

On variance stabilisation in Population Monte Carlo by double Rao-Blackwellisation

Alessandra Iacobucci, Jean-Michel Marin, Christian P. Robert

▶ **To cite this version:**

Alessandra Iacobucci, Jean-Michel Marin, Christian P. Robert. On variance stabilisation in Population Monte Carlo by double Rao-Blackwellisation. Computational Statistics and Data Analysis, 2010, 54 (3), pp.698-710. 10.1016/j.csda.2008.09.020 . inria-00260141

HAL Id: inria-00260141
https://inria.hal.science/inria-00260141

Submitted on 3 Mar 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

On variance stabilisation by double Rao-Blackwellisation

Alessandra Iacobucci — Jean-Michel Marin — Christian Robert

N° ????

February 2008

Thème COG



ISSN 0249-6399 ISRN INRIA/RR--????--FR+ENG

INRIA FUTURS

On variance stabilisation by double Rao-Blackwellisation

Alessandra Iacobucci* , Jean-Michel Marin† , Christian Robert‡

Thème COG — Systèmes cognitifs
Projets SELECT

Rapport de recherche n° ???? — February 2008 — 16 pages

Abstract: Population Monte Carlo has been introduced as a sequential importance sampling technique to overcome poor fit of the importance function. In this paper, we compare the performances of the original Population Monte Carlo algorithm with a modified version that eliminates the influence of the transition particle via a double Rao-Blackwellisation. This modification is shown to improve the exploration of the modes through an large simulation experiment on posterior distributions of mean mixtures of distributions.

Key-words: Importance Sampling mixture, adaptive Monte Carlo, Population Monte Carlo, multimodality, mixture of distributions, random walk.

* CEREMADE, Université Paris Dauphine
† INRIA Saclay, Projet SELECT, Université Paris-Sud & CREST, INSEE
‡ CEREMADE, Université Paris Dauphine & CREST, INSEE

Unité de recherche INRIA Futurs
Parc Club Orsay Université, ZAC des Vignes,
4, rue Jacques Monod, 91893 ORSAY Cedex (France)
Téléphone : +33 1 72 92 59 00 — Télécopie : +33 1 60 19 66 08

Schémas *Population Monte Carlo* et double *Rao-Blackwellisation*

Résumé : Les algorithmes *Population Monte Carlo* (PMC) sont des schémas d'échantillonnage préférentiel adaptatifs qui ont été introduits afin d'ajuster la distribution d'importance à la loi cible. Dans cet article, nous comparons les performances du schéma PMC classique à celles d'une version modifiée qui diminue la variabilité des poids d'importance en utilisant une technique de double *Rao-Blackwellisation*. Il est montré que cette modification permet d'améliorer significativement les capacités d'exploration de l'algorithme dans le cas de lois cibles multi-modales.

Mots-clés : Échantillonnage préférentiel, mélanges de lois, méthodes de Monte-Carlo adaptatives, marches aléatoires.

1 Introduction

When Cappé et al. (2004) introduced Population Monte Carlo (or PMC) as a repeated Sampling Importance Resampling procedure, the purpose was to overcome the shortcomings of Importance Sampling (abbreviated to IS in the following) procedures, while preserving the advantages of IS compared to alternatives such as Markov Chain Monte Carlo methods, namely the possibility of developing parallel implementations, which becomes more and more important with the generalisation of multiple core machines and computer clusters, and of allowing for easy assessment of the Monte Carlo error and, correlatively, for the development of on-line calibration mechanisms.

Indeed, adaptive Monte Carlo naturally answers the difficulties of finding a "good" importance function in IS by gradually improving this importance function against a given target density, based on some form of Monte Carlo approximation. Previous simulations are used (see, e.g., Douc et al., 2007a,b) to modify proposal distributions represented as mixtures of *fixed* proposals in order to increase the weights of the most appropriate components. When the proposal is inspired from random walk structures as in Metropolis–Hastings algorithms and when the update of those weights is too crude, as in Cappé et al. (2004), the improvement is so short-sighted that multiple iterations do not increase the efficiency of the method, unless a Rao-Blackwell step is added to replace the actual proposal with the mixture proposal in the importance weight (see Douc et al., 2007a). More recently, Cappé et al. (2007) have thus replaced random walk proposals based on earlier samples with a new PMC scheme with parameterised proposals whose parameters are estimated from earlier samples, leading to a monotonic decrease in the entropic distance to the target distribution. There is indeed a myopic feature in random walk proposals, namely that, once a (first) sample has exhibited some high density regions for the target distribution, the algorithm is reluctant to allow for a wider exploration of the support of the target and it may thus miss important density regions, missing the energy to reach forward to these other regions of interest.

In this paper, we nonetheless focus on random walk proposals because those kernels are more open to complex settings than on independent proposals—the later indeed require some preliminary knowledge about the target or else an major upgrade in computing power to face a much larger number of components in the mixture. More precisely, we present an experimental assessment of the use of a so-called *double Rao-Blackwellisation scheme* towards a better exploration of the modes of the target distribution. The *second* Rao-Blackwellisation step used in this *double* Rao-Blackwellisation scheme is essentially an integration over the particles of the previous sample used in the random walk proposal. While this leads to an additional computing cost in the derivation of the importance weights, double Rao-Blackwellisation undoubtedly brings a significant improvement in the stability of those importance weights and therefore justifies (in principle) the use of this additional step. Nevertheless, since an analytic proof of this improvement brought by double Rao-Blackwellisation is too delicate to contemplate, we use an intensive Monte Carlo experiment to establish the clear improvement in the case of a posterior distribution associated with a Gaussian location mixture and a sample with outliers—in order to increase the number of modes.

The paper is organised as follows: In Section 2, we recall the basics of our population Monte Carlo algorithm and we introduce the double Rao-Blackwellisation modification. In Section 3, we motivate the choice of Gaussian mean mixtures as a worthy Monte Carlo experiment to test the mode finding abilities of both algorithm. Section 4 describes the implementation of the Monte Carlo experiment and describes the results. Section 5 contains a short discussion.

2 Population Monte Carlo

Given a target distribution with density π that is known up to a normalising constant, the grand scheme of PMC is the same as with other Monte Carlo—including MCMC—methods, namely to produce a sample that is distributed from π without resorting to direct simulation from π. Let us recall that, once a sample $(X_1, \ldots, X_N)$ is produced by sampling importance resampling (see, e.g., Robert and Casella, 2004, Marin and Robert, 2007), i.e., by first simulating $X_i \sim f(x)$, second producing importance weights $\omega_i \propto \pi(X_i)/f(X_i)$, and third resampling the X_i's by multinomial/bootstrap sampling with weights ω_i, this SIR sample provides an approximation to the target distribution π and can be used as a stepping stone towards a better approximation to π.

2.1 PMC basics

In the PMC algorithm of Cappé et al. (2004), if

$$(X_1, \ldots, X_N)$$

is a sample approximately distributed from π, it is modified stochastically using an arbitrary Markov transition kernel $q(x, x')$ so as to produce a new sample

$$(X'_1, \ldots, X'_N)$$

as $X'_i \sim q(X_i, x)$. Conducting a resampling step based on the IS weights $\omega_i = \pi(X'_i)/q(X_i, X'_i)$, we then produce a new sample

$$(\tilde{X}_1, \ldots, \tilde{X}_N)$$

that equally constitutes an approximation to the target distribution π. It is however necessary to stress that, as established in Douc et al. (2007b), repeating this scheme in an iterative manner is only of interest if samples that have been previously simulated are used to update (or adapt) the kernel $q(x, x')$: a kernel q that is fixed over iterations does not modify the statistical properties of the samples.

The choice made in Douc et al. (2007b) of an adaptive proposal kernel q represented as a mixture of fixed transition kernels q_d,

$$q_\alpha(x, x') = \sum_{d=1}^{D} \alpha_d q_d(x, x')\,, \qquad \sum_{d=1}^{D} \alpha_d = 1\,, \tag{1}$$

can improve the efficiency of the kernel in terms of deviance (or relative entropy) from the target density. To achieve such an improvement, the weights $\alpha_1, \ldots, \alpha_D$ must be tuned adaptively at each iteration of the PMC algorithm.

2.2 Weight updating

If $\alpha^t = (\alpha_1^t, \ldots, \alpha_D^t)$ denote the mixture weights at the t-th iteration of the algorithm (where $t = 1, \ldots, T$), the update of the α^t's of Douc et al. (2007b) takes advantage of the latent variable structure that underlines the mixture representation, resulting in an integrated EM (Expectation-Maximisation) scheme. In the current setting, the latent variable Z is the standard component indicator in the mixture (see, e.g., Marin et al., 2005), with values in $\{1, \ldots, D\}$ such that the joint density f of x' and z given x satisfies

$$f(z) = \alpha_z \quad \text{and} \quad f(x'|z, x) = q_z(x, x')\,.$$

The updating mechanism for the α_d's then corresponds to setting the new parameter α^{t+1} equal to

$$\arg\max_\alpha \mathbb{E}_{\pi\times\pi}^{X,X'} \left[\mathbb{E}_{\alpha^t}^{Z} \left\{\log(\alpha_Z q_Z(X, X'))|X, X'\right\}\right] = \arg\max_\alpha \mathbb{E}_{\pi\times\pi}^{X,X'} \left[\mathbb{E}_{\alpha^t}^{Z} \left\{\log(\alpha_Z)|X, X'\right\}\right]\,,$$

where the inner expectation is computed under the conditional distribution of Z for the current value of the parameter, α^t, i.e.

$$f(z|x, x', \alpha^t) = \alpha_z^t q_z(x, x') \Big/ \sum_{d=1}^{D} \alpha_d^t q_d(x, x')\,,$$

with solution

$$\alpha_d^{t+1} = \mathbb{E}_{\pi\times\pi}^{X,X'} \left[f(d|X, X', \alpha^t)\right]\,. \tag{2}$$

2.3 Single Rao–Blackwellisation updates

Given that (2) is not available in closed form, the adaptivity of the proposed procedure is achieved by approximating this actualising expression based on the sample that has been previously simulated. The implementation of the standard PMC algorithm relies on an arbitrary initial proposal μ_0 that produces a pre-initial sample

$$(X_{1,-1}, \ldots, X_{N,-1})\,,$$

with importance weights $\pi(X_{i,-1})/\mu_0(X_{i,-1})$. This initial sample allows for the derivation of a sample

$$(\tilde{X}_{1,-1}, \ldots, \tilde{X}_{N,-1})$$

approximately distributed from π, using importance sampling based on those weights. The algorithm then picks arbitrary starting weights α_d^0 in the N-dimensional simplex to produce a (truly) initial sample

$$(X_{1,0}, \ldots, X_{N,0})$$

from the mixture

$$X_{i,0} \sim \sum_{d=1}^{D} \alpha_d^0 q_d(\tilde{X}_{i,-1}, x)\,.$$

In the detail of its implementation, the production of this initial sample is naturally associated with latent variables $(Z_{i,0})_{1\le i\le N}$ that indicate from which component d of the mixture the corresponding $X_{i,0}$ has been generated ($i = 1, \ldots, n$). From this stage, Douc et al. (2007b) proceed recursively. Starting at time t from a sample

$$(X_{1,t}, \ldots, X_{N,t})\,,$$

associated with $(Z_{i,t})_{1\le i\le N}$ and with the current value of the weights $\alpha^{t,N}$, the normalised importance weights of the sample point $X_{i,t}$ is provided by

$$\bar{\omega}_{i,t} = \frac{\pi(X_{i,t})}{\sum_{d=1}^{D} \alpha_d^{t,N} q_d(\tilde{X}_{i,t-1}, X_{i,t})} \Big/ \sum_{j=1}^{N} \frac{\pi(X_{j,t})}{\sum_{d=1}^{D} \alpha_d^{t,N} q_d(\tilde{X}_{j,t-1}, X_{j,t})}\,. \tag{3}$$

To approximate the update step (2) in practice, an initial possibility is

$$\alpha_d^{t+1,N} = \sum_{i=1}^{N} \bar{\omega}_{i,t} \mathbb{1}\{Z_{i,t} = d\}\,,$$

where the sum does not need renormalisation because the $\bar{\omega}_{i,t}$ sum up to 1 (over i).

This choice is however likely to be highly variable when N is small and/or the number of components D becomes larger. To robustify this update, Cappé et al. (2007) introduce a Rao-Blackwellisation step (see, e.g., Robert and Casella, 2004, Section 4.2) that consists in replacing the Bernoulli variable $\mathbb{1}\{Z_{i,t} = d\}$ with its conditional expectation given $X_{i,t}$ and $\tilde{X}_{i,t-1}$, that is,

$$\alpha_d^{t+1,N} = \sum_{i=1}^{N} \bar{\omega}_{i,t} f\left(d \,\middle|\, X_{i,t}, \tilde{X}_{i,t-1}, \alpha^{t,N}\right)\,.$$

This replacement does not involve a significant increase in the computational cost of the algorithm, while both approximations are converging to the solution of (2) as N grows to infinity.

2.4 Double Rao-Blackwellisation

While Cappé et al. (2007) showed through two experimental examples that the above Rao-Blackwellisation step does provide a significant improvement in the stability of the PMC weights (and, thus, *in fine*, a reduction in the number of iterations), there remains an extra-source of variation in the importance weights (3), namely the dependence of $\bar{\omega}_{i,t}$ on the previous sample point (or particle) $\tilde{X}_{i,t-1}$. Even though this dependence illustrates the fact that $X_{i,t}$ is indeed generated from the mixture

$$\sum_{d=1}^{D} \alpha_d^{t,N} q_d(\tilde{X}_{i,t-1}, x)\,,$$

and is thus providing a correct IS weight, it also shows a conditioning on the result of a (random) multinomial sampling in the previous iteration that led to the selection of $\tilde{X}_{i,t-1}$ with probability $\bar{\omega}_{i,t-1}$. In other words, by de-conditioning, it is also the case that the sample point $X_{i,t}$ has been generated from the (integrated) distribution

$$\sum_{j=1}^{N} \bar{\omega}_{j,t-1} \sum_{d=1}^{D} \alpha_d^{t,N} q_d(\tilde{X}_{j,t-1}, x)\,, \tag{4}$$

when averaging over all multinomial outputs. This double averaging over both the components of the mixture and the initial sample points is the reason why we call this representation *double Rao–Blackwellisation*.

The appeal of using (4) is that not only does the averaging over all possible sample points provide a most likely stabilisation of the weights, but it also eliminates a strange feature of the original approach, namely that two identical values $x_{i,t}$ and $x_{j,t} = x_{i,t}$ could have different importance weights simply because their conditioning sample values $\tilde{X}_{i,t-1}$ and $\tilde{X}_{j,t-1}$ were different.

Naturally, the replacement of the importance sampling distribution in (3) by (4) has a cost of $\mathrm{O}(N^3)$ compared with the original $\mathrm{O}(N^2)$, but this is often negligible when compared with the cost of computing $\pi(X_{i,t})$. (We also stress that some time-saving steps could be taken in order to avoid computing all the $q_d(\tilde{X}_{j,t-1}, X_{i,t})$ by considering first the distance between $X_{j,t-1}$ and $X_{i,t}$ and keeping only close neighbours within the sum although the increase in computing time in our case did not justify the filtering.) In the following experiment, the additional cost resulting from the double Rao–Blackwellisation does not induce a considerable upsurge in the computing time, even though it is not negligible. We indeed found an increase in the order of three to five times the original computing time for the same number N of sampling values. This obviously fails to account for the faster stabilisation of the IS approximation resulting from using double Rao–Blackwellisation. Note also that double Rao–Blackwellisation does not remove the need to sample the $\tilde{X}_{j,t-1}$'s: indeed, when simulating each new $X_{i,t}$ from (4), we need to first select which $\tilde{X}_{j,t-1}$ is going to be conditioned upon, then second determine which component is to be used.

3 The Gaussian mean mixture benchmark

As benchmark for our Monte Carlo experiment, we consider the case of a one-dimensional Gaussian mean mixture distribution, namely

$$x_1, \ldots, x_n \sim p\mathcal{N}(\mu_1, \sigma_1^2) + (1-p)\mathcal{N}(\mu_2, \sigma_1^2)\,, \tag{5}$$

with p, σ_1^2 and σ_2^2 known and $\theta = (\mu_1, \mu_2)$ being the parameter of the model, as in Marin et al. (2005). Using a flat prior on θ within a square region, we are thus interested in simulating from the posterior distribution associated with a given sample $(x_1, \ldots, x_n)$. The appeal of this example is that it is sufficiently simple to allow for an explicit characterisation of the attractive points for the adaptive procedure, being of dimension two, while still illustrating the variety of situations found in more realistic applications. In particular as already explained in Marin et al. (2005), the model contains at least one attractive point that does not correspond to the global minimum of the entropy criterion as well as some regions of attraction that can eventually lead to a failure of the algorithm when the data $(x_1, \ldots, x_n)$ is not simulated from the above mixture but by clumps that induce more modes in the posterior. Figures 1 and 2 illustrate quite well the diversity of posterior features when changing the repartition of the sample $(x_1, \ldots, x_n)$. We stress the fact that those samples, called artificial samples, are not resulting from simulations from (5) but from simulations from a normal mixture with five equal and well-separated components centred in $\mu_1 = 0$ and $\pm\mu_2$, $\pm 2\mu_2$, and with variances 0.1. This choice was made in order to increase the potential number of modes on the posterior surface when modelling the data with (5).

4 Monte Carlo experiment

Given the target distribution defined by (5), we want to study the improvement brought by the double Rao-Blackwellisation (4) in terms of mode degeneracy, as well as to ascertain the additional cost of using double Rao-Blackwellisation. We therefore study the performance of both single and double Rao-Blackwellisation PMC over a whole range of samples, with various values of n and of the parameters p, μ_2 and σ_2 (since we can always use $\mu_1 = 0$ and $\sigma_1 = 1$ without loss of generality).

4.1 Automatic mode finding and classification

A first difficulty, when building this experiment, is to determine the number of modes on the posterior surface for the data at hand. We achieve this goal by first discretising the parameter space in (μ_1, μ_2), and then recovering the modal points by a brute-force search for maxima and saddle-points over the grid and then identifying their basins of attraction. (The algorithm is based on multidirectional calls to the function `turnpoints()` of the `pastecs` package.) This is obviously prone to overlook some small local modes but it also allows for a quick identification of the modal regions and of their exploration by the PMC algorithm.

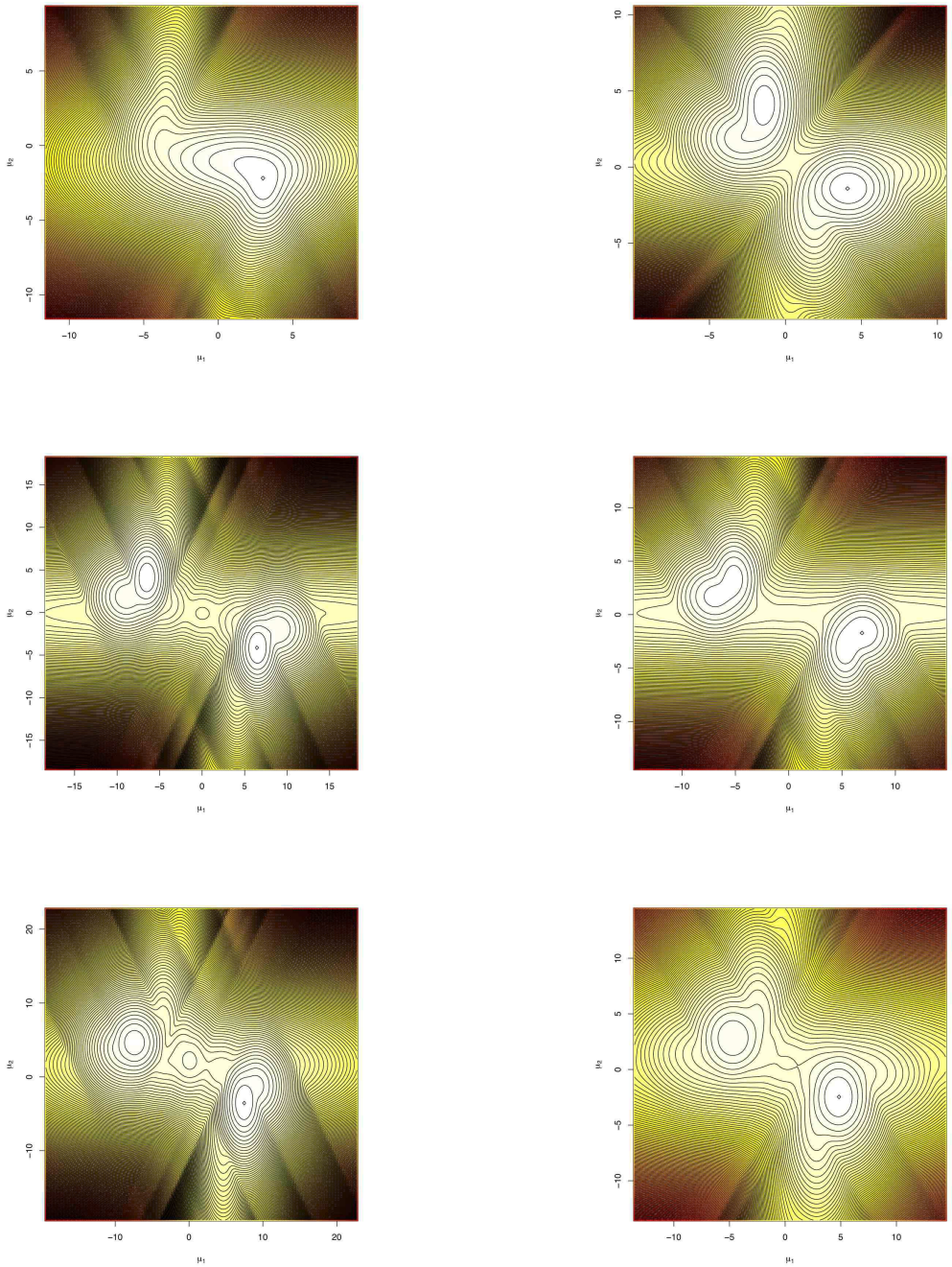


Figure 1: Series of posterior distributions associated with (5) represented as level sets for various artificial samples $(x_1, \ldots, x_n)$ and different values of n, μ_1, μ_2, σ_1^2 and σ_2^2.

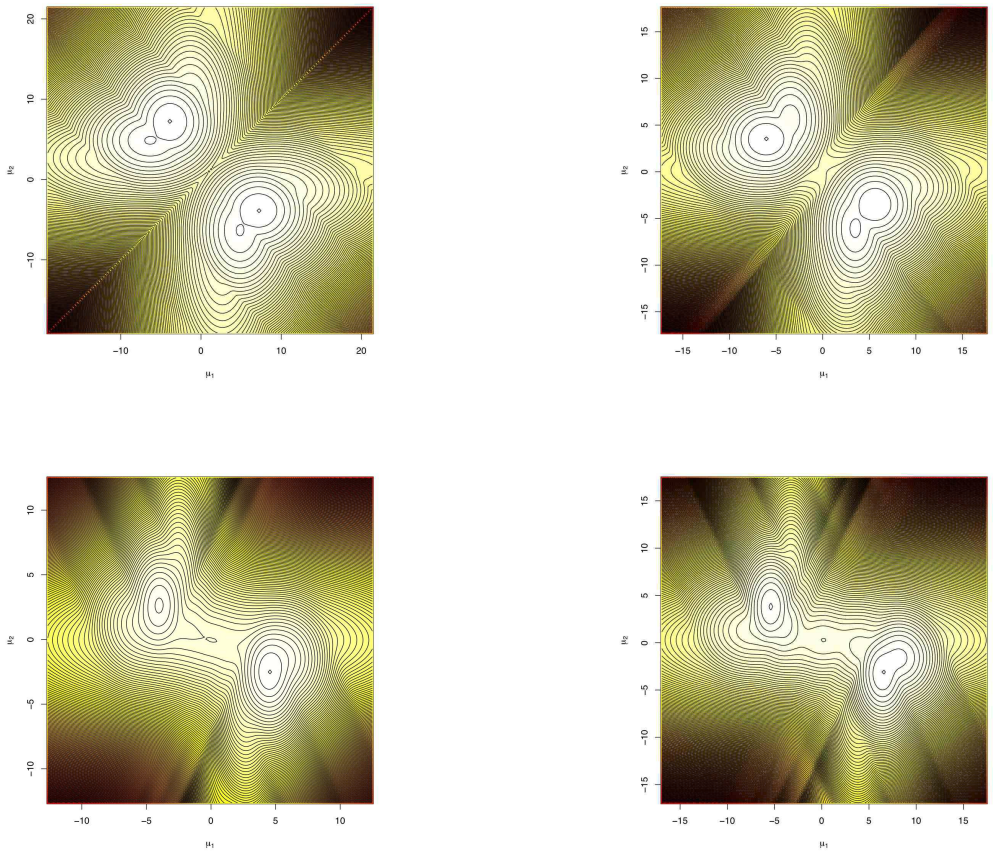


Figure 2: *(Fig. 1 cont.)* Series of posterior distributions associated with (5) represented as level sets for various artificial samples $(x_1, \ldots, x_n)$ and different values of n, μ_1, μ_2, σ_1^2 and σ_2^2.

Table 1 illustrates the results of this procedure for a range of values of (n, p, μ_2, σ_2). When conditioning upon (μ_2, σ_2) the number of modes is increasing in μ_2 (since the simulated samples include five clusters that are better separated) and slightly decreasing in σ_2 (for apparently the same reason). A similar table varying upon the pair (n, p) does not show much variation in the number of modes (which does make sense, since only the relative magnitudes of the modes are changing).

(σ_2, μ_2)	**1.0**	**1.5**	**2.0**	**2.5**	**3.0**	**3.5**	**4.0**	**4.5**	**5.0**
1.0	2.053	2.184	2.450	2.863	3.265	3.524	3.641	3.707	3.742
	(0.477)	*(0.486)*	*(1.273)*	*(1.085)*	*(0.895)*	*(0.754)*	*(0.706)*	*(0.668)*	*(0.627)*
1.5	2.037	2.124	2.135	2.165	2.294	2.494	2.731	2.958	3.182
	(0.707)	*(0.362)*	*(0.404)*	*(0.565)*	*(0.580)*	*(0.668)*	*(0.696)*	*(0.729)*	*(0.706)*
2.0	2.026	2.067	2.121	2.059	2.083	2.132	2.259	2.546	2.807
	(0.862)	*(0.337)*	*(0.359)*	*(0.342)*	*(0.350)*	*(0.369)*	*(0.505)*	*(0.648)*	*(0.636)*
2.5	1.976	2.024	2.107	2.067	2.073	2.184	2.472	2.757	2.893
	(1.027)	*(0.354)*	*(0.344)*	*(0.302)*	*(0.313)*	*(0.434)*	*(0.570)*	*(0.533)*	*(0.500)*
3.0	1.993	1.982	2.073	2.203	2.137	2.361	2.743	2.933	3.040
	(1.147)	*(0.359)*	*(0.329)*	*(0.599)*	*(0.469)*	*(0.580)*	*(0.572)*	*(0.518)*	*(0.486)*
3.5	1.979	1.964	2.076	2.186	2.367	2.662	2.973	3.181	3.354
	(1.315)	*(0.432)*	*(0.415)*	*(0.519)*	*(0.938)*	*(0.844)*	*(0.712)*	*(0.678)*	*(0.719)*
4.0	1.961	1.939	2.078	2.209	2.412	2.871	3.214	3.466	3.752
	(1.469)	*(0.507)*	*(0.490)*	*(0.500)*	*(0.851)*	*(0.922)*	*(0.807)*	*(0.806)*	*(0.860)*
4.5	1.858	1.893	2.046	2.220	2.524	3.101	3.418	3.777	4.073
	(1.471)	*(0.504)*	*(0.445)*	*(0.540)*	*(0.832)*	*(1.034)*	*(0.898)*	*(0.910)*	*(0.901)*
5.0	1.698	1.886	2.063	2.273	2.678	3.187	3.673	4.046	4.295
	(1.247)	*(0.605)*	*(0.566)*	*(0.612)*	*(1.128)*	*(1.013)*	*(1.033)*	*(0.970)*	*(0.867)*

Table 1: Average number (and standard deviation) of identified modes on a collection of 1470 samples.

The code for both implementing both versions of PMC and for evaluating their mode-finding abilities was written in R (R Development Core Team, 2006) and is available as
`http:\\www.ceremade.dauphine.fr\~xian\2RB.R`. Further documentation is available as
`http:\\www.ceremade.dauphine.fr\~xian\2RB.R.pdf`.

4.2 Output

The experiment was run on 4860 samples on seven different machines for a total of 238140 simulations, using the same machine for both single and double Rao-Blackwellised versions at a given value of the parameters in order to keep the CPU comparison sensible. As seen from Table 2, the CPU time required to implement the double Rao-Blackwellised scheme is five to three time higher than the original PMC scheme, the additional time decreasing with n. (This CPU time does not include the mode finding and storing steps that are required for the comparison. It only corresponds to the execution of a regular PMC run with 10 iterations.) Note also that the increase in computing time is certainly less than linear. Both single and double Rao-Blackwellisation PMC samplers were used on the same samples, with different values of p, σ_2 and μ_2.

Turning now to the central tables, Tables 3–6, we can see that the influence of σ_2 on the detection of the modes is relatively limited, in opposition to the influence of μ_2. As μ_2 increases (recall that $\mu_1 = 0$), a larger fraction of modes gets undetected. Both after 5 and 10 iterations of PMC, the performances of the double Rao–Blackwellised scheme are superior to those of the single Rao–Blackwellisation in terms of mode detection, by a factor of about 5%. Running the PMC algorithm longer clearly has a downward impact on the number of detected modes, even though this impact is quite limited. It however points out the major issue that importance sampling algorithms like PMC are mostly unable to recover lost modes. Another interesting feature is that single Rao–Blackwellisation suffers more from the loss of modes between 5 and 10 iterations, compared with double Rao–Blackwellisation. As expected, the later is more robust because of the average over all past values in the PMC sample. Figure 3 presents the evolution of the capture rate for the single and the double Rao–Blackwellisations as a function of μ_2 after 5 and 10 iterations, obtained by averaging Tables 3–6 over σ_2.

If we consider instead the output of this simulation experiment decomposed by the values of n and p, in Tables 7–10, a main feature is the quick deterioration in the exploration of the modes as n increases. This is obviously to be expected given that the more observations, the steeper the slopes of the posterior surfaces.

n	$t_{1\mathrm{RB}}$	$t_{2\mathrm{RB}}$
20	3.60	15.59
	(0.52)	*(1.12)*
30	3.65	15.51
	(0.52)	*(1.13)*
40	3.68	15.59
	(0.52)	*(1.13)*
50	3.72	15.61
	(0.52)	*(1.13)*
100	3.89	15.74
	(0.53)	*(1.16)*
500	5.13	17.19
	(0.62)	*(1.26)*
1000	6.85	18.92
	(0.74)	*(1.42)*

Table 2: Overall mean CPU times and their standard error (in seconds) *vs.* sample size. Each value is obtained by averaging over 4860 generated samples, for a range of parameters ($p = .1, \ldots, .8$, $\mu_2 = 1, \ldots, 5$, $\sigma_2 = 1, \ldots, 5$).

(σ_2, μ_2)	**1.0**	**1.5**	**2.0**	**2.5**	**3.0**	**3.5**	**4.0**	**4.5**	**5.0**
1.0	0.743	0.714	0.631	0.536	0.452	0.409	0.399	0.386	0.387
	(0.260)	*(0.256)*	*(0.254)*	*(0.259)*	*(0.232)*	*(0.212)*	*(0.216)*	*(0.210)*	*(0.209)*
1.5	0.697	0.697	0.612	0.573	0.515	0.472	0.425	0.386	0.350
	(0.272)	*(0.255)*	*(0.226)*	*(0.208)*	*(0.179)*	*(0.175)*	*(0.157)*	*(0.145)*	*(0.121)*
2.0	0.653	0.714	0.614	0.564	0.552	0.521	0.495	0.445	0.397
	(0.273)	*(0.257)*	*(0.226)*	*(0.182)*	*(0.173)*	*(0.151)*	*(0.147)*	*(0.147)*	*(0.131)*
2.5	0.666	0.718	0.602	0.563	0.540	0.509	0.452	0.394	0.378
	(0.288)	*(0.256)*	*(0.220)*	*(0.181)*	*(0.155)*	*(0.147)*	*(0.142)*	*(0.114)*	*(0.117)*
3.0	0.684	0.715	0.605	0.545	0.526	0.477	0.402	0.371	0.355
	(0.302)	*(0.255)*	*(0.219)*	*(0.193)*	*(0.156)*	*(0.151)*	*(0.119)*	*(0.108)*	*(0.094)*
3.5	0.706	0.718	0.611	0.538	0.500	0.434	0.373	0.345	0.322
	(0.311)	*(0.257)*	*(0.225)*	*(0.179)*	*(0.170)*	*(0.152)*	*(0.119)*	*(0.103)*	*(0.083)*
4.0	0.725	0.719	0.610	0.535	0.484	0.409	0.347	0.318	0.291
	(0.314)	*(0.258)*	*(0.225)*	*(0.187)*	*(0.155)*	*(0.151)*	*(0.108)*	*(0.094)*	*(0.084)*
4.5	0.756	0.736	0.630	0.536	0.466	0.379	0.330	0.293	0.273
	(0.307)	*(0.259)*	*(0.237)*	*(0.192)*	*(0.159)*	*(0.137)*	*(0.111)*	*(0.086)*	*(0.091)*
5.0	0.781	0.731	0.627	0.528	0.462	0.370	0.312	0.280	0.253
	(0.293)	*(0.261)*	*(0.238)*	*(0.190)*	*(0.179)*	*(0.136)*	*(0.111)*	*(0.095)*	*(0.079)*

Table 3: Average detection rate after 5 iterations of the single Rao-Blackwellised PMC algorithm as a function of σ_2 and μ_2. The number of samples per entry is 1470, obtained for 7 values of n and 6 values of p.

(σ_2, μ_2)	**1.0**	**1.5**	**2.0**	**2.5**	**3.0**	**3.5**	**4.0**	**4.5**	**5.0**
1.0	0.794	0.801	0.725	0.618	0.528	0.474	0.463	0.438	0.440
	(0.258)	*(0.247)*	*(0.266)*	*(0.285)*	*(0.265)*	*(0.241)*	*(0.238)*	*(0.232)*	*(0.229)*
1.5	0.747	0.811	0.753	0.700	0.627	0.558	0.487	0.439	0.386
	(0.275)	*(0.240)*	*(0.252)*	*(0.257)*	*(0.256)*	*(0.248)*	*(0.223)*	*(0.216)*	*(0.174)*
2.0	0.679	0.832	0.763	0.708	0.669	0.613	0.566	0.500	0.443
	(0.279)	*(0.241)*	*(0.255)*	*(0.253)*	*(0.248)*	*(0.232)*	*(0.212)*	*(0.206)*	*(0.191)*
2.5	0.679	0.832	0.756	0.697	0.648	0.586	0.514	0.447	0.416
	(0.291)	*(0.240)*	*(0.257)*	*(0.251)*	*(0.238)*	*(0.220)*	*(0.212)*	*(0.185)*	*(0.160)*
3.0	0.697	0.826	0.758	0.672	0.625	0.555	0.451	0.411	0.386
	(0.304)	*(0.242)*	*(0.256)*	*(0.263)*	*(0.237)*	*(0.226)*	*(0.179)*	*(0.159)*	*(0.137)*
3.5	0.712	0.807	0.745	0.657	0.591	0.488	0.424	0.376	0.355
	(0.312)	*(0.251)*	*(0.260)*	*(0.257)*	*(0.244)*	*(0.212)*	*(0.178)*	*(0.143)*	*(0.131)*
4.0	0.728	0.788	0.740	0.646	0.571	0.457	0.386	0.350	0.317
	(0.312)	*(0.258)*	*(0.260)*	*(0.257)*	*(0.238)*	*(0.204)*	*(0.153)*	*(0.138)*	*(0.123)*
4.5	0.758	0.795	0.738	0.627	0.540	0.431	0.363	0.318	0.298
	(0.307)	*(0.255)*	*(0.262)*	*(0.249)*	*(0.229)*	*(0.199)*	*(0.146)*	*(0.125)*	*(0.121)*
5.0	0.782	0.783	0.709	0.617	0.532	0.413	0.341	0.305	0.273
	(0.292)	*(0.259)*	*(0.264)*	*(0.253)*	*(0.241)*	*(0.182)*	*(0.141)*	*(0.124)*	*(0.105)*

Table 4: Same legend as Table 3 for the double Rao-Blackwellised PMC algorithm.

(σ_2, μ_2)	**1.0**	**1.5**	**2.0**	**2.5**	**3.0**	**3.5**	**4.0**	**4.5**	**5.0**
1.0	0.707	0.641	0.570	0.482	0.413	0.376	0.365	0.356	0.355
	(0.259)	*(0.244)*	*(0.226)*	*(0.220)*	*(0.199)*	*(0.187)*	*(0.187)*	*(0.181)*	*(0.185)*
1.5	0.656	0.599	0.537	0.516	0.478	0.443	0.403	0.371	0.340
	(0.262)	*(0.222)*	*(0.170)*	*(0.146)*	*(0.129)*	*(0.129)*	*(0.125)*	*(0.119)*	*(0.102)*
2.0	0.638	0.622	0.533	0.518	0.512	0.491	0.474	0.427	0.384
	(0.269)	*(0.231)*	*(0.157)*	*(0.119)*	*(0.115)*	*(0.094)*	*(0.108)*	*(0.115)*	*(0.107)*
2.5	0.656	0.641	0.531	0.510	0.507	0.483	0.436	0.386	0.366
	(0.285)	*(0.237)*	*(0.158)*	*(0.111)*	*(0.100)*	*(0.102)*	*(0.117)*	*(0.095)*	*(0.092)*
3.0	0.673	0.660	0.542	0.497	0.499	0.456	0.388	0.359	0.346
	(0.302)	*(0.243)*	*(0.168)*	*(0.130)*	*(0.112)*	*(0.114)*	*(0.098)*	*(0.084)*	*(0.080)*
3.5	0.700	0.660	0.554	0.499	0.472	0.416	0.362	0.335	0.315
	(0.311)	*(0.245)*	*(0.182)*	*(0.127)*	*(0.122)*	*(0.122)*	*(0.096)*	*(0.085)*	*(0.071)*
4.0	0.723	0.670	0.560	0.498	0.462	0.390	0.337	0.308	0.284
	(0.312)	*(0.248)*	*(0.186)*	*(0.139)*	*(0.126)*	*(0.121)*	*(0.095)*	*(0.075)*	*(0.073)*
4.5	0.755	0.700	0.572	0.496	0.445	0.365	0.319	0.285	0.267
	(0.308)	*(0.255)*	*(0.201)*	*(0.142)*	*(0.129)*	*(0.126)*	*(0.090)*	*(0.076)*	*(0.079)*
5.0	0.780	0.700	0.578	0.501	0.438	0.357	0.303	0.270	0.249
	(0.293)	*(0.258)*	*(0.208)*	*(0.159)*	*(0.149)*	*(0.120)*	*(0.101)*	*(0.078)*	*(0.068)*

Table 5: Average detection rate after 10 iterations of the single Rao-Blackwellised PMC algorithm as a function of σ_2 and μ_2.

(σ_2, μ_2)	**1.0**	**1.5**	**2.0**	**2.5**	**3.0**	**3.5**	**4.0**	**4.5**	**5.0**
1.0	0.786	0.785	0.712	0.610	0.521	0.467	0.453	0.431	0.432
	(0.259)	*(0.251)*	*(0.268)*	*(0.284)*	*(0.262)*	*(0.240)*	*(0.236)*	*(0.228)*	*(0.227)*
1.5	0.745	0.795	0.733	0.679	0.612	0.543	0.479	0.431	0.379
	(0.275)	*(0.246)*	*(0.253)*	*(0.255)*	*(0.249)*	*(0.240)*	*(0.216)*	*(0.208)*	*(0.167)*
2.0	0.677	0.813	0.736	0.684	0.647	0.600	0.555	0.494	0.437
	(0.280)	*(0.246)*	*(0.256)*	*(0.247)*	*(0.242)*	*(0.223)*	*(0.207)*	*(0.200)*	*(0.188)*
2.5	0.675	0.809	0.726	0.665	0.623	0.567	0.500	0.435	0.409
	(0.291)	*(0.247)*	*(0.256)*	*(0.244)*	*(0.229)*	*(0.208)*	*(0.199)*	*(0.169)*	*(0.152)*
3.0	0.694	0.804	0.723	0.643	0.606	0.537	0.443	0.404	0.380
	(0.304)	*(0.251)*	*(0.257)*	*(0.257)*	*(0.227)*	*(0.211)*	*(0.167)*	*(0.154)*	*(0.134)*
3.5	0.711	0.783	0.706	0.624	0.568	0.477	0.413	0.372	0.350
	(0.311)	*(0.257)*	*(0.256)*	*(0.244)*	*(0.233)*	*(0.202)*	*(0.169)*	*(0.135)*	*(0.124)*
4.0	0.727	0.764	0.704	0.609	0.542	0.446	0.377	0.346	0.312
	(0.313)	*(0.259)*	*(0.255)*	*(0.241)*	*(0.218)*	*(0.193)*	*(0.145)*	*(0.131)*	*(0.116)*
4.5	0.757	0.766	0.693	0.596	0.518	0.416	0.355	0.313	0.293
	(0.308)	*(0.259)*	*(0.258)*	*(0.237)*	*(0.215)*	*(0.185)*	*(0.139)*	*(0.117)*	*(0.116)*
5.0	0.782	0.764	0.665	0.589	0.512	0.404	0.334	0.300	0.268
	(0.293)	*(0.260)*	*(0.255)*	*(0.238)*	*(0.228)*	*(0.174)*	*(0.134)*	*(0.120)*	*(0.098)*

Table 6: Same legend as Table 5 for the double Rao-Blackwellised PMC algorithm.

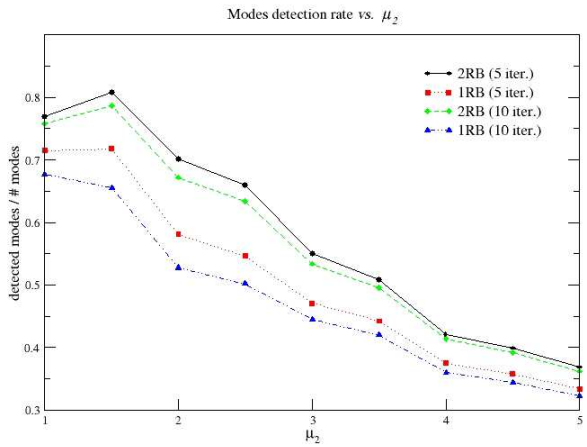


Figure 3: Evolution of the capture rate for the single and the double Rao–Blackwellised versions as a function of μ_2 after 5 and 10 iterations, obtained by averaging Tables 3–6 over σ_2.

Thus, for small values of n, both types of Rao-Blackwellised PMC algorithms achieve high coverage rates, but larger values of n lead to poorer achievements. Once again, the robustness against the number of PMC iterations is superior in the case of the double Rao-Blackwellisation. Figure 4 presents the evolution of the capture rate for the single and the double Rao–Blackwellisations as a function of the sample size n after 5 and 10 iterations, obtained by averaging Tables 7–10 over p.

$(\mathbf{p}, \mathbf{n})$	**20**	**30**	**40**	**50**	**100**	**500**	**1000**
0.10	0.602	0.565	0.542	0.519	0.478	0.423	0.411
	(0.288)	*(0.276)*	*(0.274)*	*(0.260)*	*(0.243)*	*(0.198)*	*(0.184)*
0.20	0.580	0.551	0.527	0.519	0.480	0.431	0.414
	(0.278)	*(0.263)*	*(0.249)*	*(0.246)*	*(0.222)*	*(0.193)*	*(0.173)*
0.30	0.583	0.552	0.527	0.529	0.491	0.445	0.424
	(0.276)	*(0.262)*	*(0.243)*	*(0.246)*	*(0.219)*	*(0.193)*	*(0.167)*
0.40	0.582	0.558	0.538	0.530	0.496	0.455	0.432
	(0.272)	*(0.259)*	*(0.253)*	*(0.241)*	*(0.221)*	*(0.200)*	*(0.174)*
0.50	0.602	0.572	0.564	0.553	0.533	0.476	0.455
	(0.283)	*(0.263)*	*(0.256)*	*(0.247)*	*(0.230)*	*(0.192)*	*(0.172)*
0.60	0.597	0.572	0.544	0.533	0.499	0.442	0.427
	(0.276)	*(0.266)*	*(0.256)*	*(0.242)*	*(0.220)*	*(0.174)*	*(0.153)*

Table 7: Average detection rate after 5 PMC iterations of the single Rao-Blackwellised algorithm as a function of n and p. The number of samples is 2835, obtained for 7 values of μ_1 and 5 values of σ_2.

$(\mathbf{p}, \mathbf{n})$	**20**	**30**	**40**	**50**	**100**	**500**	**1000**
0.10	0.652	0.619	0.601	0.577	0.542	0.463	0.445
	(0.291)	*(0.286)*	*(0.292)*	*(0.283)*	*(0.283)*	*(0.249)*	*(0.232)*
0.20	0.644	0.629	0.613	0.604	0.566	0.482	0.453
	(0.286)	*(0.284)*	*(0.283)*	*(0.283)*	*(0.279)*	*(0.250)*	*(0.229)*
0.30	0.653	0.645	0.630	0.628	0.588	0.500	0.467
	(0.288)	*(0.288)*	*(0.285)*	*(0.284)*	*(0.280)*	*(0.256)*	*(0.228)*
0.40	0.664	0.662	0.654	0.643	0.598	0.513	0.472
	(0.291)	*(0.289)*	*(0.290)*	*(0.281)*	*(0.283)*	*(0.260)*	*(0.227)*
0.50	0.668	0.651	0.648	0.639	0.614	0.524	0.482
	(0.298)	*(0.290)*	*(0.290)*	*(0.284)*	*(0.275)*	*(0.241)*	*(0.211)*
0.60	0.665	0.651	0.627	0.623	0.582	0.482	0.450
	(0.291)	*(0.288)*	*(0.285)*	*(0.276)*	*(0.272)*	*(0.222)*	*(0.189)*

Table 8: Same legend as Table 7 in the double Rao-Blackwellised case.

$(\mathbf{p}, \mathbf{n})$	**20**	**30**	**40**	**50**	**100**	**500**	**1000**
0.10	0.577	0.541	0.519	0.495	0.457	0.408	0.398
	(0.281)	*(0.266)*	*(0.259)*	*(0.243)*	*(0.223)*	*(0.175)*	*(0.161)*
0.20	0.545	0.514	0.492	0.485	0.447	0.413	0.402
	(0.265)	*(0.242)*	*(0.222)*	*(0.218)*	*(0.183)*	*(0.164)*	*(0.152)*
0.30	0.541	0.510	0.489	0.480	0.456	0.424	0.414
	(0.257)	*(0.233)*	*(0.214)*	*(0.201)*	*(0.180)*	*(0.159)*	*(0.146)*
0.40	0.536	0.516	0.496	0.487	0.458	0.431	0.418
	(0.252)	*(0.234)*	*(0.219)*	*(0.207)*	*(0.178)*	*(0.162)*	*(0.143)*
0.50	0.564	0.542	0.528	0.517	0.497	0.459	0.443
	(0.269)	*(0.246)*	*(0.232)*	*(0.219)*	*(0.197)*	*(0.166)*	*(0.156)*
0.60	0.544	0.519	0.499	0.487	0.463	0.426	0.420
	(0.258)	*(0.238)*	*(0.224)*	*(0.206)*	*(0.180)*	*(0.146)*	*(0.134)*

Table 9: Average detection rate after 10 PMC iterations for the single Rao–Blackwellised algorithm as a function of n and p.

$(\mathbf{p}, \mathbf{n})$	**20**	**30**	**40**	**50**	**100**	**500**	**1000**
0.10	0.639	0.608	0.590	0.564	0.529	0.458	0.441
	(0.291)	*(0.285)*	*(0.288)*	*(0.277)*	*(0.275)*	*(0.243)*	*(0.227)*
0.20	0.622	0.612	0.594	0.585	0.549	0.475	0.448
	(0.284)	*(0.281)*	*(0.278)*	*(0.276)*	*(0.267)*	*(0.246)*	*(0.220)*
0.30	0.629	0.623	0.608	0.604	0.571	0.493	0.461
	(0.284)	*(0.283)*	*(0.279)*	*(0.277)*	*(0.274)*	*(0.250)*	*(0.223)*
0.40	0.641	0.636	0.628	0.620	0.582	0.506	0.468
	(0.287)	*(0.285)*	*(0.286)*	*(0.276)*	*(0.274)*	*(0.254)*	*(0.224)*
0.50	0.642	0.630	0.627	0.619	0.600	0.518	0.480
	(0.293)	*(0.284)*	*(0.283)*	*(0.277)*	*(0.269)*	*(0.237)*	*(0.208)*
0.60	0.641	0.628	0.608	0.604	0.569	0.475	0.447
	(0.287)	*(0.281)*	*(0.278)*	*(0.271)*	*(0.264)*	*(0.215)*	*(0.184)*

Table 10: Same legend as Table 9 for the double Rao–Blackwellised PMC algorithm.

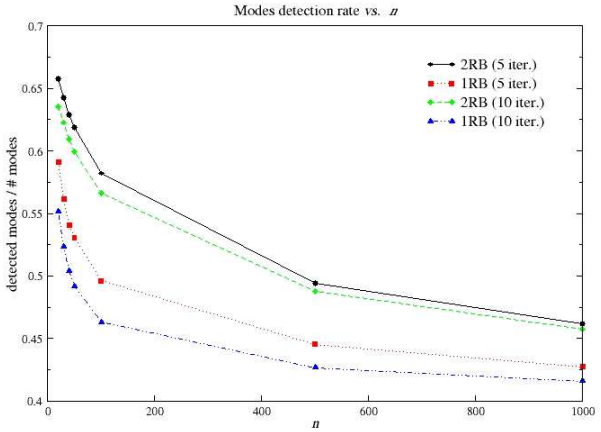


Figure 4: Evolution of the capture rate for the single and the double Rao–Blackwellisations as a function of the sample size n after 5 and 10 iterations, obtained by averaging Tables 7–10 over p.

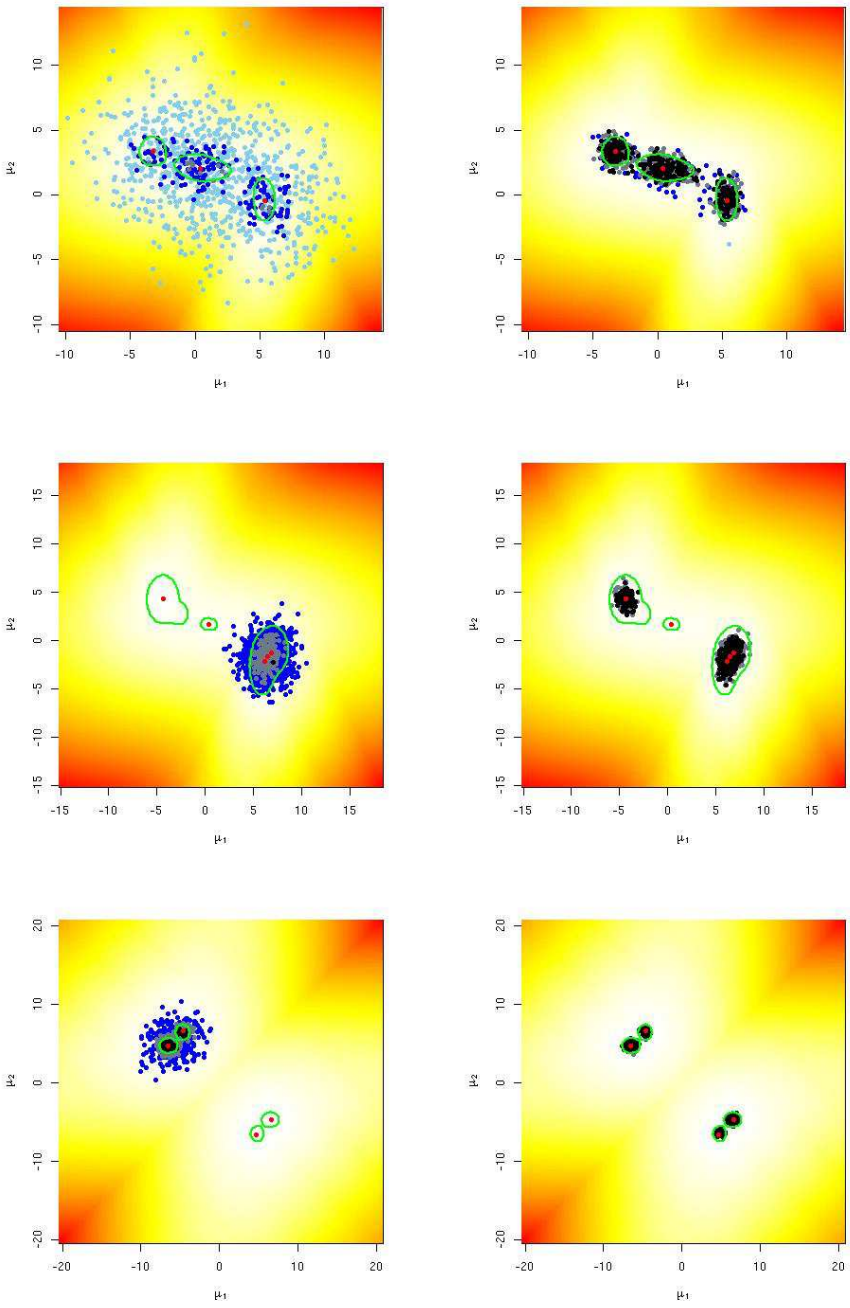


Figure 5: Comparison between PMC samples using a single Rao-Blackwellisation (left) and a double Rao-Blackwellisation (right).

Figure 5 illustrates the superior performances of the double Rao-Blackwellisation strategy, compared to the single one in terms of modes detection. Those three graphs plot side by side for a given sample the repartition of the PMC samples in the modal domains. The colour codes are associated with the five possible variances used in the 5 kernel proposal. In those three experiments, the persistence of the samples in all modes, as well as the concentration in the regions of importance, is noticeable. Both rows of Figure 5 interestingly contain a ridge structure in one of the modal basins.

5 Conclusions

The double Rao-Blackwellised algorithm provides a more robust framework for adapting general importance sampling densities represented as mixtures in the sense that the influence of the starting points diminishes when compared with the original algorithm. It is thus unnecessary to rely on a large value of the number T of iterations of the PMC algorithm: with large enough sample sizes N at each iteration—especially on the initial iteration that requires many points to counter-weight a potentially poor initial proposal—, it is quite uncommon to fail to spot a stabilisation of both the estimates and of the entropy criterion within a few iterations.

References

Cappé, O., Douc, R., Guillin, A., Marin, J.-M., and Robert, C. (2007). Adaptive importance sampling in general mixture classes. Rapport de recherche INRIA, 6332.

Cappé, O., Guillin, A., Marin, J.-M., and Robert, C. (2004). Population Monte Carlo. *J. Comput. Graph. Statist.*, 13(4):907–929.

Douc, R., Guillin, A., Marin, J.-M., and Robert, C. (2007a). Convergence of adaptive mixtures of importance sampling schemes. *Ann. Statist.*, 35(1).

Douc, R., Guillin, A., Marin, J.-M., and Robert, C. (2007b). Minimum variance importance sampling via population Monte Carlo. *ESAIM: Probability and Statistics*, 11:427–447.

Marin, J.-M., Mengersen, K., and Robert, C. (2005). Bayesian modelling and inference on mixtures of distributions. In Rao, C. and Dey, D., editors, *Handbook of Statistics*, volume 25. Springer-Verlag, New York.

Marin, J.-M. and Robert, C. (2007). *Bayesian Core*. Springer-Verlag, New York.

R Development Core Team (2006). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Robert, C. and Casella, G. (2004). *Monte Carlo Statistical Methods*. Springer-Verlag, New York, Second Edition.



Unité de recherche INRIA Futurs
Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)

Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399