Available online at www.sciencedirect.com
ScienceDirect
Mathematics and Computers in Simulation 79 (2009) 3424–3433
MATHEMATICS AND COMPUTERS IN SIMULATION
www.elsevier.com/locate/matcom

Parallel and interacting Markov chain Monte Carlo algorithm

Fabien Campillo [a], Rivo Rakotozafy [b], Vivien Rossi [c,*]

[a] *INRIA/INRA, MERE Project-Team, Montpellier, France*
[b] *University of Fianarantsoa, Fianarantsoa, Madagascar*
[c] *CIRAD, Research Unit, Dynamics of Natural Forests, Montpellier, France*

Received 1 February 2008; received in revised form 18 November 2008; accepted 9 April 2009
Available online 22 April 2009

Abstract

In many situations it is important to be able to propose N independent realizations of a given distribution law. We propose a strategy for making N parallel Monte Carlo Markov chains (MCMC) interact in order to get an approximation of an independent N-sample of a given target law. In this method each individual chain proposes candidates for all other chains. We prove that the set of interacting chains is itself a MCMC method for the product of N target measures. Compared to independent parallel chains this method is more time consuming, but we show through examples that it possesses many advantages. This approach is applied to a biomass evolution model.

Keywords: Markov chain Monte Carlo method; Interacting chains; Hidden Markov model

1. Introduction

Markov chain Monte Carlo (MCMC) algorithms [23,14,22] allows us to draw samples from a probability distribution $\pi(x)\,\mathrm{d}x$ known up to a multiplicative constant. They consist of sequentially simulating a single Markov chain whose limit distribution is $\pi(x)\,\mathrm{d}x$. There exist many techniques to speed up the convergence toward the target distribution by improving the mixing properties of the chain [15,16]. Moreover, special attention should be given to the convergence diagnosis of this method [1,8,19].

An alternative is to run many Markov chains in parallel. The simplest multiple chain algorithm is to make use of parallel independent chains [11]. The recommendations concerning this idea seem contradictory in the literature (cf. the "many short runs" *vs.*"one long run" debate described in [12]). We can note with [13] and ([22], Section 6.5) that independent parallel chains could be a poor idea: among these chains some may not converge, so one long chain could be preferable to many short ones. Moreover, many parallel independent chains can artificially exhibit a more robust behavior which does not correspond to a real convergence of the algorithm.

In practice one however makes use of several chains in parallel. It is then tempting to exchange information between these chains to improve mixing properties of the MCMC samplers [5–7,20,9,10]. A general framework of "Population Monte Carlo" has been proposed in this context [17,21,4]. A recent review of population-based simulation for static inference problems is presented in [18].

* Corresponding author.
E-mail addresses: Fabien.Campillo@inria.fr (F. Campillo), rrakotoz@univ-fianar.mg (R. Rakotozafy), vivien.rossi@cirad.fr (V. Rossi).

0378-4754/$36.00 © 2009 IMACS. Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.matcom.2009.04.010

In this paper we propose an interacting method between parallel chains which provides an independent sample from the target distribution. Contrary to papers previously cited, the proposal law in our work is given and does not adapt itself to the previous simulations. Hence, the problem of the choice of this law still remains.

The corresponding Metropolis within Gibbs (MwG) algorithm and its theoretical properties are presented in the following section. In Section 3, two simple numerical examples illustrate how the introduction of interactions can speed up the convergence.

2. Parallel/interacting Metropolis within Gibbs (MwG) algorithm

Let $\pi(x)$ be the probability density function of a target distribution defined on $(\mathbb{R}^n, \mathcal{B}(\mathbb{R}^n))$. We propose a method for sampling N independent values $X^1, \ldots, X^N \in \mathbb{R}^n$ of the law $\pi(x)\,\mathrm{d}x$. For $\ell = 1, \ldots, n$, we define the conditional laws:

$$\pi_\ell(x_\ell | x_{\neg\ell}) \stackrel{\text{def}}{=} \frac{\pi(x_{1:n})}{\int \pi(x_{1:n})}\,\mathrm{d}x_{\neg\ell}. \tag{1}$$

where $\neg\ell \stackrel{\text{def}}{=} \{m = 1 : n; m \neq \ell\}$. When we know to sample from (1), we are able to use the Gibbs sampler. It is possible to adapt our interacting method to parallel Gibbs sampler. But very often we do not know how to sample from (1) and therefore we consider proposal conditional densities $\pi_\ell^{\text{prop}}(x_\ell)$ defined for all ℓ. In this case, we use MwG algorithm, see [22]. We present in the following how to make interactions between parallel MwG algorithms. The MwG algorithm is more general than Gibbs algorithm, so a parallel/interacted Gibbs algorithm can easily be deduced from the parallel/interacted MwG algorithm.

2.1. Notations

Let

$$X = X^{1:N} = X_{1:n} \in \mathbb{R}^{n \times N},$$

so that $X_\ell \in \mathbb{R}^N$ and $X^i \in \mathbb{R}^n$ (the same for Y and Z); $x \in \mathbb{R}^n$ so that $x_\ell \in \mathbb{R}$ (the same for y and z); $\xi, \xi' \in \mathbb{R}$. Here $X^{1:N} = (X^1, \ldots, X^N)$ and $X_{1:n} = (X_1, \ldots, X_n)$. We also define $\neg\ell = \{1, \ldots, n\} \setminus \{\ell\}$. Note that the structure of the matrix X is:

$$X = \begin{array}{c} \\ X^i \\ \uparrow \\ \end{array} \begin{bmatrix} X_1^1 & \cdots & X_1^i & \cdots & X_1^N \\ \vdots & & \vdots & & \vdots \\ X_\ell^1 & \cdots & X_\ell^i & \cdots & X_\ell^N \\ \vdots & & \vdots & & \vdots \\ X_n^1 & \cdots & X_n^i & \cdots & X_n^N \end{bmatrix} \begin{array}{l} \\ \\ \\ \rightarrow\ X_\ell \\ \\ \end{array}.$$

2.2. The parallel/interacted MwG algorithm

One iteration $X \to Z$ of the parallel/interacting MwG method consists of updating the components X_ℓ successively for $\ell = 1, \ldots, n$, i.e.

$$[X_{1:n}] \to [Z_1 X_{2:n}] \to [Z_{1:2} X_{3:n}] \cdots [Z_{1:n-1} X_n] \to [Z_{1:n}].$$

For each ℓ fixed, the subcomponents X_ℓ^i are updated sequentially for $i = 1 : N$ in two steps:

(1) *Proposal step:* We sample independently N candidates $Y_\ell^j \in \mathbb{R}$ according to:

$$Y_\ell^j \sim \pi_{i,j}^{\ell,\mathrm{prop}}(\xi | [\![Z, X_\ell^i, X_\ell^i)]\!]\, \mathrm{d}\xi, \qquad 1 \le j \le n$$

where

$$[\![Z, \xi, X]\!]_\ell^i \stackrel{\mathrm{def}}{=} \left[\begin{array}{c|c|c} & Z_\ell^1 & \\ & \vdots & \\ & Z_\ell^{i-1} & \\ Z_{1:\ell-1} & \xi & X_{\ell+1:n} \\ & X_\ell^{i+1} & \\ & \vdots & \\ & X_\ell^N & \end{array}\right].$$

We also use the following lighter notation: $\pi_{i,j}^{\ell,\mathrm{prop}}(\xi|\xi') = \pi_{i,j}^{\ell,\mathrm{prop}}(\xi|[\![Z, \xi', X]\!]_\ell^i)$.

(2) *Selection step:* The subcomponent X_ℓ^i could be replaced by one of the N candidates $Y_\ell^{1:N}$ or stay unchanged according to a multinomial sampling, the resulting value is called Z_ℓ^i, i.e.:

$$Z_\ell^i \leftarrow \begin{cases} Y_\ell^1 \text{ with probability } \dfrac{1}{N}\,\alpha_\ell^{i,1}(X_\ell^i, Y_\ell^1), \\ \vdots \\ Y_\ell^N \text{ with probability } \dfrac{1}{N}\,\alpha_\ell^{i,N}(X_\ell^i, Y_\ell^N), \\ X_\ell^i \text{ with probability} \tilde{\rho}_\ell^i(X_\ell^i, Y_\ell^{1:N}) \end{cases}$$

where

$$\alpha_\ell^{i,j}(\xi, \xi') \stackrel{\mathrm{def}}{=} \frac{\pi_\ell(\xi'|X_{\neg\ell}^i)}{\pi_\ell(\xi|X_{\neg\ell}^i)} \frac{\pi_{i,j}^{\ell,\mathrm{prop}}(\xi|\xi')}{\pi_{i,j}^{\ell,\mathrm{prop}}(\xi'|\xi)} \wedge 1, \quad \tilde{\rho}_\ell^i(X_\ell^i, Y_\ell^{1:N}) \stackrel{\mathrm{def}}{=} 1 - \frac{1}{N}\sum_{j=1}^N \alpha_\ell^{i,j}(X_\ell^i, Y_\ell^j).$$

2.3. Description of the MH kernel

Lemma 1. *The Markov kernel on $\mathbb{R}^{n\times N}$ associated with the MwG algorithm is*

$$P(X, \mathrm{d}Z) \stackrel{\mathrm{def}}{=} P_1(X_{1:n}; \mathrm{d}Z_1)\ P_2(Z_1, X_{2:n}; \mathrm{d}Z_2) \cdots P_n(Z_{1:n-1}, X_n; \mathrm{d}Z_n). \tag{2}$$

At iteration ℓ, the kernel $P_\ell(Z_{1:\ell-1}, X_{\ell:n}; \mathrm{d}Z_\ell)$ generates $Z_\ell^{1:N}$ from the already updated components $Z_{1:\ell-1}^{1:N}$ and the remaining components $X_{\ell:n}^{1:N}$.

Each component Z_ℓ^i, for $i = 1, \ldots, N$, is updated independently one from each other:

$$P_\ell(Z_{1:\ell-1}, X_{\ell:n}; \mathrm{d}Z_\ell) \stackrel{\text{def}}{=} \prod_{i=1}^{N} P_\ell^i([\![Z, X_\ell^i, X]\!]_\ell^i; \mathrm{d}Z_\ell^i). \tag{3}$$

Here Z_ℓ^i is generated from $[\![Z, X_\ell^i, X]\!]_\ell^i$ according to:

$$P_\ell^i([\![Z, \xi, X]\!]_\ell^i; \mathrm{d}\xi') \stackrel{\text{def}}{=} \frac{1}{N}\sum_{j=1}^{N} \alpha_\ell^{i,j}(\xi, \xi')\pi_{i,j}^{\ell,\text{prop}}(\xi'|\xi)\,\mathrm{d}\xi' + \rho_\ell^i(\xi)\delta_\xi(\mathrm{d}\xi') \tag{4}$$

Acceptance probabilities are:

$$\alpha_\ell^{i,j}(\xi, \xi') \stackrel{\text{def}}{=} \begin{cases} r_\ell^{i,j}(\xi, \xi') \wedge 1 & if\ (\xi, \xi') \in R_\ell^{i,j}, \\ 0 & \text{otherwise}, \end{cases} \tag{5}$$

$$r_\ell^{i,j}(\xi, \xi') \stackrel{\text{def}}{=} \frac{\pi_\ell(\xi'|Z_{1:\ell-1}^i, X_{\ell+1:n}^i)}{\pi_\ell(\xi|Z_{1:\ell-1}^i, X_{\ell+1:n}^i)} \frac{\pi_{i,j}^{\ell,\text{prop}}(\xi|\xi')}{\pi_{i,j}^{\ell,\text{prop}}(\xi'|\xi)}, \tag{6}$$

$$\rho_\ell^i(\xi) \stackrel{\text{def}}{=} 1 - \frac{1}{N}\sum_{j=1}^{N} \int_{\mathbb{R}} \alpha_\ell^{i,j}(\xi, \xi')\pi_{i,j}^{\ell,\text{prop}}(\xi'|\xi)\,\mathrm{d}\xi'. \tag{7}$$

Finally, $R_\ell^{i,j}$ is the set of ordered pairs $(\xi, \xi') \in \mathbb{R}^2$ such that

$$\pi_\ell(\xi'|Z_{1:\ell-1}^i, X_{\ell+1:n}^i)\pi_{i,j}^{\ell,\text{prop}}(\xi|\xi') > 0,$$
$$\pi_\ell(\xi|Z_{1:\ell-1}^i, X_{\ell+1:n}^i)\pi_{i,j}^{\ell,\text{prop}}(\xi'|\xi) > 0.$$

Note that the functions $\alpha_\ell^{i,j}(\xi, \xi')$, $\rho_\ell^i(\xi)$, $r_\ell^{i,j}(\xi, \xi')$ and the set $R_\ell^{i,j}$ depend on $Z_{1:\ell-1}$ and $X_{\ell+1:n}$.

The proof is given in Appendix A.

2.4. Invariance property

First, assume that the following lemma holds.

Lemma 2 (conditional detailed balance). *The following equality of measures defined on $\mathbb{R} \times \mathbb{R}$*

$$P_\ell^i([\![Z, \xi, X]\!]_\ell^i; \mathrm{d}\xi') \times \pi_\ell(\xi|Z_{1:\ell-1}^i, X_{\ell+1:n}^i)\,\mathrm{d}\xi = P_\ell^i([\![Z, \xi', X]\!]_\ell^i; \mathrm{d}\xi) \times \pi_\ell(\xi'|Z_{1:\ell-1}^i, X_{\ell+1:n}^i)\,\mathrm{d}\xi' \tag{8}$$

holds true for any $\ell = 1, \ldots, n$, $i = 1, \ldots, N$ and $Z_{1:\ell-1} \in \mathbb{R}^{N\times(\ell-1)}$, $X_{\ell+1:n} \in \mathbb{R}^{N\times(n-\ell)}$.

The proof is given in Appendix B.

Proposition 3 (invariance). *The measure*

$$\Pi(\mathrm{d}X) = \pi(X^1)\,\mathrm{d}X^1, \ldots, \pi(X^N)\,\mathrm{d}X^N$$

is invariant for the kernel P, that is $\Pi P = \Pi$ i.e.:

$$\int_X P(X, \mathrm{d}Z) \left\{ \prod_{i=1}^{N} \pi(X^i)\,\mathrm{d}X^i \right\} = \prod_{i=1}^{N} \pi(Z^i)\,\mathrm{d}Z^i. \tag{9}$$

Proof.

$$\begin{aligned}\int_X P(X,\mathrm{d}Z)\left\{\prod_{i=1}^N \pi(X^i)\,\mathrm{d}X^i\right\} &= \int_X P_1(X_{1:n};\mathrm{d}Z_1)P_2(Z_1,X_{2:n};\mathrm{d}Z_2)\cdots P_n(Z_{1:n-1},X_n;\mathrm{d}Z_n)\prod_{i=1}^N\{\pi_1(X_1^i|X_{2:n}^i)\,\mathrm{d}X_1^i\pi_{\neg 1}(X_{2:n}^i)\,\mathrm{d}X_{2:n}^i\}\\ &= \int_X P_1(X_{1:n};\mathrm{d}Z_1)\left\{\prod_{i=1}^N \pi_1(X_1^i|X_{2:n}^i)\,\mathrm{d}X_1^i\right\}P_2(Z_1,X_{2:n};\mathrm{d}Z_2)\cdots P_n(Z_{1:n-1},X_n;\mathrm{d}Z_n)\left\{\prod_{i=1}^N \pi_{\neg 1}(X_{2:n}^i)\,\mathrm{d}X_{2:n}^i\right\}\\ &= \int_X \left\{\prod_{i=1}^N P_1^i([\![Z,X_1^i,X]\!]_1^i;\mathrm{d}Z_1^i)\right\}\left\{\prod_{i=1}^N \pi_1(X_1^i|X_{2:n}^i)\,\mathrm{d}X_1^i\right\}P_2(Z_1,X_{2:n};\mathrm{d}Z_2)\cdots P_n(Z_{1:n-1},X_n;\mathrm{d}Z_n)\left\{\prod_{i=1}^N \pi_{\neg 1}(X_{2:n}^i)\,\mathrm{d}X_{2:n}^i\right\}\end{aligned}$$

Moreover

$$\begin{aligned}P_1(X_{1:n};\mathrm{d}Z_1)\left\{\prod_{i=1}^N \pi_1(X_1^i|X_{2:n}^i)\,\mathrm{d}X_1^i\right\} &= \left\{\prod_{i=1}^N P_1^i([\![Z,X_1^i,X]\!]_1^i;\mathrm{d}Z_1^i)\right\}\left\{\prod_{i=1}^N \pi_1(X_1^i|X_{2:n}^i)\,\mathrm{d}X_1^i\right\}\\ &= \prod_{i=1}^N P_1^i([\![Z,X_1^i,X]\!]_1^i;\mathrm{d}Z_1^i)\pi_1(X_1^i|X_{2:n}^i)\,\mathrm{d}X_1^i\\ &= \prod_{i=1}^N P_1^i([\![Z,Z_1^i,X]\!]_1^i;\mathrm{d}X_1^i)\pi_1(Z_1^i|X_{2:n}^i)\,\mathrm{d}Z_1^i\end{aligned}$$

this last equality follows from Eq. (8) in the Lemma 2. Hence,

$$\begin{aligned}\int_X P(X,\mathrm{d}Z)\left\{\prod_{i=1}^N \pi(X^i)\,\mathrm{d}X^i\right\} = \int_X \prod_{i=1}^N \left\{P_1^i([\![Z,Z_1^i,X]\!]_1^i;\mathrm{d}X_1^i)\pi_1(Z_1^i|X_{2:n}^i)\,\mathrm{d}Z_1^i\right\}P_2(Z_1,X_{2:n};\mathrm{d}Z_2)\cdots P_n\\ \times(Z_{1:n-1},X_n;\mathrm{d}Z_n)\left\{\prod_{i=1}^N \pi_{\neg 1}(X_{2:n}^i)\,\mathrm{d}X_{2:n}^i\right\}\end{aligned}$$

In this last expression, for $i=1,\ldots,N$, the kernel $P_1^i([\![Z,Z_1^i,X]\!]_1^i;\mathrm{d}X_1^i)$ is a measure for the variable X_1^i which no longer appears in the integrand. Using the fact that the integral of the kernel w.r.t. X_1^i is 1 we get:

$$\begin{aligned}\int_X P(X,\mathrm{d}Z)\left\{\prod_{i=1}^N \pi(X^i)\,\mathrm{d}X^i\right\} &= \int_{X_{2:N}} \prod_{i=1}^N \left\{\pi_1(Z_1^i|X_{2:n}^i)\,\mathrm{d}Z_1^i\right\}P_2(Z_1,X_{2:n};\mathrm{d}Z_2)\cdots P_n(Z_{1:n-1},X_n;\mathrm{d}Z_n)\left\{\prod_{i=1}^N \pi_{\neg 1}(X_{2:n}^i)\,\mathrm{d}X_{2:n}^i\right\}\\ &= \int_{X_{2:N}} \prod_{i=1}^N P_2(Z_1,X_{2:n};\mathrm{d}Z_2)\cdots P_n(Z_{1:n-1},X_n;\mathrm{d}Z_n)\left\{\prod_{i=1}^N \pi(Z_1^i X_{2:n}^i)\,\mathrm{d}Z_1^i\,\mathrm{d}X_{2:n}^i\right\}\end{aligned}$$

Repeating this process successively for X_2 to X_n leads to (9). □

3. Numerical tests

We present two examples in the context of hidden Markov models with hidden state variable x_ℓ and unknown parameter θ. In this case the natural choice [3] for proposal distributions when using MwG samplers is (i) the transition kernel of the state variable x_ℓ and (ii) the prior distribution for the parameter θ.

3.1. A linear hidden Markov model

We apply the parallel/interacting MwG sampler to a toy problem where a good estimate $\hat{\pi}$ of the target distribution π is available. Consider

$$x_{\ell+1} = \mathbf{a}\,x_\ell + w_\ell, \qquad y_\ell = \mathbf{b}\,x_\ell + v_\ell$$

for $\ell = 1, \ldots, n$, where $x_1 \sim \mathcal{N}(\bar{x}_1, Q_1)$, $w_{1:n}$ and $v_{1:n}$ are centered white Gaussian noises with variances σ_w^2 and σ_v^2. Suppose that $\mathbf{b}$ is known and $\mathbf{a} = \theta$ is unknown with a priori law $\mathcal{N}(\mu_\theta, \sigma_\theta^2)$. We also suppose that $w_{1:n}$, $v_{1:n}$, x_1 and θ are mutually independent.

The state variable is $(x_{1:n}, \theta)$ and the target law is $\pi(x_{1:n}, \theta)\, \mathrm{d}x_{1:n}\, \mathrm{d}\theta \stackrel{\text{def}}{=} \text{law}(x_{1:n}, \theta | y_{1:n})$. This law is not Gaussian, but we can perform a Gibbs sampler:

$$\pi_{x_\ell}(x_\ell | x_{\neg\ell}, \theta)\, \mathrm{d}x_\ell \stackrel{\text{def}}{=} \text{law}(x_\ell | x_{\neg\ell}, \theta, y_{1:n}) = \mathcal{N}(\mathrm{m}_\ell, \mathrm{r}^2),$$
$$\pi_\theta(\theta | x_{1:n})\, \mathrm{d}\theta \stackrel{\text{def}}{=} \text{law}(\theta | x_{1:n}, y_{1:n}) = \mathcal{N}(\tilde{\mathrm{m}}, \tilde{\mathrm{r}}^2)$$

where r^2, m_ℓ, $\tilde{\mathrm{r}}^2$ and $\tilde{\mathrm{m}}$ are known, see [2]. We will perform three algorithms: (i) N parallel/interacting MwG samplers, (ii) N parallel/independent MwG samplers, (iii) N_{Gibbs} parallel/independent Gibbs samplers.

Our aim is to show that making parallel samplers interact could speed up the convergence toward the stationary distribution. Because of its good convergence property, method (iii) is considered as a reference method. Here we perform $k = 10,000$ iterations of $N_{\text{Gibbs}} = 5000$ independent Gibbs samplers. We obtain a kernel density estimate $\hat{\pi}$ of the target density based on the $N_{\text{Gibbs}} = 5000$ final values. Let $\hat{\pi}_{x_\ell}$ be the corresponding ℓ-th marginal density. For methods (i) and (ii) we perform $N = 50$ parallel samplers. Let $\pi^{\text{int},k}$ and $\pi^{\text{ind},k}$ be the kernel density estimates of the target density based on the final values of methods (i) and (ii), respectively. Let $\pi_{x_\ell}^{\text{int},k}$ and $\pi_{x_\ell}^{\text{ind},k}$ be the corresponding ℓ-th marginal densities.

The parameter values for the simulations are $\mathbf{a} = 2$, $\mathbf{b} = 2$, $\sigma_w^2 = 9$, $\sigma_v^2 = 25$, $x_1 \sim \mathcal{N}(4, 9)$, $\theta \sim \mathcal{N}(1, 4)$ and $n = 10$.

For this example, in case of parallel/interacting MwG samplers, the proposal distribution of the ith chain to update the ℓ th element of the jth chain is

$$\pi_{i,j}^{\ell,\text{prop}}(x_\ell^j | x_{\neg\ell}, \theta) = \mathcal{N}(\theta\, x_{\ell-1}^i, \sigma_w^2)$$

and the proposal distribution of the ith chain to update the θ component of the jth chain

$$\pi_{i,j}^{\ell,\text{prop}}(\theta^j | x_{1:n}) = \mathcal{N}(1, 4).$$

In the parallel/independent MwG samplers, these are the same distributions except that the chains are updated only with their own proposed candidate.

For each algorithm (i) and (ii), that is for $\pi_{x_\ell}^k = \pi_{x_\ell}^{\text{ind},k}$ and $\pi_{x_\ell}^{\text{int},k}$, we compute

$$\epsilon^k = \frac{1}{n+1} \sum_{\ell=1}^{n+1} \epsilon_\ell^k \quad \text{with} \quad \epsilon_\ell^k \stackrel{\text{def}}{=} \int |\pi_{x_\ell}^k(\xi) - \hat{\pi}_{x_\ell}(\xi)|\, \mathrm{d}\xi, \quad \ell = 1, \ldots, n+1. \tag{10}$$

Hence ϵ^k is a criteria of the error between the target probability distribution and its estimation provided by the algorithm used.

These estimations are based on a sample of size $N = 50$only, so they suffer from variability. This is not problematical, indeed we do not want to estimate L^1 errors but to diagnose the convergence toward the stationary distribution. So we use ϵ_ℓ^k as an indicator which must decrease and remain close to a small value when convergence occurs.

To compare fairly the parallel/independent MwG algorithm and the parallel/interacted MwG algorithm, we represent in Fig. 1 the indicator ϵ^k for each algorithm not as a function of k but as a function of the CPU time. In Fig. 1(left) we see that even if one iteration of algorithm (i) needs more CPU than one of (ii), still the first algorithm converges more rapidly than the second one. This shows the inefficiency of parallel/independent MwG on this simple model.

3.2. Ricker model

We consider the Ricker discrete-time stock-recruitment model perturbated by a noise:

$$x_{\ell+1} = x_\ell\, e^{r - b\, x_\ell} e^{w_\ell}, \quad \ell = 1, \ldots, n.$$

where r is the growth parameter and w_ℓ is a white Gaussian noise $\mathcal{N}(0, \sigma_w^2)$. We suppose that measurements satisfy:

$$y_\ell = h\, x_\ell + v_\ell$$

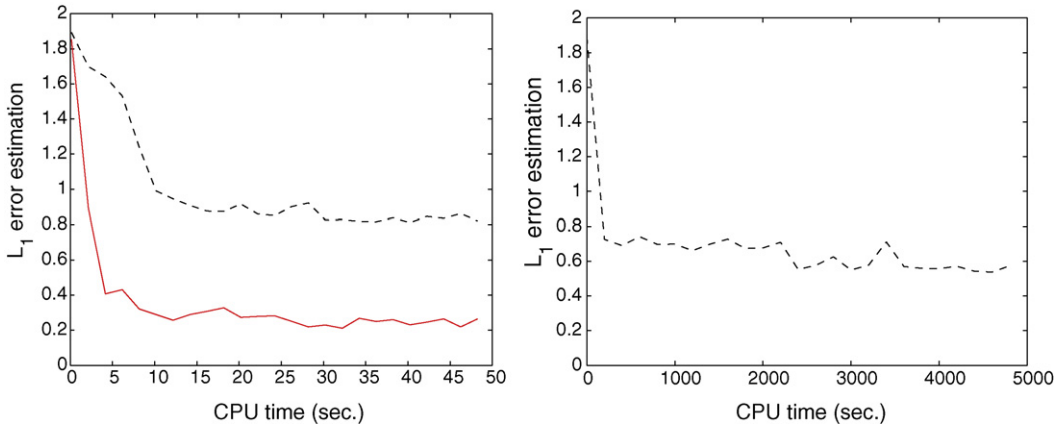

Fig. 1. Left: Evolution of the indicator ϵ^k, see (10), for the parallel/independent MwG sampler (- - -), and for the parallel/interacting MH sampler (—). This evolution is depicted as a function of the CPU time and not as a function of the iteration number k. The residual error of about 0.22 for the second method is due to the limited size of the sample. Right: Evolution of the indicator ϵ^k, see (10), for the parallel/independent MwG sampler (- - -). After 5000 s. CPU time, the convergence of this method is still unsatisfactory.

where v_ℓ is a white Gaussian noise $\mathcal{N}(0, \sigma_v^2)$. For notational convenience we assume that $h = 1$. Suppose that only r is unknown so that the target law is law$(x_{1:n}, \theta | y_{1:n})$.

We ran two parallel MwG samplers with and without interaction. The parameter values for the simulations are $b = 0.02$, $r = 1.5$, $\sigma_w^2 = 1$, $\sigma_v^2 = 0.5$, $x_1 \sim \mathcal{N}(3, 1)$, $\theta \sim \mathcal{N}(4, 2^2)$ and $n = 20$.

For this example, in case of parallel/interacting MwG samplers, the proposal distribution of the ith chain to update the ℓ th element of the jth chain is

$$\pi_{i,j}^{\ell,\text{prop}}(x_\ell^j | x_{\neg\ell}, \theta) = \text{LogNormal}(\log(x_{\ell-1}^i) + r - b\, x_{\ell-1}^i, \sigma_w^2)$$

and the proposal distribution of the ith chain to update the θ component of the jth chain

$$\pi_{i,j}^{\ell,\text{prop}}(\theta^j | x_{1:n}) = \mathcal{N}(4, 4).$$

In the parallel/independent MwG samplers, these are the same distributions except that the chains are updated only with their own proposed candidate.

To compare fairly the parallel/independent MwG algorithm and the parallel/interacted MwG algorithm, we represent in Fig. 2 the chains which estimate the parameter r for each algorithm, as a function of the CPU time. The Fig. 2 shows

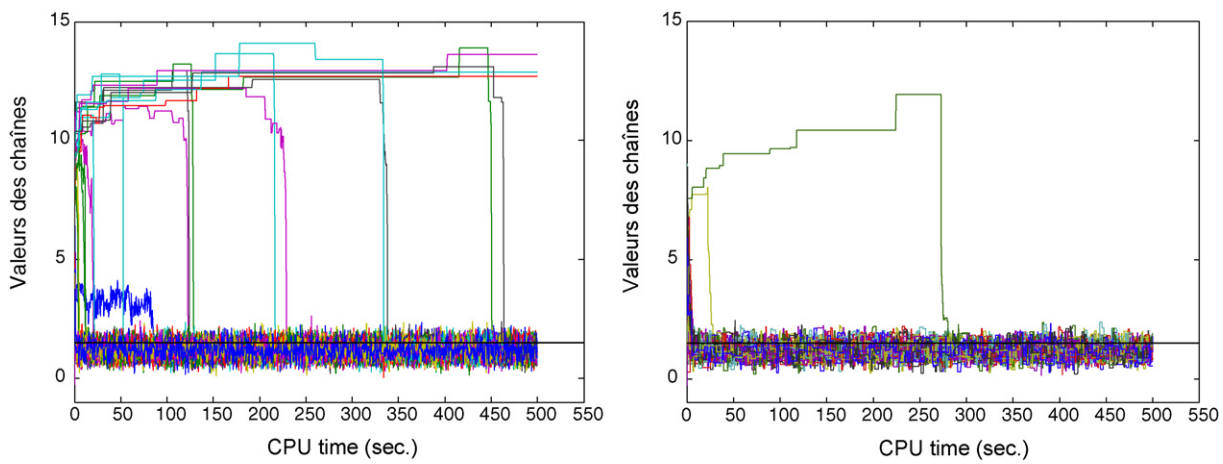

Fig. 2. Evolution of the estimation of the parameter r versus the MCMC iterations: $N = 50$ parallel samplers without interaction (left) and 50 parallel samplers with interaction (right). Interactions clearly improve the convergence behavior.

that interaction deeply improves the behavior of the algorithm. Indeed the chains of the MwG with interaction reach the neighbourhood of the true value of $r = 1.5$ faster than the chains of MwG without interaction.

4. Conclusion

This work showed that making parallel MCMC chains interact could improve their convergence properties. We presented the basic properties of the MCMC method, we did not prove that the proposed strategy speeds up the convergence. This difficult point is related to the problem of the rate of the convergence of the MCMC algorithms.

In the field of particle filtering, it has been shown that making different copies of the same Markov chain interact improves the mixture properties and thus the rate of convergence of the associated algorithms. The approach presented in the paper is inspired by such a technique. As shown by the two examples of Section 3, this argument also seems to be valid within the context of MCMC methods. Furthermore, this approach, as in the case of particle filtering, concentrates the computing effort in the relevant areas of space to be explored. Finally, the advantage of this method is to improve the convergence by mixing the parallel chains while maintaining the independence of the simulated sample.

Through simple examples we saw that the MwG strategy could be a poor strategy. In this situation our strategy improved the convergence properties.

Acknowledgements

This work is partially supported by the SARIMA project of the French Ministry of Foreign Affairs.

Appendix A. Proof of Lemma 1

This construction follows the general setup proposed by Tierney in [24]. The kernel is defined by:

$$P_\ell^i([\![Z,\xi,X]\!]_\ell^i;\mathrm{d}\xi') \stackrel{\text{def}}{=} \int_{\mathbb{R}^N} \underbrace{S_\ell^i([\![Z,\xi,X]\!]_\ell^i,\zeta^{1:N};\mathrm{d}\xi')}_{\text{selection kernel}} \times \underbrace{Q_\ell^i([\![Z,\xi,X]\!]_\ell^i;\mathrm{d}\zeta^{1:N})}_{\text{proposal kernel}}.$$

This kernel consists firstly of proposing a population of N candidates $\zeta^{1:N} \in \mathbb{R}^N$ sampled from:

$$Q_\ell^i([\![Z,\xi,X]\!]_\ell^i;\mathrm{d}\zeta^{1:N}) \stackrel{\text{def}}{=} \prod_{j=1}^N \pi_{i,j}^{\ell,\text{prop}}(\zeta^j|\xi)\,\mathrm{d}\zeta^j, \tag{A.1}$$

then secondly of selecting among these candidates or rejecting them according to a MH technique, i.e.

$$S_\ell^i([\![Z,\xi,X]\!]_\ell^i,\zeta^{1:N};\mathrm{d}\xi') \stackrel{\text{def}}{=} \frac{1}{N}\sum_{j=1}^N \alpha_\ell^{i,j}(\xi,\zeta^j)\delta_{\zeta^j}(\mathrm{d}\xi') + \tilde{\rho}_\ell^i(\xi,\zeta^{1:N})\delta_\xi(\mathrm{d}\xi') \tag{A.2}$$

where $\alpha_\ell^{i,j}$ is given by (5) and $\tilde{\rho}_\ell^i(\xi,\zeta^{1:N}) \stackrel{\text{def}}{=} 1-(1/N)\sum_{j=1}^N \alpha_\ell^{i,j}(\xi,\zeta^j)$.

Hence

$$\begin{aligned}
P_\ell^i([\![Z,\xi,X]\!]_\ell^i;\mathrm{d}\xi') &\stackrel{\text{def}}{=} \int_{\zeta^{1:N}} S_\ell^i([\![Z,\xi,X]\!]_\ell^i,\zeta^{1:N};\mathrm{d}\xi')Q_\ell^i([\![Z,\xi,X]\!]_\ell^i;\mathrm{d}\zeta^{1:N}) \\
&= \frac{1}{N}\sum_{j=1}^N \int_{\zeta^{1:N}} \alpha_\ell^{i,j}(\xi,\zeta^j)\,\delta_{\zeta^j}(\mathrm{d}\xi')\prod_{k=1}^N \pi_{i,j}^{\ell,\text{prop}}(\zeta^k|\xi)\,\mathrm{d}\zeta^k + \left\{1-\frac{1}{N}\sum_{j=1}^N \int_{\zeta^{1:N}} \alpha_\ell^{i,j}(\xi,\zeta^j)\prod_{k=1}^N \pi_{i,j}^{\ell,\text{prop}}(\zeta^k|\xi)\,\mathrm{d}\zeta^k\right\}\delta_\xi(\mathrm{d}\xi') \\
&= \frac{1}{N}\sum_{j=1}^N \int_{\zeta^j} \alpha_\ell^{i,j}(\xi,\zeta^j)\delta_{\zeta^j}(\mathrm{d}\xi')\,\pi_\ell^{\text{prop}}(\zeta^j|\xi)\,\mathrm{d}\zeta^j + \left\{1-\frac{1}{N}\sum_{j=1}^N \int_{\zeta^j} \alpha_\ell^{i,j}(\xi,\zeta^j)\pi_\ell^{\text{prop}}(\zeta^j|\xi)\,\mathrm{d}\zeta^j\right\}\delta_\xi(\mathrm{d}\xi') \\
&= \frac{1}{N}\sum_{j=1}^N \alpha_\ell^{i,j}(\xi,\xi')\pi_\ell^{\text{prop}}(\xi'|\xi)\,\mathrm{d}\xi' + \left\{1-\frac{1}{N}\sum_{j=1}^N \int_{\xi''} \alpha_\ell^{i,j}(\xi,\xi'')\,\pi_\ell^{\text{prop}}(\xi''|\xi)\,\mathrm{d}\xi''\right\}\delta_\xi(\mathrm{d}\xi')
\end{aligned}$$

which correspond to Eqs. (4)–(7).

Appendix B. Proof of Lemma 2

First let us consider the following lemma.

Lemma 4. *For almost all* $(\xi, \xi') \in \mathbb{R}^2$:

$$\alpha_\ell^{i,j}(\xi, \xi')\pi_\ell(\xi|Z_{1:\ell-1}^i, X_{\ell+1:n}^i)\pi_{i,j}^{\ell,\mathrm{prop}}(\xi'|\xi) = \alpha_\ell^{i,j}(\xi', \xi)\pi_\ell(\xi'|Z_{1:\ell-1}^i, X_{\ell+1:n}^i)\pi_{i,j}^{\ell,\mathrm{prop}}(\xi|\xi')$$

for any ℓ, i, j, $(Z_{1:\ell-1}^i, X_{\ell+1:n}^i)$, *and* $(Z_{1:\ell-1}^j, X_{\ell+1:n}^j)$.

Proof. For $(\xi, \xi') \notin R_\ell^{i,j}$, the result is obvious. For $(\xi, \xi') \in R_\ell^{i,j}$, i.e.:

$$\begin{aligned}
&(r_\ell^{i,j}(\xi, \xi') \wedge 1)\pi_\ell(\xi|Z_{1:\ell-1}^i, X_{\ell+1:n}^i)\pi_\ell^{\mathrm{prop}}(\xi'|\xi) \\
&\quad = \min\left\{\pi_\ell(\xi'|Z_{1:\ell-1}^i, X_{\ell+1:n}^i)\pi_\ell^{\mathrm{prop}}(\xi|\xi'), \pi_\ell(\xi|Z_{1:\ell-1}^i, X_{\ell+1:n}^i)\pi_\ell^{\mathrm{prop}}(\xi'|\xi)\right\} \\
&\quad = (\overset{i,j}{r}_\ell(\xi', \xi) \wedge 1)\pi_\ell(\xi'|Z_{1:\ell-1}^i, X_{\ell+1:n}^i)\pi_\ell^{\mathrm{prop}}(\xi|\xi').
\end{aligned}$$

The left hand side of equality (8) is a measure $\nu(\mathrm{d}\xi' \times \mathrm{d}\xi)$ defined on $(\mathbb{R}^2, \mathcal{B}(\mathbb{R}^2))$. For all $A_1, A_2 \in \mathcal{B}(\mathbb{R})$, we want to prove that $\nu(A_1 \times A_2) = \nu(A_2 \times A_1)$.

We have:

$$\nu(A_1 \times A_2) = \int P_\ell^i([\![Z, \xi, X]\!]_\ell^i; A_1)\mathbf{1}_{A_2}(\xi)\pi_\ell(\xi|Z_{1:\ell-1}^i, X_{\ell+1:n}^i)\,\mathrm{d}\xi$$

and

$$P_\ell^i([\![Z, \xi, X]\!]_\ell^i; A_1) = \frac{1}{N}\sum_{j=1}^N \int \mathbf{1}_{A_1}(\xi')\,\alpha_\ell^{i,j}(\xi, \xi')\pi_\ell^{\mathrm{prop}}(\xi'|\xi)\,\mathrm{d}\xi' + \rho_\ell^i(\xi)\mathbf{1}_{A_1}(\xi)$$

so that

$$\begin{aligned}
\nu(A_1 \times A_2) = \frac{1}{N}\sum_{j=1}^N \iint \mathbf{1}_{A_1}(\xi')\mathbf{1}_{A_2}(\xi)\alpha_\ell^{i,j}(\xi, \xi')\pi_\ell(\xi|Z_{1:\ell-1}^i, X_{\ell+1:n}^i)\pi_\ell^{\mathrm{prop}}(\xi'|\xi)\,\mathrm{d}\xi\,\mathrm{d}\xi' \\
+ \int \rho_\ell^i(\xi)\mathbf{1}_{A_1}(\xi)\mathbf{1}_{A_2}(\xi)\pi_\ell(\xi|Z_{1:\ell-1}^i, X_{\ell+1:n}^i)\,\mathrm{d}\xi
\end{aligned} \tag{B.1}$$

Using Lemma 4 we get:

$$\begin{aligned}
\nu(A_1 \times A_2) = \frac{1}{N}\sum_{j=1}^N \iint \mathbf{1}_{A_1}(\xi')\mathbf{1}_{A_2}(\xi)\alpha_\ell^{i,j}(\xi', \xi)\pi_\ell(\xi'|Z_{1:\ell-1}^i, X_{\ell+1:n}^i)\pi_\ell^{\mathrm{prop}}(\xi|\xi')\,\mathrm{d}\xi'\,\mathrm{d}\xi \\
+ \int \rho_\ell^i(\xi)\mathbf{1}_{A_1}(\xi)\mathbf{1}_{A_2}(\xi)\pi_\ell(\xi|Z_{1:\ell-1}^i, X_{\ell+1:n}^i)\,\mathrm{d}\xi
\end{aligned}$$

Exchanging the name of variables $\xi \leftrightarrow \xi'$ in the first term of the right hand side of the previous equality leads to the same expression as (B.1) where A_1 and A_2 were interchanged, in other words $\nu(A_1 \times A_2) = \nu(A_2 \times A_1)$. □

References

[1] S.P. Brooks, G.O. Roberts, Convergence assessment techniques for Markov chain Monte Carlo, Statistics and Computing 8 (1998) 319–335.

[2] F. Campillo, V. Rossi, Parallel and interacting Markov chains Monte Carlo method, Research Report INRIA n.6008, 2006.

[3] F. Campillo, R. Rakotozafy, V. Rossi, Computational probability modeling and Bayesian inference, in: M. Teuw Niane, G. Sallet, T. Sari, H. Touré(Eds.), ARIMA-special issue in honour of Claude Lobry, 9, 2008, 123-143.

[4] O. Cappé, A. Guillin, J.M. Marin, C.P. Robert, Population Monte Carlo, Journal of Computational and Graphical Statistics 13 (2004) 907–929.

[5] C.T. Chao, Markov chain Monte Carlo on optimal adaptive sampling selection, Environmental and Ecological Statistics 10 (2004) 129–151.
[6] D. Chauveau, P. Vandekerkhove, Algorithmes de Hastings-Metropolis en interaction, Comptes Rendus de l'Académie des Sciences, Série I Mathématique 333 (2001) 881–884.
[7] D. Chauveau, P. Vandekerkhove, Improving convergence of Hastings–Metropolis algorithm with an adaptive proposal, Scandinavian Journal of Statistics 29 (2002) 13–29.
[8] M.K. Cowles, B.P. Carlin, Markov chain Monte Carlo convergence diagnostics: a comparative review, Journal of the American Statistical Association 91 (1996) 883–904.
[9] M.M. Drugan, D. Thierens, Evolutionary Markov chain Monte Carlo, Lecture Notes in Computer Science 2936 (2004) 63–76.
[10] M.M. Drugan, D. Thierens, Recombinative EMCMC algorithms, IEEE Congress on Evolutionary Computation (2005) 2024–2031.
[11] A. Gelman, D.B. Rubin, Inference from iterative simulation using multiple sequences (with discussion), Statistical Science 7 (1992) 457–511.
[12] C.J. Geyer, Markov chain Monte Carlo maximum likelihood, in: E.M. Keramidas (Ed.), Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface, 1991.
[13] C.J. Geyer, Practical Markov chain Monte Carlo (with discussion), Statistical Science 4 (1992) 473–482.
[14] W.R. Gilks, S. Richardson, D.J. Spiegelhalter (Eds.), Markov Chain Monte Carlo in Practice, Chapman & Hall, London, 1995.
[15] W.R. Gilks, G.O. Roberts, Strategies for improving MCMC, in: W.R. Gilks, S. Richardson, D.J. Spiegelhalter (Eds.), Markov Chain Monte Carlo in Practice, Chapman & Hall, 1995.
[16] Y. Guan, R. Fleißner, P. Joyce, S.M. Krone, Markov chain Monte Carlo in small worlds, Statistics and Computing 16 (2006) 193–202.
[17] Y. Iba, Population Monte Carlo algorithms, Transactions of the Japanese Society for Artificial Intelligence 16 (2001) 279–286.
[18] A. Jasra, D.A. Stephens, C.C. Holmes, On population-based simulation for static inference, Statistics and Computing 17 (2007) 263–279.
[19] R.E. Kass, B.P. Carlin, A. Gelman, R.M. Neal, Markov chain Monte Carlo in practice: a roundtable discussion, The American Statistician 52 (1998) 93–100.
[20] K.B. Laskey, J.W. Myers, Population Markov chain Monte Carlo, Machine Learning 50 (2003) 175–196.
[21] K.L. Mengersen, C.P. Robert, Population Markov chain Monte Carlo: the pinball sampler, in: J.O. Berger, A.P. Dawid, A.F.M. Smith (Eds.), Bayesian Statistics, vol. 7, Oxford University Press, 2003.
[22] C.P. Robert, G. Casella, Monte Carlo Statistical Methods, Springer-Verlag, New York, 2004.
[23] L. Tierney, Markov chains for exploring posterior distributions (with discussion), The Annals of Statistics 22 (1994) 1701–1728.
[24] L. Tierney, A note on Metropolis–Hastings kernels for general state spaces, The Annals of Applied Probability 8 (1998) 1–9.