Population Monte Carlo

Olivier Cappé
CNRS / ENST Département TSI, Paris
Arnaud Guillin
Jean–Michel Marin
CEREMADE, Université Paris Dauphine, Paris
Christian P. Robert†
CEREMADE, Université Paris Dauphine, and CREST, INSEE, Paris

Summary. Importance sampling methods can be iterated like MCMC algorithms, while being more robust against dependence and starting values, as shown in this paper. The population Monte Carlo principle we describe here 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 first illustrate this method on a toy mixture example with multiscale importance functions. A second example reanalyses the ion channel model of Hodgson (1999), using an importance sampling scheme based on a hidden Markov representation, and compares population Monte Carlo with a corresponding MCMC algorithm.

Keywords: Adaptive algorithm, degeneracy, hidden semi-Markov model, importance sampling, ion channel model, MCMC algorithms, mixture model, multiple scales, particle system, random walk, unbiasedness.

1. Introduction

When reviewing the literature on MCMC methodology, a striking feature is that it has predominantly focussed 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

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

with empirical sums

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

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

†*Address for correspondence:* CEREMADE, Université Paris Dauphine, 16 place du Maréchal de Lattre de Tassigny, 75775 Paris cedex 16
E-mail: xian@ceremade.dauphine.fr

the x_t's is evacuated through the ergodic theorem (Meyn and Tweedie, 1993). There only exist 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 $\pi^{\bigotimes n}$ 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 realisation 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 *et al.* (2001). One clear example of this fruitful symbiosis is given by *iterated particle systems* (Doucet *et al.*, 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 *et al.*, 1993), possibly enhanced by additional sampling steps (Berzuini *et al.*, 1997; Pitt and Shephard, 1999; Gilks and Berzuini, 2001). As pointed out in 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 thus study in this paper a method, called *population Monte Carlo*, that aims at simulating from the target distribution $\pi^{\bigotimes n}$ and that tries to link these different "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 equalisation, 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 proposed sample.

We describe in Section 2 the population Monte Carlo technique, and apply these development, first to a simple mixture example in Section 3, and second to the more ambitious ion channel model that we assess in Section 4. While reasonable in complexity, the mixture example still offers an interesting media to assess the adaptivity of the population Monte Carlo 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, while 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 population Monte Carlo does not work. Section 5 contains the overall conclusions of the paper.

2. The population Monte Carlo approach

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

$$\pi^{\bigotimes n}(x_1, \ldots, x_n) = \prod_{i=1}^{n} \pi(x_i)\,, \tag{1}$$

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 $\pi^{\bigotimes n}$ in distribution, that is, get approximately distributed as an iid sample from π after a "sufficient" number of iterations. Mengersen and Robert (2003) point out that additional sampling devices can be used to construct the proposal distributions, like Gibbs-type componentwise repulsive proposals that exclude immediate neighbourhoods of the other points in the sample.

When considering, at MCMC iteration t, a sample $\mathbf{x}^{(t)} = (x_1^{(t)}, \ldots, x_n^{(t)})$, we can think of producing the next iteration of the sample $\mathbf{x}^{(t+1)}$ such that the components $x_i^{(t+1)}$ are generated from a proposal $q(x|x_i^{(t)})$. However, rather than accepting each proposed $x_i^{(t+1)}$ individually (which would be a standard form of parallel MCMC sampling) or the whole sample $\mathbf{x}^{(t+1)}$ globally (which would suffer from the curse of dimensionality), we can altogether remove the issue of assessing the convergence to the stationary distribution by correcting *at each iteration* for the use of the wrong distribution by *importance weighting.*

Thus, instead of using an asymptotic justification to an MCMC iterated simulation scheme, we can instead resort to 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, and if we associate to each point $x_i^{(t)}$ of this sample the importance weight

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

estimators of the form

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

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

This 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 does not modify their validity. As already indicated in 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 $\text{var}\left(\varrho_i^{(t)} h(x_i^{(t)})\right)$ exist for every $1 \le i \le n$, which means that the proposals q_{it} should have heavier tails than π, we have

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

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

Obviously, in most settings, the distribution of interest π is unscaled and we have to use instead

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

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 approximately hold. In fact, the estimation of the normalising constant of π improves with each iteration t, since the overall average

$$\varpi_t = \frac{1}{tn} \sum_{\tau=1}^{t} \sum_{i=1}^{n} \frac{\pi(x_i^{(\tau)})}{q_{i\tau}(x_i^{(\tau)})} \tag{3}$$

is a convergent estimator of the inverse of the normalising 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 as holding for t large enough. In addition, if the sum (3) is only based on the $(t-1)$ first iterations, that is, if

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

the variance decomposition (2) can be recovered, via the same conditioning argument.

A related point is that attention must be paid to the selection of the proposals q_{it} so that the normalising constants in these densities (or at least the part that depend on i) must be available in closed form.

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 $\varrho_i^{(t)}$ (and possibly the variance reduction device of systematic sampling, as in Carpenter *et al.*, 1998). This partially avoids the *degeneracy* phenomenon, that is, the preservation of negligible weights and corresponding irrelevant points in the sample. The sample $(y_1^{(t)}, \ldots, y_n^{(t)})$ resulting from this sampling importance resampling (SIR) step is thus akin to an iid sample extracted from the weighted empirical distribution associated with $\pi^{\bigotimes n}(x_1, \ldots, x_n)$.

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

PMCA: Population Monte Carlo Algorithm

For $t = 1, \ldots, T$

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

(i) Select the generating distribution $q_{it}(\cdot)$
(ii) Generate $x_i^{(t)} \sim q_{it}(x)$
(iii) Compute $\varrho_i^{(t)} = \pi(x_i^{(t)})/q_{it}(x_i^{(t)})$

Normalise 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 $(x_1^{(t)}, \dots, x_n^{(t)})$

Step (i) in this representation is stressed because this is an essential feature of the PMC algorithm: the proposal distributions can be individualized at each step of the algorithm without jeopardising the validity of the method. The proposals q_{it} 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 $(x_1^{(t-1)}, \dots, x_n^{(t-1)})$ or even on all the previously simulated samples, if storage allows. For instance, in the mixture setting of Section 3, the q_{it}'s are random walk proposals centered at the $x_1^{(t-1)}$'s, with various possible scales chosen from earlier performances, and they could also include large tails proposals as in the *defensive sampling* strategy of Hesterberg (1998), to ensure finite variance. Similarly, Warnes (2001) uses the previous sample to build a kernel non-parametric approximation to π.

The fact that the proposal distribution $q_{i(t-1)}$ can depend on the past iteration in any possible way without modifying the weight $\varrho_i^{(t)}$ is due to the feature that the unbiasedness equation

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

where ζ denotes the vector of past random variates that contribute to q_{it}, does not depend on the distribution $g(\zeta)$ of this random constituent.

There are similarities between PMC and earlier proposals in the particle system literature, in particular with Berzuini *et al.*'s (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. These authors are thus forced to use Markov transition kernels with given stationary distributions. There is also a similarity with Chopin (2002) who considers 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 distributions 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 $\pi^{\bigotimes n}$ in distribution. Similarly, a regular Metropolis–Hastings acceptance step for the whole vector $\mathbf{x}^{(t)}$ converges to $\pi^{\bigotimes n}$; 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. Since, 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, i.e. of proposals that correct themselves against past performances, than in MCMC setups. Indeed, while adaptive importance sampling strategies have already been considered in the pre-MCMC area, as in, e.g., Oh and Berger (1992,1993), the MCMC environment is much harsher for adaptive algorithms, because the adaptivity cancels the Markovian nature of the sequence and thus calls for more elaborate convergence studies to establish ergodicity. See, e.g., Andrieu and Robert (2001) and Haario *et al.* (1999,2001) for recent developments in this area. For PMC methods, ergodicity is not an issue since 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). This does not require a static storage of all samples produced though, since approximations like (2) 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 is not necessarily constant over iterations. As in Chopin (2002), one may increase the number of points in the sample once the algorithm seems to stabilise in a stationary regime.

3. Mixture model

Our first example is a Bayesian modelling 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; Cappé *et al.*, 2003). The mixture problem we consider is based on an iid sample $\mathbf{x} = (x_1, \ldots, x_n)$ from the distribution

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

where both $p \neq 1/2$ 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" MCMC resolution of the mixture problem is to use a Gibbs sampler based on a data augmentation step via indicator variables, recent developments (Celeux *et al.*, 2000; Chopin, 2002; Cappé *et al.*, 2003) have shown that the data augmentation step is not necessary to run an MCMC sampler. We will now demonstrate that a PMC sampler can be efficiently implemented without this augmentation step either.

Our PMC algorithm is adaptive in the following sense: The initialization step consists first 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, a very appealing feature of the PMC method is that the proposal may vary from one point of the sample to another without jeopardizing the validity of the method. At a first level, the proposals are all different, since they are normal distributions centered in every sample point. At a second level, we can also choose different variances for these normal distributions, for instance within a predetermined set of p scales v_i $(1 \le i \le p)$ ranging from 10^3 down to 10^{-3}, 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 decided to select a scale proportionally to its non-degeneracy rate on the previous iterations. (Note the formal similarity of this scheme with Stavropoulos and Titterington's (1999) *smooth bootstrap*, or *adaptive importance sampling*, 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: Initialisation

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

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

(a) generate a sample of size r_k as

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

(b) compute the weights

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

Resample the $\left((\mu_1)_j^{(i)}, (\mu_2)_j^{(i)}\right)_j$ 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.

As mentioned above, the weight associated with each variance v_k is thus proportional to the regeneration (or survival) rate of the corresponding sample. If most μ_j's associated with a given v_k are not resampled, the next step will see less generations using this variance v_k. However, to avoid the complete removal of a given variance v_k, we modified the algorithm to ensure that a minimum number of points is simulated from each variance level, namely 1% of the whole sample.

The performances of the above algorithm are illustrated on a simulated dataset of 1000 observations from the distribution $0.2\mathcal{N}(0,1)+0.8\mathcal{N}(2,1)$. We also took $\theta=1$ and $\lambda=0.1$ as hyperparameters of the prior. Applying the above PMC algorithm to this sample, we see that it produces a non-degenerate 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 $n=1050$ 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 behaviour 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 later 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, the quantities quickly stabilise. The corresponding variances are not so stable over iterations, but this is to be expected, given the regular reappearance of subsamples with large variances.

Figure 2 provides an additional insight into the performances of the PMC algorithm, 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 1050 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), and contrary to the following section. 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. That we have not used this approach is to emphasize that the PMC method has no real need of the latent structure of the model to work satisfactorily.

4. Ion channels

4.1. *The stylised model*

As a realistic example of implementation of the PMC scheme, we consider here a formalised version of the ion channel model of Hodgson (1999). We refer the reader to this paper, as

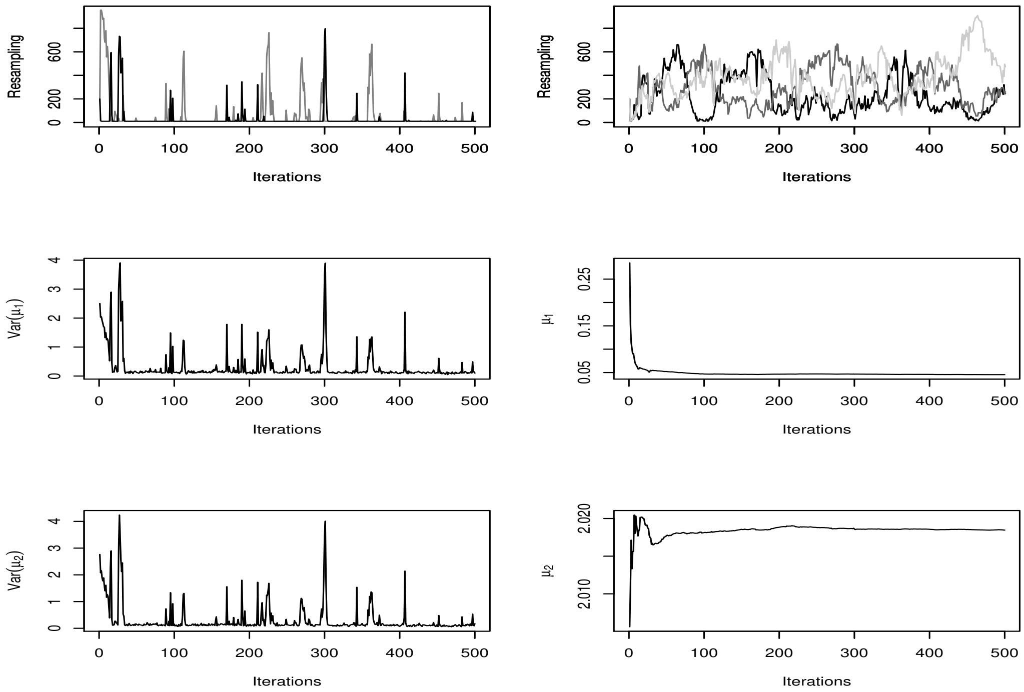

Fig. 1. Performances of the mixture PMC algorithm: *(upper left)* Number of resampled points for the variances $v_1 = 5$ (darker) and $v_2 = 2$; *(upper right)* Number of resampled points for the other variances, $v_3 = 0.1$ is the darkest one; *(middle left)* Variance of the simulated μ_1's along iterations; *(middle 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.

well as to Ball *et al.* (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 *formalised* aspect on 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 modelling of ion channel kinetics, while of importance as shown by Ball *et al.* (1999) and Hodgson and Green (1998), are not addressed by the present paper. 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.

Consider, thus, 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 $(j = 1, 2, \ldots)$ such

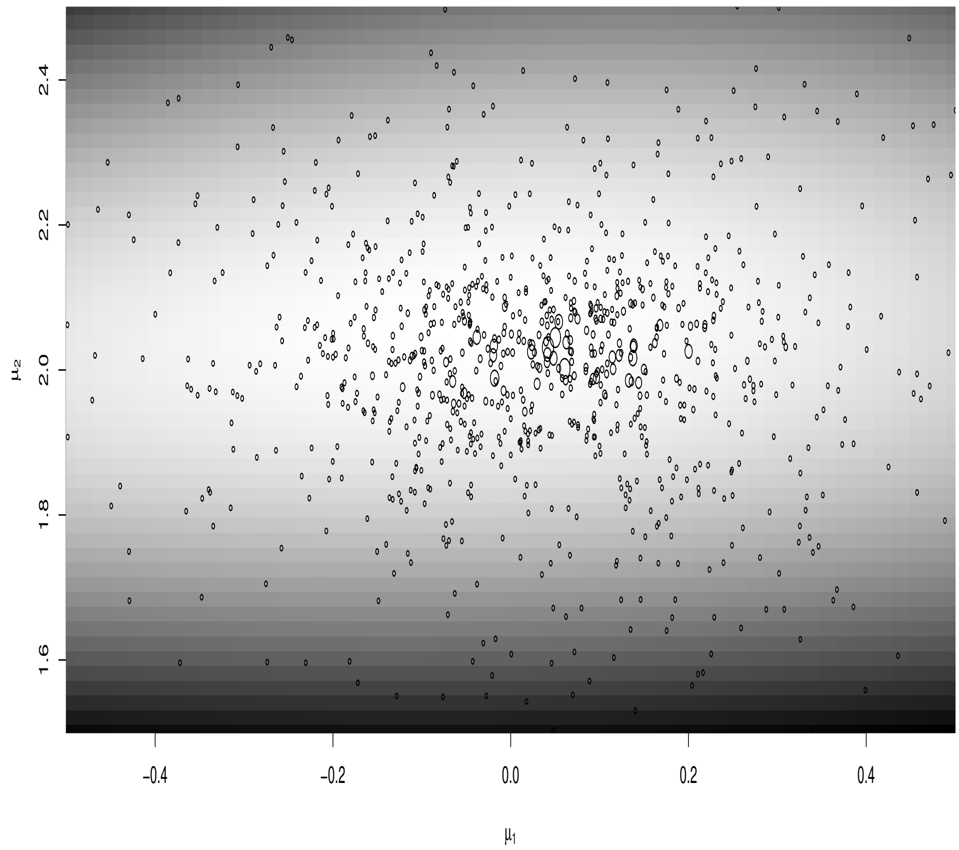

Fig. 2. Representation of the log-posterior distribution via grey levels *(darker stands for lower and lighter for higher)* and of a weighted sample of means. *(The weights are proportional to the surface of the circles.)*

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 provides a simulated sample of size 4000 from this model.

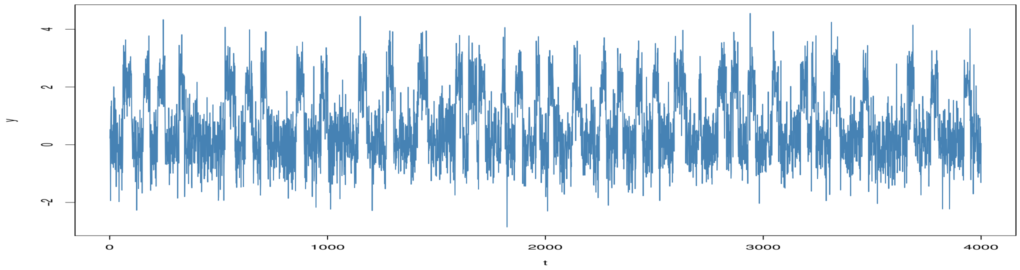

Fig. 3. Simulated sample of size 4000 from the ion channel model

A first modification of Hodgson's (1999) ion channel model is introduced at this level: we 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. The reasons for this change are that

(a) the true durations of the Gamma process are not identifiable;
(b) this model is a straightforward generalisation of the hidden Markov model where the jumps do occur at integer times (see Ball *et al.*, 1999, or Carpenter *et al.*, 1999). A natural generalisation of the geometric duration of the hidden Markov model is a negative binomial distribution, $\mathcal{N}eg(s, \omega)$, which is very close to a Gamma density $\mathcal{G}a(s+1, -\log(1-\omega))$ (up to a constant) for s small. Indeed, the former is approximately

$$\frac{d^s}{s!}(1-\omega)^d \left(\frac{\omega}{1-\omega}\right)^s$$

while the later is

$$\frac{d^s}{s!}(1-\omega)^d\{-\log(1-\omega)\}^{s+1}$$

(The simulations detailed below were also implemented using a negative binomial modelling, leading to very similar results in the restoration process.)
(c) inference on the d_j's given $(x_t)_{1\leq t\leq T}$ involves an extra level of simulations, even if it can be easily implemented via a slice sampler, as long as we do not consider the possibility of several jumps between two integer observational times. (This later possibility is actually negligible for the datasets we consider.); *and*
(d) the replacement of d_j by its integral part does not strongly modify the likelihood.

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 of Hodgson's (1999) model 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 $\mathbb{R}^+$. The reasons for this modification are that

(a) the hidden Markov process has geometric switching times, which correspond to exponential durations. A natural extension is to consider that the durations of the stays within each state (or *régime*) can be represented as the cumulated duration of s_i exponential stays, with s_i an unknown integer, which exactly corresponds to gamma durations. This representation thus removes the need to call for a level of variable dimension modelling. Carpenter et al. (1999) and Hodgson and Green (1999) use a different approach, based on the replication of the "open" and "closed" sets into several states, to approximate the semi-Markov model.
(b) the following simulations show that the parameters s_0 and s_1 are strongly identified by the observables $(y_t)_{1\le t\le T}$;
(c) the prior information on the parameters s_0 and s_1 is most likely to be sparse and thus a uniform prior is less informative than a Gamma prior when S is large; *and*
(d) the use of a finite support prior allows for the computation of the normalising 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 third 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 régimes as the data becomes potentially more volatile.

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

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

Figure 4 illustrates the dependences induced by this modelling on a DAG.

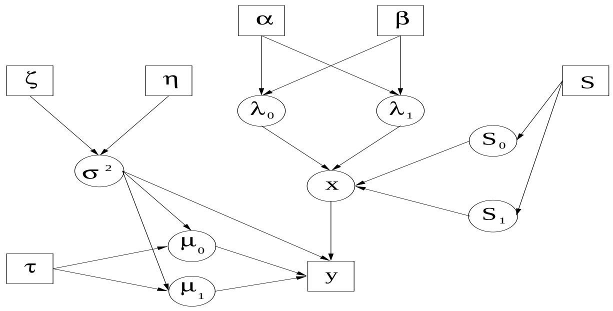

Fig. 4. DAG representation of the probabilistic dependences in the Bayesian ion channel model.

This formalised ion channel 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 $(x_t)_{1\le t\le T}$, 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 $(y_t)_{1\le t\le T}$, as was done in the hidden Markov model by, e.g., Cappé *et al.* (2003). Note also that, as opposed to Hodgson (1999), we use the *saturated* missing data representation of the model via $\mathbf{x}$ to avoid the complication of using reversible jump techniques for which PMC algorithms are more difficult to implement.

4.2. Population Monte Carlo for ion channel models

Our choice of proposal function is based on the availability of closed form formulas for the hidden Markov model. We thus 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 parameters

$$\omega = (\mu_0, \mu_1, \sigma, \lambda_0, \lambda_1, s_0, s_1)$$

constructed via the forward–backward formula: see, e.g., Cappé *et al.* (2003) for details. The simulation of the parameters 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) also consider the ion channel model in their particle filter paper, with the differences that they replace the semi-Markov structure with an approximative hidden Markov model with more than 2 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 (1998) similarly compared several hidden Markov model with duplicated "open" and "closed" states. Ball *et al.* (1999) also rely on a hidden Markov modelling with missing observations.

The subsequent use of importance sampling bypasses the exact simulation of the hidden process $(x_t)_{1\le t\le T}$ and thus avoids the recourse to variable dimension models and to more sophisticated tools like reversible jump MCMC. This *saturation* of the parameter space by the addition of the whole indicator process $(x_t)_{1\le t\le T}$ is obviously more costly in terms of storage, but it provides unrestricted moves between configurations of the process $(x_t)_{1\le t\le T}$. Since 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 rendering, where I denotes the number of iterations, T being used for the number of observations:

Ion channel PMC

Step 0: Initialisation

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

(i) generate $(\omega^{(j)}, \mathbf{x}_-^{(j)}) \sim \pi(\omega) \times \pi_H(\mathbf{x}|\mathbf{y}, \omega^{(j)})$
(ii) 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 $j = 1, \ldots, n$,

(i) generate $(\omega^{(j)}, \mathbf{x}_+^{(j)}) \sim \pi(\omega|\mathbf{y}, \mathbf{x}_-^{(j)}) \times \pi_H(\mathbf{x}|\mathbf{y}, \omega^{(j)})$
(ii) 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 take $\mathbf{x}_-^{(j)} = \mathbf{x}_+^{(j)}$ $(j = 1, \ldots, n)$.

The justification for the weights ϱ_j used in the above algorithm 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 $\pi_H(\mathbf{x}|\mathbf{y}, \omega^{(j)})$. The normalising factor of the ϱ_j's converges to the correct constant by the law of large numbers.

4.3. Normalising constants

Let us stress the specificity of the PMC method in terms of normalising constants: $\pi(\omega|\mathbf{y}, \mathbf{x})$ is available in closed form (see below in Section 4.4), including its normalising 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 normalising constant, by virtue of the forward–backward formula. The only difficulty in the ratio

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

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

$$\pi(\omega, \mathbf{x}|\mathbf{y}) \propto \pi(\omega)\, f(\mathbf{y}|\mathbf{x}, \omega)\, f(\mathbf{x}|\omega)\,. \tag{4}$$

and normalise the ϱ_j's by their sum. The foremost feature of this reweighting is that the normalising constant missing in (4) 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 (i) the points encompassing both the parameters ω and the latent data $\mathbf{x}$, and (ii) the distribution $\pi(\omega, \mathbf{x}|\mathbf{y})$ being available in closed form.

4.4. Simulation details

Figure 5 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. As obvious from the picture, the fit is quite good.

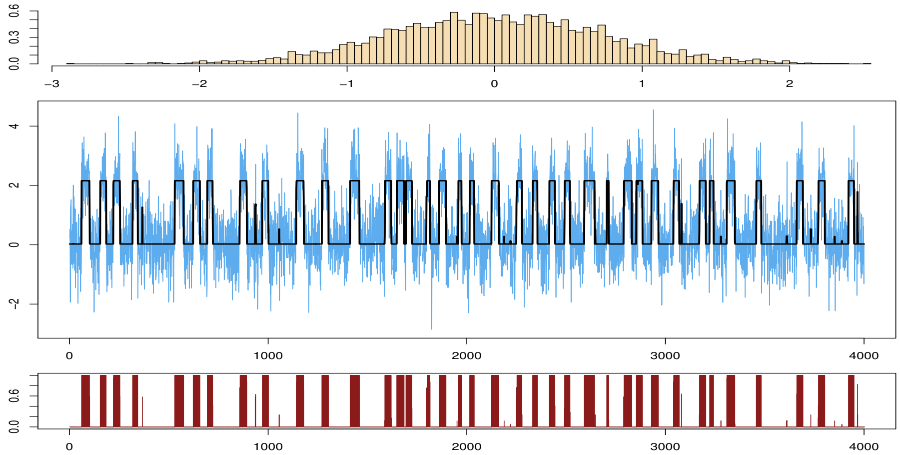

Fig. 5. (top) Histograms of residuals after fit by averaged μ_{x_t}; (middle) Simulated sample of size 4000 against fitted averaged μ_{x_t}; (bottom) Probability of allocation to first state for each observation

The unobserved Gamma process is distributed as

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

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 [corresponding to the geometric mean], v_i the total sojourn duration in state i [corresponding to the arithmetic mean]. (This is based on the assumption of no censoring, made in Section 4, 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 are the same as in Hodgson (1999).

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

$$\left[\frac{\Delta_i}{(\beta+v_i)^{n_i}}\right]^{s_i} \frac{\Gamma(n_i s_i+\alpha)}{\Gamma(s_i)^{n_i}} \tag{5}$$

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 to the Gamma function leads to (5) being equivalent to $\sqrt{n}/\sqrt{s}^n$.

Figure 6 gives the histograms of the posterior distributions of the various parameters of ω without reweighting by the importance sampling weights ϱ_j. 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) is not well identified. This is to be expected, given that we only observe a few realisations of the underlying gamma distribution, and this with added noise since 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.

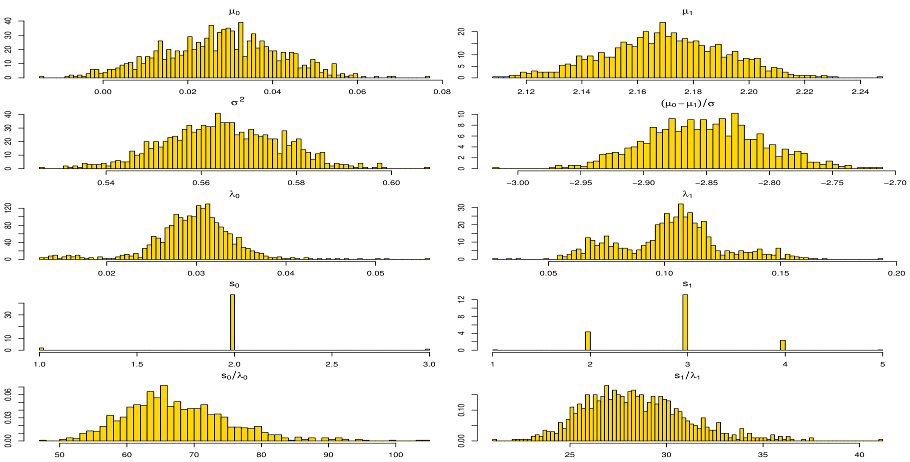

Fig. 6. Histograms of the samples produced by PMC, before resampling

4.5. Degeneracy

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

An incentive of using PMC in a static setting is thus 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 kind 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 behaviour 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 7 gives for instance the sequence of the number of points that matter at each iteration, out of $n = 1000$ 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 stabilisation in either the number of relevant points or the variance of the corresponding weights, the later being far from exhibiting a stabilisation as the number of iterations increases. Some more rudimentary signals can be considered though, like the stabilisation of the fit in Figure 8. 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.

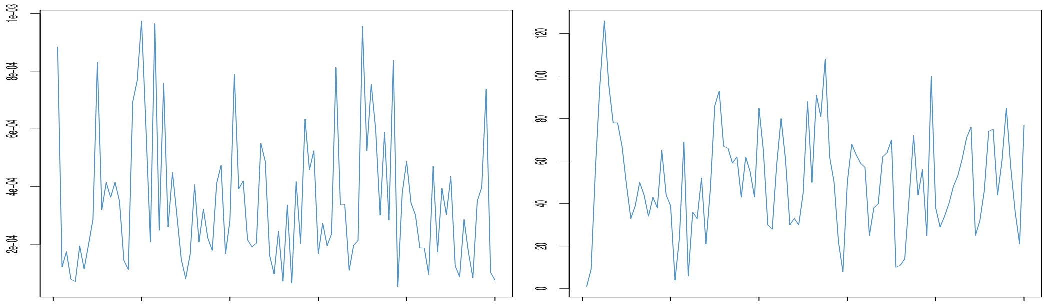

Fig. 7. (left) Variance of the weights ϱ_j along 100 iterations, and (right) Number of points with descendants along 100 iterations, for a sample of 4000 observations and 1000 points.

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 9 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 Hastings–Metropolis algorithms

As mentioned above, the 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:

MCMC Algorithm

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

(a) Generate $\omega^{(i)} \sim \pi(\omega|\mathbf{y}, \mathbf{x}^{(i-1)})$

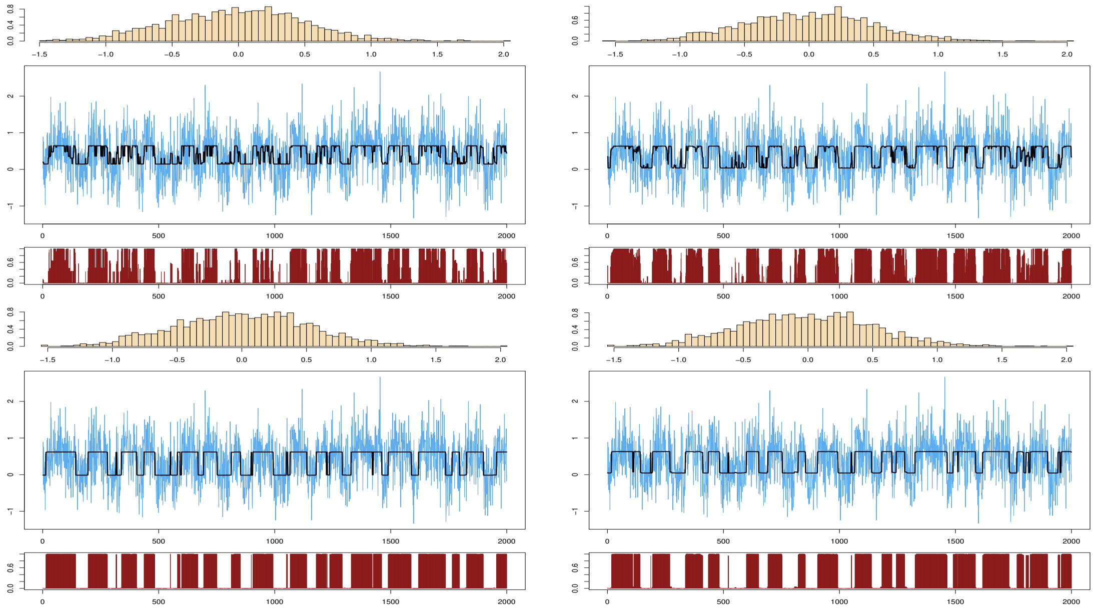

Fig. 8. Successive fits of PMC iterated by the weight-resample algorithm for 2000 observations and $n = 2000$ points, for (clockwise starting from top left) 1, 2, 5 and 10 iterations. (See the caption of Fig. 5 for a description of the displayed quantities.)

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

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

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 converges very quickly 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 durations within each state. To overcome this degeneracy of the sample, we had paradoxically to resort to a *sequential* implementation as follows: noticing that the degeneracy is only occurring 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 $\min(s_0/\lambda_0, s_1/\lambda_1)$] till the whole sample is included. The results provided in Figures 10–12 were obtained following this scheme.

For conciseness' sake, we did not reproduce the history of the allocations $\mathbf{x}$ over the iterations. The corresponding graph shows a very stable history with hardly any change, except on a few boundaries. Note the connected strong stability in the number of switches in Figure 11 *(right)*. [The cumulated means on the rhs of Figure 11 indicate that more

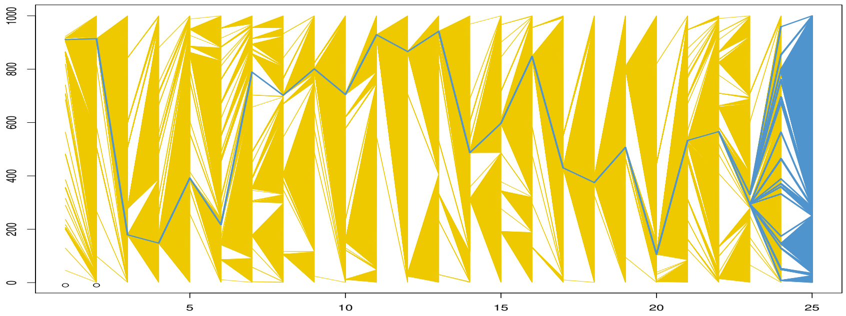

Fig. 9. Representation of the sequence of descendants (lighter grey) and ancestors (darker grey) along iterations through bars linking a given ancestor and all its descendants (light grey) or a given point and its ancestor (darker grey). In the simulation corresponding to this graph, there were 4000 observations and 1000 points.

iterations of the MCMC sampler would have been necessary but our purpose is to illustrate the proper behaviour of this sampler, provided the initialisation is adequate.]

Attempts with very mixed datasets as the one used in Figure 8 were much less successful since, even with 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, i.e., a poor mixing ability. This is further demonstrated by the following experiment: when starting the above sampler from $n = 1000$ points obtained by running PMC 20 times, the sampler always produced a satisfactory solution with two clearcut states and no degeneracy. Figure 12 compares the distributions of the PMC points and the MCMC samples via a qq-plot and shows there is very little difference between both. The same behaviour is shown by a comparison of the allocations (not represented here). This indicates that the MCMC algorithm does not lead to a better exploration of the parameter space.

For a fairly mixed dataset of 2000 observations corresponding to Figure 8, while the MCMC algorithm initialised at random could not avoid degeneracy, a preliminary run of PMC produced stable allocations to two states, as shown in Figure 13 by the fit for both PMC and MCMC samples: they are indistinguishable, even though the qq-plots in Figure 14 indicate different tail behaviours.

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

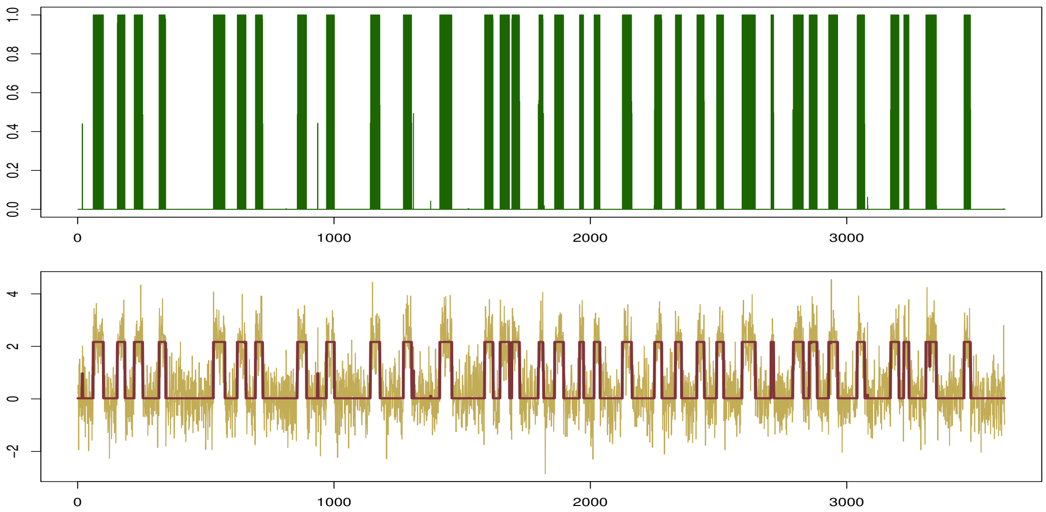

Fig. 10. Representation of a dataset of 3610 simulated values, along with the average fit *(bottom)*, and average probabilities of allocation to the upper state *(top)*. This fit was obtained using a sequential tuning scheme and 5000 MCMC iterations in the final run.

5. Conclusion

The above developments have confirmed Chopin's (2002) realisation that PMC is a useful tool in static—as opposed to sequential, rather than dynamic—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 study which scheme brings the most information about the distribution of interest. The most promising avenue seems however 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 paper 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.

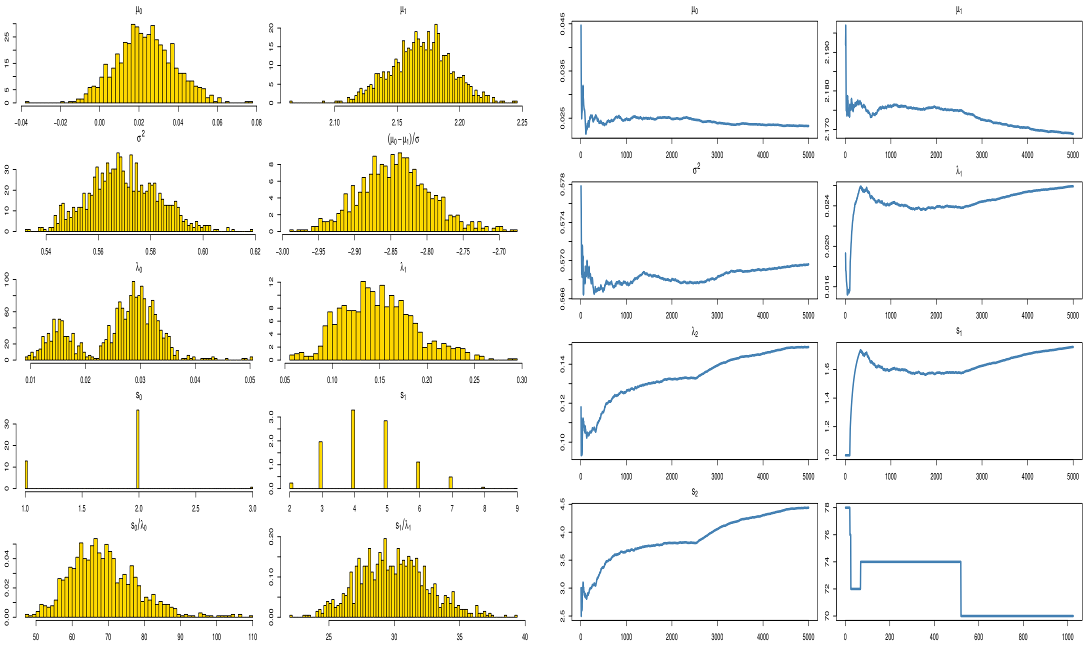

Fig. 11. Details of the MCMC sample for the dataset of Figure 10: *(left)* histograms of the components of the MCMC sample and *(right)* cumulative averages for the parameters of the models and evolution of the number of switches *(lower right graph)*.

Acknowledgments

The authors are grateful to the Friday lunch discussion group at Institut Henri Poincaré for their input. Comments from Nicolas Chopin and Gareth Roberts were also particularly helpful. Earlier conversations with Christophe Andrieu and Arnaud Doucet in Cambridge have non-coincidental connections with this work and are most sincerely acknowledged.

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. *Proc. Royal Society London* A **455**, 2879–2932.

Baum, L.E. and Petrie, T. (1966) Statistical inference for probabilistic functions of finite state Markov chains. *Annals Math. Statist.* **37**, 1554–1563.

Berzuini, C., Best, N., Gilks, N.G., and Larizza, C. (1997) Dynamic conditional independence models and Markov Chain Monte Carlo methods. *JASA* **92**, 590–599.

Billio, M., Monfort, A. and Robert, C.P. (1999) Bayesian estimation of switching ARMA Models. *J. 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. *JRSS B* (to appear).

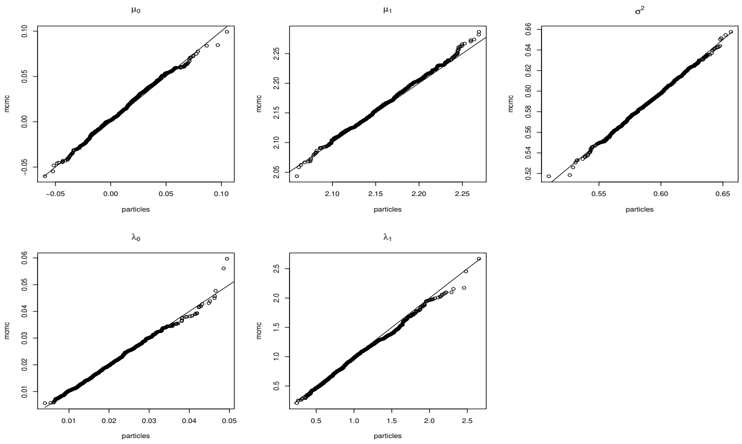

Fig. 12. QQ-plot comparing the distribution of the sample with the distribution of the MCMC sample obtained after 5000 iterations started at the PMC points.

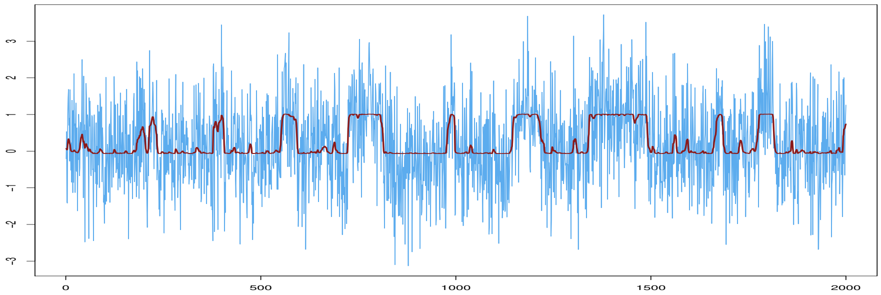

Fig. 13. Simulated dataset of 2000 points with superimposed fits by PMC and the MCMC samples, the later being initialised by PMC. Both fits are indistinguishable.

Carpenter, J., Clifford, P., and Fernhead, P. (1999) Building robust simulation based filters for evolving data sets. Technical report, Department of Statistics, Oxford University.

Celeux, G., Hurn, M. and Robert, C.P. (2000) Computational and inferential difficulties with mixture posterior distributions. *J. Amer. Statist. Assoc.* **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*. Springer-Verlag, New York.

Gilks, W.R., and Berzuini, C. (2001) Following a moving target–Monte Carlo inference for dynamic Bayesian models. *JRSS B* **63**(1), 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**(4), 711–732.

Haario, H., Saksman, E. and Tamminen, J. (1999) Adaptive proposal distribution for random walk Metropolis algorithm. *Computational Statistics* **14** 375–395.

Haario, H., Saksman, E. and Tamminen, J. (2001) An adaptive Metropolis algorithm. *Bernoulli* **7**(2), 223–242.

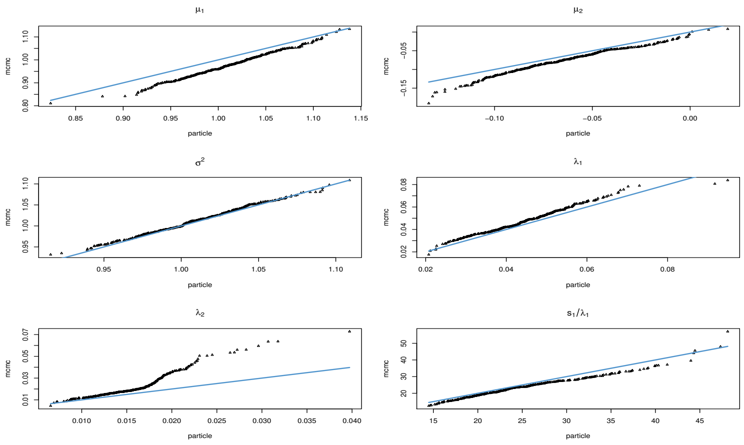

Fig. 14. QQ-plot comparing the distribution of the PMC sample with the distribution of the MCMC sample obtained after 5000 iterations started at one point.

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. *JRSS B* **61**(1), 95–114.

Hodgson, M.E.A. and Green, P.G. (1999) Bayesian choice among Markov models of ion channels using Markov chain Monte Carlo. *Proc. Royal Society London* A **455**, 3425–3448.

Kim, S., Shephard, N. and Chib, S. (1998) Stochastic volatility: Likelihood inference and comparison with ARCH models. *Rev. Econom. Stud.* **65**, 361–393.

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

Liu, J.S., Liang, F. and Wong, W.H. (2001) A theory for dynamic weighting in Monte Carlo computation. *JASA* **96**, 561–573.

MacEachern, S.N. and Peruggia, M. (2000) Importance link function estimation for Markov Chain Monte Carlo methods. *J. Comput. Graph. Statist.* **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*, J.M. Bernardo, M.J. Bayarri, J.O. Berger, A.P. Dawid, D. Heckerman, A.F.M. Smith and M. West (Eds.), Oxford University Press (to appear).

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

Oh, M.S. and Berger, J.O. (1992) Adaptive importance sampling in Monte Carlo integration. *J. Statist. Comput. Simul.* **41**, 143–168.

Oh, M.S. and Berger, J.O. (1993) Integration of multimodal functions by Monte Carlo importance sampling. *JASA* **88**, 450–456.

Pitt, M.K. and Shephard, N. (1999) Filtering via simulation: auxiliary particle filters. *JASA* **94**, 1403–1412, 1403–1412

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

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

Stavropoulos, P. and Titterington, D.M. (2001) Improved particle filters and smoothing. In *Sequential MCMC in Practice*, Doucet, A., deFreitas, N., and Gordon, N. (Eds), Springer-Verlag, New York.

Warnes, G.R. (2001) The Normal Kernel Coupler: An adaptive Markov Chain Monte Carlo method for efficiently sampling from multi-model distributions. Tech. report 395, University of Washington.