Bernoulli **11**(5), 2005, 815–828

On adaptive Markov chain Monte Carlo algorithms

YVES F. ATCHADÉ[1] and JEFFREY S. ROSENTHAL[2]

[1]*Department of Mathematics and Statistics, University of Ottawa, 585 King Edward St., Ottawa, ON, Canada K1N 6N5. E-mail: yatchade@uottawa.ca*
[2]*Department of Statistics, University of Toronto, 100 St. George Street, Room 6018, Toronto, ON, Canada M5S 3G3. E-mail: jeff@math.toronto.edu*

We look at adaptive Markov chain Monte Carlo algorithms that generate stochastic processes based on sequences of transition kernels, where each transition kernel is allowed to depend on the history of the process. We show under certain conditions that the stochastic process generated is ergodic, with appropriate stationary distribution. We use this result to analyse an adaptive version of the random walk Metropolis algorithm where the scale parameter σ is sequentially adapted using a Robbins–Monro type algorithm in order to find the optimal scale parameter σ_{opt}. We close with a simulation example.

Keywords: adaptive Markov chain Monte Carlo; Metropolis algorithm; mixingales; parameter tuning; Robbins–Monro algorithm

1. Introduction

Markov chain Monte Carlo (MCMC) methods have become an important numerical tool in statistics (see Gilks *et al*. 1996; Liu 2001). They usually require various parameters (e.g. proposal scalings) to be appropriately tuned for the algorithm to converge reasonably well. In this paper, we develop and analyse adaptive MCMC algorithms where these parameter tunings can be handled automatically.

We consider Monte Carlo algorithms based on random processes (which we shall call adaptive Markov chains) where the entire past of the process is used to make the next move in the algorithm. The set-up is a generalization of Haario *et al*. (2001). We prove two ergodicity results for such algorithms (Theorems 3.1 and 3.2). The rate of convergence obtained in Theorem 3.1 tends to indicate that these algorithms converge at a much slower rate. Nevertheless, their adaptability is an important attractive feature.

We apply these results to prove the convergence of a new adaptive random walk Metropolis (adaptive RWM) algorithm (Algorithm 4.1) with proposal kernel $q_\sigma(x, y)$, the density of the d-dimensional multivariate normal distribution $N(x, \sigma^2 I_d)$. It is well known that an effective implementation of this algorithm requires a good choice of the parameter σ^2; this choice depends on the density π. Some theoretical and empirical results (Roberts *et al*. 1997; Roberts and Rosenthal 2001) have shown that in high-dimensional spaces, under various regularity conditions, it is optimal to choose σ^2 such that the asymptotic acceptance

1350–7265

rate of the algorithm is approximately $\bar{\tau} = 0.234$. However, much trial and error may be required to find such a value for σ^2. In Section 4, we propose an adaptive version of the RWM algorithm which sequentially adapts σ^2 so as to reach the optimal acceptance rate $\bar{\tau}$. Our adaptive algorithm is based on a stochastic approximation algorithm.

A number of interesting ideas about adaptive MCMC methodology have recently been introduced. Gilks *et al*. (1998) have shown that the transition kernel used in an MCMC algorithm can be updated (without damaging the ergodicity of the algorithm) at regeneration times. The problem with this approach is that regeneration times for Markov chains are difficult to identify, particularly in high-dimensional spaces. Haario *et al*. (2001) have proposed an adaptive version of the RWM where the covariance matrix of the proposal kernel is sequentially updated. A recent paper highly comparable to this work is Andrieu and Moulines (2003). These authors have simultaneously and independently developed convergence results for adaptive MCMC algorithms. Although there is not much overlap between the two papers, both have similar assumptions.

Throughout this paper, π represents the probability measure of interest defined on some measurable space $(\mathcal{X}, \mathcal{F})$. In Section 2 we provide an example of an adaptive algorithm (Algorithm 2.1) that fails to converge. A general analysis for adaptive MCMC is developed in Section 3. The main results are Theorems 3.1 and 3.2. In Section 4 (Algorithm 4.1) we introduce a new adaptive RWM algorithm that can iteratively find the optimal scale parameter (Theorem 4.1). Simulation results are presented in Section 5.

2. Cautionary examples

We begin with a simple example due to G.O. Roberts (personal communication), where an intuitively reasonable adaptive rule fails to give the expected asymptotic distribution.

Take $\mathcal{X} = \{1, 3, 4\}$, and let π be the uniform distribution on $\mathcal{X}$. For $i = 1, 2$, and $x \in \mathcal{X}$, let $Q_i(x, \cdot)$ be the uniform distribution on $\{x - i, x + i\}$ (when $x \notin \mathcal{X}$, $Q(x, x) = 1$) and $R_i(x, \cdot) = (1 - \beta)Q_i(x, \cdot) + \beta\pi(\cdot)$, for some fixed $\beta \in [0, 1]$. Consider the following adaptive Metropolis algorithm.

Algorithm 2.1.

(i) *Start the algorithm at* $X_0 = x_0 \in \mathcal{X}$.
(ii) *Suppose that at some time* n, $X_n = x$. *If* $n = 0$, *sample* $Y_{n+1} \sim R_2(x, \cdot)$. *Otherwise:*
  (a) *if the last move was a rejection, sample* $Y_{n+1} \sim R_1(x, \cdot)$;
  (b) *if the last move was an acceptance, sample* $Y_{n+1} \sim R_2(x, \cdot)$;
  (c) *if* $Y_{n+1} \in \mathcal{X}$, *'accept'* Y_{n+1} *and set* $X_{n+1} = Y_{n+1}$; *otherwise 'reject'* Y_{n+1} *and set* $X_{n+1} = x$.

The strategy used in this algorithm is quite intuitive. Large step moves (from R_2) are proposed to help increase the mixing rate of the chain. But these moves are more likely to be rejected, and when this happens, the algorithm tries a smaller step move (from R_1). Each proposal R_i gives an ergodic Metropolis algorithm, but in fact Algorithm 2.1 fails to give the right asymptotic distribution.

To see why, let (X_n) be the stochastic process resulting from Algorithm 2.1 and define $Z_n := (X_n, X_{n-1}) \in \mathcal{X} \times \mathcal{X}$. It is easy to see that (Z_n) is a Markov chain. We can write the transition matrix of (Z_n). For $m, n \in \mathcal{X}$, note $\phi(m, n) = 1$ if $m = n$ and $\phi(m, n) = 2$ otherwise. Also define $\psi(m, n) = 1 - \beta$ if $m = n = 1$ or $(m \neq n$ and $n = 4)$ and $\psi(m, n) = (1 - \beta)/2$ otherwise. Then $P((m, n), (n, j))$, the probability that $Z_n = (n, j)$ given that $Z_{n-1} = (m, n)$, can be written:

$$P((m, n), (n, j)) = \begin{cases} (1 - \beta)Q_{\phi(m,n)}(n, j) + \beta\pi(j), & \text{if } j \neq n, \\ \beta\pi(n) + \psi(m, n), & \text{if } j = n. \end{cases}$$

It can be checked that P is irreducible and aperiodic. Since $\mathcal{X} \times \mathcal{X}$ is finite, P is ergodic. Let $\nu(i, j)$ be the invariant distribution for P. Then $\{X_n = 1\} = \{X_n = 1, X_{n-1} = 1\} \cup \{X_n = 1, X_{n-1} = 3\} \cup \{X_n = 1, X_{n-1} = 4\}$, which implies that

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} \mathbf{1}_{\{X_i=1\}} = \lim_{n\to\infty} \Pr(X_n = 1) = \nu(1, 1) + \nu(1, 3) + \nu(1, 4).$$

The computation of the matrix ν requires solution of the 9×9 linear equation $\nu P = \nu$. We do this numerically for different values of β. Table 1 summarizes the results. Clearly, for all $\beta \in [0, 1)$, $\lim_{n\to\infty}(1/n)\sum_{i=0}^{n-1} \mathbf{1}_{\{X_i=1\}} = \lim_{n\to\infty} \Pr(X_n = 1) > \frac{1}{3}$. As we shall see, this adaptive MCMC algorithm fails because the successive transition kernels in use fail to stabilize as the simulation goes along, a key requirement for an adaptive MCMC algorithm. For an interactive version of a related example, see Rosenthal (2004).

3. General ergodicity results

Assume that we have a starting transition kernel P_0 and an initial point $x_0 \in \mathcal{X}$. Consider the following generic adaptive MCMC algorithm:

Algorithm 3.1.

(i) *Suppose that at some time $n \geqslant 0$, we have $X_n = x$ and a transition kernel $P_{n,\tilde{X}_n}$ which is allowed to depend on the path $(X_0, \ldots, X_n) = \tilde{X}_n \in \mathcal{X}^{n+1}$ of the algorithm.*

(ii) *Sample $X_{n+1} \sim P_{n,\tilde{X}_n}(x, \cdot)$.*

(iii) *Use $\tilde{X}_{n+1} = (X_0, \ldots, X_{n+1})$ to build a new transition kernel $P_{n+1,\tilde{X}_{n+1}}$ to be used at time $n + 1$.*

We take $P_{0,\tilde{x}_0} = P_0$ as the starting transition kernel.

Table 1. $\lim_{n\to\infty} \Pr(X_n = 1)$ as a function of β in Algorithm 2.1.

$\lim \Pr(X_n = 1)$	0.9898	0.9088	0.5589	0.3517	0.3337	0.3334
β	0.001	0.01	0.1	0.5	0.9	0.99

To run Algorithm 3.1, we assume that we have at our disposal a family $\{P_{n,\tilde{x}_n}(x, A) : n \geqslant 0, \tilde{x}_n \in \mathcal{X}^{n+1}, x \in \mathcal{X}, A \in \mathcal{F}\}$ which is such that for $n \geqslant 0$, $\tilde{x}_n \in \mathcal{X}^{n+1}$, and $x \in \mathcal{X}$ fixed, $P_{n,\tilde{x}_n}(x, \cdot)$ is a probability measure on $(\mathcal{X}, \mathcal{F})$, and, for $A \in \mathcal{F}$, $P_{n,\tilde{x}_n}(x, A)$ is a measurable function from $(\mathcal{X}^{n+1} \times \mathcal{X}, \mathcal{F}^{n+1} \times \mathcal{F}$ to $[0, 1]$.

Let (X_n) be the random process generated by Algorithm 3.1 and P_μ its distribution on $(\mathcal{X}^\infty, \mathcal{F}^\infty)$ when $X_0 \sim \mu$. We shall write E_μ to denote the expectation with respect to P_μ. As usual, if $\mu = \delta_x$, the Dirac measure on x, we write E_x and P_x instead of E_μ and P_μ, respectively.

For a probability measure μ and a transition kernel P, the product μP defines a probability measure $\mu P(\cdot) := \int \mu(\mathrm{d}x) P(x, \cdot)$. And if f is a real-valued function on $\mathcal{X}$, the product Pf defines a function $Pf(x) := \int P(x, \mathrm{d}y) f(y)$. If P and Q are two transition kernels, the product PQ is also a transition kernel defined by $PQ(x, A) := \int P(x, \mathrm{d}y) Q(y, A)$. This allows us to define Q^n, the product of Q by itself n times, with the convention that $Q^0(x, A) = \mathbf{1}_A(x)$. Finally, for a probability measure μ and a positive function V, we define the V-norm of μ by $\|\mu\|_V := \sup_{|f| \leqslant V} |\mu(f)|$, where $\mu(f) := \int f(x) \mu(\mathrm{d}x)$.

We study the ergodicity of the stochastic process generated by Algorithm 3.1. We assume that for $n \geqslant 0$ and $\tilde{x}_n \in \mathcal{X}^{n+1}$, there exists a probability measure $\pi_{n,\tilde{x}_n}$ on $\mathcal{X}$ such that

$$\pi_{n,\tilde{x}_n} P_{n,\tilde{x}_n} = \pi_{n,\tilde{x}_n}, \tag{3.1}$$

and that the function $\tilde{x}_n \to \pi_{n,\tilde{x}_n}(A)$ is measurable for every $n \geqslant 0$ and $A \in \mathcal{F}$. In words, $\pi_{n,\tilde{x}_n}$ is an invariant distribution for $P_{n,\tilde{x}_n}$.

We require the following assumptions:

Assumption 3.1. *There exist a measurable function $V : \mathcal{X} \to [1, \infty)$ finite constants K_1, K_2, K_3 and sequences of real numbers (τ_n), (a_n), (R_n), with τ_n, $R_n \to 0$ as $n \to \infty$, such that:*

(i) *for $j \geqslant 1$, $n \geqslant 0$, $x \in \mathcal{X}$ and $\tilde{x}_n \in \mathcal{X}^{n+1}$,*

$$\|P^j_{n,\tilde{x}_n}(x, \cdot) - \pi_{n,\tilde{x}_n}(\cdot)\|_V \leqslant R_j V(x); \tag{3.2}$$

(ii) *for $x \in \mathcal{X}$, $\tilde{x}_n \in \mathcal{X}^{n+1}$, $\tilde{y}_k \in \mathcal{X}^{k+1}$, $\tilde{x}_{n+k} = (\tilde{x}_n, \tilde{y}_k)$,*

$$\|P_{n+k,\tilde{x}_{n+k}}(x, \cdot) - P_{n,\tilde{x}_n}(x, \cdot)\|_V \leqslant K_1 \tau_n a_k V(x), \tag{3.3}$$

and

$$\|\pi_{n+k,\tilde{x}_{n+k}} - \pi_{n,\tilde{x}_n}\|_V \leqslant K_2 \tau_n a_k; \tag{3.4}$$

(iii) *for $n \geqslant 0$ and $k \geqslant 1$,*

$$\int P_{n,\tilde{x}_n}(x_n, \mathrm{d}x_{n+1}) \cdots \int P_{n+k-1,\tilde{x}_{n+k-1}}(x_{n+k-1}, \mathrm{d}x_{n+k}) V(x_{n+k}) \leqslant K_3 V(x_n); \tag{3.5}$$

(iv)

$$\sup_{n,\tilde{x}_n} \pi_{n,\tilde{x}_n}(V) < \infty; \tag{3.6}$$

(v) *for finite constants c_1, c_2, defining $B(c_1, c_2, n) := \min_{1 \leqslant k \leqslant n}(c_1 \phi_k \tau_{n-k} + c_2 R_k)$, where $\phi_n = \sum_{k=1}^n a_k$, we have $B(c_1, c_2, n) = \mathcal{O}(1/n^\varepsilon)$ for some $\varepsilon > 0$.*

We would like to investigate the ergodicity of (X_n) under these assumptions. Henceforth, we write $\tilde{X}_n = (X_0, \ldots, X_n)$.

Theorem 3.1. *Assume that $X_0 = x_0 \in \mathcal{X}$. Under parts (i)–(iv) of Assumption 3.1, there are constants k_1, $k_2 < \infty$ such that for any measurable function $f : \mathcal{X} \to \mathbb{R}$ with $|f| \leqslant V$,*

$$|\mathrm{E}_{x_0}\Big(f(X_n) - \pi_{n,\tilde{X}_n}(f)\Big)| \leqslant B(k_1, k_2, n)V(x_0), \tag{3.7}$$

where V and $B(k_1, k_2, n)$ are also as in Assumption 3.1.

Theorem 3.2. *Under parts (i)–(v) of Assumption 3.1 and for any measurable function $f : \mathcal{X} \to \mathbb{R}$ with $|f| \leqslant V$, where V is also as in Assumption 3.1, we have*

$$\frac{1}{n}\sum_{i=0}^{n-1}\Big(f(X_i) - \pi_{i,\tilde{X}_i}(f)\Big) \to 0, \qquad \text{as } n \to \infty, \ P_{x_0}\text{-a.s.} \tag{3.8}$$

for any starting point $x_0 \in \mathcal{X}$.

Remark 3.1. (i) For most MCMC algorithms, one would have $\pi_{n,\tilde{x}_n} = \pi$, the invariant distribution of interest, and in this case Theorem 3.1 gives a bound on the rate of convergence of the distribution of X_n to π and Theorem 3.2 states a law of large numbers.

(ii) Assumption 3.1(i) requires a uniform-in-time (geometric or subgeometric) convergence rate of $P_{n,\tilde{x}_n}$ to $\pi_{n,\tilde{x}_n}$. This may be hard to check in practice. For example, to obtain a geometric convergence rate ($R_n = R\rho^n$ for some $0 < \rho < 1$) in Assumption 3.1(i), one possible way is to use quantitative bounds for Markov chains (e.g. Meyn and Tweedie 1994), which typically requires a drift condition of the form

$$P_{n,\tilde{x}_n}V(x) \leqslant \lambda V(x) + b\mathbf{1}_C(x), \tag{3.9}$$

for some $\lambda < 1$, $b < \infty$ and some small set C (for $P_{n,\tilde{x}_n}$) that do not depend on n, and a minorization condition

$$P_{n,\tilde{x}_n}(x, \cdot) \geqslant \varepsilon\nu(\cdot), \qquad x \in C, \tag{3.10}$$

where ε does not depend on n. It is now well known that many Markov chains satisfy a drift and a minorization condition. But the fact that the constants involved in these conditions do not depend on n makes them more difficult to establish in general. Nevertheless, there are some useful MCMC algorithms (such as the RWM algorithms) where Assumption 3.1 can be shown to hold. We return to this point in Section 4.

(iii) Assumption 3.1(ii) requires that as $n \to \infty$ the adaptation procedure results in more and more stable transition kernels. It can be shown that the example in Algorithm 2.1 satisfies all the assumptions above but Assumption 3.1(ii).

(iv) Theorem 3.1 tells us that the adaptive MCMC rate of convergence will be the worst of the rate of convergence of the (non-adaptive) transition kernels R_n and the rate of convergence of the adaptation process τ_n as in Assumption 3.1(ii). For example, taking $a_n = \mathcal{O}(n^{\lambda_2})$ for some $\lambda_2 > 0$, it is easily seen that if τ_n is geometric and R_n is geometric then $B(k_1, k_2, n) := \min_{1 \leqslant k \leqslant n}\big(c_1\phi_k\tau_{j-k} + c_2R_k\big)$ is also geometric. But for most adaptive

MCMC algorithms we typically have $\tau_n = \mathcal{O}(n^{-\lambda_1})$ for some $\lambda_1 > 0$, and assuming that $R_n = R\rho^n$ for some $0 < \rho < 1$, and taking $k = \alpha \log n$, $\alpha = -\lambda_1/\log \rho$, we obtain the polynomial rate $B(k_1, k_2, n) = \mathcal{O}(n^{-\lambda_1}(\log n)^{\lambda_2+1})$.

We conclude this section by proving these theorems. Our proofs are based on a version of the strong law of large numbers for mixingales and closely follow Haario *et al*. (2001). For an introduction to mixingales, see Hall and Heyde (1980).

Let $\mathcal{F}_n = \{\phi, \mathcal{S}\}$ be the trivial σ-algebra when $n < 0$, and $\mathcal{F}_n = \sigma(X_0, \ldots, X_n)$ be the σ-algebra generated by $(X_0, \ldots, X_n)$ when $n \geqslant 0$.

Lemma 3.1. *Suppose that parts (i)–(iv) of Assumption 3.1 hold. Then there are constants* $0 < k_1, k_2 < \infty$ *such that for any* $n \geqslant 0$, $j \geqslant 1$, *and any measurable function* f *with* $|f| \leqslant V$,

$$|\mathrm{E}_{x_0}(g_{n+j,\tilde{X}_{n+j}}(X_{n+j})]\mathcal{F}_n)| \leqslant B(k_1, k_2, j)V(X_n) \tag{3.11}$$

P_{x_0}*-a.s., where* $g_{k,\tilde{X}_k} = f - \pi_{k,\tilde{X}_k}(f)$.

Proof. We have $\pi_{k,\tilde{X}_k}(g_{k,\tilde{X}_k}) = 0$, P_{x_0}-a.s. Given $(X_0, X_1, \ldots, X_{n-1}) = \tilde{x}_{n-1}$ and $X_n = x$, we have

$$\mathrm{E}_{x_0}(g_{n,\tilde{X}_n}(X_{n+j})|\tilde{X}_{n-1} = \tilde{x}_{n-1}, X_n = x) = \sum_{k=1}^{j-1} \eta_k(\tilde{x}_{n-1}, x) + P^j_{n,\tilde{x}_n} g_{n,\tilde{x}_n}(x), \tag{3.12}$$

where

$$\eta_k(\tilde{x}_{n-1}, x) = \int P_{n,\tilde{x}_n}(x, \mathrm{d}x_{n+1}) \cdots$$

$$\int P_{n+k-1,\tilde{x}_{n+k-1}}(x_{n+k-1}, \mathrm{d}x_{n+k}) \int P^{j-k-1}_{n,\tilde{x}_n} g_{n,\tilde{x}_n}(x_{n+k+1})(P_{n+k,\tilde{x}_{n+k}}(x_{n+k}, \mathrm{d}x_{n+k+1})$$

$$- P_{n,\tilde{x}_n}(x_{n+k}, \mathrm{d}x_{n+k+1})).$$

Using Assumption 3.1(i), we can bound the second term of the left-hand side of (3.12) as follows:

$$|P^j_{n,\tilde{x}_n} g_{n,\tilde{x}_n}(x)| \leqslant R_j V(x). \tag{3.13}$$

From Assumption 3.1(ii) and using the fact that $\sup_{n,\tilde{x}_n} \pi(V) < \infty$, we have the following bounds for some finite constant r_0:

$$|\eta_k(\tilde{x}_{n-1}, x)| \leqslant r_0 \tau_n a_k \int P_{n,\tilde{x}_n}(x, \mathrm{d}x_{n+1}) \cdots \int P_{n+k-1,\tilde{x}_{n+k-1}}(x_{n+k-1}, \mathrm{d}x_{n+k}) V(x_{n+k}),$$

$$= r_0 \tau_n a_k \mathrm{E}_{x_0}\big(V(X_{n+k})|\tilde{X}_n = (\tilde{x}_{n-1}, x)\big). \tag{3.14}$$

Putting (3.13) and (3.14) together in (3.12), we obtain

$$|\mathrm{E}_{x_0}\Big(g_{n,\tilde{X}_n}(X_{n+j})|\mathcal{F}_n\Big)| \leqslant R_j V(X_n) + r_0\tau_n \sum_{k=1}^{j-1} a_k \mathrm{E}_{x_0}(V(X_{n+k})|\mathcal{F}_n). \tag{3.15}$$

Taking (3.4) into account leads to

$$|\mathrm{E}_{x_0}\Big(g_{n+j,\tilde{X}_{n+j}}(X_{n+j})|\mathcal{F}_n\Big)| \leqslant R_j V(X_n) + r_0\tau_n \sum_{k=1}^{j-1} a_k \mathrm{E}_{x_0}(V(X_{n+k})|\mathcal{F}_n) + K_2\tau_n a_j \tag{3.16}$$

$$\leqslant R_j V(X_n) + \max(r_0,\, K_2)\tau_n \sum_{k=1}^{j} a_k V(X_n)$$

$$\leqslant V(X_n)\big(r_3 R_j + r_2\tau_n\phi_j\big), \tag{3.17}$$

where in the last inequality we use Assumption 3.1(iii) and $\phi_j = \sum_{k=1}^{j} a_k$, $r_2 = \max(r_0,\, K_2)K_3$, $r_3 = K_3$ and K_3 is as defined in Assumption 3.1(iv).

Since the family $(\mathcal{F}_n)_{n=-\infty}^{\infty}$ is increasing, $\mathcal{F}_n \subseteq \mathcal{F}_{n+j-k}$ for $k = 1, \ldots, j$. Therefore,

$$\mathrm{E}_{x_0}\Big(g_{n+j,\tilde{X}_{n+j}}(X_{n+j})|\mathcal{F}_n\Big) = \mathrm{E}_{x_0}\Big[\mathrm{E}_{x_0}\Big(g_{n+j,\tilde{X}_{n+j}}(X_{n+j})|\mathcal{F}_{n+j-k}\Big)]\mathcal{F}_n\Big].$$

It follows that

$$|\mathrm{E}_{x_0}\Big(g_{n+j,\tilde{X}_{n+j}}(X_{n+j})|\mathcal{F}_n\Big)| \leqslant \mathrm{E}_{x_0}\Big[|\mathrm{E}_{x_0}\Big(g_{n+j,\tilde{X}_{n+j}}(X_{n+j})|\mathcal{F}_{n+j-k}\Big)|\,|\mathcal{F}_n\Big]. \tag{3.18}$$

Applying (3.17) to the right-hand side of (3.18) gives

$$|\mathrm{E}_{x_0}\Big(g_{n+j,\tilde{X}_{n+j}}(X_{n+j})|\mathcal{F}_n\Big)| \leqslant \min_{1\leqslant k\leqslant j}\big(r_2\tau_{n+j-k}\phi_k + r_3 R_k\big)\mathrm{E}_{x_0}(V(X_{n+j-k})|\mathcal{F}_n)$$

$$\leqslant V(X_n)B(k_1,\, k_2,\, j)$$

for some constants k_1, k_2. □

Proof of Theorem 3.2. Taking $n = 0$ in (3.11) of Lemma 3.1 gives, for $n \geqslant 1$,

$$|\mathrm{E}_{x_0}\Big(g_{n,\tilde{X}_n}(X_n)\Big)| \leqslant B(k_1,\, k_2,\, n)V(x_0). \tag{3.19}$$

Together with Assumption 3.1(v), this shows that

$$\mathrm{E}_{x_0}\Big(f(X_n) - \pi_{n,\tilde{X}_n}(f)\Big) \to 0, \qquad \text{as } n \to \infty. \tag{3.20}$$

Write $Y_n = f(X_n) - \pi_{n,\tilde{X}_n}(f) - \mathrm{E}_{x_0}(f(X_n) - \pi_{n,\tilde{X}_n}(f))$. Given Lemma 3.1, it can easily be shown that $(Y_n, \mathcal{F}_n)$ is an L^2-mixingale with mixingale sequences $c_n \equiv c$ constant and $\psi_n = B(k_1, k_2, n)$. It follows from Corollary 2.1 of Davidson and de Jong (1997) that

$$\frac{1}{n}\sum_{k=0}^{n-1} g_{k,\tilde{X}_k}(X_k) - \mathrm{E}_{x_0}\Big(g_{k,\tilde{X}_k}(X_k)\Big) \to 0, \qquad P_{x_0}\text{-a.s. as } n \to \infty. \tag{3.21}$$

Combining (3.20) and (3.21), we obtain that

$$\frac{1}{n}\sum_{k=0}^{n-1}\Big(f(X_k)-\pi_{k,\tilde{X}_k}(f)\Big)\to 0, \qquad P_{x_0}\text{-a.s. as } n\to\infty, \tag{3.22}$$

as desired. □

Proof of Theorem 3.1. Taking $n=0$ in (3.11) of Lemma 3.1, we obtain

$$|\mathrm{E}_{x_0}\Big(f(X_n)-\pi_{n,\tilde{X}_n}(f)\Big)|\leqslant B(k_1,\,k_2,\,n)V(x_0), \tag{3.23}$$

for all $|f|\leqslant V$, which is Theorem 3.1. □

4. Application to the random walk Metropolis algorithm

In this section, $\mathcal{X}$ is an open subset of $\mathbb{R}^d$, the d-dimensional Euclidean space equipped with its Borel subsets $\mathcal{B}^{\mathrm{d}}$. We let π be a positive continuous density with respect to Lebesgue measure on $\mathcal{X}$. We denote by $|\cdot|$ the Euclidean norm on $\mathcal{X}$. We consider the RWM algorithm with proposal density $q_\sigma(x,\,y)=N(x,\,\sigma^2 I_d)$. This algorithm generates a Markov chain (X_n) with invariant distribution π as follows. Given X_n, a new proposal $Y_{n+1}\sim N(X_n,\,\sigma^2 I_d)$ is made. We then either 'accept' the proposed value and set $X_{n+1}=Y_{n+1}$ with probability $\alpha(X_n,\,Y_{n+1})$, or we 'reject' it and set $X_{n+1}=X_n$ with probability $1-\alpha(X_n,\,Y_{n+1})$, where $\alpha(x,\,y)=\min(1,\,\pi(y)/\pi(x))$. This algorithm always has stationary distribution π. However, the choice of the scaling parameter σ^2 has a large effect on the algorithm's mixing time. Intuitively, if σ^2 is too small, the resulting algorithm will make very small moves, resulting in a poor mixing time. On the other hand, if σ^2 is too large, then large moves will usually be proposed, and these are likely to be rejected so the algorithm will again mix poorly. Here we propose an adaptive version of the RWM algorithm that can automatically find σ such that the asymptotic acceptance rate of the algorithm is approximately $\bar{\tau}=0.234$.

4.1. The adaptive RWM algorithm

Let P_σ be the transition kernel of the RWM algorithm with proposal $q_\sigma(x,\,y)$. Let

$$A(\sigma,\,x):=\int\alpha(x,\,y)q_\sigma(x,\,y)\mathrm{d}y \quad\text{and}\quad \tau(\sigma):=\int A(\sigma,\,x)\pi(x)\mathrm{d}x \tag{4.1}$$

be the acceptance rate at x and in stationarity, respectively. Our adaptive algorithm relies on stochastic approximation algorithms initiated by Robbins and Monro (1951). These are well-known recursive algorithms of the form $\theta_{n+1}=\theta_n+\gamma_n(h(\theta_n)+\varepsilon_{n+1})$, typically used to solve equations of the form $h(\theta)=0$ when the function h is unknown (understood to mean 'hard to compute') but can be estimated with a noise (see Kushner and Yin 2003 and the references therein).

Fix $0<\varepsilon_1<A_1$. Define $\Delta=\{\sigma:\varepsilon_1\leqslant\sigma\leqslant A_1\}$. We shall assume that there is a unique $\sigma_{\mathrm{opt}}\in\Delta$ such that $\tau(\sigma_{\mathrm{opt}})=\bar{\tau}$. Next, we need a way to contain the algorithm inside Δ. We

define the function $p(\sigma)$ such that $p(\sigma) = \sigma$ if $\sigma \in \Delta$, $p(\sigma) = \varepsilon_1$ if $\sigma < \varepsilon_1$ and $p(\sigma) = A_1$ if $\sigma > A_1$.

Let (γ_n) be a positive sequence of real numbers. Our adaptive algorithm is thus as follows:

Algorithm 4.1.

(i) *Start the algorithm at some point* $x_0 \in \mathcal{X}$ *and* $\sigma_0 \in \Delta$.
(ii) *Suppose that at time* $n \geqslant 0$, *we have* $X_n \in \mathcal{X}$ *and* $\sigma_n \in \Delta$.
  (a) *Generate* $Y_{n+1} \sim Q_{\sigma_n}(x, \cdot)$ *and* $U \sim \mathcal{U}(0, 1)$.
  (b) If $U \leqslant \alpha(X_n, Y_{n+1})$, *then set* $X_{n+1} = Y_{n+1}$. *Otherwise, set* $X_{n+1} = X_n$.
  (c) *Compute*

$$\sigma_{n+1} = p(\sigma_n + \gamma_n(\alpha(X_n, Y_{n+1}) - \bar{\tau})). \tag{4.2}$$

Remark 4.1. (i) The acceptance rate is monitored by means of (4.2). The algorithm lowers the scale parameter σ_n when the acceptance rate is too small and increases σ_n when the acceptance rate is too high. Instead of updating σ_n at each iteration, a more robust algorithm could be obtained by updating σ_n every w iterations. We tried various value of w in our simulations and did not find much improvement with $w > 1$. But this may not be the case with more complex examples.

(ii) The projection function p is used to keep σ_n inside Δ and avoid the degeneracy of the algorithm. But the drawback (as with every stochastic approximation algorithm with re-projection on a fixed compact set) is that the optimal value cannot be found if the compact set Δ is misspecified. In most MCMC contexts though, if necessary, one may run a pilot simulation at $\sigma = \varepsilon_1$ and $\sigma = A_1$ to validate these values. Another approach dating back to Chen and Zhu (1986) has been advocated and developed by Andrieu *et al*. (2002) that avoids this problem by using re-projections on a family of nested compact sets. But in MCMC settings this approach is not necessarily better.

(iii) A better way to scale the RWM algorithm is to use the proposal distribution $N(x, \sigma\Sigma)$ with $\sigma = \sigma_{\text{opt}}$ and $\Sigma = \Sigma_\pi$, the covariance matrix of the distribution π. Since $(\sigma_{\text{opt}}, \Sigma_\pi)$ is not known, an adaptive algorithm can also be applied. We do not pursue this here. See Atchadé (2005), Andrieu and Moulines (2003) and Haario *et al*. (2001).

4.2. Ergodicity of the algorithm

We assume that π is superexponential with asymptotically regular contours (Jarner and Hansen 2000) and that the function $\tau(\sigma)$ is decreasing on Δ. More precisely:

Assumption 4.1.

(i) *We assume that* π *is positive with continuous first derivative such that*

$$\lim_{|x| \to \infty} n(x) \cdot \nabla \log \pi(x) = -\infty,$$

and

$$\limsup_{|x|\to\infty} n(x)\cdot m(x) < 0,$$

where ∇ *is the gradient operator,* $n(x) = x/|x|$ *and* $m(x) = \nabla\pi(\mathrm{x})/|\nabla\pi(x)|$.

(ii) *We assume that there exists* $\sigma_{\mathrm{opt}} \in \Delta$ *such that* $\tau(\sigma_{\mathrm{opt}}) = 0$ *and* $(\sigma - \sigma_{\mathrm{opt}})(\tau(\sigma) - \bar{\tau}) < 0$ *whenever* $\sigma \neq \sigma_{\mathrm{opt}}$.

(iii) (γ_n) *is a positive sequence of real numbers such that* $\gamma_n = \mathcal{O}(n^{-\lambda_1})$ *for some constant* $1/2 < \lambda_1 \leqslant 1$.

Under Assumption 4.1(i) it follows from Proposition 9 of Andrieu and Moulines (2003) that the family $(P_\sigma)_{\sigma\in\Delta}$ satisfies a uniform (in σ) minorization and drift condition: there exist $\varepsilon > 0$, $0 < \lambda < 1$, $b < \infty$, a compact non-empty set $C \subseteq \mathcal{X}$ and a non-trivial probability measure ν such that

$$\inf_{\sigma\in\Delta} P_\sigma(x, A) \geqslant \varepsilon\nu(A)\mathbf{1}_C(x), \qquad A \in \mathcal{B},\ x \in \mathcal{X}, \tag{4.3}$$

and

$$\sup_{\sigma\in\Delta} P_\sigma W(x) \leqslant \lambda W(x) + b\mathbf{1}_C(x), \qquad x \in \mathcal{X}, \tag{4.4}$$

where $W(x) = c\pi(x)^{1/2}$, with c such that $W(x) \geqslant 1$. Moreover, there exists a constant $K_1 < \infty$ such that

$$\sup_{|f|\leqslant W^{1/2}} |P_{\sigma_2} f(x) - P_{\sigma_1} f(x)| \leqslant K_1 W^{1/2}(x)|\sigma_2 - \sigma_1|. \tag{4.5}$$

Theorem 4.1. *Let* (X_n) *be the stochastic process generated by Algorithm 4.1. Suppose Assumption 4.1 holds and take* $V = W^{1/2}$. *Then:*

(i) *there is a finite constant* k *such that for* $n \geqslant 2$,

$$\|\mathcal{L}_{x_0}(X_n) - \pi\|_V \leqslant kn^{-\lambda_1}(\log n)^2, \tag{4.6}$$

where $\mathcal{L}_{x_0}(X_n)$ *is the distribution of* X_n *given that* $X_0 = x_0$;

(ii) *for any measurable function* $f : \mathcal{X} \to \mathbb{R}$ *with* $|f| \leqslant V$,

$$\frac{1}{n}\sum_{i=0}^{n-1} f(X_i) \to \pi(f)\, P_{x_0}\text{-a.s.}, \tag{4.7}$$

(iii) $\sigma_n \to \sigma_{\mathrm{opt}}$ *as* $n \to \infty$, P_{x_0}-*a.s.*

Proof. (i) and (ii) The minorization condition (4.3) and the drift condition (4.4) imply Assumptions 3.1(iii) (with $V = W^{1/2}$), 3.1(iv) and 3.1(i). Assumption 3.1(i) actually follows from the computational bound for Markov chains in V-norm as in Meyn and Tweedie (1994). The sequence (σ_n) almost surely satisfies $|\sigma_{n+k} - \sigma_n| \leqslant Ak/n$ for some finite constant A

which, together with (4.5), implies Assumption 3.1(ii). Therefore (i) is Theorem 3.1 and (ii) is Theorem 3.2.

(iii) We have the recursion $\sigma_{k+1} = p(\sigma_k - \gamma_k(\alpha(X_n, Y_{n+1}) - \bar{\tau}))$. We let $\mathcal{F}_n$ be the σ-algebra generated by $(\sigma_0, X_0, \ldots, \sigma_n, X_n)$, $U_n = (\sigma_n - \sigma_{\mathrm{opt}})^2$ and $V_n = -(\sigma_n - \sigma_{\mathrm{opt}})(\tau(\sigma_n) - \bar{\tau})$. We recall the definition of $A(\sigma, x) = \int \alpha(x, y) q_\sigma(x, y) \mathrm{d}y$ and $\tau(\sigma) = \int A(\sigma, x)\pi(\mathrm{d}x)$. It can easily be shown that

$$\mathrm{E}_{x_0}(U_{n+1}|\mathcal{F}_n) \leqslant U_{\mathrm{n}} - 2\gamma_n V_n + \gamma_n^2 + 2\gamma_n \varepsilon_n \tag{4.8}$$

where $\varepsilon_n = (\sigma_n - \sigma_{\mathrm{opt}})(A(\sigma_n, X_n) - \tau(\sigma_n))$. We claim that $\sum \gamma_n \varepsilon_n$ converges almost surely to a finite random variable.

We are then able to apply the Robbins–Siegmund theorem (see Duflo 1997, Theorem 1.3.12) to obtain that $U_n = (\sigma_n - \sigma_{\mathrm{opt}})^2$ converges (almost surely) to some finite random variable and $\sum \gamma_n V_n < \infty$ (almost surely). That is, σ_n converges almost surely to some finite random variable $\sigma_\infty \in \Delta$. Now it is clear that the function τ is continuous so that $\tau(\sigma_n) \to \tau(\sigma_\infty)$ (almost surely). Suppose that $\sigma_\infty \neq \sigma_{\mathrm{opt}}$. Then $V_n \to -(\sigma_\infty - \sigma_{\mathrm{opt}})(\tau(\sigma_\infty) - \bar{\tau}) > 0$, which contradicts $\sum \gamma_n V_n < \infty$ since $\sum \gamma_n = \infty$. Hence $\sigma_\infty = \sigma_{\mathrm{opt}}$.

The proof of the above claim is similar to the proof of Lemma 3.1. But first observe that we can find $k_1, k_2 < \infty$ such that $|A(\sigma_2, x) - A(\sigma_1, x)| \leqslant k_1|\sigma_2 - \sigma_1|V(x)$, and $|\tau(\sigma_2) - \tau(\sigma_1)| \leqslant k_2|\sigma_2 - \sigma_1|$, for every $\sigma_1, \sigma_2 \in \Delta$. The proof follows from Proposition 9 of Andrieu and Moulines (2003). It can also be shown directly using the mean value theorem applied to $A(\sigma, x)$, x fixed.

For $n \geqslant 0$ and $k \geqslant 1$, we have

$$\begin{aligned}\varepsilon_{n+k} &= (\sigma_{n+k} - \sigma_n)(A(\sigma_{n+k}, X_{n+k}) - \tau(\sigma_{n+k})) + (\sigma_n - \sigma_{\mathrm{opt}})(A(\sigma_{n+k}, X_{n+k}) - A(\sigma_n, X_{n+k}))\\ &\quad + (\sigma_n - \sigma_{\mathrm{opt}})(A(\sigma_n, X_{n+k}) - \tau(\sigma_n)) + (\sigma_n - \sigma_{\mathrm{opt}})(\tau(\sigma_n) - \tau(\sigma_{n+k})).\end{aligned}$$

Given the recursion on (σ_n) and the fact that the functions A and τ are Lipschitz (for x fixed), non-negative and bounded from above by 1, we can find $C_1 < \infty$ such that

$$|\mathrm{E}_{x_0}(\varepsilon_{n+k}|\mathcal{F}_n)| \leqslant 3C_1 k\gamma_n V(X_n) + |\sigma_n - \sigma_{\mathrm{opt}}|\,|\mathrm{E}_{x_0}(A(\sigma_n, X_{n+k}) - \tau(\sigma_n)|\mathcal{F}_n)|. \tag{4.9}$$

Now we can apply (3.17) to $|\mathrm{E}_{x_0}(A(\sigma_n, X_{n+k}) - \tau(\sigma_n)|\mathcal{F}_n)|$ to obtain, for some constants C_2, $C_3 < \infty$ and $\rho < 1$,

$$|\mathrm{E}_{x_0}(\varepsilon_{n+k}|\mathcal{F}_n)| \leqslant V(X_n)\big(C_3\rho^k + C_2 k^2 \gamma_n\big). \tag{4.10}$$

At this point the same σ-algebra trick as used in the proof of Lemma 3.1 can be applied to obtain

$$|\mathrm{E}_{x_0}(\varepsilon_{n+k}|\mathcal{F}_n)| \leqslant C_4 \log(k)^2 \gamma_k V(X_n). \tag{4.11}$$

It follows that $(\gamma_n(\varepsilon_n - \mathrm{E}_{x_0}(\varepsilon_n)), \mathcal{F}_n)$ is a mixingale with mixingale sequence $c_n \propto \gamma_n$ and $\psi_n \propto \log(n)^2\gamma_n$. Theorem 2.7 of Hall and Heyde (1980) then asserts that for such a mixingale, $\sum \gamma_n(\varepsilon_n - \mathrm{E}_{x_0}(\varepsilon_n))$ converges almost surely to a finite random variable. The claim is thus proved since $\sum \gamma_n \mathrm{E}_{x_0}(\varepsilon_n)$ is a convergent series which follows from (4.11). □

5. Simulation example

In this section, we conduct a simulation study to illustrate the results obtained in Section 4. We take π to be the d-dimensional standard normal distribution for $d = 10$ and 50. We use $Q_\sigma(x, \cdot) \sim N(x, \sigma^2 I_d)$, and as a function of interest we take $f(x) = x_1$, the first coordinate of x. For each simulation, we start with $\sigma_0 = 10$, $a = 0.0001$, $A = 1000$, and each chain is run for 250 000 iterations. (In fact, the initial value σ_0 is not important; in any case the values of σ_n become very low before converging upwards to σ_{opt}.) With all the adaptive algorithms, we use $\gamma_n = \sigma_0/n$.

Figure 1 shows the autocorrelation functions of the adaptive RWM (ARWM) algorithm (with $\bar{\tau} = 0.234$) and the (non-adaptive) RWM with optimal scaling σ_{opt}. The performance of the adaptive and the optimal non-adaptive algorithms is very similar in term of mixing time as measured by the autocorrelation functions. This shows that our adaptive algorithm achieves essentially the same mixing time as the optimally scaled algorithm, but without

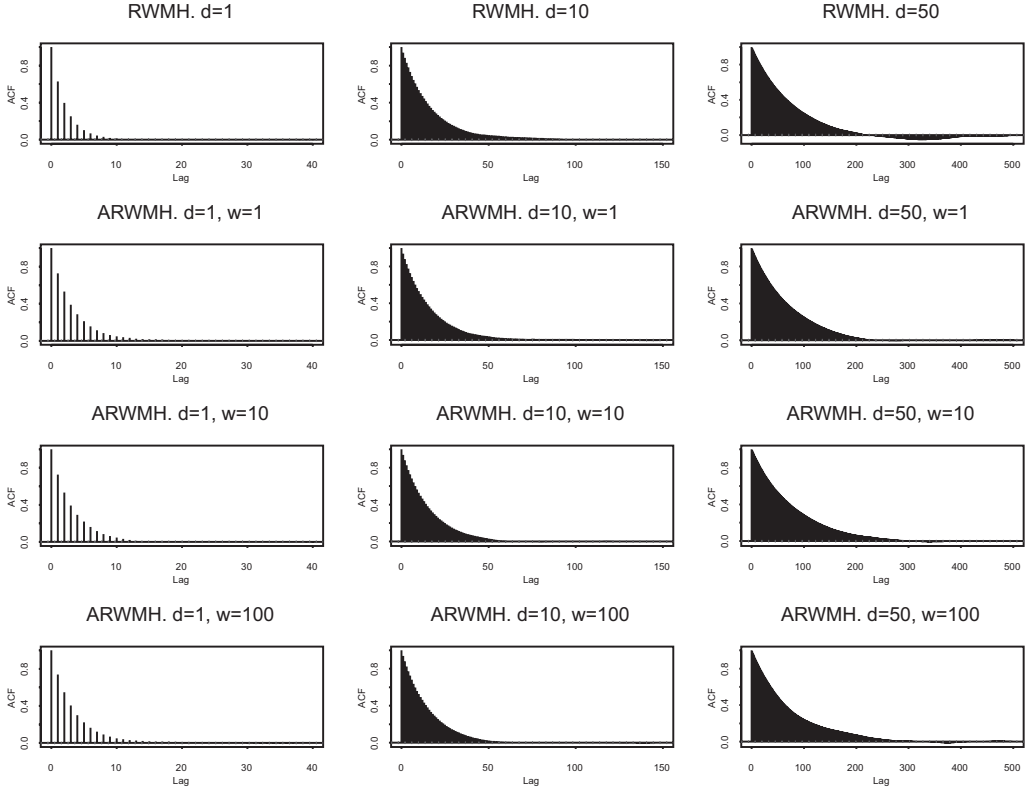

Figure 1. Autocorrelations of ergodic averages of the function $f(x) = x_1$. Target density $N(0, I_d)$, proposal density $N(x, \sigma^2 I_d)$.

requiring all the preliminary effort to manually tune the scaling parameter. For each value of d, we run the simulations with $w = 1$, 10 and 100, where w is the number of observations gathered before updating σ_n. The three values are quiet comparable.

Figure 2 shows the scale parameter process and the empirical acceptance rate obtained during the ARWM simulation for $w = 10$, and for a targeted acceptance rate of $\bar{\tau} = 0.234$. The empirical acceptance probability converges to 0.234, showing that we are indeed finding the optimal scaling parameter σ_{opt}. For large values of d, the value of σ_{opt} is consistent with the formula $2.38/\sqrt{d}$ (0.34 if $d = 50$, 0.75 if $d = 10$) given by Roberts *et al*. (1997).

Acknowledgement

This research was supported in part by NSERC of Canada.

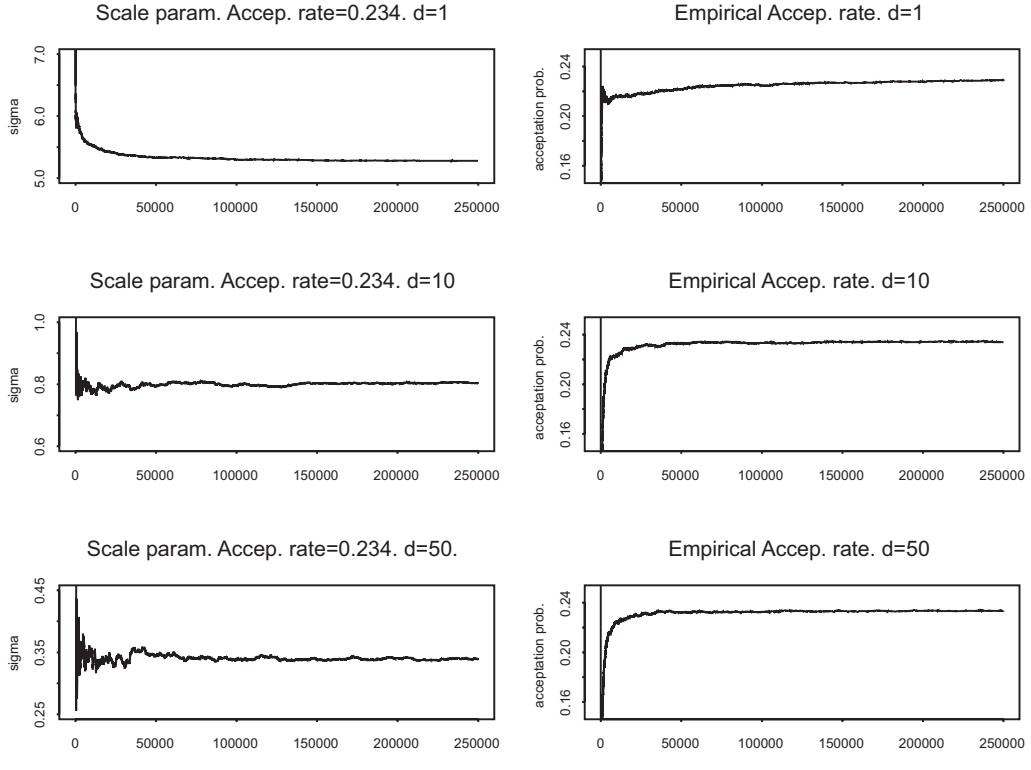

Figure 2. Scale parameter process and empirical acceptance probability for the ARWM with $w = 10$.

References

Andrieu, C. and Moulines, E. (2003) Ergodicity of some adaptive Markov Chain Monte Carlo algorithms. Technical report.

Andrieu, C., Moulines, E. and Priouret, P. (2002) Stability of stochastic approximation under verifiable conditions, *SIAM J. Control Optim*. To appear.

Atchadé, Y.F. (2005) An adaptive version for the Metropolis adjusted Langevin algorithm with a truncated drift. MCMC Preprint.

Chen, H. and Zhu, Y.-M. (1986) Stochastic approximation procedures with randomly varying truncations. *Sci. Sinica Ser. A*, **1**, 914–926.

Davidson, J. and de Jong, R. (1997) Strong laws of large numbers for dependent heteregeneous processes: a synthesis of recent and new results. *Econometric Rev.*, **16**, 251–279.

Duflo, M. (1997) *Random Iterative Models*. Berlin: Springer-Verlag.

Gilks, W.R., Richardson, S. and Spiegelhalter, D.J. (eds) (1996) *Markov Chain Monte Carlo in Practice*. London: Chapman & Hall.

Gilks, W.R., Roberts, G.O. and Sahu, S.K. (1998) Adaptive Markov chain Monte Carlo through regeneration. *J. Amer. Statist. Assoc.*, **93**, 1045–1054.

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

Hall, P. and Heyde, C.C. (1980) *Martingale Limit Theory and Its Application*. New York: Academic Press.

Jarner, S.F. and Hansen, E. (2000) Geometric ergodicity of Metropolis algorithms. *Stochastic Process. Appl.*, **85**, 341–361.

Kushner, K. and Yin, Y. (2003) *Stochastic Approximation and Recursive Algorithms and Applications*. New York: Springer-Verlag.

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

Meyn, S.P. and Tweedie, R.L. (1994) Computable bounds for convergence rates of Markov chains. *Ann. Appl. Probab.*, **4**, 981–1011.

Robbins, H. and Monro, S. (1951) A stochastic approximation method. *Ann. Math. Statist.*, **22**, 400–407.

Roberts, G.O. and Rosenthal, J.S. (2001) Optimal scaling of various Metropolis–Hastings algorithms. *Statist. Sci.*, **16**, 351–367.

Roberts, G.O. and Gelman, A. and Gilks, W. (1997) Weak convergence and optimal scaling of random walk Metropolis algorithm. *Ann. Applied Probab.*, **7**, 110–120.

Rosenthal, J.S. (2004) Adaptive MCMC Java applet. *http://probability.ca/jeff/java/adapt.html*.

Received April 2004 and revised March 2005