This document is the accepted manuscript version of the following article:
Albert, C., Künsch, H. R., & Scheidegger, A. (2015). A simulated annealing approach to approximate Bayes computations. Statistics and Computing, 25(6), 1217-1232. https://doi.org/10.1007/s11222-014-9507-8

A Simulated Annealing Approach to Approximate Bayes Computations

Carlo Albert[*], Hans R. Künsch[†] and Andreas Scheidegger[*]

August 22, 2018

Abstract

Approximate Bayes Computations (ABC) are used for parameter inference when the likelihood function of the model is expensive to evaluate but relatively cheap to sample from. In particle ABC, an ensemble of particles in the product space of model outputs and parameters is propagated in such a way that its output marginal approaches a delta function at the data 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 data. 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 and the jump distribution in parameter space according to some mean-fields of the ensemble, which preserves the statistical independence of the particles, in the limit of infinite sample size. This adaptive scheme aims at converging as close as possible to the correct result with as few system updates as possible via minimizing the entropy production of the process. The performance of this new class of algorithms is compared against two other recent algorithms on two toy examples as well as on a real-world example from genetics.

1 Introduction

One way of implementing parameter inference in the Bayesian framework is to generate parameter 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, $f(\mathbf{y}|\boldsymbol{\theta})$ is the *likelihood function*, that is, the probability density of outputs given the parameter vector $\boldsymbol{\theta}$, evaluated at the measurement vector (data) $\mathbf{y}$, and $f(\mathbf{y})$ is the corresponding prior density of the data. Numerical methods such as the *Metropolis* algorithm [15] 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 have been suggested

[*]Eawag, aquatic research, 8600 Dübendorf, Switzerland.
[†]Seminar für Statistik, ETH Zürich, 8092 Zürich, Switzerland.

that generate samples from (1) by *sampling model outputs* from the likelihood and comparing them with the data rather than evaluating the likelihood.

As far as we know, the origin of these algorithms is to be found in population genetics. Tavaré et al. [24] 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. [26] 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. [14] 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. [25] suggested using a decreasing sequence of tolerances and letting an ensemble 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 ensemble from the old one with weights and subsequent re-sampling. This re-sampling leads to a loss of effective sample size at each iteration step. There are several adaptive versions of ensemble (or particle) ABC algorithms. Beaumont et al. [2] use the empirical variances of the ensemble to adapt the jump distribution in parameter space. Del Moral et al. [5] and Lenormand et al. [11] use the particles' distance from the target to adapt the tolerance. Recent variants of the algorithm of del Moral et al appeared in [10] and [20]. 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. The effect of kernels different from the Heaviside function has been considered, e.g., in [27]. For a recent review on ABC algorithms, the reader is referred to [13].

In this paper, we present a new class of (adaptive) ensemble 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 an ensemble 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. 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. This problem is reminiscent of Simulated Annealing, which is one of our sources of inspiration. We will give a convergence proof for a schedule that satisfies

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

where n is the dimension of the output space and $\alpha > 0$ is defined in (4). Furthermore, we will present an adaptive schedule that attempts convergence to the correct posterior while minimizing the required simulations from the likelihood. Both the jump distribution in parameter space and the tolerance ϵ are adapted using mean fields of the ensemble.

The adaptation of ϵ we suggest is motivated from non-equilibrium thermodynamics, where this control parameter is naturally interpreted as a temperature. We adapt ϵ according to the particles' distance to the target (energy) in such a way that the entropy production in

the system, which is a measure for the waste of computation, is minimized. A first order approximation of the entropy production is calculated using the so-called *endoreversibility assumption*, which states that the system undergoes only reversible changes, and which is approximately satisfied if either the mixing in parameter space is fast enough or annealing is slow enough. Under this assumption the only source of entropy production is the flow of energy (or rather heat) from the system to the environment, the latter being defined by the control parameter ϵ that is used for the transitions and can be interpreted as the temperature of a heat reservoir the system is in contact with. In cases where the influence of the prior on the posterior is strong, we actively control this prior influence with a second control parameter, which allows us to extend the scope of the endoreversibility assumption. Necessary and sufficient conditions for the minimization of entropy production, for endoreversible processes, have been derived in [22]. For sufficiently slow processes, for which a linearity assumption holds, the condition is a *constant entropy production rate* [19], which has been applied to Simulated Annealing, e.g., in [18]. In cases where the prior influence on the posterior is small, we go beyond the linearity assumption and suggest a scheme with non-constant entropy production rate.

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., [7], [12]).

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. Sect. 3 contains an application to two toy models, for which the posterior is available analytically, as well as a comparison with two recent adaptive ABC algorithms [5], [11]. Sect. 4 contains an application in genetics. Conclusions are drawn in Sect. 5.

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 behind 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, $\Theta \times X$, which means to sample a parameter vector from the prior and an associated output from the likelihood and accept the particle iff the drawn output happens to coincide with the data. 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., [7], [24] and [26]).

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 in Subsects. 2.2 and 2.3.2 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 after, possibly, an initial burn-in period, which guarantees convergence. However, this is clearly inefficient. On the other hand, a too fast decrease may result in slow convergence (because the acceptance probability decreases for decreasing ϵ) or convergence to a biased result. A bias can occur, e.g., if the prior within the last factor in eq. (5) decides too seldom whether a proposal point in $\Theta \times X$ is accepted or not. In the extreme case of a constant $\epsilon = 0$, the acceptance term in (5) becomes $\chi(\rho(\mathbf{x}', \mathbf{y}) - \rho(\mathbf{x}, \mathbf{y}))$, thus, $(\boldsymbol{\theta}, \mathbf{x})$ is accepted iff $\rho(x, y) \leq \rho(x', y)$. Hence in this case, the prior has no influence, which clearly leads to convergence to a biased result. For this reason, in Subsect. 2.3.3, we will introduce a second control parameter to control the influence of the prior and replace (5) by (35).

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 $\mathbf{y} = \mathbf{0}$ 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 $\alpha > 0$ 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 $\hat{\pi}_0(\mathbf{z}) \propto f_{post}(\boldsymbol{\theta}|\mathbf{y})\exp(-\rho(\mathbf{x}))$, for $k \to \infty$.

Proof: We will apply corollary (2.34) in [8]. We start by introducing some notation. Let

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

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

By assumption (A3) and dominated convergence,

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

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 [8]. 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.

□

Remark: Convergence of inhomogeneous Markov chains has been proved in much more general settings than in [8], see e.g. [6], or Proposition A.1 in [3]. Using these techniques, it should be possible to relax the assumptions (A1)–(A2).

2.3 An adaptive scheme

2.3.1 Heuristics

As stated in Section 2.1, we construct an ensemble of particles which evolve according to a family of Markov transition kernels (P_ϵ) with a control parameter $\epsilon = \epsilon^e(t)$ that decreases to zero (the reason for the notation $\epsilon^e(t)$ will become clear later). In contrast to the algorithms in [5] and [11], we do not use importance sampling to force the distribution of the ensemble to agree with the target distribution (3) at certain time points. This has the advantage that the effective sample size of the ensemble does not decrease over time, but the disadvantage that we loose control over the transient distribution of the stochastic process defined by the algorithm. However, as Theorem 2.1 suggests, this distribution remains close to an equilibrium (3) at all times, if either the value of the control parameter $\epsilon^e(t)$ is lowered sufficiently slowly or if mixing in parameter space is sufficiently fast. In this section, we shall design an algorithm that adapts $\epsilon^e(t)$ based on the average distance of the particles from the target $\mathbf{y} = \mathbf{0}$ in such a way that the computational effort, that is the number of draws from the likelihood, is minimized. There is therefore a mean field interaction between particles.

The design of the algorithm will rely on the *assumption* that the distribution of the Markov chain is at all times t close to an equilibrium distribution $\pi_{\epsilon(t)}$, but with a parameter $\epsilon(t)$ which is somewhat higher than the value $\epsilon^e(t)$ used for the transition. How quickly we let $\epsilon^e(t)$ go to zero as the algorithm proceeds is our decision, and it determines together with the jump distribution $k(\boldsymbol{\theta}, \boldsymbol{\theta}')$ in parameter space the function $\epsilon(t)$. We have no analytical expression for $\epsilon(t)$, but it is in a one-to-one correspondence with the expected distance, $U(t)$, from the target, that we can estimate.

Since the intuition behind our adaptive algorithm stems from non-equilibrium thermodynamics, it might be helpful to imagine a gas, which is in contact with a heat bath whose *temperature* $\epsilon^e(t)$ can be controlled. The value of $\epsilon(t)$ is then the temperature of the gas at time t which is measured continuously and influences how quickly $\epsilon^e(t)$ is lowered. The superscript e stands for *"environment"* or *"equilibrium"*, because it defines the equilibrium state the system would relax to if cooling suddenly stopped, that is if $\epsilon^e(t)$ would be kept constant after some time t_0. However, if the temperature $\epsilon^e(t)$ is continuously lowered, then the gas will at any time t be warmer than the environment. In the physics community, a system which is always described by an equilibrium distribution even if it is externally driven, i.e., never at equilibrium with its environment, is called *endoreversible* (see [17]). The system is then described by the Gibbs state $\pi_{\epsilon(t)}$, and the distance of a particle to the target is interpreted as the particle's *energy*.

The question is then how $\epsilon^e(t)$ should be controlled, depending on the distribution of the system given by $\epsilon(t)$ or $U(t)$, so as to waste as little computation as possible. In physics'

terms, the cooling of the system by lowering the temperature of the environment creates a *flow of entropy* from the system to the environment. It can be split into two parts. One part is the (path-independent) reduction of the system's entropy. This is the well invested part of the computing effort, as it measures the information difference between prior and posterior. The other part is the *entropy production*, which is a measure for the wasted computing effort. We argue therefore that we have to choose the cooling or annealing schedule $\epsilon^e(t)$ such that this entropy production is minimized. Using variational calculus [22], this approach leaves us with a family of annealing schedules, parameterized by a tuning parameter v, which governs the annealing speed and expresses the optimal $\epsilon^e(t)$ in function of the expected distance $U(t)$ from the target.

In mathematical terms, entropy production equals the *Shannon entropy* of the probability distribution of the process, on the space of paths, relative to the time-inverted stochastic process [21]. It can be seen as a measure for the information loss due to rejections: With a fast cooling schedule, a typical path is likely to encounter many more rejections than a typical path under the time-reversed (heating) schedule. Thus, the probability distribution of the stochastic process on the space of all paths is in this case much more concentrated than the distribution of the reverse process, which leads to a large relative entropy.

The assumption of endoreversibility is crucial for our algorithm. If it is violated, there will be additional production of entropy due to irreversible processes. This entropy production is beyond control if only the energy of the system is measured and is thus to be avoided. Such additional entropy can even remain in the system indefinitely and lead to a biased convergence. Whether the assumption of endoreversibility is justified or not will depend on the values of the two tuning parameters of the algorithm: The covariance K of the jump distribution $k(\boldsymbol{\theta}, \boldsymbol{\theta}')$ in parameter space and the tuning parameter v that arises from variational calculus. A small value of K leads to slow mixing in parameter space, and if it is too slow compared to the decay rate of $\epsilon(t)$, the endoreversibility assumption might be violated. We derive our cooling schedule under the assumption that K is constant, but in practice it is usually advantageous to adapt K to the current distribution of the chain. We will discuss the adaptive choice of K at the end of the next subsection.

Similarly, a too large v can lead to a too fast cooling, compared to the mixing in parameter space, which bears the risk of violating the endoreversibility assumption. On the other hand, a too small value of v leads to a large amount of reversible computations which is not accounted for by the entropy production. Running the algorithm at equilibrium, i.e. setting $\epsilon^e(t) = \epsilon(t)$, does neither lead to a flow of entropy nor does it generate any entropy and would thus be considered optimal by our criterion. Because v has the dimension of an inverse time, measured in units of N computer updates of single particles, its optimal value is expected to depend not too much on details of the model.

The problem of choosing a good value of v is pronounced if the prior, $f(\boldsymbol{\theta})$, carries relevant information. Since $\epsilon^e(t)$ is by construction of the algorithm smaller than the value $\epsilon(t)$ implied by the expected distance from the target, the $\rho(\mathbf{x}, \mathbf{y})$-dependent term in (5) tends to decide more often than the prior-dependent term whether or not to accept a move. This may lead to an under-representation of the prior in the final solution. Thus, if the prior is important, we suggest introducing a second pair of "temperatures" $\epsilon_2^e(t)$ and $\epsilon_2(t)$, in order to control the relative information contributed by the prior. Obviously, in this case, tuning will become much more sophisticated and we only derive an optimal schedule for relatively slow annealing, in which case the relation between forces and fluxes (to be defined below) is approximately linear. The more information the prior contains, however, the less advantageous a sequential

scheme as ours appears compared to a brute-force acceptance/rejection algorithm. Therefore, we devote the next subsection to the simpler, yet practically relevant, case of negligible prior information.

2.3.2 The case of negligible prior information

In this subsection we consider the special case where the prior $f(\boldsymbol{\theta})$ doesn't play much of a role. This is the case if $f(\boldsymbol{\theta}) \approx \text{const}$, in the area where the likelihood function, evaluated at the data $\mathbf{y}$, is not negligible.

Our *system* is an inhomogeneous continuous time Markov process $(\mathbf{Z}_t)$ on the product space of parameters and model outputs. Its transitions occur at the random times of a Poisson process with rate 1, according to the transition kernel (5) with time dependent parameter $\epsilon^e(t)$. This means that the density $\mu(\mathbf{z}, t)$ of the system at time t satisfies

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

and for functions h on the product space we have

$$\frac{dE(h(\mathbf{Z}_t))}{dt} = \int (h(\mathbf{z}) - h(\mathbf{z}')) \mu(\mathbf{z}', t) q_{\epsilon^e(t)}(\mathbf{z}', \mathbf{z}) d\mathbf{z} d\mathbf{z}' \,. \tag{12}$$

The parameter ϵ^e which controls the cooling of the system is adaptive in the sense that $\epsilon^e(t)$ depends on the distribution $\mu(\mathbf{z}, t)$. In our algorithm, we will represent the system by a sufficiently large *ensemble*, E, of particles, $\{\mathbf{z}_i = (\boldsymbol{\theta}_i, \mathbf{x}_i)\}_{i=1}^N$ which evolve in time. Each system update consists in choosing a random member of the ensemble and updating it according to the transition kernel (5). The parameter $\epsilon^e(t)$ is then based on the current empirical distribution of the ensemble at time t.

As discussed above we will assume the process to satisfy the endoreversibility assumption

$$\mu(\mathbf{z}, t) \approx \pi_{\epsilon(t)}(\mathbf{z}) \,, \tag{13}$$

where $\pi_\epsilon(\mathbf{z})$ was defined in (3). As we have discussed the legitimacy of this assumption in the previous subsection, we take it for granted here. The system's temperature $\epsilon(t)$ is in one-to-one correspondence with the system's energy which is the system's expected distance to the target. It will be measured by the average distance of the particles from the target.

We derive now our algorithm for the choice of the cooling schedule $\epsilon^e(t)$ in a sequence of steps. In the first step we modify distance by a monotone transformation to get approximate equality of energy and temperature. We define

$$u(\mathbf{x}) = G(\rho(\mathbf{x})), \quad G(\rho) = \int_{\rho(\mathbf{x}) \le \rho} f(\boldsymbol{\theta}, \mathbf{x}) d\boldsymbol{\theta} d\mathbf{x} \,, \tag{14}$$

and we replace $\rho(\mathbf{x})$ by $u(\mathbf{x})$ in the definitions of π_ϵ and q_ϵ. Because G is the cumulative distribution function of $\rho(\mathbf{x})$ under the prior $f(\mathbf{x}, \boldsymbol{\theta})$, we obtain, for the mean energy under π_ϵ,

$$U(\epsilon) := \int u(\mathbf{x}) \pi_\epsilon(\boldsymbol{\theta}, \mathbf{x}) d\boldsymbol{\theta} d\mathbf{x} \,, \tag{15}$$

the expression

$$U(\epsilon) = \frac{\int_0^\infty G(\rho) e^{-G(\rho)/\epsilon} G'(\rho) d\rho}{\int_0^\infty e^{-G(\rho)/\epsilon} G'(\rho) d\rho} = \frac{\int_0^1 u e^{-u/\epsilon} du}{\int_0^1 e^{-u/\epsilon} du} = \epsilon \frac{1 - e^{-1/\epsilon}(1 + 1/\epsilon)}{1 - e^{-1/\epsilon}} \,. \tag{16}$$

As ϵ goes to zero, the fraction on the right is $1+o(\epsilon^k)$ for any $k>0$. By the endoreversibility assumption we therefore have

$$U(t) := \int u(\mathbf{x})\mu(\mathbf{z},t) \approx U(\epsilon(t)) \approx \epsilon(t)\,. \tag{17}$$

Our main result of this section, equation (31) below, expresses the optimal cooling schedule $\epsilon^e(t)$ as a function of $U(t)$ and a tuning parameter v of the algorithm. In order to estimate $U(t)$ we need first an approximation of the distribution function G which we construct at the beginning of the algorithm, based on the prior sample, P, that is drawn to get the initial ensemble E. If the sample size of P is not large enough or if we want to run the algorithm for a very long time, we might want to use a smooth approximation of the empirical distribution of the values $\rho(\mathbf{x}_i)$, which, for small ρ, and for $\alpha=2$ in (4), behaves as

$$G(\rho) \approx \text{const}\,\rho^{n/2}\,. \tag{18}$$

In the second step, we approximate $\dot{U}(t) = \frac{d}{dt}U(t)$, the so-called flux, as a function of $\epsilon(t)$ and $\epsilon^e(t)$. For this, we cannot use directly $U(t)\approx U(\epsilon(t))$ because then the dependence on $\epsilon^e(t)$ would be lost. Combining the endoreversibility assumption with the time evolution (12) gives

$$\begin{aligned}
\dot{U}(t) &= \int (u(\mathbf{x})-u(\mathbf{x}'))k(\boldsymbol{\theta}',\boldsymbol{\theta})f(\mathbf{x}|\boldsymbol{\theta})\min\left(1,\frac{f(\boldsymbol{\theta})\exp(-u(\mathbf{x})/\epsilon^e(t))}{f(\boldsymbol{\theta}')\exp(-u(\mathbf{x}')/\epsilon^e(t))}\right)\mu(\mathbf{z}',t)d\mathbf{z}d\mathbf{z}' \\
&\approx Z^{-1}(\epsilon(t))\int (u(\mathbf{x})-u(\mathbf{x}'))k(\boldsymbol{\theta}',\boldsymbol{\theta})f(\mathbf{x}|\boldsymbol{\theta})f(\mathbf{x}'|\boldsymbol{\theta}') \\
&\quad \times \min\left(f(\boldsymbol{\theta}'), f(\boldsymbol{\theta})\frac{\exp(-u(\mathbf{x})/\epsilon^e(t))}{\exp(-u(\mathbf{x}')/\epsilon^e(t))}\right)\exp(-u(\mathbf{x}')/\epsilon(t))d\mathbf{z}d\mathbf{z}'\,.
\end{aligned} \tag{19}$$

Since ϵ (and thus also ϵ^e) will be much smaller than 1 during most of the process, we use a Taylor expansion of (19) to quadratic order in ϵ and ϵ^e. Under the assumption that the influence of the prior is negligible, we obtain

$$\dot{U}(t) \approx \dot{U}(\epsilon,\epsilon^e) \approx -\gamma(\epsilon^2-(\epsilon^e)^2)\,, \tag{20}$$

with

$$\gamma = (f(\mathbf{y}))^{-2}\int k(\boldsymbol{\theta}',\boldsymbol{\theta})f(\mathbf{y},\boldsymbol{\theta})f(\mathbf{y},\boldsymbol{\theta}')d\boldsymbol{\theta}d\boldsymbol{\theta}'\,. \tag{21}$$

For later use, we note that from $U(t)\approx\epsilon(t)$ we obtain

$$\epsilon^e(t) \approx \sqrt{U(t)^2+\dot{U}(t)/\gamma}\,. \tag{22}$$

In the third step we approximate the derivative of the irreversible process entropy or entropy production. To simplify the notation, let us begin with the version in discrete time where we have an initial distribution μ_0 and a sequence of transition kernels P_i corresponding to a sequence ϵ_i^e of control parameters. The probability of a path $\Gamma_n=(\mathbf{z}_0,\mathbf{z}_1,\ldots,\mathbf{z}_{n-1})$ is then

$$p(\Gamma_n) = \mu_0(\mathbf{z}_0)P_0(\mathbf{z}_0,\mathbf{z}_1)\ldots P_{n-2}(\mathbf{z}_{n-2},\mathbf{z}_{n-1})\,, \tag{23}$$

whereas the probability of the same path with respect to the time-reverse schedule is

$$p^R(\Gamma_n) = \mu_{n-1}(\mathbf{z}_{n-1})P_{n-2}(\mathbf{z}_{n-1},\mathbf{z}_{n-2})\ldots P_0(\mathbf{z}_1,\mathbf{z}_0)\,, \tag{24}$$

where

$$\mu_{n-1}(\mathbf{z}_{n-1}) = \int p(\Gamma_n) d\mathbf{z}_0 \cdots d\mathbf{z}_{n-2}$$

is the distribution of the final state. The irreversible process entropy is then defined as the relative entropy of p^R with respect to p, see [21],

$$S_{irr}(n) = \int p(\Gamma_n) \ln \frac{p(\Gamma_n)}{p^R(\Gamma_n)} d\Gamma_n \,. \tag{25}$$

From this it follows easily that

$$\begin{aligned} S_{irr}(n+1) &= S_{irr}(n) \\ &+ \int \log \left(\frac{\mu_{n-1}(\mathbf{z}_{n-1}) P_{n-1}(\mathbf{z}_{n-1}, \mathbf{z}_n)}{\mu_n(\mathbf{z}_n) P_{n-1}(\mathbf{z}_n, \mathbf{z}_{n-1})} \right) \mu_{n-1}(\mathbf{z}_{n-1}) P_{n-1}(\mathbf{z}_{n-1}, \mathbf{z}_n) d\mathbf{z}_{n-1} d\mathbf{z}_n \,. \end{aligned} \tag{26}$$

Passing to a continuous time limit, we therefore obtain from (11)

$$\begin{aligned} \dot{S}_{irr}(t) &= \int \log \left(\frac{q_{\epsilon^e(t)}(\mathbf{z}, \mathbf{z}')}{q_{\epsilon^e(t)}(\mathbf{z}', \mathbf{z})} \right) \mu(\mathbf{z}, t) q_{\epsilon^e(t)}(\mathbf{z}, \mathbf{z}') d\mathbf{z} d\mathbf{z}' - \frac{d}{dt} \int \log(\mu(\mathbf{z}, t)) \mu(\mathbf{z}, t) d\mathbf{z} \\ &= \int \log \left(\frac{q_{\epsilon^e(t)}(\mathbf{z}, \mathbf{z}')}{q_{\epsilon^e(t)}(\mathbf{z}', \mathbf{z})} \right) \mu(\mathbf{z}, t) q_{\epsilon^e(t)}(\mathbf{z}, \mathbf{z}') d\mathbf{z} d\mathbf{z}' - \int \log(\mu(\mathbf{z}, t)) \frac{\partial \mu(\mathbf{z}, t)}{\partial t} d\mathbf{z} \\ &= \int \log \left(\frac{q_{\epsilon^e(t)}(\mathbf{z}, \mathbf{z}') \mu(\mathbf{z}, t)}{q_{\epsilon^e(t)}(\mathbf{z}', \mathbf{z}) \mu(\mathbf{z}', t)} \right) \mu(\mathbf{z}, t) q_{\epsilon^e(t)}(\mathbf{z}, \mathbf{z}') d\mathbf{z} d\mathbf{z}' \,. \end{aligned} \tag{27}$$

Using the endoreversibility assumption and the expression (5) for q_ϵ, we arrive at

$$\begin{aligned} \dot{S}_{irr}(t) &= \int \mu(\mathbf{z}, t) q_t(\mathbf{z}, \mathbf{z}') (u(\mathbf{z}') - u(\mathbf{z})) \left(\frac{1}{\epsilon(t)} - \frac{1}{\epsilon^e(t)} \right) d\mathbf{z} d\mathbf{z}' \\ &= \left(\frac{1}{\epsilon(t)} - \frac{1}{\epsilon^e(t)} \right) \frac{d}{dt} \int u(\mathbf{z}) \mu(\mathbf{z}, t) d\mathbf{z} = F(t) \dot{U}(t) \,, \end{aligned} \tag{28}$$

where

$$F(t) = \epsilon(t)^{-1} - \epsilon^e(t)^{-1} \tag{29}$$

is the thermodynamic force, the difference between the inverse temperatures of the system and the environment. Because of (17) and (22) $F(t)$ is a function of $U(t)$ and $\dot{U}(t)$,

In the fourth step, we determine the necessary and sufficient criterion for minimal entropy production, for fixed initial and final values of the energy:

$$\int_0^{t_f} F(U(t), \dot{U}(t)) \dot{U}(t) dt = \min!, \quad U(0) = U_0, \quad U(t_f) = U_f.$$

Using standard methods of variational calculus, see [22], one obtains the differential equation

$$\dot{U} \frac{\partial F}{\partial \dot{U}} \dot{U} = \text{const} = v \,. \tag{30}$$

From (22) it follows that

$$\frac{\partial F}{\partial \dot{U}} = -\frac{\partial \epsilon^e(t)^{-1}}{\partial \dot{U}} \approx \frac{\gamma^{1/2}}{2(\gamma U(t)^2 + \dot{U}(t))^{3/2}} \approx \frac{1}{2\gamma \epsilon^e(t)^3}.$$

If we combine this result with (20) we find the optimal cooling schedule, for small U, to be approximated by the unique solution of the quartic equation

$$\frac{(U(t)^2 - \epsilon^e(t)^2)^2}{2\epsilon^e(t)^3} = \frac{v}{\gamma} \tag{31}$$

in the interval $(0, U(t))$. It can be computed efficiently with the Newton algorithm. The leading term of the solution $\epsilon^e(U)$, for small U, is

$$\epsilon^e(U) = \Big(\frac{\gamma}{2v}\Big)^{1/3} U^{4/3} + \mathcal{O}(U^2)\,. \tag{32}$$

This means that the cooling is slowing down when $U(t)$ gets small. One can derive from this also an explicit cooling schedule,

$$\epsilon^e(t) \sim t^{-4/3}\,, \tag{33}$$

but this will not be used in our algorithm. It shows however that the cooling schedule which follows from Theorem 2.1 is different from the adaptive schedule here.

For convenience, the algorithm derived in this subsection is given as a pseudo-code in Table 1.

The algorithm presented here will not only yield a sample from an approximation of the posterior, but it will also provide information about the bias, expressed through the final value of $\epsilon(t)$. This information, of course, can be used to *reduce the bias*, at the cost of sacrificing some effective sample size, via attaching the weights $\exp(-\delta u(\mathbf{z})/\epsilon)$, with δ being a small dimensionless parameter, to the final ensemble and re-sampling a new ensemble according to these weights. The choice of δ is arbitrary and expresses the trade-off between bias and effective sample size of the ensemble. The weights were chosen such that the re-sampled ensemble still represents a distribution of the form (13). Thus, such a bias correction step can also be applied, occasionally, during the algorithm, as long as the ensemble is given enough time to recover from the loss of effective sample size between two resampling steps.

Let us conclude this subsection with comments on the adaptive choice of the covariance K of the jump distribution k. To this end, we choose, for $k(\boldsymbol{\theta}, \boldsymbol{\theta}')$, a symmetric normal jump distribution, whose covariance is adapted to the empirical covariance of the marginal of $\mu(\mathbf{z}, t)$, $\Sigma(t)$, according to eq.

$$K = \beta\Sigma(t) + s\,\mathrm{tr}(\Sigma)\mathbb{1} \tag{34}$$

where s is a small constant preventing (34) from degenerating and β is an additional tuning parameter of the algorithm that mustn't be chosen much smaller than unity in order that the mixing in parameter space is fast enough compared to the decay of the mean distance to the target. Note that our derivation of the optimal cooling schedule was based on the assumption of a time-constant $k(\boldsymbol{\theta}, \boldsymbol{\theta}')$. The adaptation (34) makes $k(\boldsymbol{\theta}, \boldsymbol{\theta}')$ time-dependent, which leads to two compensatory effects. On the one hand, due to the increased acceptance probability ensued by this adaptation, the optimal schedule would be given by a time-dependent tuning parameter $v(t)$ that increases with time. This can be seen by repeating the exercise in [22], with an explicitly time-dependent $\dot{U} = \dot{U}(U, F, t)$, and acknowledging the fact that $\partial\dot{U}/\partial t < 0$ if the adaptation (34) leads to an increase of the acceptance rate, relative to a schedule without adaptation. On the other hand, typically, adaptation makes $k(\boldsymbol{\theta}, \boldsymbol{\theta}')$ sharper over time and, therefore, γ tends to increase over time. Thus, if we set $v/\gamma = \mathrm{const}$, v tends to increase over time. In general, the optimal schedule for $\epsilon^e(t)$, if adaptation (34) is employed, cannot be

Input:

1. Algorithms to sample from the prior and the likelihood.
2. Ensemble size N and initial value ϵ_{init}.
3. Covariance K of the jump distribution $k(\boldsymbol{\theta}, \boldsymbol{\theta}') = \mathcal{N}(\boldsymbol{\theta}, K)$.
4. Tuning parameter v. The default value is $v = 0.3$.

Initialization:

1. Repeat, until the ensemble E constructed in (d) contains N particles:
   (a) Sample a parameter vector, $\boldsymbol{\theta}$, from the prior.
   (b) Sample an output, $\mathbf{x}$, from the likelihood $f(\mathbf{x}|\boldsymbol{\theta})$.
   (c) Store the particle $(\boldsymbol{\theta}, \rho(\mathbf{x}, \mathbf{y}))$ in the ensemble P.
   (d) With probability $\exp[-\rho(\mathbf{x}, \mathbf{y})/\epsilon_{init}]$ store the particle $(\boldsymbol{\theta}, \rho(\mathbf{x}, \mathbf{y}))$ also in ensemble E.
2. Estimate the distribution function $G = G(\rho)$ defined in (14) by smoothing the empirical distribution of $\rho(\mathbf{x}, \mathbf{y})$ in the ensemble P, and re-calculate all the distances in ensemble E as $u = G(\rho(\mathbf{x}, \mathbf{y}))$.
3. Initialize U as the average of the redefined distance u in ensemble E.
4. Estimate γ defined in (21) using the prior ensemble P.
5. Initialize ϵ^e solving the quartic equation (31).
6. Initialize K according to (34).

Iteration:

1. Select a random particle, $(\boldsymbol{\theta}, u)$, from the ensemble E.
2. Sample a proposal parameter vector, $\boldsymbol{\theta}^*$, from $k(\boldsymbol{\theta}, \boldsymbol{\theta}^*)$.
3. Sample a proposal output, $\mathbf{x}^*$, from the likelihood $f(\mathbf{x}^*|\boldsymbol{\theta}^*)$ and calculate its redefined distance $u^* = G(\rho(\mathbf{x}^*, \mathbf{y}))$.
4. With probability $\min\left(1, \exp\left[-(u^* - u)/\epsilon^e\right]\right)$ update E, i.e., replace particle $(\boldsymbol{\theta}, u)$ by $(\boldsymbol{\theta}^*, u^*)$.
5. Whenever a significant fraction of the ensemble has been updated, update the ensemble average U, the transition temperature ϵ^e solving equation (31) and, optionally, the jump distribution according to eq. (34).
6. Stop the algorithm if the acceptance rate drops below a certain value.

Table 1: Algorithm I for the case of a non-informative prior.

determined easily. Therefore, the best strategy seems to be to turn on adaptation (34) and check whether the gain of efficiency due to an increased acceptance rate offsets the loss due to the deviation from the minimal entropy production path.

At this time, it would be premature to come up with too many recommendations of how to choose the tuning parameters v and β, as we do not yet have enough practical experience with the algorithm (but see the recommendation given in the application part of this paper). But we want to point out again that a too large v combined with a too small β might lead to a deviation from assumption (13) and, therefore, a bias that would be impossible to correct for.

2.3.3 The case with an informative prior

As we have discussed at the beginning of this section, the transition rate (5) has the disadvantage that a too fast decrease of ϵ can lead to convergence to a biased result with under-represented prior. To account for this bias, and ultimately control it, we replace (5) by a transition rate with a two-dimensional *control parameter* $\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2)$,

$$q_{\boldsymbol{\epsilon}}((\boldsymbol{\theta}', \mathbf{x}'), (\boldsymbol{\theta}, \mathbf{x})) = k(\boldsymbol{\theta}', \boldsymbol{\theta}) f(\mathbf{x}|\boldsymbol{\theta}) \min\left(1, \exp\left[-\frac{\rho(\mathbf{x}) - \rho(\mathbf{x}')}{\epsilon_1} - (1+\epsilon_2)(\nu(\boldsymbol{\theta}) - \nu(\boldsymbol{\theta}'))\right]\right), \tag{35}$$

where

$$\nu(\boldsymbol{\theta}) = -\ln\left(f(\boldsymbol{\theta})\right) \tag{36}$$

and $\rho(\mathbf{x}) = \rho(\mathbf{x}, \mathbf{y})$. Transition rate (35) satisfies the *detailed balance condition*

$$\pi_{\boldsymbol{\epsilon}}(\boldsymbol{\theta}', \mathbf{x}') q_{\boldsymbol{\epsilon}}((\boldsymbol{\theta}', \mathbf{x}'), (\boldsymbol{\theta}, \mathbf{x})) = \pi_{\boldsymbol{\epsilon}}(\boldsymbol{\theta}, \mathbf{x}) q_{\boldsymbol{\epsilon}}((\boldsymbol{\theta}, \mathbf{x}), (\boldsymbol{\theta}', \mathbf{x}'))\,, \tag{37}$$

for the equilibrium distribution

$$\pi_{\boldsymbol{\epsilon}}(\boldsymbol{\theta}, \mathbf{x}) = Z^{-1}(\boldsymbol{\epsilon}) f(\mathbf{x}|\boldsymbol{\theta}) e^{-\rho(\mathbf{x})/\epsilon_1 - (1+\epsilon_2)\nu(\boldsymbol{\theta})}\,, \tag{38}$$

with

$$Z(\boldsymbol{\epsilon}) = \int f(\mathbf{x}|\boldsymbol{\theta}) e^{-\rho(\mathbf{x})/\epsilon_1 - (1+\epsilon_2)\nu(\boldsymbol{\theta})} d\boldsymbol{\theta} d\mathbf{x}\,. \tag{39}$$

As before, we distinguish between the parameter $\boldsymbol{\epsilon}^e(t)$ that is used in the transition at time t, thus controlling the annealing schedule, and the parameter $\boldsymbol{\epsilon}(t)$ which describes the distribution of the process at time t under the endoreversibility assumption

$$\mu(\mathbf{z}, t) \approx \pi_{\boldsymbol{\epsilon}(t)}(\mathbf{z})\,. \tag{40}$$

Again our goal is to find a cooling schedule $\boldsymbol{\epsilon}^e(t)$ depending on $\boldsymbol{\epsilon}(t)$ such that the entropy production is minimized. In addition, we want the prior bias, measured by $\epsilon_2(t)$, to go to zero.

Initially, at time $t = 0$, the distribution is chosen as (38), with a rather large $\epsilon_1(0)$ and $\epsilon_2(0) = 0$. The corresponding ensemble is generated by adopting a rejection technique. The first control parameter $\epsilon_1^e(0)$ is set somewhat smaller than $\epsilon_1(0)$ and $\epsilon_2^e(0) = 0$.

Under the endoreversibility assumption, the distribution at any time is now characterized by the following two expectations ("extensive thermodynamic quantities")

$$U_1(t) := \int \rho(\mathbf{x}) \mu(\boldsymbol{\theta}, \mathbf{x}, t) d\boldsymbol{\theta} d\mathbf{x}\,, \tag{41}$$

$$U_2(t) := \int \nu(\boldsymbol{\theta}) \mu(\boldsymbol{\theta}, \mathbf{x}, t) d\boldsymbol{\theta} d\mathbf{x}\,. \tag{42}$$

By standard results about exponential families, there is a one-to-one correspondence between the vectors $\mathbf{U}$ and the parameters (intensive quantities) $\boldsymbol{\epsilon}$. This allows us to describe the system by the time-dependent vector $\boldsymbol{\epsilon}(t) = \boldsymbol{\epsilon}(\mathbf{U}(t))$. We are however not able to achieve approximate equality of these two vectors by a simple transformation.

As in the previous subsection, the entropy production rate can be expressed as

$$\dot{S}_{irr} = \mathbf{F}(t)^T \dot{\mathbf{U}}(t)\,,$$

where the driving forces are now

$$\mathbf{F}(t) = \begin{pmatrix} \epsilon_1(t)^{-1} - \epsilon_1^e(t)^{-1} \\ \epsilon_2(t) - \epsilon_2^e(t) \end{pmatrix}.$$

In order to find a necessary condition for minimal entropy production, we need as before to express $\mathbf{F}(t)$ as a function of $\mathbf{U}(t)$ and $\dot{\mathbf{U}}(t)$ and to compute in particular the matrix of partial derivatives $\frac{\partial \mathbf{F}}{\partial \dot{\mathbf{U}}}$.

In this two-dimensional setting, it seems however infeasible to establish a non-linear relationship between $\mathbf{F}$ and $\dot{\mathbf{U}}$ as we did in (20) for the one-dimensional setting. Therefore, we shall make the *linearity assumption*

$$\dot{\mathbf{U}} \approx L(\mathbf{U})\mathbf{F}\,, \tag{43}$$

which is reasonable as long as $\mathbf{F}(t)$ is not too large. Using the detailed balance condition (37), we find

$$\begin{aligned} L_{ij}(\mathbf{U}) = Z^{-1}(\boldsymbol{\epsilon}) \int & (u_i(\mathbf{z}) - u_i(\mathbf{z}'))(u_j(\mathbf{z}) - u_j(\mathbf{z}'))k(\boldsymbol{\theta}, \boldsymbol{\theta}') \\ & \times f(\mathbf{x}|\boldsymbol{\theta}) f(\mathbf{x}'|\boldsymbol{\theta}') \exp[-\rho(\mathbf{x})/\epsilon_1 - (1+\epsilon_2)\nu(\boldsymbol{\theta})] \\ & \times \chi\left((\rho(\mathbf{x}) - \rho(\mathbf{x}'))/\epsilon_1 + (1+\epsilon_2)(\nu(\boldsymbol{\theta}) - \nu(\boldsymbol{\theta}'))\right) d\mathbf{x} d\mathbf{x}' d\boldsymbol{\theta} d\boldsymbol{\theta}'\,, \end{aligned} \tag{44}$$

with $u_1(\mathbf{z}) = \rho(\mathbf{x})$ and $u_2(\mathbf{z}) = \nu(\boldsymbol{\theta})$. The $\mathbf{U}$ dependence of the r.h.s. of (44) is through $\boldsymbol{\epsilon} = \boldsymbol{\epsilon}(\mathbf{U})$. The matrix L is *symmetric* and *positive definite* (due to the Cauchy-Schwarz inequality). In the theory of non-equilibrium thermodynamics, the entries of the matrix L are known as the *Onsager coefficients* [16].

In two dimensions, eq. (30) becomes the necessary condition for minimal entropy production

$$\dot{\mathbf{U}}^T \frac{\partial \mathbf{F}}{\partial \dot{\mathbf{U}}} \dot{\mathbf{U}} = \text{const} = v\,. \tag{45}$$

Plugging (43) into (45) we find a necessary criterion for optimality to be given by

$$\dot{\mathbf{U}}^T R(\mathbf{U}) \dot{\mathbf{U}} = v\,, \tag{46}$$

where $R(\mathbf{U}) := L^{-1}(\mathbf{U})$ defines a metric on the (U_1, U_2)-plane. Equation (46) can also be derived as follows: Under the linearity assumption (43), and due to the Cauchy-Schwarz inequality, the entropy production satisfies the inequality

$$S_{irr} = \int_0^{t_f} \dot{\mathbf{U}}(t)^T R(\mathbf{U}(t)) \dot{\mathbf{U}}(t) dt \geq \frac{\mathcal{K}}{t_f}\,, \tag{47}$$

where $\mathcal{K}$ is the length of the process-path in the (U_1, U_2)-plane, measured with the metric $R(\mathbf{U})$. The lower bound of (47) is assumed if the integrand is constant, i.e., if the entropy

production rate is constant [19]. Thus, finding the optimal schedule consists in (i) finding the shortest path in the (U_1, U_2)-plane and (ii) traveling along this path such that the entropy production rate is constant. Therefore, condition (46) completely determines the optimal trajectory, which is of course a consequence of the linearity assumption (43).

In order to define our algorithm, we have to continuously estimate the following quantities during run-time: (i) the ensemble means $\mathbf{U}(t)$, (ii) the intensities $\boldsymbol{\epsilon}(t) = \boldsymbol{\epsilon}(\mathbf{U}(t))$ that determine our system under assumption (40) and (iii) the metric $L(\mathbf{U}(t))$. As (i) is trivial, we now discuss (ii) and (iii).

Given a small change, $\Delta\mathbf{U}$, of the ensemble means, the corresponding change of the intensities, $\Delta\boldsymbol{\epsilon}$, is estimated by means of

$$\Delta\boldsymbol{\epsilon} \approx \left(\frac{\partial \mathbf{U}}{\partial \boldsymbol{\epsilon}}\right)^{-1} \Delta\mathbf{U}\,, \tag{48}$$

where the Jacobi matrix

$$\frac{\partial \mathbf{U}}{\partial \boldsymbol{\epsilon}} := \begin{pmatrix} \frac{1}{\epsilon_1^2}\operatorname{Var}(\rho) & -\operatorname{Cov}(\rho,\nu) \\ \frac{1}{\epsilon_1^2}\operatorname{Cov}(\rho,\nu) & -\operatorname{Var}(\nu) \end{pmatrix} \tag{49}$$

is estimated using the empirical covariance matrix of the ρ and ν components of the ensemble. However, the neglected higher order corrections will eventually lead to large deviations from the "true" state. Therefore, occasional corrections have to take place estimating $\mathbf{U}(\boldsymbol{\epsilon})$ without using the ensemble E. Such an estimate can be calculated using the ensemble P drawn initially from the joint prior $f(\mathbf{x}, \boldsymbol{\theta})$. Once $\boldsymbol{\epsilon}$ is estimated, we need to estimate $L(\mathbf{U})$ in order to determine the adaptive tuning parameters $\boldsymbol{\epsilon}^e$. Inspecting equation (44) reveals that this can be done using the prior sample P as well as the ensemble E. This estimate relies on assumption (40). At the end of this subsection we will discuss a way of improving both estimates, $\mathbf{U}(\boldsymbol{\epsilon})$ and $L(\mathbf{U})$, for small ϵ_1, when the effective sample size of P is low.

Since, at the beginning of the algorithm, neither is the target value for U_2, at $\epsilon_1 = \epsilon_2 = 0$, known exactly nor is the metric $R(\mathbf{U}) = L^{-1}(\mathbf{U})$ known globally. Therefore, it appears difficult to come up with an optimal path in the (U_1, U_2)-plane. However, it appears reasonable to force the process to be on a path such that ϵ_2 remains small. Practically, this can be achieved by applying a counter force, setting

$$\epsilon_2^e = -a\epsilon_2\,, \tag{50}$$

where a is some positive constant. Finally, in order to find the optimal trajectory, under these restrictions, we need to choose ϵ_1^e such that (46) is satisfied. Using (43), we obtain the quadratic equation

$$\mathbf{F}^T L(\mathbf{U})\mathbf{F} = v\,. \tag{51}$$

The easiest version of the algorithm presented in this subsection is summarized in Table 2.

Note that the prior-bias in the final ensemble, as expressed through $\epsilon_2(t)$, can be completely corrected via a weighted re-sampling, in much the same way as the bias due to a non-vanishing $\epsilon_1(t)$ was reduced in the last subsection.

In the remainder of this subsection, we outline two alternative ways of estimating $\boldsymbol{\epsilon}(\mathbf{U})$ and $L(\mathbf{U})$, using the information gathered during the course of the algorithm. They can be used when ϵ_1 gets very small and the prior sample P yields poor estimates. Both methods, however, will depend on the assumption (40) being satisfied. One way is to simply correct the

ensemble E with weights proportional to $e^{-\rho(\mathbf{x})/\epsilon_1-\epsilon_2\nu(\boldsymbol{\theta})}$, in order to get a new prior sample, which has a better resolution where ϵ_1 is small. The other way is to populate, during the course of the algorithm, a *transition matrix*, Q, of *attempted* moves [1]. That is, we partition an area of interest in the (U_1, U_2)-plane (which will contain the small distances ρ) into $n_{U_1} n_{U_2}$ bins and increment the matrix element ${Q^{ij}}_{i'j'}$, whenever a particle in bin $U_{1,i'} \times U_{2,j'}$ attempts to move into bin $U_{1,i} \times U_{2,j}$. In order to get the correct transition matrix the diagonal entries ${Q^{i'j'}}_{i'j'}$ must be incremented whenever a particle from bin $U_{1,i'} \times U_{2,j'}$ attempts to jump outside the area of interest. Furthermore, the columns of Q must be normalized so that their sums equals unity. Under assumption (40) it holds that

$${Q^{ij}}_{i'j'} = \frac{\int_{\rho(\mathbf{x})\in U_{1,i}\,,\rho(\mathbf{x}')\in U_{1,i'}\,,\nu(\boldsymbol{\theta})\in U_{2,j}\,,\nu(\boldsymbol{\theta}')\in U_{2,j'}} k(\boldsymbol{\theta},\boldsymbol{\theta}')f(\mathbf{x}|\boldsymbol{\theta})f(\mathbf{x}'|\boldsymbol{\theta}')d\mathbf{x}d\mathbf{x}'d\boldsymbol{\theta}d\boldsymbol{\theta}'}{\int_{\rho(\mathbf{x}')\in U_{1,i'}\,,\nu(\boldsymbol{\theta}')\in U_{2,j'}} f(\mathbf{x}'|\boldsymbol{\theta}')d\mathbf{x}'d\boldsymbol{\theta}'}\,.$$

The eigenvector, $\mathbf{g}$, corresponding to the largest eigenvalue, 1, of Q, is a discretization of the likelihood function on the (U_1, U_2)-plane:

$$g_{i'j'} = \frac{\int_{\rho(\mathbf{x}')\in U_{1,i'}\,,\rho(\boldsymbol{\theta}')\in U_{2,j'}} f(\mathbf{x}'|\boldsymbol{\theta}')d\mathbf{x}'d\boldsymbol{\theta}'}{\int f(\mathbf{x}|\boldsymbol{\theta})d\mathbf{x}d\boldsymbol{\theta}}\,. \tag{52}$$

This holds true even if k is adapted during the algorithm. At a later stage of the algorithm, when the prior sample becomes insufficient but Q is sufficiently well populated to estimate (52), the latter can be used to estimate both $\mathbf{U}(\boldsymbol{\epsilon})$ and $L(\mathbf{U})$, for small values of ϵ_1. Furthermore, if k is not adapted, the matrix Q can be used directly to estimate $L(\mathbf{U})$, without the need of calculating the jump density matrix K.

Of course, a similar matrix of attempted moves can also be used in the one-dimensional setting, subsection 2.3.2, to replace prior sample P at a later stage of the algorithm.

3 Toy examples

In this section, we apply our adaptive scheme to two examples. The prior of the first one has almost no influence on the posterior, in the second this influence is large. As a shorthand for our adaptive scheme we use the acronym SABC, which merges SA, for Simulated Annealing, with ABC.

SABC is compared against the sequential Monte Carlo samplers (SMC) from del Moral et al. (2012) [5] and adaptive population Monte Carlo (APMC) from Lenormand et al. (2013) [11]. For the latter two the implementations in the R-package "EasyABC" [9] were used.

For SMC and APMC the same tuning parameters were used for both examples. The population size N for all algorithms was 1000. The parameter α of APMC was set to 0.5 following the recommendation of Lenormand et al. (2013). The tuning parameters for SMC are the same del Moral et al. (2012) used for the first toy example ($\alpha = 0.95$, $M = 1$, $N_T = 500$).

In real applications the computational costs are often dominated by sampling from the likelihood. Therefore, the number of samples drawn from the likelihood was used as measure of the computational effort.

Input:

1. Algorithms to sample from the prior and the likelihood.
2. Ensemble size N and initial value ϵ_{init}.
3. Tuning parameters β, s, v and a with default values, $\beta = 2$, $s = 0.01$, $v = 0.3$ and $a = 2$.

Initialization:

1. Repeat, until the ensemble E in (d) contains N particles:
   (a) Sample a parameter vector, $\boldsymbol{\theta}$, from the prior.
   (b) Sample an output, $\mathbf{x}$, from the likelihood $f(\mathbf{x}|\boldsymbol{\theta})$.
   (c) Store the vector $(\boldsymbol{\theta}, \rho(\mathbf{x}, \mathbf{y}), v(\boldsymbol{\theta}))$ in ensemble P.
   (d) With probability $\exp[-\rho(\mathbf{x}, \mathbf{y})/\epsilon_{init}]$ store the vector $(\boldsymbol{\theta}, \rho(\mathbf{x}, \mathbf{y}), \nu(\boldsymbol{\theta}))$ also in ensemble E.
2. Initialize metric $L(\mathbf{U})$ defined in (44), using the prior ensemble P.
3. Initialize $\mathbf{U}$ as the ensemble E averages.
4. Initialize $\epsilon_2^e = 0$ and ϵ_1^e solving the quadratic eq. (51).
5. Initialize K according to (34).

Iteration:

1. Select an arbitrary particle, $(\boldsymbol{\theta}, \rho, \nu)$, from the ensemble E.
2. Sample a proposal parameter vector, $\boldsymbol{\theta}^*$, from $k(\boldsymbol{\theta}, \boldsymbol{\theta}^*)$ and a proposal output, $\mathbf{x}^*$, from the likelihood $f(\mathbf{x}^*|\boldsymbol{\theta}^*)$.
3. With probability $r = \min\left(1, \exp\left[-(\rho^* - \rho)/\epsilon_1^e - (1 + \epsilon_2^e)(\nu^* - \nu)\right]\right)$, update E, i.e., replace $(\boldsymbol{\theta}, \rho, \nu)$ by $(\boldsymbol{\theta}^*, \rho^*, \nu^*)$.
4. Whenever a significant fraction of the ensemble has been updated, perform the following mean-field updates:
   - Save the old ensemble means $\mathbf{U}_{old}$ and denote the new ones by $\mathbf{U}_{new}$.
   - Update the Jacobi matrix (49) via calculation of the empirical covariance matrix of the ρ and ν components of E.
   - Save the old intensities $\boldsymbol{\epsilon}_{old}$ and calculate the new ones iterating the following two steps:
     (a) Compute the change $\Delta\boldsymbol{\epsilon} = \boldsymbol{\epsilon}_{new} - \boldsymbol{\epsilon}_{old}$ according to equation (48).
     (b) If $\mathbf{U}$ is close (say within a relative error of 1%) to the theoretical ensemble averages, $\mathbf{U}(\boldsymbol{\epsilon}_{new})$, as calculated from P, set $\boldsymbol{\epsilon} = \boldsymbol{\epsilon}_{new}$ and stop, otherwise, replace $\boldsymbol{\epsilon}_{new} \to \boldsymbol{\epsilon}_{old}$ and $\mathbf{U}(\boldsymbol{\epsilon}_{new}) \to \mathbf{U}_{old}$ and go back to (a).
   - Update the metric $L(\mathbf{U})$, according to $\boldsymbol{\epsilon}$, using prior ensemble P.
   - Update $\epsilon_2^e = -a\epsilon_2$ and ϵ_1^e solving (51).
   - Optionally: update K according to (34).
5. Stop the algorithm if the acceptance rate drops below a certain value.

Table 2: Algorithm II, for the case of an informative prior.

3.1 Example 1

The first example is a traditional example of the ABC literature (e.g. [5], [11]). The prior is uniformly distributed on the interval $[-10, 10]$, and the likelihood is given by the sum of two normal distributions with very different standard deviations:

$$f(x|\theta) \propto \exp\left[-\frac{(x-\theta)^2}{2}\right] + \frac{1}{\sigma}\exp\left[-\frac{(x-\theta)^2}{2\sigma^2}\right], \tag{53}$$

with $\sigma = 0.1$. Thus, the posterior for $y = 0$ is given by

$$f(\theta|y) \propto \mathbb{1}_{[-10,10]}\left(\exp\left[-\frac{\theta^2}{2}\right] + \frac{1}{\sigma}\exp\left[-\frac{\theta^2}{2\sigma^2}\right]\right). \tag{54}$$

As the prior has almost no influence on the posterior, the non-linear algorithm from Table 1, with one final bias correction, has been employed. Furthermore, $k(\boldsymbol{\theta}, \boldsymbol{\theta}')$ has been continuously adapted according to (34). Since the prior has a much bigger variance than the posterior, this turns out to be beneficial, for the convergence of the algorithm. The optimal choice for the dimensionless parameter β, defined in (34), is expected to depend little on details of the model. For our examples we choose $\beta = 2$, which is large enough to ensure a fast enough mixing in parameter space compared to the decay of the mean distance to the target. The tuning parameter v/γ governs the annealing speed. As this example is so simple, its choice is not very critical. With the choice $\beta = 2$ a violation of the endoreversibility assumption is not to be expected, even for high annealing speeds. A slowing down of the convergence due to a trapping of particles is observed only at very high values of v/γ. We choose the value $v/\gamma = 3$.

Figure 1 shows the results for all three samplers after, approximately, 10 000 and 40 000 simulations from the likelihood. It is clearly visible that SMC has not yet converged, while the results of APMC and SABC look much better. After 40 000 likelihood samples, the histogram of SABC looks slightly smoother than the one of APMC. As APMC is an importance sampling algorithm, the sample generated after 40 000 simulations is an exact sample from a closer approximation of the posterior than the sample generated after 10 000 simulations. Therefore, we attribute the slight deterioration of the histogram to the loss of effective sample size (ESS) due to resampling. The ESS, for APMC and SABC, are summarized in Table 3. For APMC, the ESS was calculated under the optimistic assumption that, before the last resampling is made, the ensemble has completely recovered from the loss of ESS. For SABC, the loss of ESS after 10000 simulations is due to the final bias correction step. The parameter δ, used for the final bias correction as described towards the end of subsection 2.3.2, was chosen such that the ESS of SABC and APMC are comparable.

	APMC	SABC
10 000 simulations	306	240
40 000 simulations	323	1000

Table 3: Comparison of effective sample sizes of the APMC and the SABC algorithm for example 1, after 10 000 and 40 000 likelihood simulations.

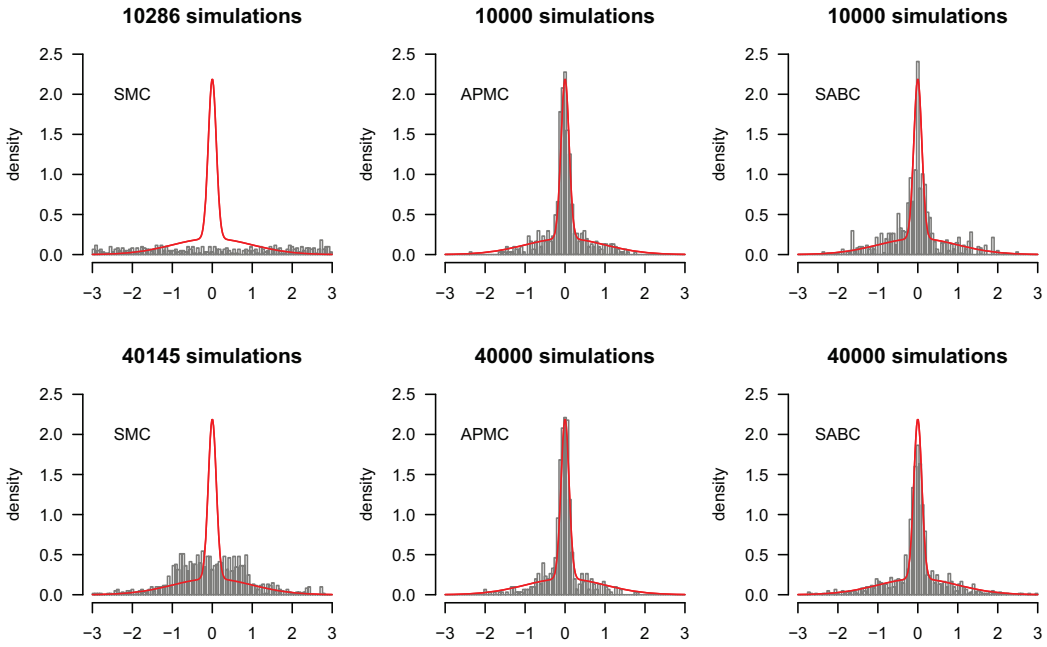

Figure 1: Histograms of an ensemble of 1000 particles for example 1 generated with SMC, APMC and SABC. The solid curve is the exact posterior density. Note that "simulations" refers to single draws from the likelihood.

3.2 Example 2

In contrast to the first example, the prior in the second example has a large influence on the posterior. The prior shall be given as the normal distribution

$$f(\theta) = \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{\theta^2}{2}\right] ,$$

and the likelihood as the normal distribution

$$f(x|\theta) = \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{(x-\theta)^2}{2}\right] .$$

Thus, the posterior is given as

$$f(\theta|y) = \frac{1}{\sqrt{\pi}} \exp\left[-(\theta - y/2)^2\right] .$$

To investigate if the algorithms can handle severe *prior-data conflicts*, we set $y = 3$.

In this example it is important for SABC to properly control $\epsilon_2(t)$ while annealing $\epsilon_1(t)$ as prior and likelihood "pull from opposite directions". Therefore, we employ the linear algorithm as described in Table 2, with one final bias correction. The tuning parameter v, which now has the interpretation of an entropy production rate, was chosen to be 0.3. For β we chose the same value as in the previous example, namely $\beta = 2$. In this example, continuously adapting $k(\boldsymbol{\theta}, \boldsymbol{\theta}')$ has a negligible effect on the convergence speed.

The results are shown in figure 2. Again, the results from SMC have not yet converged and are heavily biased towards the prior. APMC seems to converge slightly faster then SABC (compared at 10 000 simulations). However, the quality of the APMC sample decreases for more simulations, which is attributed to the loss of ESS. As SABC is avoiding resampling, this effect is not observed. Effective sample sizes, for APMC and SABC are summarized in Table 4. After 10000 simulations, we chose the bias-correcting parameter δ such that the ESS of SABC and APMC are similar. After 40000 simulations, the loss of ESS for SABC is due solely to the correction of the prior bias, expressed through the final value of ϵ_2.

	APMC	SABC
10 000 simulations	404	408
40 000 simulations	322	982

Table 4: Comparison of effective sample sizes of the APMC and the SABC algorithm for example 2, after 10 000 and 40 000 likelihood simulations.

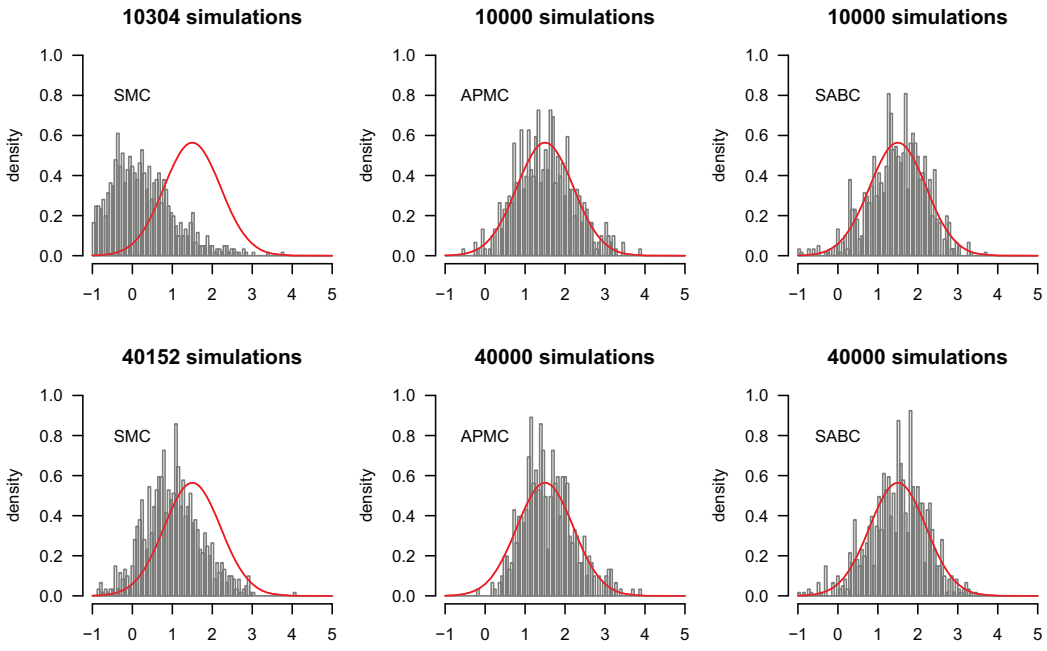

Figure 2: Histograms of an ensemble of 1000 particles for example 2 generated with SMC, APMC and SABC. The solid curve is the exact posterior density.

4 Real-world example: tuberculosis bacteria

Tanaka et al. [23] analyzed genotype data of tuberculosis bacteria with a stochastic model to infer death, birth and mutation rates by means of ABC. In the 473 analyzed tuberculosis bacteria cultures, 326 distinct genotypes where found. Cultures with the same genotype form a cluster. The data in table 5 describe how many clusters with a certain number of cultures

were found. For example one cluster consisting of 30 cultures with the same genotype was observed, two clusters with five cultures each, and so forth.

This data contains only information on the rates relative to each other, because no time information is available. Therefore, only birth, death, and mutation events are simulated until the population reaches a (arbitrarily defined) size of 10 000 living bacteria (see [23] for details). Therefrom a random sample without replacement of size 473 is taken. We used the parametrization proposed by Fearnhead and Prangle [7] which reduces the inference to a two dimensional problem with $a = P(\text{birth}|\text{event})$ and $d = P(\text{death}|\text{event})$. The probability that an event is a mutation is given by $1 - a - d$. Also the same flat prior is used $\pi(a, d) \propto \mathbf{1}_{a>d}\mathbf{1}_{0<a+d\leq 1}$.

The data are summarized by two statistics as described by [23]: the number of distinct genotypes g in the sample and a measure of gene diversity $H = 1 - \sum_{i=1}^{g} (n_i/473)^2$, where n_i is the number of bacteria in the ith cluster. The distance between simulated and observed data is measured as $|g^* - g|/473 + |H^* - H|$, where the asterisks indicate the statistics of the simulated data.

Because of the flat prior the non-linear version of the SABC, Table 1, was used, with a final re-sampling step, with $\delta = 0.2$. As in the previous examples, we chose $\beta = 2$ and tuned v/γ. A high convergence speed is achieved with $v/\gamma = 7$ but we found that the algorithm is remarkably robust w.r.t. the choice of this tuning parameter.

We compared the performance of our algorithm with the adaptive population Monte Carlo (APMC) from Lenormand et al. [11], which we ran with the same sample size $N = 200$ and with the choice of the tuning parameter $\alpha = 0.5$, as recommended in [11]. Similarly to the results from the previous section, we found that, for short simulation times, APMC shows a slightly better performance than SABC due to a faster convergence, but, for longer simulation times, APMC suffers from a deterioration of the sample due to a loss of ESS, which is not observed with SABC. Figure 3 shows the results after 3800 iterations (approximately 2000 simulations from the likelihood). The result of SABC is in excellent agreement with the result reported in [7], whereas the final sample from APMC shows some signs of deterioration, which is attributed to the loss of ESS, in each iteration step. The time-course of the ESS, for APMC, is shown in Figure 3. The jump to an ESS of about 80, before the last resampling step, is based on the (presumably unrealistic) assumption that each population update leads to a complete recovery of the ESS. For SABC, the final ESS is 129 and due to the final resampling step.

5 Conclusions

We have presented a framework of particle algorithms for Approximate Bayes Computations that is inspired by Simulated Annealing. Its main advantage compared to the sequential ABC algorithms the authors are aware of is the fact that it is not based on importance sampling. Therefore, the effective sample size of our algorithms does not decrease over time. As the interactions between the particles in the adaptive algorithm are of *mean-field type*, the

Table 5: Tuberculosis genotype data.

number of cultures per cluster	30	23	15	10	8	5	4	3	2	1
number of clusters	1	1	1	1	1	2	4	13	20	282

statistical independence of the particles is preserved (see, e.g., [4]).

The cost for this gain of efficiency is the fact that our system is necessarily out of equilibrium. That is, in addition to the bias due to non-zero equilibrium tolerances ϵ_1^e and ϵ_2^e, we have a bias due to our system being out of equilibrium (i.e. ϵ_1 being larger than ϵ_1^e). There is a trade-off between these two kinds of bias reflected in the choice of the tuning parameter v. Choosing a larger v might result in a smaller ϵ_1^e, for a given computation time, but in a larger bias of the second kind. Choosing v too large, in combination with too slow mixing in parameter space, expressed through a too small β, might lead to a third kind of bias, a violation of the endoreversibility assumption (13) or (40). This kind of bias is impossible to correct for and has to be avoided by a careful choice of tuning parameters.

In Sect. 2.2 we proved convergence to the correct posterior, for cooling that is slower than a certain inverse power of time. In Sect. 2.3 we presented an adaptive cooling scheme that is designed to achieve convergence to the correct posterior with a minimum of computational effort. Therefore, the control variable ϵ_1^e is adjusted according to the particles' distance to the target in such a way that the entropy production in the system, which is a measure for the waste of computation, is minimized. If the prior is important, a second control variable is used to control its influence. Using this adaptive scheme, tuning essentially reduces to the choice of β, related to the mixing speed in parameter space, and v, related to the annealing speed.

In our scheme the characteristic function $\chi(\epsilon - \rho(\mathbf{x}, \mathbf{y}))$, which is often used in ABC calculations, is replaced by the Boltzmann factor $\exp(-\rho(\mathbf{x}, \mathbf{y})/\epsilon)$. 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 $\mathcal{O}(N)$, with some overhead due to occasional mean-field updates needed for the update of the tolerance(s) and the jump distribution. Importance sampling algorithms are typically of the order $\mathcal{O}(N^2)$, due to the weighting step, but see the algorithm by del Moral et al [5], which scales like $\mathcal{O}(N)$. However, all the algorithms mentioned in this article scale like $\mathcal{O}(N)$ with the number of simulations from the likelihood, which is usually the most costly step. Like all sequential ABC algorithms, our scheme is well suited for *parallelization*.

The overhead, in our scheme, is significantly larger if the prior is informative. Furthermore, in this case, we can only derive an optimal schedule for relatively slow annealing (linearity assumption). For strongly informative priors, a simple ABC rejection algorithm should be considered as an alternative to a sequential schedule.

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 (see, e.g., [7] and [12]).

Drawing the initial sample for our adaptive algorithm generates, as a side product, a larger sample from the joint prior. In our adaptive scheme we use this prior information, for the redefinition of the metric (14) or to estimate the sample average $\mathbf{U}(\boldsymbol{\epsilon})$ and the metric $L(\mathbf{U})$. Note that, at the same time, this information can be used to establish appropriate summary statistics, as described in [7].

Acknowledgements

The first author is indebted to Bjarne Andresen for valuable comments on the adaptive algorithm.

References

[1] B. Andresen, KH. Hoffmann, K. Mosegaard, J. Nulton, JM. Pedersen, and P. Salamon. On lumped models for thermodynamic properties of simulated annealing problems. *J. Physique.*, 49(9):1485–1492, 1988.

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

[3] A. Beskos, D. Crisan, and A Jasra. On the Stability of Sequential Monte Carlo Methods in High Dimensions. *arXiv: 1103.3965v2*, 2012.

[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.

[5] P. Del Moral, A. Doucet, and A. Jasra. An adaptive sequential Monte Carlo method for approximate Bayesian computation. *Statistics and Computing*, 22(5):1009–1020, 2012.

[6] R. Douc, E. Moulines, and J.S. Rosenthal. Quantitative bounds on convergence of time-inhomogeneous Markov chains. *The Annals of Applied Probability*, 14(4):1643–1665, 2004.

[7] P. Fearnhead and D. Prangle. Constructing summary statistics for approximate Bayesian computation: semi-automatic approximate Bayesian computation. *J. Roy. Stat. Soc. B*, 74(3):419–474, 2012.

[8] 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.

[9] F. Jabot, T. Faure, and N. Dumoullin. *EasyABC: EasyABC: performing efficient approximate Bayesian computation sampling schemes*, 2013. R package version 1.2.2.

[10] A. Lee. On the choice of MCMC kernels for approximate Bayesian computation with SMC samplers. In *Proceedings of the 2012 Winter Simulation Conference (WSC 2012)*, page 12 pp. IEEE Syst., Man, Cybernetics Soc., 2012 2012. 2012 Winter Simulation Conference (WSC 2012), 9-12 Dec. 2012, Berlin, Germany.

[11] M. Lenormand, F. Jabot, and Deffuant G. Adaptive approximate Bayesian computation for complex models. *Comp. Stat.*, 28(6):2777–2796, 2013.

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

[13] J.M. Marin, P. Pudlo, C.P. Robert, and R.J. Ryder. Approximate Bayesian computational methods. *Statistics and Computing*, 22(6, SI):1167–1180, 2012.

[14] 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.

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

[16] L. Onsager. Reciprocal relations in irreversible processes. I. *Phys. Rev.*, 37(4):405–426, 1931.

[17] M.H. Rubin. Optimal Configuration of a Class of Irreversible Heat Engines I. *Phys. Rev. A*, 19(3):1272–1276, 1979.

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

[19] P. Salamon, A. Nitzan, B. Andresen, and RS. Berry. Minimum Entropy Production and the Optimization of Heat Engines. *Phys. Rev. A*, 21(6):2115–2129, 1980.

[20] M. Sedki, P. Pudlo, Marin J.M., C.P. Robert, and J.M. Cornuet. Efficient learning in ABC algorithms. *arXiv: 1210.1388v2 [stat.CO]*, 2013.

[21] U. Seifert. Entropy production along a stochastic trajectory and an integral fluctuation theorem. *Phys. Rev. Lett.*, 95, 2005.

[22] W. Spirkl and H. Ries. Optimal Finite-Time Endoreversible Processes. *Phys. Rev. E*, 52(4, A):3485–3489, 1995.

[23] M.M. Tanaka, A.R. Francis, F. Luciani, and S.A. Sisson. Using approximate bayesian computation to estimate tuberculosis transmission parameters from genotype data. *Genetics*, 173(3):1511–1520, 2006.

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

[25] 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.

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

[27] R.D. Wilkinson. Approximate Bayesian computation (ABC) gives exact results under the assumption of model error. *Stat. App. in Gen. and Mol. Biol.*, 12(2):129–141, 2013.

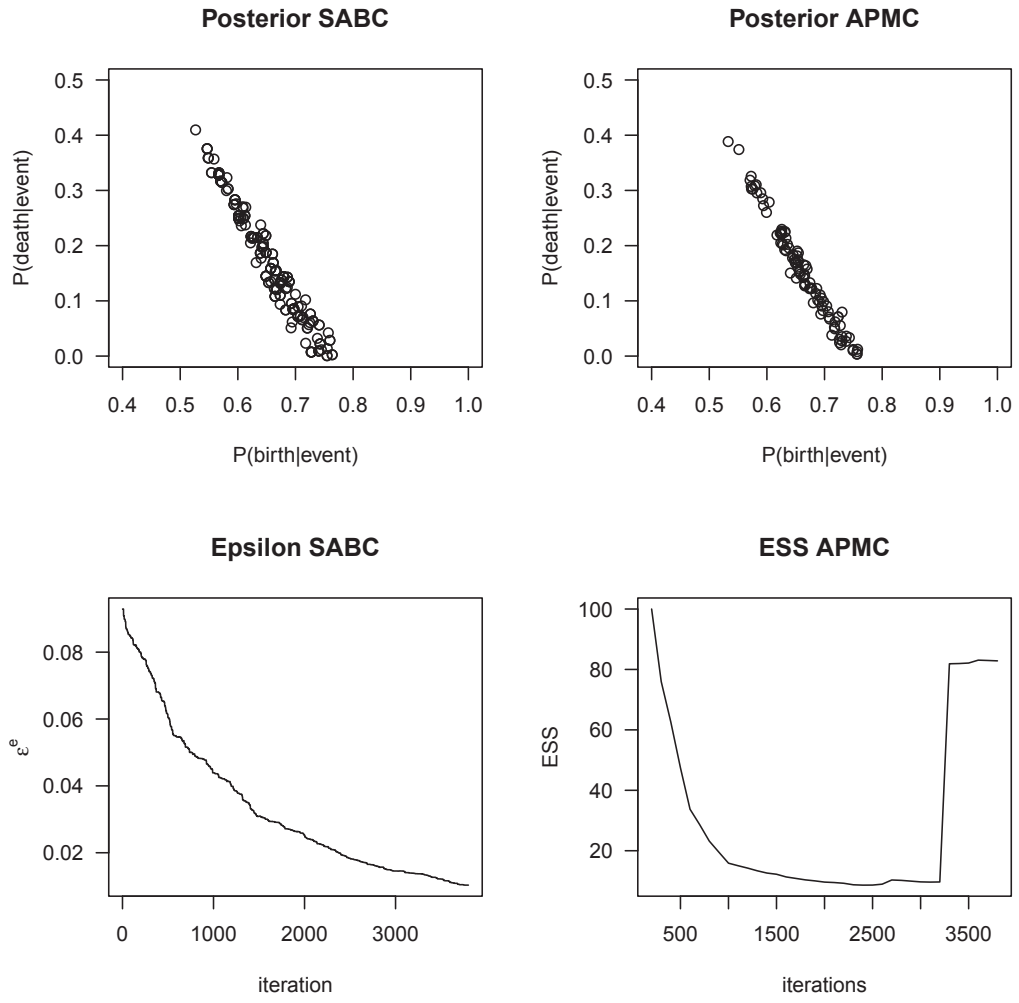

Figure 3: Top row: Final population of 200 particles after a total of 3800 updates (approximately 2000 simulations from the likelihood, the rest were jumps into forbidden parameter regions), for SABC (left) and APMC (right). Bottom row: Time-course of $\epsilon^e(t)$, for SABC (left) and time-course of the ESS, for APMC (right). The final ESS, for SABC, is 129 and due to a single resampling step.