An ergodicity result for Adaptive Langevin Algorithms

Tristan Marshall, Gareth Roberts

March 2009

Abstract

We consider a class of adaptive MCMC algorithms using a Langevin-type proposal density. We prove that these are algorithms are ergodic when the target density has exponential tail behaviour. Unlike previous results, our approach does not require bounding the drift function.

1 Introduction

Metropolis-Hastings algorithms [15] are an important class of Markov Chain Monte Carlo (MCMC) algorithms. When we wish to simulate from a density $\pi(x), x \in \mathcal{X}$ (the 'target') these algorithms simulate a Markov Chain $X_0, X_1, X_2, \ldots$ having π as its stationary distribution.

To be more specific, we start the algorithm in an arbitrary state X_0, and propose a move to a state y according to some proposal distribution $q(x, y)$. With probability $\alpha(x, y) = \frac{q(y,x)\pi(y)}{q(x,y)\pi(x)}$ we accept this move and set $X_1 = y$, otherwise we reject it and set $X_1 = x$. Repeat this process to obtain the sequence $X_0, X_1, X_2, \ldots$. It is straightforward to show that this process forms a Markov chain with stationary distribution π, and with some additional weak conditions on $q(x, y)$ (see [22]) it follows that $(X_n)_{n\geq 0}$ converges in distribution to π.

Since the conditions on $q(x, y)$ are very weak, we therefore have a great degree of freedom in our choice of proposal. We shall focus on proposals arising from a discretisation of a Langevin diffusion; such algorithms are commonly known as Metropolis-adjusted Langevin algorithms (MALA).

1.1 Metropolis-Adjusted Langevin Algorithms (MALA)

Metropolis-adjusted Langevin algorithms (MALA) are a class of Metropolis-Hastings algorithms using a proposal use a proposal distribution of the type

$$q_{h,\Gamma}(x,y) = \phi_d(x + \frac{1}{2}h\Gamma\nabla \log \pi(x), h\Gamma), \tag{1}$$

where $x \in \mathbb{R}^d, h \in \mathbb{R}^+$, Γ is a $d \times d$ symmetric positive definite matrix with unit determinant, and $\phi_d(u, \Sigma)$ is the density at u of a d-dimensional Normal distribution with mean 0 and covariance matrix Σ. This proposal arises from a discrete-time approximation of continuous-time processes known as Langevin diffusions; these are diffusion processes satisfying the stochastic differential equation $d\mathbf{L}_t = \Gamma^{\frac{1}{2}} d\mathbf{W}_t + \frac{1}{2}\nabla \log \pi(\mathbf{L}_t)$. Under suitable regularity conditions, Langevin diffusions converge to π in the sense that $\mathbb{P}(\mathbf{L}_t \in \cdot) \to \pi(\cdot)$ as $t \to \infty$, so at least intuitively it is plausible that a Metropolis-Hastings algorithm using a proposal that is already approximately convergent will have good convergence properties. Under suitable regularity conditions this is indeed true; more detail can be found in [20].

The proposal density (1) contains two unspecified 'tuning' parameters: a positive real h (the 'scaling coefficient') that determines the length of proposed jumps, and a symmetric positive-definite $d \times d$ matrix Γ with determinant 1 that controls the direction (the reason for using this parameterisation will become apparent in section 4). As we will see in section 2.1, the values we choose for these parameters can significantly affect the algorithm's rate of convergence.

1.2 Choosing appropriate proposal parameters

Theorem 7 of [18] states that it is 'usually' optimal to choose Γ proportional to Σ, the covariance matrix of the target π, and adjust the scaling coefficient h so that the average acceptance rate is 0.574. However since Σ is typically not available, in practice we must choose $\Gamma = h\hat{\Sigma}$, an estimate of Σ. One option is to use a sequence of 'pilot' MCMC runs to estimate Σ, but a large number of runs may be needed to obtain a good estimate $\hat{\Sigma}$ and there is no easy way to determine how many runs are necessary or how long they need to be.

An alternative to pilot runs is to use so-called 'adaptive' methods, where we continuously update the choice of Γ using the past history of the run.

More precisely, we start with an initial state x_0, initial proposal parameters (h_0, Γ_0), and proceed as follows:

Algorithm 1 (Prototype Adaptive Langevin MCMC). `1. Simulate` Y `according to the density` $q_{h_{n-1},\Gamma_{n-1}}(x_{n-1}, y)$

2. `With probability` $\frac{\pi(y)q_{h_{n-1},\Gamma_{n-1}}(y,x_{n-1})}{\pi(x_{n-1})q_{h_{n-1},\Gamma_{n-1}}(x_{n-1},y)}$ `set` $X_n = Y$. `Otherwise set` $X_n = X_{n-1}$

3. `Construct new parameters` $(h_n, \Gamma_n) = (h_n(X_0, \ldots, X_n), \Gamma_n(X_0, \ldots, X_n))$

4. `Set` $n = n + 1$ `and repeat from step 1`

(We are deliberately vague for now about how to choose $\hat{\Gamma}_n$). While an algorithm of this type might seem like a natural solution to the problem of tuning simulation parameters, finding an algorithm that converges properly requires more care. The dependence of the update density $q_{\hat{\Gamma}_{n-1}}(x_{n-1}, y)$ the past history $X_0, X_1, \ldots, X_n$ means that the algorithm 1 is no longer a Markov Chain on $\mathbb{R}^d$, and so existing results on the ergodicity of (non-adaptive) MCMC do not apply here. In [19] the authors demonstrate several 'intuitively plausible' adaptive MCMC algorithms that nonetheless fail to converge.

Several approaches to adaptive MCMC have already been proposed. Gilks et. al. [10] demonstrate an ergodic algorithm that adapts only at certain *regeneration times*; this work is extended in [8]. Unfortunately this method is often impractical: regeneration times can be difficult to identify, and the time between successive regenerations can be extremely long. Other approaches focus specifically on adaptive Random Walk Metropolis (RWM) proposals; in [11] the authors propose an adaptive Metropolis algorithm that attempts to tune the proposal variance to $(2.38)^2\Sigma/d$, where d is the dimension and Σ is the (presumably unknown) covariance matrix of the co-ordinates of the target. (This variance is optimal in certain cases by the main result of [21]; see also [6] for a generalisation of this result). Convergence is proved for this algorithm using an argument based on mixingales (see [12]). These results have since been generalised in [5] and [3], where a central limit theorem is also proved. In [19] the authors give general regularity conditions for an adaptive MCMC algorithm to converge, but it is not immediately obvious whether these will hold for an adaptive Langevin algorithm.

Adaptive Langevin algorithms have been studied previously, in [4]. Here it is proved that a particular adaptive Langevin Algorithm with a truncated drift component (similar to the T-MALA algorithm of [20] converges to $\pi(\cdot)$ if it satisfies the following conditions:

Condition 1.2.1 (Atchadé)**.** *1. The target density $\pi(\cdot)$ has finite second moment.*

2. $\pi(\cdot)$ satisfies the following conditions:

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

and

$$\limsup_{|x|\to\infty} \frac{x}{|x|} \cdot \frac{\nabla \log \pi(x)}{|\nabla \log \pi(x)|} < 0,$$

3. The 'acceptance rate in stationarity,' τ, viewed as a function of the adaptive step-size parameter h, is linear near its maximum, i.e.

$$(h - h_{\text{opt}})(\tau(h) - \tau(h_{\text{opt}})) < -\delta|h - h_{\text{opt}}|$$

for some δ, where $\tau(h)$ is defined as

$$\tau(h) = \int \int \alpha_h(x, y) q_\Gamma(x, y) \pi(\,\mathrm{d}x)\,\mathrm{d}y.$$

We will refer to this algorithm as 'AT-MALA' (Adaptive Truncated MALA).

In this paper, we develop regularity conditions under which a Langevin algorithm with the proposal form in (1) will be ergodic. Our approach involves extending a result in [20] for (non-adaptive) Langevin algorithms in order to demonstrate that the regularity conditions for a general ergodicity result in [19] hold for a given class of adaptive Langevin algorithms. We also correct a minor error in one of the key theorems of [20].

2 Adaptive MCMC algorithms

In this section we take a look at adaptive MCMC algorithms in general before focusing on adaptive Langevin algorithms in later sections. We will need some notation.

Let $\pi(x)$ be our target density, where x lies in some measurable space $(\mathcal{X}, \mathcal{F})$. As before, let $q_\lambda(x, \cdot)$ be a family of proposal density and let $\mathcal{P}_\lambda(x, \cdot)$ be the corresponding transition kernels. The subscript λ is some parameter of q that we will refer to as the *adaptive parameter* (in our prototype algorithm 1 λ is the pair (h, Γ)). We also restrict λ to lie in some set Λ - this will turn out to be necessary for convergence. Assume we have some strategy for updating λ at each time n based on the past history of $\lambda_0, \ldots, \lambda_{n-1}$ and $X_0, \ldots, X_{n-1}$; λ is then a $\sigma((X_0, \lambda_0), (X_1, \lambda_1), \ldots, (X_{n-1}, \lambda_{n-1}))$-measurable random variable taking values in Λ. We will refer to the sequence of pairs $(X_0; \lambda_0), \ldots, (X_n; \lambda_n)$ as the *adaptive chain*, and to $\lambda_0, \ldots, \lambda_n$ as the *adaptive sequence*.

Our aim is to prove that a suitable adaptive algorithm satisfies the following two conditions:

Condition 2.0.2 (Diminishing Adaptation)**.**

$$\lim_n \sup_{x \in \mathcal{X}} \|\mathcal{P}_{\lambda_{n+1}}(x, \cdot) - \mathcal{P}_{\lambda_n}(x, \cdot)\| = 0 \text{ in probability.}$$

Condition 2.0.3 (Simultaneous Geometric ergodicity)**.** *There is* $C \in \mathcal{F}, V : \mathcal{X} \to [1, \infty), \delta > 0, \lambda < 1,$ *and* $b < \infty$*, such that* $\sup_C V = v$*, and*

1. *(marginal minorisation condition) for each* $\lambda \in \Lambda$ *there is some probability measure* $\nu_\lambda(\cdot)$ *on* C *with* $\mathcal{P}_\lambda(x, \cdot) \geq \delta \nu_\lambda(\cdot)$ *for all* $x \in C$*, and*

2. *(simultaneous geometric drift condition)* $\mathcal{P}_\lambda V \leq \lambda V + b\mathbb{I}_c$ *for all* $x \in \mathcal{X}$*,*

where this latter statement holds simultaneously for all $\lambda \in \Lambda$*.*

If these conditions are satisfied, then the following theorem from [19] implies that the adaptive algorithm converges:

Theorem 2.0.4. *Suppose an adaptive chain* (X_n, λ_n) *on a state-space* $(\mathcal{X}, \mathcal{F})$ *with a family of transition kernels* $\mathcal{P}_\lambda, \lambda \in \Lambda$ *satisfies conditions 2.0.2 and 2.0.3.*

Then $\sup_{A \in \mathcal{F}} \|\mathbb{P}(X_n \in A) - \pi(A)\| \to 0$ *as* $n \to \infty$*.*

i.e. the adaptive chain (X_n, λ_n) *converges to* π *in total variation distance.*

The *diminishing adaptation* condition intuitively means that we adapt less and less as time increases. Though this condition can sometimes be awkward to verify in practice, most adaptive algorithms can be made to

satisfy it: either by forcibly reducing the absolute amount of adaptation as time increases, or by adapting at step n with probability $p_A(n)$, and forcing $p_A(n) \to 0$ as $n \to \infty$.

The simultaneous geometric ergodicity condition is a combination of two conditions: that each kernel $\mathcal{P}_\lambda(x, \cdot)$ is geometrically ergodic, and that the collection of kernels $\mathcal{P}_\lambda, \lambda \in \Lambda$ are all geometrically ergodic 'in the same way'. These two properties are less intuitive than diminishing adaptation, and are consequently harder to verify. Most of our effort in section 3 will focus on establishing these properties.

2.1 Effects of heterogeneous scaling on Langevin algorithms

The following theorem from [18] gives the optimal acceptance rate and a measure of the lost efficiency for a Langevin algorithm that uses an improperly specified covariance parameter Γ. This loss of efficiency is the motivation for adaptively adjusting Γ.

Theorem 2.1.1 (Heterogenous Langevin Scaling)**.** *Let $(X_n)_{n\geq 0}$ be a Metropolis-Hastings chain with target density π, where*

$$\pi(x) = \prod_{i=1}^{d} C_i \pi(C_i x^{(i)}), \tag{2}$$

where the C_i are i.i.d. random variables with $\mathbb{E}(C_i)^6 < \infty$, and with Langevin proposals Y of the form

$$Y \sim N(X_n + \frac{\sigma_d^2}{2} \nabla \log \pi(X_n), \sigma^2 I_d)$$

as before. Letting $\sigma_d^2 = l^2/d^{1/3}$, and defining

$$Z_t^d = X^{(1)}_{[d^{1/3}t]}.$$

Assume that the densities π satisfy the regularity conditions as in [17]. Then:

1. *Z_t^d converges weakly to*

$$\mathrm{d}Z_t = h(l)^{1/2}\, \mathrm{d}W_t + \frac{h(l) \nabla \log \pi(Z_y)}{2}\, \mathrm{d}t$$

as $d \to \infty$, *where*

$$h(l) = 2l^2\Phi(-Jkl^3),$$

where J *is the quantity*

$$J = \sqrt{\mathbb{E}_\pi\left(\frac{5(\log\pi)'''(X)^2 - 3(\log\pi)''(X)^3}{48}\right)}$$

and k *is given by*

$$k = \sqrt{\mathbb{E}(C_1^6)/\mathbb{E}(C_1)^6}$$

2. *The asymptotic acceptance rate is* $2\Phi(-kJl^3)$*, and the optimal algorithm is that having acceptance rate 0.574.*

3. *The asymptotic efficiency of the algorithm is reduced by a factor of* $k^{1/3}$ *compared to the homogeneous algorithm with* $C_i = 1 \quad \forall i$.

Asymptotic efficiency here refers to the speed measure of the limiting diffusion Z_t. Since Z_t arises as a scaled limit of the first component of X_n, this particular measure only applies to functionals of the first component. Corresponding expressions exist for all the other components.

Proof. The proof is virtually identical to the proof of Theorems 1 and 2 in [17]. If we change the likelihood used in those results to the expression in 2, replace expectation with respect to π, $\mathbb{E}_\pi[\cdot]$ with an iterated conditional expectation conditioned on the C_i, $\mathbb{E}_{(C_i)_{i\in I}}[\mathbb{E}_\pi[\cdot|(C_i)_{i\in I}]]$, and follow the (lengthy) argument through, the expressions in parts (i) and (ii) emerge without any additional complications. We find the inefficiency factor k by comparing the speed measures of the limiting diffusions in the homogeneous and non-homogeneous cases. □

Strictly speaking this result only applies to the particular family of densities in 2; in particular it requires a random structure on the scaling factors C_i that would not be present in most conventional simulation problems. However it illustrates how sensitive Langevin algorithms can be to a mis-specified scaling Γ.

If we wish to use this result to measure the inefficiency of an algorithm, we will need an empirical version of the relative inefficiency $k^{1/3}$. If we use a Langevin algorithm with covariance parameter Γ to simulate from a

multivariate Gaussian density with covariance Σ, then we define the *empirical relative inefficiency* $\hat{K}$ as:

$$\hat{K} := \sqrt[6]{\frac{\frac{1}{d}\sum_{i=1}^{d} \kappa_i^6}{\min(\kappa_i)^6}}, \tag{3}$$

where the $(\kappa_i)_{i=1}^d$ are the inverses of the eigenvalues of the matrix $\Gamma - 1/2 \Sigma \Gamma^{-1/2}$ (in the Gaussian case this is what the natural interpretation of the C_i). Our reason for using [17]).

3 Ergodicity of the adaptive algorithm

We now state and prove our main result: that an adaptive Langevin algorithm satisfying certain (intuitively plausible) conditions on the acceptance rate possesses the simultaneous geometric ergodicity condition 2.0.3 . It then only remains to prove the diminishing adaptation property 2.0.2; this turns out to be more straightforward and we address this in section 4. We examine when the regularity conditions hold for a toy family of densities in section 5.

Our argument is essentially a modified version of the proof of Theorem 4.1 of [20] (see also Appendix C, where we correct an error in one of the regularity conditions used in that paper). From this point onward we return to the parameterisation in (1), taking the adaptive parameter to be the pair $\{h, \Gamma\}$.

Let $\pi(x)$ be our target density, $c_{h,\Gamma}(x) = \frac{1}{2} h \Gamma \nabla \log \pi(x)$ be the mean proposal, and

$$q_{h,\Gamma}(x, y) := (2\pi h^d)^{-d/2} \exp\left\{h^{-1}(x - c_{h,\Gamma}(x))^T \Gamma^{-1} (x - c_{h,\Gamma}(x))\right\}$$

be our Langevin proposal density.

For a given parameter value h, Γ, we define $A_{h,\Gamma}(x)$ to be the acceptance region of $\mathcal{P}_{h,\Gamma}(x, \cdot)$ from x; the set of points such that $\mathcal{P}_{h,\Gamma}$-proposed moves from x to $A_{h,\Gamma}(x)$ are always accepted. In other words:

$$A_{h,\Gamma}(x) = \{Y : \pi(x) q_{h,\Gamma}(x, y) \leq \pi(y) q_{h,\Gamma}(y, x)\}. \tag{4}$$

We also use the expressions $R_{h,\Gamma}(x)$ and $I(x)$ for the 'potential rejection region' from x and for the interior of x respectively:

$$R_{h,\Gamma}(x) = A_{h,\Gamma}(x)^C$$
$$I(x) = \{y : |y| \leq |x|\}.$$

We will assume the following three conditions:

Condition 3.0.2. *There is an $\eta > 0$ such that:*

$$\eta \leq \liminf_{\|x\| \to \infty} (\|x\| - \|c_{h,\Gamma}(x)\|), \quad \forall \{h, \Gamma\} \in \mathcal{L}.$$

Condition 3.0.3.

$$\lim_{\|x\| \to \infty} \int_{A_{h,\Gamma}(x) \Delta I(x)} q_{h,\Gamma}(x, y) \, \mathrm{d}y = 0,$$

uniformly for all $\{h, \Gamma\} \in \mathcal{L}$.

Condition 3.0.4. *The eigenvalues λ_i^Γ of Γ are uniformly bounded above and below for all $\Gamma \in \mathcal{L}$, i.e. there are constants e and E such that $e \leq \lambda_i^\Gamma \leq E$ for all $1 \leq i \leq d$ and over all $\Gamma \in \mathcal{L}$.*

Condition 3.0.5. *The target density $\pi(x)$ is bounded away from 0 and ∞ on compact sets.*

We now state our core result:

Lemma 3.0.6. *Consider a Langevin algorithm with mean proposal $c_{h,\Gamma}(x)$ where the adaptive parameter $\{h, \Gamma\}$ is a member of some set $\mathcal{L}$. Suppose that the conditions (1)-(4) hold. Then the algorithm is simultaneously geometrically ergodic.*

Proof. We show directly that the function $V_s(x) = e^{s\|x\|}$ satisfies the conditions of Theorem 15.0.1 of [16] for suitable values of the constant s, which implies simultaneous V_s-uniform ergodicity. Consider a fixed Γ, and split the

integral $\mathcal{P}V_s(x)/V_s(x)$ over the acceptance and rejection regions:

$$\begin{aligned}\mathcal{P}V_s(x)/V_s(x) \leq (2\pi h^d)^{-\frac{d}{2}} \int_{a_{h,\Gamma}} \exp\Big\{ &-\frac{1}{2}\left(y - c_{h,\Gamma}(x)\right)^T \Gamma^{-1} \left(y - c_{h,\Gamma}(x)\right) \\ &+ s(\|y\| - \|x\|)\Big\}\, \mathrm{d}y \\ + (2\pi h^d)^{-\frac{d}{2}} \int_{R_\Gamma} \exp\Big\{ &-\frac{1}{2}\left(y - c_{h,\Gamma}(x)\right)^T \Gamma^{-1} \left(y - c_{h,\Gamma}(x)\right) \\ &+ s(\|y\| - \|x\|)\Big\}\, \alpha(x,y)\, \mathrm{d}y \\ + (2\pi h^d)^{-\frac{d}{2}} \int_{R_\Gamma} \exp\Big\{ &-\frac{1}{2}\left(y - c_{h,\Gamma}(x)\right)^T \Gamma^{-1} \left(y - c_{h,\Gamma}(x)\right) \\ &+ s(\|y\| - \|x\|)\Big\}(1 - \alpha(x,y))\, \mathrm{d}y.\end{aligned}$$

Now by intersecting the integration region of the second and third terms with $I(x)$, adding the remainder to the first term, and using the obvious upper bounds, we obtain:

$$\begin{aligned}\leq (2\pi h^d)^{-\frac{d}{2}} \int_{\mathbb{R}^d} \exp\Big\{ &-\frac{1}{2}\left(y - c_{h,\Gamma}(x)\right)^T \Gamma^{-1} \left(y - c_{h,\Gamma}(x)\right) \\ &+ s(\|y\| - \|x\|)\Big\}\, \mathrm{d}y \\ + (2\pi h^d)^{-\frac{d}{2}} \int_{R_\Gamma \cap I(x)} \exp\Big\{ &-\frac{1}{2}\left(y - c_{h,\Gamma}(x)\right)^T \Gamma^{-1} \left(y - c_{h,\Gamma}(x)\right) \\ &+ s(\|y\| - \|x\|)\Big\}\, \mathrm{d}y\end{aligned}$$

Multiplying the first term by $\exp(s(\|x\| - \|c_{h,\Gamma}(x)\|))$ and taking limits as $\|x\| \to \infty$, this product asymptotes to $\exp(s^2 \mathbf{1}^T \Gamma \mathbf{1}/2)$, where $\mathbf{1}$ is the d-dimensional vector of ones. By our third assumption, $\mathbf{1}^T\Gamma\mathbf{1}$ is bounded above, by K say, so combining this with the first condition, we see that the lim sup of the first term is less than $\exp(s^2K/2)\exp(-\eta)$, and that this bound is uniform over all $\Gamma \in \mathcal{L}$.

By our second assumption, the second term also converges to zero uni-

formly in $\mathcal{L}$, so for $s < \sqrt{2\eta/K}$ we have that

$$\limsup_{\|x\|\to\infty} \mathcal{P}\frac{V_s(x)}{V_s(x)} < 1.$$

From this it follows that for large enough r, there are constants b and c such that

$$\mathcal{P}V_s(x) \leq bV_s(x) + c\mathbb{I}_{\|x\|\leq r},$$

and that we can use the same V_s, b, c and r for all $\Gamma \in \mathcal{L}$. To complete the proof, we need to show that the set $\{x : \|x\| \leq r\}$ is small. This follows from condition 3.0.5, the continuity of $q_{h,\Gamma}(x, y)$, and Theorem 2.2 of [23] that compact sets are small. □

This result has an immediate corollary:

Corollary 3.0.7. *An adaptive chain possessing diminishing adaptation and satisfying the conditions of* (3.0.6) *is ergodic.*

Proof. This follows immediately from applying Lemma 3.0.6 to Theorem (2.0.4). □

3.0.1 Alternative regularity conditions

Although Lemma 3.0.6 describes sufficient conditions for ergodicity, these conditions are not always easy to verify. We will find the following alternative definition of the acceptance region useful in section 5:

Theorem 3.0.8 (Alternative acceptance region)**.** *For a given* Γ*, the acceptance region* $a_{h,\Gamma}(x)$ *is the set of* y *such that*

$$\int_y^x \nabla \log \pi(\mathrm{z})\, \mathrm{dz} \leq \frac{1}{2}(x-y)^T(\nabla \log \pi(x) + \nabla \log \pi(y)) + \frac{h}{8}(\nabla \log \pi(x)^T \Gamma \nabla \log \pi(x) - \nabla \log \pi(y)^T \Gamma \nabla \log \pi(y)),$$

where the left hand side is a line integral.

Proof. In appendix A. □

In section 5 we use this result to study the convergence of our algorithm on a particular family of densities.

4 Description of the Algorithm

We now outline the basic form of our adaptive Langevin algorithm.

Definition 4.0.9 (Basic Adaptive algorithm)**.** Let $\pi(x)$, $c_{h,\Gamma}(x)$, and $q_{h,\Gamma}(x,y)$ be as in section 3, and let

$$\alpha_{h,\Gamma}(X,Y) = \frac{\pi(Y)q_{h,\Gamma}(Y,X)}{\pi(X)q_{h,\Gamma}(X,Y)}$$

be the probability that a proposed move from X to Y is accepted. In a slight abuse of notation, let $\alpha(n) = 1$ if the proposed move at time $n-1$ is accepted, 0 otherwise. Let E and ϵ be large and small real constants respectively, and M be a positive integer (with a value of approximately 10). Let $\bar{h}, \underline{h}$ be upper and lower bounds on our step size h. Define the *E-truncated covariance estimator* cov^E as:

$$\mathrm{cov}^E(Z_1,\ldots,Z_n) := \frac{1}{n-1}\sum_i (Z_i^E - \bar{Z}^E)(Z_i^E - \bar{Z}^E)^T, \tag{5}$$

where $Z^E = (Z^1 \wedge E, \ldots, Z^d \wedge E)^T$ is the result of truncating all the elements of the $d \times 1$ vector Z by E, and $\bar{Z}^E$ is the mean of $(Z_1^E, \ldots Z_n^E)$.

Finally, define an increasing sequence $(t_i)_{i=1}^{\infty}$ of *covariance adaptation times*, such that $t_i > 0$, $\lim_{i\to\infty}(t_i - t_{i-1}) = \infty$ and $(t_i - t_{i-1}) > d \quad \forall i$. Also define a sequence $c(n)$ such that $c(n) \to 0$ as $n \to \infty$.

The algorithm then proceeds as follows. We use two index variables in this description - n keeps track of the number of iterations, while i records the number of times we have adapted Γ:

Algorithm 2. *1. Set $n = 1$, $i = 0$*

2. *Choose an (arbitrary) initial state X_0, multiplicative constant $h_0 > 0$ and $d \times d$ positive-definite covariance matrix Γ_0 with unit determinant.*

3. *Simulate $Y = \frac{1}{2}\Gamma_i^{1/2}\nabla \log \pi(x)\Gamma_i^{1/2} + \Gamma_i^{1/2}Z$, where Z is a standard d-dimensional normal random variable.*

4. *Set $X_n = Y$, with probability $\alpha_{h_{n-1},\Gamma_i}(X_{n-1}, Y)$, otherwise set $X_n = X_{n-1}$.*

5. *If $n = t_i$, adapt Γ:*

(a) *Compute a new covariance matrix* $\Gamma_{i+1} = \frac{\mathrm{cov}^E(X_{t_{i-1}+1},\ldots,X_{t_i})+\epsilon I_d}{\det(\mathrm{cov}^E(X_{t_{i-1}+1},\ldots,X_{t_i})+\epsilon I_d)}$, *where* I_d *is the* $d \times d$ *identity matrix.*

(b) *Set* $i = i + 1$.

6. *Compute a new step-size constant* h_n*:*

(a) *Let* $h^* = (0.001 \times h) \wedge c(n)$

(b) *If* $\frac{1}{M}\sum_{j=n-M}^{n} \alpha(j) < 0.574$, *set* $h_n = h_{n-1} - h^*$

(c) *If* $\frac{1}{M}\sum_{j=n-M}^{n} \alpha(j) \geq 0.574$, *set* $h_n = h_{n-1} + h^*$

7. *Set* $n = n + 1$ *and repeat from step 2.*

4.1 Justification of the steps

This is not the only possible adaptive Langevin algorithm satisfying the conditions of Lemma 3.0.6, but there are a number of reasons why we have constructed the algorithm in this way. We outline the key points below.

We adapt Γ_n only at the times $(t_i)_{i=0}^{\infty}$ since adapting this parameter is computationally expensive. By requiring that the time between adaptations $(t_i - t_{i-1})$ tends to infinity we reduce the computational cost and ensure that the adaptation of Γ_n satisfies the diminishing adaptation condition 2.0.2. By contrast we adapt the step-size h_n at every iteration since the computational cost is much lower. Here we ensure diminishing adaptation by defining a maximum abount of adaptation h^* and reducing this with n.

The use of the truncated covariance estimator cov^E and the addition of ϵI_d to the covariance estimate ensures that we satisfy condition 3.0.4 that the eigenvalues of Γ_n are bounded above and below. This also prevents degenerate behaviour such as Γ_n becoming singular due to numerical roundoff errors.

The updated covariance parameter Γ_{i+1} is calculated using only the subsample $X_{t_{i-1}+1}, \ldots, X_{t_i}$; the X-values simulated using the previous covariance parameter Γ_i. This is done since Γ_i is *presumably* the closest estimate so far of the true covariance Γ; using a larger sample would therefore only add noise to our covariance estimate. Empirical simulations seem to support our use of this subsample, though a formal proof does not seem possible in the absence of a proof that Γ_i converges.

The step-size h_n is adapted at each step using only the acceptance rate over the last M samples; the maximum amount of adjustment is reduced

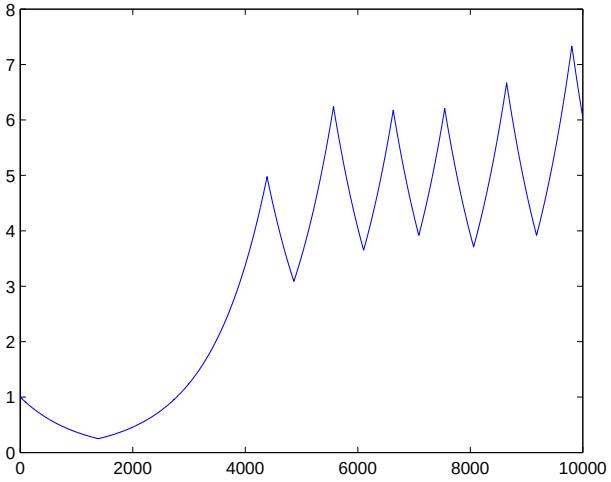

Figure 1: Step sizes against n from an adaptive simulation on a 20-dimensional Gaussian target density with $M = 500$. After a non-adaptive 'burn-in' period were the covariance multiplier is kept unchanged, the algorithm keeps 'over-correcting' the step size, leading to the observed instability.

with n to satisfy diminishing adaptation. In our simulations in section 6 we used $M = 10$; the reason for such a small M is to prevent the h-adaptation from 'overshooting'. If M is large then the empirical acceptance rate $\frac{1}{M}\sum_{j=n-M}^{n}\alpha(j)$ changes relatively slowly, which can cause h_n to rpeatedly under- and over-adapt (Fig. 1). A small value of M prevents this (Fig 2).

It is worth noting that it is straightforward to implement an adaptive-RWM algorithm with essentially the same structure as 2, the only necessary modification to the adaptation is to alter the target acceptance rate to 0.234, in line with the main result of [21]. We compare the performance of the adaptive Langevin and RWM algorithms in Section 6.

4.2 Implementing the algorithm

In practical simulations (see section 6) we have found that a good choice for the covariance adaptation times (t_i) is to take $t_1 = K$ for some constant K, then let $t_2 - t_1 = d(d-1)/2$, and increase the length of each subsequent

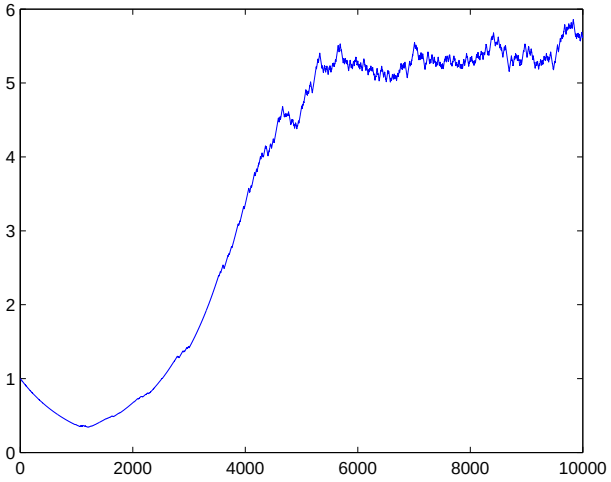

Figure 2: Step sizes against n from an adaptive simulation on a 20-dimensional Gaussian target density with $M = 10$. The instability in Fig. 1 is no longer evident.

interval geometrically , so that $t_i - t_{i-1} = (1 + a)(t_{i-1} - t_{i-2})$, for some $a > 0$. The constant K defines an 'adaptive-burn-in' interval for Γ; this allows the algorithm to find a good initial step size before choosing a new Γ. The choice of a controls the frequency of the covariance adaptations - smaller values mean more frequent adaptations using smaller subsample sizes. We have generally found it better to use small values of a; in section 6 we used $a = 0.03$.

The choice of step size adaptation bound $c(n)$ is somewhat arbitrary. In our simulations we used $c(n) = bn^{-r}$, choosing b and r so that the diminishing effect began at a point about halfway through the run.

Leaving the choice of these adaptive hyperparameters up to the user is in some ways unsatisfactory; the goal of adaptive MCMC is to remove the need for such tuning. However, since there are only a small number of adaptive hyperparameters and these will typically be much less sensitive to misspecification than h and Γ, the algorithm does substantially reduce the need for tuning by the user.

5 Application to a family of exponential models

We demonstrate our results when our algorithm is applied to the target family of exponential models given by

$$\pi_s(x) \propto e^{-\gamma\|x\|^\beta}. \quad \gamma > 0, \beta > 0 \tag{6}$$

While this may seem quite a restrictive class, this behaviour need only occur in the tails since our ergodicity results are only concerned with the limiting behaviour as $\|x\| \to \infty$. All of our upcoming remarks apply equally to any density $\pi(\cdot)$ satisfying

$$\pi(x)e^{-\gamma\|x\|^\beta} \to c \text{ as } \|x\| \to \infty. \tag{7}$$

We also note that essentially identical arguments apply to families of the form

$$\pi_s(x) \propto e^{-\gamma(x^T\Sigma^{-1}x)^{\beta/2}}. \quad \gamma > 0, \beta > 0, \tag{8}$$

where Σ is a positive definite matrix.

The behaviour of the algorithm on this family depends on the value of β (in what follows, we assume condition 3.0.4 is satisfied):

- If $0 < \beta < 1$ (the 'super-exponential' case) then we can see from Theorem 3.0.8 that for large $\|x\|$ the acceptance region $A_{\Gamma,h}(x)$ contains the region $\|y\| \geq K\|x\|$ for some K. From this it is straightforward to show that condition 3.0.3 does not hold.

- If $\beta = 1$ (the exponential case), then a similar argument shows that condition 3.0.3 still does not hold.

- If $1 < \beta < 2$ (the 'sub-exponential' case) then both condition 3.0.2 and condition 3.0.3 hold; we prove this in Appendix B.

- $\beta = 2$ (the 'Gaussian case') is a threshold case. Condition 3.0.4 together with Theorem 3.0.8 together show that for large $|x|$, the acceptance region $A_{\Gamma,h}(x)$ contains the ball $\{\|y\| \leq K\|x\|\}$, for some constant K that is independent of Γ; this is enough to imply condition 3.0.3. Condition 3.0.2 is satisfied when $\bar{h}\gamma E < 2$, where E is the upper bound from condition 3.0.4.

- If $\beta > 2$ (the 'extremely light-tailed' case) neither condition 3.0.2 nor 3.0.3 hold.

It follows that our adaptive algorithms will only be ergodic in the light-tailed case, which is consistent with the behaviour of non-adaptive algorithms. In the extremely light-tailed case, our algorithm is not ergodic.

5.1 Comparison with AT-MALA

We now compare the ergodicity properties of our algorithm with those of the adaptive T-MALA from [4]. Recall the following conditions of [4] from (1.2.1). In particular we consider the following:

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

and

$$\limsup_{|x|\to\infty} \frac{x}{|x|} \cdot \frac{\nabla \log \pi(x)}{|\nabla \log \pi(x)|} < 0.$$

For the exponential family 6 we see that:

$$\begin{aligned} \frac{x}{|x|} \cdot \nabla \log \pi(x) &= -s|x|^{s-1} \\ \frac{\nabla \pi(x)}{|\nabla \pi(x)|} \cdot \frac{x}{|x|} &= -\frac{x.x}{|x|^2} \\ &= -1 \end{aligned}$$

so that this condition is satisfied when $s > 1$. This is identical to the convergence criterion for (non-adaptive) truncated-MALA algorithms for this family as described in [20], and should therefore not be surprising.

The differences between our algorithm and the AT-MALA of [4] therefore seems to be very similar to the differences between non-adaptive MALA and T-MALA. In both cases the truncated algorithm displays more robust convergence properties, at the expense of making smaller jumps.

6 Simulation results

6.1 Gaussian target density

We tested our algorithm using a 100-dimensional Gaussian distribution as the target $\pi(\cdot)$. The mean of the target was chosen to be zero, and a covariance matrix Σ was generated by simulating a vector u of 100 independent $U[-2, 2]$ random variables, and forming the products:

$$S = uu^T$$
$$\Sigma = S^T S.$$

This produced a matrix Σ with large heterogeneities of scale (smallest and largest eigenvalues differed by a factor of 20,000). The algorithm was started with initial state $X_0 = 0$ and initial adaptive parameters $h_0 = 1.0$ and $\Gamma_0 = I_{100}$. The simulation was conducted over 2×10^6 iterations. We compared the output of this algorithm with that of an adaptive RWM algorithm started with the same X_0, h_0, and Γ_0 and run for the same number of iterations.

For comparison purposes we also performed the same simulation using an adaptive Random Walk Metropolis algorithm.

Figure 3 shows the trace of the first component over the first 100,000 iterations; the adaptive behaviour is immediately apparent. The mixing is initially extremely poor, but improves rapidly after a few iterations. This effect can also be seen in Figure 4, which shows the improvement in the mean-squared jump distance (MSJD) after each adaptation, and in Figure 5, which shows the step-size parameter h at each iteration. The improvement is surprisingly rapid; almost all the improvement in MSJD occurs after just a few iterations.

Comparison with the adaptive Random Walk Metropolis simulation show that the Langevin algorithm performs significantly better. Figure 6 shows sample autocorrelations of of selected components of the final 1×10^6 iterations from both the Langevin and RWM simulations (the first 1×10^6 iterations were discarded in order to reduce bias from early, badly tuned parts of the simulation). The Langevin autocorrelations decrease rapidly, and are statistically insignificant at lags greater than 35. The RWM autocorrelations by contrast decrease much more slowly; a more detailed autocorrelation plot shows that RWM still exhibits significant correlations at lag 1000. Also worth noting is that all components display the same autocorrelation structure: this is consistent with the theory.

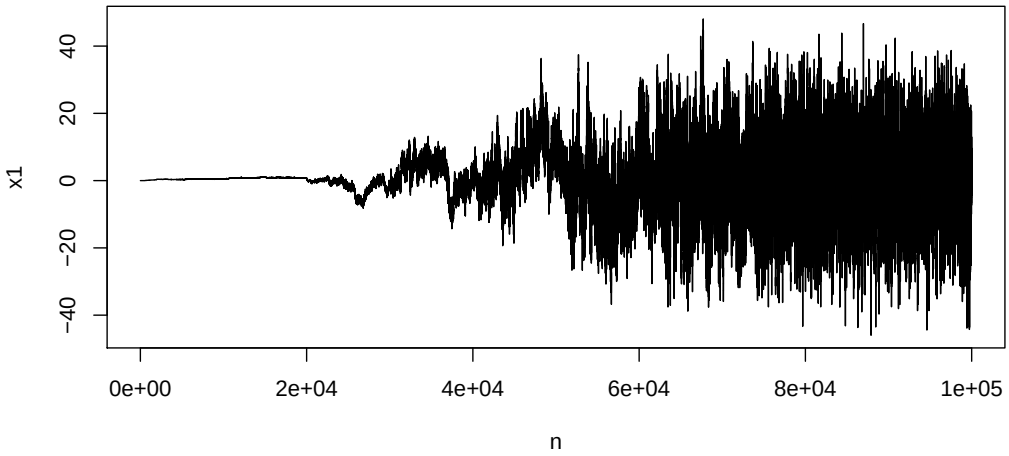

Figure 3: Trace plot of the adaptive algorithm applied to a 100-dimensional Gaussian target distribution with extremely heterogenous scaling. The plot shows the first component of the first 10^5 iterations. The algorithm is started with a step size h that is too large and spends the first 20000 iterations reducing h to obtain an optimal acceptance rate. The covariance parameter Γ is adapted at times $n = 30120, 40544, 51281, 62340, 73731, 85463, 97547$. Only the early adaptations produce any dramatic improvements in the mixing; the algorithm seems to reach its optimum efficiency quite quickly.

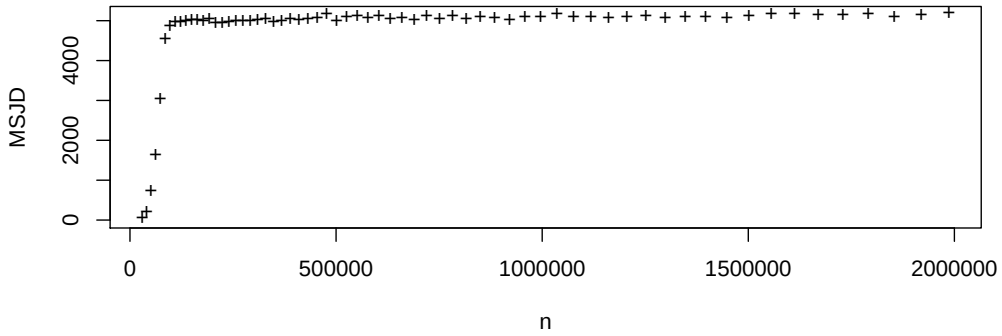

Figure 4: Mean-Squared Jump Distance (MSJD) attained by successive adaptive estimates Γ_i of the covariance parameter Γ. Total simulation length was 2×10^6 iterations, target was a 100-dimensional Gaussian. The 'x-axis' position of each '+' is the time at which Γ_i was adaptively updated to Γ_{i+1}, its 'y-axis' value is the MSJD the algorithm achieved using Γ_i. The improvement due to adaptation occurs very rapidly; the MSJD after eight adaptations is near to the maximum attained during the whole simulation. Initial MSJD is 1.655×10^{-3}, the final (and maximal) value attained is 5151.2.

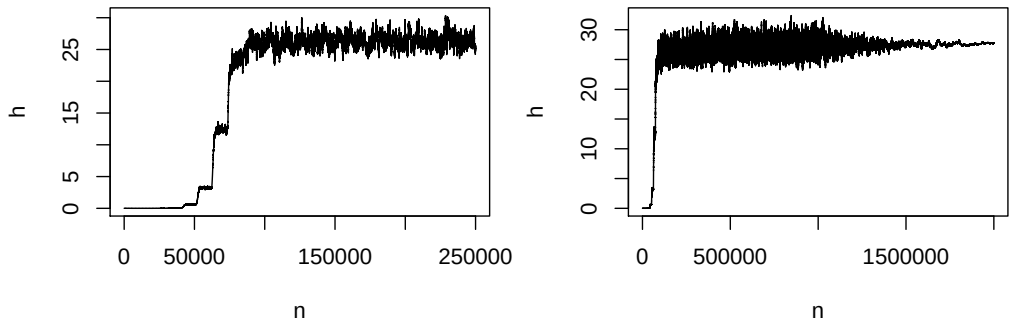

Figure 5: Values of the step size parameter h chosen by the adaptive Langevin algorithm over the course of a simulation of 2×10^6 iterations. The plot on the left shows the first 250000 iterations, that on the right shows the entire simulation. The initial h-value of 0.01 is too large - over the first 30,000 iterations the algorithm reduces h to around 3×10^{-5} (this behaviour is not visible on this scale). The rapid jumps in h occur after adaptations of Γ - this is consistent with the theory. Note also the effect of diminishing adaptation towards the end of the simulation. Final Step size is 27.708

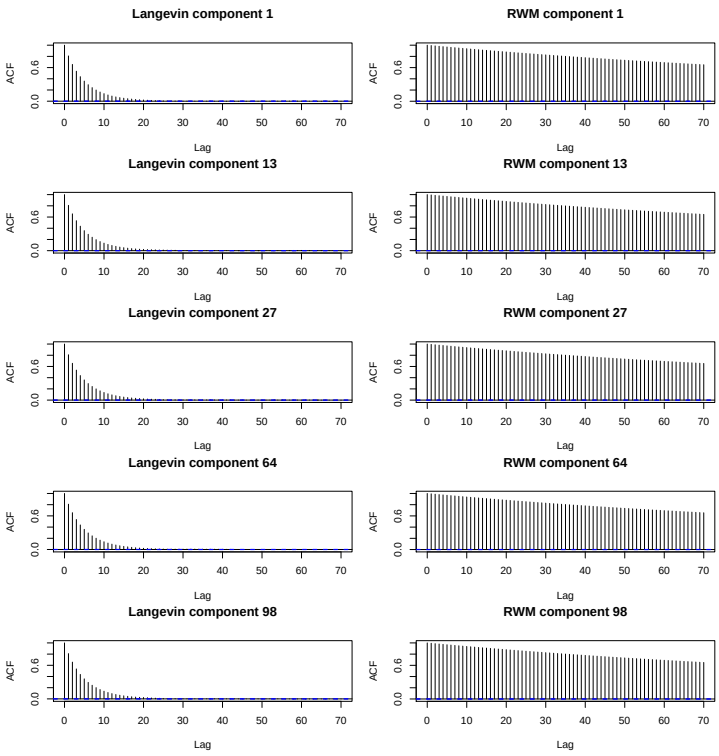

Figure 6: Sample autocorrelations of selected components of the output from the adaptive Langevin (left) and adaptive RWM (right) algorithms. The algorithms were run for 2×10^6 iterations on the same 100-dimensional Gaussian target distribution; the sample autocorrelations were computed using the final 1×10^6 iterations. The Langevin algorithm displays negligible correlation in all these components for lags greater than 30; a more detiled examination shows that the RWM algorithm still shows significant corelations even for a lag of 1000.

6.2 Log-Gaussian Cox Process

Log-Gaussian Cox process (LGCPs), introduced independently in [14] and [9], are a type of spatial point process - a finite random subset of some bounded set $S \subseteq \mathbb{R}^2$ (this is not the most general definition, but it is enough for our purposes). A point process X is a *Cox process* with random intensity $F = \{F(u) : u \in S\}$ if, conditionally on F, X is a Poisson process (see [13]) with intensity function F. X is said to be a *log-Gaussian Cox Process* if it is a Cox process and additionally $\log(F)$ is a Gaussian random field.

We consider the following parameterisation of the Gaussian field $Y := \log(F)$. Let Y have a constant mean β, and let the covariance $\mathrm{cov}(Y(u), Y(v)$ between two points u and v be given by the function:

$$c_\sigma(u, v) = \sigma^2 \exp\left(-\|u - v\|\right), \tag{9}$$

so that the covariance is stationary and isotropic.

LGCPs have found modelling applications in a variety of biological point-data problems, including forestry, weed modelling, and disease mapping (see [14], [1], [2] and [7]). A common feature of the cited examples is that the data consists of point-pattern observations $\mathbf{x} := x_1, \ldots, x_n$ of some quantity of interest within a fixed observation window W. In particular the field Y and intensity $F = \exp(Y)$ are not observed. This makes inference challenging, since there is a large quantity of missing data. An additional problem for Bayesian analysis is that the conditional density of Y given the observations $\mathbf{x}$, given by

$$\begin{aligned} f_{Y|X}((y_s)_{s\in W}|\mathbf{x}) \propto \mathbb{E}[\mathbb{P}(\mathbf{x}|(y_s)_{s\in W}) \\ \times f_Y((y_s)_{s\in W}) \end{aligned} \tag{10}$$

is not analytically tractable, since the field Y involves an infinite number of random variables. In practice we partition the window W into (rectangular) cells $(C_i)_{i\in I}$ and approximate Y by a step function $\hat{Y}$ that is constant on each C_i. This allows us to approximate the true posterior density $f_{\beta,\sigma,Y|X}$ with a computable estimate $\tilde{f}_{\beta,\sigma,\tilde{Y}|X}$ given by:

$$\begin{aligned} \tilde{f}_{\beta,\sigma,\tilde{Y}|X} \propto p(\beta)p(\sigma)\sigma^{-d} \exp\left\{-\sum_{i\in I} |C_i| \exp(\tilde{y}_i) + \sum_{i\in I} n_i \log(|C_i|) + \sum_{i\in I} n_i \tilde{y}_i\right\} \\ \times \exp\left\{-\sigma^{-2}(\tilde{\mathbf{y}} - \beta\mathbf{1})^T P(\tilde{\mathbf{y}} - \beta\mathbf{1})\right\}. \end{aligned} \tag{11}$$

Here n_i is the number of points in cell C_i, $p(\beta), p(\sigma)$ are the prior densities on β, σ respectively, and the precision matrix P is given by $P_{i,j} := \exp(\|u_i - u_j\|)$, where u_i is the point in the centre of the cell C_i. We would expect the approximate likelihood (11) to converge to the true likelihood as we refine the partition, and indeed this turns out to be true ([24]).

There is a closed form expression for $\nabla \log(\tilde{f})$, so it is possible to sample from $\tilde{f}$ using a Langevin algorithm. However there is the problem that the tails of $\tilde{f}$ are doubly exponential in Y (there is an $\exp - \exp y$ term), and the results of section 5 therefore do not apply in this example. There are several possible ways of restoring ergodicity; we choose to do so by truncating the support of $\tilde{f}$ and restricting $\beta, sigma, y_i$ to the (large) bounded region $\{\beta, \sigma, \mathbf{y} : -100 \leq \beta \leq 100, 0 < \sigma \leq 200, -720 \leq y_i \leq 720\}$. As it turned out, none of the parameters approached these bounds during our simulation. Alternative means of restoring ergodicity include replacing every Rth iteration with a Random-Walk Metropolis or Independence Sampler update.

We simulated point data $(\tilde{x})$ from $\tilde{f}$ with $\beta = 1.6$ and $\sigma = 0.9$. The observation window W was the set $[0, 30] \times [0, 10]$ and the step-function $\hat{Y}$ used was a 30×10 grid of cells of unit size; the dimension d of $\tilde{f}$ was therefore 302. In total the simulation produced $n = 2292$ points in W. On this data $\tilde{x}$ we then simulated 5×10^6 iterations of the adaptive algorithm with target density $\tilde{f}$, started from initial state $\beta_0 = 0.2, \sigma_0 = 0.5, y_i = 0.2 \ \forall i$. As outlined at the end of section 4 we used an adaptive burn-in period of 45602 $(= d^2/2)$ iterations. The first covariance adaptation was at time $n = 91204$ and the time between adaptations was increased by a factor of 1.03 after each adaptation.

Figure 8 shows trace plots from β and σ at four different points during the simulation. Both parameters show a rapid initial improvement due to adaptation of the step-size h. However, only *sigma* shows an obvious improvement after this. This is supported by the sample ACF plots (Figure 9), and by the trace plot of the step sizes (Figure 7), all of which indicate that the benefits of adaptation occur within the first 2×10^5 iterations. This is surprisingly quick given the dimensionality of the problem.

7 Concluding Remarks

We have presented an algorithm for adaptively choosing the step-size and covariance multiplier of a MALA algorithm. The ergodicity requirements for

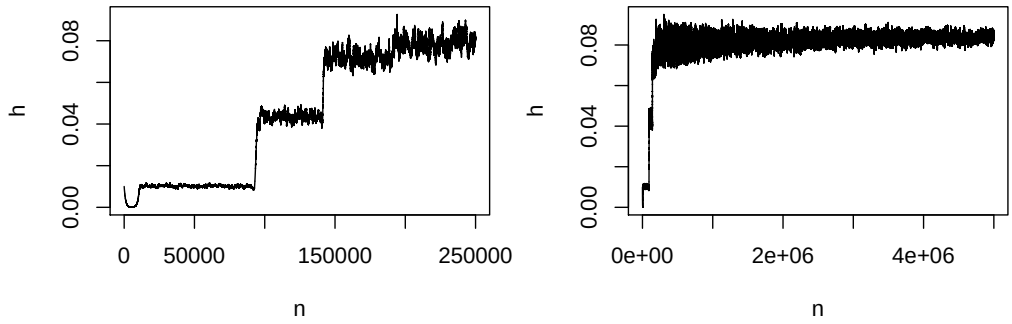

Figure 7: Trace plots of the step size parameter h over the first 100000 iterations (left), and over all 5000000 iterations (right). The first three covariance adaptations have a significant effect on the step size; the effect of subsequent adaptations is less obvious. The effects of diminishing adaptation can also be clearly seen.

the adaptive algorithm are not much stricter than those for non-adaptive MALA, and it significantly outperforms adaptive Random Walk Metropolis algorithms in situations where both algorithms are ergodic. These theoretical results are confirmed by simulations.

Our algorithm differs from the AT-MALA introduced in [4] in several key aspects. AT-MALA has weaker conditions for convergence than our algorithm; this is consistent with the behaviour of the corresponding non-adaptive algorithms (described in [20]). However, in the case of a target density where both algorithms converge, the additional truncation step in AT-MALA will only serve to reduce the mean squared jump distance and increase autocorrelations.

It is possible that the regularity conditions in Lemma 3.0.6 can be weakened. In particular the bounds placed on h_n and Γ_n seem to be too strong a condition - experimental simulations without these restrictions in place do not behave observably differently.

This is related to the question of whether the adaptive parameters (h_n, Γ_n) converge to some optimal value (h^*, Γ^*) as $n \to \infty$. Lemma 3.0.6 does not address this issue, since convergence of the adaptive parameters is unnecessary for convergence of X_n. In the simulation results of section 6 these parameters do appear to converge, but proving this convergence under suit-

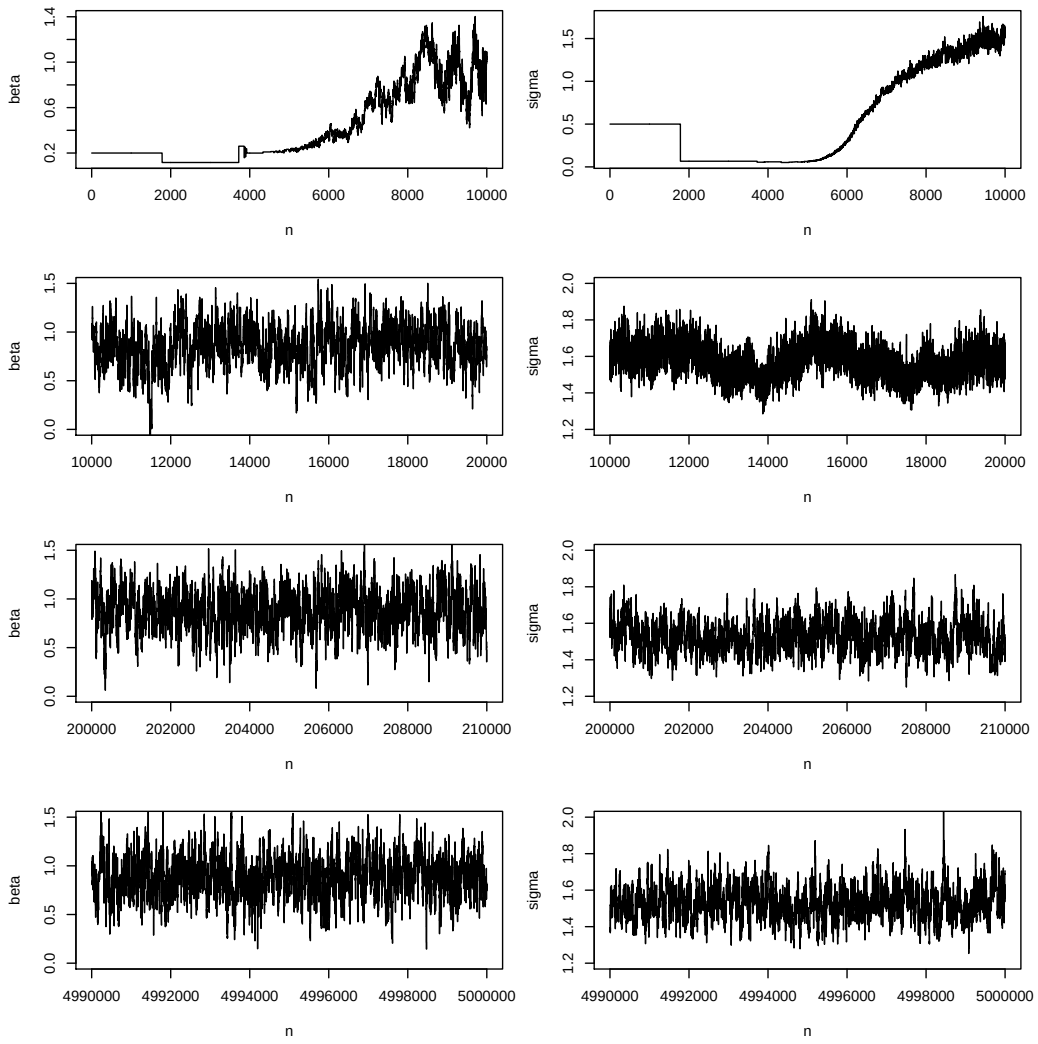

Figure 8: Trace plots of β (left) and σ (right) at different points during the simulation: iterations 1–10000 (top), 10001 – 20000 (second row), iterations 200001 – 210001, and iterations 4999001 – 5000000 (bottom). The first two adaptations of the covariance parameter Γ occur between the second and third rows. Both parameters show an improvement from the first to second row - this is due to step size adaptation. Only *sigma* shows an obvious improvement from the second to third row, and neither parameter shows a clear improvement from third to fourth row

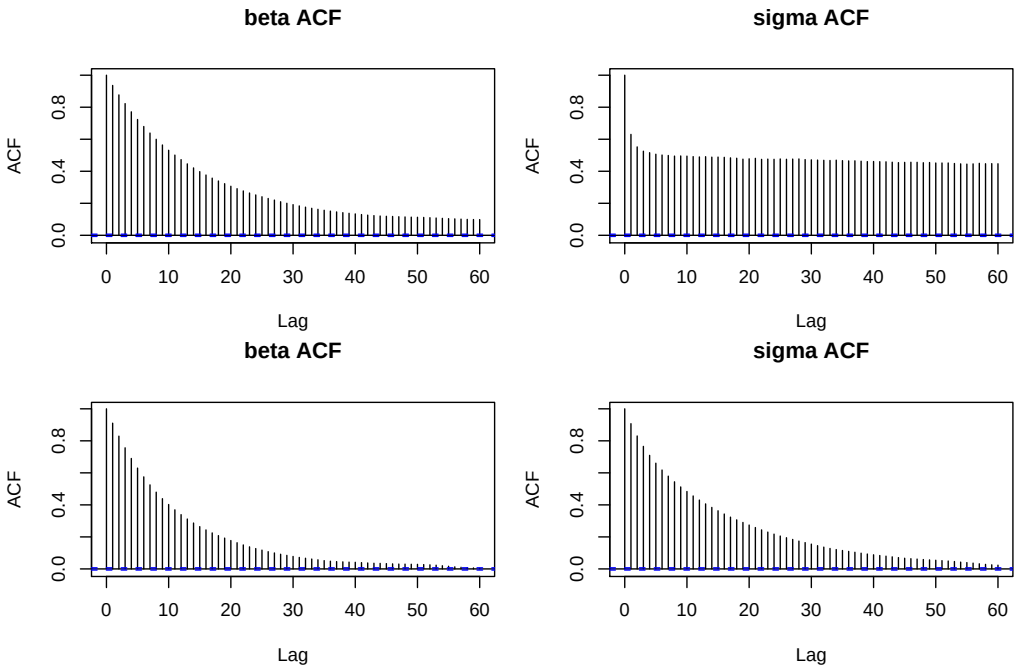

Figure 9: Sample autocorrelation functions (ACF) for *beta* and *sigma* from three different subsamples: iterations 10001 – 90000 (top), 200001 – 280000 (middle) and 4920001 – 5000000 (bottom). Both parameters show a clear reduction in large-lag autocorrelations from top to middle, but that for σ is much more pronounced. *sigma* also shows a slight increase in small-lag autocorrelations in the middle row. Neither shows any significant improvement from the middle to bottom rows.

ably weak conditions is more difficult. The author of [4] proves that these parameters converge for AT-MALA, but only by assuming strong conditions on the acceptance rates in stationarity that would be difficult to verify in practice.

Questions also remain over the optimal

A Proof of Theorem 3.0.8

By definition, $A_{h,\Gamma}(x) = \{y : \pi(x)q_{h,\Gamma}(x,y) \leq \pi(y)q_{h,\Gamma}(y,x)\}$. We prove the result by substituting in our choice of $q_{h,\Gamma}$ and manipulating this inequality.

As before, we define $c_{h,\Gamma}(x) = x + \frac{1}{2}\Gamma\nabla \log \pi(x)$ to be the mean of the Langevin proposal. Our proposal density $q_{h,\Gamma}(x,y)$ is then:

$$q_{h,\Gamma}(x,y) = (2\pi h)^{-d/2} \exp\left\{-\frac{1}{2h}(y - c_{h,\Gamma}(x))^T\Gamma^{-1}(y - c_{h,\Gamma}(x))\right\} \tag{12}$$

Substituting this into our definition of $A_{h,\Gamma}(x)$ in (4), we see that $A_{h,\Gamma}(x)$ is the set of y such that

$$\frac{\pi(x)}{\pi(y)} \leq \exp\left\{\frac{-1}{2h}\Big[(y - c_{h,\Gamma}(x)^T\Gamma^{-1}(y - c_{h,\Gamma}(x)) - (x - c_{h,\Gamma}(y))^T\Gamma^{-1}(x - c_{h,\Gamma}(y))\Big]\right\}.$$

Taking logs and using polarisation gives:

$$\log \pi(x) - \log \pi(y) \leq \frac{-1}{2h}\big[(y + x - c_{h,\Gamma}(x) - c_{h,\Gamma}(y))^T\Gamma^{-1} (y - x - c_{h,\Gamma}(x) + c_{h,\Gamma}(y))\big].$$

We now manipulate this expression further.

$$\begin{aligned}\log \pi(x) - \log \pi(y) &\leq \frac{1}{2h}\left(\frac{h}{2}\Gamma(\nabla log\pi(x) + \nabla \log \pi(y))\right)^T \Gamma^{-1} \\ &\qquad \left(2(x-y) + \frac{h}{2}\Gamma(\nabla log\pi(x) - \nabla \log \pi(y))\right) \\ &\leq \frac{1}{2}(x-y)(\nabla log\pi(x) + \nabla \log \pi(y)) \\ &\quad + \frac{h}{8}\left(\Gamma(\nabla log\pi(x) - \nabla \log \pi(y))\right)^T\left((\nabla log\pi(x) + \nabla \log \pi(y))\right).\end{aligned}$$

This simplifies to:

$$\log \pi(x) - \log \pi(y) \leq \frac{1}{2}(x-y)^T \left(\nabla log\pi(x) + \nabla \log \pi(y)\right) + \frac{h}{8}\left(\nabla \log \pi(x)^T \Gamma \nabla \log \pi(x) - \nabla \log \pi(y)^T \Gamma \nabla \log \pi(y)\right).$$

Since $\nabla \log \pi(x)$ is the gradient of the scalar field $\log \pi(x)$, we can rewrite the left hand side as a line integral.

B Proof of regularity conditions (1) and (2) for the exponential family $\pi(x) = e^{-\gamma \|x\|^\beta}$ when $1 < \beta < 2$

We seek to prove that the following conditions hold for the exponential family $\pi(x) = e^{-\gamma\|x\|^\beta}$ when $1 < \beta < 2$.

1. $\exists \eta > 0$ s.t. $\eta \leq \liminf_{\|x\|\to\infty}(\|x\| - \|c_\Gamma(x)\|), \qquad \forall \Gamma \in \mathcal{L}$
2. $\lim_{\|x\|\to\infty} \int_{A_\Gamma(x)\Delta I(x)} q_\Gamma(x,y)\,\mathrm{d}y = 0, \qquad$ uniformly for all $\Gamma \in \mathcal{L}$.

B.0.1 Condition 1

To prove the first condition for all $\Gamma \in \mathcal{L}$, it is sufficient to prove that there is a constant K such that

$$\left\| x - \gamma\frac{\beta}{2}\Gamma x \|x\|^{\beta-2} \right\|^2 < \|x\|^2 \tag{13}$$

for all $\|x\| > K$, and that K can be chosen independently of Γ.

Recall that from condition 3.0.4, we have global lower and upper bounds e, E on the eigenvalues $(\lambda_i^\Gamma)_{i=1}^d$ of all $\Gamma \in \mathcal{L}$, so that $e \leq \lambda_i^\Gamma \leq E$ for all i, Γ.

We start by writing the norm as an inner product

$$\begin{aligned}\left\| x - \gamma\frac{\beta}{2}\Gamma x \|x\|^{\beta-2} \right\|^2 &= \left\langle x - \gamma\frac{\beta}{2}\Gamma x\|x\|^{\beta-2}, x - \gamma\frac{\beta}{2}\Gamma x\|x\|^{\beta-2} \right\rangle \\ &= \|x\|^2 - \gamma\beta\|x\|^{\beta-2}\langle x, \Gamma\rangle + \frac{\gamma^2\beta^2}{4}\|x\|^{2(\beta-2)}\langle \Gamma x, \Gamma x\rangle.\end{aligned}$$

So if we can show that

$$\frac{\gamma\beta}{4}\|x\|^{\beta-2}\langle\Gamma x,\Gamma x\rangle < \langle x,\Gamma x\rangle, \tag{14}$$

then this will be enough to imply 13. We continue by observing that

$$\begin{aligned}\frac{\gamma^\beta}{4}\|x\|^{\beta-2}\langle\Gamma x,\Gamma x\rangle &\leq \frac{\gamma^\beta}{4}\|x\|^{\beta-2}E^2\|x\|^2\\ &= \frac{\gamma^\beta}{4}E^2\|x\|^\beta\end{aligned}$$

and

$$\langle x,\Gamma x\rangle \geq e\|x\|^2, \tag{15}$$

and so it is sufficient to prove that

$$e\|x\|^2 > \frac{\gamma^\beta}{4}E^2\|x\|^\beta \tag{16}$$

Since $\beta < 2$, it follows that we can choose K such that (16) holds for all $\|x\| > K$. Since none of the constants in (16) involve Γ, it follows that K does not depend on Γ, and the result follows.

B.0.2 Condition 2

Substituting $\pi(x) = e^{-\gamma\|x\|^\beta}$ into Theorem 3.0.8 we see that a point y is in the acceptance region $A(x)$ if and only if:

$$\begin{aligned}-\gamma\|x\|^\beta + \gamma\|y\|^\beta \leq \frac{-\gamma\beta}{2}(x-y)^T(\|x\|^{\beta-2}x + \|y\|^{\beta-2}y)\\ + \frac{h}{8}\gamma^2\beta^2(\|x\|^{2(\beta-2)}x^T\Gamma x - \|y\|^{2(\beta-2)}y^T\Gamma y).\end{aligned}$$

Re-arranging this expression we see that $y \in A(x)$ iff:

$$\begin{aligned}0 \leq (\gamma - \frac{1}{2}\gamma\beta)(\|x\|^\beta - \|y\|^\beta) + \frac{1}{2}\gamma\beta(\|x\|^{\beta-2} - \|y|^{\beta-2})\langle x,y\rangle\\ + \frac{h}{8}\gamma^2\beta^2(\|x\|^{2(\beta-2)}x^T\Gamma x - \|y\|^{2(\beta-2)}y^T\Gamma y).\end{aligned}$$

The first term on the right hand side is positive for all $\|x\| > \|y\|$. An application of Cauchy-Schwarz to the second term shows that it is dominated by the first term as $\|x\| \to \infty$. Condition 3.0.4 allows us to place uniform (in Γ) Lipschitz bounds on the matrix products in the third term, from which we can deduce that this term is also positive for sufficiently large $\|x\|$.

Together these imply that for every r we can choose $K = K(r)$ such that $\{y : \|y\| < r\}$, the ball of radius r is in the acceptance region $A(x)$ for all $\|x\| > K(r)$, and that we can choose K such that it does not depend on Γ. This together with the fact that $\int_{\mathcal{X}} q_{h,\Gamma}(x, y)\, \mathrm{d}y = 1$ implies the result.

C Correction to Theorem 18 of [20]

We begin by defining non-adaptive versions of the quantities in section 3

As before, let $\pi(x)$ be our target density, $c(x) = \frac{1}{2}\nabla \log \pi(x)$ be the mean proposal, and

$$q(x, y) := (2\pi)^{-d/2} \exp\left\{(x - c(x))^T (x - c(x))\right\}$$

be our Langevin proposal density.

Similarly, define $A(x)$ and $R(x)$, as the obvious analogues to $A_{h,\Gamma}(\cdot)$ and $R_{h,\Gamma}(x)$, and let $I(x)$ be the interior as before.

With these quantities defined, we can now state the corrected version of the earlier theorem:

Theorem C.0.1 (Roberts and Tweedie, 1996, corrected)**.** *Suppose that* $c(x) = x + \frac{1}{2}h\nabla \log \pi(x)$ *is the mean "next candidate position", and that*

$$\eta \equiv \liminf_{|x|\to\infty} (|x| - |c(x)|) > 0; \tag{17}$$

and assume $A(\cdot)$ *converges inwards in* q*. If* $V_s(x) = e^{s|x|}$*, then the algorithm is* V_s *uniformly ergodic for* $s < 2\eta/h$*.*

Proof. Omitted, since this result is a special case of Lemma 3.0.6 and the proofs are nearly identical. □

The uncorrected version stated that the result held for $s < 2h\eta$.

References

[1] Jesper Mø ller Anders Brix. Space-time multi type log gaussian cox processes with a view to modelling weeds. *Scandinavian Journal of Statistics*, 28(3):471–488, 2001.

[2] Peter J. Diggle Anders Brix. Spatiotemporal prediction for log-gaussian cox processes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(4):823–841, 2001.

[3] C. Andrieu and E. Moulines. On the ergodicity properties of some adaptive markov chain monte carlo algorithms. *Ann. Appl. Probab.*, 16(3):1462–1505, 2006.

[4] Yves Atchadé. An adaptive version for the metropolis adjusted langevin algorithm with a truncated drift. *Methodology and Computing in Applied Probability*, 2005.

[5] Yves Atchadé and Jeffrey Rosenthal. On adaptive markov chain monte carlo algorithms. *Bernoulli*, 11(5):815–828, 2005.

[6] Mylène Bédard. *On the Robustness of Optimal Scaling for Random Walk Metropolis Algorithms*. PhD thesis, University of Toronto, 2006.

[7] V. Benes, K. Bodlak, J. Møller, and R.P. Waagepetersen. A case study on point process modelling in disease mapping. *Image Analysis and Stereology*, 24:159–168, 2005.

[8] A. E. Brockwell and J. B. Kadane. Identification of regeneration times in mcmc simulation, with aplication to adaptive schemes. *J. Comp. Graph. Stat.*, 14:436–458, 2005.

[9] P. Coles and B. Jones. A lognormal model for the cosmological mass distribution. *Royal Astronomical Society, Monthly Notices*, 248:1–13, January 1991.

[10] Walter R. Gilks, Gareth O. Roberts, and Sujit K. Sahu. Adaptive markov chain monte carlo through regeneration. *Journal of the American Statistical Association*, 93(443):1045–, 1998.

[11] H. Haario, E. Saksman, and J. Tamminen. An adaptive metropolis algorithm. *Bernoulli*, 7:223–242, 2001.

[12] Peter Hall and C. C. Heyde. *Martingale Limit Theory and its Application*. Academic Press, 1980.

[13] J. F. C. Kingman. *Poisson processes*, volume 3 of *Oxford Studies in Probability*. The Clarendon Press Oxford University Press, New York, 1993. Oxford Science Publications.

[14] Jesper Møller, Anne Randi Syversveen, and Rasmus Plenge Waagepetersen. Log gaussian cox processes. *Scandinavian Journal of Statistics*, 25(3):451–482, 1998.

[15] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *J. Chem Phys.*, 21:1087–1092, 1953.

[16] S.P. Meyn and R.L. Tweedie. *Markov Chains and Stochastic Stability*. Springer-Verlag, London, 1993. Available at probability.ca/MT.

[17] Gareth Roberts and Jeffrey Rosenthal. Optimal scaling of discrete approximations to langevin diffusions. *Journal of the Royal Statistical Society. Series B*, 60(1):255–268, 1995.

[18] Gareth Roberts and Jeffrey Rosenthal. Optimal scaling for various metropolis-hastings algorithms. *Statistical Science*, 15(4):351–367, 2001.

[19] Gareth Roberts and Jeffrey Rosenthal. Coupling and ergodicity of adaptive mcmc. *J. Appl. Prob.*, 44(2):458–477, 2007.

[20] Gareth Roberts and Richard Tweedie. Exponential convergence of langevin diffusions and their discrete approximations. *Bernoulli*, 2(4):341–363, 1996.

[21] G.O. Roberts, A. Gelman, and W.R. Gilks. Weak convergence and optimal scaling of random walk metropolis algorithms. *Ann. Appl. Prob.*, 7(1):110–120, 1996.

[22] G.O. Roberts and A.F.M. Smith. Simple conditions for the convergence of the gibbs sampler and metropolis-hastings algorithms. *Stochastic Processes and their Applications*, 49:207–216, 1994.

[23] G.O. Roberts and R.L. Tweedie. Geometric convergence and central limit theorems for multidimensional hastings and metropolis algorithms. *Biometrika*, 83(1):95–110, 1996.

[24] Rasmus Waagepetersen. Convergence of posteriors for discretized log gaussian cox processes. *Statistics and Probability Letters*, 66(3):229 – 235, 2004.