An Interacting Particle Method for Approximate Bayes Computations

Carlo Albert*and Hans R. Künsch†

August 13, 2012

Abstract

Approximate Bayes Computations (ABC) are used for parameter inference when the likelihood function is expensive to evaluate but relatively cheap to sample from. In ABC, a population of particles in the product space of outputs and parameters is propagated in such a way that its output marginal approaches a delta function at the measured output and its parameter marginal approaches the posterior distribution. Inspired by simulated annealing, we present a new class of particle algorithms for ABC, based on a sequence of Metropolis kernels, associated with a decreasing sequence of tolerances w.r.t. the measured output. Unlike other algorithms, our class of algorithms is not based on importance sampling. Hence, it does not suffer from a loss of effective sample size due to re-sampling. We prove convergence under a condition on the speed at which the tolerance is decreased. Furthermore, we present a scheme that adapts the tolerance according to the mean and the standard deviation of the distance of the particles from the measured output, and the jump distribution in parameter space according to the covariance of the population. These adaptations can be interpreted as mean-field interactions between the particles. Thus, the statistical independence of the particles is preserved, in the limit of infinite sample size. The performance of this new class of algorithms is investigated with a toy example, for which we have an analytical solution.

1 Introduction

One way of implementing parameter inference in the Bayesian framework is to generate samples from the *posterior distribution*

$$f_{post}(\boldsymbol{\theta}|\mathbf{y}) = \frac{f(\mathbf{y}|\boldsymbol{\theta})f(\boldsymbol{\theta})}{f(\mathbf{y})}\,, \tag{1}$$

where $f(\boldsymbol{\theta})$ denotes the *prior distribution* encoding our knowledge about the parameter vector $\boldsymbol{\theta}$ before the experiment and $f(\mathbf{y}|\boldsymbol{\theta})$ is the *likelihood function*, that is, the probability density of outputs given the parameter vector $\boldsymbol{\theta}$, evaluated at the measurement vector $\mathbf{y}$. Numerical methods such as the *Metropolis* algorithm [8] require many evaluations of the likelihood function to generate such a sample. However, for complex stochastic models, the likelihood function is often prohibitively expensive to evaluate. Therefore, in recent years, algorithms

[*] Eawag, aquatic research, 8600 Dübendorf, Switzerland.

[†] Seminar für Statistik, ETH Zürich, 8092 Zürich, Switzerland.

have been suggested that generate samples from (1) by *sampling from* the likelihood rather than calculating its value.

As far as we know, the origin of these algorithms is to be found in population genetics. Tavaré et al. [10] replaced the output of a genetic model by a summary statistic and adopted a rejection technique to generate samples from the posterior. Weiss et al. [12] extended this method sampling a vector of summary statistics and introducing a *tolerance* for its distance from the observed summary statistics. Thus, their algorithm generates samples from an *approximate* posterior. Algorithms that generate samples from an approximate posterior via sampling outputs from the likelihood are nowadays called *Approximate Bayes Computations* (ABC). Marjoram et al. [7] used *Markov chains* to produce samples from an approximate posterior. Their algorithm combines a random walk in parameter space with drawing from the likelihood and an acceptance/rejection step that accounts for the prior and only accepts moves into an ϵ ball around the target $\mathbf{y}$. However, a small static tolerance leads to a high rejection rate. Therefore, Toni et al. [11] suggested using a decreasing sequence of tolerances and letting a population of particles of constant size N evolve towards an approximate posterior. Their algorithm consists of an iteration of *importance sampling* steps, where each iteration consists of drawing a new population from the old one with weights and subsequent re-weighting. This re-weighting leads to a loss of effective sample size at each iteration step and, furthermore, computational costs of the order $\mathcal{O}(N^2)$. An adaptive version of Toni's algorithm, which uses the empirical variances of the population to adapt the jump distribution in parameter space, was presented by Beaumont [3]. All of the mentioned algorithms generate samples from the probability distribution proportional to $f(\boldsymbol{\theta})f(\mathbf{x}|\boldsymbol{\theta})\chi(\epsilon - \rho(\mathbf{x}, \mathbf{y}))$, where ρ is some metric on the output space and χ denotes the Heaviside function whose value is unity if its argument is non-negative and 0 otherwise.

In this paper, we present a new class of (adaptive) population algorithms that are of order $\mathcal{O}(N)$ and do not suffer from a loss of effective sample size. The idea is to start with a population of particles drawn from an arbitrary distribution (e.g. the prior) in the product space of parameters and outputs and apply a sequence of Markov kernels, (P_{ϵ_k}), each of which having

$$Z^{-1}(\epsilon_k)f(\boldsymbol{\theta})f(\mathbf{x}|\boldsymbol{\theta})e^{-\rho(\mathbf{x},\mathbf{y})/\epsilon_k}$$

as equilibrium distribution. A choice for P_ϵ, which does not require evaluation of $f(\mathbf{x}|\boldsymbol{\theta})$ for simulation is given in (5). The key question is then how fast we should decrease ϵ_k in order to have a fast convergence and at the same time not to acquire an additional bias due to a too fast convergence. We will give a convergence proof for a schedule that satisfies

$$\epsilon_k \geq \mathrm{const}\, k^{-\alpha/n}\,,$$

where n is the dimension of the output space and α is defined in (4). Furthermore, we will present an adaptive schedule that attempts to stay close to equilibrium, at all times. Both the jump distribution in parameter space and the tolerance ϵ are adapted using the empirical covariance of the population in parameter space and both the average and standard deviation of the distance from the target, respectively. The adaptation of ϵ we suggest was developed for simulated annealing and motivated from thermodynamics [9]. It is a heuristic scheme, for which there is no convergence proof, yet. The adaptation can be interpreted as a mean-field interaction between the particles. As a consequence, the particles remain statistically *independent*, in the limit of an infinite sample size.

The tolerance ϵ that can be achieved in reasonable time is limited by the dimension of the output space. This deficiency is inherent to all ABC algorithms simply because drawing an output from an ϵ-ball around $\mathbf{y}$ scales like ϵ^n. Methods to reduce this bias are investigated elsewhere (see, e.g., Leuenberger et al. [6]).

The paper is organized as follows: In Subsect. 2.1, we explain the main idea behind our class of algorithms. In Subsect. 2.2, the explicit scheme together with a convergence proof is given. The adaptive scheme is developed in Subsect. 2.3, at the end of which a version of it is provided in pseudo-code for convenience. In Sect. 2.4, a comparison with the Metropolis algorithm and population algorithms that are based on importance sampling is made. Sect. 3 contains an application to a toy model, for which the posterior is available analytically.

2 A new class of ABC algorithms

2.1 Basic idea

Our aim is to sample from the posterior distribution (1), without evaluating the likelihood function. The basic idea of ABC is to rewrite (1) as the marginalization

$$f_{post}(\boldsymbol{\theta}|\mathbf{y}) \propto \int f(\mathbf{x}|\boldsymbol{\theta})f(\boldsymbol{\theta})\delta(\mathbf{x}-\mathbf{y})d\mathbf{x} \tag{2}$$

and sample from the joint density $f(\mathbf{x}|\boldsymbol{\theta})f(\boldsymbol{\theta})\delta(\mathbf{x}-\mathbf{y})$ in the $(\boldsymbol{\theta},\mathbf{x})$-space, $\Theta \times X$. If the output space has a high cardinality or is continuous, sampling from $f(\mathbf{x}|\boldsymbol{\theta})f(\boldsymbol{\theta})\delta(\mathbf{x}-\mathbf{y})$ becomes inefficient or impossible, respectively. In these cases, we approximate it by the following family of distributions

$$\pi_\epsilon(\boldsymbol{\theta},\mathbf{x}) = \frac{1}{Z(\epsilon)} f(\mathbf{x}|\boldsymbol{\theta})f(\boldsymbol{\theta})e^{-\rho(\mathbf{x},\mathbf{y})/\epsilon}\,, \tag{3}$$

where $\rho(\mathbf{x},\mathbf{y})$ measures how close $\mathbf{x}$ is to the observation $\mathbf{y}$. For simplicity, we set $X = \mathbb{R}^n$ and

$$\rho(\mathbf{x},\mathbf{y}) = \frac{1}{\alpha}\sum_{i=1}^{n} |x_i - y_i|^\alpha\,, \tag{4}$$

for some $\alpha > 0$, but our results could easily be extended to more general manifolds equipped with distance measures obeying suitable regularity conditions. This might become necessary, if *summary statistics* are used to map the output space to some smaller-dimensional manifold (see, e.g., [10], [12]).

Under the assumption that $f(\mathbf{x}|\boldsymbol{\theta})$ is uniformly bounded and, as a function of $\mathbf{x}$, continuous at $\mathbf{y}$, π_ϵ converges weakly to $f(\mathbf{x}|\boldsymbol{\theta})f(\boldsymbol{\theta})\delta(\mathbf{x}-\mathbf{y})d\boldsymbol{\theta}d\mathbf{x}$, for $\epsilon \searrow 0$. Our idea is to choose a family of Markov transition kernels (P_ϵ) on the space $\Theta \times X$, which have π_ϵ as stationary distribution and apply them recursively on members of a sample drawn from an arbitrary initial distribution, for a decreasing sequence of ϵ's. If ϵ is decreased sufficiently slowly, we expect to end up with an approximate sample from the posterior distribution. This is analogous to the simulated annealing algorithm, although in simulated annealing the limiting distribution is usually concentrated on a finite set. Still, we will strongly rely on ideas developed in the context of simulated annealing. The transition kernels (P_ϵ) that we will use are defined by the transition densities

$$q_\epsilon((\boldsymbol{\theta}',\mathbf{x}'),(\boldsymbol{\theta},\mathbf{x})) = k(\boldsymbol{\theta}',\boldsymbol{\theta})f(\mathbf{x}|\boldsymbol{\theta})\min\left(1, \frac{f(\boldsymbol{\theta})e^{-\rho(\mathbf{x},\mathbf{y})/\epsilon}}{f(\boldsymbol{\theta}')e^{-\rho(\mathbf{x}',\mathbf{y})/\epsilon}}\right)\,, \tag{5}$$

combined with a multiple of a Dirac delta distribution at $(\boldsymbol{\theta}', \mathbf{x}')$ such that $P_\epsilon((\boldsymbol{\theta}', \mathbf{x}'), \Theta \times X) = 1$. Here, k is a symmetric transition density on Θ. It is straightforward to check that π_ϵ is the equilibrium distribution for P_ϵ.

The main question now is how fast ϵ should be decreased. Obviously, an arbitrarily slow decrease of ϵ allows to stay arbitrarily close to equilibrium at all times, which guarantees convergence. However, this is clearly inefficient. On the other hand, a decrease that is much faster than the relaxation velocity of the transition kernel may result in slow convergence (because the acceptance probability decreases for decreasing ϵ) or convergence to a biased result. A bias can be the result of the process not having enough time to explore the Θ space while converging in the X space. For instance if we set $\epsilon = 0$, $(\boldsymbol{\theta}, \mathbf{x})$ is accepted iff $\rho(x, y) \leq \rho(x', y)$. Hence in this case, the prior has no influence, which leads to convergence to a biased result.

In the next subsection we will present an explicit schedule (ϵ_k) that ensures convergence to an unbiased result. A potentially better performance can be achieved when the state of the system is used to adapt the tolerance ϵ and the jump distribution k. This idea will be developed in Subsect. 2.3.

2.2 An explicit scheme with convergence proof

In this subsection, we use a time discrete description. That is, we start with a sample from an arbitrary distribution μ_0 and then recursively make transitions of the whole sample with the kernel P_{ϵ_k}, for an explicitly given decreasing sequence $\epsilon_k \searrow 0$. In this way, we generate samples distributed according to

$$\mu_{k+1} = \mu_k P_{\epsilon_{k+1}} = \int P_{\epsilon_{k+1}}(\boldsymbol{\theta}, \mathbf{x}; .) d\mu_k(\boldsymbol{\theta}, \mathbf{x}). \tag{6}$$

We expect that for a suitable choice of (ϵ_k), μ_k will converge weakly to $f(\mathbf{x}|\boldsymbol{\theta}) f(\boldsymbol{\theta}) \delta(\mathbf{x} - \mathbf{y}) d\boldsymbol{\theta} d\mathbf{x}$, and thus in particular the marginal will converge weakly to the posterior distribution (1).

In order to ease the notation we set $\mathbf{z} = (\boldsymbol{\theta}^T, \mathbf{x}^T)^T$ and write, for the joint prior,

$$f(\mathbf{z}) := f(\mathbf{x}|\boldsymbol{\theta}) f(\boldsymbol{\theta}) \,.$$

Furthermore, w.l.o.g. we will assume $\mathbf{y} = 0$ and replace $\rho(\mathbf{x}, \mathbf{y})$ by $\rho(\mathbf{x})$. For our main result, we make the following assumptions about the parameter space Θ and the functions $k(\boldsymbol{\theta}', \boldsymbol{\theta})$, $f(\boldsymbol{\theta})$ and $f(\mathbf{x}|\boldsymbol{\theta})$ thereon:

(A1) $\exists c_1 > 1$ such that $c_1^{-1} \leq f(\boldsymbol{\theta}) / f(\boldsymbol{\theta}') \leq c_1$, for all $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \Theta$.

(A2) $\exists c_2 > 0$ such that $k(\boldsymbol{\theta}', \boldsymbol{\theta}) \geq c_2 f(\boldsymbol{\theta})$, for all $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \Theta$.

(A3) $f(\mathbf{x}|\boldsymbol{\theta})$ is continuously differentiable w.r.t. $\mathbf{x}$ for all $\boldsymbol{\theta}$, and the function and all partial derivatives are bounded uniformly in $\mathbf{x}$ and $\boldsymbol{\theta}$.

These conditions essentially restrict the parameter space to be compact. We will in fact prove stronger than weak-convergence results, namely convergence in total variation of the distributions of $(\boldsymbol{\theta}, \epsilon_k^{-1/\alpha} \mathbf{x})$, with α as defined in (4). The densities of these scaled distributions are

$$\hat{\mu}_k(\boldsymbol{\theta}, \mathbf{x}) := \epsilon_k^{n/\alpha} \mu_k(\boldsymbol{\theta}, \epsilon_k^{1/\alpha} \mathbf{x})$$

and

$$\hat{\pi}_\epsilon(\boldsymbol{\theta}, \mathbf{x}) := \epsilon^{n/\alpha}\pi_\epsilon(\boldsymbol{\theta}, \epsilon^{1/\alpha}\mathbf{x}) = \frac{1}{C(\epsilon^{1/\alpha})} f(\epsilon^{1/\alpha}\mathbf{x}|\boldsymbol{\theta}) f(\boldsymbol{\theta}) \exp(-\rho(\mathbf{x}))$$

where

$$C(\epsilon^{1/\alpha}) = \int f(\epsilon^{1/\alpha}\mathbf{x}|\boldsymbol{\theta}) f(\boldsymbol{\theta}) \exp(-\rho(\mathbf{x})) d\mathbf{z}\,,$$

and the transition densities for the scaled variables are

$$\hat{q}_{\epsilon^{1/\alpha}}(\mathbf{z}, \mathbf{z}') = \epsilon^{n/\alpha} q_\epsilon((\boldsymbol{\theta}, \epsilon^{1/\alpha}\mathbf{x}), (\boldsymbol{\theta}', \epsilon^{1/\alpha}\mathbf{x}'))\,.$$

Theorem 2.1. *If the assumptions (A1) – (A3) above are satisfied and if*

$$\epsilon_k \geq \text{const}\, k^{-\alpha/n}\,, \tag{7}$$

for an arbitrary constant (where n denotes the dimension of X and α is defined by (4)), then, for any absolutely continuous initial distribution $\hat{\mu}_0$ the distribution $\hat{\mu}_k$ converges in total variation to $\hat{\pi}_0(\mathbf{z}) \propto f_{post}(\boldsymbol{\theta}|0)\exp(-\rho(\mathbf{x}))$, for $k \to \infty$.

Proof: We will apply corollary (2.34) in [5]. We start by introducing some notation. Let

$$\hat{\pi}_k = \hat{\pi}_{\epsilon_k}\,, \quad \hat{P}_k = \hat{P}_{\epsilon_k}\,, \quad \hat{P}_{s:t} = \hat{P}_s\hat{P}_{s+1}\dots\hat{P}_t\,,$$

where $\hat{P}_\epsilon$ is defined by the transition density $\hat{q}_\epsilon$.

By assumption (A3) and dominated convergence,

$$\hat{\pi}_k(\boldsymbol{\theta}, \mathbf{x}) \to \hat{\pi}_0(\boldsymbol{\theta}, \mathbf{x}) = \frac{f(0|\boldsymbol{\theta}) f(\boldsymbol{\theta}) \exp(-\rho(\mathbf{x}))}{\int f(0|\boldsymbol{\theta}) f(\boldsymbol{\theta}) d\boldsymbol{\theta} \int \exp(-\rho(\mathbf{x})) d\mathbf{x}}$$

pointwise and thus by Scheffé's theorem also in L^1-norm, that is in total variation. In order to deduce

$$||\hat{\mu}_0\hat{P}_{0:t} - \hat{\pi}_0||_{TV} \to 0,$$

we have to verify conditions (2.31) and (2.33) in [5]. These conditions are

$$\prod_k c(\hat{P}_k) = 0\,, \tag{8}$$

where

$$c(\hat{P}_k) = \sup_{\mathbf{z},\mathbf{z}'} ||\hat{P}_k(\mathbf{z}, .) - \hat{P}_k(\mathbf{z}', .)||_{TV}\,,$$

and

$$\sum_k ||\hat{\pi}_{k+1} - \hat{\pi}_k||_{TV} < \infty. \tag{9}$$

Replacing $\epsilon^{1/\alpha}$ by ϵ, we may set, without loss of generality, $\alpha = 1$. To get an upper bound for $c(\hat{P}_\epsilon)$ we use

$$c(\hat{P}_\epsilon) = \sup_{\mathbf{z}',\mathbf{z}''} \left(1 - \int \min(\hat{q}_\epsilon(\mathbf{z}', \mathbf{z}), \hat{q}_\epsilon(\mathbf{z}'', \mathbf{z})) d\mathbf{z}\right).$$

By (A1) and (A2), for any $\mathbf{z}'$,

$$\hat{q}_\epsilon(\mathbf{z}', \mathbf{z}) \geq \epsilon^n \frac{c_2}{c_1} f(\boldsymbol{\theta}) f(\epsilon\mathbf{x}|\boldsymbol{\theta}) \exp(-\rho(\mathbf{x}))\,.$$

Hence we obtain

$$\int \min(\hat{q}_\epsilon(\mathbf{z}', \mathbf{z}), \hat{q}_\epsilon(\mathbf{z}'', \mathbf{z}))d\mathbf{z} \geq \epsilon^n \frac{c_2}{c_1} C(\epsilon).$$

Because $C(\epsilon) \to C(0) > 0$ as $\epsilon \to 0$, it follows that, for ϵ sufficiently small ϵ,

$$c(\hat{P}_\epsilon) \leq 1 - \frac{c_2}{c_1}\frac{C(0)}{2}\epsilon^n \,, \tag{10}$$

and (8) holds for the choice (7).

In order to show (9), we start with

$$|\hat{\pi}_\epsilon(\mathbf{z}) - \hat{\pi}_{\epsilon'}(\mathbf{z})| \leq \frac{|f(\epsilon\mathbf{x}|\boldsymbol{\theta}) - f(\epsilon'\mathbf{x}|\boldsymbol{\theta})| f(\boldsymbol{\theta}) \exp(-\rho(\mathbf{x}))}{C(\epsilon)} + \hat{\pi}_{\epsilon'}(\mathbf{z}) \frac{|C(\epsilon') - C(\epsilon)|}{C(\epsilon)} \,.$$

By (A3) and the intermediate value theorem, we obtain that

$$|f(\epsilon\mathbf{x}|\boldsymbol{\theta}) - f(\epsilon'\mathbf{x}|\boldsymbol{\theta})| \leq \text{const}\, ||\mathbf{x}||_1 |\epsilon - \epsilon'|$$

and, moreover, that $C(\epsilon)$ is differentiable with

$$|C'(\epsilon)| \leq \text{const} \int ||\mathbf{x}||_1 \exp(-\rho(\mathbf{x}))d\mathbf{x} \,,$$

where const is the bound for the partial derivatives of $f(.|\boldsymbol{\theta})$. Hence we find that

$$||\hat{\pi}_\epsilon - \hat{\pi}_{\epsilon'}||_{TV} \leq \frac{\text{const}}{C(\epsilon)} \int ||\mathbf{x}||_1 \exp(-\rho(\mathbf{x}))d\mathbf{x}\, |\epsilon - \epsilon'| \,.$$

Therefore (9) holds for any sequence (ϵ_k) which converges monotonically to zero.

□

2.3 An adaptive scheme

In this subsection, we adopt an optimal adaptive cooling strategy that has been developed for simulated annealing [9]. It is naturally expressed in the language of thermodynamics using a continuous time description. We propagate a distribution, $\mu(\mathbf{z}, t)$, with (5) , which is now interpreted as a transition *rate* and whose tolerance ϵ is time-dependent. Then, the time dependence of $\mu(\mathbf{z}, t)$ is described by the master equation (or Kolmogorov forward equation)

$$\frac{d}{dt}\mu(\mathbf{z}, t) = \int (\mu(\mathbf{z}', t) q_{\epsilon(t)}(\mathbf{z}', \mathbf{z}) - \mu(\mathbf{z}, t) q_{\epsilon(t)}(\mathbf{z}, \mathbf{z}'))d\mathbf{z}' \,. \tag{11}$$

Now, the idea is to decrease $\epsilon(t)$ adaptively in such a way that $\mu(\mathbf{z}, t)$ stays close to the target distribution

$$\pi(\mathbf{z}, t) = Z^{-1}(\epsilon(t)) f(\mathbf{x}|\boldsymbol{\theta}) f(\boldsymbol{\theta}) e^{-\rho(\mathbf{x},\mathbf{y})/\epsilon(t)} \,, \tag{12}$$

at all times, while minimizing the number of computer iterations needed to reach a given final state. Thus, $\epsilon(t)$ will be dependent on $\mu(t)$ and the master equation (11) will become *non-linear.* As a measure for the number of computer iterations, the *total entropy production* is used. We introduce the notation $\rho(\pi)$ for the mean distance to $\mathbf{y}$ under distribution π, i.e., we set

$$\rho(\pi(t)) = \int \rho(\mathbf{x}, \mathbf{y}) \pi(\mathbf{z}, t) d\mathbf{z} \,,$$

and, analogously, for $\rho(\mu(t))$. It can be shown that (see, e.g., [2]), as long as the difference $\rho(\mu(t)) - \rho(\pi(t))$ is relatively small, minimal entropy production is approximately satisfied if $\epsilon(t)$ satisfies the differential equation

$$\frac{d\epsilon(t)}{dt} = -\frac{v\epsilon(t)}{\sqrt{C(t)}\eta(t)}\,, \tag{13}$$

where v is *constant* and relatively small. In (13), $C(t)$ is the derivative of the mean particle distance from $\mathbf{y}$ w.r.t. ϵ, i.e.,

$$C(t) = \frac{d\rho(\pi(t))}{d\epsilon(t)}\,. \tag{14}$$

If ρ is interpreted as an *energy* and ϵ as a *temperature* then $C(t)$ can be interpreted as a *heat capacity.* Furthermore, $\eta(t)$ is a *relaxation time*, namely the time the system's mean energy would take to reach the target's mean energy at current velocity. That is,

$$\eta(t) = \frac{\rho(\pi(t)) - \rho(\mu(t))}{d\rho(\pi(t))/dt}\,. \tag{15}$$

In terms of natural time units defined by $\eta(t)$ and rescaling energy with $\sigma(\pi(t))$ (see (17)), the tuning parameter v can be interpreted as a *thermodynamic speed.* In accordance with [9], we shall refer to the algorithm presented here as the *constant thermodynamic speed (CTS) algorithm.* Using eqn. (12)

$$C(t) = \frac{\sigma^2(\pi(t))}{\epsilon^2(t)}\,, \tag{16}$$

where

$$\sigma^2(\pi(t)) = \int (\rho(\mathbf{x}, \mathbf{y}) - \rho(\pi(t)))^2 \pi(\mathbf{z}, t) d\mathbf{z}\,, \tag{17}$$

and using

$$\frac{d\rho(\pi(t))}{dt} = C(t)\frac{d\epsilon(t)}{dt}\,, \tag{18}$$

eqn. (13) can be expressed as

$$\rho(\pi(t)) = \rho(\mu(t)) - v\sigma(\pi(t))\,, \tag{19}$$

that is, the equilibrium mean energy is kept a constant multiple of the equilibrium standard deviation below the current system's mean energy. Assuming that $\sigma(\mu(t))$ doesn't vary a lot we approximate $\sigma(\pi(t))$ by $\sigma(\mu(t))$. Using (19) we can then determine the time dependence of $\rho(\pi(t))$ from $\mu(t)$. The time dependence of $\epsilon(t)$ is then derived from (16) and (18):

$$\frac{d\epsilon(t)}{dt} = \frac{\epsilon^2(t)}{\sigma^2(\pi(t))}\frac{d\rho(\pi(t))}{dt} \approx \frac{\epsilon^2(t)}{\sigma^2(\mu(t))}\frac{d\rho(\pi(t))}{dt}\,. \tag{20}$$

Additionally to the adaptation of ϵ we may also adapt the jump distribution k. We suggest to use a Gaussian distribution whose covariance is adapted to the covariance, Σ, of $\mu(t)$ w.r.t. Θ space, according to

$$k = \beta\Sigma + s\mathbb{1}\,. \tag{21}$$

The small multiple of the identity in eq. (21) is added to prevent the process from degenerating, and β is a tuning parameter of the algorithm.

In order to initialize the algorithm, we draw a population from (12), for a relatively large $\epsilon = \epsilon_0$, using a rejection technique, i.e., we draw particles $(\boldsymbol{\theta}, \mathbf{x})$ from $f(\boldsymbol{\theta})f(\mathbf{x}|\boldsymbol{\theta})$ and accept them with probability $e^{-\rho(\mathbf{x},\mathbf{y})/\epsilon_0}$. In order to stay as close as possible to the time continuous process defined by (13) we should update a single particle (randomly chosen from the population) at a time according to (5) and update the mean fields $\rho(\mu(t))$, $\sigma(\mu(t))$ and $\Sigma(\mu(t))$ after each update, which can be done using recursion relations as shown in eqs. (25) through (28).

There is always a small fraction of particles that get stuck on their way towards the target. Therefore, for certain applications, it might be helpful to insert a *resampling step* every once in a while. However, resampling means a loss of effective sample size. Therefore, the population needs to be given enough time between two resampling steps to recover. E.g., a resampling step can be made after a sufficiently large number of accepted updates. In a resampling step, we draw a new population from the old with weights

$$\exp\left(\frac{-\rho(\mathbf{x}_i, \mathbf{y})\delta}{\epsilon(t)}\right) , \tag{22}$$

where δ is a (small) tuning parameter. After this step, the new population is a sample from the distribution

$$\tilde{\mu}(\mathbf{z}) = \mu(\mathbf{z})e^{-\rho(\mathbf{x},\mathbf{y})\delta/\epsilon} .$$

To first order in δ,

$$\rho(\tilde{\mu}) \approx \rho(\mu) - \frac{\delta}{\epsilon}\sigma^2(\mu) . \tag{23}$$

Maintaining (19) we define $\tilde{\epsilon}$ such that

$$\rho(\pi_{\tilde{\epsilon}}) = \rho(\tilde{\mu}) - \upsilon\sigma(\pi_{\tilde{\epsilon}}) .$$

Neglecting terms of order $\mathcal{O}(\upsilon\delta)$ and $\mathcal{O}(\delta^2)$, and using (18) and (23), we then find that

$$\tilde{\epsilon} \approx \epsilon(1 - \delta) . \tag{24}$$

Thus, we suggest the following algorithm:

1. Initialization of the algorithm:

   (a) Repeat the following steps until a population of N particles is obtained:

      i. Draw a parameter vector, $\boldsymbol{\theta}$, from the prior.

      ii. Draw an output, $\mathbf{x}$, from the likelihood $f(\mathbf{x}|\boldsymbol{\theta})$.

      iii. Accept the particle $(\boldsymbol{\theta}, \mathbf{x})$ with probability

      $$e^{-\rho(\mathbf{x},\mathbf{y})/\epsilon_0} ,$$

      for a sufficiently large ϵ_0.

   (b) Set the initial mean particle distance of the equilibrium, ρ_0, equal to the average particle distance $\bar{\rho}$:

   $$\rho_0 = \bar{\rho} = \frac{1}{N}\sum_{i=1}^{N} \rho(\mathbf{x}_i, \mathbf{y}) .$$

(c) Calculate the standard deviation of the particles from the target

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(\rho(\mathbf{x}_i, \mathbf{y}) - \bar{\rho})^2}\,.$$

(d) Set the initial ϵ according to eqn.

$$\epsilon = \epsilon_0 \left(1 - \frac{\epsilon_0 v}{\sigma}\right)\,,$$

where v is a small tuning parameter (by default $v = 0.1$).

(e) Calculate the parameters' average and empirical covariance according to

$$\bar{\boldsymbol{\theta}} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\theta}_i\,,$$

and

$$\Sigma = \frac{1}{N-1}\left(\sum_{i=1}^{N}\boldsymbol{\theta}_i^T\boldsymbol{\theta}_i - N\bar{\boldsymbol{\theta}}^T\bar{\boldsymbol{\theta}}\right)\,.$$

(f) Initialize an acceptance counter

$$a = 0\,.$$

2. Iterate the following steps:

(a) Draw a random particle, $(\boldsymbol{\theta}_j, \mathbf{x}_j)$, from the population.

(b) Draw an independent proposal parameter from the normal distribution

$$\boldsymbol{\theta}^* \sim N(\boldsymbol{\theta}_j, k)\,,$$

where k is calculated according to (21).

(c) Draw a proposal output, $\mathbf{x}^*$, from the likelihood $f(\mathbf{x}|\boldsymbol{\theta}^*)$.

(d) Draw a uniform random number r.

(e) If

$$r < \min\left(1, \exp\left(\frac{\rho(\mathbf{x}_j, \mathbf{y}) - \rho(\mathbf{x}^*, \mathbf{y})}{\epsilon}\right)\frac{f(\boldsymbol{\theta}^*)}{f(\boldsymbol{\theta}_j)}\right)$$

do the following updates:

i. Update the mean particle distance from the target

$$\bar{\rho}_{new} = \bar{\rho}_{old} + \frac{1}{N}(\rho(\mathbf{x}^*, \mathbf{y}) - \rho(\mathbf{x}_j, \mathbf{y}))\,. \tag{25}$$

ii. Update the variance of the distances

$$\sigma^2_{new} = \sigma^2_{old} + \frac{N}{N-1}(\bar{\rho}^2_{old} - \bar{\rho}^2_{new}) - \frac{1}{N-1}(\rho^2(\mathbf{x}_j, \mathbf{y}) - \rho^2(\mathbf{x}^*, \mathbf{y}))\,. \tag{26}$$

iii. Update the mean of the parameters

$$\bar{\boldsymbol{\theta}}_{new} = \bar{\boldsymbol{\theta}}_{old} + \frac{1}{N}(\boldsymbol{\theta}^* - \boldsymbol{\theta}_j)\,. \tag{27}$$

iv. Update the covariance of the parameters

$$\Sigma_{new} = \Sigma_{old} + \frac{N}{N-1}(\bar{\boldsymbol{\theta}}_{old}^T\bar{\boldsymbol{\theta}}_{old} - \bar{\boldsymbol{\theta}}_{new}^T\bar{\boldsymbol{\theta}}_{new}) + \frac{1}{N-1}((\boldsymbol{\theta}^*)^T\boldsymbol{\theta}^* - \boldsymbol{\theta}_j^T\boldsymbol{\theta}_j)\,. \tag{28}$$

v. Update the equilibrium particle distance according to eq. (19) as

$$(\rho_0)_{new} := \bar{\rho}_{new} - v\sigma_{new}\,. \tag{29}$$

vi. Update the tolerance according to eq. (13) as

$$\epsilon_{new} = \epsilon_{old} - \epsilon_{old}^2\frac{(\rho_0)_{old} - (\rho_0)_{new}}{\sigma_{new}^2}\,. \tag{30}$$

vii. Set $\boldsymbol{\theta}_j = \boldsymbol{\theta}^*$ and $\mathbf{x}_j = \mathbf{x}^*$.

viii. Increment the acceptance counter a.

(f) (optional) If $a = lN$, for a sufficiently large l (e.g. $l = 10$), draw a new population of size N from the old one with weights (22) and update the tolerance according to (24) as

$$\epsilon_{new} = \epsilon_{old}(1-\delta)\,.$$

Remark:

Implementing eqn. (20) does not guarantee that we stay close to equilibrium at all times. The risk to drift away from equilibrium naturally increases once the relaxation time $\eta(t)$ (which can be estimated from (15)) isn't small anymore compared to the observation time. If this happens, it might be a good idea to either stop the algorithm or then switch to constant ϵ.

2.4 Comparison with other algorithms

An important property of our algorithm is that, in the limit $N \to \infty$, the particles remain uncorrelated, if they were drawn independently at the beginning, even though the particles interact. This property is called *propagation of chaos* and is a well known consequence of the fact that the interaction is of *mean-field type* (see, e.g., [4]).

Whether or not our algorithm is to be preferred over the *Metropolis algorithm* depends on the "degree of stochasticity" of our model and the desired precision of the result: A model is said to have a high degree of stochasticity if it is much cheaper to draw a sample from the likelihood than to evaluate the likelihood function. In the limit of infinitely many particles, as we have just seen, our algorithm yields *independent* samples from (an approximation of) the posterior, whereas the sample generated from the Metropolis algorithm suffers from autocorrelation. On the other hand, contrary to Metropolis, the whole history of the particles in our population has to be discarded. Which of these (dis-)advantages is more dominant depends on the tuning of the algorithms and the desired sample size. However, no matter how well our algorithm is tuned, its acceptance rate decreases dramatically at a certain tolerance level.

A well tuned Metropolis algorithm, on the other hand, has a constant acceptance rate of $20 - 50\%$.

In our algorithm, the *sample size remains constant* (except possibly at the few resampling steps that were suggested in Subsect. 2.3). This is the main advantage compared to population methods based on *importance sampling*, such as [11] or [3], where each importance sampling step entails a reduction in effective sample size. Efficiency is further gained replacing $\chi(\epsilon - \rho(\mathbf{x}, \mathbf{y}))$ by $\exp(-\rho(\mathbf{x}, \mathbf{y})/\epsilon)$. With this replacement, moves are not only accepted if they end up in an ϵ-ball around the target but they are more likely accepted if they move *closer* to the target. Finally, our algorithm is of the order $\mathcal{O}(N)$, whereas importance sampling algorithms are of the order $\mathcal{O}(N^2)$, due to the weighting step. However, both algorithms scale like $\mathcal{O}(N)$ with the number of simulations from the likelihood, which is usually the most costly step.

3 Toy Example

In order to test the performance of our algorithm, we consider a case where the analytical equilibrium solution as a function of ϵ is available. The prior shall be given by a univariate normal distribution,

$$f(\theta) \propto \exp\left[-\frac{1}{2}\theta^2\right] . \tag{31}$$

The output, y, is assumed to be normally distributed around θ, and we assume to have n independent measurements, $\mathbf{y} = (y_1, \dots, y_n)$. Thus, the likelihood function reads as

$$f(\mathbf{y}|\theta) \propto \exp\left[-\frac{1}{2}\sum_{i=1}^{n}(y_i - \theta)^2\right] . \tag{32}$$

In order to be able to calculate the analytical solution, we set $\alpha = 2$ in (4), i.e., we set

$$\rho(\mathbf{x}, \mathbf{y}) = \frac{1}{2}\sum_{i=1}^{n}(x_i - y_i)^2 .$$

Then, the equilibrium solution (12) is given by the normal distribution

$$\pi_\epsilon(\theta, \mathbf{x}) = \pi_\epsilon(\mathbf{z}) \propto \exp\left[-\frac{1}{2}(\mathbf{z} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{z} - \boldsymbol{\mu})\right] , \tag{33}$$

with

$$\boldsymbol{\mu} = \frac{1}{\epsilon}\Sigma \begin{pmatrix} 0 \\ \mathbf{y} \end{pmatrix} \tag{34}$$

and

$$\Sigma^{-1} = \left(\begin{array}{c|c} (1+n) & -2\mathbf{1}^T \\ \hline -2\mathbf{1} & \frac{1+\epsilon}{\epsilon}\mathbb{1} \end{array}\right) . \tag{35}$$

A tedious but straightforward calculation shows that

$$\Sigma = \left(\begin{array}{c|c} (1+\epsilon)/(n+1+\epsilon) & \epsilon(1+\epsilon)n_\epsilon^{-1}\mathbf{1}^T \\ \hline \epsilon(1+\epsilon)n_\epsilon^{-1}\mathbf{1} & \lambda\mathbb{1} + \nu O \end{array}\right) , \tag{36}$$

where

$$\lambda = \epsilon(n+1+2\epsilon)n_\epsilon^{-1}\,, \quad \nu = \epsilon^2 n_\epsilon^{-1}\,, \quad n_\epsilon = n+1+\epsilon(n+2)+\epsilon^2\,,$$

and O denotes the matrix with zeroes on the diagonal and ones on the off-diagonals. From this and (34) we find that

$$\boldsymbol{\mu} = n_\epsilon^{-1}\begin{pmatrix} n(1+\epsilon)\bar{y} \\ (n+1)(1+\epsilon)\mathbf{y}\,. \end{pmatrix} \tag{37}$$

From this, we read off the posterior's mean

$$E_{\pi_0}(\Theta) = \frac{n\bar{y}}{n+1}\,, \tag{38}$$

and variance

$$\mathrm{Var}_{\pi_0}(\Theta) = \frac{1}{n+1}\,. \tag{39}$$

The mean particle distance is calculated from eq.

$$\langle\rho(\mathbf{x},\mathbf{y})\rangle_{\pi_\epsilon} = \frac{1}{2}\sum_{i=2}^{n+1}\left(\Sigma_{ii} + (\mu_i - y_{i-1})^2\right)\,. \tag{40}$$

Using (36) and (37)this yields

$$\langle\rho(\mathbf{x},\mathbf{y})\rangle_{\pi_\epsilon} = \frac{n\epsilon}{2n_\epsilon}\left(n+1+2\epsilon+\frac{1}{n}\sum_{i=1}^{n} y_i^2\epsilon(1+\epsilon)^2 n_\epsilon^{-1}\right)\,. \tag{41}$$

We spare the reader with the expression for the heat capacity $C(\epsilon)$ but comment on its important properties. For small values of $\sum_{i=1}^n y_i^2$, C is monotonously decreasing as a function of ϵ. Thus, it assumes its maximum at $\epsilon = 0$, namely

$$C(0) = \frac{n}{2}\,. \tag{42}$$

For $\sum_{i=1}^n y_i^2$ larger than a certain value, $C(\epsilon)$ starts forming a peak before it decays to zero, for $\epsilon \to \infty$.

In Figs. 1 and 2 we compare the convergence of three different schemes: the CTS scheme explained in Subsect. 2.3, the explicit scheme for which we have proven convergence in Subsect. 2.2, and an explicit scheme with a constant and small ϵ. For simplicity, we have chosen $y_i = y$. Furthermore, the population size was chosen to be $N = 1000$. For the explicit schedule, we've chosen, according to Theorem 2.1,

$$\epsilon(t) = \epsilon_0 t^{-1/10}\,, \tag{43}$$

because $\alpha = 2$ and $n = 20$, for an initial $\epsilon_0 = 2.7$. (Note that, for large N, we can approximate the time continuous process used in this section by a sequence of transitions (6) upon multipliying (5) by a small time step Δt.) For the constant scheme we have chosen $\epsilon(t) = 0.1$. The decay of $\epsilon(t)$, for these three schedules, is plotted in the left panel of Fig. 1. The explicit scheme (43) shows the slowest decay, whereas the tolerance of the constant scheme was chosen considerably smaller than the tolerance that is reached by the CTS after a long simulation time of $3000N$ particle updates. For this example, resampling was found not to lead to a significant increase in convergence speed of the CTS schedule and was thus not applied.

The right panel in Fig. 1 shows the mean energy (particle distance) as a function of ϵ. The solid black line corresponds to the equilibrium and shows the functional dependence of $\rho(\pi)$ from ϵ, as given by eq. (41). For the CTS schedule, the mean energy of $\mu(t)$ is always slightly above the mean energy of the target $\pi(t)$, as dictated by (19). The slope of the bold black curve is the heat capacity (14). A much simpler adaptive scheme [1] would be to set

$$\epsilon(t) = C_0^{-1} \rho(\mu(t)) \,, \tag{44}$$

where C_0 can be interpreted as a constant heat capacity. The schedule (44) is associated with a straight line in the right panel of plot 1. If we don't want the process to converge for an $\epsilon > 0$, we have to set C_0 equal to the maximum of the heat capacity, which means that, for $\sum_{i=1}^{n} y_i^2$ not too large, we have to set $C_0 = n/2$, according to (42). But then, at the beginning of the process, we would have a huge discrepancy between $\rho(\mu)$ and $\rho(\pi)$, and, therefore, potentially pick up an additional bias. Therefore, (44) is not advised whenever $\rho(\pi_\epsilon(t))$ is strongly non-linear as a function of ϵ. The explicit scheme (43) starts off slightly off equilibrium but then quickly reaches equilibrium and stays close to it, as we would expect given our convergence proof. The constant schedule is obviously very far from equilibrum, at least at the beginning of the algorithm. That this is still so at the end of the simulation period is revealed by the left panel of Fig. 2, which shows the convergence of the expectation value and the standard deviation of the θ-marginal of $\mu(t)$, respectively. The bold black lines show the corresponding values of the target distribution (12). For the explicit schedule (43) and the CTS, mean and standard deviation of $\mu(t)$ closely follow the corresponding values of $\pi(t)$. For a constant ϵ, however, still at the end of the simulation time, the expectation value of the marginal of $\mu(t)$ shows a bias that is much larger than the error given by a finite ϵ and a finite population size. The former is given by the upper solid black line in the left panel of Fig. (2) and the latter can be estimated as

$$\frac{\sigma_{\boldsymbol{\theta}}(\pi_\epsilon)}{\sqrt{N}} \approx 0.007 \,. \tag{45}$$

This demonstrates the effect a too fast cooling can have.

Finally, due to the relatively large output dimension $n = 20$, all schedules have a relatively large bias in the standard deviation, as revealed by the right panel of Fig. (2). This, however, is a problem inherent to all ABC schedules.

4 Conclusions

The interacting particle algorithm presented in this paper has several advantages compared to the other Approximate Bayes Computations the authors are aware of. Firstly, our algorithm is not based on importance sampling. Consequently, there is no loss of effective sample size due to re-sampling. Secondly, the Heaviside function $\chi(\epsilon - \rho(\mathbf{x}, \mathbf{y}))$ has been replaced by the smooth kernel $\exp(-\rho(\mathbf{x}, \mathbf{y})/\epsilon)$. Thus, moves are more likely to be accepted if they move closer to the target and not only if they move into an ϵ-ball around the target. Thirdly, our algorithm not only adapts the jump distribution in parameter space but also the tolerance on the output space. The tuning thus restricts to the two parameters v and β (apart from the resampling parameters δ and l). Since these adaptations can be interpreted as a mean-field interaction between the particles, the particles remain statistically independent if they were drawn independently at the beginning, in the limit of an infinite sample size.

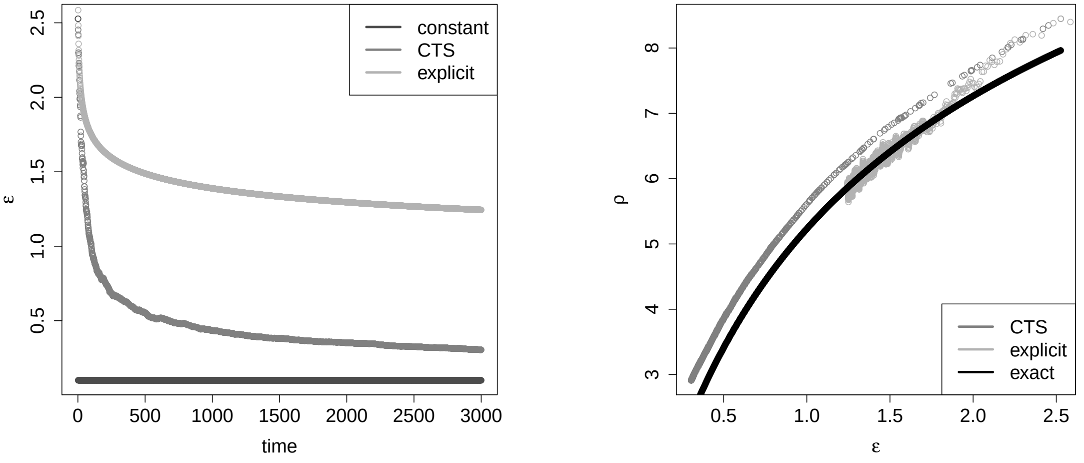

Figure 1: The left panel shows the decay of ϵ, as a function of time, for three different cooling schedules. The right panel shows the mean energy, as a function of ϵ, for two different cooling schedules. The black line in the right panel shows the mean energy of the equilibrium solution, as a function of ϵ. We set $n = 20$ and $y = 0.5$.

The disadvantage of the adaptive algorithm presented here is that it is based on a heuristic designed to keep the population close to equilibrium at all times, but does not guarantee convergence. Dragging along all the particles might lead to the disadvantage that we invest too much computation into a small fraction of outliers that got stuck on their way towards the target. This can be remedied employing a resampling step every once in a while.

The biggest disadvantage inherent to all ABC algorithms is that the tolerance leads to a bias that grows with the dimension of the output space n. Therefore, it is important to use *summary statistics* to reduce the output dimension or employ *local approximations of the likelihood*, for ABC to be useful for problems with large output dimensions.

References

[1] C. Albert. An interacting particle method for approximate bayes computations. *NOLTA Conference Proceedings*, 2011.

[2] B. Andresen and J.M. Gordon. Constant thermodynamic speed for minimizing entropy production in thermodynamic processes and simulated annealing. *Phys. Rev. E*, 50(6):4346–4351, 1994.

[3] M. A. Beaumont, J.M. Cornuet, J.M. Marin, and C. P. Robert. Adaptive approximate Bayesian computation. *Biometrika*, 96(4):983–990, 2009.

[4] D. Burkholder, E. Pardoux, and A. Sznitman. Topics in propagation of chaos. In *Ecole d'Ete de Probabilites de Saint-Flour XIX — 1989*, volume 1464 of *Lecture Notes in Mathematics*, pages 165–251. Springer Berlin / Heidelberg, 1991. 10.1007/BFb0085169.

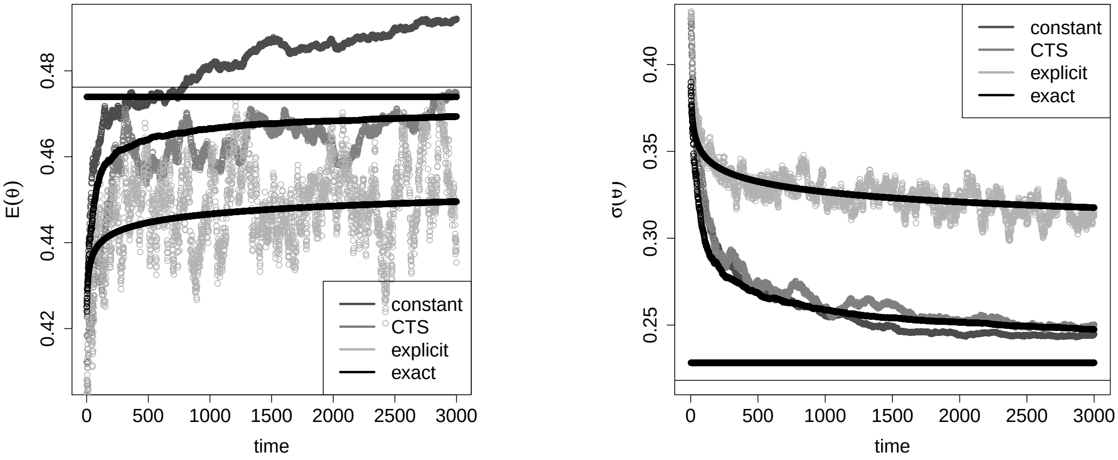

Figure 2: These plots show the convergence of the mean and the standard deviation of of the θ-marginal of $\mu(t)$, for three different cooling schedules. The three bold black lines in each plot show the mean and the standard deviation of the corresponding equilibrium $\pi(t)$. The CTS and the explicit schedule follow their respective equilibrium solutions, while the schedule with constant ϵ is obviously far off equilibrium, even at the end of the simulation period. The thin black lines indicate mean and standard deviation of the posterior, i.e. the equilibrium solution for $\epsilon = 0$.

[5] H. Föllmer. Random fields and diffusion processes. In *Ecole d'Ete de Probabilites de Saint-Flour XV–XVII, 1985–87*, volume 1362 of *Lecture Notes in Mathematics*, pages 101–203. Springer Berlin / Heidelberg, 1988.

[6] C. Leuenberger and D. Wegmann. Bayesian computation and model selection without likelihoods. *Genetics*, 184(2):243–252, 2010.

[7] P. Marjoram, J. Molitor, V. Plagnol, and S. Tavaré. Markov chain monte carlo without likelihoods. *Proc. Natl. Acad. Sci. U.S.A.*, 100(2):15324–15328, 2003.

[8] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *J. Chem. Phys.*, 21(6):1087–1092, 1953.

[9] G. Ruppeiner, Pedersen J.M., and Salamon P. Ensemble approach to simulated annealing. *J. Phys. I*, 1:455–470, 1991.

[10] S. Tavaré, D.J. Balding, R.C. Griffiths, and P. Donnelly. Inferring Coalescence Times From DNA Sequence Data. *Genetics*, 145:505–518, 1997.

[11] T. Toni, D. Welch, N. Strelkowa, A. Ipsen, and M. P. H. Stumpf. Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems. *J. R. Soc. Interface*, 6(31):187–202, 2009.

[12] G. Weiss and A. Haeseler. Inference of population history using a likelihood approach. *Genetics*, 149:1539–1546, 1998.