Particle Metropolis adjusted Langevin algorithms

Christopher Nemeth, Chris Sherlock and Paul Fearnhead

Lancaster University

December 24, 2014

Abstract

Particle MCMC has recently been introduced as a class of algorithms that can be used to analyse state space models. They use MCMC moves to update the parameters of the model, and particle filters to both propose values for the latent state and to obtain estimates of the posterior density that are used to calculate the acceptance probability. For many applications it is easy to adapt the particle filter so that it also gives an estimate of the gradient of the log-posterior, and this estimate can then be used within the proposal distribution for the parameters. This results in a particle version of a Metropolis adjusted Langevin algorithm, which we call particle MALA. We investigate the theoretical properties of particle MALA under standard asymptotics, which correspond to an increasing dimension of the parameters, \( n \). Our results show that the behaviour of particle MALA depends crucially on how accurately we can estimate the gradient of the log-posterior. If the error in the estimate of the gradient is not controlled sufficiently well as we increase dimension then asymptotically there will be no advantage in using particle MALA over a particle MCMC algorithm using a random-walk proposal. However if the error is well-behaved, then the optimal scaling of particle MALA proposals will be \( O(n^{-1/6}) \) as compared to \( O(n^{-1/2}) \) when a random walk proposal is used. Furthermore, we show that asymptotically the optimal acceptance rate is 15.47% and that we should tune the number of particles so that the variance of our estimate of the log-posterior is roughly 3. We also propose a novel implementation of particle MALA, based on the approach of Nemeth et al. (2013) for estimating the gradient of the log-posterior. Empirical results suggest that such an implementation is more efficient than other recently proposed particle MALA algorithms.

1 Introduction

Markov chain Monte Carlo (MCMC) algorithms are a popular and well-studied methodology that can be used to draw samples from posterior distributions. Over the past few years MCMC methodology has been extended to tackle problems where the model likelihood is intractable. For such models it is often possible to replace the intractable likelihood with an estimate (Beaumont, 2003), which can be obtained from Monte Carlo simulations. Andrieu and Roberts (2009) showed that within the MCMC sampler, if the likelihood is replaced with an unbiased estimate, then the sampler still targets the correct stationary distribution. Andrieu et al. (2010) extended this work further to create a class
of MCMC algorithms based on sequential Monte Carlo methods (also known as particle filters). This class of algorithms is referred to as particle MCMC. In this paper we shall focus on one particular algorithm, the particle marginal Metropolis Hastings algorithm, which replaces the likelihood term in the Metropolis Hastings (MH) sampler with an unbiased particle filter estimator.

The current default implementation of this method uses random walk proposals to update the parameters (e.g. [Golightly and Wilkinson 2011], [Knape and de Valpine 2012]) and shall be referred to herein as the particle random walk Metropolis algorithm, or particle RWM. However, we can often obtain further information about the posterior from our run of the particle filter, at little or no computational overhead. So it is natural to see if we can use this information to make better proposals for the parameters. We focus on using particle filter methods that will produce estimates of the gradient of the log posterior, and then use this gradient information to guide the proposed parameters towards regions of higher posterior probability. This results in a particle version of the Metropolis adjusted Langevin algorithm (MALA), which we call particle MALA.

For standard MCMC it is known that MALA has better theoretical properties than the random walk Metropolis (RWM). For example, the mixing of such algorithms has been studied in the asymptotic limit as the dimension of the parameter, \( n \), increases. For this asymptotic regime the optimal proposal step-size scales as \( n^{-1/2} \) for the RWM, but as \( n^{-1/6} \) for MALA; and the optimal asymptotic acceptance rate is higher for MALA than for the RWM; see [Roberts et al. 1997], [Roberts and Rosenthal 1998] and [Roberts and Rosenthal 2001] for more details. It is natural to ask whether these advantages of MALA over RWM extend to particle MCMC algorithms, and, in particular, how they are affected by the fact that within particle MALA we only have a noisy estimate of the gradient of the log-posterior.

We investigate the asymptotic properties of particle MALA. We show that the behaviour of particle MALA depends crucially on the behaviour of the accuracy of the estimate of the gradient of the log-posterior as \( n \) increases. If the error in the estimate of a component of the gradient does not decay sufficiently quickly with \( n \), then particle MALA will be no better than particle RWM. If the error is well-behaved then we find that particle MALA inherits the same asymptotic advantages over particle RWM that MALA has over RWM, Namely, the optimal proposal scales like \( n^{-1/6} \), rather than \( n^{-1/2} \), and there is a higher optimal acceptance rate. Furthermore, we obtain explicit results for the optimal acceptance rate, 15.47\%, and the required accuracy of the estimate of the log-posterior, results which are themselves important for tuning particle MALA.

We are not the first to propose particle MALA (see also [Dahlin et al. 2013]), though we do propose a novel implementation of particle MALA which uses a recent algorithm ([Nemeth et al. 2013]) to estimate the gradient of the log-posterior. Results from [Nemeth et al. 2013] suggest that, for a given computational cost, this algorithm is often more accurate than either the approach used by [Dahlin et al. 2013] or alternative approaches ([Poyiadjis et al. 2011]). For concreteness we present our implementation of particle MALA specifically for analysing state space models, which are the most common models for which particle MCMC is used. However both the algorithms and theory extend beyond this class of models to others for which particle MCMC is possible.

The outline of the paper is as follows. We first give an introduction to state space models, and to MCMC and sequential Monte Carlo (particle filter) algorithms for analysing these models. In Section 3 we introduce particle MCMC, and show that information from running the particle filter can be used to guide the choice of proposal distribution for the
parameters. We then introduce our particle MALA algorithm and in Section 4 present
important theoretical results. Section 5 presents empirical results which show the ro-
bustness of the asymptotic results established in Section 4 for finite dimensional models.
Furthermore, the simulation results show, in terms of effective sample size, the improve-
ment of particle MALA over both particle RWM and a popular alternative gradient-based
proposal. The paper ends with a discussion.

2 Inference for state space models

Consider the general state space model where there is a latent Markov process \( \{ X_t; 1 \leq t \leq T \} \) that takes values on some measurable space \( \mathcal{X} \subseteq \mathbb{R}^n \). The process is fully
characterised by its initial density \( p(x_1|\theta) = \mu_\theta(x_1) \) and transition probability density
\[
p(x_t|x_{t-1},\theta) = p(x_t|x_{t-1},\theta) = f_\theta(x_t|x_{t-1}),
\]
where \( \theta \in \Theta \) represents a vector of model parameters. For an arbitrary sequence \( \{ z_i \} \) the
notation \( z_{i:j} \) corresponds to \( (z_i, z_{i+1}, \ldots, z_j) \) for \( i \leq j \).

We assume that the process \( \{ X_t \} \) is not directly observable, but partial observations
are received via a second process \( \{ Y_t; 1 \leq t \leq T \} \subseteq \mathcal{Y}^n \). The observations \( \{ Y_t \} \) are
conditionally independent given \( \{ X_t \} \) and are defined by the probability density
\[
p(y_t|y_{1:t-1},x_{1:t},\theta) = p(y_t|x_t,\theta) = g_\theta(y_t|x_t).
\]

The marginal likelihood of observations for a given \( \theta \) can be decomposed as
\[
p(y_{1:T}|\theta) = p(y_1|\theta) \prod_{t=2}^{T} p(y_t|y_{1:t-1},\theta),
\]
where,
\[
p(y_t|y_{1:t-1},\theta) = \int g_\theta(y_t|x_t) \int f_\theta(x_t|x_{t-1}) p(x_{t-1}|y_{1:t-1},\theta) dx_{t-1} dx_t
\]
is the predictive likelihood.

Aside from a few special cases, it is generally not possible to evaluate the likelihood
analytically, but it is often possible to approximate the likelihood using importance sam-
pling (Pitt, 2002). Model parameters \( \theta \) can be estimated by maximising the likelihood
using expectation maximisation (EM) or gradient based maximum likelihood methods
(Nemeth et al., 2013; Poyiadjis et al., 2011; Dempster et al., 1977). Alternatively, within
the Bayesian framework, MCMC techniques (Andrieu et al., 2010; Fearnhead, 2011) can
be applied to estimate the posterior density \( \pi(\theta) = p(\theta|y_{1:T}) \) of the parameters condi-
tional on the observed data. Within this paper we shall consider only the latter case of
applying MCMC to state space models.

2.1 MCMC for state space models

We start by considering the generic MCMC algorithm used to perform Bayesian inference
on the parameters \( \theta \). Firstly, we introduce a prior distribution for the parameters, \( p(\theta) \).
Our goal is then to estimate the posterior density \( \pi(\theta) \propto p(y_{1:T}|\theta)p(\theta) \), which is known
only up to a constant of proportionality. In this setting we are considering the ideal case,
where we assume that the likelihood \( \mathbb{I} \) is known and tractable.
Samples from the posterior \((\theta_1, \theta_2, \ldots, \theta_j, \ldots, \theta_T)\) are generated using the Metropolis Hastings algorithm, where proposed values \(\theta'\) are sampled from a proposal distribution \(q(\cdot|\theta_{j-1})\) and accepted with probability

\[
\alpha(\theta'|\theta_{j-1}) = \min \left\{ 1, \frac{p(y_{1:T}|\theta')p(\theta'|\theta_{j-1})}{p(y_{1:T}|\theta_{j-1})p(\theta'|\theta_{j-1})} \right\}.
\]

(2)

The samples \(\{\theta_j\}_{j=1}^T\) generated by the MH algorithm form a Markov chain of correlated samples. The choice of proposal distribution \(q(\theta'|\theta)\) is important as it affects the mixing of the MCMC sampler. A standard choice of proposal is the Gaussian random walk proposal. This proposal generates new parameter values by perturbing the current parameters with noise sampled from a Gaussian distribution with zero mean and covariance matrix \(\Sigma\). The covariance matrix can be tuned to account for correlations in the parameter vector by running a pilot simulation to get an approximation of the posterior. Or in the simplest case, each element of \(\theta\) is perturbed independently by replacing the covariance matrix with \(\lambda^2 I\), where \(I\) is the identity matrix and \(\lambda^2\) is a user chosen step size parameter. For this simple case, new parameters \(\theta'\) are sampled from

\[\theta' = \theta_{j-1} + \lambda z \quad \text{where} \quad z \sim \mathcal{N}(0, I).\]

(3)

Alternatively, we can use local information about the posterior within the proposal. One such approach is the MALA proposal \(\text{(Roberts and Rosenthal} 1998)\) which uses the gradient of the log posterior \(\nabla \log \pi(\theta)\) within the proposal

\[\theta' = \theta_{j-1} + \lambda z + \frac{\lambda^2}{2} \nabla \log p(x_{1:T}|\theta_{j-1}) \quad z \sim \mathcal{N}(0, I).\]

The benefit of using MALA over the RWM is most notable when \(n\), the number of parameters, is large. This is because the scaling parameter \(\lambda\) is \(O(n^{-1/2})\) for RWM \(\text{(Roberts et al.} 1997)\), but is \(O(n^{-1/6})\) for MALA \(\text{(Roberts and Rosenthal} 1998)\). This means that, for large \(n\), MALA proposes larger jumps in the posterior compared to RWM. This leads to reduced first order auto-correlation and better mixing of the Markov chain.

### 2.2 Sequential Monte Carlo

Sequential Monte Carlo algorithms represent a class of simulation methods for the sequential approximation of posterior probability distributions. In the context of state space modelling, we are interested in approximating the posterior \(p(x_t|y_{1:t}, \theta)\) of the filtered latent state \(x_t\), given a sequence of observations \(y_{1:t}\). In this section we shall assume that the model parameters \(\theta\) are fixed. Approximations of \(p(x_t|y_{1:t}, \theta)\) can be calculated recursively by first approximating \(p(x_1|y_{1:t}, \theta)\), then \(p(x_2|y_{1:t}, \theta)\) and so forth for \(t = 1, \ldots, T\).

At time \(t\) the posterior of the filtered state is

\[p(x_t|y_{1:t}, \theta) \propto g_\theta(y_t|x_t) \int f_\theta(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1}, \theta)dx_{t-1}\]

(4)

where \(p(x_{t-1}|y_{1:t-1}, \theta)\) is the posterior at time \(t - 1\).

The posterior at time \(t\) can be approximated if we assume that at time \(t - 1\) we have a set of particles \(\{x_{t-1}^{(i)}\}_{i=1}^N\) and corresponding weights \(\{w_{t-1}^{(i)}\}_{i=1}^N\) which produce a discrete approximation of \(p(x_{t-1}|y_{1:t-1}, \theta)\). This induces the following approximation to (4),

\[
\hat{p}(x_t|y_{1:t}, \theta) \approx cg_\theta(y_t|x_t) \sum_{i=1}^N w_{t-1}^{(i)} f_\theta(x_t|x_{t-1}^{(i)}).
\]

(5)
where $c$ is a normalising constant. The filtered density, as given above, can be updated recursively by propagating and updating the particle set using importance sampling techniques. The resulting algorithms are called particle filters, see [Doucet et al., 2000] and Cappé et al. (2007) for a review.

In this paper the particle approximations of the latent process are created with the auxiliary particle filter of [Pitt and Shephard, 1999]. This filter can be viewed as a general filter from which simpler filters are given as special cases (Fearnhead et al., 2008). The aim is to view the target (5) as defining a joint distribution on the particle at time $t$ and the value of a new particle at time $t$. The probability of sampling particle $x_{t-1}^{(i)}$ and $x_t$ is

$$cw_{t-1}^{(i)}g_\theta(y_t|x_t)f_\theta(x_t|x_{t-1}^{(i)}).$$

We approximate this with $\xi_t^{(i)}q(x_t|x_{t-1}^{(i)}, y_t, \theta)$, where $q(x_t|x_{t-1}^{(i)}, y_t, \theta)$ is a density function that can be sampled from and $\{\xi_t^{(i)}\}_{i=1}^N$ is a set of probabilities. This defines a proposal which we can simulate from by first choosing particle $x_{t-1}^{(i)}$ with probability $\xi_t^{(i)}$, and then, conditional on this, a new particle value, $x_t$, is sampled from $q(x_t|x_{t-1}^{(i)}, y_t, \theta)$. The weight assigned to our new particle is then

$$\tilde{w}_t = \frac{w_{t-1}^{(i)}g_\theta(y_t|x_t)f_\theta(x_t|x_{t-1}^{(i)})}{\xi_t^{(i)}q(x_t|x_{t-1}^{(i)}, y_t, \theta)}.$$

Details are summarised in Algorithm 1.

The optimal proposal density, in terms of minimising the variance of the weights [Doucet et al., 2000], is available when $q(x_t|x_{t-1}^{(i)}, y_t, \theta) = p(x_t|x_{t-1}^{(i)}, y_t, \theta)$ and $\xi_t^{(i)} \propto w_{t-1}^{(i)}p(y_t|x_{t-1}^{(i)})$. This filter is said to be fully adapted as all the weights $w_t^{(i)}$ will equal $1/N$. Generally it is not possible to sample from the optimal proposal, but alternative proposals can be used which approximate the fully adapted filter.

Algorithm 1 Auxiliary Particle Filter

Step 1: Iteration $t = 1$.
(a) For $i = 1, \ldots, N$, sample particles $\{x_1^{(i)}\}$ from the prior $p(x_1|\theta)$ and set $\tilde{w}_1^{(i)} = p(y_1|x_1^{(i)})$.
(b) Calculate $C_1 = \sum_{i=1}^N \tilde{w}_1^{(i)}$; set $\tilde{p}(y_1) = C_1/N$; and calculate normalised weights $w_1^{(i)} = \tilde{w}_1^{(i)}/C_1$ for $i = 1, \ldots, N$.

Step 2: Iteration $t = 2, \ldots, T$. Assume a user-defined set of proposal weights $\{\xi_t^{(i)}\}_{i=1}^N$ and family of proposal distributions $q(x_t|x_{t-1}^{(i)}, y_t, \theta)$.

(a) Sample $\{k_1, k_2, \ldots, k_N\}$ from $\{1, \ldots, N\}$ with probabilities $\xi_t^{(i)}$.
(b) Propagate particles $x_t^{(i)} \sim q(\cdot|x_{t-1}^{(k_i)}, y_t, \theta)$.
(c) Weight particles $\tilde{w}_t^{(i)} = \frac{w_{t-1}^{(k_i)}g_\theta(y_t|x_t^{(i)})f_\theta(x_t^{(i)}|x_{t-1}^{(k_i)})}{\xi_t^{(i)}q(x_t^{(i)}|x_{t-1}^{(k_i)}, y_t, \theta)}$ and calculate $C_t = \sum_{i=1}^N \tilde{w}_t^{(i)}$.
(d) Obtain an estimate of the predictive likelihood, $\tilde{p}(y_t|y_{1:t-1}, \theta) = C_t/N$, and calculate normalised weights $w_t^{(i)} = \tilde{w}_t^{(i)}/C_t$ for $i = 1, \ldots, N$.

One of the benefits of using the particle filter is that an estimate for the likelihood $p(y_{1:T}|\theta)$ is given for free from the particle filter output. We can estimate $p(y_t|y_{1:t-1}, \theta)$ by

$$\tilde{p}(y_t|y_{1:t-1}, \theta) = \frac{\sum_{i=1}^N \tilde{w}_t^{(i)}}{N}.$$

(6)
where, \( \tilde{w}_t^{(i)} \) are unnormalised weights. An unbiased estimate of the likelihood (Del Moral, 2004) is then

\[
\hat{p}(y_{1:T}|\theta) = \hat{p}(y_1|\theta) \prod_{t=2}^{T} \hat{p}(y_t|y_{1:t-1}, \theta).
\]

See Algorithm 1 and Pitt et al. (2012) and Del Moral (2004) for further details.

3 Particle MCMC

The auxiliary particle filter given in Algorithm 1 provides a positive, unbiased estimate of the likelihood based on the importance weights (6). Andrieu and Roberts (2009) and Andrieu et al. (2010) have shown how we can use such estimates in place of the likelihood function within MCMC. The idea is to run Algorithm 1 at each iteration of an MCMC algorithm to get an estimate of the likelihood at the current parameters. We then use this estimate instead of the true likelihood value within the accept-reject probability. If interest lies just in the posterior for the parameter, this results in the particle marginal Metropolis Hastings (PMMH) algorithm (see Algorithm 2). We will focus on this algorithm in the following (see Andrieu et al., 2010, for alternative particle MCMC algorithms).

**Algorithm 2 Particle Marginal Metropolis Hastings (PMMH) Algorithm**

*Step 1: Iteration \( j = 1 \).*

(a) Set \( \theta_1 \) arbitrarily.

(b) Run Algorithm 1 and compute the marginal likelihood \( \hat{p}(y_{1:T}|\theta_1) \) from the importance weights (6).

*Step 2: Iteration \( j = 2, \ldots, M \).*

(a) Sample \( \theta' \sim q(\cdot|\theta_{j-1}) \)

(b) Run Algorithm 1 and compute the marginal likelihood \( \hat{p}(y_{1:T}|\theta') \) from the importance weights (6).

(c) Set \( \theta_j = \theta' \) and \( \hat{p}(y_{1:T}|\theta_j) = \hat{p}(y_{1:T}|\theta') \)

with probability \( 1 \wedge \frac{\hat{p}(y_{1:T}|\theta_j)p(\theta_j)p(\theta')}{\hat{p}(y_{1:T}|\theta_{j-1})q(\theta'|\theta_{j-1})} \) else set \( \theta_j = \theta_{j-1} \) and \( \hat{p}(y_{1:T}|\theta_j) = \hat{p}(y_{1:T}|\theta_{j-1}) \).

A key result is that PMMH has \( \pi(\theta) \) as its stationary distribution (Andrieu and Roberts, 2009; Andrieu et al., 2010). To see this, let \( U \) denote the random variables used in the particle filter to generate the estimate of the likelihood, and \( p(U|\theta) \) their conditional density given \( \theta \). We can define a target distribution on the joint space \((\theta, U)\) as,

\[
\tilde{\pi}(\theta, U) \propto \hat{p}(y_{1:T}|\theta, U)p(U|\theta)p(\theta).
\]

It is straightforward to show that PMMH is a standard MCMC algorithm targeting (7) with a proposal \( q(\theta'|\theta)p(U|\theta') \), and that samples from the marginal target distribution \( \pi(\theta) \),

\[
\int \tilde{\pi}(\theta, U) dU \propto \int \hat{p}(y_{1:T}|\theta, U)p(U|\theta)p(\theta)dU = p(y_{1:T}|\theta)p(\theta),
\]

are given as a byproduct. The last equality follows from the fact that \( \hat{p}(y_{1:T}|\theta, U) \) is an unbiased estimator of the likelihood. Note that implementation of PMMH does not require storing all details of the particle filter just the resulting estimate of the likelihood.
Whilst PMMH admits $\pi(\theta)$ as the invariant density regardless of the variance of the likelihood estimator, the variance does affect the mixing properties of the algorithm; see Pitt et al. (2012) and Sherlock et al. (2014) for details. The choice of proposal distribution for the parameter will also have an important impact on the mixing properties of the algorithm. We now show that information from the particle filter can be used to guide this choice of proposal.

### 3.1 Efficient use of the particle filter output

Consider using some information from the particle filter, which we will denote $\mathcal{I}_\theta(\mathcal{U})$, within the proposal distribution. So if the current state of the Markov chain is $(\theta, \mathcal{U})$, our proposal will be $q(\theta' | \mathcal{U}, \mathcal{I}_\theta(\mathcal{U}))$. The acceptance probability of a new state $(\theta', \mathcal{U}')$ will then be

$$
\alpha(\theta', \mathcal{U}' | \theta, \mathcal{U}) = \min \left\{ 1, \frac{\hat{p}(y_{1:T} | \theta', \mathcal{U}') p(\theta') q(\theta | \theta', \mathcal{I}_\theta(\mathcal{U}'))}{\hat{p}(y_{1:T} | \theta, \mathcal{U}) p(\theta) q(\theta | \theta, \mathcal{I}_\theta(\mathcal{U}))} \right\}.
$$

It is straightforward to show that such an algorithm admits $\pi(\theta)$ as the invariant density.

**Proposition 3.1** Implementing PMMH with proposal distribution $q(\theta' | \mathcal{U}, \mathcal{I}_\theta(\mathcal{U}))$, and acceptance probability given by (8), gives an MCMC algorithm which admits $\pi(\theta)$ as the invariant density.

**Proof** As before the PMMH is a standard MCMC algorithm with $\hat{\pi}(\theta, \mathcal{U})$ as its invariant distribution, but now the proposal distribution is $q(\theta' | \mathcal{U}, \mathcal{I}_\theta(\mathcal{U}))$ as its invariant density. The acceptance probability for such an MCMC algorithm is

$$
\alpha(\theta', \mathcal{U}' | \theta, \mathcal{U}) = \min \left\{ 1, \frac{\hat{p}(y_{1:T} | \theta', \mathcal{U}') p(\mathcal{U}' | \theta') q(\theta | \theta', \mathcal{I}_\theta(\mathcal{U}')) p(\mathcal{U} | \theta)}{\hat{p}(y_{1:T} | \theta, \mathcal{U}) p(\mathcal{U} | \theta) q(\theta | \theta, \mathcal{I}_\theta(\mathcal{U}))} \right\}
$$

which simplifies to (8) as required.

Again, when implementing this version of PMMH we do not need to store all details of the particle filter. All we need is to store our estimate of the likelihood $\hat{p}(y_{1:T} | \theta, \mathcal{U})$ and the information $\mathcal{I}_\theta(\mathcal{U})$.

Our choice of information will be an estimate of the score, $\mathcal{I}_\theta(\mathcal{U}) = \nabla \log \hat{p}(y_{1:T} | \theta)$, where we give details of how to obtain such an estimate in the next section. We then use this estimate in place of the true score within a MALA proposal:

$$
\hat{q}(\theta' | \theta, \mathcal{I}_\theta(\mathcal{U})) = \mathcal{N} \left( \theta_{j-1} + \frac{\lambda^2}{2} \nabla \log \hat{\pi}(\theta), \lambda^2 \right),
$$

where $\nabla \log \hat{\pi}(\theta) = \nabla \log \hat{p}(y_{1:T} | \theta) + \nabla \log p(\theta)$, and $\lambda$ is the step-size parameter. The new proposal (9) can be used in place of $q(\theta' | \theta)$ in Algorithm 2 to give the particle MALA algorithm.

### 3.2 Particle approximations of the score vector

We can create a particle approximation of the score vector based on Fisher’s identity (Cappé et al. 2005)

$$
\nabla \log p(y_{1:T} | \theta) = \mathbb{E} [ \nabla \log p(x_{1:T}, y_{1:T} | \theta) | y_{1:T}, \theta ]
$$

(10)
which is the expectation of

\[ \nabla \log p(x_{1:T}, y_{1:T} | \theta) = \nabla \log p(x_{1:T-1}, y_{1:T-1} | \theta) + \nabla \log g_0(y_T | x_T) + \nabla \log f_0(x_T | x_{T-1}) \]

over the path \( x_{1:T} \).

The particle approximation to the score vector is obtained by replacing \( p(x_{1:T} | y_{1:T}, \theta) \) with a particle approximation \( \hat{p}(x_{1:T} | y_{1:T}, \theta) \). Here we outline this idea, but see Poyiadjis et al. (2011) for more details.

For each particle at a time \( t - 1 \), there is an associated path, defined by tracing the ancestry of each particle back in time. With slight abuse of notation denote this path by \( x^{(i)}_{1:t-1} \). We can thus associate with particle \( i \) at time \( t - 1 \) a value \( \alpha^{(i)}_{t-1} = \nabla \log p(x^{(i)}_{1:t-1}, y_{1:t-1} | \theta) \). These values can be updated recursively. Remember that in step 2(b) of Algorithm 1 we sample \( k_i \), which is the index of the particle at time \( t - 1 \) that is propagated to produce the \( i \)th particle at time \( t \). Thus we have

\[
\alpha^{(i)}_t = \alpha^{(k_i)}_{t-1} + \nabla \log g_0(y_t | x^{(i)}_t) + \nabla \log f_0(x^{(i)}_t | x^{(k_i)}_{t-1}). \tag{11}
\]

The problem with this approach is that the variance of the score estimate \( \nabla \log p(y_{1:t} | \theta) \) increases quadratically with \( t \) (Poyiadjis et al., 2011). Poyiadjis et al. (2011) suggest an alternative particle filter algorithm, which avoids a quadratically increasing variance but at the expense of a computational cost that is quadratic in the number of particles. Instead we will use the algorithm of Nemeth et al. (2013), which uses kernel density estimation and Rao-Blackwellisation to substantially reduce the Monte Carlo variance, but still maintains an algorithm whose computational cost is linear in the number of particles. See Poyiadjis et al. (2011), Ionides et al. (2011) and Dahlin et al. (2014) for some alternative approaches.

An outline of their approach is as follows. We first use kernel density estimation to replace each discrete \( \alpha^{(i)}_{t-1} \) value by a Gaussian distribution:

\[
\alpha^{(i)}_{t-1} \sim \mathcal{N}(m^{(i)}_{t-1}, V_{t-1}). \tag{12}
\]

The mean of this distribution is obtained by shrinking \( \alpha^{(i)}_{t-1} \) towards the mean of \( \alpha_{t-1} \),

\[
m^{(i)}_{t-1} = \zeta \alpha^{(i)}_{t-1} + (1 - \zeta) \sum_{i=1}^{N} w^{(i)}_{t-1} \alpha^{(i)}_{t-1}.
\]

Here \( 0 < \zeta < 1 \) is a user-defined shrinkage parameter. The idea of this shrinkage is that it corrects for the increase in variability introduced through the kernel density estimation of West (1993). For a definition of \( V_{t-1} \) see Nemeth et al. (2013), however its actual value does not affect the following details.

The resulting model for the \( \alpha_i \)’s, including their updates (11), is linear Gaussian. Hence we can use Rao-Blackwellisation to avoid sampling \( \alpha^{(i)}_t \), and instead calculate the parameters of the kernel (12) directly. This gives the following recursion for the means,

\[
m^{(i)}_t = \zeta m^{(k_i)}_{t-1} + (1 - \zeta) \sum_{i=1}^{N} w^{(i)}_{t-1} m^{(i)}_{t-1} + \nabla \log g_0(y_t | x^{(i)}_t) + \nabla \log f_0(x^{(i)}_t | x^{(k_i)}_{t-1}).
\]

The final score estimate depends only on these means, and is

\[
\nabla \log \hat{p}(y_{1:t} | \theta) = \sum_{i=1}^{N} w^{(i)}_t m^{(i)}_t.
\]
Algorithm 3 Rao-Blackwellised Kernel Density Estimate of the Score Vector

Add the following steps to Algorithm 1.

Step 1: (c) Set $\nabla \log \hat{p}(y_1|\theta) = \nabla \log g_\theta(y_1|x_1^{(i)}) + \nabla \log \mu_\theta(x_1^{(i)})$.

Step 2: (e) For $i = 1, \ldots, N$, calculate

$$m_t^{(i)} = \zeta m_{t-1}^{(k_i)} + (1 - \zeta) \sum_{i=1}^{N} \frac{w_t^{(i)} m_{t-1}^{(i)}}{N} + \nabla \log g_\theta(y_t|x_t^{(i)}) + \nabla \log f_\theta(x_t^{(i)}|x_{t-1}^{(k_i)})$$

(f) Update and store the score vector

$$\nabla \log \hat{p}(y_{1:t}|\theta) = \sum_{i=1}^{N} w_t^{(i)} m_t^{(i)}.$$

See Algorithm 3 for a summary. When $\zeta = 1$ the recursion simplifies to the method given by Poyiadjis et al. (2011), where the variance of the score estimate will increase quadratically with $t$. The use of a shrinkage parameter $\zeta < 1$ alleviates the degeneracy problems that affect the estimation of the score and significantly reduces the estimate’s variance. As a rule of thumb, setting $\zeta = 0.95$ produces reliable estimates (Nemeth et al., 2013). We shall use this tuning for all examples given in the Section 5.

4 High-dimensional limit results

For comparability with the existing literature on the analysis of high-dimensional Metropolis Hastings chains we shall use a change of notation for this section only. We shall now refer to our target density of dimension $n$ as $\pi_n(x_n)$, rather than $\pi(\theta)$.

In this section we obtain limiting forms for the acceptance rate and expected squared jumping distance (ESJD) for the particle MALA proposal. The ESJD has been used extensively as a measure of mixing of MCMC algorithms (e.g. Beskos et al., 2009; Sherlock and Roberts, 2009; Pasarica and Gelman, 2010; Sherlock, 2013), where maximising the ESJD is equivalent to minimising the first order auto-correlation of the Markov chain. We start by considering the idealised particle MALA which, for any given $x_n$, uses an unbiased stochastic estimate of the target density, but the exact value of the gradient of the log-target, $\nabla \log \pi_n(x_n)$. This algorithm is unlikely to be usable in practice, but provides a useful reference point for the more general particle MALA proposal, where we assume that we have a noisy and possibly biased estimate of $\nabla \log \pi_n(x_n)$. Introducing the possibility of both noise and bias in the estimate allows our results to be applied to a wider range of algorithms that could be used to estimate the gradient of the log-target.

Our assumptions on the form of the target, (13), (14) and (15) are the same as those in Roberts and Rosenthal (1998) and our Proposition A.1 resembles Lemma 1 from that article. We consider three different levels of control for the bias and variance of the errors in the estimate of each component of the gradient. For each level of control we then investigate the scaling that is necessary to achieve a non-degenerate limiting acceptance rate, and the behaviour of the efficiency function in that limit. A natural corollary of our analysis is that these scaling requirements, and the resulting general forms for the limiting acceptance rate and ESJD, would persist even if we were able to use an unbiased
estimate of each component of the gradient.

Our assumption on the distribution of the unbiased noise in the estimation of the target, (18), is the same as in Pitt et al. (2012) and is one of the key cases considered in Sherlock et al. (2014) and Doucet et al. (2014). As in each of these articles, the computational cost of the algorithm is assumed to have the form in (21).

4.1 Idealised particle MALA

Let \( x^n \) denote the \( i^{th} \) component an \( n \) dimensional vector \( x \) and let the target density be

\[
\pi^n(x^n) = \prod_{i=1}^{n} f(x^n_i). 
\]  

(13)

Let \( g(x) = \log f(x) \) and assume that \( g(x) \) and its derivatives \( g^{(i)}(x) \) satisfy

\[
|g(x)|, |g^{(i)}(x)| \leq M_0(x), 
\]  

(14)

for \( i = 1, \ldots, 8 \), where \( M_0(x) \) is some polynomial and that

\[
\int_{\mathbb{R}} x^k f(x) \, dx < \infty, \quad k = 1, 2, 3, \ldots. 
\]  

(15)

Let the scaling for the proposal on the target \( \pi^n \) be

\[
\lambda_n = \ell n^{-1/6}, 
\]  

(16)

where \( \ell > 0 \) is a tuning parameter. As mentioned earlier, for the idealised particle MALA we make the unrealistic assumption that the gradient in the log-posterior may be evaluated precisely so that the \( i^{th} \) component of the proposal is

\[
Y^n_i = x^n_i + \lambda_n Z_i + \frac{1}{2} \lambda_n^2 g'(x^n_i), 
\]  

(17)

with \( Z_i \sim \mathcal{N}(0, 1) \) \( (i = 1, \ldots, n) \) and independent of all other sources of variation.

We assume that the additive noise in the estimate of the log-target, \( V \), is Gaussian with a variance that is independent of the current or proposed position in the state space of the target, and the current noise, \( w \):

\[
V| x^n, y^n, w \sim \mathcal{N}\left(-\frac{1}{2} \sigma^2, \sigma^2\right), 
\]  

(18)

for some fixed \( \sigma > 0 \). It is straightforward to then show (e.g. Pitt et al., 2012) that when the chain is stationary,

\[
W \sim \mathcal{N}\left(\frac{1}{2} \sigma^2, \sigma^2\right). 
\]  

(19)

Let \( \alpha_n(x, w; y, v) \) be the acceptance probability for the idealised particle MALA with current value \( (x, w) \) and proposed value \( (y, v) \). We are interested in the expected acceptance rate and the expected squared jump distance,

\[
\overline{\alpha}_n(\ell, \sigma^2) := \mathbb{E} [\alpha_n(X^n, W; Y^n, V)], 
\]

\[
J_n(\ell, \sigma^2) = \mathbb{E} \left[ ||Y^n - X^n||^2 \alpha_n(X^n, W; Y^n, V) \right], 
\]

where expectation is over \( X^n, W, Y^n, V \) with distributions as defined in (13), (17) (for each component), (18) and (19). Our first result is as follows:
Theorem 4.1 As $n \to \infty$, the following limits hold in probability:

$$
\alpha_n(\ell, \sigma^2) \to \alpha(\ell, \sigma^2) := 2\Phi\left(-\frac{1}{2}\sqrt{\ell^6 K^2 + 2\sigma^2}\right) \quad \text{and} \quad n^{-2/3} J_n(\ell, \sigma^2) \to \ell^2 \alpha(\ell, \sigma^2),
$$

where

$$
K := \sqrt{\frac{1}{48} \mathbb{E} [5g'''(X)^2 - 3g''(X)^3] > 0}, \quad (20)
$$

and expectation is with respect to the density $f(x)$. Theorem 4.1 is proved in Appendix A.

We next consider the overall efficiency of the algorithm in terms of both ESJD and the computational cost. We assume that the computational cost (CPU time) is inversely proportional to the variance of the additive noise in the log-target, $T_{CPU} \propto 1/\sigma^2$; (21)

this is justified for a particle filter with a large number of particles acting on a large number of observations [Bérard et al., 2014]. Hence our measure of efficiency is, up to a constant of proportionality,

$$
\text{Eff}(\ell, \sigma^2) := \sigma^2 \ell^2 \Phi\left(-\frac{1}{2}\sqrt{\ell^6 K^2 + 2\sigma^2}\right). \quad (22)
$$

The following corollary details the parameters that optimise this efficiency.

Corollary 4.2 The efficiency function in equation (22) is maximised when the scaling and noise variance are respectively given by $\ell_{opt} \approx 1.125 K^{-1/3}$, and $\sigma_{opt}^2 \approx 3.038$, at which point the limiting acceptance rate is $\alpha_{opt} \approx 15.47\%$.

Proof Set $a^2 = K^2 \ell^6$ and $b^2 = 2\sigma^2$ then

$$
\text{Eff}(\ell, \sigma^2) \propto a^{2/3} b^2 \Phi\left(-\frac{1}{2}\sqrt{a^2 + b^2}\right).
$$

Given $a^2 + b^2$, $a^{2/3} b^2$ is maximised when $b^2 = 3a^2$, at which point the efficiency is proportional to $a^{8/3} \Phi(-a)$. Numerical optimisation shows that this function is maximised at $\hat{a} \approx 1.423$, and thus the optimal acceptance rate is $\hat{\alpha} = 2\Phi(-\hat{a}) \approx 15.47\%$. As a result, the optimal scaling and variance are $\ell_{opt} \approx 1.125 K^{-1/3}$ and $\sigma_{opt}^2 \approx 3.038$, as given in the statement.

Corollary 4.2 establishes that for the idealised particle MALA the optimal scaling rule is $\lambda \approx 1.125 K^{-1/3} n^{-1/6}$, contrasting with the optimal scaling for standard MALA, $\lambda \approx 0.825 K^{-1/3} n^{-1/6}$ [Roberts and Rosenthal, 1998]. The optimal variance of the noise in the log-target differs only slightly from the optimal variance when using the particle RWM, where $\sigma_{opt}^2 \approx 3.283$ [Sherlock et al., 2014]. However, it is important to note, that compared to the RWM, not only is the optimal asymptotic acceptance rate increased to 15.47% from 7.00%, but the scaling of the particle MALA is $O(n^{-1/6})$ rather than $O(n^{-1/2})$. This means that, for large $n$, particle MALA allows and accepts larger jumps and will therefore lead to a more efficient MCMC sampler.

11
Figure 1 shows a contour plot of the relative efficiency as a function of the scaling and the standard deviation, $\sigma$, of the noise (left panel), and a plot of the optimal acceptance rate as a function of $\sigma$ (right panel). The left-hand panel shows that for any given variance, the optimal scaling $\ell$ is close to $1.125K^{-1/3}$, and for any given scaling the optimal variance $\sigma^2$ is close to $3.038$. This relative insensitivity between the scaling and variance means that the scaling which maximises the expected squared jump distance will be close to the overall optimal scaling for a range of noise variances. The right-hand panel of Figure 1 gives the acceptance rate for a range of variances, assuming that we use the optimal $\ell$. From this we see that the optimal acceptance rate varies considerably over a range of ‘sensible’ noise variances. This suggests that, given a sensible, but not necessarily optimal noise variance, tuning to achieve an acceptance rate of about $15.47\%$ may lead to a relatively inefficient algorithm. Instead, one should either choose a scaling which optimises the effective sample size directly, or estimate the variance in the noise in the log-posterior, find the acceptance rate that corresponds to the optimal scaling conditional on the estimated variance, and tune to this.

4.2 Scaling conditions for particle MALA

In the particle MALA setting we do not have an exact estimate for the gradient of the log-target. In fact, depending on the approach used to estimate the gradient, the estimate may be both biased and noisy. In this section we give conditions on the bias and noise of the gradient estimate that would lead to an efficient particle MALA proposal.

We start by considering the $i$th component of the particle MALA proposal ($i = 1, \ldots, n$):

$$Y^n_i = x^n_i + \lambda Z_i + \frac{1}{2} \lambda^2 n \left( g(x^n_i) + \frac{1}{n^k} (b(x^n_i) + \tau U_x) \right),$$  \hspace{1cm} (23)

where, for all $i$, $Z_i \sim \mathcal{N}(0, 1)$ ($i = 1, \ldots, n$) and $U_x$ are independent of each other and of all other sources of variation. For any $x$, $U_x$ is a random variable with a distribution that is independent of $x$ and with $\mathbb{E}[U_x] = 0$, $\text{Var}[U_x] = 1$ and

$$\mathbb{E}\left[U_x^k\right] < \infty \ \forall \ k > 0.$$  \hspace{1cm} (24)
\( \tau U_x \) represents the noise in a particular gradient estimate at \( x \). Whilst the variance of the noise is kept fixed, we do allow the bias \( b(x_i^n) \) in the estimate of the \( i \)th component of the gradient (at \( x_i^n \)) to be position specific. Furthermore, we assume that \( b(x) \) and its derivatives \( b^{(i)}(x) \) (\( i = 1, \ldots, 7 \)) satisfy
\[
|b(x)|, |b^{(i)}(x)| \leq M_0(x), \quad (25)
\]
where \( M_0(x) \) is (without loss of generality) the same polynomial as in \((14)\).

The particle MALA proposal \((23)\) can be viewed as a generalisation of the MALA proposal, where the original MALA proposal can be retrieved by setting \( b(x) = \tau = 0 \). The bias and noise components of \((23)\) are scaled by an \( n^{-\kappa} \) term, where \( \kappa \geq 0 \). Without this control, as shall be shown in Part (i) of Theorem 4.3 in order to achieve a non-degenerate limiting acceptance rate, the scaling of the proposal must be chosen so that the particle MALA proposal has the same limiting behaviour as the particle random walk. Therefore, the conditions in Parts (ii) and (iii) of Theorem 4.3 that are imposed on the bias and noise terms of the particle MALA proposal are necessary for the algorithm to exhibit MALA-like limiting behaviour. In the following theorem, for ease of notation, we define
\[
K_i^2 := E_f [b(X)^2] + \frac{x^2}{2}, \quad (26)
\]
where, by assumptions \((15)\) and \((25)\), \( E_f [b(X)^2] < \infty \).

**Theorem 4.3** As \( n \to \infty \), the following limits hold in probability.

(i) If \( \kappa = 0 \), for a non-degenerate limiting acceptance rate \( \lambda_n = \ell n^{-1/2} \), whence
\[
\alpha_n(\ell, \sigma^2) \to \alpha^{(i)}(\ell, \sigma^2) := 2 \Phi \left( -\frac{1}{2} \sqrt{\ell^2 K_i^2 + 2 \sigma^2} \right) \quad \text{and} \quad n^{-1} J_n(\ell, \sigma^2) \to \ell^2 \alpha^{(i)}(\ell, \sigma^2),
\]

(ii) If \( \kappa = \frac{1}{3} \), for a non-degenerate limiting acceptance rate \( \lambda_n = \ell n^{-1/6} \), whence
\[
\alpha_n(\ell, \sigma^2) \to \alpha^{(ii)}(\ell, \sigma^2) := 2 \Phi \left( -\frac{1}{2} \sqrt{\ell^6 K_i^2 + \ell^2 K_i^2 + 2 \sigma^2} \right)
\]
and \( n^{-2/3} J_n(\ell, \sigma^2) \to \ell^2 \alpha^{(ii)}(\ell, \sigma^2) \), where \( K_i \) is defined in \((20)\),

(iii) If \( \kappa = \frac{1}{2} \), for a non-degenerate limiting acceptance rate \( \lambda_n = \ell n^{-1/6} \), whence
\[
\alpha_n(\ell, \sigma^2) \to \alpha^{(iii)}(\ell, \sigma^2) := 2 \Phi \left( -\frac{1}{2} \sqrt{\ell^6 K_i^2 + 2 \sigma^2} \right) \quad \text{and} \quad n^{-2/3} J_n(\ell, \sigma^2) \to \ell^2 \alpha^{(iii)}(\ell, \sigma^2).
\]

A proof of Theorem 4.3 is given in Appendix B where we note that condition (iii) of the theorem could be relaxed to \( \kappa > \frac{1}{3} \) at the expense of a considerably messier proof.

As in Section 4.1 we assume that the computational cost of estimating the log-target is inversely proportional to its Monte Carlo variance, \((21)\). Combining this assumption with the three distinct regimes of Theorem 4.3 leads to three distinct functions describing the overall efficiency, and hence to three different criteria for optimality.

**Corollary 4.4** In the limit as \( n \to \infty \) the overall efficiency is maximised when the scaling is \( \ell = \ell_{opt} \), the variance of the additive noise in the log-target is \( \sigma^2 = \sigma_{opt}^2 \), and the acceptance rate is \( \alpha = \alpha_{opt} \), where \( \ell_{opt} \), \( \sigma_{opt}^2 \) and \( \alpha_{opt} \) are as follows.
(i) If \( \kappa = 0 \), then \( \ell_{\text{opt}} \approx 2.562K_s^{-1} \), \( \sigma_{\text{opt}}^2 \approx 3.283 \) and \( \alpha_{\text{opt}} \approx 7.00\% \).

(ii) If \( \kappa = \frac{1}{3} \), the optimal variance and acceptance rate depend only on \( K_{\text{frac}} := K^2/(K^2 + K_s^6) \). In the limit as \( K_{\text{frac}} \to 0 \) the optimal values approach those in Part (i) of this corollary; in the limit as \( K_{\text{frac}} \to 1 \) the optimal values approach those in Part (iii) of this corollary; the behaviour of \( \ell^K := K^2\ell_{\text{opt}}^6 + K_s^2\ell_{\text{opt}}^2 \), \( \sigma_{\text{opt}}^2 \) and \( \alpha_{\text{opt}} \) as functions of \( K_{\text{frac}} \) is shown in Figure 2.

(iii) If \( \kappa = \frac{1}{2} \), then \( \ell_{\text{opt}} \approx 1.125K^{-1/3} \), \( \sigma_{\text{opt}}^2 \approx 3.038 \) and \( \alpha_{\text{opt}} \approx 15.47\% \).

Figure 2: Optimal values for \( \ell^K := K^2\ell_{\text{opt}}^6 + K_s^2\ell_{\text{opt}}^2 \), \( \sigma_{\text{opt}}^2 \) and \( \alpha_{\text{opt}} \) as functions of \( K_{\text{frac}} := K^2/(K^2 + K_s^6) \) for regime (ii) of Corollary 4.4.

Proof The acceptance rate in Part (i) of Theorem 4.3 leads to an efficiency function of the same form as that of the particle RWM that is studied in [Sherlock et al.] (2014); Part(i) of the corollary therefore follows directly from the results in that article. Part (iii) follows from Corollary 4.2. For Part (ii) the efficiency function is

\[
2\ell^2\sigma^2\Phi\left(-\frac{1}{2}\sqrt{\ell^6 K^2 + \ell^2 K_s^2 + 2\sigma^2}\right),
\]

and is maximised at some \((\ell_{\text{opt}}, \sigma_{\text{opt}})\). For any \( a \in \mathbb{R} \), a different scenario with constants \( \overline{K} = a^3K \) and \( \overline{K}_s = aK_s \) has an efficiency function proportional to

\[
2\overline{\ell}^2\sigma^2\Phi\left(-\frac{1}{2}\sqrt{\overline{\ell}^6 K^2 + \overline{\ell}^2 K_s^2 + 2\sigma^2}\right),
\]

where we have substituted \( \overline{\ell} = a\ell \). This is maximised at \((\overline{\ell}, \sigma) = (\ell_{\text{opt}}, \sigma_{\text{opt}})\), and so also possesses the same optimal acceptance rate. The optimal values for \( \sigma \) and \( \alpha \) therefore depend only on the ratio \( K/K_s^3 \). To obtain the results when \( K_{\text{frac}} \) is 0 or 1 we may therefore fix one of \( K \) or \( K_s \) and set the other to zero. The plots were obtained by numerical maximisation of the efficiency function.
Theorem 4.3 reveals some interesting features about the particle MALA proposal. Firstly, it is evident that some control of the bias and noise of the gradient estimate is essential. Without such control, the chain exhibits behaviour akin to that of the particle RWM, resulting in an optimal acceptance rate of 7%. The intuition behind this result is that if the bias and noise are not controlled as \( n \to \infty \), then these terms will dominate the estimate of the gradient, resulting in a noisy proposal similar to that in the RWM. If, however, the bias and noise components decrease in proportion to \( n^{\kappa} \) with \( \kappa \geq \frac{1}{3} \), then the proposal will again exhibit MALA-type behaviour, allowing the scaling of the proposal to decrease as \( n^{-1/6} \) rather than \( n^{-1/2} \).

Figure 2 shows that in the case with \( \kappa = \frac{1}{3} \) the optimal \( \ell^K \), variance and acceptance rate are continuous and monotonic functions of \( K \) which vary from the values that are optimal in the regime that resembles that of a particle RWM, to those which are optimal in case (iii) with \( \kappa = 1/2 \), which resembles the regime of the idealised particle MALA. Indeed, fixing \( K \) and letting \( K_n \downarrow 0 \) in case (ii) leads directly to case (iii). However fixing \( K \) and letting \( K_n \uparrow \infty \) does not lead directly to case (i) as, in the limit, the scaling of \( \lambda_n = \ell/n^{1/6} \) is inappropriate. Nevertheless since \( \ell^K \) is \( O(1) \) and, for large \( K \), \( \ell^K \sim \ell^K/K^2 \), as \( K_n \uparrow \infty \), \( \ell^K \downarrow 0 \) and so \( \lim_{n \to \infty} n^{-2/3} J_0(\ell, \sigma^2) \downarrow 0 \). For case (ii) we also see that over a large range of \( K \) (e.g. \( 0.1 < K < 0.9 \)), the optimal \( \ell^K \), variance and acceptance rate change slowly. For example, for a large range of relative values of the induced bias and noise in the gradient we see that the acceptance rate lies between 12% and 14%.

If we use the method outlined in Section 3.2 to estimate the gradient of the log-posterior, then to which of the three scenarios outlined in Theorem 4.3 do we expect our method to apply?

Under our assumption on the target (13) it would be natural to expect that condition (iii) of Theorem 4.3 would hold. This is because, for particle MCMC, we need to control the variance of the estimate of the posterior as \( n \) increases. This would require the number of particles used to estimate each component of (13) to increase linearly with \( n \) (this is the normal scaling of particle MCMC; see Andrieu et al. (2010), so that the Monte Carlo variance of the estimate of each term in the product (13) would be of order 1/n. Under this regime, the Monte Carlo error of the estimate of each component of the gradient would be of order \( n^{-1/2} \), which corresponds to \( \kappa = \frac{1}{2} \).

The method presented in Section 3.2 for estimating the gradient of the log-target is one of many possible approaches that can be applied. The general assumption that the gradient estimate is both biased and noisy means that the results given in Theorem 4.3 can be applied more generally to a variety of gradient estimate procedures (see Poyiadjis et al. (2011), Ionides et al. (2011) and Dahlin et al. (2014) for some alternatives). Furthermore, as a result of Theorem 4.3, using an unbiased gradient estimate within the particle MALA proposal does not eliminate the need for scaling, as the noise of the estimate still needs to be controlled to avoid a random walk type proposal. This raises an interesting question about how to best trade off between the bias and variance in the estimate of the gradient (see Olsson et al. (2008) for a discussion).

5 Simulation studies

In this section we provide simulation results to support the theory outlined in Section 4. In particular, we show that while our theory is based on the limit as the number of
possible to estimate the marginal likelihood $p(y_{1:T}|\theta)$ and score vector exactly with the Kalman filter [Durbin and Koopman 2001],

$$y_t = \alpha + \beta x_t + \tau \epsilon_t, \quad x_t = \mu + \phi x_{t-1} + \sigma \eta_t, \quad x_0 \sim \mathcal{N}(\mu/(1-\phi), \sigma^2/(1-\phi^2)),$$

where $\epsilon_t$ and $\eta_t$ are standard independent Gaussian random variables and $\theta = (\alpha, \beta, \tau, \mu, \phi, \sigma)^\prime$ is a vector of model parameters.

For this model it is possible to use the fully adapted particle filter [Pitt and Shephard 1999] using the optimal proposal for the latent states. Compared to the simpler bootstrap filter, this will reduce the variance of the likelihood estimate meaning that fewer particles are required to attain the same level of accuracy.

For this experiment we simulated 500 observations from the model with parameters $\theta = (0.2, 1, 1, 0.1, 0.9, 0.15)$. The particle MALA algorithm was run for 100,000 iterations, discarding the first half as burn-in. At each iteration, estimates of the likelihood and score vector were calculated from the particle filter (Alg. [1] and [3]) using $N \in \{200, 100, 70, 40, 20, 5, 10, 1\}$ particles. The parameters where then updated using the particle MALA proposal [9] with $\lambda^2 = \gamma^2 \times 1.125^2/6^{-1/3} \times \hat{V}$, where $\gamma \in \{0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2\}$ and $\hat{V}$ is the posterior covariance estimated from a pilot run. Note that increasing the number of particles leads to a reduction in $\sigma^2$.

The MCMC sampler was run assuming the following prior distributions

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0.3 \\ 1.2 \end{pmatrix}, \tau_\epsilon^2 \begin{pmatrix} 0.25 & 0 \\ 0 & 0.5 \end{pmatrix}\right), \quad \tau_\epsilon^2 \sim \mathcal{IG}(1, 7/20),$$

$\mu \sim \mathcal{N}(0.15, 0.5)$, $(\phi + 1)/2 \sim \text{Beta}(20, 5)$ and $\sigma^2 \sim \mathcal{IG}(2, 1/40)$, where $\mathcal{IG}$ is an inverse gamma distribution.

The parameters $(\phi, \sigma, \tau_\epsilon)$ are constrained such that $|\phi| < 1$, $\sigma > 0$ and $\tau_\epsilon > 0$. These parameters are transformed for the MCMC sampler as $\text{tanh}(\phi)$, $\log(\sigma)$ and $\log(\tau_\epsilon)$, noting that this transformation now introduces a Jacobian term into the MH acceptance ratio [2].

The efficiency of the particle MALA algorithm is assessed against various scalings $\gamma$ and noise $\sigma^2$ options. We use the effective sample size (ESS) as a measure of empirical efficiency and scale this by the computational time of the algorithm to give a measure of efficiency that approximates the theoretical measure of efficiency given in Corollary 4.2.

Figure [x] shows the efficiency of particle MALA for a range of scaling and noise options as detailed above. The left-hand panel gives the efficiency of particle MALA against various scalings $\gamma$ with a varying number of particles $N$, and the centre panel gives the efficiency against the number of particles for varying $\gamma$. From the left-hand panel it can be seen that, initially, increasing the number of particles leads to a more efficient MCMC sampler. However, beyond 20 particles the increase in computational cost outweighs the further improvement in mixing. Setting $N = 20$ results in a noisy posterior with $\sigma^2 \approx 2.6$, supporting Corollary 4.2, the optimal acceptance rate was $\approx 19\%$, slightly above the theoretical optimum. The left-hand panel of Figure [x] also shows an insensitivity of the optimal scaling to the noise variance, with efficiency maximised for $\gamma$ between 1 and 1.5 regardless of the number of particles; similarly the centre panel shows an insensitivity of the optimal variance to the scaling, with efficiency maximised for $\sigma^2$.
between 1.5 and 3, regardless of the scaling. Both of these insensitivities are predicted by the theory established in Section 4. The right-hand panel shows that the observed optimal acceptance rate as a function of $\sigma^2$ is also reasonably well predicted by the theory.

### 5.1 Mixture model of autoregressive experts

We now consider a real data example from the econometrics literature that was presented by Pitt et al. (2012). We use this example to illustrate the improvement of using the particle MALA (9) compared to particle RWM. Moreover, we show that using the method of Nemeth et al. (2013) to estimate the gradient, as outlined in Section 3.2, is more efficient than the algorithm proposed by Poyiadjis et al. (2011). This competing algorithm has been shown to have the nice property that, if the length of the data set increases linearly, the variance in the estimate of the gradient increases only linearly, as is the case with our algorithm (Alg. 3). However, while our algorithm has an $O(N)$ computational cost, where $N$ is the number of particles, the Poyiadjis et al. (2011) algorithm has a cost which is $O(N^2)$.

This example uses a two component mixture of experts model that is assumed to be observed with noise. Each of the experts is represented by a first order autoregressive process, where the mixing of the experts is probabilistic rather than deterministic. The model is defined as

$$
y_t = x_t + \tau_t \nu_t, \quad x_t = \psi_{J_t} + \phi_{J_t} x_{t-1} + \sigma_{J_t} \eta_t \quad J_t = 1, 2$$

with

$$
P(J_t = 1|x_{t-1}, x_{t-2}) = \frac{\exp(\xi_1 + \xi_2 x_{t-1} + \xi_3 (x_{t-1} - x_{t-2}))}{1 + \exp(\xi_1 + \xi_2 x_{t-1} + \xi_3 (x_{t-1} - x_{t-2}))},$$

where $\nu_t$ and $\eta_t$ are standard independent Gaussian random variables and there are 10 model parameters $\theta = (\tau, \psi_1, \psi_2, \phi_1, \phi_2, \sigma_1, \sigma_2, \xi_1, \xi_2, \xi_3)$.

Pitt et al. (2012) used the mixture of autoregressive experts to model the growth of US gross domestic product (GDP) from the second quarter of 1984 to the third quarter of 2010. This model follows previous observations that economic cycles display nonlinear and non-Gaussian features (Hamilton, 1989). By assuming measurement noise we are...
modelling the recorded GDP more accurately than by assuming no noise. This is because, as has been noted in the literature (Zellner, 1992), adjustments to GDP data (e.g. seasonal adjustments) between first and final release can change the data significantly. We impose the constraint $\psi_1(1 - \phi_1) < \psi_2(1 - \phi_2)$ to ensure that the mean of expert one is less than that of expert two. This implies that the first expert is identified as a low growth regime.

A particle filter approach to this problem is ideal if we assume measurement error for the GDP data. Standard MCMC methods could be applied on this model where the latent states are sampled conditional on the parameters and vice-versa (see Pitt et al. (2010)). However, this would cause the sampler to mix slowly for this model and would ultimately be less efficient than a particle MCMC implementation, whereby the latent states are integrated out. We compare particle MALA with the particle RWM implemented in Pitt et al. (2012). For both methods we implement a fully adapted particle filter, where 20 particles were used to give a variance of less than three.

We run the particle MCMC algorithm (Alg. 2) for 100,000 iterations, discarding the first half as burn-in, and use the random walk and particle MALA proposals. We scale the particle MALA proposal as in the previous example, where $\lambda^2 = 1.125^2/10^{-1/3} \times \hat{V}$ and $\hat{V}$ is an estimate of the posterior covariance taken from a pilot run. The random walk proposal is scaled as $\lambda^2 = 2.526^2/10 \times \hat{V}$. Each of the ten parameters has a Gaussian prior, where constrained parameters are transformed appropriately, and the hyper parameters are given in Pitt et al. (2012). Table 1 gives a comparison of the proposals: random walk, particle MALA and particle MALA using the Poyiadjis et al. (2011) $O(N^2)$ algorithm. The minimum and maximum effective sample size per CPU minute, taken over 10 independent simulations, is given for each of these proposals.

| Algorithm         | $\tau$ | $\psi_1$ | $\psi_2$ | $\phi_1$ | $\phi_2$ | $\sigma_1$ | $\sigma_2$ | $\xi_1$ | $\xi_2$ | $\xi_3$ |
|------------------|--------|----------|----------|----------|----------|-------------|-------------|-------|--------|--------|
| Particle RWM     | Min    | 3.39     | 2.96     | 1.65     | 2.15     | 1.96        | 1.38        | 2.16   | 2.54   | 2.05   |
|                  | Max    | 4.65     | 3.68     | 3.15     | 3.68     | 3.48        | 2.82        | 3.56   | 4.20   | 3.32   |
| Particle MALA $O(N)$ | Min    | 4.11     | 3.21     | 4.77     | 3.57     | 4.18        | 2.60        | 3.68   | 4.59   | 3.32   |
|                  | Max    | 5.12     | 5.71     | 6.37     | 6.12     | 6.43        | 5.47        | 6.22   | 7.34   | 7.02   |
| Poyiadjis $O(N^2)$ | Min    | 0.76     | 0.60     | 1.00     | 0.96     | 0.47        | 0.33        | 0.90   | 1.06   | 0.59   |
|                  | Max    | 1.25     | 1.19     | 1.37     | 1.35     | 1.35        | 1.17        | 1.26   | 1.82   | 1.23   | 0.96   |

Table 1: Effective sample size per CPU time given from MCMC output

The results from the MCMC simulation are summarised in Table 1. We can see that there is a significant improvement in terms of effective sample size when using the particle MALA algorithm compared to the random walk proposal. When using the Poyiadjis et al. (2011) $O(N^2)$ algorithm to calculate the gradient of the log-posterior, the effective sample size is approximately equal to that of particle MALA. However, when taking into account the computational cost, this proposal performs worse than the random walk. Therefore, if we take into account computational cost, then in order to benefit from using gradient information within the proposal, we need to be able to estimate the gradient of log-posterior with the same computational cost that is required to estimate the log-posterior.
6 Discussion

This paper presents theoretical and empirical results illustrating the improvement of the particle MALA proposal over the particle RWM when applied within the pseudo-marginal setting using a particle filter. Compared to the particle RWM, the particle MALA proposal improves the mixing of the Markov chain and increases the optimal acceptance rate from 7.00% to 15.47%. This is achieved with a small increase in computational cost that is required to estimate the gradient of the log-posterior. Our theoretical results establish an optimal scaling for the proposal and an optimal variance for the estimate of the log-posterior. These results provide useful guidelines for practitioners wishing to implement this class of proposal.

We have shown that it is possible to create MALA proposals using estimates of the log-posterior given by the $O(N)$ and $O(N^2)$ algorithms of Nemeth et al. (2013) and Poyiadjis et al. (2011), respectively. Compared to the $O(N^2)$ algorithm, our particle MALA algorithm offers significant improvements in terms of computational cost. The benefit of this computational saving is highly significant as the particle MALA algorithm can be executed with a larger number of particles, thus improving the approximation of the gradient estimate and also reducing the variance of the estimate of log-posterior.

Our results are the latest in a number of results looking at optimal implementations of particle MCMC algorithms (Pitt et al., 2012; Sherlock et al., 2014; Doucet et al., 2014). One important observation is that the optimal choice for the number of particles is similar for a range of different proposals. We found that the number of particles should be chosen so that the variance of the estimator of the log-likelihood was around 3.04, which is similar to, but slightly smaller than the value found in Sherlock et al. (2014) when using the particle RWM. Doucet et al. (2014) analysed bounds on the integrated autocorrelation time and suggested that for any Metropolis-Hastings algorithm the optimal variance should be between 0.85 and 2.82. Given that our assumption of constant variance (which is made in all of the above papers) is unlikely to hold in practice, and that the number of parameters is finite, we would recommend two possible tuning strategies: either evaluating the variance at several points in the main posterior mass and ensure that the largest of these is slightly lower than 3, then tune the scaling to achieve the acceptance rate given by Figure 1 or, for a sensible scaling, find the number of particles that optimises the overall efficiency (ESS/sec) then with this number of particles, find the scaling which optimises efficiency (ESS).

Acknowledgements

The authors would like to thank Prof. Jeff Rosenthal for providing Mathematica scripts from Roberts and Rosenthal (1998). This research was supported by EPSRC through grant EP/K014463 and the STOR-i Centre for Doctoral Training.

A Proof of Theorem 4.1

The proposal density for any given component is

$$ q(x, y) = \frac{1}{\lambda_n \sqrt{2\pi}} \exp \left[ -\frac{1}{2\lambda_n^2} \left( y - x - \frac{1}{2} \lambda_n^2 g'(x) \right) \right]. $$
Define
\[
R(x^n_i, Y^n_i) := \log \left\{ \frac{f(Y^n_i)q(Y^n_i, x^n_i)}{f(x^n_i)q((x^n_i, Y^n_i))} \right\},
\]
and \(T_n(X^n, Y^n) := \sum_{i=1}^n R(X^n_i, Y^n_i)\), so that the acceptance probability is
\[
\alpha_n(x^n, w; Y^n, V) = 1 \wedge \exp [V - w + T_n(x^n, Y^n)],
\]
where \(V\) and \(W\) are given in \((18)\) and \((19)\).

**Proposition A.1**
\[
R(x^n_i, Y^n_i) = C_3(x^n_i, Z_i)n^{-1/2} + C_4(x^n_i, Z_i)n^{-2/3} + C_5(x^n_i, Z_i)n^{-5/6}
\]
\[
+ C_6(x^n_i, Z_i)n^{-1} + C_7(x^n_i, Z_i, \lambda_n),
\]
where
\[
C_3(x^n_i, Z_i) = -\frac{1}{12} \ell^3 \left\{ 3Z_ig'(x^n_i)g''(x^n_i) + Z_i^2 g'''(x^n_i) \right\},
\]
and where \(C_4(x^n_i, Z_i), C_5(x^n_i, Z_i)\) and \(C_6(x^n_i, Z_i)\) are also polynomials in \(Z_i\) and the derivatives of \(g\). Furthermore, if \(\mathbb{E}_Z\) denotes expectation with \(Z \sim \mathcal{N}(0, 1)\) and \(\mathbb{E}_X\) denotes expectation with \(X\) having the density \(f(\cdot)\), then
\[
\mathbb{E}_X [\mathbb{E}_Z [C_3(X, Z)]] = \mathbb{E}_X [\mathbb{E}_Z [C_4(X, Z)]] = \mathbb{E}_X [\mathbb{E}_Z [C_5(X, Z)]] = 0,
\]
whereas
\[
\mathbb{E}_X [\mathbb{E}_Z [C_3(X, Z)^2]] = \ell^6 K^2 = -2\mathbb{E}_X [\mathbb{E}_Z [C_6(X, Z)]] > 0.
\]

Also
\[
\text{Var}[C_4(X, Z)] < \infty, \quad \text{Var}[C_5(X, Z)] < \infty, \quad \text{and} \quad \text{Var}[C_6(X, Z)] < \infty,
\]
where \(\text{Var}\) denotes variance over both \(Z\) and \(X\). Finally
\[
\mathbb{E}_Z [|C_7(x^n_i, Z_i, \lambda_n)|] \leq n^{-7/6} p(x^n_i),
\]
where \(p(x)\) is a polynomial in \(x\).

**Proof** As in Roberts and Rosenthal (1998), equation \((29)\) follows by Taylor expansion of \(g\) and its derivatives using MATHEMATICA (Wolfram, 2014) and collecting terms in powers of \(n\). Straightforward inspection shows that \(C_3\) has the claimed form and that \(C_4, C_5\) and \(C_6\) are also polynomials in \(Z\) and the derivatives of \(g\), as claimed. All terms in both \(C_3\) and \(C_5\) contain odd powers of \(Z\) and so their expectations are zero. Equation \((31)\), and the fact that the expectation of \(C_4\) is zero, follows after integrating by parts where expectations of products of the derivatives of \(g\) are being taken with respect to the density \(e^{g(x)}\). Equation \((32)\) follows from the polynomial form for \(C_4, C_5\) and \(C_6\) and assumptions \((14)\) and \((15)\).

Using the remainder formula of the Taylor series expansion we may derive the bound
\[
|C_7(x^n_i, Z_i, \lambda_n)| \leq n^{-7/6} p_*(x^n_i, w_i),
\]
for some polynomial \(p_*\), with \(|w_i| \leq |Z_i|\). But for any polynomial \(p_*(x, w) \leq A(1 + x^N)(1 + w^N)\), with a sufficiently large \(A\) and for a sufficiently large even integer \(N\), \((33)\) follows with \(p(x) = A\mathbb{E} \left[ (1 + Z^N) \right] (1 + x^N)\).
Proposition A.1 allows us to find the limiting distribution of one of the key terms in the acceptance probability of the algorithm when the Markov chain on \((X, W)\) is stationary, \((28)\).

**Lemma A.2**

\[ T_n(X^n, Y^n) \Rightarrow T \sim \mathcal{N}\left(-\frac{1}{2} \ell^6 K^2, \ell^6 K^2\right). \]

**Proof** First note that, by \((33)\),

\[
\mathbb{E}\left[\left|\sum_{i=1}^{n} C_7(X_i^n, Z_i, \lambda_n)\right|\right] \leq \sum_{i=1}^{n} \mathbb{E}[|C_7(X_i^n, Z_i, \lambda_n)|] \leq n^{-1/6}\mathbb{E}[p(X)].
\]

However \(\mathbb{E}[p(X)] < \infty\) by \((15)\) so, by Markov’s inequality, \(\sum_{i=1}^{n} C_7(X_i^n, Z_i, \lambda_n) \to 0\).

By Slutsky’s Theorem it is therefore sufficient to show that \(T'_n \to 0\), where

\[
T'_n \defeq \sum_{i=1}^{n} \{ C_3(X_i^n, Z_i)n^{-1/2} + C_4(X_i^n, Z_i)n^{-2/3} + C_5(X_i^n, Z_i)n^{-5/6} + C_6(X_i^n, Z_i)n^{-1}\}.
\]

Combining \((30), (31)\) and \((32)\),

\[
\mathbb{E}[T'_n] = -\frac{1}{2} \ell^6 K^2, \\
\text{Var}[T'_n] = \text{Var}[C_3(X, Z)] + O(n^{-1/6}) \to \ell^6 K^2
\]

in probability. Moreover \(T'_n\) is the sum of \(n\) independent and identically distributed terms, so the result follows by the Central Limit Theorem.

The distribution of the other key term in \((28)\) follows directly from \((18)\) and \((19)\):

\[ V - W \sim \mathcal{N}(-\sigma^2, 2\sigma^2). \] (34)

Thus

\[ T + V - W \Rightarrow \mathcal{N}\left(-\sigma^2 + \frac{1}{2} \ell^6 K^2, 2\sigma^2 + \ell^6 K^2\right). \] (35)

Now if \(U \sim \mathcal{N}(a, b^2)\) then \(\mathbb{E}[\min(1, e^U)] = \Phi(a/b) + e^{a+b^2/2}\Phi(-b - a/b)\) (e.g. Roberts et al. \(1997\)). Since \(\alpha_n = \mathbb{E}[\min(1, e^{T_n+V-W})]\), we may apply the Bounded Convergence Theorem to see that

\[
\lim_{n \to \infty} \alpha_n = \mathbb{E}[\min(1, e^{T+V-W})] = 2\Phi\left(-\frac{1}{2} \sqrt{\ell^6 K^2 + 2\sigma^2}\right),
\]

proving the first part of Theorem 4.1.

To prove the second result, we first note the following

**Proposition A.3**

\[
\lim_{n \to \infty} \mathbb{E}\left[n^{-2/3} ||Y^n - X^n||^2\right] = \ell^2, \\
\lim_{n \to \infty} \mathbb{E}\left((n^{-2/3} ||Y^n - X^n||^2 - \ell^2)^2\right) = 0.
\]
Proof To simplify the exposition we suppress the superscripts, \( n \), in \( X^n \) and \( Y^n \). Firstly, \[
-\frac{2}{3} \mathbb{E} \left[ |Y - X|^2 \right] = n^{\frac{1}{3}} \mathbb{E} \left[ (Y_1 - X_1)^2 \right] \\
= n^{\frac{1}{3}} \mathbb{E} \left[ \left( \ell n^{-1/6} Z_1 + \frac{1}{2} \ell^2 n^{-1/3} g'(X_1) \right)^2 \right] \\
= \ell^2 + \frac{1}{4} \ell^4 n^{-1/3} \mathbb{E} \left[ g'(X_1)^2 \right].
\]

By assumptions (14) and (15), \( \mathbb{E} \left[ g'(X_1)^2 \right] < \infty \) and the first result follows. Also as \( n \to \infty \), \[
\text{Var} \left[ n^{-\frac{2}{3}} |Y - X|^2 - \ell^2 \right] = n^{-\frac{4}{3}} \text{Var} \left[ \sum_{i=1}^{n} (Y_1 - X_1)^2 \right] \\
= n^{-\frac{1}{3}} \text{Var} \left[ \left( \ell n^{-1/6} Z_1 + \frac{1}{2} \ell^2 n^{-1/3} g'(X_1) \right)^2 \right] \to 0,
\]

by (14) and (15). This, combined with the first part of this proposition proves the second part.

To complete the proof, we abbreviate \( \alpha_n(X^n, W; Y^n, V) \) to \( A_n \). Now \[
\mathbb{E} \left[ n^{-\frac{2}{3}} |Y^n - X^n|^2 A_n \right] - \ell^2 A \leq \mathbb{E} \left[ (n^{-\frac{2}{3}} |Y^n - X^n|^2 - \ell^2) A \right] + \mathbb{E} \left[ \ell^2 (A_n - A) \right].
\]

The second term on the right hand side converges to zero by the first part of Theorem 4.1. The Cauchy-Schwarz inequality bounds the first term on the right hand side by \[
\mathbb{E} \left[ (n^{-\frac{2}{3}} |Y^n - X^n|^2 - \ell^2)^2 \right]^{1/2} \mathbb{E} \left[ A_n^2 \right]^{1/2}.
\]

The first term converges to zero by Proposition A.3 and the second term is bounded.

B Proof of Theorem 4.3

For the sake of brevity, we shall prove statements (i), (ii) and (iii) of Theorem 4.3 together rather than separately. Throughout the proof, therefore, the superscript * will be used to denote a superscript that could be replaced by \((i), (ii)\) or \((iii)\) respectively according to the case in the statement of Theorem 4.1 that is being considered.

For \( * \in \{(i), (ii), (iii)\} \) let \( R^*(x^n_i, y^n_i) \) be the log Metropolis Hastings ratio when the proposal, \( q^*(x^n_i, y^n_i) \), is the particle MALA proposal given in (23), and let \[
T^*_n(X^n, Y^n) = \sum_{i=1}^{n} R^*(X^n_i, Y^n_i).
\]

We also define \( U_i = (U_{x_i}, U_{y_i}) \), to be the vector of (zero mean and unit variance) noise terms in the \( i \)th component of the gradient estimate used, respectively, in the particle MALA proposal from the current value and the proposal for the corresponding reverse move from the proposed value.
Remark In the analysis of the idealised particle MALA in Appendix \ref{sec:ideal-particle-MALA}, the terms $C_3, \ldots, C_6$ arise respectively from the 3rd, \ldots, 6th terms in Taylor expansions in $\lambda_n$ of $g$ and $g'$ about $x_i^n$, the first two terms being zero. A general analysis of particle MALA would need to consider the double Taylor expansions in both $\lambda_n$ and $n^{-k}$. Since the usual MALA scaling is $\lambda_n \propto n^{-1/6}$ we simplify the exposition considerably by only examining cases where $\kappa$ is a multiple of $\frac{1}{3}$. Here, therefore, $C_k^* (k = 3, \ldots, 6)$ will simply refer to a coefficient multiplying $n^{-k/6}$.

Remark In the idealised particle MALA setting (Proposition \ref{prop:ideal-particle-MALA}), the leading term, $C_3$, is $O(n^{-1/2})$, the expectations of all terms of a lower order than $n^{-1}$ are zero, the expectation of the $O(n^{-1})$ term is non-zero and the expectation of the remainder term is bounded. In order to have a non-degenerate acceptance rate for the particle MALA proposal similar conditions on the terms in $R^*(x_i^n, y_i^n)$ must be satisfied. If the standard MALA scaling is used, then with no control over the bias and noise of the gradient estimate (i.e. with $\kappa = 0$) the leading term is of $O(\lambda_n)$; as we shall see, setting $\lambda_n = \ell n^{-1/6}$ in fact ensures that all of the conditions listed above are satisfied. Alternatively, if we fix $\lambda_n = \ell n^{-1/6}$, then to have a non-degenerate acceptance rate we must set $\kappa \geq \frac{1}{3}$.

The proof commences with an analogous result to Proposition \ref{prop:ideal-particle-MALA} from Appendix \ref{sec:ideal-particle-MALA}

Proposition B.1

\begin{equation}
R^*(x_i^n, y_i^n) = C_3^*(x_i^n, U_i, Z_i) n^{-1/2} + C_4^*(x_i^n, U_i, Z_i) n^{-2/3} + C_5^*(x_i^n, U_i, Z_i) n^{-5/6} \tag{37}
+ C_6^*(x_i^n, U_i, Z_i) n^{-1} + C_7^*(x_i^n, U_i, Z_i, \lambda_n),
\end{equation}

where the terms $C_3^*, \ldots, C_7^*$ possess the following properties.

\begin{align*}
C_3^{(i)}(x_i^n, U_i, Z_i) & = -\frac{1}{2} \ell \tau U x_i^n Z_i - \frac{1}{2} \ell \tau U y_i^n Z_i - b(x_i^n) \ell Z_i, \\
C_3^{(ii)}(x_i^n, U_i, Z_i) & = C_3(x_i^n, Z_i) - \frac{1}{2} \ell \tau U x_i^n Z_i - \frac{1}{2} \ell \tau U y_i^n Z_i - b(x_i^n) \ell Z_i, \\
C_3^{(iii)}(x_i^n, U_i, Z_i) & = C_3(x_i^n, Z_i),
\end{align*}

and where $C_3(x_i^n, Z_i)$ is the MALA term from Proposition \ref{prop:ideal-particle-MALA}

For $* \in \{ (i), (ii), (iii) \}$, $C_4^*, C_5^*$ and $C_6^*$ are all polynomials in $Z$, in derivatives of $g(x)$, and in $b(x)$ and its derivatives. Furthermore, let $E_{U,Z}$ denotes expectation with respect to $Z \sim N(0,1)$, and with respect to $U_x$ and $U_y$ with $E[U_x] = E[U_y] = 0$ and $\text{Var}[U_x] = \text{Var}[U_y] = 1$, and let $E_X$ denote expectation with $X$ having the density $f(\cdot)$. Then

\begin{equation}
E_X \left[ E_{U,Z} \left[ C_3^*(X, U, Z) \right] \right] = E_X \left[ E_{U,Z} \left[ C_4^*(X, U, Z) \right] \right] = E_X \left[ E_{U,Z} \left[ C_5^*(X, U, Z) \right] \right] = 0. \tag{38}
\end{equation}

\begin{equation}
E_X \left[ E_{U,Z} \left[ C_3^*(X, U, Z)^2 \right] \right] = -2E_X \left[ E_{U,Z} \left[ C_4^*(X, U, Z)^2 \right] \right] > 0
\end{equation}

with,

\begin{align*}
E_X \left[ E_{U,Z} \left[ C_3^{(i)}(X, U, Z)^2 \right] \right] & = \ell^2 K_i^2 =: a^{(i)} \\
E_X \left[ E_{U,Z} \left[ C_3^{(ii)}(X, U, Z)^2 \right] \right] & = \ell^6 K^2 + \ell^2 K_i^2 =: a^{(iv)} \\
E_X \left[ E_{U,Z} \left[ C_3^{(iii)}(X, U, Z)^2 \right] \right] & = \ell^6 K^2 =: a^{(iii)}.
\end{align*}

23
where $K_*^2$ is defined in (26). Also

$$
\text{Var}[C_4^*(X, U, Z)] < \infty, \quad \text{Var}[C_5^*(X, U, Z)] < \infty, \quad \text{and} \quad \text{Var}[C_6^*(X, U, Z)] < \infty,
$$

where $\text{Var}$ denotes variance over $Z$, $U$ and $X$. Finally

$$
\mathbb{E}_{U,Z} [|C_7(x_i^n, U_i, Z_i, \lambda_n)|] \leq n^{-7/6}p(x_i^n),
$$

where $p(x)$ is a polynomial in $x$.

**Proof** As with Proposition A.1, the polynomial forms for $C_3^*, \ldots, C_6^*$ are produced using MATHEMATICA (Wolfram, 2014), but this time by also Taylor expanding the term $b(y)$ in $y - x$.

Clearly $\mathbb{E}[C_3^*] = 0$ as $\mathbb{E}[C_3] = 0$ and the other terms are multiples of $U_x$, $U_y$ and odd powers of $Z$. The same argument can be used for the expectations of $C_4^*$ and $C_5^*$ and the relationship between $\mathbb{E}[C_3^2]$ and $\mathbb{E}[C_6]$, however it must be used in tandem with integration (by parts) with respect to the target density ($e^{\varrho(x)}$) and using assumptions [14], [15] and [25]. We illustrate this by providing the form for $C_4^*$; the forms and calculations for the other two relationships are straightforward but messier. With $\kappa = 0$ we have $C_4^{(i)} = 0$ (indeed $C_5^{(i)} = 0$ also). With $\kappa = \frac{1}{3}$ and $\kappa = \frac{1}{2}$, we have, respectively,

$$
C_4^{(i)}(X, U, Z) = C_4 - \frac{1}{2} \ell^2 b'(X)Z^2 - \frac{1}{2} \ell^2 b(X)g'(X) - \frac{1}{2} \ell^2 \tau U_y g'(x),
$$

$$
C_4^{(ii)}(X, U, Z) = C_4 - \tau b(X)Z - \frac{1}{2} \ell \tau U_x Z - \frac{1}{2} \ell \tau U_y Z.
$$

(42)

However [14] and [25] imply that $\int b'(x)e^{\varrho(x)} \, dx = -\int b(x)g'(x)e^{\varrho(x)}$.

The final two parts of the proposition follow from analogous arguments to those used in Proposition A.1 provided (24) holds.

**Remark** The bias and noise terms that are present in $C_3^{(ii)}$ are absent from $C_3^{(iii)}$; unsurprisingly they instead appear in $C_4^{(iii)}$ (see (42)). In fact, for $\kappa > \frac{1}{3}$, these extra terms must be of a higher order than $n^{-1/2}$.

Proposition B.1 allows us to state an analogous result to Lemma A.2; the proof is very similar to that of Lemma A.2 and so is omitted.

**Lemma B.2** For $* \in \{(i), (ii), (iii)\}$

$$
T_*^n(X^n, Y^n) \Rightarrow T \sim \mathcal{N}\left(-\frac{1}{2}a^*, a^*\right),
$$

where $T^*$ is defined in (36) and $a^{(i)}$, $a^{(ii)}$ and $a^{(iii)}$ are defined in (39).

The proof for the asymptotic acceptance rate, $\tau_n(\ell, \sigma^2) \rightarrow \alpha^*(\ell, \sigma^2)$, is completed using Lemma B.2 as in the proof of Theorem 4.1 by accounting for the distribution of the noise of the log-target.

Finally, as in the proof of Theorem 4.1, the limit for the squared jump distance, $J_n(\ell, \sigma^2)$, follows from Proposition A.3.
References

Andrieu, C., Doucet, A., and Holenstein, R. (2010). Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342.

Andrieu, C. and Roberts, G. O. (2009). The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics*, 37(2):697–725.

Beaumont, M. a. (2003). Estimation of population growth or decline in genetically monitored populations. *Genetics*, 164(3):1139–60.

Bérard, J., Del Moral, P., and Doucet, A. (2014). A lognormal central limit theorem for particle approximations of normalizing constants. *Electronic Journal of Probability*, 19:1–28.

Beskos, A., Roberts, G., and Stuart, A. (2009). Optimal scalings for local Metropolis-Hastings chains on non-product targets in high dimensions. *Annals of Applied Probability*, 19(3):863–898.

Cappé, O., Godsill, S., and Moulines, E. (2007). An Overview of Existing Methods and Recent Advances in Sequential Monte Carlo. *Proceedings of the IEEE*, 95(5):899–924.

Cappé, O., Moulines, E., and Ryden, T. (2005). *Inference in Hidden Markov Models*. Springer Series in Statistics. Springer.

Dahlin, J., Lindsten, F., and Schönh, T. B. (2013). Particle Metropolis Hastings Using Langevin Dynamics. In *Proceedings of the 38th International Conference on Acoustics and Speech and Signal Processing*.

Dahlin, J., Lindsten, F., and Schönh, T. B. (2014). Particle Metropolis Hastings using gradient and Hessian information. *Statistics and Computing*.

Del Moral, P. (2004). *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*. Probability and Its Applications. Springer.

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38.

Doucet, A., Godsill, S., and Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and computing*, 10(3):197–208.

Doucet, A., Pitt, M., Deligiannidis, G., and Kohn, R. (2014). Efficient implementation of Markov chain Monte Carlo when using an unbiased likelihood estimator. *arXiv preprint arXiv:1210.1871v4*.

Durbin, J. and Koopman, S. (2001). *Time Series Analysis by State Space Methods*. Oxford Statistical Science Series. Oxford University Press.

Fearnhead, P. (2011). MCMC for state space models. In Brooks, S., Gelman, A., Jones, G. L., and Meng, X.-L., editors, *Handbook of Markov Chain Monte Carlo*, pages 513–529. Chapman and Hall.
Fearnhead, P., Papaspiliopoulos, O., and Roberts, G. O. (2008). Particle filters for partially observed diffusions. *Journal of the Royal Statistical Society, Series B*, 70:1–28.

Golightly, A. and Wilkinson, D. J. (2011). Bayesian parameter inference for stochastic biochemical network models using particle Markov chain Monte Carlo. *Interface focus*, 1(6):807–20.

Hamilton, J. (1989). A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica*, (57):357–384.

Ionides, E. L., Bhadra, A., Atchadé, Y., and King, A. (2011). Iterated filtering. *The Annals of Statistics*, 39(3):1776–1802.

Knape, J. and de Valpine, P. (2012). Fitting complex population models by combining particle filters with Markov chain Monte Carlo. *Ecology*, 93(2):256–263.

Nemeth, C., Fearnhead, P., and Mihaylova, L. (2013). Particle approximations of the score and observed information matrix for parameter estimation in state space models with linear computational cost. *arXiv preprint arXiv:1306.0735*.

Olsson, J., Cappé, O., Douc, R., and Moulines, E. (2008). Sequential Monte Carlo smoothing with application to parameter estimation in nonlinear state space models. *Bernoulli*, 14(1):155–179.

Pasarica, C. and Gelman, A. (2010). Adaptively scaling the Metropolis algorithm using expected squared jumped distance. *Statistica Sinica*, 20:343–364.

Pitt, M., Giordani, P., and Kohn, R. (2010). Bayesian inference for time series state space models. In Geweke, J., Koop, G., and van Dijk, H., editors, *The Oxford Handbook of Bayesian Econometrics*. Oxford University Press.

Pitt, M. K. (2002). Smooth Particle Filters for Likelihood Evaluation and Maximisation.

Pitt, M. K. and Shephard, N. (1999). Filtering via Simulation: Auxiliary Particle Filters. *Journal of the American Statistical Association*, 94(446):590–599.

Pitt, M. K., Silva, R., Giordani, P., and Kohn, R. (2012). On some properties of Markov chain Monte Carlo simulation methods based on the particle filter. *Journal of Econometrics*, 171(2):134–151.

Poyiadjis, G., Doucet, A., and Singh, S. S. (2011). Particle approximations of the score and observed information matrix in state space models with application to parameter estimation. *Biometrika*, 98(1):65–80.

Roberts, G. O., Gelman, A., and Gilks, W. (1997). Weak Convergence and Optimal Scaling of the Random Walk Metropolis Algorithms. *The Annals of Applied Probability*, 7(1):110–120.

Roberts, G. O. and Rosenthal, J. S. (1998). Optimal scaling of discrete approximations to Langevin diffusions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(1):255–268.
Roberts, G. O. and Rosenthal, J. S. (2001). Optimal scaling for various Metropolis-Hastings algorithms. *Statistical Science*, 16(4):351–367.

Sherlock, C. (2013). Optimal scaling of the random walk Metropolis: general criteria for the 0.234 acceptance rule. *J. App. Prob.*, 50(1):1–15.

Sherlock, C. and Roberts, G. (2009). Optimal scaling of the random walk Metropolis on elliptically symmetric unimodal targets. *Bernoulli*, 15(3):774–798.

Sherlock, C., Thiery, A. H., Roberts, G. O., and Rosenthal, J. S. (2014). On the efficiency of pseudo-marginal random walk Metropolis algorithms. *Annals of Statistics*.

West, M. (1993). Approximating posterior distributions by mixture. *Journal of the Royal Statistical Society. Series B*, 55(2):409–422.

Wolfram, S. (2014). *Mathematica 10.0*. Champaign, Illinois.

Zellner, A. (1992). Commentary. In Belagia, M. and Garfinkel, M., editors, *The Business Cycle: Theories and Evidence: Proceedings of the Sixteenth Annual Economic Policy Conference of the Reserve Bank of St Louis*. 