The Use of a Single Pseudo-Sample in Approximate Bayesian Computation

Luke Bornn∗1, Natesh S. Pillai†1, Aaron Smith‡2, and Dawn Woodard§3

1Department of Statistics, Harvard University
2Department of Mathematics and Statistics, University of Ottawa
3School of Operations Research and Information Engineering, Cornell University

April 6, 2015

Abstract

We analyze the computational efficiency of approximate Bayesian computation (ABC), which approximates a likelihood function by drawing pseudo-samples from the associated model. For the rejection sampling version of ABC, it is known that multiple pseudo-samples cannot substantially increase (and can substantially decrease) the efficiency of the algorithm as compared to employing a high-variance estimate based on a single pseudo-sample. We show that this conclusion also holds for a Markov chain Monte Carlo version of ABC, implying that it is unnecessary to tune the number of pseudo-samples used in ABC-MCMC. This conclusion is in contrast to particle MCMC methods, for which increasing the number of particles can provide large gains in computational efficiency.

1 Introduction

Approximate Bayesian computation (ABC) is a family of algorithms for Bayesian inference that address situations where the likelihood function is intractable to evaluate but where one can obtain samples from the model. These methods are now widely used in population genetics, systems biology, ecology, and other areas, and are implemented in an array of popular software packages (Tavare et al., 1997; Marin et al., 2012). Let \( \theta \in \Theta \) be the parameter of interest with prior density \( \pi(\theta) \), \( y_{\text{obs}} \in Y \) be the observed data and \( p(y|\theta) \) be the model. The simplest ABC algorithm first generates a sample \( \theta' \sim \pi(\theta) \) from the prior, then generates a pseudo-sample \( y_{\theta'} \sim p(\cdot|\theta') \). Conditional on \( y_{\theta'} = y_{\text{obs}} \), the distribution of \( \theta' \) is the posterior distribution \( \pi(\theta|y_{\text{obs}}) \propto \pi(\theta)p(y_{\text{obs}}|\theta) \). For all but the most trivial discrete problems, the probability that \( y_{\theta'} = y_{\text{obs}} \) is either zero or very small. Thus the condition of exact matching of pseudo-data to the observed data is typically relaxed to \( \| \eta(y_{\text{obs}}) - \eta(y_{\theta'}) \| < \epsilon \), where \( \eta \) is a low-dimensional summary statistic, \( \| \cdot \| \) is a distance function, and \( \epsilon > 0 \) is a tolerance level. The resulting algorithm gives samples from the target density \( \pi_\epsilon \) that is proportional to \( \left[ \pi(\theta) \int 1_{\{\| \eta(y_{\text{obs}}) - \eta(y) \| < \epsilon \}} p(y|\theta) \, dy \right] \). If the tolerance \( \epsilon \) is small enough and the statistic(s) \( \eta \) good enough, then \( \pi_\epsilon \) can be a good approximation to \( \pi(\theta|y_{\text{obs}}) \).

∗bornn@stat.harvard.edu
†pillai@stat.harvard.edu
‡asmif28@uottawa.ca
§http://people.orie.cornell.edu/woodard
A generalized version of this method (Wilkinson 2013) is given in Algorithm 1, where \( K(\eta) \) is an unnormalized probability density function, \( M \) is an arbitrary number of pseudo-samples, and \( c \) is a constant satisfying \( c \geq \sup_\eta K(\eta) \).

\[
\text{for } t = 1 \text{ to } n \text{ do} \\
\quad \text{repeat} \\
\quad \quad \text{Generate } \theta' \sim \pi(\theta), \ y_i \sim p(y|\theta') \text{ for } i = 1, \ldots, M, \text{ and } u \sim \text{Uniform}[0,1]. \\
\quad \text{until } u < \frac{1}{cM} \sum_{i=1}^{M} K(\eta(y_{obs}) - \eta(y_i, \theta')); \\
\quad \text{Set } \theta(t) = \theta'. \\
\]

\textbf{Algorithm 1: Generalized ABC}

Using the “uniform kernel” \( K(\eta) = 1_{\{||y||<\epsilon\}} \) and taking \( M = c = 1 \) yields the version described above. Algorithm 1 yields samples \( \theta(t) \) from the kernel-smoothed posterior density

\[
\pi_K(\theta|y_{obs}) \propto \pi(\theta) \int K(\eta(y_{obs}) - \eta(y)) \ p(y|\theta) \ dy. 
\] (1)

Although Algorithm 1 is nearly always presented in the special case with \( M = 1 \) (Wilkinson 2013), it is easily verified that \( M > 1 \) still yields samples from \( \pi_K(\theta|y_{obs}) \).

Many variants of Algorithm 1 exist. Algorithm 2 is a Markov chain Monte Carlo (MCMC) version, constructed using essentially any transition kernel \( q \) on \( \Theta \) (Marjoram et al., 2003; Wilkinson, 2013; Andrieu and Vihola, 2014); it uses the nonnegative, unbiased estimator

\[
\hat{\pi}_{K,M}(\theta|y_{obs}) \equiv \pi(\theta) \frac{1}{M} \sum_{i=1}^{M} K(\eta(y_{obs}) - \eta(y_i, \theta)) 
\] (2)

of (an unnormalized version of) \( \pi_K(\theta|y_{obs}) \). Despite the fact that the approximation (2) depends on \( M \), as \( n \to \infty \) the distribution of \( \theta^{(n)} \) in Algorithm 1 converges to the same distribution \( \pi_K \) under mild conditions (Andrieu and Vihola, 2014). Although the choice of \( M \) generally does not affect the limiting distribution of \( \theta^{(n)} \), it does affect the evolution of the stochastic process \( \{\theta^{(n)}\}_{n \in \mathbb{N}} \). Under mild conditions on \( K \) and \( p(y|\theta) \) (cf. Wied and Weißbach, 2012), the kernel density estimator (2) converges uniformly. When the estimator converges uniformly, as \( M \to \infty \) the finite sample paths \( \theta^{(1)}, \ldots, \theta^{(n)} \) obtained from Algorithm 2 converge in probability to those of the usual Metropolis-Hastings algorithm with target \( \pi_K \) (i.e., for each \( t \) generate \( \theta' \sim q(\cdot|\theta^{(t-1)}) \) and \( u \sim \text{Uniform}[0,1] \), then if \( u \leq \frac{\pi_K(\theta'|y_{obs})q(\theta^{(t-1)}|\theta')}{\pi_K(\theta^{(t-1)}|y_{obs})q(\theta'|y_{obs})} \) set \( \theta^{(t)} = \theta' \) and otherwise set \( \theta^{(t)} = \theta^{(t-1)} \).

We address the effect of \( M \) on the efficiency of Algorithm 2. Increasing the number of pseudo-samples \( M \) improves the accuracy of the estimator (2), which one might think could improve the efficiency of Algorithms 1 and 2. This is suggested by the fact that Monte Carlo estimates based on runs of the Metropolis-Hastings chain with target \( \pi_K \) have lower (asymptotic) variance than estimates based on runs of Algorithm 2 (see Section 2). However, it is widely known that the choice \( M = 1 \) minimizes the running time of Algorithm 1 (see Section 2). We show that a similar result holds for Algorithm 2: that the choice \( M = 1 \) yields a running time within a factor of two of optimal. We use natural definitions of running time in the contexts of serial and parallel computing, which are extensions of those used by Pitt et al. (2012), Sherlock et al. (2013), and Doucet et al. (2014), and which capture the time required to obtain a particular Monte Carlo accuracy. Our definition in the serial computing case is the number of pseudo-samples generated, which is an appropriate measure of running time when drawing pseudo-samples is more computationally intensive than
Initialize $\theta^{(0)}$ and set $T^{(0)} = \tilde{\pi}_{K,M}(\theta^{(0)}|y_{obs})$, as in (2):

for $t = 1$ to $n$ do

  Generate $\theta' \sim q(.|\theta^{(t-1)})$, $y_i, \omega_i \sim p(y_i|\theta')$ for $i = 1, \ldots, M$, and $u \sim \text{Uniform}[0,1]$;
  Compute $T' = \tilde{\pi}_{K,M}(\theta'|y_{obs})$;
  if $u \leq \frac{T' q(\theta'^{(t-1)}|\theta')}{T q(\theta^{(t-1)}|\theta')}$ then
    Set $(\theta^{(t)}, T^{(t)}) = (\theta', T')$;
  else
    Set $(\theta^{(t)}, T^{(t)}) = (\theta^{(t-1)}, T^{(t-1)})$;

Algorithm 2: ABC-MCMC

the other steps in Algorithm [2] and when the expected computational cost of drawing each pseudo-sample is the same, i.e. when there is no computational discounting due to having pre-computed relevant quantities.

A number of authors have recommended choosing $M > 1$ in Algorithm [2] because this improves the accuracy of $\tilde{\pi}_{K,M}(\theta|y_{obs})$. Andrieu and Vihola (2014) showed that increasing $M$ decreases the autocorrelation of the Markov chain, and improves the accuracy of the resulting Monte Carlo estimators for a fixed number of Markov chain iterations. However, increasing $M$ also increases the running time of each iteration of the chain (if the $M$ pseudo-samples are drawn serially), or increases the number of processors assigned to draw samples (if the pseudo-samples are drawn in parallel). It is not immediately clear how to select $M$ to optimize this computational tradeoff, in the sense of minimizing the running time required to obtain a desired Monte Carlo accuracy. The problem is particularly hard because the accuracy of Markov chain Monte Carlo estimators depends on the amount of autocorrelation of the Markov chain, which itself depends in a complex way on the characteristics of the target distribution and the construction of the Markov chain.

Several authors have drawn the conclusion that the approximately optimal value of $M$ in pseudo-marginal algorithms (a class of algorithms that includes Algorithm [2]) is obtained by tuning $M$ to achieve a particular variance for the estimator $\tilde{\pi}_{K,M}(\theta|y_{obs})$ (Pitt et al., 2012; Sherlock et al., 2013; Doucet et al., 2014). We demonstrate that in Algorithm [2] such a tuning process is unnecessary, since near-optimal efficiency is obtained by using low-cost, high-variance estimates based on a single pseudo-sample (Proposition 4 and Corollary 5). This result assumes only that the kernel $K(\eta)$ is the uniform kernel $1_{||\eta|| < \epsilon}$ (the most common choice). In particular, and in contrast to earlier work, it does not make any assumptions about the target distribution $\pi(\theta|y_{obs})$.

Our result is in contrast to particle MCMC methods (Andrieu et al., 2010), for which Flury and Shephard (2011) demonstrated that often millions of particles are required to obtain sufficient accuracy. This difference between particle MCMC and ABC-MCMC is largely due to the interacting nature of the particles in particle MCMC, allowing for better path sampling.

The intuition behind our result (in the serial computation case) is that the computational effort of constructing $\tilde{\pi}_{K,M}$ with $M > 1$ pseudo-samples can instead be used to propose $M$ values of $\theta$ and perform the accept-reject step with the less accurate estimator $\tilde{\pi}_{K,1}$. In both cases, the probability of at least one acceptance is essentially the probability that at least one good pseudo-sample is drawn. But the expected number of good pseudo-samples does not depend too much on whether the pseudo-samples are drawn one-at-a-time or in batches of size $M > 1$. Thus, the acceptance rate per pseudo sample should not be much smaller for the $M = 1$ chain than for an $M > 1$ chain, which suggests that the asymptotic variance should not be much smaller either (see, e.g., Tierney (1998)). Since the $M = 1$ chain can accept many more proposals per pseudo-sample than an $M > 1$ chain, this suggests that the asymptotic variance can be much smaller for the $M = 1$ chain.
We also give a simulation study that supports our theoretical results and explores the case where pseudo-samples after the first have a reduced computational cost. Our proofs are based on tools developed by Andrieu and Vihola (2014) that bound the relative efficiency of two pseudo-marginal algorithms. In particular, they involve comparing the distribution of the error in the estimated target density \( \hat{π}_{K,M} \), between ABC-MCMC chains with two different values of \( M \). We also provide an extension (Theorem 3) of the results of Andrieu and Vihola (2014), which may be useful in characterizing other aspects of the efficiency of ABC-MCMC, or the efficiency of other types of pseudo-marginal MCMC methods.

2 Efficiency of ABC and ABC-MCMC

For a measure \( \mu \) on space \( X \) let \( \mu(f) \equiv \int f(x)\mu(dx) \) be the expectation of a real-valued function \( f \) with respect to \( \mu \), let \( L^2(\mu) = \{ f : \mu(f^2) < \infty \} \) be the space of functions with finite variance, and let \( \langle f, g \rangle_\mu \equiv \int f(x)g(x)\mu(dx) \) be the usual inner product for \( f, g \in L^2(\mu) \). For any reversible Markov transition kernel \( H \) with stationary distribution \( \mu \), any function \( f \in L^2(\mu) \), and Markov chain \( X^{(t)} \) evolving according to \( H \) with \( X^{(0)} \sim \mu \), the Markov chain Monte Carlo estimator of \( \mu(f) \) is \( \bar{f}_n \equiv \frac{1}{n} \sum_{t=1}^n f(X^{(t)}) \). The error of this estimator can be measured by the asymptotic variance:

\[
v(f, H) = \lim_{n \to \infty} n \text{Var}_H(\bar{f}_n)
\]

which is closely related to the autocorrelation of the samples \( f(X^{(t)}) \) (Tierney, 1998).

If \( H \) is geometrically ergodic, \( v(f, H) \) is guaranteed to be finite for all \( f \in L^2(\mu) \), while in the non-geometrically ergodic case \( v(f, H) \) may or may not be finite (Roberts and Rosenthal, 2008). When \( v(f, H) \) is infinite our results still hold but are not informative. The fact that our results do not require geometric ergodicity distinguishes them from many results on efficiency of MCMC methods (Guan and Krone, 2007; Woodard et al., 2009). When \( X^{(t)} \overset{\text{d}}{\sim} \mu \), we define \( v(f) = \text{Var}(f(X^{(t)})) \).

We will describe the running time of Algorithms 1 and 2 in two cases: first, when the computations are done serially, and second, when they are done in parallel across \( M \) processors. Using (3), the variance of \( \bar{f}_n \) from a single (reversible) Markov chain \( H \) is roughly \( v(f, H)/n \), so to achieve variance \( \delta > 0 \) in the serial context we need \( v(f, H)/\delta \) iterations. Similarly, the variance of \( \bar{f}_n \) from \( M \) parallel Markov chains \( H \) is roughly \( v(f, H)/(nM) \), so to achieve variance \( \delta > 0 \) in the parallel context we need \( v(f, H)/(\delta M) \) iterations of each Markov chain.

Although our definitions make sense for any function \( f \) of the two values \( (\theta^{(t)}, T^{(t)}) \) described by Algorithm 2 throughout the rest of the note we restrict our attention to functions that depend only on the first coordinate, \( \theta^{(t)} \). That is, when discussing Algorithm 2 or other pseudo-marginal algorithms, we restrict our attention to functions that satisfy \( f(\theta, T_1) = f(\theta, T_2) \) for all \( \theta \) and all \( T_1, T_2 \). We slightly abuse this in our notation, not distinguishing between a function \( f : \Theta \to \mathbb{R} \) of a single variable \( \theta \) and a function \( f : \Theta \times \mathbb{R}^+ \to \mathbb{R} \) of two variables \( (\theta, T) \) that only depends on the first coordinate.

Let \( Q_M \) be the transition kernel of Algorithm 2 like all pseudo-marginal algorithms, \( Q_M \) is reversible (Andrieu and Roberts, 2009). We assume that drawing pseudo-samples is the slowest part of the computation, and that drawing each pseudo-sample takes the same amount of time on average (as also assumed in Pitt et al. 2012, Sherlock et al. 2013, Doucet et al. 2014). Then the running time of \( Q_M \) in the serial context can be measured as the number of iterations times the number of pseudo-samples drawn in each iteration, namely \( C_{f,Q_M,\delta} \equiv Mv(f, Q_M)/\delta \). In the context of parallel computation across \( M \) processors, we compare two competing approaches that utilize all the processors. These are: (a) a single chain with transition kernel \( Q_M \), where the \( M > 1 \) pseudo-samples in each iteration are drawn independently across \( M \) processors; and
(b) $M$ parallel chains with transition kernel $Q_1$. The running time of these methods can be measured as the number of required Markov chain iterations to obtain accuracy $\delta$, namely $C_{f,Q,M,\delta}^{\text{ser}} \equiv v(f,Q_M)/\delta$ utilizing method (a) and $C_{f,Q_1,\delta}^{\text{par}} \equiv v(f,Q_1)/\delta M$ utilizing method (b). Since both measures of computation time are based on the asymptotic variance of the underlying Markov chain, they ignore the initial ‘burn-in’ period and are most appropriate when the desired error $\delta$ is small. Other measures of computation time should be used if the Markov chains are being used to get only a very rough picture of the posterior (e.g. to locate, but not explore, a single posterior mode). Note, however, that in practice there is typically no burn-in period for Algorithm 2, since it is usually initialized using samples from ABC (Marin et al., 2012).

For Algorithm 1 denoted by $R_M$, the running time is defined analogously. However, we must account for the fact that each iteration of $R_M$ yields one accepted value of $\theta$, which may require multiple proposed values of $\theta$ (along with the associated computations, including drawing pseudo-samples). The number of proposed values of $\theta$ to get one acceptance has a geometric distribution with mean equal to the inverse of the marginal probability $p_{\text{acc}}(R_M)$ of accepting a proposed value of $\theta$. So, similarly to $Q_M$, the running time of $R_M$ in the context of serial computing can be measured as $C_{f,R,M,\delta}^{\text{ser}} \equiv M v(f)/(\delta p_{\text{acc}}(R_M))$, and the computation time in the context of parallel computing can be measured as $C_{f,R,M,\delta}^{\text{par}} \equiv v(f)/(\delta M p_{\text{acc}}(R_M))$ utilizing method (a) and $C_{f,R_1,\delta}^{\text{par}} \equiv v(f)/(\delta M p_{\text{acc}}(R_1))$ utilizing method (b).

Using these definitions, we first state the result that $M = 1$ is optimal for ABC (Algorithm 1). This result is widely known but we could not locate it in print, so we include it here for completeness.

**Lemma 1.** The marginal acceptance probability of ABC (Algorithm 1) does not depend on $M$. For $M > 1$ the running times $C_{f,R,M,\delta}^{\text{ser}}$ and $C_{f,R,M,\delta}^{\text{par}}$ of ABC in the serial and parallel contexts satisfy $C_{f,R,M,\delta}^{\text{ser}} = MC_{f,R_1,\delta}^{\text{ser}}$ and $C_{f,R,M,\delta}^{\text{par}} = MC_{f,R_1,\delta}^{\text{par}}$ for any $f \in L^2(\pi_K)$ and any $\delta > 0$.

**Proof.** The marginal acceptance probability of Algorithm 1 is

$$
\int \pi(\theta) \left[ \prod_{i=1}^{M} p(y_i,\theta) \right] \left[ \frac{1}{cM} \sum_{i=1}^{M} K(\eta(y_{\text{obs}}) - \eta(y_i,\theta)) \right] \, d\theta \, dy_1,\theta \ldots dy_M,\theta
$$

which does not depend on $M$. The results for the running times follow immediately. \hfill \square

Our contribution is to show a similar result for ABC-MCMC. We note below that Algorithm 2 is no more accurate (in terms of asymptotic variance) than the Metropolis-Hastings algorithm that uses the exact target distribution $\pi_K$ in equation (1). This suggests that accurately approximating the target by using a large value of $M$ in Algorithm 2 might lead to better performance than using a small $M$. In this section we conclude that this is not the case when the uniform kernel $K(\eta) = 1_{\{||\eta||<\epsilon\}}$ is used. We demonstrate that selecting $M > 1$ in Algorithm 2 is never much better than choosing $M = 1$, show that choosing $M = 1$ can be substantially better in some situations, and conclude that one should choose $M = 1$. It is of interest to extend our results to other kernels.

A potential concern regarding Algorithm 2 is raised by Lee and Latuszynski (2013), who point out that this algorithm is generally not geometrically ergodic when $q$ is a local proposal distribution, such as a random walk proposal. This is due to the fact that in the tails of the distribution $\pi_K$, the pseudo-data $y_{i,\theta}$ are very different from $y_{\text{obs}}$ and so the proposed moves are rarely accepted. This problem can be fixed in several ways. Lee and Latuszynski (2013) give a sophisticated solution that involves choosing a random number of pseudo-samples at every step of Algorithm 2 and they show that this modification increases the class
of target distributions for which the ABC-MCMC algorithm is geometrically ergodic. One consequence of our Proposition 4 is that a simpler ‘obvious’ fix to the problem of geometric ergodicity does not work: increasing the number of pseudo-samples used in Algorithm 2 from 1 to any fixed number $M$ has no impact on the geometric ergodicity of the algorithm.

Algorithm 2 is a special case of pseudo-marginal MCMC, given in Algorithm 3 [Andrieu and Roberts (2009)]. This is used when one cannot evaluate the target density $\mu(x)$ up to a normalizing constant, but does have access to an unbiased and nonnegative estimator $\hat{\mu}(x)$ for each $x \in \mathcal{X}$. Algorithm 3 is a Markov chain on the augmented state space $\mathcal{X} \times \mathbb{R}^+$ and not necessarily Markov in $\mathcal{X}$. Its stationary distribution has marginal distribution $\mu$ for $x$, so under mild regularity conditions the distribution of $x^{(n)}$ converges to $\mu$ as $n \to \infty$. Algorithm 2 is the case with $\mu = \pi_K$ and $\hat{\mu} = \hat{\pi}_{K,M}$.

Our main tool in analyzing Algorithm 2 will be the results of Andrieu and Vihola (2014). Two random variables $X$ and $Y$ are convex ordered if $\mathbb{E}[\phi(X)] \leq \mathbb{E}[\phi(Y)]$ for any convex function $\phi$, we denote this relation by $X \leq_{cx} Y$. Let $H_1, H_2$ be the transition kernels of two pseudo-marginal algorithms with the same proposal kernel $q$ and the same target marginal distribution $\mu$. Denote by $T_{1,x}$ and $T_{2,x}$ the estimators of the unnormalized target used by $H_1$ and $H_2$ respectively. Recall the asymptotic variance $v(f,H)$ from [3]; although $f$ could be a function on $\mathcal{X} \times \mathbb{R}^+$, we restrict our attention to functions $f$ on the non-augmented state space $\mathcal{X}$. Then if $T_{1,x} \leq_{cx} T_{2,x}$ for all $x$, Theorem 3 of Andrieu and Vihola (2014) shows that $v(f,H_1) \leq v(f,H_2)$ for all $f \in L^2(\mu)$.

This tool can be used to show that Algorithm 2 is no more accurate than the Metropolis-Hastings chain with proposal kernel $q$ and target $\pi_K$ (call its transition kernel $Q_\infty$). The following result is a special case of Theorem 3 of Andrieu and Vihola (2012), but we include the very short proof for completeness.

**Corollary 2.** For any $f \in L^2(\pi_K)$ and any $M \geq 1$ we have $v(f,Q_M) \geq v(f,Q_\infty)$.

**Proof.** $Q_\infty$ is a pseudo-marginal algorithm with target marginal distribution $\pi_K$ and proposal $q$, where the estimate of $\pi_K(\theta|y_{\text{obs}})$ is a point mass at the true value. $Q_M$ is also a pseudo-marginal algorithm with target distribution $\pi_K$ and proposal $q$.

A point mass at $\pi_K(\theta|y_{\text{obs}})$ is convex upper-bounded by any random variable with expectation $\pi_K(\theta|y_{\text{obs}})$, by Theorem 3 of Leskelä and Vihola (2014). The result follows from Theorem 3 of Andrieu and Vihola (2014). □

In the appendix we show that this result also holds for the alternative version of ABC-MCMC described in Wilkinson (2013).

Although Corollary 2 suggests that it is advantageous to use a large value of $M$ in Algorithm 2, we will show that this is not the case. To do this, we first give a general result related to Theorem 3 of Andrieu and
Vihola (2014). For any $0 \leq \alpha < 1$ and $i \in \{1, 2\}$, define the estimator

$$T_{i,x,\alpha} = \begin{cases} 0 & \text{with probability } \alpha \\ T_{i,x}/(1 - \alpha) & \text{otherwise} \end{cases}$$

(4)

that is “handicapped” relative to $T_{i,x}$ in the sense that it estimates the target density to be zero with probability $\alpha$, and otherwise uses $T_{i,x}$ (adjusted so that $T_{i,x,\alpha}$ is unbiased). Theorem 3 shows that convex ordering of $T_{1,x} = T_{1,x,0}$ and $T_{2,x,\alpha}$ is enough to obtain a relative bound on the asymptotic variances associated with the pseudo-marginal algorithms $H_1$ and $H_2$ defined above.

**Theorem 3.** Assume that $H_2$ is nonnegative definite. For any $0 \leq \alpha < 1$ and if $T_{1,x} \leq_{cx} T_{2,x,\alpha}$ for all $x$, then for any $f \in L^2(\mu)$ we have

$$v(f, H_1) \leq \frac{1 + \alpha}{1 - \alpha} v(f, H_2).$$

Theorem 3 is proven in the appendix, and assumes that the transition kernel $H_2$ is nonnegative definite. This is a common technical assumption in analyses of the efficiency of Markov chains (Woodard et al., 2009; Narayanan and Rakhlin, 2010), and is done for analytical convenience. It can easily be achieved, for example, by incorporating a “holding probability” (chance of proposing to stay in the same location) of 1/2 into the proposal kernel $q$ (Woodard et al., 2009); also see the related Remark 3.2 of Baxendale et al., 2005).

Theorem 3 yields the following bound for Algorithm 2, proven in the appendix.

**Proposition 4.** If $K(\eta) = 1_{||\eta||<\epsilon}$ for some $\epsilon > 0$, and if the transition kernel $Q_M$ of Algorithm 2 is nonnegative definite, then for any $f \in L^2(\pi_K)$ we have

$$v(f, Q_1) \leq (2M - 1)v(f, Q_M).$$

(5)

If additionally $\int 1_{||\eta(\theta)\eta(\eta(y)dy > a > 0$ for all $\theta$ and some $a > 0$, then $v(f, Q_1) \leq (2/a + 2) v(f, Q_M)$ uniformly in $M$ (i.e., $Q_1$ is order $M$ times faster than $Q_M$).

Proposition 4 implies that it is only possible to get an efficiency gain of two times from running $Q_M$ rather than $Q_1$.

**Corollary 5.** Using the uniform kernel and assuming that $Q_M$ is nonnegative definite, the running time of $Q_1$ is at most twice that of $Q_M$, for both serial and parallel computing:

$$C_{f,Q_1,\delta}^{ser} \leq 2C_{f,Q_M,\delta}^{ser} \quad \text{and} \quad C_{f,Q_1,\delta}^{M,par} \leq 2C_{f,Q_M,\delta}^{M,par}.$$ 

If additionally $\int 1_{||\eta(\theta)\eta(\eta(y)dy > a > 0$ for all $\theta$ then

$$C_{f,Q_1,\delta}^{ser} \leq \left(\frac{2 + 2a}{aM}\right) C_{f,Q_M,\delta}^{ser} \quad \text{and} \quad C_{f,Q_1,\delta}^{M,par} \leq \left(\frac{2 + 2a}{aM}\right) C_{f,Q_M,\delta}^{M,par}.$$ 

So for the uniform kernel there is never a strong reason to use $Q_M$ over $Q_1$, and in fact there can be a strong reason to use $Q_1$ over $Q_M$. 
2.1 Simulation study

We now demonstrate these results through a simple simulation study, showing that choosing \( M > 1 \) is seldom beneficial. We consider the model \( y | \theta \sim N(\theta, \sigma_y^2) \) for a single observation \( y \), where \( \sigma_y \) is known and \( \theta \) is given a standard normal prior, \( \theta \sim N(0,1) \). We apply Algorithm 2 using proposal distribution \( q(\theta | \theta_{(t-1)}) = N(\theta; 0,1) \), summary statistic \( \eta(y) = y \), and \( K(\eta) = 1\{||\eta||<\epsilon\} \) equal to the uniform kernel with bandwidth \( \epsilon \). Each iteration of this algorithm involves simulating \( \theta' \) from \( q(\cdot | \theta_{(t−1)}) \), generating pseudo-samples \( y_1, \theta', \ldots, y_M, \theta' | \theta' \sim N(\theta', \sigma_y^2) \), and accepting or rejecting the move based on the approximated likelihood

\[
\frac{1}{M} \sum_{i=1}^{M} 1\{||y_{\text{obs}} - y_i, \theta'||<\epsilon\}.
\]  

(6)

We start by exploring the case where \( y_{\text{obs}} = 2 \) and \( \sigma_y = 1 \), simulating the Markov chain for 5 million iterations. Figure 1 (left) shows the acceptance rate per pseudo-sample as a function of \( M \). Large \( M \) does not provide a benefit in terms of accepted \( \theta \) samples per generated pseudo-sample, and can even decrease this measure of efficiency, supporting the result of Corollary 5.

In certain cases, there is an initial fixed computational cost to generating pseudo-samples, after which generating subsequent pseudo-samples is computationally inexpensive. If \( y \) is a length-\( n \) sample from a Gaussian process, for instance, there is an initial \( O(n^3) \) cost to decomposing the covariance, after which each pseudo-sample may be generated at \( O(n^2) \) cost. In Figure 1 (right) we look at the \( \epsilon \) which results from requiring that a fixed percentage (0.4%) of \( \theta \) samples are accepted per unit computational cost. In the non-discounted case, for example, this means that for \( M = 1 \) we require that 0.4% of samples are accepted, while for \( M = 64 \) we require that 0.4 \times 64 = 25.6% of samples are accepted. In the case with discount factor \( \delta > 1 \) (representing a \( \delta \times \) cost reduction for all pseudo-samples after the first), the pseudo-sampling cost is \( (1 + (M-1)/\delta) \), so we require that 0.4 \times (1 + (M-1)/\delta)% of \( \theta \) samples are accepted. For example, a discount of \( \delta = 16 \) with \( M = 64 \) requires that 2.0% of samples are accepted. Figure 1 shows that, for \( \delta = 1 \), larger \( M \) results in larger \( \epsilon \); in other words, for a fixed computational budget and no discount, \( M = 1 \) gives the smallest \( \epsilon \). For discounts \( \delta > 1 \), however, increasing \( M \) can indeed lead to reduced \( \epsilon \).

![Figure 1: Left: Acceptance rate per pseudo-sample. Lines correspond to \( \epsilon = 0.5^2 \) (black) through \( \epsilon = 0.5^6 \) (light grey), spaced uniformly on log scale. Right: The \( \epsilon \) resulting from requiring that 0.4% of \( \theta \) samples are accepted per unit computational cost. Lines correspond to different computational savings for pseudo-samples beyond the first, ranging from \( \delta = 1 \) (black, no cost savings) to \( \delta = 16 \) (light grey).](image-url)
We further explore discounting in Figure 2, which uses a discount of $\delta = 8$ and varies $y_{\text{obs}}$ from 2 to 8 (left plot) and $\sigma_y$ from 0.01 to 2 (right plot). In both cases the changes are meant to induce a divergence between the prior and the likelihood, and hence the prior and posterior. In these figures, the requisite $\epsilon$ are scaled such that at $M = 1$ all normalized $\epsilon$ are 1. Figure 2 (left) shows that as $y_{\text{obs}}$ grows, the benefit associated with using higher values of $M$ shrinks and eventually disappears. This is because for large $y_{\text{obs}}$ a large value of $\epsilon$ is required in order to frequently get a nonzero value for the approximated likelihood (6) and thus a reasonable acceptance rate; for instance, the unnormalized value of $\epsilon$ is 0.08 when $y_{\text{obs}} = 2$ and $M = 1$, while $\epsilon = 7.64$ when $y_{\text{obs}} = 8$ and $M = 1$. As such, the increased diversity from multiple samples is dwarfed by the scale of $\epsilon$.

In Figure 2 (right) we examine sensitivity of our conclusions to $\sigma_y$. For large $\sigma_y$, additional (discounted-cost) pseudo-samples provide a benefit, because they improve the accuracy of the approximated likelihood (6). However, for small values of $\sigma_y$, the variability of the pseudo-samples $y_i, \theta_i$ is low and so additional pseudo-samples do not provide much incremental improvement to the likelihood approximation. In summary, we only find a benefit of increasing the number of pseudo-samples $M$ in cases where there is a discounted cost to obtain those pseudo-samples, and even then the benefit can be decreased or eliminated when $y_{\text{obs}}$ is extreme under typical proposed values of $\theta$, or when the variability of $y$ under the model is low.

![Figure 2: Sensitivity of the relationship between the bias and $M$ to changes in the likelihood. Each curve is normalized by dividing by its respective $\epsilon$ for $M = 1$. Left: Varying $y_{\text{obs}} = 2$ (light grey), 4, 6, 8 (black). Right: Varying $\sigma_y = 0.01$ (black), 0.05, 0.1, 0.5, 1, 2 (light grey).](image)

3 Discussion

In this paper, we have shown that despite the true likelihood leading to reduced asymptotic variance relative to the approximated likelihood constructed through ABC, in practice one should stick with simple, single pseudo-sample approximations rather than trying to accurately approximate the true likelihood with multiple pseudo-samples. Our results are obtained by bounding the asymptotic variance of Markov chain Monte Carlo estimates, which takes into account the autocorrelation of the Markov chain. Intuitively, for a fixed number of pseudo-samples $M$, it is seldom beneficial to propose a single proposal $\theta$ and use all $M$ pseudo-samples to accurately approximate the likelihood. In fact, this approach can lead to drastically reduced performance relative to proposing $M \theta$’s each with their own (single) pseudo-sample.
This is not to say that $M = 1$ is optimal in all situations, however. In many cases, there is a large initial cost to the first pseudo-sample, with subsequent samples drawn at a much reduced computational cost. In this case $M > 1$ can lead to improved performance, though that performance disappears when $y_{\text{obs}}$ is extreme (and hence proposal acceptance is based on the unlikely event of having 1 pseudo-sample drawn near $y_{\text{obs}}$) or when the variance of the likelihood is small relative to the approximation bandwidth $\epsilon$ (in which case the kernels from subsequent pseudo-samples are simply stacking on top of earlier pseudo-sample kernels).

We hope that this work not only provides practical guidance on the choice of the number of pseudo-samples when using ABC, but also that it might lead to future research in the analysis of these algorithms. As a specific example, it would be fruitful to pursue the results in this paper extended to non-uniform kernels.

Acknowledgement

The authors thank Alex Thiery for his careful reading of an earlier draft, as well as Pierre Jacob, Rémi Bardenet, Christophe Andrieu, Matti Vihola, Christian Robert, and Arnaud Doucet for useful discussions. This research was supported in part by U.S. National Science Foundation grants DMS-1209103 and DMS-1406599, and by DARPA under Grant No. FA8750-14-2-0117.

A Proofs

Proof of Theorem 3. For a reversible Markov kernel $H$ with stationary distribution $\mu$ and $g \in L^2(\mu)$, we follow Tierney (1998, proof of Theorem 4) by writing the asymptotic variance as

$$v(g, H) = \lim_{n \to \infty} \int_{-1}^{1} \left( 1 + 2 \sum_{i=1}^{n} \frac{n - i}{n} x^i \right) e_{g, H}(dx) = \int_{-1}^{1} \frac{1}{1 - x} e_{g, H}(dx),$$

(7)

where $e_{g, H}$ is the spectral measure associated with $g$ and $H$, i.e., the finite positive measure on $[-1, 1]$ such that $\langle g, H^i g \rangle_{\mu} = \int_{-1}^{1} x^i e_{g, H}(dx)$ for all $i \in \mathbb{N}$. For nonnegative definite transition kernels $H$ the spectral measure, and the integrals, are over the narrower range $[0, 1]$.

Denote by $H_{2, \alpha}$ the transition kernel of the pseudo-marginal algorithm with proposal kernel $q$, target marginal distribution $\mu$, and estimator $T_{2,x,\alpha}$ of the unnormalized target. Denote by $I$ the identity operator. If we denote by $\{(X_{t}^{(1)}, T_{t}^{(1)})\}_{t \in \mathbb{N}}$ and $\{(X_{t}^{(2)}, T_{t}^{(2)})\}_{t \in \mathbb{N}}$ the Markov chains driven by the kernels $H_{2, \alpha}$ and $\alpha I + (1 - \alpha)H_2$ respectively, then $\{X_{t}^{(1)}\}_{t \in \mathbb{N}}$ and $\{X_{t}^{(2)}\}_{t \in \mathbb{N}}$ have the same distribution. This implies that

$$\langle f, H_{2,\alpha}^k f \rangle_{\mu} = \langle f, (\alpha I + (1 - \alpha)H_2)^k f \rangle_{\mu}$$

(8)

for any $k \in \mathbb{N}$ and $f \in L^2(\mu)$ (i.e. for any function $f$ that does not depend on $T_t$). If $T_{1,x} \leq_{cx} T_{2,x,\alpha}$ then by Theorem 3 of Andrieu and Vihola (2014),

$$v(f, H_1) \leq v(f, H_{2, \alpha})$$

(9)

for any $f \in L^2(\mu)$. Using (7)–(8) and the nonnegative-definiteness of $H_2$,

$$v(f, H_{2, \alpha}) = \lim_{n \to \infty} \int_{-1}^{1} \left( 1 + 2 \sum_{i=1}^{n} \frac{n - i}{n} x^i \right) e_{f, H_{2, \alpha}}(dx)$$

(10)
\[
\begin{align*}
&= \lim_{n \to \infty} \left[ (f, f)_{\mu} + 2 \sum_{i=1}^{n} \frac{n-i}{n} \langle f, H_{2, \alpha}^i f \rangle_{\mu} \right] \\
&= \lim_{n \to \infty} \left[ (f, f)_{\mu} + 2 \sum_{i=1}^{n} \frac{n-i}{n} \sum_{k=0}^{i} \binom{i}{k} \alpha^{i-k} (1 - \alpha)^k \langle f, H_{2}^k f \rangle_{\mu} \right] \\
&= \lim_{n \to \infty} \int_{-1}^{1} \left( 1 + 2 \sum_{i=1}^{n} \frac{n-i}{n} (\alpha + (1 - \alpha)x)^i \right) e_{f, H_{2}}(dx) \\
&= \int_{-1}^{1} \frac{1 + \alpha + (1 - \alpha)x}{1 - \alpha} e_{f, H_{2}}(dx) \\
&= \int_{0}^{1} \left( 1 + \alpha - x \right) \frac{1}{1 - \alpha} \frac{1 + x}{1 - x} e_{f, H_{2}}(dx) \\
&\leq \frac{1 + \alpha}{1 - \alpha} \int_{0}^{1} \frac{1 + x}{1 - x} e_{f, H_{2}}(dx) \\
&= \frac{1 + \alpha}{1 - \alpha} v(f, H_{2}).
\end{align*}
\]

Combining this with (9) yields the desired result.

Remark: The conclusion of inequality (10) is almost a special case of Latuszyński and Roberts (2013, Theorem 2), whose proof is quite similar to our calculation. However, since our assumptions are not quite identical to that of Latuszyński and Roberts (2013), we include our argument here.

Proof of Proposition 4 For any \( M \geq 1 \), let \( T_{M, \theta} \) be the estimator \( \hat{\pi}_{K, M}(\theta | y_{\text{obs}}) \) of the target \( \pi_{K} \), so that \( T_{M, \theta, \alpha} \) is \( T_{M, \theta} \) handicapped by \( \alpha \) as defined in (4) of the main document. To obtain (5) of the main document, by Theorem 3, it is sufficient to take \( \alpha = 1 - \frac{1}{M} \) and show that \( T_{1, \theta} \leq_{cx} T_{M, \theta, \alpha} \). By Proposition 2 of Leskelä and Vihola (2014), it is furthermore sufficient to show that, for all \( c \in \mathbb{R} \),

\[
\mathbb{E}[|T_{1, \theta} - c|] \leq \mathbb{E}[|T_{M, \theta, \alpha} - c|].
\]

Let Bin\((n, \psi)\) denote the binomial distribution with \( n \) trials and success probability \( \psi \). For a given point \( \theta \in \Theta \), let \( \tau = \tau(\theta) \equiv \mathbb{P}[T_{1, \theta} \neq 0] = 1 \mathbb{1}_{\{||y(y_{\text{obs}}) - y(y)|| < \epsilon\}} p(y | \theta) dy \). Noting that \( \frac{T_{1, \theta}}{\pi(\theta)} \in \{0, 1\} \), we may then write \( T_{1, \theta} \) and \( T_{M, \theta, \alpha} \) as the following mixtures

\[
\begin{align*}
\frac{T_{1, \theta}}{\pi(\theta)} &\overset{D}{=} \text{Bin}(1, \tau) \\
\frac{T_{M, \theta, \alpha}}{\pi(\theta)} &\overset{D}{=} \frac{M - 1}{M} \delta_{0} + \frac{1}{M} \text{Bin}(M, \tau),
\end{align*}
\]

where \( \delta_{0} \) is the unit point mass at zero. Denote \( T'_{1, \theta} = \frac{T_{1, \theta}}{\pi(\theta)} \) and \( T'_{M, \theta, \alpha} = \frac{T_{M, \theta, \alpha}}{\pi(\theta)} \). We will check condition (11) for \( T'_{1, \theta}, T'_{M, \theta, \alpha} \) and \( 0 \leq c \leq 1 \), then separately for \( c < 0 \) and \( c > 1 \). For \( 0 \leq c \leq 1 \), we compute:

\[
\mathbb{E}[|T'_{M, \theta, \alpha} - c|] = \left( 1 - \frac{1}{M} \right) c + \frac{1}{M} (1 - \tau) M c + \frac{1}{M} \left( \sum_{j=1}^{M} \frac{M!}{j!(M-j)!} \tau^{j} (1 - \tau)^{M-j} (j-c) \right)
\]

\[
= \tau + c \left( 1 - \frac{2}{M} (1 - (1 - \tau)^{M}) \right).
\]
\[
\begin{align*}
\geq \tau + c(1 - 2\tau) \\
= \mathbb{E}[|T'_{1,\theta} - c|].
\end{align*}
\]
For \( c < 0 \), we have
\[
\mathbb{E}[|T'_{M,\theta,\alpha} - c|] = \mathbb{E}[T'_{M,\theta,\alpha}] - c = \mathbb{E}[T'_{1,\theta}] - c = \mathbb{E}[|T'_{1,\theta} - c|],
\]
and the analogous calculation gives the same conclusion for \( c \geq M \). To prove the result for \( 1 < c < M \), note that
\[
\begin{align*}
\mathbb{E}[|T'_{1,\theta} - 1|] &\leq \mathbb{E}[|T'_{M,\theta,\alpha} - 1|] \\
\mathbb{E}[|T'_{1,\theta} - M|] &= \mathbb{E}[|T'_{M,\theta,\alpha} - M|].
\end{align*}
\]
Also, the functions \( f_1(c) \equiv \mathbb{E}[|T'_{1,\theta} - c|] \) and \( f_2(c) \equiv \mathbb{E}[|T'_{M,\theta,\alpha} - c|] \) are continuous, convex and piecewise linear. For \( c \geq 1 \), they satisfy
\[
\frac{d}{dc} f_1(c) = 1 \geq \frac{d}{dc} f_2(c)
\]
where the derivative of \( f_2 \) exists. Combining inequalities (12) and (13), we conclude that
\[
\mathbb{E}[|T'_{1,\theta} - c|] \leq \mathbb{E}[|T'_{M,\theta,\alpha} - c|]
\]
for all \( 1 < c < M \). Thus we have verified (11) and the first claim follows. Analogous calculations yield the second claim of Proposition 4.

\section{Analysis of an Alternative ABC-MCMC Method}

Here we give a result analogous to Corollary 2 for an alternative version of ABC-MCMC proposed in Wilkinson (2013), given in Algorithm 4 below. The constant \( c \) can be any value satisfying \( c \geq \sup_y K(\eta(y_{\text{obs}}) - \eta(y)) \).

\begin{algorithm}
Initialize \( \theta^{(0)} \);
for \( t = 1 \) to \( n \) do
\begin{itemize}
\item Generate \( \theta' \sim q(\theta^{(t-1)}) \), \( y_{\theta'} \sim p(y|\theta') \), and \( u \sim \text{Uniform}[0, 1] \):
\item if \( u \leq r(\theta^{(t-1)}, \theta'|y_{\theta'}) \equiv \frac{K(\eta(y_{\text{obs}}) - \eta(y_{\theta'}))}{c} \min \left\{ 1, \frac{\pi(\theta')q(\theta^{(t-1)}|\theta')}{\pi(\theta^{(t-1)})q(\theta'|\theta^{(t-1)})} \right\} \), then
\item Set \( \theta^{(t)} = \theta' \)
\else
\item Set \( \theta^{(t)} = \theta^{(t-1)} \)
\end{itemize}
\end{algorithm}

\textbf{Algorithm 4:} Alternative ABC-MCMC Method

Lemma 6 compares Algorithm 4 (call its transition kernel \( \tilde{Q} \)) to \( Q_{\infty} \).

\textbf{Lemma 6.} For any \( f \in L^2(\pi_K) \) we have \( v(f, \tilde{Q}) \geq v(f, Q_{\infty}) \).

\textbf{Proof.} Both \( \tilde{Q} \) and \( Q_{\infty} \) have stationary density \( \pi_K \), so by Peskun ordering results such as Theorem 4 of Tierney (1998), it suffices to show that \( Q_{\infty}(\theta, A\setminus\{\theta\}) \geq \tilde{Q}(\theta, A\setminus\{\theta\}) \) for every \( \theta \in \Theta \) and every measurable \( A \subset \Theta \). Since \( \tilde{Q} \) and \( Q_{\infty} \) use the same proposal density \( q(\theta'|\theta^{(t-1)}) \), it is furthermore sufficient
to show that for every $\theta^{(t-1)}$ and $\theta'$, the acceptance probability of $Q_\infty$ is at least as large as that of $\tilde{Q}$, where computing the latter requires marginalizing over $y_{\theta'}$.

Since $p(\cdot|\theta)$ is a probability density,

$$
\int K(\eta(y_{\text{obs}}) - \eta(y)) p(y|\theta) dy \leq \sup_y K(\eta(y_{\text{obs}}) - \eta(y)) \quad \forall \theta \in \Theta.
$$

(14)

So the acceptance probability of $\tilde{Q}$, marginalizing over $y_{\theta'}$, is

$$
a_{\text{ABC}} = \int r(\theta^{(t-1)}, \theta'|y)p(y|\theta')dy
\leq \frac{\int K(\eta(y_{\text{obs}}) - \eta(y)) p(y|\theta')dy}{\sup_y K(\eta(y_{\text{obs}}) - \eta(y))} \min \left\{ 1, \frac{\pi(\theta') q(\theta^{(t-1)}|\theta')}{\pi(\theta^{(t-1)}) q(\theta'|\theta^{(t-1)})} \right\}
\leq \min \left\{ 1, \frac{\int K(\eta(y_{\text{obs}}) - \eta(y)) p(y|\theta')dy}{\sup_y K(\eta(y_{\text{obs}}) - \eta(y))} \left( \frac{\pi(\theta') q(\theta^{(t-1)}|\theta')}{\pi(\theta^{(t-1)}) q(\theta'|\theta^{(t-1)})} \right) \right\}.
$$

(15)

The acceptance probability of the Metropolis-Hastings algorithm is

$$
a_{\text{MH}} = \min \left\{ 1, \frac{\int K(\eta(y_{\text{obs}}) - \eta(y)) p(y|\theta')dy}{\int K(\eta(y_{\text{obs}}) - \eta(y)) p(y|\theta^{(t-1)})dy} \left( \frac{\pi(\theta') q(\theta^{(t-1)}|\theta')}{\pi(\theta^{(t-1)}) q(\theta'|\theta^{(t-1)})} \right) \right\}.
$$

(16)

Using (14), (15), and (16), $a_{\text{ABC}}/a_{\text{MH}} \leq 1.$

**References**

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

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

Andrieu, C. and M. Vihola (2012). Convergence properties of pseudo-marginal markov chain monte carlo algorithms. *arXiv preprint arXiv:1210.1484*.

Andrieu, C. and M. Vihola (2014). Establishing some order amongst exact approximations of MCMCs. *arXiv preprint, arXiv:1404.6909v1*.

Baxendale, P. H. et al. (2005). Renewal theory and computable convergence rates for geometrically ergodic markov chains. *The Annals of Applied Probability* 15(1B), 700–738.

Doucet, A., M. Pitt, G. Deligiannidis, and R. Kohn (2014). Efficient implementation of Markov chain Monte Carlo when using an unbiased likelihood estimator. *arXiv preprint, arXiv:1210.1871v3*.

Flury, T. and N. Shephard (2011). Bayesian inference based only on simulated likelihood: particle filter analysis of dynamic economic models. *Econometric Theory* 27(5), 933–956.

Guan, Y. and S. M. Krone (2007). Small-world MCMC and convergence to multi-modal distributions: from slow mixing to fast mixing. *Annals of Applied Probability* 17, 284–304.
Latuszyński, K. and G. O. Roberts (2013). Clts and asymptotic variance of time-sampled markov chains. *Methodology and Computing in Applied Probability* 15(1), 237–247.

Lee, A. and K. Latuszynski (2013). Variance bounding and geometric ergodicity of Markov chain Monte Carlo kernels for approximate Bayesian computation. *arXiv preprint arXiv:1210.6703*.

Leskelä, L. and M. Vihola (2014). Conditional convex orders and measurable martingale couplings. *arXiv preprint, arXiv:1404.0999*.

Marin, J.-M., P. Pudlo, C. P. Robert, and R. J. Ryder (2012). Approximate Bayesian computational methods. *Statistics and Computing* 22, 1167–1180.

Marjoram, P., J. Molitor, V. Plagnol, and S. Tavaré (2003). Markov chain Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences* 100(26), 15324–15328.

Narayanan, H. and A. Rakhlin (2010). Random walk approach to regret minimization. In *Advances in Neural Information Processing Systems*.

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

Roberts, G. O. and J. S. Rosenthal (2008). Variance bounding Markov chains. *Annals of Applied Probability* 18, 1201–1214.

Sherlock, C., A. H. Thiery, G. O. Roberts, and J. S. Rosenthal (2013). On the efficiency of pseudo-marginal random walk Metropolis algorithms. *arXiv preprint arXiv:1309.7209*.

Tavare, S., D. J. Balding, R. Griffiths, and P. Donnelly (1997). Inferring coalescence times from DNA sequence data. *Genetics* 145(2), 505–518.

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

Wied, D. and R. Weiβbach (2012). Consistency of the kernel density estimator: a survey. *Statistical Papers* 53(1), 1–21.

Wilkinson, R. D. (2013). Approximate Bayesian computation (ABC) gives exact results under the assumption of model error. *Statistical applications in genetics and molecular biology* 12(2), 129–141.

Woodard, D. B., S. C. Schmidler, and M. Huber (2009). Conditions for rapid mixing of parallel and simulated tempering on multimodal distributions. *Annals of Applied Probability* 19, 617–640.