Stat Comput (2006) 16:339–354
DOI 10.1007/s11222-006-9438-0

DRAM: Efficient adaptive MCMC

Heikki Haario · Marko Laine · Antonietta Mira · Eero Saksman

Received: July 2004 / Accepted: June 2006

Abstract We propose to combine two quite powerful ideas that have recently appeared in the Markov chain Monte Carlo literature: adaptive Metropolis samplers and delayed rejection. The ergodicity of the resulting non-Markovian sampler is proved, and the efficiency of the combination is demonstrated with various examples. We present situations where the combination outperforms the original methods: adaptation clearly enhances efficiency of the delayed rejection algorithm in cases where good proposal distributions are not available. Similarly, delayed rejection provides a systematic remedy when the adaptation process has a slow start.

Keywords Adaptive Markov chain Monte Carlo · Adaptive Metropolis-Hastings · Delayed rejection · Efficiency ordering

1 Introduction and motivation

Markov chain Monte Carlo (MCMC) methods allow to estimate $E_\pi f$, the expectation of a function f with respect to a distribution π, possibly known up to a normalizing constant. A Markov chain that has π as its unique stationary and limiting distribution is constructed and simulated. The mean of f along a realized path of the chain of length N, $\frac{1}{N}\sum_{i=1}^{N} f(X_i)$, is the MCMC estimator. Typically the mean is computed after a burn-in to allow the chain to reach its stationary regime. Under mild regularity condition (Tierney, 1994) the MCMC sampler is asymptotically unbiased and normally distributed.

In this paper we propose various strategies to combine two quite powerful ideas that have recently appeared in the MCMC literature: adaptive Metropolis samplers (Haario et al., 1999, 2001) and delayed rejection (Tierney and Mira, 1999; Green and Mira, 2001; Mira, 2002).

Delayed rejection (DR) is a way of modifying the standard Metropolis-Hastings algorithm (MH) (Tierney, 1994) to improve efficiency of the resulting MCMC estimators relative to Peskun (1973) and Tierney (1998) asymptotic variance ordering. The basic idea is that, upon rejection in a MH, instead of advancing time and retaining the same position, a second stage move is proposed. The acceptance probability of the second stage candidate is computed so that reversibility of the Markov chain relative to the distribution of interest, π, is preserved. The process of delaying rejection can be iterated for a fixed or random number of stages. Higher stage proposals are allowed to depend on the candidates so far proposed and rejected. Thus DR allows partial local adaptation of the proposal within each time step of the Markov chain still retaining the Markovian property and reversibility.

DR can also be considered as a way of combining different proposals for MH or different kernels for MCMC. There are other strategies suggested in the MCMC literature to combine kernels all having the proper stationary distribution, namely mixing and cycling (Tierney, 1994). The advantage of DR over these alternatives is that a hierarchy between kernels

H. Haario (✉) · M. Laine
Lappeenranta University of Technology, Lappeenranta, Finland
e-mail: haario@csc.fi

A. Mira
University of Insubria, Varese, Italy

E. Saksman
University of Jyväskylä, Jyväskylä, Finland

can be exploited so that kernels that are easier to compute (in terms of CPU time) are tried first, thus saving in terms of simulation time. Or moves that are more "bold" (bigger variance of the proposal, for example) are tried at earlier stages thus allowing the sampler to explore the state space more efficiently following a sort of "first bold" versus "second timid" tennis-service strategy. Similarly, again to allow for better exploration of the state space, global moves (i.e., updating all coordinates at once) could be tried first and local moves (updating single or groups of coordinates) could be attempted later.

The Adaptive Metropolis (AM) algorithm is the global adaptive strategy we will combine with the local adaptive strategy provided by the DR.

The intuition behind the AM is that, on-line tuning the proposal distribution in a MH can be based on the past sample path of the chain. Due to this form of adaptation the resulting sampler is neither Markovian nor reversible. In Haario et al. (2001) the authors prove, from first principles, that, under some regularity conditions on the way adaptation is performed and if the target distribution is bounded on a bounded support, the AM retains the desired stationary distribution.

The paper is organized as follows. In Sections 2 and 3 we give the details of the DR and of the AM strategies respectively.

We then propose different ways of combining DR with AM (Section 4) and prove the ergodicity of the resulting non-Markovian algorithms (Section 5).

In Section 6, various examples will be used to compare the proposed strategies in terms of efficiency of the resulting MCMC estimators as well as in terms of CPU simulation time.

2 Delayed rejection

In this section we give the details of DR. Suppose the current position of the Markov chain is $X_n = x$. As in a regular MH, a candidate move, Y_1, is generated from a proposal $q_1(x, \cdot)$ and accepted with the usual probability

$$\begin{aligned}\alpha_1(x, y_1) &= 1 \wedge \frac{\pi(y_1)q_1(y_1, x)}{\pi(x)q_1(x, y_1)}\\ &= 1 \wedge \frac{N_1}{D_1}. \qquad (1)\end{aligned}$$

Upon rejection, instead of retaining the same position, $X_{n+1} = x$, as we would do in a standard MH, a second stage move, Y_2, is proposed. The second stage proposal is allowed to depend not only on the current position of the chain but also on what we have just proposed and rejected: $q_2(x, y_1, \cdot)$.

The second stage proposal is accepted with probability

$$\begin{aligned}&\alpha_2(x, y_1, y_2)\\ &= 1 \wedge \frac{\pi(y_2)q_1(y_2, y_1)q_2(y_2, y_1, x)[1 - \alpha_1(y_2, y_1)]}{\pi(x)q_1(x, y_1)q_2(x, y_1, y_2)[1 - \alpha_1(x, y_1)]}\\ &= 1 \wedge \frac{N_2}{D_2}. \qquad (2)\end{aligned}$$

This process of delaying rejection can be iterated. If q_i denotes the proposal at the i-th stage, the acceptance probability at that stage is, following (Mira, 2001),

$$\begin{aligned}&\alpha_i(x, y_1, \ldots, y_i) = 1 \wedge\\ &\left\{\frac{\pi(y_i)q_1(y_i, y_{i-1})q_2(y_i, y_{i-1}, y_{i-2})\ldots q_i(y_i, y_{i-1}, \ldots, x)}{\pi(x)q_1(x, y_1)q_2(x, y_1, y_2)\ldots q_i(x, y_1, \ldots, y_i)}\right.\\ &\left.\frac{[1 - \alpha_1(y_i, y_{i-1})][1 - \alpha_2(y_i, y_{i-1}, y_{i-2})]\cdots[1 - \alpha_{i-1}(y_i, \ldots, y_1)]}{[1 - \alpha_1(x, y_1)][1 - \alpha_2(x, y_1, y_2)]\cdots[1 - \alpha_{i-1}(x, y_1, \ldots, y_{i-1})]}\right\}\\ &= 1 \wedge \frac{N_i}{D_i}. \qquad (3)\end{aligned}$$

If the ith stage is reached, it means that $N_j < D_j$ for $j = 1, \ldots, i-1$, therefore $\alpha_j(x, y_1, \ldots, y_j)$ is simply N_j/D_j, $j = 1, \ldots, i-1$ and we obtain the recursive formula

$$D_i = q_i(x, \ldots, y_i)(D_{i-1} - N_{i-1})$$

which leads to

$$\begin{aligned}D_i = {} & q_i(x, \ldots, y_i)[q_{i-1}(x, \ldots, y_{i-1})[q_{i-2}(x, \ldots, y_{i-2})\cdots\\ & [q_2(x, y_1, y_2)[q_1(x, y_1)\pi(x) - N_1]\\ & -N_2] - N_3]\cdots - N_{i-1}]. \qquad (4)\end{aligned}$$

Since all acceptance probabilities are computed so that reversibility with respect to π is preserved separately at each stage, the process of delaying rejection can be interrupted at any stage that is, we can, in advance, decide to try at most, say, 3 times to move away from the current position, otherwise we let the chain stay where it is. Alternatively, upon each rejection, we can toss a p-coin (i.e., a coin with head probability equal to p), and if the outcome is head we move to a higher stage proposal, otherwise we stay put.

In Tierney and Mira (1999) the DR strategy is proved to outperform the standard MH in the Peskun absolute efficiency ordering. This means that, using the DR, we obtain MCMC estimators that have a smaller asymptotic variance for every function f whose expectation relative to π we want to estimate (provided f has finite variance under π).

3 Adaptive MCMC

In this section we briefly introduce the AM strategy, for more details and theory see (Haario et al., 2001). The basic idea is to create a Gaussian proposal distribution with a covariance matrix calibrated using the sample path of the MCMC chain. The crucial point regarding the AM adaptation is how the covariance of the proposal distribution depends on the history of the chain. We take, possibly after an initial non-adaptation period, the Gaussian proposal to be centered at the current position of the Markov chain, X_n, and set its covariance to be: $C_n = s_d \mathrm{Cov}(X_0, \ldots, X_{n-1}) + s_d \varepsilon I_d$, where s_d is a parameter that depends only on the dimension d of the state space on which π is defined and $\varepsilon > 0$ is a constant that we may choose very small. Here I_d denotes the d-dimensional identity matrix. In order to start the adaptation procedure an arbitrary strictly positive definite initial covariance, C_0, is chosen according to a priori knowledge (which may be quite poor). A time index, $n_0 > 0$, defines the length of the initial non-adaptation period and we let

$$C_n = \begin{cases} C_0, & n \le n_0 \\ s_d \mathrm{Cov}(X_0, \ldots, X_{n-1}) + s_d \varepsilon I_d, & n > n_0. \end{cases} \tag{5}$$

Recall the definition of the empirical covariance matrix determined by points $X_0, \ldots, X_k \in \mathbb{R}^d$:

$$\mathrm{Cov}(X_0, \ldots, X_k) = \frac{1}{k}\left(\sum_{i=0}^{k} X_i X_i^T \;-(k+1)\overline{X}_k\ \overline{X}_k^T\right), \tag{6}$$

where $\overline{X}_k = \frac{1}{k+1}\sum_{i=0}^{k} X_i$ and the elements $X_i \in \mathbb{R}^d$ are considered as column vectors. Substituting (6) in definition (5), we get that, for $n > n_0$, the covariance C_n satisfies the recursive formula:

$$C_{n+1} = \frac{n-1}{n}C_n + \frac{s_d}{n}\big(n\overline{X}_{n-1}\overline{X}_{n-1}^T - (n+1)\overline{X}_n\overline{X}_n^T + X_n X_n^T + \varepsilon I_d\big). \tag{7}$$

which permits the calculation of the covariance matrix without excessive computational cost since the mean, $\overline{X}_n$, also satisfies an obvious recursive formula.

This form of adaptation was proved to be ergodic in Haario et al. (2001). In numerical applications, some helpful observations have emerged. The choice for the length of the initial non-adaptive portion of the simulation, n_0, is free, but, obviously, the larger it is, the longer it takes for the effect of adaptation to take place. In the earlier, non-ergodic version of the algorithm presented in Haario et al. (1999), it was found that the adaptation should not be done at each time step, but only at given time intervals. This form of adaptation improves the mixing properties of the algorithm also with AM. So the index n_0, in fact, can be used to define the length of non-adaptation periods during the whole chain.

The role of the parameter ε is just to ensure that, theoretically, C_n will not become singular. In most practical cases ε can be safely set to zero. Following (Gelman et al., 1995), we take the scaling parameter to be $s_d = 2.4^2/d$. In Gelman et al. (1995) the authors show that, in a certain sense, this choice optimizes the mixing properties of the MH search in the case of Gaussian targets and Gaussian proposals.

4 DRAM: Combining DR and AM

The success of MCMC methods, in general, depends on how well the proposal distribution fits the target distribution. In its basic formulation, DR employs a given number of fixed proposals that are used at the different stages. Therefore, the success of the DR strategy depends largely on the fact that at least one of the proposals is successfully calibrated. The intuition behind adaptive strategies is to learn from the information obtained during the run of the chain, and, based on this, to efficiently tune the proposals. There are, in principle, numerous ways of combining AM or MH within the DR framework, as indicated in the discussion in Section 7.

We shall follow here a rather direct way of combining AM adaptation with an m-stages DR algorithm:

- The proposal at the first stage of DR is adapted just as in AM: the covariance C_n^1 is computed from the points of the sampled chain, no matter at which stage these points have been accepted in the sample path.
- The covariance C_n^i of the proposal for the ith stage ($i = 2, \ldots, m$) is always computed simply as a scaled version of the proposal of the first stage, $C_n^i = \gamma_i C_n^1$.

The scale factors γ_i can be freely chosen: The proposals of the higher stages can have a smaller or larger variance than the proposal at earlier stages. The simulation results in Green and Mira (2001) suggest that it is more beneficial, in terms of asymptotic variance reduction of the resulting estimators, to have larger variance at earlier stages and then reduce the variance upon rejection. This is also confirmed by our simulations.

Guidelines on how to construct more elaborated proposals that make explicit use of rejected candidates are given in Tierney and Mira (1999), Green and Mira (2001) and Mira (2001). In particular, rejected proposals can be used to recenter and rescale higher stage proposals. These methods are, however, not employed here, since the aim of this paper is to point out that already a straightforward combination of DR and AM, with a very basic adaptation mechanism, may be very helpful.

From the DR strategy point of view, the rational of the approach we propose is to adapt, via AM, the first stage proposal to better fit the target distribution. If the variance is too large or small, the points obtained from earlier iterations will change it in the right direction (either increasing or decreasing it)—this indeed is how AM typically works.

From the AM point of view, clear benefits are expected, too. It sometimes may be difficult to get the adaptation process started. This happens if the initial guess for the proposal distribution is far from a correct one. AM typically recovers well if the initial proposal variances are too small, see Example 1 below. However, if the variance of the proposal is too large, or if the covariance for the proposal is nearly singular, practically no proposals are accepted, and as a consequence the adaptation process does not get started, see Examples 2 and 3 below. The DR framework provides a natural remedy for these situations: By reducing the variance of the proposals at higher stages we ensure that some points will be accepted. Once this happens, the AM adaptation usually starts working properly.

In Section 6, we shall present the merits of the DRAM combination in light of concrete examples. The coding work required for DRAM is slightly more involved than the rather straightforward extension that AM implies on top of the basic MH. However, if one already has codes for both AM and DR, the combination of them is easily coded. The likelihood function is Gaussian in all the examples here, so all the simulations could be performed by the same generic DRAM code. Naturally, the code contains AM, DR and MH as special cases.

5 Ergodicity of DRAM

In order to study the properties of the simulation provided by the non-Markovian DRAM algorithm we first fix some notation and define the stochastic process corresponding to the algorithm. We follow mainly the approach and notation of Haario et al. (2001), to which we refer for unexplained concepts.

In this section we focus on two-stages DR algorithms but the theory can be generalized to multiple-stages DR strategies with more than two attempts to move.

Denote by $q_C(x, y)$ the density of a d-dimensional Gaussian proposal with covariance matrix C centered at x (thus y has the distribution $N(x, C)$). We shall assume that $S \subset \mathbb{R}^d$ is a Borel-measurable subset of the Euclidean space, and the stationary distribution of the Markov chain, $\pi : S \to [0, \infty)$, is a probability density on S (we will also denote by π the associated measure). As explained in Section 2, given two proposals one may always define a corresponding delayed rejection transition probability function (DR-tpf). We formalize this into a definition:

Definition 1. Let π and S be as defined above and let C^1, C^2 be given covariance matrices. The corresponding two-stages Gaussian DR-tpf is denoted by Q_{C^1,C^2}.

In order to give an explicit formula for Q_{C^1,C^2} we write (compare with Section 2)

$$\alpha_1(x, y) = 1 \wedge \frac{\pi(y)}{\pi(x)}, \tag{8}$$

where we understand that $\pi(x) = 0$ for $x \notin S$ and α_1 takes the value 1 if $\pi(x) = 0$. Moreover,

$$\alpha_2(x, y, y') = 1 \wedge \frac{\pi(y) q_{C^1}(y, y')(1 - \alpha_1(y, y'))}{\pi(x) q_{C^1}(x, y')(1 - \alpha_1(x, y'))}. \tag{9}$$

Comparing the above formula with (2) one should notice the cancellation of the second stage proposals due to its symmetry. We are now able to define, for any Borel-measurable subset $A \subset S$ such that $x \notin A$

$$\begin{aligned} Q_{C^1,C^2}(x; A) = &\int_A q_{C^1}(x, y')\alpha_1(x, y')dy' \\ &+ \int_A \Big(\int_{\mathbb{R}^d} q_{C^1}(x, y')(1 - \alpha_1(x, y')) \\ &\times q_{C^2}(x, y)\alpha_2(x, y', y)\, dy' \Big)\, dy. \end{aligned} \tag{10}$$

The definition of the transition probability function is completed by setting

$$Q_{C^1,C^2}(x; \{x\}) = 1 - Q_{C^1,C^2}(x; S\backslash\{x\}). \tag{11}$$

For later need we quantify the dependence of Q_{C^1,C^2} on the covariance matrices. The following technical lemmas serves this purpose. The derivative D^k_{ij} in the Lemma 1 are taken with respect to the (i, j)th element $(i, j = 1, \ldots, d)$ of the covariance matrix C^k $(k = 1, 2)$. The easy proof of Lemma 1 is left to the reader. In the sequel we denote by $a_1, a_2, \ldots$ generic positive constants whose actual value is of no direct interest.

Lemma 1. *Let $S \subset \mathbb{R}^d$ be bounded. Assume that the covariances C^1, C^2 satisfy the matrix inequality ($A \le B$ means that $B - A$ is a non-negative definite matrix)*

$$a_1 I_d \le C^1, C^2 \le a_2 I_d, \tag{12}$$

where $0 < a_1 < a_2 < \infty$. Then there are finite positive constants a_3, a_4 that depend only on S, a_1, a_2 such that the inequalities

$$\frac{|D^k_{ij} q_{C^k}(x, y')|}{q_{C^k}(x, y')} \le a_3(1 + |y'|^2) \tag{13}$$

and

$$\frac{|D^1_{ij} \alpha_2(x, y', y)|}{\alpha_2(x, y', y)} \le a_4(1 + |y'|^2) \tag{14}$$

hold for all $y' \in \mathbb{R}^d$ and $x, y \in S$. Here $1 \le k \le 2$ and $1 \le i, j \le d$ are arbitrary.

We should remark here that α_2 is not necessarily differentiable in the strict sense, but (14) should be interpreted as an estimate of the local Lipschitz constant.

Lemma 2. *Let $S \subset \mathbb{R}^d$ be bounded and assume that the covariances $C^1, \widetilde{C}^1, C^2, \widetilde{C}^2$ satisfy the matrix inequality* (12). *Then there is a constant a_5 such that*

$$\begin{aligned} &|Q_{C^1,C^2}(x, A) - Q_{\widetilde{C}^1,\widetilde{C}^2}(x, A)| \\ &\quad \le a_5(\|C^1 - \widetilde{C}^1\| + \|C^2 - \widetilde{C}^2\|) \end{aligned} \tag{15}$$

for all $x \in S$ and measurable $A \subset S$.

Proof: In order to prove (15) we first consider the case $C^2 = \widetilde{C}^2$. By (11) we may also assume that $x \notin A$. We obtain, from (10), that

$$\begin{aligned} &|Q_{C^1,C^2}(x, A) - Q_{\widetilde{C}^1,C^2}(x, A)| \\ &\quad \le \int_0^1 \left|\frac{d}{ds} h_1(s)\right| ds + \int_0^1 \left|\frac{d}{ds} h_2(s)\right| ds, \end{aligned}$$

where

$$h_1(s) = \int_A q_{C(s)}(x, y)\alpha_1(x, y)dy$$

with $C(s) = s\widetilde{C}^1 + (1 - s)C^1 = C^1 + s(\widetilde{C}^1 - C^1)$, and

$$\begin{aligned} h_2(s) = \int_A \bigg(\int_{\mathbb{R}^d} & q_{C(s)}(x, y')(1 - \alpha_1(x, y')) \\ & \times q_{C^2}(x, y)\alpha_2(x, y', y)\, dy'\bigg) dy. \end{aligned}$$

Observe that α_2 depends on $C(s)$. The matrix $C(s)$ clearly satisfies the inequalities (12) for all $s \in [0, 1]$. Hence the previous lemma applies and we obtain the estimate

$$\begin{aligned} \left|\frac{d}{ds} h_1(s)\right| &\le a_6 a_3 \|C^1 - \widetilde{C}^1\| \sup_{y \in S}(1 + |y|^2) h_1(s) \\ &\le a_7 \|C^1 - \widetilde{C}^1\| \end{aligned}$$

since $h_1 \le 1$ and S is bounded. Here the distance between the covariances is measured in the usual matrix (L2) norm.

Similarly we compute

$$\begin{aligned} &\left|\frac{d}{ds} h_2(s)\right| \\ &\le a_6 \|C^1 - \widetilde{C}^1\| \int_A \bigg(\int_{\mathbb{R}^d} (a_3 + a_4)(1 + |y'|^2) q_{C(s)}(x, y') \\ &\qquad (1 - \alpha_1(x, y')) q_{C^2}(x, y)\alpha_2(x, y', y)\, dy'\bigg) dy \\ &\le a_8 \|C^1 - \widetilde{C}^1\| \int_{\mathbb{R}^d} (1+|y'|^2) q_{C(s)}(x, y')dy' \int_A q_{C^2}(x, y)dy \\ &\le a_9 \|C^1 - \widetilde{C}^1\| \sup_{x \in S} \int_{\mathbb{R}^d} (1 + |y'|^2) q_{C(s)}(x, y')dy' \\ &\quad \le a_{10} \|C^1 - \widetilde{C}^1\|. \end{aligned}$$

In the last estimate we used the fact that $C(s)$ satisfies bounds similar to (12).

By combining the obtained estimates, the claim follows in the case $C^2 = \widetilde{C}^2$. The case $C^1 = \widetilde{C}^1$ is similar, although easier since α_2 does not depend on C^2. By combining the two cases the general statement is proved. □

The sequence (K_n) of generalized transition probability functions defining the DRAM algorithm (with second covariance proportional to the first one) is given by

$$K_n(x_0, \dots, x_{n-1}; A) = Q_{C_n, \gamma C_n}, \tag{16}$$

where C_n is the covariance obtained from the history of the algorithm as defined in (5). The constant $\gamma > 0$ is fixed, see Section 4. The proof of ergodicity of DRAM algorithm is based on Theorem 2 in Haario et al. (2001) which we recall here (Theorem 3 below) for the readers convenience. First we need to define a "freezed" transition probability. Given a generalized transition probability K_n (where $n \ge 2$) and a *fixed* $(n - 1)$-tuple, $(y_0, y_1, \dots y_{n-2}) \in S^{n-1}$, we denote $\widetilde{y}_{n-2} = (y_0, y_1, \dots y_{n-2})$ and define the transition probability $K_{n,\widetilde{y}_{n-2}}$ by

$$K_{n,\widetilde{y}_{n-2}}(x; A) = K_n(\widetilde{y}_{n-2}, x; A) \tag{17}$$

for $x \in S$ and $A \subset S$. Above S^{n-1} stands for the $(n - 1)$-fold product space. For the definition of the (Dobrushin) coefficient of ergodicity, $\delta(K)$, we refer to Haario et al. (2001) (p. 228).

Theorem 3. *Assume that (K_n) satisfies the following three conditions* (i)–(iii):

(i) *There is an integer k_0 and a constant $\lambda \in (0, 1)$ such that*

$$\delta((K_{n,\widetilde{y}_{n-2}})^{k_0}) \le \lambda < 1 \quad \text{for all } \widetilde{y}_{n-2} \in S^{n-1} \text{ and } n \ge 2.$$

(ii) *There is a probability measure π on S and a constant $c_0 > 0$ so that*

$$\|\pi K_{n,\widetilde{y}_{n-2}} - \pi\| \leq \frac{c_0}{n} \quad \text{for all} \quad \widetilde{y}_{n-2} \in S^{n-1} \quad \text{and} \quad n \geq 2.$$

(iii) *The estimate for the operator norm*

$$\|K_{n,\widetilde{y}_{n-2}} - K_{n+k,\widetilde{y}_{n+k-2}}\|_{\mathcal{M}(S)\to\mathcal{M}(S)} \leq c_1 \frac{k}{n},$$

holds, where c_1 is a positive constant, $n, k \geq 1$ and we assume that the $(n+k-1)$-tuple $\widetilde{y}_{n+k-2}$ is a direct continuation of the $(n-1)$-tuple $\widetilde{y}_{n-2}$. Here $\mathcal{M}(S)$ stands for the finite signed measures on S, and $\|\cdot\|_{\mathcal{M}(S)\to\mathcal{M}(S)}$ denotes the operator norm in the space of bounded measures.

Then, if $f : S \to \mathbb{R}$ is bounded and measurable, it holds almost surely that

$$\lim_{N\to\infty} \frac{1}{N+1}(f(X_0) + f(X_1) + \cdots + f(X_N)) = \int_S f(x)\pi(dx). \tag{18}$$

We are ready to show that the DRAM algorithm yields asymptotically unbiased estimators for expected values of functions f with respect to π. The conditions of the theorem are commented in the remark after the proof.

Theorem 4. *Let π be the density of a target distribution supported on a bounded measurable subset $S \subset \mathbb{R}^d$ and assume that π is bounded from above. Then the DRAM algorithm, as described in Section* 4 (*see also* (16)) *is ergodic in the sense of* (18).

Proof: We show that the transition probabilities (16) fulfill conditions (i)–(iii) of Theorem 3. Observe first that by (5) and boundedness of S the empirical covariances C_n satisfy a uniform estimate as in (12) with constants depending only on S, d and ε. Hence the corresponding densities $q_{C_n}(x, y)$ are uniformly bounded from below for $x, y \in S$ and the first term in the formula (10) easily yields the estimate

$$K_{n,\widetilde{y}_{n-2}}(x \; A) \geq a_3\pi(A)$$

since π is bounded from above. This is well known to yield condition (i) with $k_0 = 1$ (compare (Haario et al., 2001, p. 230)).

In order to check condition (ii) we fix $\widetilde{y}_{n-2} \in S^{n-1}$ and denote $C^* = C_{n-1}(\widetilde{y}_{n-2})$. By the very definitions (5), (6) it follows that

$$\|C^* - C_n(\widetilde{y}_{n-2}, y)\| \leq a_{10}/n, \tag{19}$$

where a_{10} does not depend on $y \in S$. We may hence apply Lemma 2 to deduce, for all measurable $A \subset S$, that $|K_{n,\widetilde{y}_{n-2}}(y; A) - Q_{C^*,\gamma C^*}(y; A)| \leq a_{11}/n$, which in turn implies that $\|K_{n,\widetilde{y}_{n-2}} - Q_{C^*,\gamma C^*}\|_{\mathcal{M}(S)\to\mathcal{M}(S)} \leq 2a_{11}/n$. By Tierney and Mira (1999) the delayed rejection kernel satisfies $\pi Q_{C^*,\gamma C^*} = \pi$, and we obtain

$$\|\pi - \pi K_{n,\widetilde{y}_{n-2}}\| = \|\pi(Q_{C^*,\gamma C^*} - K_{n,\widetilde{y}_{n-2}})\| \leq \frac{2a_{11}}{n},$$

as desired.

Finally, the verification of condition (iii) is based on Lemma 2, which gives that

$$\begin{aligned}
&\|K_{n,\widetilde{y}_{n-2}} - K_{n+k,\widetilde{y}_{n+k-2}}\|_{\mathcal{M}(S)\to\mathcal{M}(S)} \\
&\quad \leq 2 \sup_{y\in S, A\in\mathcal{B}(S)} |K_{n,\widetilde{y}_{n-2}}(y; A) - K_{n+k,\widetilde{y}_{n+k-2}}(y; A)| \\
&\quad \leq 2a_5(1+\gamma) \sup_{y_1,\ldots,y_{n+k-2}\in S} \|C_n - C_{n+k}\| \leq a_{12}k/n,
\end{aligned}$$

where the last inequality follows from definition (5). □

Remark. One can generalize the proof of (Haario et al., 2001, Theorem 2) and consequently the proof of Theorem 3 above, to obtain a stronger result with less restrictive (but more implicit) assumptions, see (Atchade and Rosenthal, 2005). Moreover, Andrieu and Moulines (2002) present an approach that can be used to study a modified algorithm in the case where the support of π can be unbounded (see also (Andrieu and Robert, 2001) for a general framework of adaptation where the connection to stochastic optimization was observed). We expect the ergodicity of DRAM to hold under quite minimal assumptions, especially without the extra smoothness and strong decay of π assumed in Andrieu and Moulines (2002).

The proof given here is valid, without changes, also for the modifications of the DRAM algorithm, where for example the second stage covariance is kept fixed all the time, or adapted only after prescribed periods.

6 Examples

In this section we present three examples of which the last one is a real high dimensional application to real data. The first two examples are, instead, artificially constructed to show that when either one of the two building blocks of DRAM, namely DR and AM, are poorly calibrated, the combination of them almost automatically solves the problems that would appear running each one of them separately. In particular we stress the fact that, in our examples, the second stage proposal in the DR strategy has been scaled down by a constant factor always taken to be $\gamma = 0.01$ while in

Green and Mira (2001) suggest using a down scale factor of 0.5 starting from a first stage proposal which is overdispersed.

We first employed the target distributions already used in Haario et al. (1999, 2001) as test cases. More specifically, we used both correlated Gaussian distributions, and 'nonlinear banana-shaped' distributions. These distributions allow to an exact computation for the, e.g. 50% and 90% probability regions, so the correctness of the MCMC runs can be easily verified. As representative cases we present the results of test sets in Example 1 below.

In Example 2 we demonstrate with a simple but realistic case a situation where neither DR nor AM alone works properly but the combination of the two performs quite well.

Example 3 concludes with a real high dimensional modeling problem, where convergence to the target distribution seems to be difficult to achieve by any method. However, we see that again DRAM provides the most reliable tool among the choices tested.

In all the test runs we have compared the results obtained from the basic Metropolis-Hastings (MH), Adaptive Metropolis (AM), basic Delayed Rejection (DR) and the combination DR+AM (DRAM). In all the examples we always update all the coordinates together.

6.1 Example 1

We first want to test situations where the proposal distributions are selected to have clearly too small or too large variances with respect to the target distribution. We use correlated Gaussian targets in various dimensions. The condition number of the covariance matrix (the ratio between the largest and smallest eigenvalue of the matrix, the largest being scaled to 1) is fixed to be $c = 2$ here. We start with a Gaussian proposal that has unit diagonal covariance matrix scaled by the optimal factor $s_d = 2.4^2/d$ following (Gelman et al., 1995). We will refer to this as the 'basic proposal'. To create a proposal with too small variance, this matrix is multiplied by the factor 0.01, both for MH, AM, DR and DRAM. In DR and DRAM we used a second stage proposal obtained by further scaling down the proposal of the first stage by a factor $\gamma = 0.01$.

As it is well known, in this setting the MH algorithm tends to "walk around" the target distribution with small steps, without effectively exploring the state space. The same is naturally true for DR if all the proposals have a variance which is too small. The point here is to see how the AM adaptation is able to overcome this problem.

Figures 1 and 2 present typical outcomes of the runs in a two dimensional setting. Figure 1 gives the results of MH and AM, while Fig. 2 exhibits the results produced by DR and DRAM, respectively. The top parts exhibit the sampled points as well as the 50% and 90% probability regions. The lower parts of the figures give the proportion of the chain points within the 50% and 90% probability regions during the runs. We can see that the adaptation indeed seems to remove the problem caused by too small variance in proposal distributions for both the MH and the DR.

For more reliable statistics, we performed the runs repeatedly with increasing dimensions, $d = 2, 5, 10, 15, \ldots, 50$, for the correlated Gaussian target distribution described above. The chain length was fixed to 20000 steps for all dimensions. Otherwise the settings are the same as above, the variance of the basic proposal distribution was again scaled down by 0.01. For each dimension, the runs were repeated 100 times. The simulations were started by randomly generating a point from the target.

For moderately low dimensions, $d < 15$, roughly, the results were not sensitive with respect to the length n_0 of the non-adaptation period, one might simply take $n_0 = 1$. For higher dimensions, it seems that AM with $n_0 = 1$ might become 'too' adaptive, and works better if the adaptation is slowed down with a larger value of n_0. As a rule of thumb, a value of a few hundred for n_0 may used for low dimensional problems, and a few thousand for larger dimensions ($d > 15$ here).

The true mean of the target distribution was always set at the origin, so the norm (the Euclidean L2 distance from the origin) of the average value of the chain can be used as a measure of the error of the estimate for the expectation. The mean values over the Monte Carlo repetitions were also computed for the proportion of the chain points inside the 50% and 90% probability regions.

Figure 3 shows the estimated errors (computed over 100 independent chains) of our estimators for the center point of the distributions. We see that the adaptive algorithms clearly outperform the MH and DR runs, where the estimate of the center point of the distribution may be strongly biased.

Figure 4 shows the results for the 50% and 90% regions. We can see that for moderate dimensions, up to around $d = 15$, the performance of all algorithms are comparable. For higher dimensions, the adaptation seems to put too many points in the central part of the target distribution. Increasing the chain length to, e.g., 200,000 for the higher dimensions, would provide a remedy for this problem (see Haario et al., 2005), but here we have intentionally kept the chain length fixed regardless of the dimension in order to reveal differences between the various methods. The combination, DRAM, improves but does not remove this feature of adaptation here, since the second stage proposal was chosen to have a "too small" covariance matrix. Methods for adaptation in high dimensional problems are studied elsewhere, e.g., in Haario et al. (2005), therefore here we will focus on simulations in moderate dimensions.

The computational times greatly depend on implementation—here the runs were performed with a

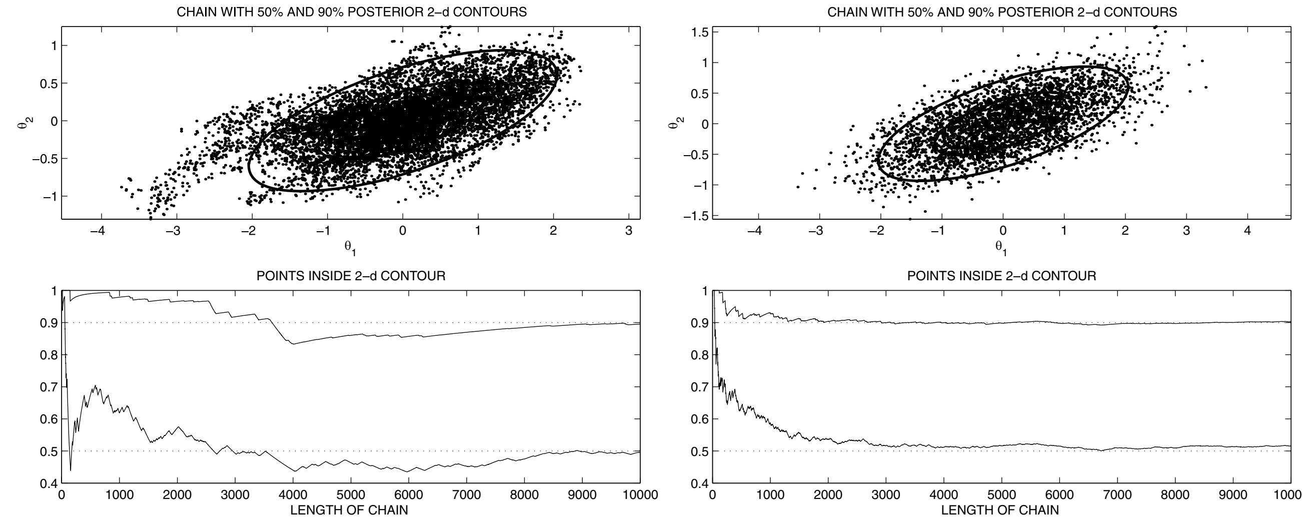


Fig. 1 Left figures: results by MH, with too small variance for proposal. Right figures: results by AM, started with the same proposal distribution as with MH. The upper figures present the sampled points as well as the 50% and 90% probability regions. The lower figures give the proportion of the chain points within the 50% and 90% probability regions during the runs

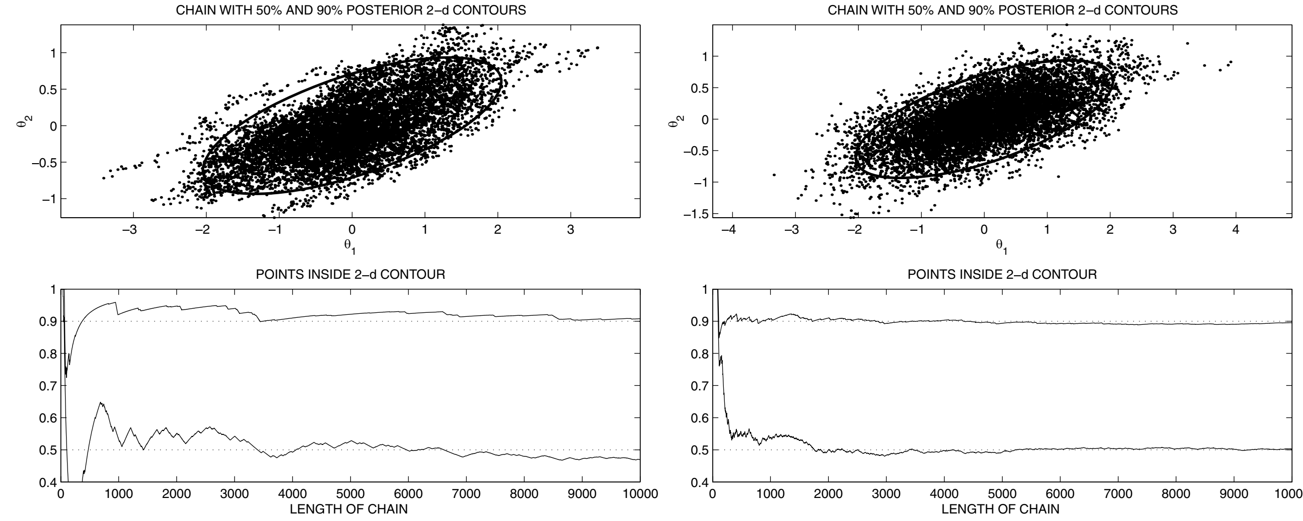
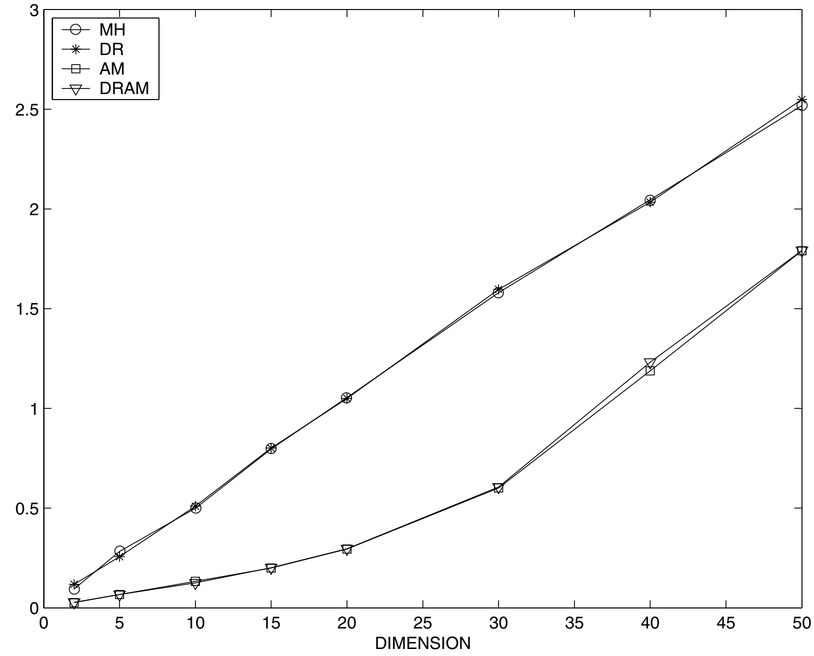


Fig. 2 Left figures: results by DR, with too small variance for proposals. Right figures: results by DRAM, started with the same proposal distributions as with DR. The upper figures present the sampled points as well as the 50% and 90% probability regions. The lower figures give the proportion of the chain points within the 50% and 90% probability regions during the runs

Matlab code—and we only give relative figures. The basic MH algorithm always took the least CPU time. The relative CPU times of AM, DR and DRAM were, roughly, larger by factors 1.1, 1.2 and 2.3, respectively. This is natural, since the adaptation slightly increases the computational cost of AM compared to MH. As for DR, the cost here is not essentially larger, since the first stage proposal, intentionally selected to have a very small variance, is typically accepted. DRAM adapts the proposal of the first stage to the 'correct' size, so that the second stage is often also used. This explains its higher computational cost factor 2.3. The acceptance rates were about 90% for MH and nearly 100% for DR (calculated as the combined acceptance from both stages) for all dimensions. With DRAM the acceptance rates slightly decreased from 94% to 80% as the dimension increases, while, for AM, the acceptance rates increase with dimensions from around 30% to 50%, indicating that, for adaptations, a longer chain would have been needed for the higher dimensional settings.

Next, we run basically the same tests as above, but select proposals which have too large variances, so that the AM adaptation has difficulties to get started. The covariance of the 'basic proposal distribution' is now multiplied by a factor of 4 (instead of 0.01 as above) to obtain the fixed proposal for MH, initial for AM, first stage proposal for DR, and initial first stage for DRAM. The size of this factor was

Fig. 3 Errors in the estimates of the center point of the Gaussian distribution with $c = 2$. Average results from 100 simulations for cases with too small proposal variances



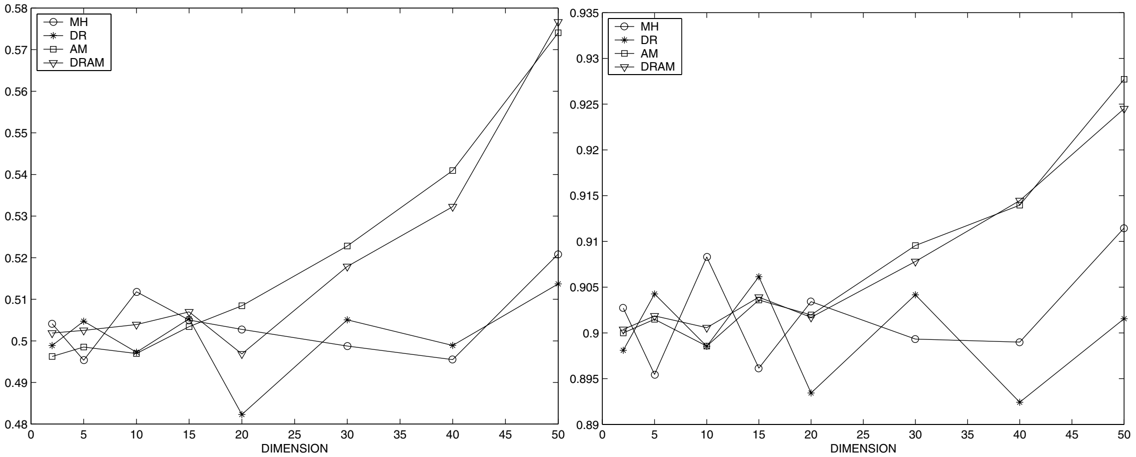


Fig. 4 Proportions of sampled points in the 50% (left) and 90% (right) probability regions. Average results from 100 simulations for cases with too small proposal variances

mainly chosen to get the AM adaptation at least started in higher dimensions. In fact, with an essentially larger initial proposal, practically no new points would be accepted, and no adaptation would take place. The shrinking factor, γ, for the second stage proposal in DR and DRAM was kept at the value 0.01.

As above, we run repeated simulations for increasing dimensions. We see that AM seems to outperform MH and, likewise, DRAM seems to outperform DR in computing the expected value of the distribution, see Fig. 5. The picture, however, does not reveal the true behavior of all the algorithms. In fact, for the MH algorithm in high dimensions, essentially no proposal is accepted and the results in Fig. 5 mainly reflect the fact that the starting point was sampled from the target distribution itself. The reason of the rejection is that the variance of the proposal is too large, and its covariance does not fit the one of the target, so the acceptance rate strongly decreases as the dimensions increases. Due to the same reason, the AM adaptation has difficulties getting started in high dimensions. The decreasing acceptance rates with AM again reflect the fact that longer chains would have been needed.

We can observe that the results of both MH and AM are clearly improved by the DR strategy. Now DRAM properly works in all dimensions tested and gives the best estimates for the center point of the target distribution.

The relative computational times with AM were again, roughly, larger than those with MH by a factor of 1.1. With DR and DRAM the factors were about 2.9 and 2.6 respectively. Now DR typically uses both proposal stages, which

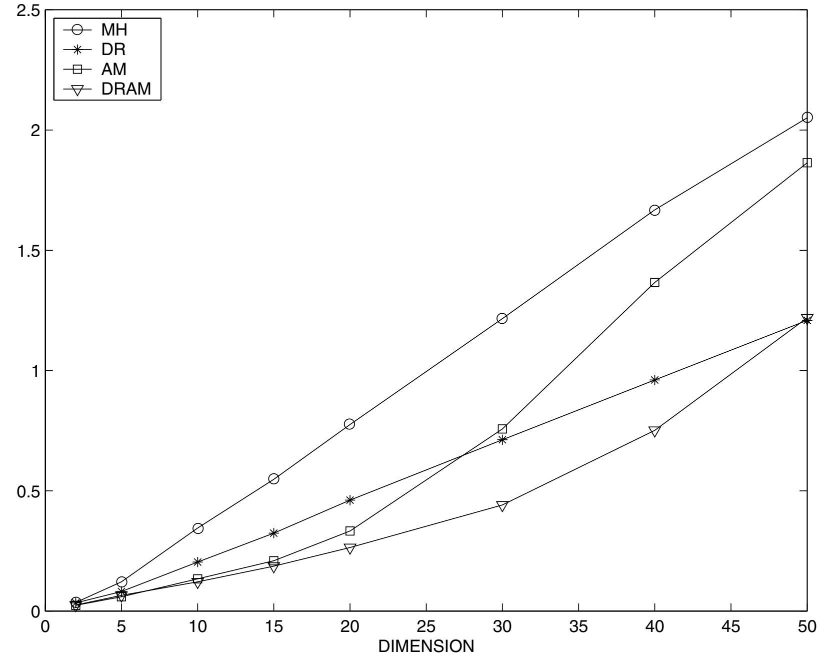


Fig. 5 Errors in the estimates of the center point of the distribution. Average results from 100 simulations for cases with too large proposal variances

explains the higher factor. DRAM adapts the first stage proposal and thus uses the second stage less frequently, which explains the somewhat lower CPU times.

6.2 Example 2

Our next example presents a situation where neither AM nor DR work properly alone, but their combination, DRAM, is quite efficient. Consider a simple chemical reaction $A \rightleftarrows^{k_1}_{k_2} B$, where a component A goes to B in a reversible manner, with reaction rate coefficients k_1 and k_2. The dynamics are given by the ODE system:

$$\frac{dA}{dt} = -k_1 A + k_2 B, \quad \frac{dB}{dt} = k_1 A - k_2 B,$$

with initial values fixed as $A_0 = 1$, $B_0 = 0$ at $t = 0$. We are interested in estimating k_1 and k_2 when data for, e.g., $A(t) = k_2/(k_1 + k_2) + (k_1/(k_1 + k_2))e^{-(k_1+k_2)t}$ has been obtained at given sampling times of t. Suppose now that the data has been sampled too late, in the sense that the reaction has already reached a steady-state equilibrium at the sampling times, cf. Fig. 6. It is clear that from such data the values of the parameters can not be separately determined, only the ratio k_1/k_2 may be identified, as well as lower bounds for k_1 and k_2. Without priors, the possible values for k_1 and k_2 would lie in a practically infinite "zone" in a direction where k_1/k_2 is constant.

Synthetic data was created for parameter values $k_1 = 2$, $k_2 = 4$ at time points $t = 2, 4, 6, 8, 10$. Zero mean Gaussian noise with standard deviation of size 0.01 was added. For the prior we set a broad Gaussian distribution with center point at (2, 4) and deviation equal to 200 for both parameters. The goal is to sample from the posterior with MH, DR, AM and DRAM.

While given here in a simplistic setting, situations of this type are, in fact, rather often faced in real-life parameter estimation of dynamical systems, see, for instance, Haario et al. (1999). Some parts of the dynamics are very fast, or internal structural characteristics of the model lead to strongly correlated parameter combinations. In more complex situations, it may not be easy to observe the correlations beforehand. MCMC methods should work in these situations, too. Indeed, they can provide a good tool for studying the identifiability of the parameters.

A standard procedure would be to estimate the parameters by least squares fitting, compute the covariance matrix of the parameters by the approximate Hessian matrix, use it to construct the proposal distribution for MH, and perform the MCMC run using the fitted parameters values as the starting point for the chain. This is what is done in the test runs below. However, in the setting of our example there is no unique minimum for the least squares function, and the covariance matrix is nearly singular.

Figure 6 shows typical runs with DR and AM. The computed approximate covariance does not provide a good proposal, and the efficiency remained very low for DR. The acceptance rate with the first stage proposal—that is, with the MH proposal—was around 0.6%. For the second stage we used scaled versions of the first one, both with smaller and larger variances. At the second stage the acceptance rate was at best around 5–7%. We may conclude that while DR is

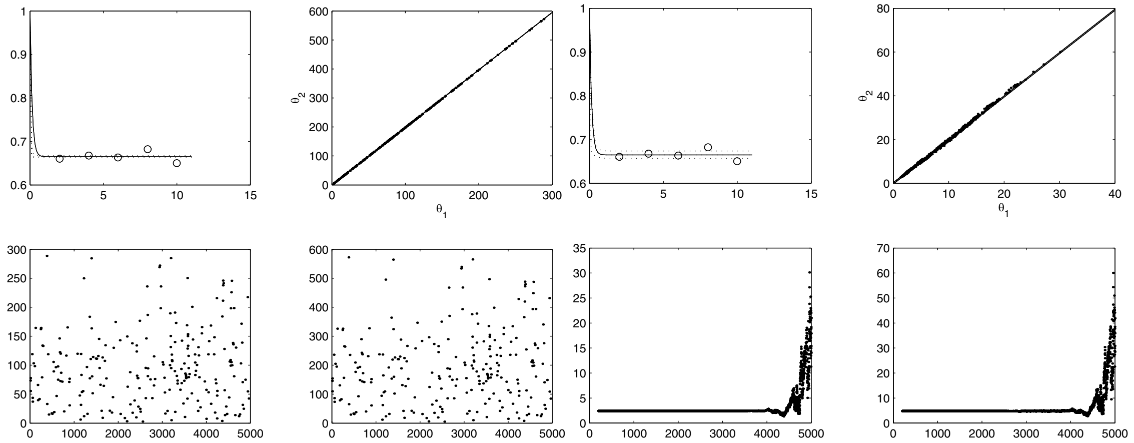
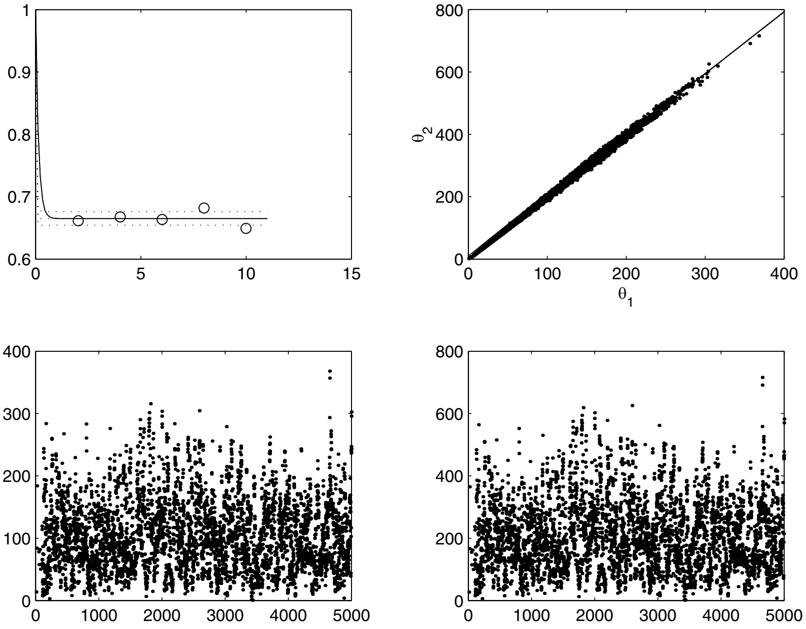


Fig. 6 4 figures on the left: results by DR. 4 figures on the right: results by AM. Figures 1 and 3 in the upper row: the data, the fit and the 95% probability values of the model predictions as computed by DR and AM, respectively. Figures 2 and 4 in the upper row: the computed 2D MCMC chain by DR and AM, respectively. Lower Figures: the 1D parameter chains

Fig. 7 Results by DRAM. Top left: the data, the fit, the 95% probability values of the model predictions as computed by the MCMC chain. Top right: the computed 2D MCMC chain. Lower figures: the 1D parameter chains



somewhat better than MH here, the 'easy' way of producing second stage proposals by scaling does not lead to essential improvement in this situation.

One could expect that AM would find, possibly after some initial trials, a well calibrated proposal distribution. However, since the initial proposal is so poor, it can take a very long time for AM to start working. Figure 6 illustrates a typical case. The time it takes to start the adaptation may be long or short, making the success of AM, at best, uncertain and the resulting estimates unreliable.

The combination of AM and DR was employed as outlined before. The proposal obtained from the covariance of the fit was used at the first stage. For the second stage the proposal was scaled down by a factor of 0.1. This gives a dramatic improvement, see Fig. 7. The second DR stage is able to find acceptable proposals right from the beginning, the AM adaptation starts immediately. The acceptance rate and the mixing of the chain are clearly improved. The first and second stage acceptance rates were around 30% and 60% respectively.

In such a low dimensional situation, the chain typically is not sensitive with respect to the choice of the non-adaptation period n_0, once the adaptation has properly started. However, here we have to start with a poor initial proposal, which might not produce accepted points at the very beginning. Using, e.g., the value $n_0 = 100$ guaranteed a smooth start: the successful results reported above were obtained without a single exception in repeated simulations.

Updating the covariance does not bring any substantial increase of computational burden in this low dimensional example. So the CPU times of MH and AM were roughly equal. The CPU times of DR and DRAM were approximately twice of those of MH and AM. Again, DRAM uses the second stage proposal less frequently than DR, and the CPU time is consequently slightly decreased.

6.3 Example 3

In our final example we discuss a real, high-dimensional modelling situation: the algae population dynamics in the shallow, mesotrophic Lake Pyhäjärvi in southwest Finland. Phytoplankton is modelled with a non-linear dynamics which describes the succession of four different algae groups as a function of total phosphorus, total nitrogen, temperature, global irradiance, and crustacea zooplankton. The noise level in the data is high, and the model is loaded with several correlated parameters. More details on the biological background and modelling issues is given in Malve et al. Here our purpose is to use the model to demonstrate how MH, DR, AM and DRAM typically behave in such a challenging situation.

The growth and decay mechanisms are integrated into a minimal mass-balance equation system for the wet weight concentration of algae. Phytoplankton is divided into four groups (Diatoms, Chrysophycea, nitrogen fixing Cyanobacteria and several minor species summed together in the fourth group), the concentrations of which are denoted as $A_i, i = 1, \ldots, 4$ ([mg L^{-1}]). The model is given by the following system of ordinary differential equations

$$\frac{dA_i}{dt} = \left(\tilde{\mu}_i - \frac{\tilde{\sigma}_i}{h} - \frac{Q}{V} - p_i Z\right) A_i, \quad i = 1, 2, 3, 4; \qquad (20)$$

where the growth limiting factors and loss rates are given by

$$\begin{aligned} \tilde{\mu}_i &= \mu_i \theta_i^{T - T_{\text{ref}}} \frac{I}{K_{Ii} + I} \frac{P}{K_{Pi} + P} \frac{N}{K_{Ni} + N}, \\ \tilde{\sigma}_i &= \sigma_i \theta_\sigma^{T - T_{\text{ref}}}. \end{aligned} \qquad (21)$$

Here P and N denote the total amounts of phosphorus and nitrogen minus that included in the phytoplankton: $P = P_{\text{tot}} - \sum_{i=1}^{4} \alpha_i A_i$ and $N = N_{\text{tot}} - \sum_{i=1}^{4} \beta_i A_i$, where the constants α_i and β_i give the nutrition content of the corresponding phytoplankton species. The terms P_{tot}, N_{tot}, T, I and Z are treated as control variables, given by measurements. The notations and roles of the various variables are listed in Table 1.

Table 1 Notations and units for the algae model parameters, data and constants

Estimated parameters $i = 1, \ldots, 4$	
μ_i	maximum growth rate at 20°C [d^{-1}]
σ_i	maximum non-predatory loss rate at 20°C [md^{-1}]
θ_i, θ_σ	temperature coefficients for growth and non-predatory loss rate
K_{Ii}	global irradiance half-saturation coefficient [W m^{-2}]
K_{Pi}	phosphorus half-saturation coefficient [µg L^{-1}]
K_{Ni}	nitrogen half-saturation coefficient [µg L^{-1}]
p_i	zooplankton filtration rate [mgC L^{-1} d^{-1}]
Control variables	
P_{tot}	total phosphorus concentration [g L^{-1}]
N_{tot}	total nitrogen concentration [g L^{-1}]
Z	zooplankton herbivore (crustacea) carbon mass concentration [mgC L^{-1}]
T	temperature [°C],
Q	outflow [m^{3}s^{-1}]
I	global irradiance [W m^{-2}]
Known constants	
T_{ref}	the reference temperature (20°C)
α_i	phosphorus content of A_i
β_i	nitrogen content of A_i
V	volume of lake [m^3]
h	depth of lake [m]

Eight years of observations from the lake, collected between 1992 and 2000, were used for this study. The observational error is modelled using i.i.d. Gaussian distribution. To stabilize the residual variances and to guarantee the positivity of the simulated observations, a square root transformation of the concentrations is used. Separate error variances are estimated for each of the four algae groups. For the variances of the error terms of the model we use non-informative conjugate priors defined by the inverse gamma distribution. As an example, in Fig. 10, the algae observations for two of the years are shown, together with the fitted model and simulated observations.

The values of the control variables are plotted, for all the years measured, in Fig. 8.

Although the model is rather reduced, we still have eight parameters to be estimated for each of the four phytoplankton groups. After a minor simplification (a common temperature coefficient for all the non-predatory losses, θ_σ) and the addition of four unknown initial algae concentration values for each of the eight summers considered, we have total of 61 model parameters to be estimated.

For several of the parameters we use non-informative priors, with positivity constraints only. Truncated Gaussian distributions are adopted as priors for the initial algae

Fig. 8 Time series of observed variables used in the algae model: P_{tot}: total phosphorus concentration [g L^{-1}], N_{tot}: total nitrogen concentration [g L^{-1}], T: water temperature [°C], I: global irradiance [W m^{-2}], Z: grazing zooplankton biomass concentration [g L^{-1}], Q: outflow rate [m^{3}s^{-1}]

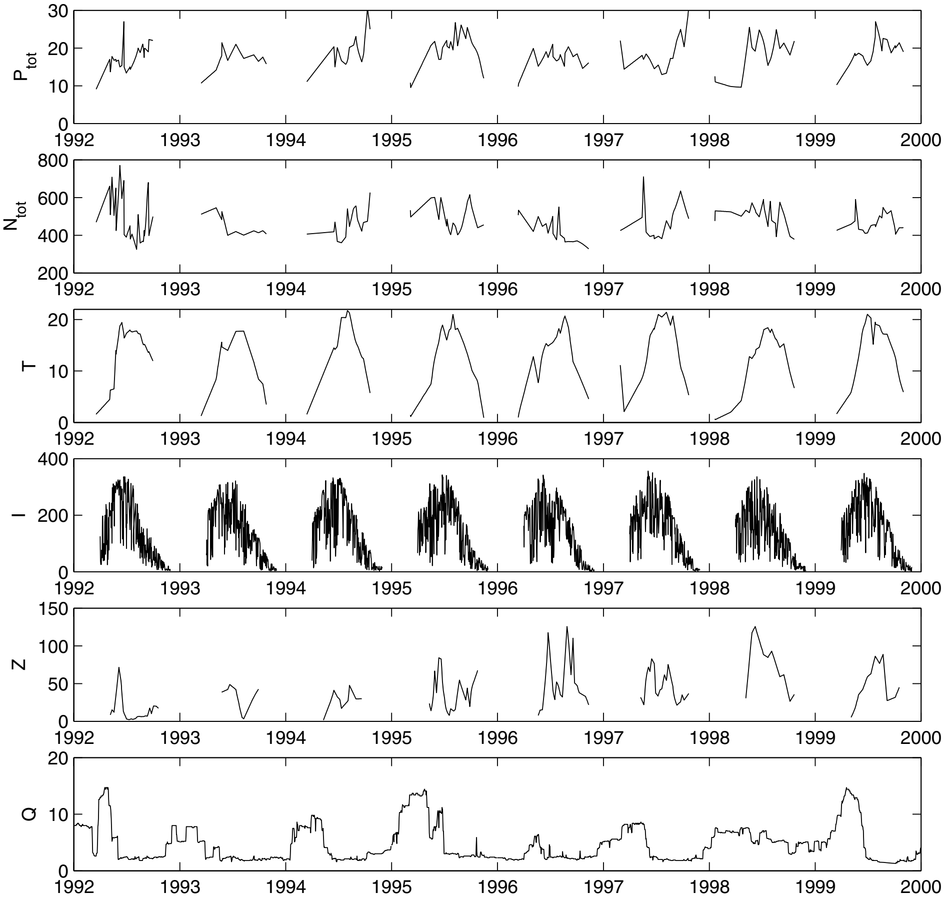


Fig. 9 Convergence of chain by MH, AM, DR and DRAM towards the target distribution, starting with the parameters values found in literature and using a crudely tuned proposal distribution. The pooled sum of residual squares is plotted against simulation time

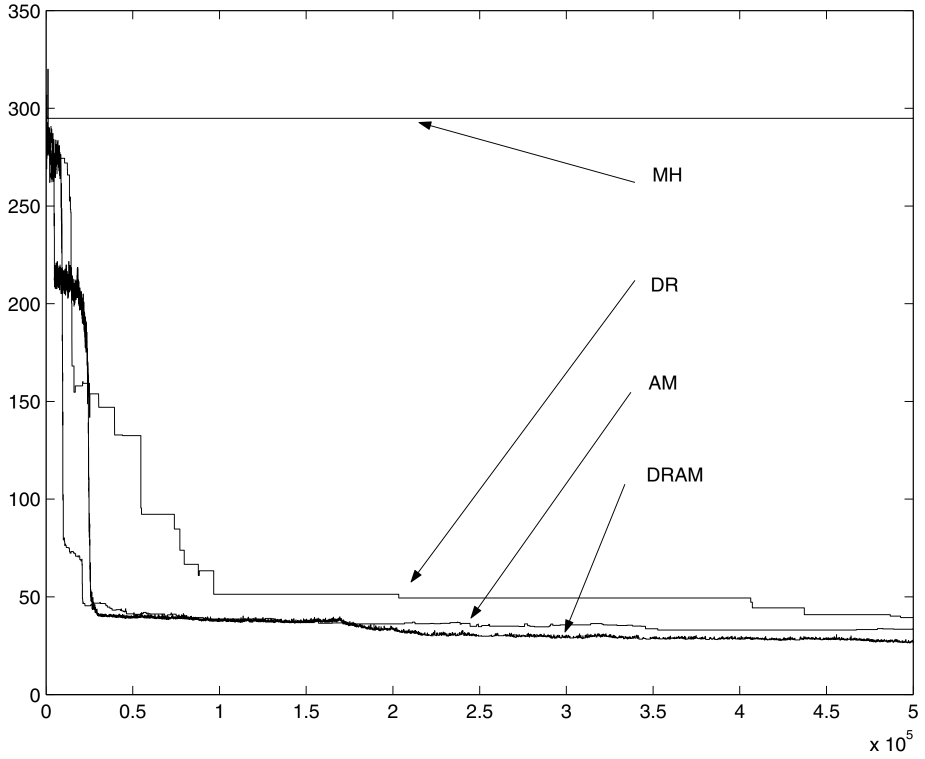


Table 2 Initial values, prior and posterior means and standard deviations of the estimated parameters of the phytoplankton model. A single dot (·) in the table indicates that a Uniform prior with positivity constraint is used for the corresponding parameter

Parameter	Prior mean/ initial value	Prior Std	Posterior mean	Posterior std
μ_1	1.07	.	0.0886	0.043
μ_2	1.00	.	0.0465	0.033
μ_3	1.31	.	0.329	0.089
μ_4	1.12	.	0.212	0.10
σ_1	0.015	.	0.0845	0.062
σ_2	0.052	.	0.137	0.077
σ_3	0.317	.	0.349	0.20
σ_4	0.036	.	0.0463	0.025
θ_1	1.01	0.08	1.14	0.051
θ_2	1.07	0.08	1.07	0.049
θ_3	1.16	0.08	1.16	0.060
θ_4	1.07	0.08	1.13	0.051
θ_σ	1.20	0.09	1.05	0.060
K_{I1}	100	100	61.9	48
K_{I2}	100	100	115	60
K_{I3}	100	100	16.4	11
K_{I4}	100	100	134	67
K_{P1}	2.5	10	10.4	5.3
K_{P2}	5	10	8.27	4.9
K_{P3}	10	20	5.50	3.2
K_{P4}	20	50	77.3	28
K_{N1}	10	20	14.5	11
K_{N2}	10	40	32.9	22
K_{N3}	10	20	21.0	13
K_{N4}	10	40	45.7	28
p_1	0.051	.	0.0438	0.036
p_2	0.044	.	0.0665	0.046
p_3	2.075	.	1.09	0.33
p_4	0.057	.	0.0802	0.033

concentrations, $A_i(t_0)$, with variances approximated by noise level of the measurements. Gaussian priors could also be used for the temperature coefficients for the growth and the non–predatory loss terms, e.g. as in Bowie et al. (1985). Table 2 gives the numerical values of the parameters used for constructing the priors, as well as the means and standard deviations computed from the posterior.

Initial values for the parameters are derived from the literature or obtained from discussions with experts in the field. It turned out that the initial guess was rather far from the MAP point of the posterior distribution. Several simulations were performed for testing the algorithms with various options for the starting point of the chain as well as for the initial proposal. We summarize below the experience, and present representative examples of the runs.

It turns out that it is possible to sample rather effectively the posterior with all the methods—plain MH, DR, AM and DRAM—using Gaussian proposals—once the chain has converged to the target distribution, and a suitable shape and orientation for the proposal is first found by pilot runs. But crucial differences are encountered in how the algorithms are capable to converge to the target and find an effective proposal without pilot runs. When using MH, a fixed proposal naturally has to be selected by the user. In the present high dimensional situation, this requires exhaustive care and off-line tuning. In the absence of a successfully selected proposal, the acceptance rate typically remains extremely low, as demonstrated in the example runs below. One possible strategy that could be adopted here is to estimate the parameters by least squares fitting and use the approximated covariance matrix of the parameters in the proposal distribution. But standard optimization routines do not easily converge in the present situation, often resulting in a singular covariance matrix leading, again, to ineffective sampling.

The DR algorithm generally performs better here, and is less sensitive with respect to non-optimally selected initial proposals.

The AM algorithm also performs reasonably well. Two types of problems may be encountered, however. As in the previous examples, if the variance of the initial proposal is too large, only few points are accepted at the beginning, and the adaptation has difficulties to get started. On the other hand, the acceptance rate may start decreasing during the adaptation. The reason is the basically non-Gaussian, slightly 'banana-shaped', correlated character of the posterior. With AM, we try to explore it using a single Gaussian proposal. This may lead to a proposal with too large variances and, consequently, small acceptance rates. As mentioned above, all the methods work rather well after convergence, the problems only appear during the warm-up phase.

The DRAM algorithm turns out to be the most reliable method. That is, it guarantees the mixing of the chain even with moderately ill-posed initial parameter values and proposal covariances. Basically, the success is due to the use of the proposal with small variances in situations where AM alone would use a proposal with too large variance.

Let us then present runs that demonstrate the convergence properties of the methods tested. We run a chain of length 500,000 using each of the four methods. The initial state of the chain is fixed using the parameter values given by the literature The proposal distribution (the initial proposal for the adaptive methods) is also fixed to be the same for all methods. A non-correlated Gaussian distribution is employed, with only a crude 'tuning' in the sense that the variances are given as the variances of the Gaussian prior distributions, when available. For parameters with positivity prior only (e.g., with a large upper bound for the flat prior), the variances are chosen so as to give standard deviations of the same order of magnitude as the prior guesses of the respective parameters. In DR and DRAM we use a second stage proposal obtained by scaling down the proposal of the first stage by the factor $\gamma = 0.01$. For AM and DRAM the value $n_0 = 200$ was used as the non-adaptation period, a larger value might have been used as well.

In order to exhibit the convergence properties of the sampling strategies, we plot the residual sum-of-squares obtained by the various methods. Separate sum-of-squares for the four model variables were employed, with the different unknown error variances sampled from non-informative conjugate priors. For clarity, we only present the pooled sums of the residuals.

Figure 9 gives a representative example for each different method. For the MH method the proposal is clearly too ill posed, the chain does not move at all. Due to the smaller variance of the second stage proposal, DR is able to make progress, although the acceptance rate remains very low. The

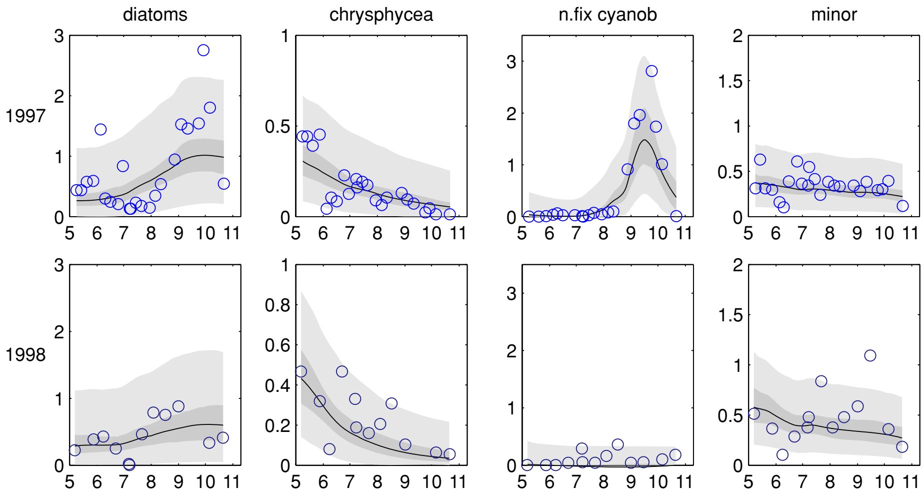


Fig. 10 Plots of the fitted algae model together with the uncertainties for two years 1997 and 1998. Circles (o) present the observed algae wet biomass concentrations [mg L^{-1}]. The solid lines show the median fits. Darker areas correspond to the 95% posterior limits of the model uncertainty, while the lighter areas present the same uncertainty level in predicting new observations. The horizontal axis give the months of the year

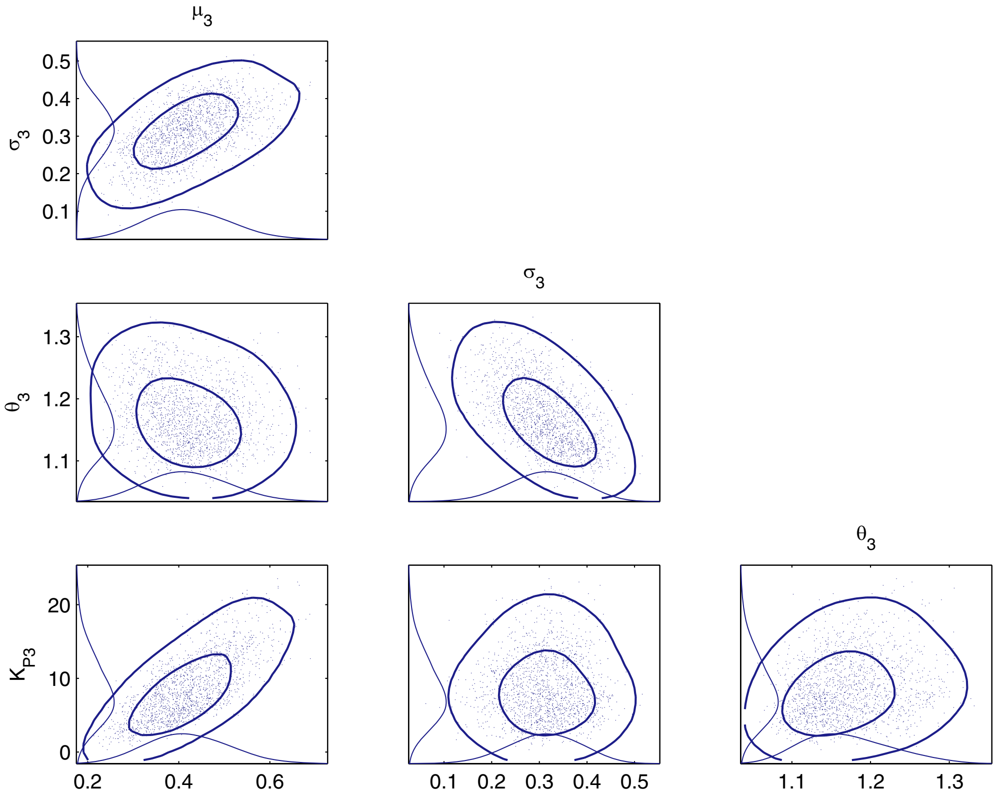


Fig. 11 Two dimensional marginal posterior distributions for parameters μ_3, σ_3, θ_3, and K_{P3} for the Cyanobacteria (see Table 1). The dots give the points of the MCMC chain from which the distribution contour lines (for the 50% and 95% regions of the distribution) are calculated using kernel density estimation. The distributions drawn along the axis are the corresponding one dimensional marginal densities

same is true here for AM: the initial proposal has too large variances, an even longer chain would be needed before the adaptation would find an effective scale for the proposal. DRAM clearly is the only method whose acceptance rate remains reasonable high, and which shows a monotone convergence towards lower values of the sum-of-squares. One should note however, that in this example the convergence is slow even with DRAM—the target distribution is typically reached only with a chain whose length is a few million samples.

After convergence, the parameters are, somewhat surprisingly, rather well identified and not too correlated. The situation can certainly be attributed to the a priori bounds available for several of the parameters. Figure 11 gives examples of two dimensional marginal posteriors, obtained by DRAM.

7 Conclusions

We show how two ways of modifying the standard MH sampler can be successfully combined. The first modification, AM, aims at adapting the proposal distribution based on the past history of the chain. The second modification, DR, aims at improving the efficiency of the resulting MCMC estimators. While AM allows for "global" adaptation, based on all the previously accepted proposals, DR may allow for "local" adaptation, only based on rejected proposals within each time-step. While DR retains the Markovian property and reversibility, AM is neither Markovian nor reversible. There are different ways of combining AM and DR. We tried some basic but very effective ones, as the simulation results show. One could use AM only at the first stage and employ fixed MH proposals at higher stages of the delaying rejection process. The choice of the fixed MH proposals could be based on separate pilot runs. Or one might adapt the proposals at different stages separately, with the aim of attempting "global" moves at the first stage (update all coordinates at once) and "local" moves at higher stages (update single coordinates of groups of them). As an alternative, at different stages of the delaying rejection, different values of n_0 and s_d could be used. We plan to further investigate other ways of combining DR and AM: DRAM is basically AM with-in DR. AMDR could also be tried (that is DR with-in AM): a basic AM algorithm is run, with some values for the parameters n_0 and s_d at the first stage and, upon rejection, a second stage adaptive proposal is used, with different values of these parameters. Further variations can be investigated. The key feature is that, as pointed out by Green and Mira (2001), DR works better if the variance of the proposal is too big at first stages and down scaled at higher stages. On the other hand AM recovers well and starts adapting even if the variance of the initial proposal is too small (clearly if the variance is too big no proposals are accepted and adaptation is almost impossible to get started). Thus, a combination of the two, as in DRAM or other variations of it, clearly provides protection against both over and under calibrated proposals.

Acknowledgments We would like to thank Johanna Tamminen for helpful discussions during this work.

References

Andrieu C. and Robert C.P. 2001. Controlled MCMC. Preprint.

Andrieu C. and Moulines E. 2002. On the ergodicity properties of some adaptive MCMC algorithms. To appear in Annals of Applied Probability.

Atchade Y.F. and Rosenthal J.S. 2005. On adaptive Markov chain Monte carlo algorithms. Bernoulli 11(5): 815–282.

Bowie G.L., Mills W.B., et al. 1985. Rates, constants, and kinetic formulations in surface water modeling. Technical Report EPA/600/3-85/040, U.S. Environmental Agency, ORD, Athens, GA, ERL.

Gelman A.G., Roberts G.O., and Gilks W.R. 1996. Efficient Metropolis jumping rules. In: Bernardo J.M., Berger J.O., David A.F., and Smith A.F.M. (Eds.), Bayesian Statistics V. Oxford University Press, pp. 599–608.

Green, P.J. and Mira, A. 2001 Delayed rejection in reversible jump Metropolis-Hastings. Biometrika 88: 1035–1053.

Haario H., Kalachev L., Lehtonen J., and Salmi T. 1999. Asymptotic analysis of chemical reactions. Chem. Eng. Sci. 54: 1131–1143.

Haario H., Saksman E., and Tamminen J. 1999. Adaptive proposal distribution for random walk Metropolis algorithm. Comp. Stat. 14: 375–395.

Haario H., Saksman E., and Tamminen J. 2001. An adaptive Metropolis algorithm. Bernoulli 7: 223–242.

Haario H., Saksman E., and Tamminen J. 2005. Componentwise adaptation for high dimensional MCMC. Computational Statistics 20(2): 265–274.

Malve O., Laine M., Haario H., Kirkkala T., and Sarvala J. Bayesian modeling of algae mass occurrences—using adaptive MCMC methods with a lake water quality model. To appear in Environmental Modelling and Software, 2006.

Mira A. 2001. On Metropolis-Hastings algorithms with delayed rejection. Metron, Vol. LIX, (3–4): 231–241.

Mira A. 2002. Ordering and improving the performance of Monte Carlo Markov Chains. Statistical Science 16: 340–350.

Peskun P.H. 1973. Optimum Monte Carlo sampling using markov chains. Biometrika 60: 607–612.

Sokal A.D. 1998. Monte carlo methods in statistical mechanics: Foundations and new algorithms. Cours de Troisième Cycle de la Physique en Suisse Romande. Lausanne.

Tierney L. 1994. Markov chains for exploring posterior distributions. Annals of Statistics 22: 1701–1762.

Tierney L. 1998. A note on Metropolis-Hastings kernels for general state spaces. Annals of Applied Probability 8: 1–9.

Tierney L. and Mira A. 1999. Some adaptive Monte Carlo methods for bayesian inference. Statistics in Medicine 18:2507–2515.