Bayesian inversion by parallel interacting Markov chains

Thomas Romary**
Centre de Géosciences, Equipe Géostatistique, Ecole des Mines de Paris,
35 rue Saint-Honoré ,
77300, Fontainebleau, France

April 15, 2009

Abstract

Markov chains Monte-Carlo (MCMC) methods are known to produce samples of virtually any distribution. They have already been widely used in the resolution of non-linear inverse problems where no analytical expression for the forward relation between data and model parameters is available, and where linearization is unsuccessful. However, in Bayesian inversion, the total number of simulations we can afford is highly related to the computational cost of the forward model. Hence, the complete browsing of the support of the posterior distribution is hardly performed at final time, especially when the posterior is high dimensional and/or multimodal. In the latter case, the chain may stay stuck in one of the modes. Recently, the idea of making interact several Markov chains at different temperatures has been explored. These methods improve the mixing properties of classical single MCMC. Furthermore, these methods can make efficient use of large CPU clusters, without increasing the global computational cost with respect to classical MCMC.

Keywords

Inverse problem ; Bayesian inversion ; MCMC ; interacting Markov chains ; tempering ; History matching

**Corresponding author. Email: thomas.romary@mines-paristech.fr

1 Introduction

Monte-Carlo methods are becoming increasingly important for the solution of nonlinear inverse problems. Typically, the inverse problem is formulated as a search for solutions fitting the data within a certain tolerance, given by data uncertainties. In a non-probabilistic setting this means that we search for solutions with calculated data whose distance from the observed is less than a fixed, positive number. In a Bayesian context, the tolerance is soft: a large number of samples of statistically near-independent models from the a posterior probability distribution are sought. Such solutions are consistent with data and prior information, as they fit the data within error bars, and adhere to soft prior constraints given by a prior probability distribution.

Precisely, we consider the study of a system $X \in \mathcal{X}$, on which we have an indirect measurement d, that is function of the state of X, modeled by $F(X)$, and some a priori information under the form of the prior distribution $\mathbb{P}(X)$. We also consider that the measurement d is affected by an error and that we know how to simulate F up to an approximation error, both errors being accounted for by $\mathbb{P}(d|X)$. We also define the joint distribution $\mathbb{P}(d, X)$. Then, assuming that all these distributions admit a density with respect to the Lebesgue measure, denoted $f(\cdot)$, the conditional density of X with respect to d takes the following form:

$$f(X|d) = \frac{f(d|X)f(X)}{\int_{\mathcal{X}} f(d, X)\mathrm{d}X}. \tag{1}$$

This is the Bayesian formulation of inverse problem and $\mathbb{P}(X|d)$, whose density $f(X|d)$, is the posterior distribution, see [1]. The formula (1) shows that this problem can be viewed as a classical statistical inference problem, where we want to sample independent realizations from the posterior distribution. Note that the normalization constant in (1) is generally intractable in high-dimensional problems. Therefore, we consider that the posterior is known up to a constant, being defined from the prior knowledge on the system studied and the data with its associated measurement error.

There exists several methods for solving (1) such as the Kitanidis-Oliver algorithm (see [2] and [3]), developed for petroleum engineering applications and the neighbourhood algorithm ([4] and [5]), developed for geophysical inverse problems. In spite of its universality the speed of convergence of the first one is controversial: it consists in performing a large number of optimizations with an observed datum perturbed according to its measurement error. It is particularly difficult to know how many optimizations should be performed. The second one seems to be limited for low-dimensional problems: it can be seen as a geometric version of an iterated importance sampling scheme (see *e.g.* [6], chapter 14). This article focus on Monte-Carlo Markov chains (MCMC) methods for their universality and the relative ease of their implementation.

MCMC methods suit indeed particularly for this problem, as they are known to produce samples of virtually any posterior distribution. Two problems may arise then. On one hand, the dimension of the problem may be so large that the chain has to be run for an intractable number of iterations to converge and to achieve an efficient sampling of the posterior, we say that they have weak mixing properties. On the other hand, an evaluation of the forward operator F can be very computer demanding so that the practitioner wishes to minimize the number of iterations. Moreover, when the posterior has several disconnected modes in a high-dimensional space, which is often the case in nonlinear Bayesian inversion, the problem of exploring the whole support of the posterior is a difficult one. It can be shown that even for very simple problems most classical Markov chain algorithms can fail at identifying the main modes of the posterior, because of their lack of mixing (see [7]).

We expose a method to improve the global efficiency of the Markov chain by generating a collection of chains in parallel at different temperatures and allowing them to interact. This method is not more computer demanding than classical MCMC since it can be easily parallelized.

This paper aims at providing researchers and engineers with some recipes to apply interacting MCMC methods. Thus, it begins in section 2 with basics for MCMC methods, some examples of classical algorithm and earlier attempts to improve mixing properties like annealing and tempering techniques, which rely on the same basic principles as interacting MCMC techniques, exposed in section 3. In section 4, we will show an application to a reservoir engineering problem. The paper ends with some conclusions and perspective of future work.

2 Markov chains Monte-Carlo methods

MCMC, introduced by Metropolis et al. [8], is a popular method for generating samples from virtually any distribution π defined on $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$, where $\mathcal{B}(\mathcal{X})$ stands for the Borel sets of $\mathcal{X}$. In particular there is no need for the normalizing constant of π to be known and the space $\mathcal{X} \subseteq \mathbb{R}^d$ (for some integer d) on which it is defined can be high dimensional. We recall here some classical results on MCMC methods. For a comprehensive review of MCMC, see [6], chapters 6 to 13. For a more detailed account on Markov chains theory, see [9].

2.1 Principles

The method consists in simulating an ergodic Markov chain $\{X_n, n \geq 0\}$ on $\mathcal{X}$ with transition probability P such that π is a *stationary* density for this chain, *i.e.* $\forall A \in \mathcal{B}(\mathcal{X})$:

$$\int_{\mathcal{X}} P(x, A)\pi(x)\mathrm{d}x = \pi(A). \tag{2}$$

Such samples can be used *e.g.* to compute integrals

$$\pi(h) = \int_{\mathcal{X}} h(x)\pi(x)\mathrm{d}x, \tag{3}$$

estimating this quantity by

$$S_n(h) = \frac{1}{n}\sum_{i=1}^{n} h(X_i), \tag{4}$$

for some $h : \mathcal{X} \to \mathbb{R}$. A very useful concept in constructing ergodic Markov chains is *reversibility*. A Markov chain is reversible if it satisfies the *detailed balance condition*:

$$P(x, dy)\pi(dx) = P(y, dx)\pi(dy). \tag{5}$$

This means that, if started in stationarity, the Markov chain has the same chance of starting at x and jumping to y as starting at y and jumping to x.

We illustrate the principles of MCMC with the Metropolis-Hastings (MH) update. It requires the choice of a *proposal distribution* q. The role of q consists in proposing potential transitions for the Markov chain. Given that the chain is currently at x, a candidate y is accepted with probability $\alpha(x, y)$ defined as:

$$\alpha(x, y) = \begin{cases} \min\left\{1, \frac{\pi(y)}{\pi(x)}\frac{q(y,x)}{q(x,y)}\right\} & \text{if } \pi(x)q(x, y) > 0, \\ 1 & \text{otherwise.} \end{cases} \tag{6}$$

Otherwise, it is rejected and the Markov chain stays at its current location x. The transition kernel P of this Markov chain takes the form, for $(x, A) \in \mathcal{X} \times \mathcal{B}(\mathcal{X})$:

$$P(x, A) = \int_A \alpha(x, y) q(x, y) \mathrm{d}y + \mathbb{1}_A(x) \int_{\mathcal{X}} (1 - \alpha(x, y)) q(x, y) \mathrm{d}y. \tag{7}$$

The Markov chain defined by P is reversible with respect to π and therefore admits π as invariant distribution. Conditions on the proposal distribution q that guarantee irreducibility and positive recurrence are easy to meet and many satisfactory choices are possible.

2.2 Some examples of Metropolis-Hastings samplers

The arbitrariness of the choice of $q(x, \cdot)$ allows considerable freedom to design a multitude of different chains, each with stationary distribution π, although in the Bayesian inversion framework, q should rely on the a priori distribution. Some examples include (see [6], chapter 7, for more examples):

1. the independent sampler (IMH): $q(x, y) = q(y)$, where q is generally the prior in Bayesian inversion,

2. the symmetric increments random-walk sampler (SIMH): $q(x, y) = q(|y - x|)$, where q can be a zero-mean version of the prior,

3. the Langevin sampler (LMH): assuming that π is differentiable on $\mathcal{X}$, it allows to take advantage of the gradient information to give the sampling direction, q takes the form:

$$q(x, y) \sim \mathcal{N}\left(x + \frac{h^2}{2} \nabla \log(\pi(x)), h^2 I_d\right), \tag{8}$$

   where h is a parameter to choose according to *e.g.* [10] or [11]. Note that a bad choice of h can induce erratic behaviour of the chain,

4. The adaptive algorithm of [12] (ASIMH): In this algorithm, y is proposed according to $q_{\theta_n}(x, \cdot) = \mathcal{N}(x, \Gamma_n)$, where $\theta = (\mu, \Gamma)$. We also consider a non-decreasing sequence of positive step sizes $\{\gamma_n\}$, such that $\sum_{n=1}^{\infty} \gamma_n = \infty$ and $\sum_{n=1}^{\infty} \gamma_n^{1+\delta} < \infty$ for some $\delta > 0$. In practice, we generally use: $\gamma_n = 1/n$, as suggested in [12]. The parameter estimation algorithm takes the following form:

$$\begin{aligned} \mu_{n+1} &= \mu_n + \gamma_{n+1} \left(X_{n+1} - \mu_n\right), \, n \geq 0, \\ \Gamma_{n+1} &= \Gamma_n + \gamma_{n+1} \left(\left(X_{n+1} - \mu_n\right) \left(X_{n+1} - \mu_n\right)^{tr} - \Gamma_n\right), \end{aligned} \tag{9}$$

5. The Gibbs sampler: Here $\mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_d$, and $q = q_i$ leaves all coordinates fixed except the i^{th} one, which it proposes according to the conditional distribution $(x_i | \{x_j\}_{j \neq i})$. This implies that $\alpha(x, y) = 1$ for all x and y, so there are no rejections. If the resulting i^{th} component Gibbs sampler is called P_i, then these components can be combined to yield the *random-scan Gibbs sampler* which is the average $P_{RS} = \frac{1}{d}(P_1 + \ldots + P_d)$, or the *deterministic-scan Gibbs sampler* which is the product $P_{DU} = P_1 \cdots P_d$.

2.3 Comments

One of the problems with Metropolis-Hastings algorithms is the abundance of choice available for choosing the proposal distribution $q(x, \cdot)$. For instance even if the type of algorithm

(perhaps the SIMH) has been chosen, it is necessary to scale the proposal variance to be appropriate for $\pi(\cdot)$. Such a problem is known as a scaling problem. To make this question more concrete, consider the following problem. Suppose that $q(x,\cdot)$ is distributed as the d-dimensional normal distribution $\mathcal{N}(x,\sigma^2 I_d)$, for some $\sigma^2 > 0$. We recall that the acceptance probabilities for this algorithm are given by (6). For very small values of σ^2, small jumps are attempted by the algorithm, and because of the form of (6), these moves are almost always accepted. The Markov chain mixes very slowly because its increments are so small. On the other hand, if σ^2 is chosen to be very large, long distance jumps are attempted by the algorithm, most of which are rejected. The algorithm therefore spends long periods of time in the same state, and thus the algorithm still converges slowly. For this problem, "very large" and "very small" have to be interpreted in a way related to the particular form of π. It seems reasonable that "moderate" values of σ^2 should be preferred. However, it is difficult to see how to figure out what values are "moderate", especially if π is very complicated. In Bayesian inversion context, the random walk type algorithms, like the SIMH or the LMH, generally fail at identifying different modes. In large dimensional space, the scaling factor generally has to be "small" so as to get an acceptable acceptance rate (6). Therefore, these two algorithms perform generally a local exploration and are hardly able to jump from one mode to another. We will then refer to them as "local" samplers. Note that they are also generally really slow to converge towards the stationary regime in Bayesian inversion context.

Conversely, the IMH does not need any tuning. It will explore largely the surface of the posterior distribution and we will refer to it as a "global" sampler. Nevertheless, in practical applications, unless q is the posterior distribution, the transitions will obviously almost always be rejected.

Finally, we can notice here that the chain generated by the adaptive algorithm is no longer homogeneous, but it can be proved (see [12], [13] and [14] in a more general framework) that it has the correct ergodic properties. The idea of adaptive sampling is to improve the proposal efficiency, making it as close as possible to the posterior density. However, it should be stressed here that the algorithm presented above generally fails in multi-modal context for a low number of iterations (see *e.g.* [7]). Regarding the Gibbs sampler, it does not seem to be well adapted to the Bayesian inversion problem: the important number of calls of the forward model limits its relevancy.

Due to the sequential nature of MCMC algorithm and to tackle multi-modality problems, MCMC practitioners generally use several chains that they run in parallel. By simulating several chains, variability and dependence on the initial value are reduced and it should be easier to control convergence to the stationary distribution by comparing the estimation, using different chains, of quantities of interest. However, good performances of these parallel methods require a degree of a priori knowledge on the distribution of interest π, in order to construct an initial distribution on $\mathcal{X}$ which takes into account the features of π (modes, shape of high density regions, etc.). This is rarely the case in Bayesian inversion. Moreover, in highly non-linear setups, like in Bayesian inversion, a slow mixing chain will presumably stay in the neighborhood of the starting point with a high probability (see [6] chapter 12 for a more thorough discussion).

Due to the complexity of the posterior distribution (*e.g.* multi-modality and/or disconnected support) in Bayesian inversion problems and classical limitations of MH algorithms, other methods than classical MH algorithm should be investigated. Simulated annealing and tempering, which are presented in the next paragraph, consists in studying modified versions of the posterior.

2.4 Simulated annealing and tempering

The simulated annealing algorithm has been introduced by [8], then generalized by [15] for optimization problems. It can be applied to both optimization and simulation problems (see [6] and reference therein). The simulated tempering has been introduced independently in [16] and [17].
The fundamental idea of these algorithms is that a change of scale, named temperature, allows larger moves on the surface of the distribution to explore, compared with classical MCMC methods. Indeed, this change of scale allows to avoid the chain to remain trapped in a local mode.
The name and inspiration of the first one come from annealing in metallurgy, a technique involving heating and controlled cooling of a material to increase the size of its crystals and reduce their defects. The heat causes the atoms to become unstuck from their initial positions (a local minimum of the internal energy) and wander randomly through states of higher energy; the slow cooling gives them more chances of finding configurations with lower internal energy than the initial one. Conversely the tempering is a brutal cooling followed by a a controlled reheating of the work piece to a temperature below its lower critical temperature. Precipitation hardening alloys, like many grades of aluminum and super alloys, are tempered to precipitate intermetallic particles which strengthen the metal.
These two methods aim particularly at generating samples from Gibbs distribution.

Definition 2.1 *A $\mathcal{X}$-valued random field X, is a Gibbs field of energy E, if its probability density function (with respect to the Lebesgue measure) is:*

$$f(x) = \frac{1}{\mathcal{Z}} e^{-E(x)},\ \mathcal{Z} = \int_{\mathcal{X}} e^{-E(x)} \mathrm{d}x, \tag{10}$$

named Gibbs density.

For practical problems, the constant $\mathcal{Z}$ is generally intractable due to the dimension of $\mathcal{X}$. Note that in a wide variety of inverse problems, the posterior distribution (1) takes the form (10); for instance when both prior and measurement error are assumed Gaussian. Note also that simulated annealing and tempering are not confined to cope with Gibbs distributions. We present here both algorithms in this framework for sake of simplicity. For other kind of target distributions π, the practitioner has to consider flattened versions given by $\pi_T = \pi^{1/T}$.

2.4.1 Simulated annealing

Given a positive temperature T, a Markov chain X is generated from the following Gibbs density:

$$\pi_T(x) \propto \exp\left(-E(x)/T\right). \tag{11}$$

The simulated annealing is performed by gradually lowering the temperature T from a high value to near-zero. Close to $T = 0$ the Gibbs distribution approximates a delta function at the global minimum for $E(x)$ (if it is unique). For simulation purposes, the cooling can be stopped at the value $T = 1$.

This algorithm can be viewed as a non-homogeneous version of the MH algorithm. Indeed, since T decreases along the algorithm, the kernel of the chain varies with time. Classical theoretical results on Markov chains does not apply for this algorithm. Heuristic rules are generally applied to ensure the validity of simulated annealing: the starting temperature must be high enough and its decrease slow enough. A convergence result exists, for optimization purpose, with the condition that T decreases as $1/\log(n)$. In practice, a geometrically decreasing sequence is generally used.

2.4.2 Simulated tempering

The principle of simulated tempering is linked to the simulated annealing one in the sense that we will again consider Gibbs distributions scaled by a temperature parameter T. However, this algorithm aims at sampling from a Gibbs distribution π rather than minimizing the energy of the system. We consider here a finite sequence of temperatures and the associated Gibbs distributions. In this algorithm, we authorize the chain to change temperature level according to a given probability. This will allow the chain to go back to higher temperatures, escaping eventual local modes of the target distribution, that results in better mixing.
We first define an increasing sequence of temperatures $1 = T_0 < \ldots < T_K$, with its associated Gibbs densities $\pi_i(x) \propto e^{-\frac{E(x)}{T_i}}$, an auxiliary $\{0, \ldots, K\}$-valued variable M, and the joint distribution:

$$\mu(x, m) = \rho_m \pi(x), \quad \sum_{i=0}^{K} \rho_m = 1. \tag{12}$$

We also define the probabilities p_U and p_D of moving "up", from m to $m+1$, and "down", from m to $m-1$, with only T changing, the chain being at temperature T_m, and the probability of choosing a fixed level move $(1 - p_U - p_D)$.
The principle is to simulate a $\mathcal{X} \times \{0, \ldots, K\}$-valued chain (X_n, M_n). Denoting respectively $q_{i \to i+1}(X_{n+1}|X_n = x_n)$ and $q_{i+1 \to i}(X_{n+1}|X_n = x_n)$ the probabilities of transition proposition towards the superior and the inferior temperature level, the acceptance probability of a transition from T_i to T_{i+1} is proportional to:

$$\rho_{i \to i+1}(x_n, x_{n+1}) = \frac{p_D}{p_U} \frac{\pi_{i+1}}{\pi_i} \frac{q_{i \to i+1}(X_{n+1} = x_{n+1}|X_n = x_n)}{q_{i+1 \to i}(X_{n+1} = x_{n+1}|X_n = x_n)}. \tag{13}$$

To maintain the detailed balance condition, it is then necessary that $\rho_{i+1 \to i}(x_{n+1}, x_n) = 1/\rho_{i \to i+1}(x_n, x_{n+1})$ and to choose the proposition distributions $q_{i \to i+1}(X_{n+1}|X_n = x_n)$ and $q_{i+1 \to i}(X_{n+1}|X_n = x_n)$ accordingly. Still, it is important to notice that (13) depends on the normalization constants of π_{i+1} and π_i. We can bypass this difficulty by designing an equal number of moves from m to $m+1$ and from $m+1$ to m and by accepting the entire sequence as a single proposal, thus canceling the normalizing constants in the acceptance probability, as described *e.g.* in [18].

2.4.3 Comments

Attempts to improve mixing properties of the chain by simulated annealing fail generally because of the monotonous decrease in temperature; if the chain gets in a local mode, it may be impossible to escape it if the temperature is already too low.

Concerning the simulated tempering, the potential gain in a better exploration of the support of the target distribution, so as to say, a better mixing, does not seem to compensate for the increased amount of forward operator evaluations for inverse problems ($2K$ bigger, for the scheme presented above). However, the presentation of this method is a good introduction to the interacting Markov chains algorithms exposed in the next section.

3 Parallel interacting Markov chains

The principle of making interact Markov Chains first appears in [19] under the name parallel tempering (PT). It has been mostly applied in physico-chemical simulations, see [20] and references therein. It is known in the literature under different names such as: exchange

Monte-Carlo, Metropolis coupled-chain, see [21] for a review. The principle of PT is to simulate a number $(K+1)$ of replica of the system of interest by MCMC, each at a different temperature, in the sense of the simulated annealing, and to allow the chains to exchange information, swapping their current state. The high temperature systems are generally able to sample large volumes of state space, whereas low temperature systems, whilst having precise sampling in a local region of state space, may become trapped in local energy minima during the timescale of a typical computer simulation. Parallel tempering achieves good sampling by allowing the systems at different temperatures to exchange their state. Thus, the inclusion of higher temperature systems ensures that the lower temperature systems can access a representative set of low-temperature regions of state space.

Simulation of $(K+1)$ replicas, rather than one, requires on the order of $(K+1)$ times more computational effort. This *extra expense* of PT is one of the reasons for the initially slow adoption of the method. Eventually, it became clear that a PT simulation is more than $(K+1)$ times more efficient than a standard, single-temperature Monte- Carlo simulation. This increased efficiency derives from allowing the lower temperature systems to sample regions of state space that they would not have been able to access, even if regular sampling had been conducted for a single-temperature simulation that was $(K+1)$ times as long. It is also worth noticing that PT can make efficient use of large CPU clusters, where different replicas can be run in parallel, unlike classical MCMC sampling that are sequential methods. An additional benefit of the PT method is the generation of results for a range of temperatures, which may also be of interest to the investigator. It is now widely appreciated that PT is a useful and powerful computational method.

More recently, some researchers in the statistical community took attention on PT and more generally on interacting Markov Chains. They propose a general theoretical framework and new algorithms in order to improve the exchange information step addressed above. Two main algorithms drawn our attention: the equi-energy sampler (EES) of [22] and the population importance-resampling MCMC sampler (PIR) of [23], which allows to go back in the history of the chain. More precisely, these two last algorithms are based on self interacting approximations of non-linear Markov kernels, defined by Andrieu et al. [23]. We now describe these methods in the Bayesian inversion context.

3.1 Description of the algorithms

We first recall that our aim is to simulate realizations from the posterior distribution (1). We assume that the posterior distribution $\pi(X) = f(X|d)$ takes the form of a Gibbs distribution, that is:

$$\pi(X) = \exp(-E(X)), \tag{14}$$

where $E(X)$ is the energy of the system at the state X. We first define the family $\{\pi^{(l)}, l = 0 \dots K\}$ of distributions we want to simulate from, such that:

$$\pi^{(l)}(x) \propto e^{-E_l(x)}, \tag{15}$$

where $E_l(x) = \frac{E(x)}{T_l}$, where T_l is the temperature at which the system under study is considered. The T_l satisfy: $T_0 = 1 < T_1 < \ldots < T_K < +\infty$, so that $\pi^{(0)} = \pi$. These distributions are thus a family of *tempered* versions of $\mathbb{P}(X|d)$. To go back to the analogy with the metallurgy, these distributions represent the states of the metal at each considered temperature. At high temperatures, the system can access to high energy states, whereas at low ones, it will attain lower energy, *i.e.* more stable states. We will also talk of *tempered* energies to

denote the E_l. The parallel algorithms aim to simulate from:

$$\Pi(x) = \prod_{l=0}^{K} \pi^{(l)}(x), \tag{16}$$

whilst allowing exchanges between states at different temperatures. Flattened versions of $\pi^{(0)}$: $\pi^{(1)}, \ldots, \pi^{(K)}$ are easier to simulate. Thus they can provide information on $\pi^{(0)}$. Particularly, the system at T_0 can exhibit a wide range of disconnected meta-stable states (*i.e.* the different modes of the posterior) and typically, a single Markov chain is not able to visit all of them in the time of the simulation. So, exchanging with states generated at higher temperature allows to explore better the support of the posterior.

Different strategies can be adopted to exchange information between chains at adjacent temperatures. For $l = 0, \ldots, K-1$, we define the importance function:

$$r^{(l)}(x) = e^{-(E_l(x) - E_{l+1}(x))}, \tag{17}$$

which is the un-normalized ratio of the distributions $\pi^{(l)}$ and $\pi^{(l+1)}$ at a given state x.
From now on, we denote by $x = (x^{(0)}, \ldots, x^{(K)}) \in \mathcal{X}^{K+1}$ the current state of the Markov chain that aims at simulating from Π, defined in (16). The method can be formalized by defining the following kernel P_n at time n, given all the previous states $x_{0:n-1} = (x_0, \ldots, x_{n-1})$ and for $A_0 \times \ldots \times A_K \in \mathcal{B}(\mathcal{X}^{K+1})$:

$$P_n(x_{0:n-1}; A_0 \times \ldots \times A_K) = P^{(K)}(x^{(K)}, A_K) \prod_{l=0}^{K-1} P^{(l)}_{x^{(l+1)}_{0:n-1}}(x^{(l)}, A_l), \tag{18}$$

where we simulate from $\pi^{(K)}$, the chain at the highest temperature T_K, using the classical MH kernel $P^{(K)}(\cdot, \cdot)$, whereas at the other temperatures, for $x^{(l+1)}_{0:n-1} \in \mathcal{X}^n$, $x^{(l)} \in \mathcal{X}$ and $A \in \mathcal{B}(\mathcal{X})$, we will use the heterogeneous Markov kernel:

$$P^{(l)}_{x^{(l+1)}_{0:n-1}}(x^{(l)}; A) = \theta P^{(l)}(x^{(l)}, A) + (1-\theta) \int_{\mathcal{X}} \nu^{(l)}_{x^{(l+1)}_{0:n-1}}(x^{(l)}, dy) T^{(l)}(y, x^{(l)}; A), \tag{19}$$

where,

$$\nu^{(l)}_{x^{(l+1)}_{0:n-1}}(x^{(l)}, dy) = \frac{\sum_{i=0}^{n-1} \omega^{(l)}_{n,i}(x^{(l)}, x^{(l+1)}_i) \delta_{x^{(l+1)}_i}(dy)}{\sum_{i=0}^{n-1} \omega^{(l)}_{n,i}(x^{(l)}, x^{(l+1)}_i)} \tag{20}$$

and in the three algorithms considered here $T^{(l)}$ will take the following form:

$$T^{(l)}(y, x^{(l)}; A) = \min\left\{1, \frac{r^{(l)}(y)}{r^{(l)}(x^{(l)})}\right\} \mathbb{1}_A(y) + \left(1 - \min\left\{1, \frac{r^{(l)}(y)}{r^{(l)}(x^{(l)})}\right\}\right) \mathbb{1}_A(x^{(l)}). \tag{21}$$

In other words, equation (19) states that at time step n, temperature T_l, with probability θ, a classical MH move will be performed according to the Markov kernel $P^{(l)}(x^{(l)}, A)$. Otherwise, with probability (1-θ), an exchange move will be proposed. It consists in choosing a state y among $x^{(l+1)}_{0:n-1}$, the past states of the chain at temperature T_{l+1}, from the empirical distribution $\nu^{(l)}_{x^{(l+1)}_{0:n-1}}$ (20). This move is then accepted or rejected according to $T^{(l)}$ (21). Precisely, going back to (17), it is accepted with probability:

$$\min\left\{1, \frac{r^{(l)}(y)}{r^{(l)}(x^{(l)})}\right\} = \min\left\{1, \exp\left(\left(\frac{1}{T_l} - \frac{1}{T_{l+1}}\right)\left(E(x^{(l)}) - E(y)\right)\right)\right\},$$

that is, if the energy of the proposed state y is lower than that of $x^{(l)}$, the exchange will be systematically accepted.

The empirical distribution $\nu^{(l)}_{x^{(l+1)}_{0:n-1}}$ can be viewed as an importance sampling estimate of $\pi^{(l)}$ with the instrumental law $\pi^{(l+1)}$ constructed from the past states $x^{(l+1)}_{0:n-1}$ of the chain at temperature T_{l+1}. Then, an exchange amounts to simulate directly from an approximate form of $\pi^{(l)}$. Note that this will *regenerate* the chain and hence reduce the autocorrelation along time.

The three algorithms (PT, EES, PIR), considered in this article can be written in this framework, and differ only in the formulation of the weights $\omega^{(l)}_{n,i}$. For some $(y,z) \in \mathcal{X}^2$, we have:

1. for the PT algorithm:
$$\omega^{(l)}_{n,i}(y,z) = \mathbb{1}_{i=n-1},$$
it is only possible to go to the current state of the chain at the adjacent higher temperature,

2. for the EES algorithm, given a sequence of energy levels $E_0 < E_1 < \ldots < E_K < E_{K+1} = \infty$ defining a partition: $\mathcal{X} = \bigcup_{l=0}^{K} \mathcal{X}_l$ of energy rings: $\mathcal{X}_l = \{x \in \mathcal{X} : E_l < E(x) < E_{l+1}\}$ and the function $I(x) = l$ if $x \in \mathcal{X}_l$, then the $\omega_{n,i}$ take the form:
$$\omega^{(l)}_{n,i}(y,z) = \mathbb{1}_{\mathcal{X}_{I(y)}}(z),$$
that is, the new state of the chain at temperature T_l will be taken uniformly among the states $x^{(l+1)}_{0:n-1}$ of the chain at temperature T_{l+1} that are in the same energy ring as the current state,

3. for the PIR algorithm, the weights $\omega_{n,i}$ take the following form:
$$\omega^{(l)}_{n,i}(y,z) = r^{(l)}(z),$$
i.e. we obtain the new state by resampling from $x^{(l+1)}_{0:n-1}$ with the weights ω.

The main idea behind the last two algorithms is that the kernel defined in (19) will converge towards the following limiting kernel:

$$P^{(l)}_{x^{(l+1)}_{0:n-1}}(x^{(l)}; A) = \theta P^{(l)}(x^{(l)}, A) + (1-\theta) R^{(l)}(x^{(l)}, A), \tag{22}$$

where $R^{(l)}$ is a MH kernel, whose proposal distribution is given by:

- $Q^{(l)}_{EES}(x, dy) \propto \pi^{(l+1)}(y) \mathbb{1}_{\mathcal{X}_{I(x)}}(y) \lambda(dy)$ for the EES,
- $Q^{(l)}_{PIR}(x, dy) = \pi^{(l)}(dy)$ for the PIR algorithm.

Obviously the convergence towards $R^{(l)}$ will not be achieved in the time of the simulation, but its approximation at time n will help to sample from the posterior, particularly to span a larger part of the state space.

Finally, it is worth noting that for all three algorithms, we can use the entire sample generated, reweighting the states according to the temperature by the following importance weights:

$$\eta^{(l)}(x) = e^{-(E_0(x) - E_l(x))}, \tag{23}$$

in order to compute estimates of $I_h = E_{\pi_0}[h(X)]$, for some h. Hence, the estimate $\hat{I}_h$, after N iteration of the algorithm will take the form:

$$\hat{I}_h = \sum_{l=0}^{K} \frac{\sum_{i=0}^{N} \eta^{(l)}(x_i^{(l)}) h(x_i^{(l)})}{\sum_{i=0}^{N} \eta^{(l)}(x_i^{(l)})}. \tag{24}$$

It has been shown numerically in [22] that using the reweighted entire sample will provide better estimates than using only $x_{0:N}^{(0)}$. Ergodic properties of the whole chain $X \in \mathcal{X}^{K+1}$ and asymptotic results (law of large number, central limit theorem) regarding (24) follow directly from the properties of each chain used (see [22] and [23]).

Concerning the choice of the parameters, some heuristic rules exist and are discussed in e.g. [21] for the PT algorithm and in [22] for the EES. Unfortunately, this kind of information does not exist yet in the literature for the PIR. The choice depends mainly on the problem addressed. We give below a few recipes to tune the parameters.

3.2 Tuning the parameters

As the algorithms proposed here are fairly new, we think that some comments from our experience can be useful for future practitioners. These guidelines are purely empirical, based on numerical experiments and our own reflection. We will focus on four different points for the EES and the PIR algorithms:

1. the kernel to choose, as a function of the temperature,
2. the sequence of temperature to choose,
3. the number of chains,
4. the probability of proposing exchange between chains.

As already claimed, the idea of applying these methods is to improve the mixing of the chain. Then we have to choose kernels that will make effective this assumption. At the highest temperature, large moves tend to be accepted, even though the energy level reached is not as low as the one finally aimed. Thus, it is of great interest to use a fast mixing kernel that cannot be used at lower energy levels because its transition would be rejected. We then recommend to use a "global" sampler like the independent sampler presented in section 2.2. However, the highest temperature has to be chosen so that the transition acceptance rate is high enough (see below). Conversely, at low temperatures, it is of interest to have a kernel with good local properties, like the Langevin sampler or a random walk with small steps, that will explore the posterior around the currently identified mode. The point is then to design the kernels between the highest and the lowest temperature levels. The difficulty is to choose kernels that progressively worsen their global properties, while increasing local properties, when descending the temperature ladder. We mean progressively in the sense that the exchange proposal acceptation rate has to stay at a satisfactory level between each chain. In this regard, the kernels proposed in [24] should be useful. In high-dimensional problems, the number of components affected at each transition should vary according to the temperature, modifying more components at high levels than at lower ones, see the application in section 4.

The sequence of temperatures has to be chosen so as to obtain a satisfactory exchange acceptation rate. In the literature about PT (*e.g* [20], [21]), most authors propose to distribute the temperature geometrically. In our applications, we followed this advice and it appeared to

work well. The problem is then to choose the highest temperature T_K and the number K, the lowest temperature being always 1. T_K has to be chosen according to the problem considered. Practically, a preliminary study of the energy has to be conducted. This study consists in the computation the energies for an i.i.d. sample $(x_1, \ldots, x_n)$ generated from the prior and calculating its mean $\frac{1}{n}\sum_{i=1}^{n} E(x_i)$. More precisely, assuming that the prior distribution on X is given by $g(X)$ and that the measurement error is Gaussian with identity covariance, F denoting the forward model, the posterior will take the following form:

$$\pi(X) \propto \exp(-E(x)) = \exp\left(-\frac{1}{2}\|d - F(X)\|^2 + \log(g(X))\right). \tag{25}$$

Considering that the realizations X from the posterior show no error, that is $F(X) = d$, the first term in the expression above vanishes and the energy of the system conditioned by d should be around $\mathbb{E}[-\log(g(X))]$, where $\mathbb{E}$ stands for the mathematical expectation, if the prior has been chosen correctly. The idea is then to choose the highest temperature T_K so as to have $\frac{1}{nT_K}\sum_{i=1}^{n} E(x_i) \approx \mathbb{E}[-\log(g(X))]$. Note that this rule works also when the measurement error is not Gaussian; it needs however to be centered. This choice will ensure a satisfactory transition acceptance rate when using the independent sampler affecting all the components at the highest level.

These considerations about kernels and temperatures are closely related to the number K of chains you use. Particularly, it is important not to employ a too big number. Indeed, using more chains will slow the input of information from the highest temperature level to the lowest, the one of interest. Conversely, the number of chains has to be large enough to allow them to exchange information at a good rate. The temperature ladder is then constructed distributing the temperatures geometrically between T_K and $T_0 = 1$. If the number of chains is sufficient, it allows generally a good overlapping of the histograms of the tempered energies, inducing the occurrence of exchanges. The number of temperatures to use should then be the minimum number that ensures a good overlapping of the histograms of the tempered energies.

Regarding the proposal rate of information exchange, there is again a balance to do between high and low rates. A high rate will encourage information exchange, but will slow local exploration. Conversely, a low rate will hamper the process of exchanging information. This depends highly on the dimension of the problem: local exploration is obviously slower in high dimensional spaces. It should generally be between 0.05 and 0.3.

As a conclusion, we can say that on each four points addressed here, there is a balance to make. The idea is to tune the different parameters in order to allow efficient information exchange, while allowing good local exploration at low temperatures and fast mixing at high ones. It may depend strongly on the problem to solve. However, as explained above, a preliminary study of the energies of an i.i.d. sample generated from the posterior should allow to tune satisfactorily the parameters. Like MCMC methods, every applied use of these methods requires instinct and understanding both about the underlying model and about the Markov chains being used.

3.3 PT, EES or PIR ?

Among the three algorithms proposed here, we claim that the PIR outperforms the two others for Bayesian inversion problems. It is clear that the PT has weaker properties than the two others because it does not account for the history of the chains. Comparing the EES and the PIR, we think the latter is the most suited for our problem. Indeed, considering that the lower is the temperature, the slower the chain will enter the stationary regime, we can remark that the PIR does not need the chains to be in stationary regime before to allow exchanges,

contrarily to the EES algorithm. Indeed, in the EES, the exchange proposal is made in the same energy ring as the current state. Then, if the chain of interest (*i.e.* at T_0) has not reached the stationary regime and is still at high energy levels, the exchange proposal will be in the same energy ring as the current state. Therefore, it will not help to attain stationary regime. Conversely, the PIR proposes exchange proposals according to an importance sampling step, constructed on the states generated at the higher adjacent temperature. The proposals are then more likely at low energy levels and helps the chain to enter faster the stationary regime.

4 Application to reservoir engineering

4.1 Introduction

In oil industry and subsurface hydrology, geostatistical models ([25]) are often used to represent the spatial distribution of different lithofacies in the reservoir. Two main model families exist: multiple point ([26]) and truncated Gaussian models ([27]). We focus here on the latter.

Conditioning the spatial distribution of different lithofacies in the reservoir to production data, such as cumulative oil production, water cut, is a highly challenging task in reservoir modeling. It consists in solving an ill-posed inverse problem: given a prior knowledge on the random field governing the lithofacies spatial distribution in the reservoir, typically a geostatistical model, we aim at finding multiple realizations of this model that will exhibit the same dynamical behaviour of the true reservoir. In other words, we want to sample from the posterior distribution defined in the Bayesian inversion framework. This will improve our knowledge on the reservoir and indicate us what should be the best exploitation strategy, where to dig new wells, in function of all the information gathered. The dynamical behaviour of a given realization is computed by a fluid-flow simulator F.

4.2 The case

We consider a case where the prior on the lithofacies distribution is a 2-dimensional thresholded Gaussian model (see *e.g.* [25]), with the following characteristics:

- its size is $2500 \times 2500 m^2$,
- it is discretized on a regular grid of $N = 50 \times 50$ blocks,
- it is 10 m thick,
- the underlying Gaussian random field X has an isotropic spherical covariance structure with a range equal to 600 m (a quarter the field edge size),
- it is composed of two lithofacies: A (50% with permeability 500md) and B (50% with permeability 10md),
- we put two wells in this field: an injector at grid node $(3, 3)$ and a producer at $(48, 48)$,
- the porosity is assumed constant at 0.25.

Practically, X is a Gaussian random field with mean zero and spherical covariance:

$$\Gamma(u, v) = 1 - 3\frac{\|u - v\|}{2a} - \frac{\|u - v\|^3}{2a^3}\mathbb{1}_{\|u-v\|<a},$$

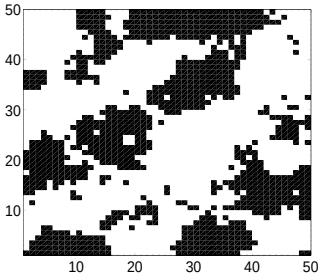


Figure 1: Reference lithofacies map.

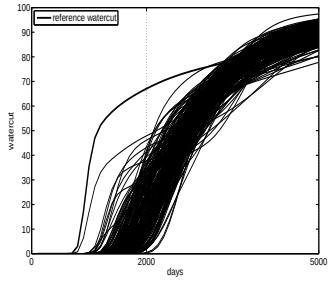


Figure 2: Water cut curves of an i.i.d. sample and reference.

whose range a is a quarter the edge of the field (see [25]). The lithofacies field is constructed thresholding X:

$$T(X) = \mathbb{1}_{X<0}.$$

When $T(X) = 1$, the corresponding cells will be facies A, otherwise B, with its associated permeability value.
The field is assumed to be saturated in oil at time zero. The fluid flow is simulated with *3DSL*© [28], a streamline fluid flow simulator, during 5000 days with an injection rate at 5000 m^3/day and a pressure of 200 *bars* at the producer.

Given a reference realization of the field X^* and water cut D^* computed on 2000 days, we attempt to condition the geostatistical model X to D^* (the water cut being the proportion of water in the oil produced at each time step). According to the methodology introduced in [29], [7], and [30], we choose to use a truncated Karhunen-Loève ([31]) expansion with $M = 100$ components to represent the field. Hence, this approximation reduces the dimension of the inference problem from 2500 to 100, whereas the fluid flow results remain slightly unchanged. The posterior distribution takes the following form:

$$P(X^{(M)}|D^*) \propto \mathrm{e}^{\left(-\frac{1}{2}\|D^*-F(X^{(M)})\|^2-\frac{1}{2}\|X^{(M)}-\mu\|^2_{\Gamma^{-1}_{(M)}}\right)}, \tag{26}$$

where $\Gamma_{(M)} = \Phi_{(M)}\Lambda\Phi^{tr}_{(M)}$,
$\Phi_{(M)}$ is a $L \times M$ matrix whose column vectors are the $\phi_i(x)$,
Λ is a diagonal $M \times M$ matrix whose diagonal components are the λ_i,

where $\phi_i(x)$ and λ_i are respectively the eigenfunctions and eigenvalues of $\Gamma_{(M)}$. Here, D^* and $F(X^{(M)})$ are both functions of time. The covariance of the measurement error on the water cut is assumed to be the identity matrix. We represent the reference realization of the field considered here in figure 1. We also represent in figure 2 the reference water cut curve together with a sample of curves computed for a sample of 200 independent realizations of the prior. This sample is used to tune the parameters of the algorithm as explained in section 3.2.

4.3 Choice of the parameters of the PIR algorithm

We can see in figure 1 that there is an important portion of highly permeable (500 md) facies (in white) in the diagonal axis linking the two wells. Figure 2 shows its particular water cut

profile: after the early water breakthrough (time when the water cut becomes strictly positive), the water cut increases very fast, then slows down. This profile is very different from that of the curves of 200 independent realizations of X. Indeed, the minimum energy calculated for this sample is about 3000, with an average around 20000, whereas we expect the energies to be around 50 for the matched sample (see section 3.2). That proves how challenging is our problem. In order to solve it, that is to sample from (26), we implement the PIR algorithm and a classical component-wise independent MH algorithm, that we will call single chain (SC) algorithm. The choice of the different parameters, set after some experiments, of the PIR algorithm is inspired by practical considerations given in section 3.2.

We use 5 different temperatures, distributed geometrically between $T_0 = 1$ and $T_4 = 400$. A geometric distribution of the temperatures is then chosen between the two extremal ones. Namely, we take $T_l = T_0 \left(\frac{T_4}{T_0}\right)^{l/4}$ for $l = 1, 2, 3$. Hence, we use the the following temperature ladder:

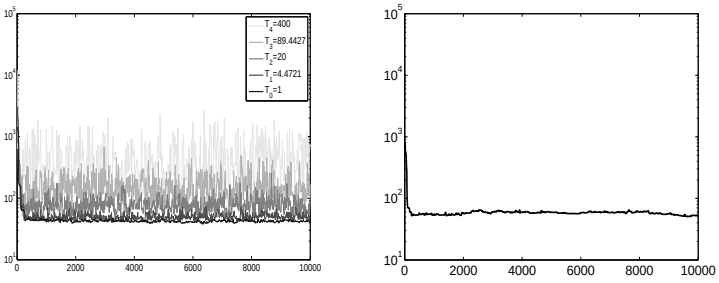
$$T_0 = 1.000 < T_1 = 4.729 < T_2 = 22.361 < T_3 = 105.737 < T_4 = 400.000.$$

Thus, we simulate the 5 Markov chains $(X^{(l)})$ at the temperature T^l. At T_0, T_1, T_2, we simulate from a symmetric increments random walk MH algorithm with a step variance $0.15\sqrt{T_l}$, affecting respectively 5, 20 and 50 components. At T_3, we simulate from an independent sampler affecting 80 components. At T_4, we simulate from a global independent sampler. In other words, proportionally to the temperature, we propose larger moves, using global samplers at the two highest temperatures. Modifying less components at low temperature results in better acceptance rates in our high dimensional space ($M = 100$) and allows local exploration of the posterior. Moreover, the moves at the highest temperatures affect more components, thus improve the mixing of these chains and feed the chains $(X^{(0)})$, $(X^{(1)})$, $(X^{(2)})$ with states, that they could not have attained without the exchange steps. After a few experiments, we allowed the chains to exchange information according to the PIR scheme just after the first iteration with a probability of 0.05, to ensure local exploration between exchange steps.

4.4 Results

We ran both algorithm for 10000 iterations. The PIR algorithm took 50 hours to run on a desktop computer with a single processor AMD Opteron 146 2.0GHz and the SC algorithm took about 10 hours. Note that having implemented the PIR algorithm on a parallel computer architecture, it would have taken the same time as the SC.
In figures 3*a* and 3*b*, we represent respectively the energy of the states of the 5 chains used in the PIR algorithm, and the energy of the states generated by the single chain.

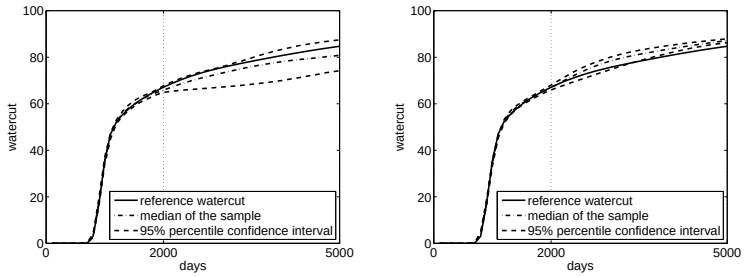
Figure 3*a* shows the energy of the states of the 5 chains, as a function of the number of iterations. For the lower curve, corresponding to T_0, we observe a stabilization after about 200 iterations, around levels of energy corresponding to the expected order of magnitude of the posterior mean energy. Indeed, allowing exchanges since the beginning of the chain accelerates its convergence. As all the other chains show a stabilized profile of energy after this number of iterations, we consider it as the end of the burn-in period, namely each chain is assumed in stationary regime beyond this number of iterations. Moreover, we can see that each couple of chains at adjacent temperatures show overlapping energy profiles, allowing the exchanges between the two chains. Indeed, the empirical exchange acceptance rate has been found between 0.6 and 0.8 for each couple of adjacent chains.
Figure 3*b* shows that the SC algorithm exhibits a rather fast convergence towards the stationary regime, attaining energy levels around 50 in about 250 iterations. This amazingly fast convergence is probably due to the starting state generated. It has an energy below 1000,



a. *b.*

Figure 3: Energy of the states, as a function of the number of iterations (a: 5 chains by PIR; b: SC).

much lower than those observed in our preliminary sample.
Figures 4*a* and 4*b*, show some statistics computed from the samples generated respectively by the PIR algorithm and by the SC algorithm, namely, the median and the 95% percentile confidence interval of the water cut curves generated together with the reference water cut.



a. *b.*

Figure 4: *Median, 95% percentile confidence interval and reference water cut (a. PIR algorithm, b. SC).*

In figure 4*a*, we can see that for the matched period (up to 2000 days), the median of the sample water cut perfectly matches the reference. Moreover, the 95% confidence interval is extremely thin around the reference water cut until 2000 days. Then it widens for the next 3000 days. In addition, the reference water cut stays in the 95% confidence interval and is quite close to the median. This validates our sample for prediction purposes.
Conversely, in figure 4*b*, although the reference water cut is also correctly matched by the sample generated by the SC algorithm, its prediction abilities are rather weak with respect to the PIR algorithm: the confidence interval generated is still thin beyond 2000 days and does not include the reference water cut. This is due to the only local exploration performed by this algorithm.

Figure 5 shows 7 realizations by the PIR and one by SC. First, the aspect of the realizations is far smoother than the reference. This is due to the approximation by a truncated Karhunen-Loève expansion with only $M = 100$ components. Second, the realizations generated by PIR (a to g) are clearly different between each other (we did not reproduce here the whole variety of maps generated). This illustrates the good exploration of the posterior (26) carried out by the PIR, due to the improved mixing properties with respect to classical single MCMC. Finally, all realizations generated by the SC are similar between each other (figure 5*h*). It

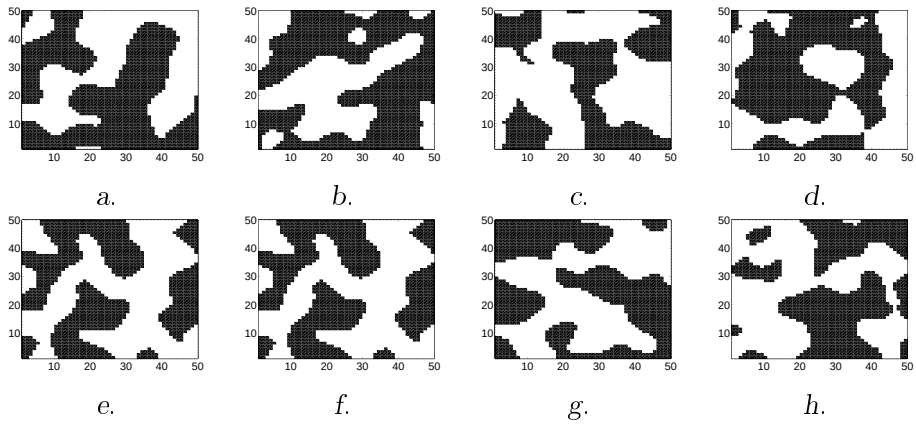


Figure 5: *7 realizations from the posterior generated by the PIR (a. to g.) and one generated by the SC (h.).*

has only performed a local exploration. Note that all the maps generated by both algorithms reproduce a link of highly permeable facies between the two wells.

Besides, it is worth noting that the PIR exhibits a global empirical acceptance rate of about 0.4, whereas the SC shows a empirical acceptance rate around 0.1. In other words, comparatively, we throw away twice more fluid-flow simulations with the SC than with the PIR.

To sum up the results on this synthetic test case, the PIR has shown improved mixing properties compared with the SC. It has provided a sample with good predictive properties, representative of different modes of the posterior.

5 Conclusion

In this work, we have first described the main principles of classical MCMC methods and related techniques simulated annealing and simulated tempering. We have then proposed an innovative application of a recent stochastic simulation method, based on parallel interacting Markov chains. We also provided some general guidelines for the tuning of the parameters of these algorithms. Finally, an application on a synthetic case of reservoir characterization has been performed. The numerical results show clearly that the PIR algorithm outperforms the single Markov chain for sampling the posterior. The sampling carried out by PIR explores better the posterior, therefore the sample produced has a better capacity of prediction. Moreover, this method is well suited for parallel computing, thus comparable with the classical MCMC in terms of computation costs.

The parallel interacting methods presented here, like other MCMC methods, aim at generating samples approximately distributed from a given distribution, without directly simulating from it. Although presented in the Bayesian inversion context, these methods can be applied in a wider range of application, for instance simulation problems in statistical mechanics (see the literature about parallel tempering and reference therein, *e.g.* [20] and [21]). Moreover, parallel interacting Markov chains algorithms can be easily combined with surrogate or approximate models approach, where a faster version of the forward model is used (see *e.g.* [32], [33], [34]).

Further improvements can be made on the parameterization of the parallel algorithm. It should be of great interest to imagine an automatic tuning of the kernels parameters, namely

the number of components affected at each iteration and the variance step used in random walks, according to the temperature.

Finally, the problem of integrating new data in an existing model can be performed in the following way: we could use either the above method with the kernel given by the final estimation of (22) or an importance sampling resampling scheme ([6]). The latter consists in proposing a realization with the weights given by (23), then reweighting them according to the adequation of new data.

References

[1] A. Tarantola, *Inverse Problem Theory and Model Parameter Estimation*, SIAM, Philadelphia, 2005.

[2] P.K. Kitanidis, *Quasi-linear geostatistical theory for inversing*, Water Resources Research 31 (1995), pp. 2411–2419.

[3] D.S. Oliver, *On conditional simulation to inaccurate data*, Mathematical Geology 28 (1996), pp. 811–817.

[4] M. Sambridge, *Geophysical inversion with a neighbourhood algorithm: I. Searching a parameter space*, Geophys. J. Int. 138 (1999), pp. 479–494.

[5] ———, *Geophysical inversion with a neighbourhood algorithm: II. Appraising the ensemble*, Geophys. J. Int. 138 (1999), pp. 727–746.

[6] C.P. Robert and G. Casella, *Monte-Carlo Statistical Methods*, 2nd edition, Springer, Berlin, 2004.

[7] T. Romary, *Integrating production data under uncertainty by parallel interacting Markov chains on a reduced dimensional space*, Computational Geosciences 13 (2009), pp. 103–122.

[8] N. Metropolis, A. Rosenbluth, M. Rosenbluth, and A.T.M. Teller, *Equations of state calculations by fast computing machines*, Journal of Chemical Physics 21 (1953), pp. 1087–1091.

[9] S.P. Meyn and R.L. Tweedie, *Markov Chains and Stochastic Stability*, Springer, Berlin, 1993.

[10] O. Stramer and R.L. Tweedie, *Self-targeting candidates for Metroppolis-Hastings algorithms*, Methodology and Computing in Applied Probability 16 (1999), pp. 307–328.

[11] G.O. Roberts and J.S. Rosenthal, *Optimal scaling of discrete approximations to Langevin diffusions*, Journal of the Royal Statistical Society B 60 (1998), pp. 255–268.

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

[13] C. Andrieu and C. P. Robert, *Controlled MCMC for Optimal Sampling*, Tech. Report, Cérémade, Université de PARIS - DAUPHINE, 2001.

[14] C. Andrieu and E. Moulines, *On the ergodicity properties of some adaptive MCMC algorithms*, Annals of Applied Probability 16 (2003), pp. 1462–1505.

[15] S. Kirkpatrick, C.D. Gelatt Jr., and M.P. Vecchi, *Optimization by simulated annealing*, Science 220 (1983), pp. 671–680.

[16] E. Marinari and G. Parisi, *Simulated tempering: a new Monte-Carlo scheme*, Europhysics Letters 19 (1992), pp. 451–458.

[17] C.J. Geyer and E.A. Thompson, *Annealing Markov chain Monte-Carlo with applications to ancestral inference*, Journal of the American Statistical Association 90 (1995), pp. 909–920.

[18] G. Celeux, M. Hurn, and C.P. Robert, *Computational and Inferential Difficulties with Mixture Posterior Distributions*, Tech. Report RR-3627, INRIA, 1999.

[19] C.J. Geyer, *Markov chain Monte-Carlo maximum likelihood*, in *Computing Science and Statistics: Proceedings of 23rd Symposium on the Interface Interface Foundation*, Fairfax Station, New-York, 1991, p. 156.

[20] D.J. Earl and M.W. Deem, *Parallel tempering : theory, applications, and new perspectives*, Physical Chemistry Chemical Physics 7 (2005), pp. 3910 – 3916.

[21] Y. Iba, *Extended ensemble Monte-Carlo*, International Journal of Modern Physics C 12 (2001), pp. 623 – 656.

[22] S.C. Kou, Q. Zhou, and W.H. Wong, *Equi-energy sampler with applications in statistical inference and statistical mechanics*, The Annals of Statistics 34 (2006), pp. 1581–1619.

[23] C. Andrieu, A. Jasra, A. Doucet, and P.D. Moral, *On Non-linear Markov Chain Monte-Carlo via self-interacting approximations*, Tech. Report, University of Bristol, 2007.

[24] L. Tierney, *Markov chains for exploring posterior distributions*, The Annals of Statistics 22 (1994), pp. 1701–1762.

[25] J.P. Chilès and P. Delfiner, *Geostatistics, Modeling Spatial Uncertainty*, John Wiley & Sons, New York, 1999.

[26] S. Strebelle, *Conditional simulation of complex geological structures using multiple-point geostatistics*, Mathematical Geology 34 (2002), pp. 1–22.

[27] C. Lantuéjoul, *Geostatistical Simulation*, Springer, Berlin, 2002.

[28] *3dsl User Manual*, StreamSim Technologies, Inc. Version 3.00 ed., (2008).

[29] T. Romary and L.Y. Hu, *Assessing the dimensionality of random fields with Karhunen-Loève expansion*, in *Petroleum Geostatistics 2007*, 2007.

[30] ———, *History matching of truncated Gaussian models by parallel interacting Markov chains on a reduced dimensional space*, in *ECMOR XI, 11th European Conference on the Mathematics of Oil Recovery*, 2008.

[31] M. Loève, *Probability Theory*, Van Nostrand, New York, 1955.

[32] D. Higdon, H. Lee, and Z. Bi, *A Bayesian approach to characterizing uncertainty in inverse problems using coarse and fine-scale information*, IEEE Transactions on Signal Processing 50 (2002), pp. 389–399.

[33] S. Balakrishnan, A. Roy, M.G. Ierapetritou, G.P. Flach, and P.G. Georgopoulos, *Uncertain reduction and characterization for complex environmental fate and transport models: an empirical Bayesian framework incorporating the stochastic response surface method*, Water Resources Research 39 (2003), pp. 1350.

[34] B. Jin, *Fast Bayesian approach for parameter estimation*, International Journal for Numerical Methods in Engineering 76 (2008), pp. 230–252.