionally on the previous sample of (μ_1, μ_2) and then simulate a new sample of means conditionally on the latent indicator variables. Iterating this PMC scheme seems to constitute a sort of parallel Gibbs sampling, but this scheme is valid at any iteration and can thus be stopped at any time. We have not used this approach in order to emphasize that the PMC method has no real need of the latent structure of the model to work satisfactorily.

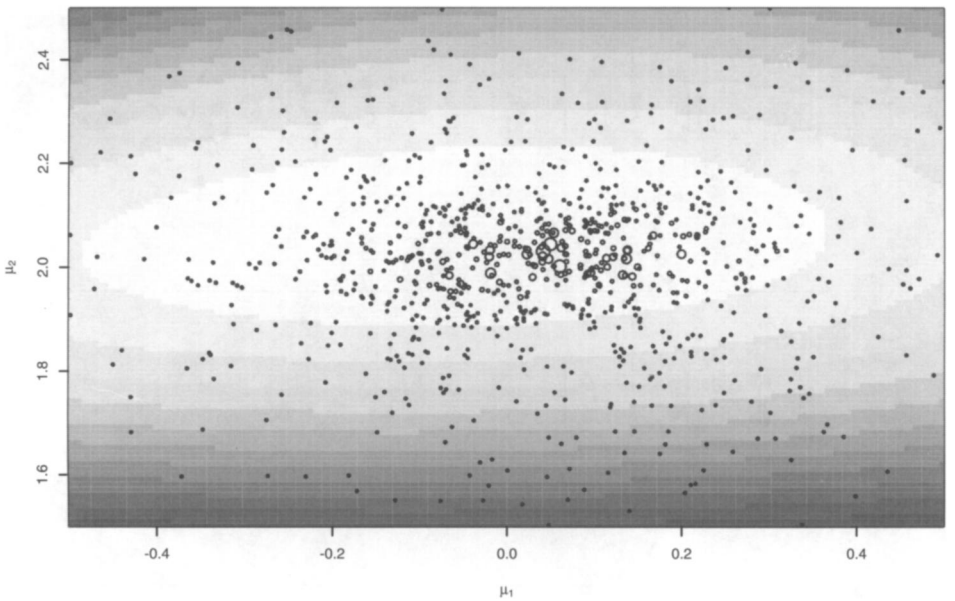

Figure 2. Representation of the log-posterior distribution via gray levels (darker stands for lower and lighter for higher) and of a weighted sample of means. (The weights are proportional to the surface of the circles.)

4. ION CHANNELS

4.1 THE STYLIZED MODEL

As a realistic example of implementation of the PMC scheme, we consider here a formalized version of the ion channel model of Hodgson (1999). We refer the reader to this article, as well as to Ball, Cai, Kadane, and O'Hagan (1999) and Carpenter et al. (1999), for a biological motivation of this model, alternative formulations, and additional references. Let us insist at this point on the *formalized* aspect of our model, which predominantly serves as a realistic support for the comparison of a PMC approach with a more standard MCMC approach in a semi-Markov setting. The finer points of model choice and model comparison for the modeling of ion channel kinetics, while of importance as shown by Ball et al. (1999) and Hodgson and Green (1999), are not addressed by this article. Note also that, while a Bayesian analysis of this model provides a complete inferential perspective, the focus of attention is generally set on the restoration of the true channel current, rather than on the estimation of the parameters of the model.

Hodgson's (1999) model is made of observables $\mathbf{y} = (y_t)_{1 \le t \le T}$ directed by a hidden Gamma (indicator) process $\mathbf{x} = (x_t)_{1 \le t \le T}$ in the following way:

$$y_t | x_t \sim \mathcal{N}(\mu_{x_t}, \sigma^2),$$

while $x_t \in \{0, 1\}$, with durations $d_j \sim \mathcal{G}a(s_i, \lambda_i)$ $(i = 0, 1)$. More exactly, the hidden process $(x_t)_t$ is a (continuous time) Gamma jump process with jump times t_j $(j = 1, 2, \ldots)$

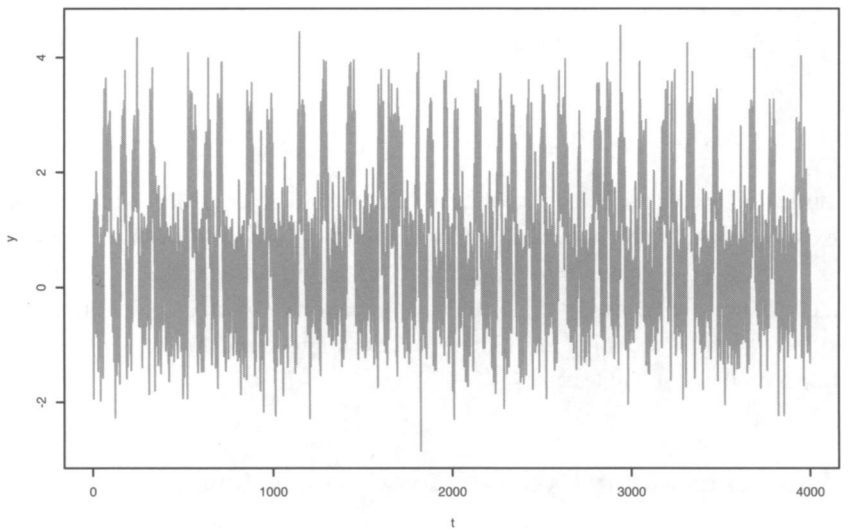

Figure 3. Simulated sample of size 4,000 from the ion channel model, with μ_0 =0, $\mu_1 = 2$, and $\sigma = .5$

such that

$$d_{j+1} = t_{j+1} - t_j \sim \mathcal{G}a(s_i, \lambda_i)$$

if $x_t = i$ for $t_j \leq t < t_{j+1}$, that is, $\mathbb{E}[d_{j+1}] = s_i/\lambda_i$. Figure 3 shows a simulated sample of length 4,000 from this model. (In this article, we will solely ressort to datasets simulated from the true model.) As explained by Hodgson (1999), the two states of x_t represent two different levels of current intensity of channels in a neuron, intensities whose measurement y_t is imprecise.

Our first modification of Hodgson's (1999) model is to assume that the durations d_j—that is, the time intervals in which the process $(x_t)_{1 \leq t \leq T}$ remains in a given state—are integer valued, rather than real valued, and distributed from a negative binomial distribution, $\mathcal{N}eg(s, \omega)$. The main reason for this change is that it is a straightforward generalization of the hidden Markov model where the jumps occur geometrically at integer times (see Ball et al. 1999). In a similar vein, we omit the censoring effect of both the first and the last intervals, given that the influence of this censoring on a long series is bound to be small. A second modification is that we choose a uniform prior for the shape parameters s_0 and s_1 on the finite set $\{1, \ldots, S\}$, rather than an exponential $\mathcal{E}xp(\xi)$ prior on $\mathbb{R}^+$. Two major reasons for this modification are that the following simulations show that the parameters s_0 and s_1 are strongly identified by the observables $(y_t)_{1 \leq t \leq T}$ and that the use of a finite support prior allows for the computation of the normalizing constant in the posterior conditional distribution of the parameters s_0 and s_1, a feature that is paramount for the implementation of PMC. A last modification, when compared with both Hodgson (1999) and Carpenter et al. (1999), is that the observables are assumed to be independent, given the x_t's, rather than distributed from either an AR(15) (Hodgson 1999) or an ARMA(1,1) (Carpenter et al. 1999) model. This modification somehow weakens the identifiability of both regimes as

the data becomes potentially more volatile.

The other parameters of the model are distributed as in Hodgson (1999), using conjugate priors,

$$\mu_0, \mu_1 \sim \mathcal{N}(\theta_0, \tau\sigma^2) \quad \sigma^{-2} \sim \mathcal{G}(\zeta, \eta) \quad \lambda_0, \lambda_1 \sim \mathcal{G}(\alpha, \beta).$$

This model is thus a special case of discrete time hidden semi-Markov model for which there exists no explicit polynomial time formula for the posterior distribution of the hidden process $(x_t)_{1\leq t\leq T}$, as opposed to the hidden Markov model with the forward–backward formula of Baum and Petrie (1966). From a computational (MCMC) point of view, there is therefore no way of integrating this hidden process out to simulate directly the parameters conditional on the observables $(y_t)_{1\leq t\leq T}$.

4.2 POPULATION MONTE CARLO FOR ION CHANNEL MODELS

Our choice of proposal is based on the availability of closed form formula for the hidden Markov model: We create a pseudo hidden Markov model based on the current values of the parameters for the ion channel model, simply by building the Markov transition matrix from the average durations in each state,

$$\mathbb{P} = \begin{pmatrix} 1 - \frac{\lambda_0}{s_0} & \frac{\lambda_0}{s_0} \\ \frac{\lambda_1}{s_1} & 1 - \frac{\lambda_1}{s_1} \end{pmatrix},$$

since, for a hidden Markov model, the average sojourn within one state is exactly the inverse of the transition probability to the other state. We denote by $\pi_H(\mathbf{x}|\mathbf{y}, \omega)$ the full conditional distribution of the hidden Markov chain $\mathbf{x}$ given the observables $\mathbf{y}$ and the parameter $\omega = (\mu_0, \mu_1, \sigma, \lambda_0, \lambda_1, s_0, s_1)$ constructed via the forward–backward formula: see, for example, Cappé, Robert, and Rydén (2003) for details. The simulation of the parameter ω proceeds in a natural way by using the full conditional distribution $\pi(\omega|\mathbf{y}, \mathbf{x})$ since it is available. In order not to confuse the issues, we do not consider the possible adaptation of the approximation matrix $\mathbb{P}$ over the iterations, that is, a modification of the switch probabilities from λ_i/s_i $(i = 1, 2)$.

Note that Carpenter et al. (1999) considered the ion channel model via a particle filter, with the differences that they replace the semi-Markov structure with an approximative hidden Markov model with more than two states, and that they work in a dynamic setting based on this approximation. The observables $\mathbf{y}$ are also different in that they come from an ARMA(1,1) model with only the location parameter depending on the unknown state. Hodgson and Green (1999) similarly compared several hidden Markov model with duplicated "open" and "closed" states. Ball et al. (1999) also relied on a hidden Markov modeling with missing observations.

Because importance sampling bypasses the exact simulation of the hidden process $(x_t)_{1\leq t\leq T}$, it thus avoids the recourse to variable dimension models and to sophisticated tools like reversible jump MCMC (Green 1995). This saturation of the parameter space by the addition of the whole indicator process $\mathbf{x}$ is obviously more costly in terms of storage,

but it provides unrestricted moves between configurations of the process $\mathbf{x}$. Because we do not need to define the corresponding jump moves, we are thus less likely to encounter the slow convergence problems of Hodgson (1999).

We therefore run PMC as in the following pseudo-code, where I denotes the number of iterations, T being used for the number of observations:

Ion channel PMC

Step 0: Initialization

For $j = 1, \ldots, n$,
1. generate $(\omega^{(j)}, \mathbf{x}_-^{(j)}) \sim \pi(\omega) \times \pi_H(\mathbf{x}|\mathbf{y}, \omega^{(j)})$
2. compute the weight $\varrho_j \propto \pi(\omega^{(j)}, \mathbf{x}_-^{(j)}|\mathbf{y})/\pi(\omega^{(j)})\pi_H(\mathbf{x}_-^{(j)}|\mathbf{y}, \omega^{(j)})$

Resample the $(\omega^{(j)}, \mathbf{x}_-^{(j)})_j$ using the weights ϱ_j

Step i: Update $(i = 1, \ldots, I)$

For $j = 1, \ldots, n$,
1. generate $(\omega^{(j)}, \mathbf{x}_+^{(j)}) \sim \pi(\omega|\mathbf{y}, \mathbf{x}_-^{(j)}) \times \pi_H(\mathbf{x}|\mathbf{y}, \omega^{(j)})$
2. compute the weight $\varrho_j \propto \pi(\omega^{(j)}, \mathbf{x}_+^{(j)}|\mathbf{y})/\pi(\omega^{(j)}|\mathbf{y}, \mathbf{x}_-^{(j)})\pi_H(\mathbf{x}_+^{(j)}|\mathbf{y}, \omega^{(j)})$

Resample the $(\omega^{(j)}, \mathbf{x}_+^{(j)})_j$ using the weights ϱ_j, and set $(\mathbf{x}_-^{(j)})_j = (\mathbf{x}_+^{(j)})_j$.

The justification for the weights ϱ_j above is that conditional on the $\mathbf{x}_-^{(j)}$'s, $\omega^{(j)}$ is simulated from $\pi(\omega|\mathbf{y}, \mathbf{x}_-^{(j)})$ and, conditional on $\omega^{(j)}$, $\mathbf{x}_+^{(j)}$ is simulated from $\pi_H(\mathbf{x}|\mathbf{y}, \omega^{(j)})$.

4.3 NORMALIZING CONSTANTS

Let us stress the specificity of the PMC method in terms of normalizing constants: $\pi(\omega|\mathbf{y}, \mathbf{x})$ is available in closed form (see Section 4.4), including its normalizing constant, due to the conjugacy of the distributions on $\mu_0, \mu_1, \sigma, \lambda_0, \lambda_1$ and the finiteness of the support of s_0, s_1. The conditional distribution $\pi_H(\mathbf{x}|\mathbf{y}, \omega)$ is also available with its normalizing constant, by virtue of the forward–backward formula. The only difficulty in the ratio

$$\pi(\omega, \mathbf{x}|\mathbf{y})/\pi(\omega|\mathbf{y}, \mathbf{x})\pi_H(\mathbf{x}|\mathbf{y}, \omega)$$

lies within the numerator $\pi(\omega, \mathbf{x}|\mathbf{y})$ whose normalized version is unknown. We therefore use instead the proportional term

$$\pi(\omega, \mathbf{x}|\mathbf{y}) \propto \pi(\omega)\, f(\mathbf{y}|\mathbf{x}, \omega)\, f(\mathbf{x}|\omega)\,, \tag{4.1}$$

and normalize the ϱ_j's by their sum. The foremost feature of this reweighting is that the normalizing constant missing in (4.1) only depends on the observables $\mathbf{y}$ and is therefore truly a *constant*, that is, does not depend on the previous value of the point $\mathbf{x}_-^{(j)}$. This scheme crucially relies on (1) the simulation space encompassing both the parameters ω and the latent data $\mathbf{x}$, and (2) the distribution $\pi(\omega, \mathbf{x}|\mathbf{y})$ being available in closed form.

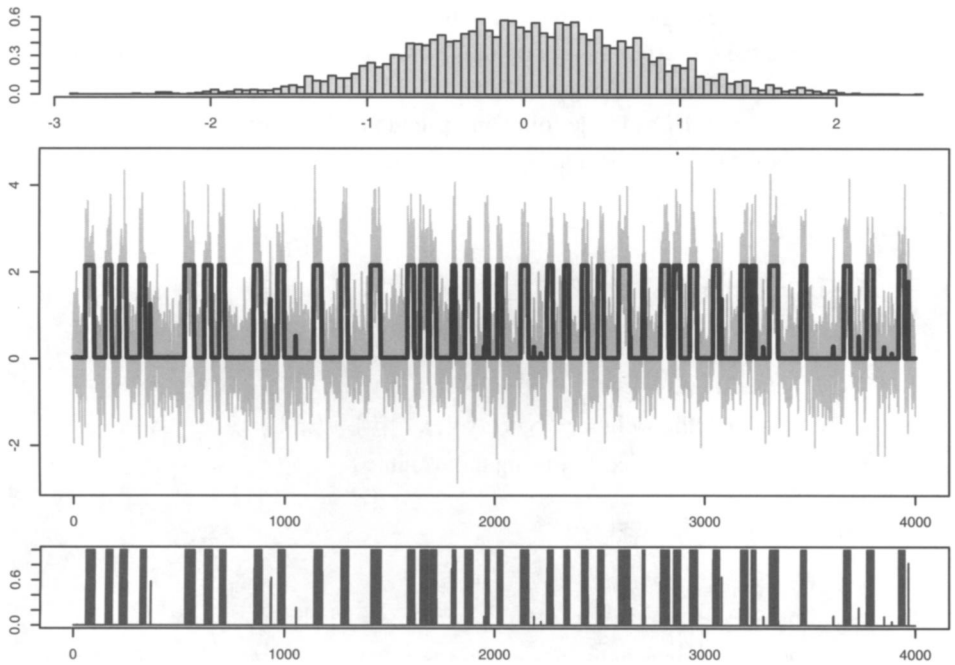

Figure 4. (top) Histograms of residuals after fit by averaged μ_{x_t}. (center) Simulated sample of size 4,000 with $\mu_0 = 0$, $\mu_1 = 2$, $\sigma = .2$, against fitted averaged μ_{x_t}. (bottom) Probability of allocation to first state for each observation. (This graph was obtained after 10 PMC iterations.)

4.4 SIMULATION AND RESULTS

In the joint distribution $\pi(\omega, \mathbf{x}|\ \mathbf{y})$, the unobserved Gamma process $\mathbf{x}$ is distributed as

$$\prod_{m=1}^{M} \frac{(t_{m+1}-t_m)^{s_m-1}\,\lambda_m^{s_m}\,e^{-\lambda_m(t_{m+1}-t_m)}}{\Gamma(s_m)} = \frac{\lambda_0^{n_0 s_0}\,e^{-\lambda_0 v_0}\,\Delta_0^{s_0-1}}{\Gamma(s_0)^{n_0}}\,\frac{\lambda_1^{n_1 s_1}\,e^{-\lambda_1 v_1}\,\Delta_1^{s_1-1}}{\Gamma(s_1)^{n_1}},$$

with obvious notations: M is the number of changes, the t_m's are the successive times when the gamma process changes state, the s_m's, λ_m's are the corresponding sequences of s_0, s_1 and λ_0, λ_1, n_i is the number of visits to state i, Δ_i is the product of the sojourn durations in state i, v_i the total sojourn duration in state i. (This is based on the assumption of no censoring, made in Section 4.1, namely that $t_1 = 1$ and $t_{M+1} = T + 1$.)

The posterior distributions on the μ_i's and σ^{-2} (conditional on the hidden process) are thus the standard Normal-Gamma conjugate priors while

$$\begin{aligned} \lambda_i|s_i, \mathbf{x} &\sim \mathcal{G}a(\alpha + n_i s_i, \beta + v_i) \\ s_i|\mathbf{x} &\sim \pi(s_i|\mathbf{x}) \propto \left[\frac{\Delta_i}{(\beta+v_i)^{n_i}}\right]^{s_i} \frac{\Gamma(n_i s_i + \alpha)}{\Gamma(s_i)^{n_i}}\,\mathbb{I}_{\{1,2,\ldots,S\}}(s_i). \end{aligned}$$

Therefore, except for the s_i's, the posterior distributions on the parameters of the model, that is, $\pi(\omega|\ \mathbf{y},\ \mathbf{x})$, are the same as in Hodgson (1999).

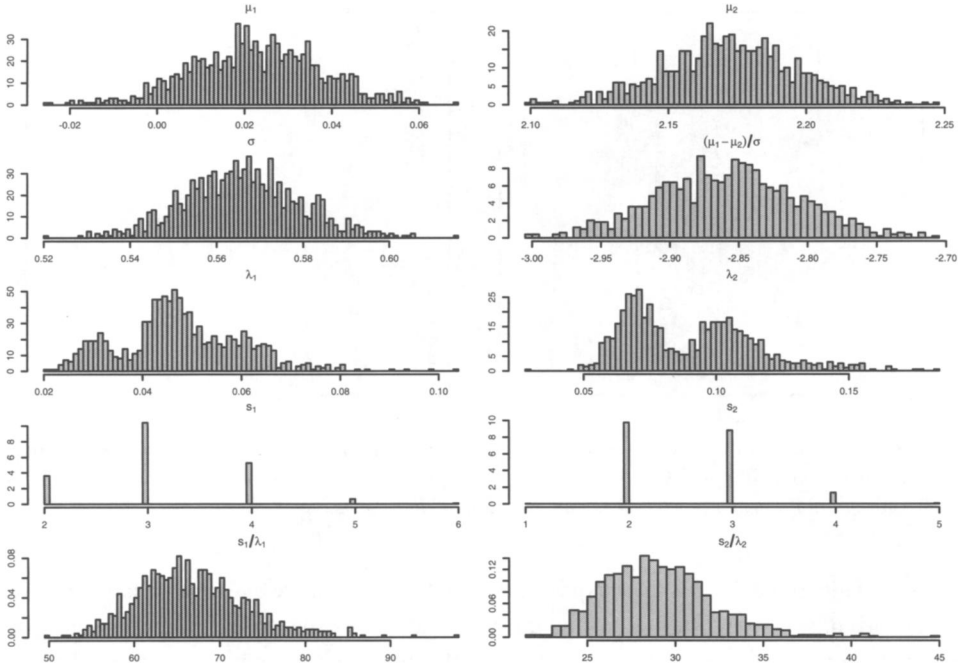

Figure 5. Histograms of the samples produced by PMC, for the dataset of 3,610 points represented on Figure 9, after 10 PMC iterations, before the resampling step of the PMC algorithm. The weighted averages are $\hat{\mu}_1 = .03$, $\hat{\mu}_2 = 2.15$, $\hat{\sigma} = .57$, $\hat{\lambda}_1 = .04$, $\hat{\lambda}_2 = .11$, $\hat{s}_1 = 3$, *and* $\hat{s}_2 = 3$.

The distribution on the s_i's is highly variable, given that the product

$$\left[\frac{\Delta_i}{(\beta + v_i)^{n_i}}\right]^{s_i} \frac{\Gamma(n_i s_i + \alpha)}{\Gamma(s_i)^{n_i}} \tag{4.2}$$

often leads to a highly asymmetric distribution, which puts most of the weight on the minimum value of s. Indeed, when the geometric and arithmetic means, $\Delta_i^{1/n}$ and v_i/n, are similar, a simple Stirling approximation leads to (4.2) being equivalent to $\sqrt{n}/\sqrt{s^n}$.

Figure 4 (center) illustrates the performances of PMC by representing the graph of the dataset against the fitted average

$$\sum_{j=1}^{J} \varrho_j \mu_{x_t^{(j)}}$$

for each observation y_t. The resulting fit is quite good and, besides, the fitted residuals (top) are approximately normal, while the estimated x_t's, that is, the allocations to the states (bottom), are well-marked for almost all observations.

The PMC algorithm provides a sample of ω's which is weighted by the importance weights ϱ_j in the last step of each iteration. Figure 5 gives histograms of the distributions of the components of ω before this reweighting step. As seen from this graph, the histograms in μ_i and σ are well concentrated, while the histogram in λ_1 exhibits two modes which correspond to the two modes of the histogram of s_1 and indicate that the parameter (λ_i, s_i)

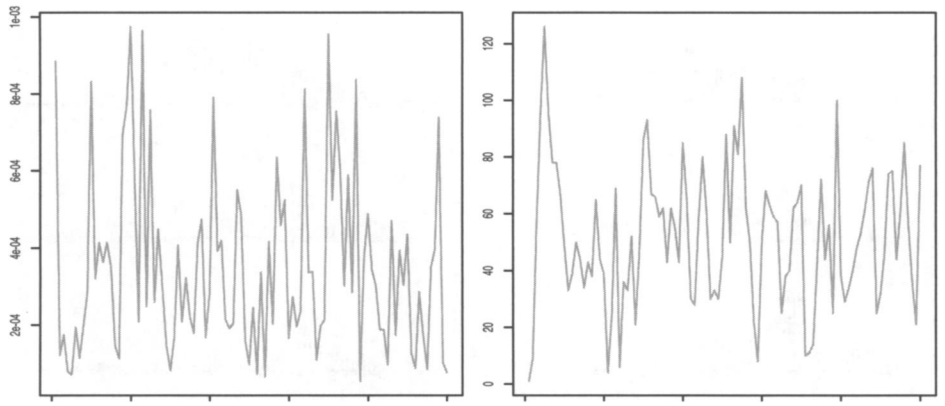

Figure 6. Evolution over 100 consecutive iterations of the PMC algorithm of (left) the variance of the weights ϱ_j along 100 iterations, and (right) the number of points resampled at least once during the resampling step of the PMC algorithm, for a sample of 4,000 observations and 1,000 PMC points.

is not well identified. This is to be expected, given that we observe only a few realizations of the underlying gamma distribution, and this with added noise because the durations are not directly observed. However, the histograms of the average durations s_i/λ_i do not exhibit such multimodality and are well-concentrated around the values of interest. The weighted averages of the different parameters give the following estimators $\hat{\mu}_1 = .03$, $\hat{\mu}_2 = 2.15$, $\hat{\sigma} = .57$, for true values $\mu_1 = .0$, $\mu_2 = 2.0$, $\sigma = .5$, and $\hat{\lambda}_1 = .04$, $\hat{\lambda}_2 = .11$, $\hat{s}_1 = 3$, and $\hat{s}_2 = 3$.

4.5 DEGENERACY

As noted earlier, PMC is simply an importance sampling algorithm when implemented only once, that is, for a single collection of n points $(\omega^{(j)}, \mathbf{x}^{(j)})$. As such, this algorithm provides an approximation device for the target distribution but it is also well known that a poor choice of the importance sampling distribution can jeopardize the interest of the approximation, as for instance when the weights ϱ_j have infinite variance.

An incentive for using PMC in a static setting is to overcome a poor choice of the importance function by recycling the best points and discarding the worst ones. This point of view makes PMC appear as a primitive (but valid) type of *adaptive algorithm*, in that the support of the importance function is adapted to the performance of the previous importance sampler.

The difficulty with this approach is in determining the long-term behavior of the algorithm and, correlatively, the *stopping rule* that decides that nothing is gained in running the algorithm any longer. For instance, it often happens that only a few points are kept after the resampling step of the algorithm, because only a few weights ϱ_j are different from 0. Figure 6 gives the sequence over iterations of the number of points that are resampled at least once, out of $n = 1{,}000$ original points: the percentage of relevant points is thus less than 10% on average and in fact much closer to 5%. In addition, there is no clearcut stabilization in

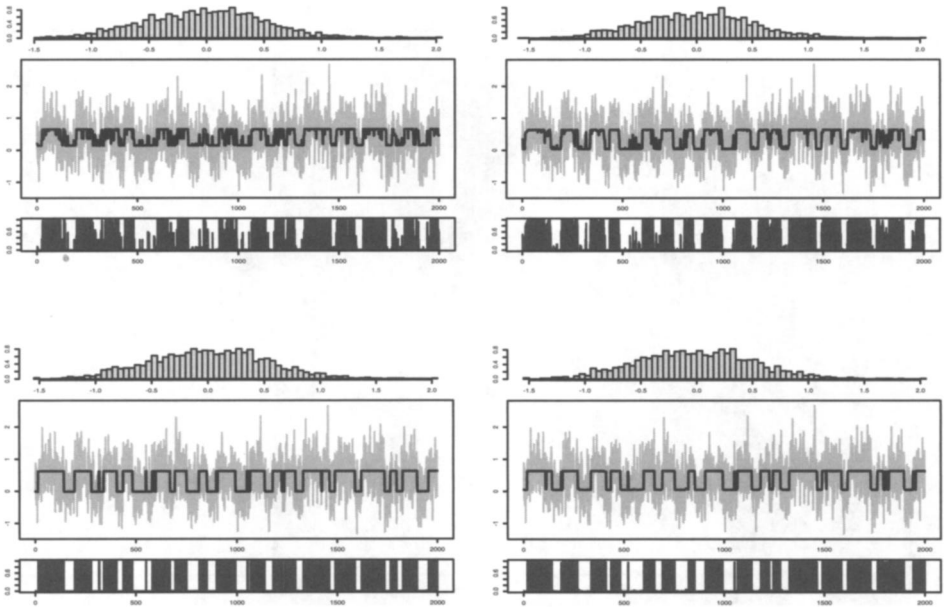

Figure 7. Successive fits of PMC iterated by the weight-resample algorithm for 2,000 simulated observations with $\mu_0 = 0$, $\mu_1 = .5$, $\sigma = .5$ and $n = 2{,}000$ points, for (clockwise starting from top left) 1, 2, 10, and 5 iterations. (See the caption of Figure 4 for a description of the displayed quantities.)

either the number of relevant points or the variance of the corresponding weights, the latter being far from exhibiting a stabilization as the number of iterations increases. Additional assessments can also be considered, like the stabilization of the fit in Figure 7: While the averages for 1 and 2 iterations are quite unstable for most observations, the two states are much more clearly identified for 5 and 10 iterations, and hardly change over subsequent iterations.

A related phenomenon pertains to the degeneracy of ancestors observed in the iterations of our algorithm: as the number of steps increases, the number of points from the first generation used to generate points from the last generation diminishes and, after a few dozen iterations, reduces to a single ancestor. This is for instance what occurs in Figure 8 where, after only two iterations, there is a single ancestor to the whole sample. (Note also the iterations where the whole sample originates from a single point.) This phenomenon appears in every setting and, while it cannot be avoided, since some points are bound to vanish at each iteration even when using the systematic sampling of Carpenter et al. (1999) the surprising factor is the speed with which the number of ancestors decreases.

4.6 A Comparison With a Hastings-Metropolis Algorithm

As mentioned above, the (same) proposal distribution associated with the pseudo hidden Markov model could alternatively be used as a proposal distribution in a Metropolis–Hastings algorithm of the following form:

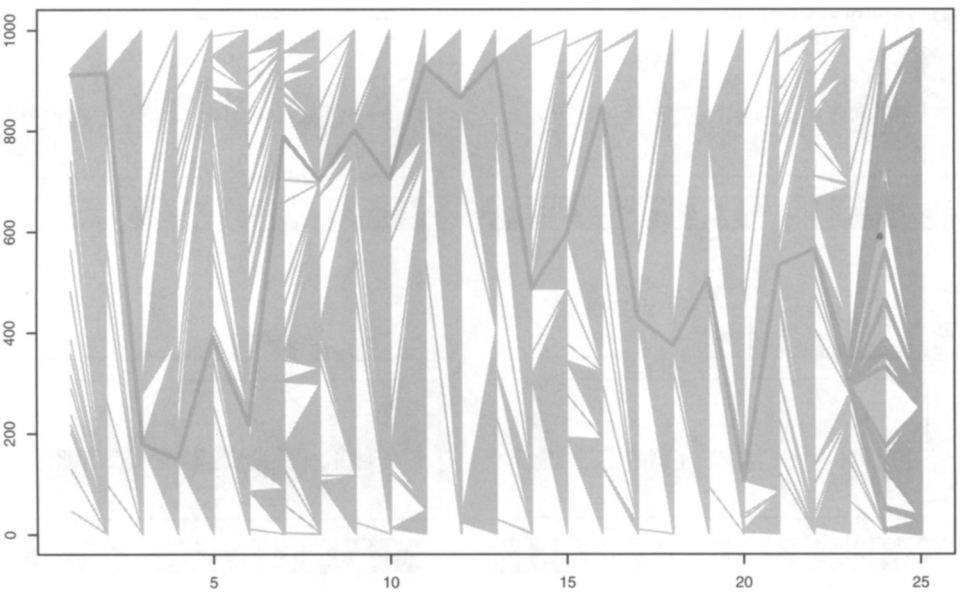

Figure 8. Representation of the sequence of descendants (lighter gray) and ancestors (darker gray) along iterations through bars linking a given ancestor and all its descendants (light gray) or a given point and its ancestor (darker gray). In the simulation corresponding to this graph, there were 4,000 observations and 1,000 points in the PMC sample.

MCMC Algorithm

Step i $(i = 1, \ldots, I)$

1. Generate $\omega^{(i)} \sim \pi(\omega|\mathbf{y}, \mathbf{x}^{(i-1)})$
2. Generate $\mathbf{x}^\star \sim \pi_H(\mathbf{x}|\mathbf{y}, \omega^{(i)})$, $u \sim \mathcal{U}([0,1])$ and take

$$\mathbf{x}^{(i)} = \begin{cases} \mathbf{x}^\star & \text{if} \quad u \leq \dfrac{\pi(\mathbf{x}^\star|\omega^{(i)}\mathbf{y})}{\pi_H(\mathbf{x}^\star|\mathbf{y}, \omega^{(i)})} \Big/ \dfrac{\pi(\mathbf{x}^{(i-1)}|\omega^{(i)}\mathbf{y})}{\pi_H(\mathbf{x}^{(i-1)}|\mathbf{y}, \omega^{(i)})}, \\ \mathbf{x}^{(i-1)} & \text{otherwise} \end{cases}$$

The performances of this alternative algorithm are, however, quite poor. Even with a well-separated dataset like the simulated dataset represented in Figure 3, the algorithm requires a very careful preliminary tuning not to degenerate into a single-state output. More precisely, the following occurs: when started at random, the algorithm quickly converges to a configuration where both means μ_0 and μ_1 of the ion channel model are very close to one another (and to the overall mean of the sample), with, correlatively, a large variance σ^2 and very short stays within each state. To overcome this degeneracy of the sample, we paradoxically had to resort to a *sequential* implementation as follows: noticing that the degeneracy only occurs with large sample sizes, we start the MCMC algorithm on the first 100 observations $y_{1:100}$ and, once a stable configuration has been achieved, we gradually increase the number of observations taken into account (by a factor of $\min(s_0/\lambda_0, s_1/\lambda_1)$) until the whole sample is included. The results provided in Figures 9–11 were obtained following this pseudo-sequential scheme. As for the PMC output, the average fit and allocation

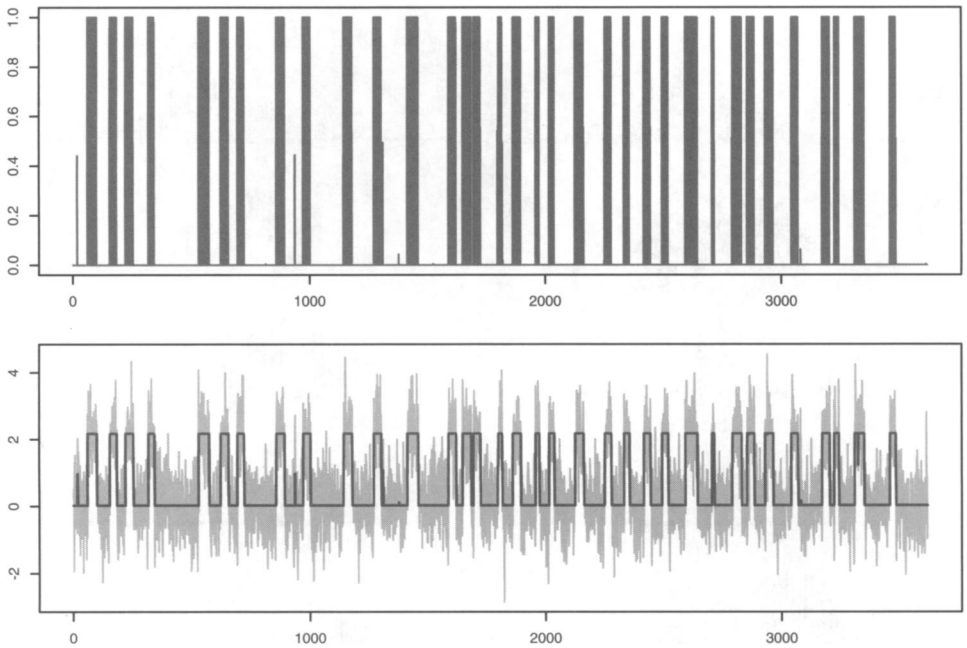

Figure 9. Representation of a dataset of 3,610 simulated values, with $\mu_0 = 0$, $\mu_1 = 2$, and $\sigma = .5$, along with the average fit (bottom), and average probabilities of allocation to the upper state (top). This fit was obtained using a sequential tuning scheme and 5,000 MCMC iterations in the final run, that is, for the complete dataset.

are quite in agreement with the dataset, while the averaged parameters are close to the true values of $\mu_1 = 0$, $\mu_2 = 2$, and $\sigma = .5$. From an MCMC point of view, the cumulated means on the rhs of Figure 10 indicate that more iterations of the MCMC sampler would have been necessary but our purpose in this section is simply to illustrate the correct behaviour of this sampler, under proper initialization.

Attempts with very mixed datasets as the one used in Figure 7 were much less successful since, despite a careful tuning of the starting values (we even tried starting with the known values of the parameters), we could not avoid the degeneracy to a single state. The problem with the Metropolis–Hastings algorithm in this case is clearly a strong dependence on the starting value, that is, a poor mixing ability. This is further demonstrated by the following comparative experiment: when starting the above MCMC sampler from a sample of $n = 1{,}000$ points obtained by running PMC 20 times, the MCMC sampler always produces a satisfactory solution with two clearcut states/means and no degeneracy. For instance, Figure 11 compares the PMC points with the MCMC sample obtained this way via a QQ-plot and shows there is very little difference between both. The same behavior is shown by a comparison of the allocations (not represented here). This indicates that the MCMC algorithm does not lead to a different exploration of the parameter space, while requiring a more careful tune-up to avoid degeneracy.

For a fairly mixed dataset of 2,000 observations, although an MCMC algorithm initialized at random could not avoid degeneracy, a preliminary run of the PMC algorithm provided stable enough allocations to two states to reach the same level of exploration

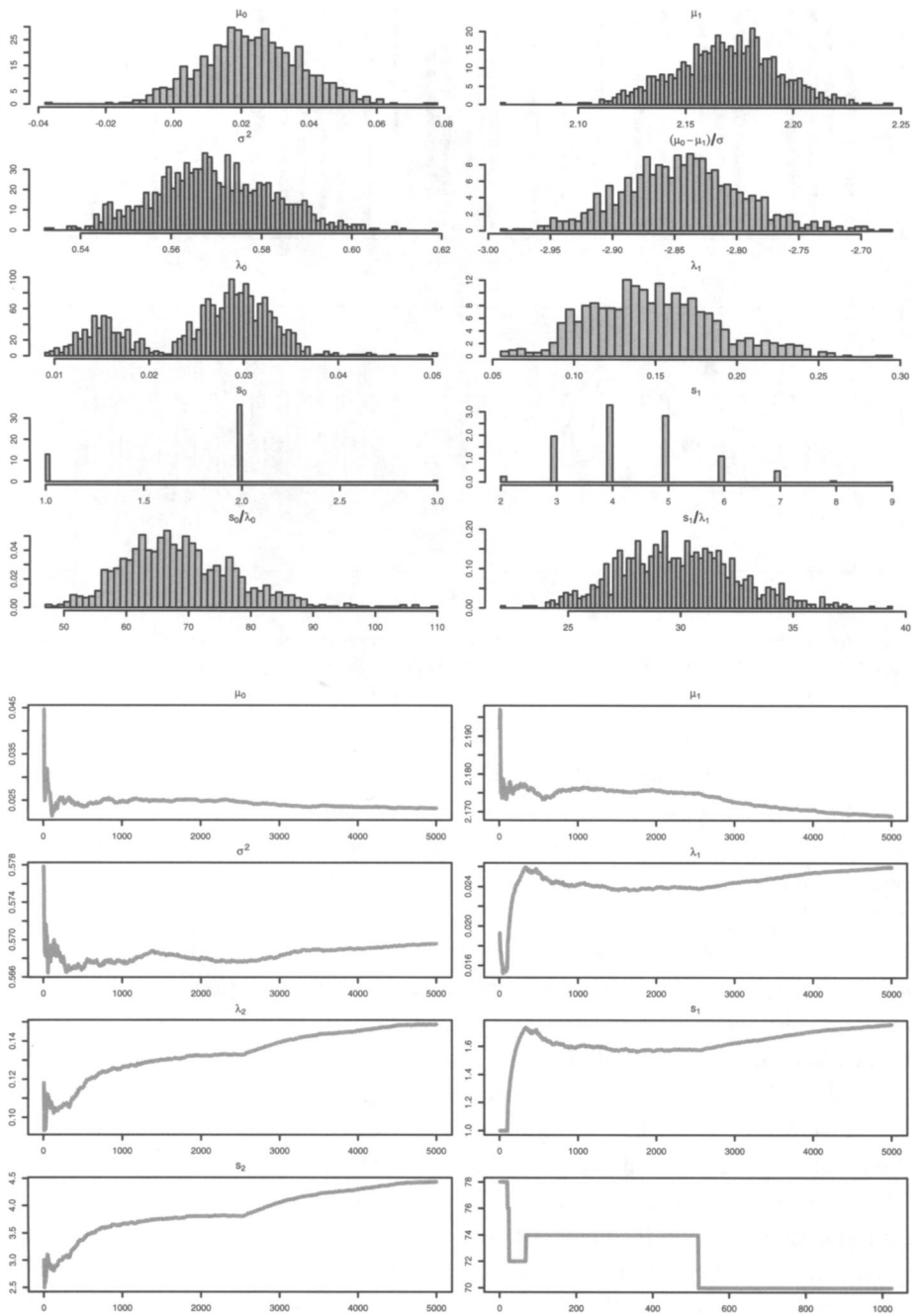

Figure 10. Details of the MCMC sample for the dataset of Figure 9: (top) histograms of the components of the MCMC sample and (bottom) cumulative averages for the seven parameters of the model and evolution of the number of switches (lower right hand side graph).

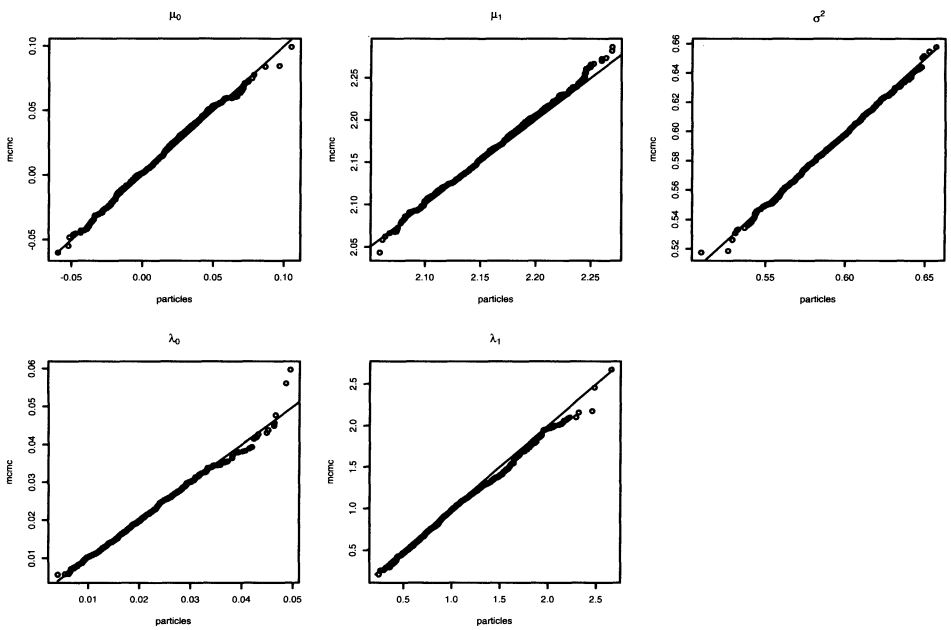

Figure 11. QQ-plot comparing the PMC sample with an MCMC sample for all parameters, obtained after 5,000 iterations of 1,000 MCMC parallel steps started at the 1,000 PMC points.

of the posterior distribution, as shown in Figure 12 by the fit for both PMC and MCMC samples: they are indistinguishable.

This is not to say that an MCMC algorithm cannot work in this setting, since Hodgson (1999) and others demonstrated the opposite, but this shows that *global* updating schemes—that is, proposals that update the whole missing data $\mathbf{x}$ at once—are difficult to work with and, further, that one has to instead rely on more *local* moves as those proposed by Hodgson (1999). A similar conclusion was drawn by Billio, Monfort, and Robert (1999) in the setup of switching ARMA models. (See also Kim, Shephard, and Chib 1998.)

5. CONCLUSION

The above developments have confirmed Chopin's (2002) realization that sequential Monte Carlo is also a useful tool in static setups. Quite obviously, the specific Monte Carlo scheme we built can be used in a sequential setting in a very similar way. The comparison with the equivalent MCMC algorithm in Section 4.6 is also very instructive in that it shows the superior robustness of PMC to a possibly poor choice of the proposal distribution.

There still are issues to explore about PMC scheme. In particular, a more detailed assessment of the iterative and adaptive features is in order, to decide to which extent this is a real asset. When the proposal distribution is not modified over iterations, as in Section 4, it is possible that there is an equivalent to the "cutoff phenomenon": after a given t_0, the distribution of $\mathbf{x}^{(t)}$ may be very similar for all $t \geq t_0$. Further comparisons with full Metropolis–Hastings moves based on similar proposals would also be of interest, to

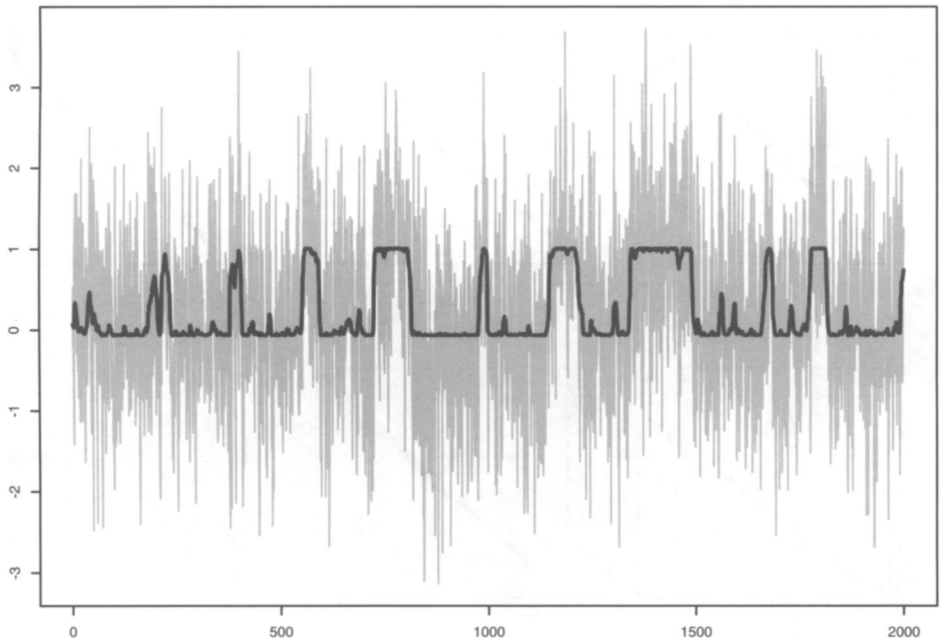

Figure 12. Simulated dataset of 2,000 points with superimposed fits by PMC and the MCMC samples, the later being initialized by PMC. Both fits are indistinguishable.

study which scheme brings the most information about the distribution of interest. The most promising avenue, however, seems to be the development of adaptive proposals as in Section 3, where one can borrow from earlier work on MCMC algorithms to build assessments of the improvement brought by modifying the proposals. Our feeling at this point is that a limited number of iterations is necessary to achieve stability of the proposals.

An extension not studied in this article is that the PMC algorithm can be started with a few points that explore the parameter space and, once the mixing is well-established, the sample size can be increased to improve the precision of the approximation to the integrals of interest. This is a straightforward extension in terms of programming, but the selection of the duplication rate and increase schedule is more delicate.

ACKNOWLEDGMENTS

The authors are grateful to the IHP Friday lunch discussion group for its input, as well as to N. Chopin and G. Roberts. Easter conversations in Cambridge with C. Andrieu and A. Doucet in connection with this work are most sincerely acknowledged. We are thankful to the editorial team for their constructive comments on an earlier version. Research partially supported by EU TMR network ERB–FMRX–CT96–0095 on *"Computational and Statistical Methods for the Analysis of Spatial Data"* and CREST, Insee, Paris.

[Received November 2002. Revised July 2003.]

REFERENCES

Andrieu, C., and Robert C. P. (2001), "Controlled Markov Chain Monte Carlo Methods for Optimal Sampling," *Cahiers du Ceremade* 0125, Université Paris Dauphine.

Ball, F. G., Cai, Y., Kadane, J. B., and O'Hagan, A. (1999), "Bayesian Inference for Ion Channel Gating Mechanisms Directly from Single Channel Recordings, Using Markov Chain Monte Carlo," *Proceedings of the Royal Society of London*, A, 455, 2879–2932.

Baum, L. E., and Petrie, T. (1966), "Statistical Inference for Probabilistic Functions of Finite State Markov Chains," *Annals of Mathematical Statistics*, 37, 1554–1563.

Berzuini, C., Best, N., Gilks, N. G., and Larizza, C. (1997), "Dynamic Conditional Independence Models and Markov Chain Monte Carlo Methods," *Journal of the American Statistical Association*, 92, 590–599.

Billio, M., Monfort, A., and Robert, C. P. (1999), "Bayesian Estimation of Switching ARMA Models," *Journal of Econometrics*, 93, 229–255.

Cappé, O., Robert, C. P., and Rydén, T. (2003), "Reversible Jump, Birth-and-Death, and More General Continuous Time MCMC Samplers," *Journal of the Royal Statistical Society*, Ser. B, 65, 679–700.

Carpenter, J., Clifford, P., and Fernhead, P. (1999), "Building Robust Simulation Based Filters for Evolving Datasets," technical report, Department of Statistics, Oxford University.

Celeux, G., Hurn, M., and Robert, C. P. (2000), "Computational and Inferential Difficulties With Mixture Posterior Distributions," *Journal of the American Statistical Association*, 95, 957–979.

Chopin, N. (2002), "A Sequential Particle Filter Method for Static Models," *Biometrika*, 89, 539–552.

Doucet, A., deFreitas, N., and Gordon, N. (2001), *Sequential MCMC in Practice*, New York: Springer-Verlag.

Gilks, W. R., and Berzuini, C. (2001), "Following a Moving Target—Monte Carlo Inference for Dynamic Bayesian Models," *Journal of the Royal Statistical Society*, Ser. B, 63, 127–146.

Gordon, N., Salmond, D., and Smith, A. F. M. (1993), "Novel Approach to Nonlinear/non-Gaussian Bayesian State Estimation," *IEEE Proceedings F*, 140, 107–113.

Green, P. J. (1995), "Reversible Jump MCMC Computation and Bayesian Model Determination," *Biometrika*, 82, 711–732.

Haario, H., Saksman, E., and Tamminen, J. (1999), "Adaptive Proposal Distribution for Random Walk Metropolis Algorithm," *Computational Statistics*, 14, 375–395.

——— (2001), "An Adaptive Metropolis Algorithm," *Bernoulli*, 7, 223–242.

Hesterberg, T. (1998), "Weighted Average Importance Sampling and Defensive Mixture Distributions," *Technometrics*, 37, 185–194.

Hodgson, M. E. A. (1999), "A Bayesian Restoration of a Ion Channel Signal," *Journal of the Royal Statistical Society*, Ser. B, 61, 95–114.

Hodgson, M. E. A., and Green, P. G. (1999), "Bayesian Choice Among Markov Models of Ion Channels Using Markov Chain Monte Carlo," *Proceedings of the Royal Society of London*, Ser. A, 455, 3425–3448.

Iba, Y. (2000), "Population-Based Monte Carlo Algorithms," *Journal of Computational and Graphical Statistics*, 7, 175–193.

Kim, S., Shephard, N., and Chib, S. (1998), "Stochastic Volatility: Likelihood Inference and Comparison with ARCH Models," *The Review of Economic Studies*, 65, 361–393.

Liu, J. S. (2001), *Monte Carlo Strategies in Scientific Computing*, New York: Springer-Verlag.

Liu, J. S., Liang, F., and Wong, W. H. (2001), "A Theory for Dynamic Weighting in Monte Carlo Computation," *Journal of the American Statistical Association*, 96, 561–573.

MacEachern, S. N., and Peruggia, M. (2000), "Importance Link Function Estimation for Markov Chain Monte Carlo Methods," *Journal of Computational and Graphical Statistics*, 9, 99–121.

Mengersen, K. L., and Robert, C. P. (2003), "Iid Sampling with Self-Avoiding Particle Filters: The Pinball Sampler," in *Bayesian Statistics 7*, eds. J. M. Bernardo, M. J. Bayarri, J. O. Berger, A. P. Dawid, D. Heckerman, A. F. M. Smith, and M. West, Cambridge, MA: Oxford University Press, pp. 277–292.

Meyn, S. P., and Tweedie, R. L. (1993), *Markov Chains and Stochastic Stability*, New York: Springer-Verlag.

Oh, M. S., and Berger, J. O. (1992), "Adaptive Importance Sampling in Monte Carlo Integration," *Journal of Statistical Computing and Simulation*, 41, 143–168.

Pitt, M. K., and Shephard, N. (1999), "Filtering via Simulation: Auxiliary Particle Filters," *Journal of the American Statistical Association*, 94, 1403–1412.

Robert, C. P., and Casella, G. (1999), *Monte Carlo Statistical Methods*, New York: Springer-Verlag.

Rubin, D. B. (1987), *Multiple Imputation for Nonresponse in Surveys*, New York: Wiley.

Stavropoulos, P., and Titterington, D. M. (2001), "Improved Particle Filters and Smoothing," in *Sequential MCMC in Practice*, eds. A. Doucet, N. deFreitas, and N. Gordon, New York: Springer-Verlag, pp. 295–318.

Warnes, G. R. (2001), "The Normal Kernel Coupler: An Adaptive Markov Chain Monte Carlo Method for Efficiently Sampling from Multi-model Distributions," Technical Report 395, University of Washington.