Scaling Limits for the Transient Phase of Local Metropolis-Hastings Algorithms

by

Ole F. Christensen*
Department of Mathematics and Statistics
University of Lancaster

and

Gareth O. Roberts*
Department of Mathematics and Statistics
University of Lancaster

and

Jeffrey S. Rosenthal**
Department of Statistics
University of Toronto

Scaling Limits for the Transient Phase
of Local Metropolis-Hastings Algorithms

by

Ole F. Christensen* and Gareth O. Roberts* and Jeffrey S Rosenthal**

Abstract. This paper considers high-dimensional Metropolis and Langevin algorithms in
their initial transient phase. In stationarity, these algorithms are well-understood and it is
now well-known how to scale their proposal distribution variances. For the random walk
Metropolis algorithm, convergence during the transient phase is extremely regular - to
the extent that the algorithm's sample path actually resembles a deterministic trajectory.
In contrast, the Langevin algorithm with variance scaled to be optimal for stationarity,
performs rather erratically. We give weak convergence results which explain both of these
types of behaviour, and give practical guidance on implementation based on our theory.

1. Introduction.

Markov chain Monte Carlo (MCMC) algorithms are a very popular method for sam-
pling from complicated probability distributions $\pi(\cdot)$ (see e.g. Gilks, Richardson and Spiegel-
halter, 1996). One very common MCMC algorithm is the Metropolis-Hastings algorithm
(Metropolis, Rosenbluth, Rosenbluth, Teller and Teller, 1953; Hastings, 1970). This algo-

* Department of Mathematics and Statistics, Fylde College, Lancaster University, Lan-
caster, LA1 4YF, England. Email: o.christensen@lancaster.ac.uk and
g.o.roberts@lancaster.ac.uk.

** Department of Statistics, University of Toronto, Toronto, Ontario, Canada M5S 3G3.
Email: jeff@math.toronto.edu. Supported in part by NSERC of Canada.

The result of Roberts et al. (1997) says that, for random-walk Metropolis algorithms, to achieve optimal mixing speed of the algorithm, the proposal standard deviation should scale with dimension $d$ like $O(d^{-1})$, and the optimal acceptance rate should be 0.234. By contrast, the result of Roberts and Rosenthal (1998) says that, for Langevin Metropolis-Hastings algorithms, the proposal standard deviation should scale with dimension $d$ like $O(d^{-1/3})$, and the optimal acceptance rate should be 0.574. These results provide very useful practical guidance.

However, these results do have limitations. Firstly, these results are limiting results as the dimension of the problem goes to infinity, and this restricts the class of target distribution for which formal scaling limits can be demonstrated. For instance, Roberts et al. (1997) and Roberts and Rosenthal (1998) only prove results rigorously when $\pi(\cdot)$ has components which are i.i.d. (at least asymptotically), though subsequent theoretical and empirical work shows that these results apply somewhat more generally (see Roberts and Rosenthal, 2001 for a survey). Secondly, and sometimes overlooked, the results assume that the chain is started in stationarity, i.e. they consider mixing properties only in the stationary phase of the chain.

In this paper, we consider what happens in the transient (pre-stationary) phase of the chain. Specifically, we consider chains which are started far out in the tails of $\pi(\cdot)$, and study their approach to the “center” of $\pi(\cdot)$. We see that, surprisingly, the behaviour of the algorithms are quite different in this transient phase. In fact, the approach to the “center” of $\pi(\cdot)$ happens deterministically as long as the proposal variance is scaled appropriately. In particular, for Langevin algorithms, the proposal standard deviation must only scale as $O(d^{-1/2})$ or smaller. If the proposal variance recede to zero more slowly than $O(d^{-1/2})$, then the convergence time becomes exponentially large as a function of dimension, exhibiting erratic behaviour and rejecting a large proportion of proposed moves in the tail region.

To illustrate that these scaling problems for the Langevin algorithm are relevant in practice we consider an example of a high-dimensional target density related to inference
for a log-Gaussian Cox point-process. For this example we demonstrate that for two
natural starting values (both being near the mode of the distribution) the algorithm is
stuck at the starting value, whereas for a starting value near to the stationary distribution
the algorithm mixes very well.

2. Definitions.

Given a $d$-dimensional probability density function of interest, $\pi$, and a proposal
kernel $Q(\mathbf{x}, \cdot)$, the Metropolis-Hastings algorithm proceeds as follows. Suppose that at
the $t$'th iteration, the current state of the algorithm is given by $\mathbf{X}_t$. Then the algorithm
proposes a new value $\mathbf{Y}_{t+1} \sim Q(\mathbf{X}_t, \cdot)$, which has proposal density with respect to the
Lebesgue measure given by $q(\mathbf{X}_t, \cdot)$. The proposal $\mathbf{Y}_{t+1}$ is accepted as the new value (and
we therefore set $\mathbf{X}_{t+1} = \mathbf{Y}_{t+1}$) with probability $\alpha(\mathbf{X}_t, \mathbf{Y}_{t+1})$ where

$$\alpha(\mathbf{x}, \mathbf{y}) = \min \left\{ 1, \frac{\pi(\mathbf{y})q(\mathbf{y}, \mathbf{x})}{\pi(\mathbf{x})q(\mathbf{x}, \mathbf{y})} \right\},$$

and otherwise rejected (where we set $\mathbf{X}_{t+1} = \mathbf{X}_t$).

Throughout this paper we will consider two algorithms. The symmetric random-walk
Metropolis algorithm takes $q(\mathbf{x}, \mathbf{y})$ to be a spherically symmetric function of $||\mathbf{y} - \mathbf{x}||$, often
denoted by $q(||\mathbf{y} - \mathbf{x}||)$. In this case, $\alpha(\mathbf{x}, \mathbf{y})$ simplifies to

$$\alpha(\mathbf{x}, \mathbf{y}) = \min \left\{ 1, \frac{\pi(\mathbf{y})}{\pi(\mathbf{x})} \right\}.$$

Therefore left to its own devices, $Q$ would carry out a random walk with increment density
$q(\cdot)$. Most of what we describe below requires only that $q(\cdot)$ be a density of a square
integrable random variable. However for simplicity we shall assume that $q$ is multivariate
Gaussian, i.e. $Q(\mathbf{x}, \cdot) \sim \text{MVN}(\mathbf{x}, hI_d)$ where $I_d$ is the $d$-dimensional identity matrix and $h$
is a proposal variance parameter.

The second algorithm we consider is the Langevin algorithm (see for example Roberts
and Tweedie, 1996) which takes $Q(\mathbf{x}, \cdot) \sim \text{MVN}(\mathbf{x} + \nabla \log \pi(\mathbf{x})h/2, hI_d)$. 

3
3. A Simple Example.

To further motivate the results we shall describe later, we consider the case where $\pi(\cdot)$ is a $d$-dimensional standard normal distribution, so that $\pi(x) = \exp(-\frac{1}{2}\|x\|^2)$.

We consider a random-walk Metropolis algorithm with Gaussian proposal distribution where the proposal variance is scaled to be proportional to $d^{-1}$ in each dimension. Thus $Y_{t+1} \sim N(X_t, (\ell^2/d) I_d)$. We also consider the Langevin algorithm with proposal variance $h$ scaled to be proportional to $d^{-1/3}$. Thus in this case $Y_{t+1} \sim N(X_t + \nabla \log \pi(X_t) \ell^2/(2d^{1/3}), (\ell^2/d^{1/3}) I_d)$. Both algorithms have variances scaled optimally as a function of $d$, with the constant in the proposal variance, $\ell$, chosen as the stationary optimal scaling (see Roberts et. al, 1997, and Roberts and Rosenthal, 1998).

The figure below shows output from simulating the target density $\pi(x) \propto \exp(-\|x\|^2/2)$ where $x$ has dimension $d = 1000$.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure1.png}
\caption{Traceplots of $\|x\|^2$ for simulating a 1000 dimensional normal distribution starting at the origin. Left: Random walk Metropolis with solid line being solution to $w'(t) = a_\ell(t)$, where $a_\ell(t)$ is defined in (1). Right: Langevin.}
\end{figure}

For the random walk Metropolis method, the initial convergence appears to be almost
deterministic, governed by the function $a(\cdot)$ defined below in (1). However once the algorithm reaches the stationary region, its trajectory becomes more obviously stochastic, displaying behaviour characteristic of a diffusion process. Note that convergence to the stationary region is fairly quick, but subsequent mixing is displaying high serial correlation.

In contrast, the Langevin method takes more than 20000 iterations to move at all, and then proceeds in a sequence of unpredictable irregular jumps to move towards the stationary region. The algorithm displays 'sticky patches' when in tail regions, where large numbers of successive iterations are rejected. However once the algorithm has succeeded in finding the stationary region, its mixing is very rapid indeed.

The theory in Roberts and Rosenthal (1998) predicts accurately the comparative performance of these two algorithms in stationarity: mixing time for the Langevin algorithm ought to be $O(d^{1/3})$ comparing favourably with $O(d)$ for the random walk Metropolis algorithm. However from this output, it is clear that the transient phase of the algorithms needs further investigation, since here, the random walk Metropolis method appears to be doing better.

4. Scaling limit for random-walk Metropolis algorithms.

We now consider the random-walk Metropolis algorithm on the above Gaussian example analytically. There is nothing particularly important about the target density being Gaussian here, but its use makes calculations explicit and easy to understand.

We let $W^d_{t+1} = (1/d)\|X_{t+1}\|^2$, so that $W^d$ is a process which keeps track of the normal-squared of $X$, with space shrunk by a factor $d$ and time speeded up by a factor $d$.

We have the following calculation.

Lemma 1. 

$$\lim_{d \to \infty} \mathbb{E} \left[ W^d_{(t+1)/d} - W^d_{t/d} \mid W^d_t = w \right] d = a_d(w),$$

where 

$$a_d(w) = \ell^2 \Phi(N^*) + \exp((\ell^2/2)(w - 1))(1 - 2w)\ell^2 \Phi(-N^* - \ell w^{1/2}),$$

(1) with $N^* = -\ell w^{-1/2}/2$, $\phi(s) = (2\pi)^{-1/2} \exp(-s^2/2)$, and $\Phi(s) = \int_{-\infty}^{s} \phi(u)du$. 

5
Proof. We write $Y_{t+1} = X_t + (\ell^2/d)^{1/2}Z$, where $Z$ is standard normal. Then the proposed value for $W_{(t+1)/d}^d$ (given $W_{t/d}^d$) is given by

$$\|X_t + (\ell^2/d)^{1/2}Z\|^2 = \|X_t\|^2 + 2(\ell^2/d)^{1/2}X_t^T Z + (\ell^2/d)\|Z\|^2$$

$$\approx \|X_t\|^2 + 2(\ell^2/d)^{1/2}X_t^T Z + \ell^2,$$

since $\|Z\|^2 \approx d$ for large $d$ by the law of large numbers. Hence, the difference $\|Y_{t+1}\|^2 - \|X_t\|^2$ is (to first order) normally distributed with mean $\ell^2$, and variance $4(\ell^2/d)\|X_t\|^2 = 4\ell^2 W_t^d = 4\ell^2 w$. We can thus write

$$\|Y_{t+1}\|^2 - \|X_t\|^2 \approx \ell^2 + 2\ell w^{1/2}U,$$

where $U \sim N(0,1)$.

Now, this proposed value is then accepted with probability equal to the minimum of 1 and

$$\frac{\pi(Y_{t+1})}{\pi(X_t)} = \frac{\exp(-\|Y_{t+1}\|^2/2)}{\exp(-\|X_t\|^2/2)} = \exp(-((\|Y_{t+1}\|^2 - \|X_t\|^2)/2)) \approx \exp(-((\ell^2 + 2\ell w^{1/2}U)/2)).$$

We conclude that, for large $d$,

$$\mathbb{E} \left[ W_{(t+1)/d}^d - W_{t/d}^d | W_{t/d}^d = w \right] \approx \mathbb{E} \left[ (\ell^2 + 2\ell w^{1/2}U) \min \left( 1, \exp(-(\ell^2 + 2\ell w^{1/2}U)/2) \right) \right],$$

with the second expectation taken with respect to $U \sim N(0,1)$. The remainder of the proof is concerned with computing this expectation, which is straightforward but tedious.

We first recall that $N^* = -\ell w^{-1/2}/2$, and note that the above “min” equals 1 for $U \leq N^*$ only. Hence, as $d \to \infty$,

$$\mathbb{E} \left[ W_{(t+1)/d}^d - W_{t/d}^d | W_{t/d}^d = w \right] \approx \int_{N^*}^\infty \phi(u) \exp(-\ell^2/2 - \ell w^{1/2} u)(\ell^2 + 2\ell w^{1/2} u) du$$

$$+ \int_{-\infty}^{N^*} \phi(u)(\ell^2 + 2\ell w^{1/2} u) du = I_1 + I_2.$$

Then

$$I_1 = \int_{N^*}^\infty \phi(u + \ell w^{1/2}) \exp(\ell^2 w/2 - \ell^2/2)(2\ell w^{1/2}(u + \ell w^{1/2}) - 2\ell^2 w + \ell^2) du$$

$$I_2 = \int_{-\infty}^{N^*} \phi(u + \ell w^{1/2}) \exp(\ell^2 w/2 - \ell^2/2)(2\ell w^{1/2}(u + \ell w^{1/2}) - 2\ell^2 w + \ell^2) du$$
\[
= \exp(\ell^2 w/2 - \ell^2/2) \left\{ 2\ell w^{1/2} \phi(N^* + \ell w^{1/2}) + (-2\ell^2 w + \ell^2) \left[ 1 - \Phi(N^* + \ell w^{1/2}) \right] \right\},
\]

\[
= 2\ell w^{1/2} \phi(N^*) + \exp(\ell^2 w/2 - \ell^2/2)(-2\ell^2 w + \ell^2) \Phi \left(-N^* - \ell w^{1/2}\right),
\]

where we first used \( \phi(u) \exp(-\ell w^{1/2}u) = \phi(u+\ell w^{1/2}) \exp(\ell^2 w/2) \), second that \( \int_u^\infty u \phi(u) du = \phi(u) \), and third that \( \phi(N^* + \ell w^{1/2}) = \phi(N^*) \exp(\ell^2/2 - \ell^2 w/2) \).

Similarly,

\[
I_2 = \ell^2 \Phi(N^*) - 2\ell w^{1/2} \phi(N^*).
\]

Combining the expressions for \( I_1 \) and \( I_2 \) the result follows.

Figure 2 shows the function \( a_\ell(w) \) for different values of \( \ell \). Convergence of the algorithm is quicker when the modulus of \( a_\ell \) is large. We see that there exists no \( \ell \) which is uniformly maximising the speed of convergence.

![Graph of the function \( a_\ell(w) \) for different values of \( \ell \).](image)

**Figure 2.** A collection of the \( a_\ell(\cdot) \) functions defined in (1) for various different values of \( \ell \). Convergence is quicker when \( a_\ell \) is as large as possible in modulus and positive for \( w < 1 \), negative for \( w > 1 \). The thick solid curve for \( \ell = 2.38 \), represents the scaling corresponding to optimal mixing for a chain started in stationarity. The dashed curve for \( \ell = \sqrt{2} \), represents the scaling produced by optimising \( a_\ell(0) \). Both thick solid and dashed
curves perform close to optimally for all values of \( w \). The four remaining curves are for \( \ell = 1, \ldots, 4 \).

**Lemma 2.**

\[
\limsup_{d \to \infty} \mathbb{E} \left[ \left( W_{(t+1)/d}^d - W_{t/d}^d \right)^2 \mid W_{t/d}^d = w \right] d^2 < \infty.
\]

**Proof.** Letting \( U \sim N(0,1) \), then for \( d \) large

\[
\mathbb{E} \left[ (W_{(t+1)/d}^d - W_{t/d}^d)^2 \mid W_{t/d}^d = w \right] d^2 \approx \mathbb{E} \left[ (\ell^2 + 2\ell w^{1/2} U)^2 \min \left( 1, \exp(-((\ell^2 + 2\ell w^{1/2} U)/2)) \right) \right]
\]

\[
\leq \mathbb{E} \left[ (\ell^2 + 2\ell w^{1/2} U)^2 \right] = \ell^4 + 4\ell^2 w,
\]

which proves the lemma.

\[\blacksquare\]

From Lemmas 1 and 2, we obtain the following.

**Theorem 3.** When \( W_0^d = w_0 \neq 1 \), then as \( d \to \infty \), we have \( W^d \Rightarrow f \), where \( \Rightarrow \) is weak convergence, and where \( f \) is a deterministic function satisfying \( f(0) = w_0 \) and

\[
f'(t) = a_\ell(f(t)),
\]

with \( a_\ell(\cdot) \) as in (1).

**Proof.** Let \( G^d_\ell \) denote the discrete time generator of \( W^d_\ell \). Then for an arbitrary bounded \( C^2 \) function with bounded second derivative, \( h(\cdot) \), and for \( w \neq 1 \)

\[
G^d_\ell h(w) = d\mathbb{E} \left[ h(W_{(t+1)/d}^d) - h(w) \mid W_{t/d}^d = w \right]
\]

\[
= d\mathbb{E} \left[ h'(w)(W_{(t+1)/d}^d - w) + h''(w)(W_{(t+1)/d}^d - w)^2/2 \mid W_{t/d}^d = w \right]
\]

where \( W^* \) represents a value in between \( w \) and \( W_{(t+1)/d}^d \). However the second term on the right hand side converges to 0 by Lemma 2, so that by Lemma 1

\[
\lim_{d \to \infty} G^d_\ell h(w) = a_\ell(w)h'(w).
\]
If \( w \neq 1 \), then \( a_\ell(w) > 0 \), so this equation describes the first-order limiting behaviour of \( W^d \).

It remains to show that the limiting process, \( f \), never reaches 1 in finite time when started at \( w_0 \), since that would violate the assumption \( w \neq 1 \) used above. However this is a straightforward property of the differential equation (2) given the following properties of the function \( u_\ell(\cdot) \) which are easily verified: \( u_\ell(\cdot) \) is a \( C^1 \) function on \([0, \infty]\) with

\[
\lim_{w \to 1} a_\ell(w) = 0.
\]

Theorem 3 therefore explains the apparent deterministic sample path behaviour of the random-walk Metropolis algorithm during its transient phase, as illustrated in Figure 1.

Started in stationarity, the mixing time of optimally scaled random walk Metropolis algorithms on reasonably behaved target distributions is \( O(d) \). What does Theorem 3 have to say about convergence starting from the transient phase? The first thing to note is that the solution to (2) does not reach state 1 in finite time started from some value \( w \neq 1 \). However for finite \( d \), the distinction between the deterministic transient phase, and the stochastic mixing phase is blurred. Again, Figure 1 illustrates this nicely.

We now argue informally that the convergence time of the algorithm is \( O(d) \) started from the transient phase. For the Markov chain \( \|X\|^2 \) it can easily be shown (for instance by recalling that \( \|X\|^2 \) is asymptotically Gaussian with variance \( d \)) that the set \( [d-1, d+1] \) is \( \epsilon(d) \)-small for \( \epsilon(d) \to 1 \) as \( d \to \infty \). However from Theorem 1, the time taken for \( W \) to reach \( 1 - 1/d \) (which of course corresponds to \( \|X\|^2 \) reaching \( d - 1 \)), is \( O(\log d) \), which corresponds to \( \|X\|^2 \) reaching a particular small set from which regeneration is almost certain, in time \( O(d + \log d) = O(d) \), and thus this must be the order of the mixing time.

5. Scaling limits for Langevin algorithms.

We now move on to study in more detail the erratic behaviour of the Langevin algorithm in its transient phase, as illustrated in Figure 1. For the example in Section 2 we first provide theoretical justification for the problematic behaviour observed when the proposal variance is \( h = \ell^2 d^{-1/3} \). Secondly, we motivate a different scaling limit for the algorithm.
For the Langevin algorithm with scaling $O(1/d^{1/3})$, we obtain a result similar to Lemma 1.

**Lemma 4.** Consider the Langevin algorithm with scaling $h = \ell^2 d^{-1/3}$. Let $W^d_t = (1/d)\|X_{t+d^{1/3}}\|^2$. Then

$$
E\left[ W^d_{(t+1)/d^{1/3}} - W^d_{t/d^{1/3}} \left| W^d_{t/d^{1/3}} = w \right. \right] d^{1/3} \approx \ell^2 (1-w) \text{acc}_d(w),
$$

where $\text{acc}_d(w) = \min\{1, \exp(-d^{1/3} \ell^4 (1-w)/8)\}$ is the acceptance probability of moves from $w$.

**Proof.** We use techniques similar to the proof of Lemma 1. For the Langevin algorithm, $Y_{t+1} = (1-h/2)X_t + \sqrt{h}Z$ with $Z$ being a standard normal random variable. From this we get

$$
\|Y_{t+1}\|^2 = (1-h/2)^2\|X_t\|^2 + 2(1-h/2)h^{1/2}X_t^T Z + h\|Z\|^2
$$

$$
\approx (1-h+h^2/4)\|X_t\|^2 + 2h^{1/2}(\|X_t\|^2)^{1/2}U + hd
$$

where $U \sim N(0,1)$.

Using that $h \to 0$ and $hd \to \infty$ we can write

$$
\|Y_{t+1}\|^2 - \|X_t\|^2 \approx hd(1-w)
$$

The acceptance probability for $Y_{t+1}$ becomes the minimum of 1 and

$$
\frac{\pi(Y_{t+1}) q(Y_{t+1}, X_t)}{\pi(X_t) q(X_t, Y_{t+1})} = \frac{\exp(-\|Y_{t+1}\|^2/2) \exp(-\|X_t - (1-h/2)Y_{t+1}\|^2/2h)}{\exp(-\|X_t\|^2/2) \exp(-\|Y_{t+1} - (1-h/2)X_t\|^2/2h)}
$$

$$
= \exp(-h(\|Y_{t+1}\|^2 - \|X_t\|^2)/8) \approx \exp(-h^2 d(1-w)/8).
$$

Thus for large $d$

$$
E[W^d_{(t+1)/d^{1/3}} - W^d_{t/d^{1/3}} \left| W^d_{t/d^{1/3}} = w \right. \right] \approx hd(1-w) \min\{1, \exp(-h^2 d(1-w)/8)\}/d.
$$

proving (by substituting $h = \ell^2 d^{-1/3}$) the result.
From Lemma 4 we see the reason for the problems reported in Section 2. The acceptance probability of moves from the origin are receding exponentially in $d^{1/3}$ leading to severe mixing problems. We also observe that for $w \geq 1$ the algorithm behaves well when using the scaling $O(d^{1/3})$. Therefore it is only starting values too close to the mode which lead to severe convergence problems.

By studying the acceptance probabilities in the proof above we see that we may fix the problem by choosing the scaling $h = \ell^2/d^{1/2}$.

**Lemma 5.** Consider the Langevin algorithm with scaling $h = \ell^2d^{-1/2}$. Let $W_t^d = (1/d)\|X_{t/d^{1/2}}\|^2$. Then

$$\lim_{d \to \infty} \left( \mathbb{E}[W_{(t+1)/d^{1/2}}^d - W_t^d \mid W_{t/d^{1/2}}^d = w] \right) d^{1/2} = b_t(w),$$

where

$$b_t(w) = \ell^2(1 - w) \min\{1, \exp(-\ell^4(1 - w)/8)\}. \tag{4}$$

$$\lim_{d \to \infty} \sup \mathbb{E} \left( (W_{(t+1)/d}^d - W_t^d)^2 \mid W_{t/d}^d = w \right) d < \infty.$$

**Proof.** The first part of the lemma follows from the proof of Lemma 4, using $h = \ell^2/d^{1/2}$. The second part of the lemma follows in a similar manner to the proof of Lemma 2 and is therefore omitted.

Using this Lemma we see that for the Langevin algorithm using a scaling $O(d^{-1/2})$ we have a result similar to Theorem 3.

**Theorem 6.** When $W_0^d = w_0 \neq 1$, then as $d \to \infty$, we have $W^d \Rightarrow f$, where $\Rightarrow$ is weak convergence, and where $f$ is a deterministic function satisfying $f(0) = w_0$ and

$$f'(t) = b_t(f(t)), \tag{5}$$

with $b_t(\cdot)$ as in (4).
We therefore should expect to see apparent deterministic sample path behaviour of the Langevin algorithm during its transient phase. However, first we need to discuss how to choose the constant \( \ell \) in \( h = \ell^2/d^{1/2} \). Figure 3 shows the function \( b_\ell(w) \) for different choices of \( \ell \). We see that there exists no \( \ell \) which is uniformly maximising the speed of convergence.

**Figure 3.** A collection of the \( b_\ell(\cdot) \) functions defined in (4) for various different values of \( \ell \). Convergence is quicker when \( b_\ell \) is as large as possible in modulus and positive for \( w < 1 \), negative for \( w > 1 \). The dashed curve is for \( \ell = 1.519671 \), and the five solid curves are for \( \ell = 1.25, 1.75, 2, 2.5, 3 \).

The value \( \ell = 1.519671 \) maximises \( b_\ell(0) \), and in Figure 4 we show a traceplot for the target density in the example in Section 3 with this choice of \( \ell \).
Figure 4. Traceplots of $\|x\|^2$ for simulating a 1000 dimensional normal distribution starting at the origin using the Langevin algorithm with proposal variance $h = 1.519671^2/d^{1/2}$. Left: Traceplot. Right: Traceplot for the first 500 iterations, where the solid line is the solution to $w'(t) = b_\ell(t)$, where $b_\ell(t)$ is defined in (4).

The initial convergence appears now to be almost deterministic and quick. Also when the algorithm has found the stationary region, its mixing is rapid. However comparing the mixing in the stationary region of the Langevin algorithm scaled as $O(d^{1/2})$ and the Langevin algorithm scaled as $O(d^{1/3})$ one sees that the mixing for the former is slower than the mixing for the later (figure omitted).

An obvious and important question is if the result in Lemma 5 is specific to the normal target density. Clearly, for other target densities, the form of (4) is different, since in the derivation we have used extensively that $\frac{\partial}{\partial x} \log \pi(x) = -x$ for the normal density. However, our conclusion about using the scaling $h = \ell^2 d^{-1/2}$ during the transient phase for the Langevin algorithm also holds for other distributions. Indeed, in the Appendix we prove the following.
Theorem 7. Suppose that $\pi(x) = \exp(\sum_{i=1}^{d} g(x_i))$ where $g$ is any four times differentiable function with mode at 0. Consider the Langevin algorithm with variance scaling $h = \ell^2d^{-1/2}$. Then $\lim_{d \to \infty} P_0(\text{move accepted})$ exists and is strictly positive.

Theorem 7 shows that for a general class of target distributions $\pi(\cdot)$, the Langevin algorithm with scaling $O(d^{-1/2})$, in contrast with $O(d^{-1/3})$, will not get “stuck” when starting from the mode.

6. Log-Gaussian Cox point-process example.

We now consider an example of a high-dimensional target density related to inference for a log-Gaussian Cox point-process. The example is from Møller, Syversveen and Waagepetersen (1998) and consists of locations of 126 Scots pine saplings in a natural forest in Finland. The locations are shown in the left plot in Figure 5.

The discretised version of the model used in the paper can be defined as follows. First the area of interest, $[0,1]^2$, is discretised into a $64 \times 64$ regular grid, where the random variables $X = \{X_{i,j}\}$ are the number of points in grid-cells $(i,j)$, $i,j = 1, \ldots, 64$. The dimension of the problem is thus $d = 64^2 = 4096$. Note that due to the fine discretisation used, most of the grid cells contain no points, and only a few contain more than one point. Given an unobserved intensity process $\Lambda(\cdot) = \{\Lambda(i,j) \mid i,j = 1, \ldots, 64\}$, the random variables $X_{i,j}, i,j = 1, \ldots, 64$, are assumed to be conditionally independent and Poisson distributed with means $m\Lambda(i,j)$, $i,j = 1, \ldots, 64$, where $m = 1/4096$ is the area of each grid-cell. The prior assumed for $\Lambda(\cdot)$ is

$$\Lambda(i,j) = \exp(Y_{i,j}),$$

where $Y = (Y_{i,j}, i,j = 1, \ldots, 64)$ is multivariate Gaussian with mean $E[Y] = \mu 1$, and covariance matrix $\text{Cov}(Y) = \Sigma$, where

$$\Sigma_{(i,j),(i',j')} = \exp(-(i-i')^2 + (j-j')^2)/(64\beta)).$$

In the paper they estimated parameter values $\beta = 1/33, \sigma^2 = 1.91$ and $\mu = \log(126) - \sigma^2/2$, which we will use here.
Møller et al. (1998) considered simulation of the intensity $\Lambda(\cdot)$ given the data $X = x$, or equivalently $Y$ given $X = x$. Using Bayes formula, the target density of interest is

$$f(y \mid x) \propto \prod_{i,j=1}^{64} \exp(x_{i,j}y_{i,j} - m \exp(y_{i,j})) \exp(-0.5(y - \mu 1)^T \Sigma^{-1}(y - \mu 1)).$$

In the paper they reparameterised $Y = \mu 1 + \Sigma^{1/2}\Gamma$ where $\Gamma$ is a vector of i.i.d. standard normal distributed random variables, and $\Sigma^{1/2}$ is a Cholesky factorisation such that $\Sigma = \Sigma^{1/2}(\Sigma^{1/2})^T$. They used a Langevin algorithm for simulating from $f(\gamma \mid x)$; the gradient of the log target density being $\nabla \Lambda(\gamma) = -\gamma + (\Sigma^{1/2})^T(x_{i,j} - m \exp(y_{i,j})\{i,j \}$ where the vector $\{y_{i,j}\}_{i,j} = \Sigma^{1/2}\gamma$. The right hand plot in Figure 5 shows the estimated intensity $\mathbb{E}[\Lambda(\cdot) \mid x]$ based on 100000 iterations, subsampling every 10'th observation and using the starting value $\Pi$ below; the plot is similar to Figure 12, upper left plot, in Møller et al. (1998), the only difference being that more grey-scale colours are used here.

Note that using the Cholesky factorisation as above is computationally slow, and is only used here for simplicity of presentation. Møller, Syversveen and Waagepetersen (1998) use a circulant embedding technique, where they extend the grid to a torus and use the two-dimensional fast Fourier transform (FFT) to reduce the computational burden. We refer to the paper for further details on this.

Note also, it is well-known that the Langevin algorithm performs much better than the random walk Metropolis algorithm for this and similar problems (see e.g. Christensen and Waagepetersen, 2002). We therefore omit any comparison with the random walk Metropolis algorithm for this problem.
Now we compare the performance of the Langevin algorithm above for three different starting values. The starting values expressed in terms of $Y$ (which have to be transformed to starting values for $\Gamma$) are

$I$: $Y_{i,j} = \mu$ for $i,j = 1, \ldots, 64$.

$II$: a random starting value, simulated from the prior $Y \sim N(\mu 1, \Sigma)$.

$III$: a starting value near the posterior mode. Let $Y_{i,j}$ solve the equation $0 = x_{i,j} - \exp(Y_{i,j}) - (Y_{i,j} - \beta)/\sigma^2$.

In all three cases we use the scaling $\hat{\ell}^2/(4096)^{1/3} = 0.16$ where $\hat{\ell} = 1.6$ is derived in the following ad hoc way. From Roberts and Rosenthal (1998) we know that $\hat{\ell}$ solves the equation $2\Phi(-J\ell^2/2) = 0.574$ where $J$ is defined in formula (4.2) in that paper. We note that for the target density consisting of a vector of standard normal distributed random variates, $J = 0.25$, and from this we would get $\ell = 1.65$. Making a normal approximation to the target density $f(\gamma \mid x)$, we see that the variance is approximately $I_{4096} + (\Sigma^{1/2})^T \text{diag}(m \exp(y_{i,j}), i,j = 1, \ldots, 64) \Sigma^{1/2}$, which we approximate by $(1 + \sigma^2 E[m \exp(y_{i,j})])I_{4096} = (1 + \sigma^2 m \exp(\beta + \sigma^2/2))I_{4096}$. From this we see that
\hat{\ell} = 1.65/(1 + \sigma^2 m \exp(\beta + \sigma^2/2))^{1/2} \approx 1.6 \text{ may be appropriate.}

For the starting value II we observed that the convergence to equilibrium was fast, say less than a hundred iterations. The overall acceptance rate was approximately 50.4% which is close to the maximal efficiency of 57.4%. For the starting values I and III many proposals were rejected before the algorithm eventually started moving. Figure 6 shows the traceplots of \log(\gamma | x) from the algorithm with these three different starting values, and we see that the reason the algorithm rejected many of the proposals for starting values I and III is because they both are far away from the equilibrium distribution, in the sense of being nearer to the mode of the distribution.

![Figure 6. Scots pine saplings. Traceplots \log(\gamma | x) when using the scaling 0.16. Left: starting value I. Middle: starting value II. Right: starting value III.](image)

For comparison Figure 7 show traceplots for the starting values I, II and III where we instead used the scaling \hat{\ell}^2/(4096)^{1/2} = 0.034. Here \hat{\ell} is derived in a similar ad hoc way as before, \hat{\ell} = 1.519671/(1 + \sigma^2 m \exp(\beta + \sigma^2/2))^{1/2} where \ell = 1.519671 was found in Section 5 to maximise the speed of convergence when starting at the mode for the standard normal distribution, i.e. maximise \hat{b}_\ell(0) with \hat{b}_\ell(\cdot) in (4). We see that the algorithm approaches the equilibrium distribution very rapidly for all three starting values. The acceptance rate was in all three cases approximately 95%, and the scaling is therefore too small in stationarity.
Figure 7. Scots pine saplings. Traceplots $\log(\gamma \mid x)$ when using the scaling 0.034. Left: starting value I. Middle: starting value II. Right: starting value III.

7. Discussion.

The results above introduce the intriguing suggestion that proposals for Langevin algorithms should be scaled very differently in the transient and stationary phases - that is once stationarity has been reached, the proposal variance can increase from $O(d^{-1/2})$ to $O(d^{-1/3})$. In practice, implementing a purely adaptive strategy which does this automatically is not a feasible solution, since it could destroy the stationarity of $\pi(\cdot)$. One simple solution to this problem is to alternate $O(d^{-1/2})$ and $O(d^{-1/3})$ moves in order to cover the possibility of being in either regime at each time-point. This strategy would resemble somewhat that of using a heavy tailed proposal distribution, a suggestion considered in Stramer and Tweedie (1999a,b) to improve the convergence of MCMC algorithms.

In contrast, optimal scalings of Metropolis algorithms have rather stable properties in the tails and it is always optimal to set the proposal variance to be $O(d^{-1})$. However even in this case, the optimal acceptance rate of the algorithm can vary. To consider this further, we note that to maximise the speed of movement towards convergence, we need to maximise $a_\ell(w)$ as a function of $\ell$ (separately for each $0 \leq w < 1$). The resulting optimal value, $\ell^*(w)$, can be numerically computed, but it depends explicitly on $w$. Fortunately,
both \( \ell = \ell^*(0) \) and \( \ell \) equal to the optimal scaling in stationarity are not too far from being optimal for all values of \( 0 \leq \omega < 1 \), as is illustrated in Figure 2.

The results also raise the question of how to chose default starting values, for algorithms. This is a very practical issue, clearly of interest for routine use of MCMC algorithms. In particular for the Langevin algorithm we have seen that natural candidates for default starting values are too near the mode of the distribution and the scaling of the algorithm becomes difficult.

As has been seen in other contexts (see for example Roberts and Rosenthal, 2001) our results demonstrate the simple Metropolis algorithm in a very positive light - it may not be quick but it has relatively robust convergence properties, requiring no special adaptive scaling strategies. In contrast, the use of Langevin methods can always be quicker (and indeed this is crucial in high dimensional examples such as that in Section 6) but great care is required in scaling. We believe these broad conclusions are of direct relevance to practical MCMC usage within statistics.


We now briefly sketch the proof of Theorem 7.

The acceptance probability for \( Y_{t+1} \) becomes the minimum of 1 and

\[
\frac{\pi(Y_{t+1}) g(Y_{t+1}, X_t)}{\pi(X_t) g(X_t, Y_{t+1})} = \exp(\sum_{i=1}^{d} g(Y_{t+1,i}) \exp(-\sum_{i=1}^{d} (X_{t,i} - Y_{t+1,i} - h/2g'(Y_{t+1},i))^2/2h) \exp(\sum_{i=1}^{d} g(X_{t,i}) \exp(-\sum_{i=1}^{d} (Y_{t+1,i} - X_{t,i} - h/2g'(X_{t},i))^2/2h)

\]

\[= \exp(-(h/8) \sum_{i=1}^{d} (g'(Y_{t+1,i})^2 - g'(X_{t,i})^2)) \times \exp(\sum_{i=1}^{d} (g(Y_{t+1,i}) - g(X_{t,i}) - (g'(Y_{t+1,i}) + g'(X_{t,i}))(Y_{t+1,i} - X_{t,i})/2))\]

This first term is similar to the term we got for the normal target density. Making a third and fourth order Taylor expansion of \( g'(Y_{t+1,i}) \) and \( g(Y_{t+1,i}) \), respectively, we get that the last term is approximately

\[\exp(-\sum_{i=1}^{d} (g''(X_{t,i})(Y_{t+1,i} - X_{t,i})^3/12 + g'''(X_{t,i})(Y_{t+1,i} - X_{t,i})^4/24)) \].
Now we study the case where $X_i$ is equal to the mode of the distribution, i.e. $g'(X_{t,i}) = 0$ and $X_{t,i} = 0$, $i = 1, \ldots, d$. This gives us

$$\frac{\pi(Y_{t+1})}{\pi(0)} \frac{q(Y_{t+1},0)}{q(0,Y_{t+1})} \approx \exp(-(h/8)g''(0)^2 \sum_{i=1}^{d} Y_{t+1,i}^2)$$

$$\times \exp(-g'''(0) \sum_{i=1}^{d} Y_{t+1,i}^3/12 - g''''(0) \sum_{i=1}^{d} Y_{t+1,i}^4/24).$$

The proposal in this case is $Y_{t+1} = h^{1/2}Z$, which implies that

$$Y_{t+1,i} = h^{1/2}Z_i$$

where $\{Z_i\}$ denote a collection of i.i.d. standard normal variables. Now observe that $\sum_{i=1}^{d} Y_{t+1,i}^3$ has mean zero, and therefore by the Central Limit Theorem is of order $d^{1/2}h^{3/2}$. Using the Law of Large Numbers we see that the terms $\sum_{i=1}^{d} Y_{t+1,i}^2$ and $\sum_{i=1}^{d} Y_{t+1,i}^4$ are asymptotically $dh$ and $3dh^2$, respectively. Therefore in the acceptance probability expression, the $Y_{t,i}^3$ terms are negligible as $d \to \infty$, and

$$\frac{\pi(Y_{t+1})}{\pi(0)} \frac{q(Y_{t+1},0)}{q(0,Y_{t+1})} \approx \exp(-h^2d(g''(0)^2 + g''''(0))/8)$$

$$= \exp(-\ell^4(g''(0)^2 + g''''(0))/8) = c, \quad \text{say},$$

when $d$ is large and $h = \ell^2d^{-1/2}$. Thus, asymptotically as $d \to \infty$, all proposed moves will be accepted with the same positive acceptance probability $c$. This proves Theorem 7.■
REFERENCES


Stramer, O. and Tweedie, R.L. (1999b), Langevin-Type Models II: Self-Targeting