Improving Approximate Bayesian Computation via Quasi-Monte Carlo

Alexander Buchholz, Nicolas Chopin

Improving approximate Bayesian computation via quasi-Monte Carlo

Alexander Buchholz Nicolas Chopin
ENSAE-CREST

Abstract

ABC (approximate Bayesian computation) is a general approach for dealing with models with an intractable likelihood. In this work, we derive ABC algorithms based on QMC (quasi-Monte Carlo) sequences. We show that the resulting ABC estimates have a lower variance than their Monte Carlo counter-parts. We also develop QMC variants of sequential ABC algorithms, which progressively adapt the proposal distribution and the acceptance threshold. We illustrate our QMC approach through several examples taken from the ABC literature.

The computer code used to perform our numerical experiments is available at `https://github.com/alexanderbuchholz/ABC`.

Keywords: Approximate Bayesian computation, Likelihood-free inference, Quasi-Monte Carlo, Randomized Quasi-Monte Carlo, Adaptive importance sampling

1 Introduction

Since its introduction by Tavaré et al. (1997) approximate Bayesian computation (ABC) has received growing attention and has become today a major tool for Bayesian inference in settings where the likelihood of a statistical model is intractable but simulations from the model for a given parameter value can be generated. The approach of ABC is as convincing as intuitive: We first sample a value from the prior distribution, conditional on this prior simulation an observation from the model is generated. If the simulated observation is sufficiently close to the observation that has been observed in nature, we retain the simulation from the prior distribution and assign it to the set of posterior simulations. Otherwise the simulation is discarded. We repeat this procedure until enough samples have been obtained.

Since then several computational extensions related to ABC have been proposed. For instance the use of MCMC as by Marjoram et al. (2003) has improved the simulation of ABC posterior samples over the simple accept–reject algorithm. The use of sequential approaches by Beaumont et al. (2009), Sisson et al. (2009), Del Moral et al. (2012) and Sedki et al. (2012) made it possible to exploit the information from previous iterations and eventually to choose adaptively the schedule of thresholds ϵ. Besides the question of an efficient simulation of high posterior probability regions, the choice of summary statistics, summarizing the information contained in the observation and the simulated observation, has been investigated (Fearnhead and Prangle, 2012). See Marin et al. (2012) and Lintusaari et al. (2017) for two recent reviews. Moreover, the introduction of more machine learning driven approaches like random forests (Marin et al., 2016), Gaussian processes (Wilkinson, 2014), Bayesian optimization (Gutmann and Corander, 2016), expectation propagation

(Barthelmé and Chopin, 2014) and neural networks (Papamakarios and Murray, 2016) have been proposed. A post-processing approach based on nonparametric regression was studied in Blum (2010).

In this paper we take a different perspective and approach the problem of reducing the variance of ABC estimators. We achieve this by introducing so called low discrepancy sequences in the simulation of the proposal distribution. We show that this allows to reduce significantly the variance of posterior estimates.

The rest of the paper is organized as follows. Section 2 reviews the basic ideas of approximate Bayesian computation and sets the notation. Section 3 introduces the concept of low discrepancy sequences. Section 4 brings the introduced concepts together and provides the theory that underpins the proposed idea. Section 5 presents a first set of numerical examples. Section 6 explains how to use our ideas in a sequential procedure which adapts progressively the proposal distribution and the value of ϵ. Section 7 illustrates the resulting sequential ABC procedure. Section 8 concludes.

2 Approximate Bayesian computation

2.1 Reject-ABC

Approximate Bayesian computation is motivated by models such that (a) the likelihood function is difficult or expensive to compute; (b) simulating from the model (for a given parameter θ) is feasible.

The most basic ABC algorithm is called reject-ABC. It consists in simulating pairs (θ, y), from the prior $p(\theta)$ and the likelihood $p(y|\theta)$, and keeping those pairs such that $\delta(y, y^\star) \leq \epsilon$, where $y^\star$ is the actual data, and $\delta : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$ is some distance (e.g. Euclidean). This is done until N pairs are accepted. The target density of this rejection algorithm is:

$$p_\epsilon(\theta, y) = \frac{1}{Z_\epsilon} p(\theta) p(y|\theta) \mathbb{1}\left\{\delta(y, y^\star) \leq \epsilon\right\},$$

and its marginal density with respect to θ is:

$$p_\epsilon(\theta) = \frac{1}{Z_\epsilon} p(\theta) \mathbb{P}_\theta\left(\delta(y, y^\star) \leq \epsilon\right) \tag{1}$$

where $\mathbb{P}_\theta$ denotes a probability with respect to $y \sim p(y|\theta)$, and $Z_\epsilon = \int_\Theta p(\theta)\mathbb{P}_\theta\left(\delta(y, y^\star) \leq \epsilon\right) \mathrm{d}\theta$ is the normalising constant.

As $\epsilon \to 0$, (1) converges to the true posterior density. Actually, δ is often not a distance but a pseudo-distance of the form: $\delta(y, y^\star) = \|s(y) - s(y^\star)\|_2$, where $\|\cdot\|_2$ is the Euclidean norm, and $s(y)$ is a low-dimensional, imperfect summary of y. In that case, $p_\epsilon(\theta) \to p(\theta|s(y^\star))$. This introduces an extra level of approximation, which is hard to assess theoretically and practically. However, in this paper we focus on how to approximate well (1) for a given δ (and ϵ), and we refer to e.g. Fearnhead and Prangle (2012) for more discussion on the choice of δ or s.

2.2 Pseudo-marginal importance sampling

A simple generalisation of reject-ABC is described in Algorithm 1. For $n = 1, \ldots, N$, we sample the parameter $\theta_n \sim q(\theta)$, the latent variable $x_n \sim q_{\theta_n}(x)$, and reweight (θ_n, x_n)

according to

$$w_n = \frac{p(\theta_n)}{q(\theta_n)} \times \hat{L}_\epsilon(x_n)$$

where, for $x \sim q_\theta$, $\hat{L}_\epsilon(x)$ is an unbiased estimate of the probability $\mathbb{P}_\theta\left(\delta(y, y^\star) \leq \epsilon\right)$:

$$\int q_\theta(x)\hat{L}_\epsilon(x)\mathrm{d}x = \mathbb{P}_\theta\left(\delta(y, y^\star) \leq \epsilon\right).$$

Input: Observed $y^\star$, prior distribution $p(\theta)$, proposal distribution $q(\theta)$, simulator $q_{\theta_n}(x)$, distance function $\delta(\cdot,\cdot)$, target threshold ϵ, number of simulations N
Result: Set of weighted samples $(\theta_n, x_n, w_n)_{n\in 1:N}$
for $n = 1$ *to* N **do**
  Sample $\theta_n \sim q(\theta)$
  Sample $x_n \sim q_{\theta_n}(x)$
  Set $w_n = p(\theta_n)\hat{L}_\epsilon(x_n)/q(\theta_n)$
end

Algorithm 1: ABC importance sampling algorithm

The marginal density (with respect to θ) of the target density of this importance sampling scheme is again (1). In particular, the quantity

$$\hat{\phi}_N = \frac{\sum_{n=1}^N w_n \phi(\theta_n)}{\sum_{n=1}^N w_n}, \tag{2}$$

is a consistent (as $N \to \infty$ and under appropriate conditions) estimate of expectation $\mathbb{E}_{p_\epsilon(\theta)}[\phi(\theta)]$, for $\phi : \Theta \to \mathbb{R}$. Since the importance weight involves an unbiased estimator, the whole procedure may be viewed as a pseudo-marginal sampler, in the spirit of Andrieu and Roberts (2009).

A a special case, take the proposal $q(\theta)$ to be equal to the prior, $p(\theta)$, and take $x = y$, $\hat{L}_\epsilon(x) = \mathbb{1}\left\{\delta(y, y^\star) \leq \epsilon\right\}$; then we recover essentially the same procedure as reject-ABC (except that N stands for the number of proposed points, rather than the number of accepted points). However, the generalized scheme allows us (a) to sample θ_n from a distribution $q(\theta)$ which may be more likely (than the prior) to generate high values for the probability $\mathbb{P}_\theta\left(\delta(y, y^\star) \leq \epsilon\right)$; and (b) to use a more sophisticated unbiased estimate for $\mathbb{P}_\theta\left(\delta(y, y^\star) \leq \epsilon\right)$.

Regarding (b), we consider two unbiased schemes in this work. In the first part, we focus on:

$$x = y_{1:M}, \quad q_\theta(x) = \prod_{m=1}^M p(y_m|\theta), \quad \hat{L}_\epsilon(x) = \frac{1}{M}\sum_{m=1}^M \mathbb{1}\{\delta(y_m, y^\star) \leq \epsilon\}. \tag{3}$$

for a certain $M \geq 1$. The possibility to associate more than one datapoints to each parameter θ_n was considered in e.g. Del Moral et al. (2012). Bornn et al. (2015) showed that $M = 1$ usually represents the best variance vs CPU time trade-off when using Monte Carlo sampling, however we shall see that this result does not hold when using QMC.

Later on in the paper, we shall consider an alternative unbiased estimator, based on properties of the negative binomial distribution. More precisely, assume that, for a given θ, we sample sequentially $y_1, y_2, \ldots \sim p(y|\theta)$, until we reach the time k where $r \geq 2$ datapoints

are such that $\delta(y_n, y^\star) \leq \epsilon$; then k is distributed according to a negative binomial distribution with parameters r and $p = \mathbb{P}_\theta\left(\delta(y, y^\star) \leq \epsilon\right)$, and the minimum-variance unbiased estimator of $\mathbb{P}_\theta\left(\delta(y, y^\star) \leq \epsilon\right)$ is (Johnson et al., 2005, Chap. 8):

$$\hat{L}_\epsilon(x) = \frac{r-1}{k-1}$$

where $x = y_{1:k}$.

The second unbiased estimator is closely related, but not equivalent to, the r-hit kernel of Lee (2012); see also Lee and Łatuszyński (2014). Specifically, Lee (2012) proposed an MCMC kernel that generates *two* negative binomial variates (one for the current point, and one for the proposed point) at each iteration. The invariant distribution of this kernel is such that, marginally, θ is distributed according to (1).

In more practical terms, we shall use the latter estimator in situations where we would like to set ϵ beforehand to some value such that $\mathbb{P}_\theta\left(\delta(y, y^\star) \leq \epsilon\right)$ may be small. In that case, this estimator automatically adjusts the CPU budget (i.e. the number of simulations from the likelihood) so as to ensure that the number of simulated $y-$values is non-zero. But we shall return to this point in Section 6.

3 Quasi-Monte Carlo

3.1 QMC Overview

This section gives a brief overview of QMC and the underlying theory; for a more in-depth presentation, see e.g. the book of Lemieux (2009), the book of Leobacher and Pillichshammer (2014) or Chapter 5 in Glasserman (2013).

QMC sequences (also called low discrepancy sequences), are used to approximate integrals over the $[0,1]^d$ hypercube:

$$\mathbb{E}\left[\psi(U)\right] = \int_{[0,1]^d} \psi(u)\mathrm{d}u,$$

that is the expectation of the random variable $\psi(U)$, where $U \sim \mathcal{U}\left([0,1]^d\right)$. The basic Monte Carlo approximation of the integral is $\hat{I}_N := N^{-1}\sum_{n=1}^N \psi(\mathbf{u}_n)$, where each $\mathbf{u}_n \sim \mathcal{U}\left([0,1]^d\right)$. The error of this approximation is $\mathcal{O}_P(N^{-1/2})$, since $\mathrm{Var}[\hat{I}_N] = \mathrm{Var}[\psi(U)]/N$.

It is possible to improve on this basic approximation, by replacing the random variables $\mathbf{u}_n$ by a low-discrepency sequence; that is, informally, a deterministic sequence that covers $[0,1]^d$ more regularly. This idea is illustrated in Figure 1.

More formally, the general notion of discrepancy of a given sequence is defined as follows:

$$D(\mathbf{u}_{1:N}, \mathcal{A}) := \sup_{A\in\mathcal{A}} \left| \frac{1}{N}\sum_{n=1}^N \mathbb{1}\left\{\mathbf{u}_n \in A\right\} - \lambda_d(A) \right|,$$

where $\lambda_d(A)$ is the volume (Lesbegue measure on $\mathbb{R}^d$) of A and $\mathcal{A}$ is a set of measurable sets. When we fix the sets A to be intervals anchored at 0 we obtain the so called star discrepancy:

$$D^*(\mathbf{u}_{1:N}) := \sup_{[0,\mathbf{b}]} \left| \frac{1}{N}\sum_{n=1}^N \mathbb{1}\left\{\mathbf{u}_n \in [0,\mathbf{b}]\right\} - \prod_{i=1}^d b_i \right|,$$

where $[0, \mathbf{b}] = \prod_{i=1}^{d}[0, b_i], 0 \leq b_i \leq 1$. The importance of the notion of discrepancy and in particular the star discrepancy is highlighted by the Koksma-Hlawka inequality (Hickernell, 2006), which relates the error of the integration to the coverage of the space and the variation of the function that is integrated:

$$\left| \int_{[0,1]^d} \psi(\mathbf{u}) \mathrm{d}\mathbf{u} - \frac{1}{N} \sum_{n=1}^{N} \psi(\mathbf{u}_n) \right| \leq V(\psi) D^*(\mathbf{u}_{1:N}),$$

where $V(\psi)$ is the variation in the sense of Hardy and Krause (Hardy, 1905). The actual definition of this quantity is a bit involved, but essentially it measures in some way the smoothness of the function ψ; see Kuipers and Niederreiter (2012) and Leobacher and Pillichshammer (2014) for more details.

It is possible to construct sequences $\mathbf{u}_n$ such that, when N is fixed in advance, $D^*(\mathbf{u}_{1:N})$ is $\mathcal{O}\left(N^{-1}(\log N)^{d-1}\right)$, and, when N is allowed to grow, i.e., the sequence must be generated iteratively, then $D^*(\mathbf{u}_{1:N}) = \mathcal{O}\left(N^{-1}(\log N)^{d}\right)$. Then $\forall \tau > 0$ the error rate is $\mathcal{O}\left(N^{-1+\tau}\right)$. Consequently, QMC integration schemes are asymptotically more efficient than MC schemes. One observes in practice that QMC integration outperforms MC integration even for small N in most applications, see e.g. the examples in Chapter 5 of Glasserman (2013).

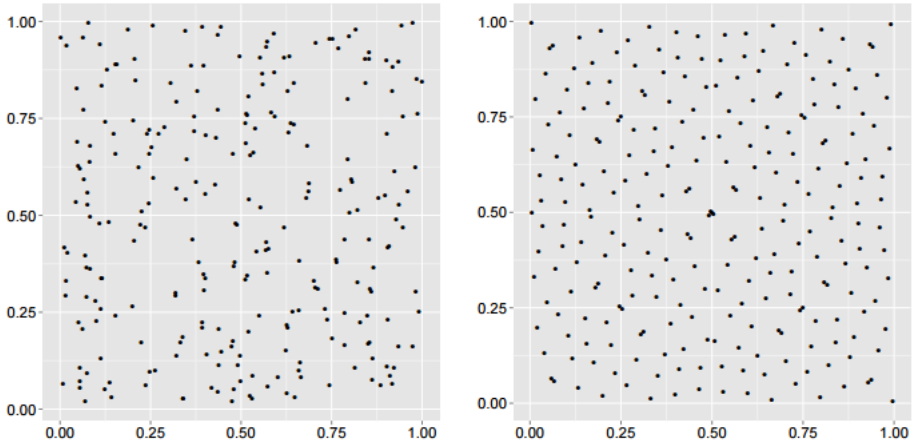


Figure 1: Uniform random (left) and QMC (right) point sets of length 256 in $[0, 1]^2$. The QMC sequence covers the target space more evenly than the random uniform sequence.

3.2 Randomized quasi-Monte Carlo

A drawback of QMC is that it does not come with an easy way to assess the approximation error. RQMC (randomized quasi-Monte Carlo) amounts to introduce randomness in a QMC sequence, in such a way that $\mathbf{u}_n \sim \mathcal{U}\left([0, 1]^d\right)$, marginally. The quantity $\hat{I}_N = N^{-1} \sum_{n=1}^{N} \psi(\mathbf{u}_i)$ is then an unbiased estimate of the integral of interest. One may assess the approximation error by computing the empirical variance over repeated simulations.

The simplest way to obtain an RQMC sequence is to randomly shift a QMC sequence: Let $\mathbf{v} \sim \mathcal{U}\left([0, 1]^d\right)$, and $\mathbf{u}_{1:N}$ a QMC sequence; then

$$\hat{\mathbf{u}}_n := \mathbf{u}_n + \mathbf{v} \mod 1 \text{ (component wise)}$$

is an RQMC sequence.

A more sophisticated approach, called scrambled nets, was introduced by Owen (1997) and later refined in Owen (2008). The main advantage of this approach is that under the assumption of smoothness of the derivatives of the function, the speed of convergence can be even further improved, as stated in the following Theorem.

Theorem 1 *(Owen, 2008) Let $f : [0,1]^d \to \mathbb{R}$ be a function such that its cross partial derivatives up to order d exist and are continuous, and let $(\mathbf{u}_n)_{n\in 1:N}$ be a relaxed scrambled (λ, t, m, d)-net in base b with dimension d with uniformly bounded gain coefficients. Then,*

$$\mathrm{Var}\left(\frac{1}{N}\sum_{n=1}^{N} f(\mathbf{u}_n)\right) = \mathcal{O}\left(N^{-3}\log(N)^{(d-1)}\right),$$

where $N = \lambda b^m$.

In words, $\forall \tau > 0$ the RQMC error rate is $\mathcal{O}(N^{-3/2+\tau})$ when a scrambled (λ, t, m, d)-net is used. This result has the only inconvenience that the rate of convergence only holds for certain N. However, a more general result has recently been shown by Gerber (2015)[Corollary 1], where if $f \in L^2$ and $(\mathbf{u}_n)_{n\in 1:N}$ is a scrambled (t, d)-sequence, then $\forall N \in \mathbb{Z}^+$,

$$\mathrm{Var}\left(\frac{1}{N}\sum_{n=1}^{N} f(\mathbf{u}_n)\right) = o\left(N^{-1}\right).$$

The construction of scrambled nets and sequences is quite involved. As the focus of our paper is the application and not the construction of these sequences, we refer the reader for more details to L'Ecuyer (2016) or Dick et al. (2013). In the following, when speaking about an RQMC sequence, we will assume that this sequence is a scrambled (t, d)-sequence.

3.3 Mixed sequences and a central limit theorem

One drawback of low discrepancy sequences is that the speed of convergence deteriorates with the dimension. In some situations, a small number of components contributes significantly to the variance of the target. One then might choose to use a low discrepancy sequence for those components and an ordinary Monte Carlo approach for the rest. This idea of using a *mixed sequence* is closely linked to the concept of effective dimension, see Owen (1998). Based on the randomness induced by the Monte Carlo part a central limit theorem (CLT) may be established:

Theorem 2 *(Ökten et al., 2006) Let $u_k = (q_k^{1:d}, X_k^{d+1:s})$ be a mixed sequence of dimension s where $q_k^{1:d}$ denotes the deterministic QMC part and $X_k^{d+1:s}$ denotes the random independent MC part. Let $f : [0,1]^s \to \mathbb{R}^t, t \in \mathbb{Z}^+$ a bounded, square integrable function, $Y_k = f(u_k)$,*

$\hat{I}_N = N^{-1}\sum_{k=1}^N Y_k$, *and*

$$\begin{aligned}
\mu_k &:= \mathbb{E}[Y_k] = \int_{[0,1]^{s-d}} f(u_k)\mathrm{d}X^{d+1:s},\\
S_N &:= \frac{1}{N}\left(\sum_{k=1}^N Y_k - \sum_{k=1}^N \mu_k\right) = \left(\hat{I}_N - \frac{1}{N}\sum_{k=1}^N \mu_k\right),\\
\sigma_k^2 &:= \mathrm{Var}[Y_k] = \int_{[0,1]^{s-d}} f(u_k)f(u_k)^T\mathrm{d}X^{d+1:s}\\
&\quad - \left(\int_{[0,1]^{s-d}} f(u_k)\mathrm{d}X^{d+1:s}\right)\left(\int_{[0,1]^{s-d}} f(u_k)\mathrm{d}X^{d+1:s}\right)^T,\\
C_N^2 &:= \mathrm{Var}[N\hat{I}_N] = \sum_{k=1}^N \sigma_k^2.
\end{aligned}$$

Then, as $N \to +\infty$, $C_N^2/N \to C^2_{\mathrm{qmc-mixed}}$ *and*

$$N^{1/2}S_N \xrightarrow{\mathcal{L}} \mathcal{N}\left(0, C^2_{\mathrm{qmc-mixed}}\right),$$

where

$$\begin{aligned}
C^2_{\mathrm{qmc-mixed}} &= \int_{[0,1]^s} f(x)f(x)^T\mathrm{d}x\\
&\quad - \int_{[0,1]^d}\left(\int_{[0,1]^{s-d}} f(u)\mathrm{d}X^{d+1:s}\right)\left(\int_{[0,1]^{s-d}} f(u)\mathrm{d}X^{d+1:s}\right)^T \mathrm{d}q^{1:d}.
\end{aligned}$$

As a direct corollary of the previous Theorem we obtain that, provided f has a finite variation in the sense of Hardy and Krause, $N^{1/2}(\hat{I}_N - I) \xrightarrow{\mathcal{L}} \mathcal{N}(0, C^2_{\mathrm{qmc-mixed}})$, where $I = \int f(u)\mathrm{d}u$. This is due to the fact that

$$N^{1/2}\left(\hat{I}_N - I\right) = N^{1/2}S_N + N^{1/2}\left(\frac{1}{N}\sum_{k=1}^N \mu_k - I\right)$$

and the second term on the right hand side converges deterministically to 0. Ökten et al. (2006) present only a univariate version of their central limit theorem; the extension to the multivariate case is straightforward.

Moreover, their work shows that the asymptotic variance of the mixed sequence estimator is smaller than for the same estimator based on Monte Carlo sequences in dimension one. We extend this result to the multivariate case.

Corollary 1 *Let* $C^2_{\mathrm{qmc-mixed}}$ *be the asymptotic variance of an estimator based on a mixed sequence as defined in Theorem 2. Let* C^2_{mc} *be the variance of the same estimator based on a pure MC sequence, e.g., when* $d = 0$. *Then*

$$C^2_{\mathrm{qmc-mixed}} \preceq C^2_{\mathrm{mc}}$$

in the sense of positive definite matrices.

Moreover, we present a result here that allows us to apply the same technique to mixed sequences that combine Monte Carlo and randomized quasi-Monte Carlo sequences.

Theorem 3 *Let S_N^{RQMC} be the MC-RQMC equivalent of S_N under the same conditions as in Theorem 2. Then*

$$N^{1/2} S_N^{\mathrm{RQMC}} \xrightarrow{\mathcal{L}} \mathcal{N}(0, C^2_{\mathrm{rqmc-mixed}}),$$

where $C^2_{\mathrm{rqmc-mixed}} = C^2_{\mathrm{qmc-mixed}}$.

These results may be understood as follows. The randomness in the Monte Carlo sequence allows the construction of a central limit theorem. The part associated to the (R)QMC sequences converges faster to zero than the part associated to the Monte Carlo sequence. This leads to a reduced asymptotic variance for estimators based on mixed sequences.

4 Improved ABC via (R)QMC

Recall that we described our ABC importance sampler as an algorithm that samples pairs (θ_n, x_n) from $q(\theta)q_\theta(x)$, where x_n consists of datapoints generated from the model. In most ABC problems, using (R)QMC to generate the θ_n should be easy, but this should not be the case for the x_n's. Indeed, the simulator used to generate datapoints from the model may be a complex black box, which may require a very large, or random, number of uniform variates. Thus, we contemplate from now on generating the θ_n's using (R)QMC. That is, $\theta_n = \Gamma(\mathbf{u}_n)$, where $\mathbf{u}_{1:N}$ is a QMC or RQMC sequence, and Γ is a function such that $\Gamma(U)$, $U \sim \mathcal{U}\left([0,1]^d\right)$, is distributed according to the proposal $q(\theta)$; and $x_n|\theta_n \sim q_{\theta_n}$ is a random variate. In other words, (θ_n, x_n) is a mixed sequence.

We already know from the previous section that an estimate based on a mixed sequence converges at the Monte Carlo rate, $\mathcal{O}_P(N^{-1/2})$, but has a smaller asymptotic variance than the same estimate based on Monte Carlo. In fact, a similar result may be established directly for the actual variance. Let $\hat{I}_N := \sum_{n=1}^N \varphi(\theta_n, x_n)/N$ be an empirical average for some measurable function φ. For simplicity, we assume here that the θ_n's are either random variates, or RQMC variates. That is, in both cases, $\theta_n \sim q$ marginally. Then

$$\begin{aligned}
\mathrm{Var}[\hat{I}_N] &= \mathrm{Var}\left[\mathbb{E}\{\hat{I}_N|\theta_{1:N}\}\right] + \mathbb{E}\left[\mathrm{Var}\{\hat{I}_N|\theta_{1:N}\}\right] \\
&= \mathrm{Var}\left[\frac{1}{N}\sum_{n=1}^N \mathbb{E}_{x_n\sim q_{\theta_n}}\left\{\varphi(\theta_n, x_n)|\theta_n)\right\}\right] + \frac{1}{N} \times \mathbb{E}_{\theta_n\sim q}\left[\mathrm{Var}_{x_n\sim q_{\theta_n}}\left\{\varphi(\theta_n, x_n)|\theta\right\}\right]
\end{aligned} \tag{4}$$

The first term is $\mathcal{O}(N^{-1})$ when the θ_n's are generated using Monte Carlo, and should be $o(N^{-1})$ under appropriate conditions when the θ_n's are an RQMC sequence. On the other hand, the second term is $\mathcal{O}(N^{-1})$ in both cases. As a corrolary, the variance of $\hat{I}_N$ is smaller when using a mixed sequence, for N large enough.

The point of the following sections is to generalize this basic result to various ABC estimates of interest.

4.1 Improved estimation of the normalization constant

We first consider the approximation of the normalization constant of the ABC posterior:

$$Z_\epsilon = \int \mathbb{P}_\theta \left(\delta(y, y^\star) \leq \epsilon\right) p(\theta) \mathrm{d}\theta = \int \hat{L}_\epsilon(x) q_\theta(x) p(\theta) \mathrm{d}x \mathrm{d}\theta.$$

Recall that, for the moment, we take $x = y_{1:M}$, $q_\theta(x) = \prod_{m=1}^M p(y_m|\theta)$ and

$$\hat{L}_\epsilon(x) = \frac{1}{M} \sum_{m=1}^M \mathbb{1}\left\{\delta(y_m, y^\star) \leq \epsilon\right\}.$$

Thus, a natural estimator of Z_ϵ is

$$\hat{Z}_N := \frac{1}{N} \sum_{n=1}^N \frac{p(\theta_n)}{q(\theta_n)} \left[\frac{1}{M} \sum_{m=1}^M \mathbb{1}\left\{\delta(y_{n,m}, y^\star) \leq \epsilon\right\}\right] \tag{5}$$

where the θ_n's are either a Monte Carlo or RQMC sample from the proposal $q(\theta)$, and $y_{n,m} \sim p(y|\theta_n)$ for $n = 1, \ldots, N$, $m = 1, \ldots, M$.

When the θ_n's are a Monte Carlo sample, it is always best to take $M = 1$, as noted by Bornn et al. (2015). This may be seen by calculating both terms of decomposition (4) when applied to the estimator of the normalization constant $\hat{Z}_N$:

$$\mathrm{Var}\left[\mathbb{E}\{\hat{Z}_N|\theta_{1:N}\}\right] = \frac{1}{N} \times \mathrm{Var}_q \left[\frac{p(\theta)}{q(\theta)} \mathbb{P}_\theta \left(\delta(y, y^\star) \leq \epsilon\right)\right] \tag{6}$$

$$\mathbb{E}\left[\mathrm{Var}\{\hat{Z}_N|\theta_{1:N}\}\right] = \frac{1}{NM} \times \int_\Theta \frac{p(\theta)^2}{q(\theta)} \mathbb{P}_\theta \left(\delta(y, y^\star) \leq \epsilon\right) \left\{1 - \mathbb{P}_\theta \left(\delta(y, y^\star) \leq \epsilon\right)\right\} \mathrm{d}\theta. \tag{7}$$

Increasing M increases the CPU cost and decreases the variance of $\hat{Z}_N$. To account for both simultaneously, we look at the adjusted variance, $M \times \mathrm{Var}\left[\hat{Z}_N\right]$. From (6) and (7), we see that the adjusted variance increases with M, hence the best CPU time vs error trade-off is obtained by taking $M = 1$.

Now, consider the situation where the θ_n's form an RQMC sequence. As noted in the previous section, (7) still holds due to the unbiasedness property of RQMC sequences, however the first (6) term of the decomposition should converge faster.

Proposition 1 *Let $f(\theta) = \{p(\theta)/q(\theta)\}\mathbb{P}_\theta \left(\delta(y, y^\star) \leq \epsilon\right)$, assume that $\theta_n = \Gamma(\mathbf{u}_n)$ where $\mathbf{u}_{1:N}$ is a scrambled (λ, t, m, d)-net, and assume that $f \circ \Gamma \in L^2$. Then,*

$$\mathrm{Var}\left[\mathbb{E}\{\hat{Z}_N|\theta_{1:N}\}\right] = o\left(N^{-1}\right).$$

This result is a direct consequence of Corollary 1 of Gerber (2015) and the fact

$$\mathbb{E}\{\hat{Z}_N|\theta_{1:N}\} = \frac{1}{N} \sum_{n=1}^N f(\theta_n) = \frac{1}{N} \sum_{n=1}^N f \circ \Gamma(\mathbf{u}_n).$$

It has two corollaries. First, the variance of $\hat{Z}_N$ is smaller when using a RQMC sequence for the θ_n's (for N large enough). Second, in that case, the adjusted variance is such that

$M \operatorname{Var}[\hat{Z}_N] = \mathcal{O}(N^{-1})$, with a constant that does not depend on M. Thus taking $M > 1$ (within a reasonable range) should have basically no impact on the CPU time vs error trade-off in the RQMC case.

Taking $M > 1$ has the following advantage: it makes it possible to consistently estimate (7) with the quantity

$$\hat{\sigma}^2(Z_\epsilon) := \frac{1}{N^2(M-1)} \times \sum_{n=1}^{N} \frac{p(\theta_n)^2}{q(\theta_n)^2} \hat{L}_\epsilon(x_n)\{1 - \hat{L}_\epsilon(x_n)\}. \tag{8}$$

where $\hat{L}_\epsilon(x_n) = M^{-1} \sum_{m=1}^{M} \mathbb{1}\{\delta(y_{n,m}, y^\star) \leq \epsilon\}$. As (7) corresponds to the non-negligible part of the variance of $\hat{Z}_N$, this allows us to obtain asymptotic confidence intervals for $\hat{Z}_N$.

We have focused on the RQMC case for now on, but a similar result holds for QMC sequences. Note, however, that we cannot use directly decomposition (4) when the θ_n's are deterministic.

Proposition 2 *Assume that $\mathbf{u}_{1:N}$ is a deterministic low-discrepancy sequence, that $f \circ \Gamma$ (where f and Γ are defined as in Proposition 1) has a finite variation in the sense of Hardy and Krause, and that the ratio p/q is upper-bounded, $p(\theta)/q(\theta) \leq C$, then*

$$M \times \mathbb{E}\left[\left(\hat{Z}_N - Z_\epsilon\right)^2\right] = \mathcal{O}(N^{-1})$$

with a constant that does not depend on M. Furthermore, the mean square error above is smaller than in the Monte Carlo case, for N large enough.

4.2 Improved estimation of general importance sampling estimators

We now turn to the analysis of general importance sampling estimators of the form

$$\hat{\phi}_N = \frac{\sum_{n=1}^{N} w_n \phi(\theta_n)}{\sum_{n=1}^{N} w_n}. \tag{9}$$

As these estimators are ratios, we cannot apply decomposition (4) directly. However, we may apply the following inequality, due to Agapiou et al. (2015):

$$\mathbb{E}\left\{\hat{\phi}_N - \mathbb{E}_{p_\epsilon}\phi\right\}^2 \leq \frac{2}{Z_\epsilon^2}\left(\mathbb{E}\left\{\frac{1}{N}\sum_{n=1}^{N} w_n\phi(\theta_n) - Z_\epsilon \mathbb{E}_{p_\epsilon}\phi(\theta)\right\}^2 + \mathbb{E}\left\{\frac{1}{N}\sum_{n=1}^{N} w_n - Z_\epsilon\right\}^2\right)$$

provided $|\phi| \leq 1$. Both terms are mean squared errors of empirical averages, and hence may be bounded directly using a decomposition of variance and the results of the previous section. Thus, we see that, again, when the θ_n are generated with (R)QMC, the mean squared error of estimate $\hat{\phi}_N$ is $\mathcal{O}(M^{-1}N^{-1})$ as $N \to +\infty$. However, this inequality does not make it possible to compare the performance of our RQMC-ABC procedure with Monte Carlo-based ABC. For this, we now consider the asymptotic behavior of these estimators.

Theorem 4 *Let $\phi : \Theta \to \mathbb{R}$ be a bounded function, $\bar{\phi} = \phi - \mathbb{E}_{p_\epsilon}\phi$, $\hat{\phi}_N$ defined as (9), then, under the same conditions as Proposition 2, and assuming further that function $\mathbf{u} \to \bar{\phi}(\Gamma(\mathbf{u}))f(\Gamma(\mathbf{u}))$ has a finite variation (in the sense of Hardy and Krause), one has that*

$$N^{1/2}\left(\hat{\phi}_N - \mathbb{E}_{p_\epsilon}\phi\right) \xrightarrow{\mathcal{L}} \mathcal{N}\left(0, \sigma^2_{mixed}(\phi)\right),$$

where, using the short-hand $b(\theta) = \mathbb{P}_\theta\left(\delta(y, y^\star) \leq \epsilon\right)$,

$$\sigma^2_{\text{mixed}}(\phi) = \frac{1}{MZ_\epsilon^2} \int_\Theta \frac{p(\theta)^2}{q(\theta)} \bar{\phi}(\theta)^2 b(\theta)\{1 - b(\theta)\} \mathrm{d}\theta. \tag{10}$$

Alternatively, if the parameter values θ_n were generated through Monte Carlo sampling, one would obtain a similar central limit theorem, but with asymptotic variance

$$\sigma^2_{\text{MC}}(\phi) = \frac{1}{Z_\epsilon^2} \int_\Theta \frac{p(\theta)^2}{q(\theta)} \bar{\phi}(\theta)^2 \left[\frac{b(\theta)\{1 - b(\theta)\}}{M} + b(\theta)^2\right] \mathrm{d}\theta$$

which is larger than or equal to $\sigma^2_{\text{mixed}}(\phi)$.

It is possible to obtain a similar result for RQMC sequences by using Theorem 3.

As for the normalising constant, we observe that the adjusted (asymptotic) variance, i.e. $M \times \sigma^2_{\text{mixed}}(\phi)$, is constant with respect to M. Thus, taking $M > 1$ does not deteriorate the performance of the algorithm (in terms of variance relative to CPU time). And it makes it possible to estimate consistently the asymptotic variance (10) (and thus compute confidence intervals) using

$$\widehat{\sigma}^2_{\text{mixed}}(\phi) = \frac{1}{(\hat{Z}_N)^2 N(M-1)} \sum_{n=1}^{N} \frac{p(\theta_n)^2}{q(\theta_n)} \{\phi(\theta_n) - \hat{\phi}_N\}^2 \hat{L}_\epsilon(x_n) \left\{1 - \hat{L}_\epsilon(x_n)\right\}.$$

5 Numerical examples

We illustrate in this section the improvement brought by (R)QMC through several numerical examples. Code for reproducing the results of this section and of Section 7 is available at `https://github.com/alexanderbuchholz/ABC`.

Thus we compare three different approaches, all corresponding to Algorithm 1, but with particles generated using either Monte Carlo (ABC-IS), Quasi-Monte Carlo (ABC-QMC), or randomised QMC (ABC-RQMC). For the generation of the (R)QMC sequences we use the `R` package `randtoolbox` (Christophe and Petr, 2015) and generate Sobol sequences (QMC), or Owen-type scrambled Sobol sequences (RQMC), see Owen (1998).

We take $q(\theta) = p(\theta)$, i.e. points are generated from the prior, and, unless explicitely stated, we take $M = 1$. (The problem of adaptively choosing q will be considered in the next section.)

In this case, weights w_n are either 0 or 1 (according to whether $\delta(y_n, y^\star) \leq \epsilon$), and we set ϵ so that the proportion of non-zero weights is close to some pre-specified value, e.g. 10^{-3}.

5.1 Toy model

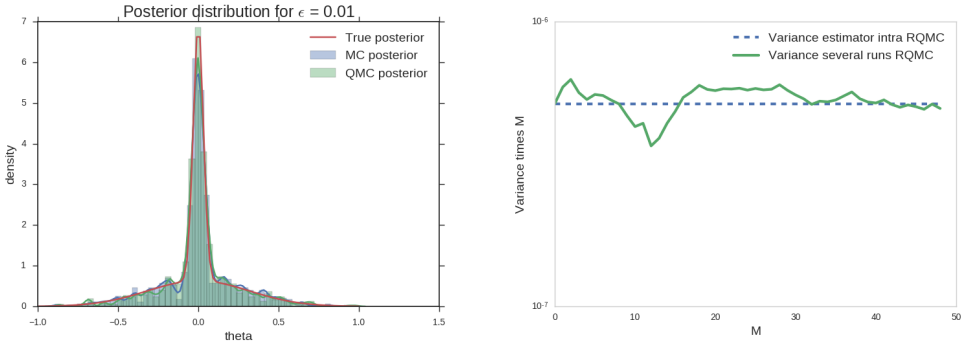
The first model we consider is the toy model used in Marin et al. (2012) that tries to recover the mean of a superposition of two Gaussian distributions with identical mean and different variances:

$$\begin{aligned} \theta &\sim \mathcal{U}\left([-10, 10]^d\right), \\ y|\theta &\sim \frac{1}{2}\mathcal{N}(\theta; 0.1 I_d) + \frac{1}{2}\mathcal{N}(\theta; 0.001 I_d). \end{aligned}$$

The use of this model is motivated by the fact that the dimension of the model d can be scaled up easily. We set $y^\star = 0_d$ and $\delta(y, y^\star) = \|y - y^\star\|_2$. Posterior density (1) may be calculated exactly in this particular case. (The resulting expression depends on the cdfs of non-central χ^2 distributions.)

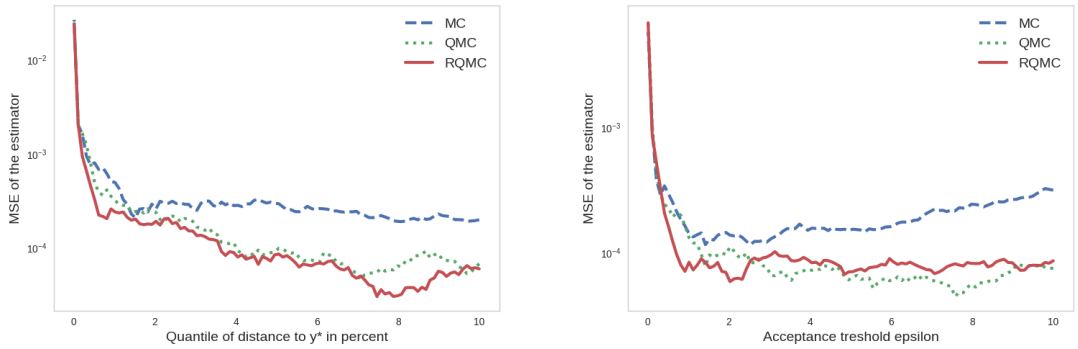
We run the three considered algorithms with $N = 10^6$. Figure 2a shows that the MC and QMC approximations match closely; for this plot, $\epsilon = 0.01$ (leading to a proportion of non-zero weights close to 10^{-3}), and $d = 1$.

Figure 3 compares the empirical variance (over 50 runs) obtained with the three considered approaches, as a function of ϵ, when estimating the expectation (left pane) and variance (right pane) of the ABC posterior. Here, $N = 10^6$, $d = 2$, and ϵ is chosen so as to generate a proportion of non-zero weights that vary from 0 to 10%.



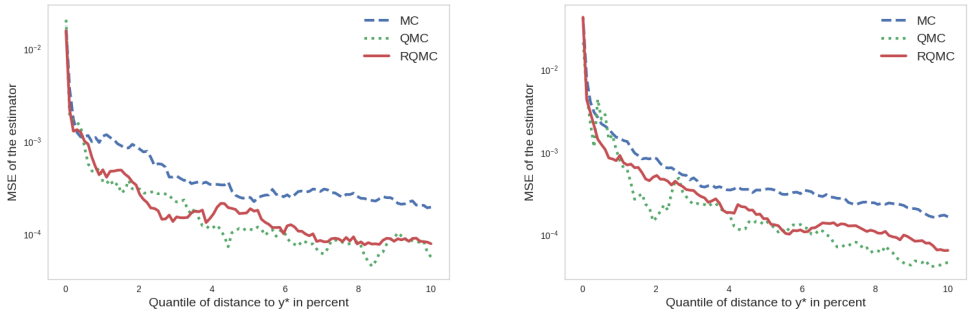
(a) (b)

Figure 2: Left: Kernel density estimation of the approximation of the posterior distribution based on $N = 10^6$ simulations and the threshold $\epsilon = 0.01$ for $d = 1$. The exact posterior can be calculated analytically. The approaches based on MC and QMC essentially recover the same distribution. Right: Adjusted variance (variance times M) of the normalization constant as a function of M: the dashed line corresponds to the variance estimator given by (8), the solid line corresponds to the empirical variance of the estimator based on 75 runs. The results are based on $N = 10^5$ simulations, $\epsilon = 1$, $d = 1$, and an RQMC sequence for the θ_n's. The adjusted variance stays roughly constant for $M > 1$.



(a) (b)

Figure 3: MSE of posterior estimates as ϵ varies (Left: ABC posterior mean; Right: ABC posterior variance). The plots are based on 50 runs, with $N = 10^6$ simulations and $d = 2$. The $x-$axis corresponds to a varying ϵ, which is set so that the proportion of non-zero weights (i.e. the proportion of simulated y_n such that $\delta(y_n, y^\star) \leq \epsilon$) varies from 0 to 10%. (R)QMC sequences lead to a reduced MSE. The effect vanishes as ϵ goes to 0.



(a) Left (b) Right

Figure 4: Same caption as for Figure 3b, except left (resp. right) panel corresponds to $d = 4$ (resp. $d = 8$); posterior estimate is the ABC posterior expectation in both cases.

We observe a variance reduction when using either QMC or RQMC and for not too small values of ϵ, but the variance reduction vanishes as $\epsilon \to 0$. However, interestingly, the variance reduction (again for not too small values of ϵ) remains significant when we increase the dimension, see Figures 4. (For $d > 1$, the considered estimated quantity is the expectation of the average of the d components of θ with respect to the ABC posterior.)

Finally, we consider increasing M, so as to be able to estimate the variance of a given ABC estimate from a single run of Algorithm 1, when using (R)QMC, as explained at the end of Section 4.1. The considered estimate is that of the normalising constant of the ABC posterior. We see that the variance estimate is fairly stable even for small values of M, and that it is close to the actual variance (over 75 runs) of the estimate as can be seen in Figure 2b.

Note that both quantities are multiplied by M in Figure 2b. This allows us to check that

the adjusted variance (accounting for CPU time) remains constant, as expected. As already explained, this means that taking $M > 1$ is not sub-optimal (in terms of the variance vs CPU time trade-off), while it allows us to estimate the variance of any estimate obtained from the (R)QMC version of Algorithm 1.

5.2 Lotka-Volterra-Model

The Lotka-Volterra model, see Toni et al. (2009), is commonly used in population dynamics to study the interaction in predator-prey models, for example. The model is characterized by the respective size of the populations evolving over time and denoted by (X_1, X_2), taking values in $\mathbb{Z}^2$.

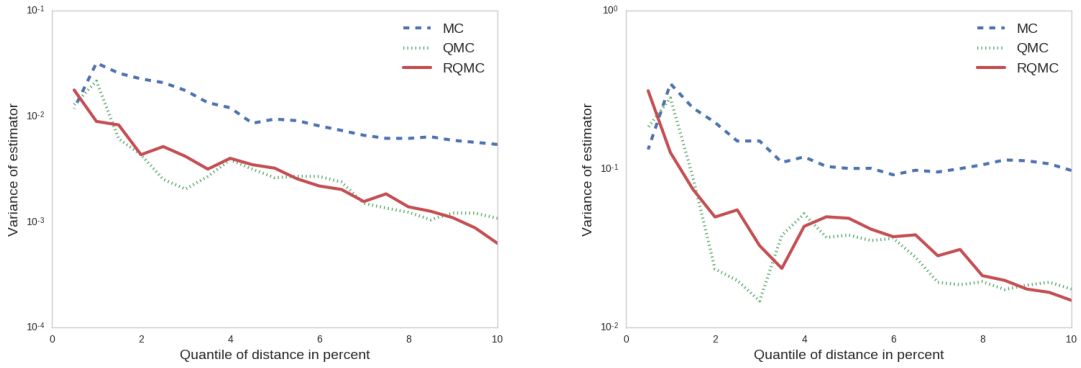
There are three possible transitions: the prey (denoted by X_1) may grow by one entity with rate α, a predation may happen with rate β, that reduces the prey by one unit and increases the predator population (denoted by X_2) by one unit, or the predator may die with rate γ. The system is summarized by the following rate equations:

$$\begin{aligned}(X_1, X_2) &\xrightarrow{\alpha} (X_1 + 1, X_2),\\ (X_1, X_2) &\xrightarrow{\beta} (X_1 - 1, X_2 + 1),\\ (X_1, X_2) &\xrightarrow{\gamma} (X_1, X_2 - 1),\end{aligned}$$

with the corresponding hazard rates αX_1, $\beta X_1 X_2$ and γX_2, respectively. The hazard rates characterize the instantaneous probability that the system changes to a new state. The parameter of the model is $\theta = (\alpha, \beta, \gamma)$. The initial population is fixed to $(50, 100)$.

We simulate from the model using Gillespie's algorithm, see Toni et al. (2009), for $T = 30$ time steps, and record the size of the population at times $t_i = 2i$, where $i = 0, \cdots, 15$. This gives two discrete time series of length 16. As a distance function for comparing our true observation and the pseudo-observations, we use the Euclidean norm $\|\cdot\|_2$ applied to the differences of the series. As a prior we use $\mathbf{u} \sim \mathcal{U}[-6, 2]^3$, which is then transformed to $\theta = \exp(\mathbf{u})$.

As in the previous section, we compare the empirical variance over 50 runs of a given estimate obtained from the different approaches. The estimated quantity is the expectation of $(\alpha + \beta + \gamma)/3$ with respect to the ABC posterior.



(a) (b)

Figure 5: Variance of the mean and variance estimator for the Lotka–Volterra model. The plots are based on 50 repetitions of 10^5 simulations from the prior and the model. The accepted observations correspond to quantiles based on the smallest distances $\delta(y_n, y^\star)$. Left: Variance of the posterior mean estimator. Right: Variance of the posterior variance estimator

We observe the same phenomenon as in the previous example: the variance reduction brought by either QMC or RQMC is significant for not too small values of ϵ, but it vanishes as $\epsilon \to 0$.

5.3 Tuberculosis mutation

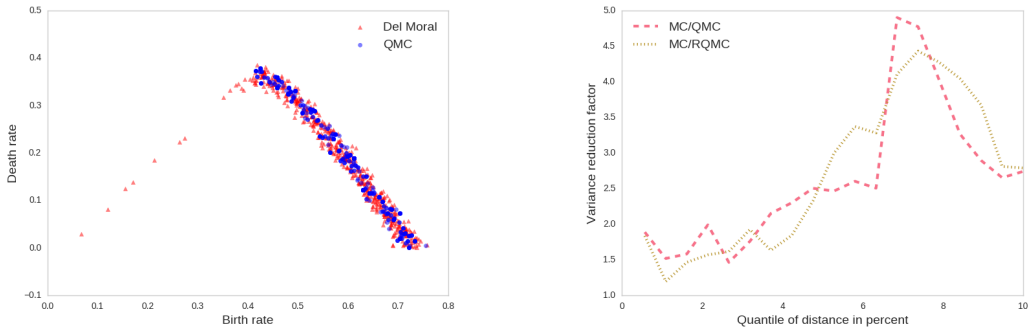
The following application is based on the estimation of tuberculosis reproduction rates as in Tanaka et al. (2006). The interest lies in recovering the posterior distribution of birth, death and mutation rates (α, β, γ) of a tuberculosis population that has been recorded in San Francisco over a period from 1991 to 1992.

The simulator of the model is based on an underlying continuous time Markov process where t denotes the time and $N(t)$ denotes the size of the population. Starting from one single bacterium the individual can either replicate itself with rate α, die with rate γ or mutate to a new genotype with rate β. The number of bacteria having the same genotype is recorded at every step and the simulation is run forward until a size of $N(t) = 10^4$ has been obtained. At every step in the simulation a bacterium is chosen uniformly at random and one of the three events (α, β, γ) is applied to it. After simulating a population of 10^4 bacteria, the simulation is stopped and a subpopulation of 473 bacteria is sampled. The ensuing population is characterized by the cluster size of bacteria that have the same genotype. The data is available in Table 1. For instance, there were 282 clusters with only one bacterium with the same genotype and there were 20 clusters that contained two bacteria with the same genotype.

Cluster size	1	2	3	4	5	8	10	15	23	30
Number of clusters	282	20	13	4	2	1	1	1	1	1

Table 1: Tuberculosis bacteria genotype data

The parameters must satisfy the conditions $\alpha + \beta + \gamma = 1$, $0 \leq \alpha, \beta, \gamma \leq 1$, and $\alpha > \gamma$. (The last constraint prevents the population from dying out.) Thus, we let $\beta = 1-\alpha-\gamma$, and assign a uniform prior to (α, γ), subject to $\alpha > \gamma$. Tanaka et al. (2006) used as a summary statistic for the data the quantities $y = (g/473, 1 - \sum_i (n_i/473)^2)$, where g denotes the number of distinct clusters in the sample and n_i is the number of observed bacteria in the ith genotype cluster. The distance between a pseudo observation and the observed data is finally calculated as the Euclidean distance between y and $y^\star$. Figure 6a shows the recovered posterior distribution after application of a sequential sampling approach, that is described in Section 7. We see our method, denoted by QMC and the method of Del Moral et al. (2012), denoted by *Del Moral* recover the same posterior distribution. There remain some artifacts in the second method, due to a slightly higher acceptance threshold $\epsilon = 0.12$ compared to $\epsilon = 0.08$ as in the QMC approach. We estimate the ABC posterior expectation of $(\alpha + \gamma)/2$ and then compare the empirical variance of this estimator. The result of the repeated simulation of this estimator is shown in Figure 6b, where we show the value of $\mathrm{Var}_{MC} \,/\, \mathrm{Var}_{(R)QMC}$, where Var_{MC} is the variance of the posterior estimator based on a MC sequence. This quantity allows to assess the variance reduction factor as a function of the acceptance threshold. Again, we observe a declining variance reduction as $\epsilon \to 0$. Nevertheless, the variance reduction even for the smallest acceptance threshold is still of factor 1.5, which means that we need 33% fewer simulations in order to achieve the same precision of the estimator.



(a) (b)

Figure 6: Left: Posterior distribution of the tuberculosis mutation model. The x–axis corresponds to birth rate α, the y–axis corresponds to the death rate β, $N = 500$. Right: Variance reduction factors (computed from 50 runs based on $N = 10^4$) as a function of the proportion of non-zero weights.

5.4 Concluding remarks

As predicted by the theory, we observed that using QMC (or RQMC) to generate the parameter values (in Algorithm 1) always reduce the variance of ABC estimates. However, the variance reduction becomes small when $\epsilon \to 0$. But it should be noted that any static ABC algorithm, such as Algorithm 1 becomes very wasteful when ϵ is small, as most simulated datapoints lies outside the ball defined by the constraint $\delta(y, y^\star) \leq \epsilon$ in such a case. In order to take ϵ smaller and smaller, it seems to make more sense to progressively refine the proposal distribution, based on past simulations. This is the point of sequential

ABC algorithms, which we discuss in the next two sections.

6 Sequential ABC

6.1 Adaptive importance sampling

One major drawback of Algorithm 1 is that the quality of the approximation in (2) depends on how well the proposal distribution $q(\theta)$ matches the target distribution $p_\epsilon(\theta)$. If, for example, the proposal is very flat and the target is spiky due to a small value of ϵ, only a small number of particles will cover the region of interest. The idea of sequential ABC algorithms is therefore to sequentially decrease ϵ over a range of time steps $t \in 0 : T$ while adapting the proposal distribution $q_t(\theta)$ so as to make it closer and closer to the true posterior.

In the current setting we will use a flexible parametric approximation $q_t(\theta)$ of the ABC posterior $p_{\epsilon_t}(\theta)$, that is estimated from the the samples $(\theta_n^{t-1}, w_n^{t-1})_{n\in 1:N}$. This distribution $q_t(\theta)$ is then used to simulate new particles $(\theta_n^t)_{n\in 1:N}$. The corresponding algorithm is given as pseudo-code in Algorithm 2.

Input: Observed $y^\star$, prior distribution $p(\theta)$, simulator $q_\theta(x)$, initial threshold ϵ_0, number of simulations N, weighting procedure $\hat{L}_\epsilon(x)$.
Result: Set of weighted samples $(\theta_n^t, x_n^t, w_n^t)_{n\in 1:N, t\in 0:T}$
for $n = 1$ *to* N **do**
  Sample $\theta_n^0 \sim p(\theta)$;
  set $w_n^0 = 1$;
end
for $t = 1$ *to* T **do**
  Set ϵ_t and $q_t(\theta)$ based on $(\theta_n^{t-1}, x_n^{t-1}, w_n^{t-1})_{n\in 1:N}$;
  for $n = 1$ *to* N **do**
    Sample $\theta_n^t \sim q_t(\theta)$;
    Sample $x_n^t \sim q_{\theta_n^t}(x)$;
    Set $w_n^t = p(\theta_n^t)\hat{L}_{\epsilon_t}(x_n^t)/q_t(\theta_n^t)$;
  end
end

Algorithm 2: ABC adaptive importance sampling algorithm

6.2 Adapting the proposal q_t

6.2.1 Gaussian proposal

The simplest strategy one may think of to adapt q_t is to set it to a Gaussian fit of the previous weighted sample. Although basic, we shall see that this approach tends to work well in practice, unless of course the actual posterior is severely multimodal, strongly skewed or has heavy tails.

6.2.2 Mixture of N components

The sequential Monte Carlo sampler (SMC) of Sisson et al. (2009) may be viewed as a particular version of Algorithm 2, where q_t is set to a mixture of N Gaussian components

centred on the N previous particles θ_n^{t-1}, with covariance matrix $\hat{\Sigma}^{t-1}$ set to twice the empirical covariance of these particles. The proposal distribution reads

$$q_t(\theta) = \frac{\sum_{n=1}^N w_n^{t-1} \mathcal{N}(\theta|\theta_n^{t-1}, 2\hat{\Sigma}^{t-1})}{\sum_{n=1}^N w_n^{t-1}}.$$

This results in an algorithm of complexity $\mathcal{O}(N^2)$ since for every proposed new particle θ_n^t, computing the corresponding weight involves a sum over N terms.

6.2.3 Mixture proposal with a small number of components

As an intermediate solution between a single Gaussian distribution and a mixture of N Gaussian distributions, we suggest to use a Gaussian mixture with a small number of components. We suggest to estimate the mixture via a Variational Bayesian procedure, see Blei et al. (2016), but other methods as Expectation Maximization could also be used. The proposal distribution reads

$$q_t(\theta) = \sum_{j=1}^J \alpha_j^{t-1} \mathcal{N}(\theta|\hat{\mu}_j^{t-1}, \lambda\hat{\Sigma}_j^{t-1}),$$

where α_j^{t-1}, $\hat{\mu}_j^{t-1}$, and $\hat{\Sigma}_j^{t-1}$ denote respectively the weight, mean, and covariance matrix of cluster j estimated at iteration $t-1$. Again, we artificially inflate the covariances with a factor $\lambda > 1$ in order to put more mass in the tails of the proposal distribution. In our numerical experiments we set $\lambda = 1.2$. Regarding J, we may either fix it arbitrarily or use the Variational Bayesian approach to choose it automatically.

In order to generate QMC or RQMC points from such a mixture distribution, we set the number of samples for each cluster j to $N_j^t = \left\lfloor \alpha_j^{t-1} N \right\rfloor$ and potentially adjust N_j^t as to make sure that $\sum_j N_j^t = N$ holds. For each cluster j, a (R)QMC sequence of length N_j^t is generated and transformed to the sample of a Gaussian distribution $\mathcal{N}(\theta|\hat{\mu}_j^{t-1}, \lambda\hat{\Sigma}_j^{t-1})$. This is achieved via the transformation of the (R)QMC sequence $(\mathbf{u}_n)_{n\in 1:N_j^t}$ via the component-wise quantile function $\Phi^{-1}(\cdot)$: $\theta_n^t = \hat{\mu}_j^{t-1} + C_{t-1}\Phi^{-1}(\mathbf{u}_n)$, where C_{t-1} is the Cholesky triangle of the covariance matrix: $C_{t-1}(C_{t-1})^T = \lambda\hat{\Sigma}_j^{t-1}$.

This approach has the following advantages. First, we maintain flexibility by allowing to cover several modes, as the posterior distribution might be multi–modal. Second, the use of a limited number of clusters makes sure that we can benefit from the better coverage of the space that comes from the use of (R)QMC sequences. Using only a small number of clusters preserves the structure of the (R)QMC point set. Other approaches based on the inverse Rosenblatt transform (Gerber and Chopin, 2015) are computationally more expensive. In contrast, using the approach of Sisson et al. (2009) would destroy the properties of the low discrepancy or scrambled net sequences and hence the variance reduction that comes from the (R)QMC sequence could vanish. (This has been found as a result of our simulation studies, not shown here.)

6.3 Adapting simultaneously ϵ_t and the number of simulations per parameter

As discussed in Section 2.2, the weights $\hat{L}_{\epsilon_t}(x_n^t)$ are unbiased estimators of the probabilities $P_{\theta_n^t}(\delta(y^\star, y) \leq \epsilon_t)$, which may be obtained in two ways: (a) as an average over a fixed

number M of simulations; or (b) as a function of the number of simulations required so that k of them are at a ϵ distance of $y^\star$; that random number follows a negative binomial distribution.

So far, we have focused on (a), and even took $M = 1$ in our first set of numerical examples in Section 5. If we use this strategy, we may follow Del Moral et al. (2012) in adapting ϵ_t according to the ESS (effective sample size, Kong et al., 1994); i.e. at iteration t, once we have simulated the θ_n^t's and the x_n^t's, we solve numerically (using bisection) in ϵ_t the equation $\text{ESS} = \alpha N$, for $\alpha \in (0, 1)$, where

$$\text{ESS} = \frac{(\sum_{n=1}^N w_n^t)^2}{\sum_{n=1}^N (w_n^t)^2}$$

and $w_t^n = p(\theta_n^t)\hat{L}_\epsilon(x_n^t)/q(\theta_n^t)$, $\hat{L}_\epsilon(x_n^t) = M^{-1}\sum_{m=1}^M \mathbb{1}\{\delta(y^\star, y_{n,m}^t) \leq \epsilon_t\}$.

This approach usually works well during the first iterations of Algorithm 2, but it is bound to collapse as ϵ gets too small: as $\epsilon \to 0$, $P_\theta(\delta(y^\star, y) \leq \epsilon) \to 0$ whatever θ, and as a result most weights w_t^n become zero when ϵ_t is too small. One remedy is to set M to a much larger value, so that weights take much longer to collapse. However, this is expensive and wasteful, given that the first iterations would work well with a much smaller M.

In that sequential context, the negative binomial strategy for computing the weights becomes appealing, as it makes it possible to adapt automatically the CPU effort to a given ϵ: we may decrease ϵ_t at each iteration, while ensuring that the variance of the weights (as estimates of the probabilities $P_{\theta_n^t}(\delta(y^\star, y) \leq \epsilon_t)$ does not blow up. Of course, the price to pay is that iterations become more and more expensive.

In practice, we found that that this approach was unwieldy during the first iterations of the algorithm: during that time, a few simulated parameters θ_n^t are such that the corresponding probability that $\delta(y^\star, y) \leq \epsilon_t$ is much smaller than for the other particles. As a result, the negative binomial estimate requires generating a lot of observations for those particles, which typically gets discarded later.

Thus, in the end, we recommend the following hybrid strategy:

- At iterations $t = 0$ to $t = T_1$ (say $T_1 = 10$), use the 'fixed M' (say $M = 10$) strategy to compute the weights, and adapt ϵ_t using the ESS.
- At iterations $t > T_1$, switch to the negative binomial strategy for computing the weights, and adapt ϵ_t as follows: set it to the median of the distance values $\delta(y^\star, y_n)$ where the y_n's denote here all the artificial observations generating during the previous iteration such that $\delta(y^\star, y_n) \leq \epsilon_{t-1}$. Stop when ϵ_t gets below a certain target value $\epsilon^\star$.

7 Numerical illustration of the sequential procedure

7.1 Toy model

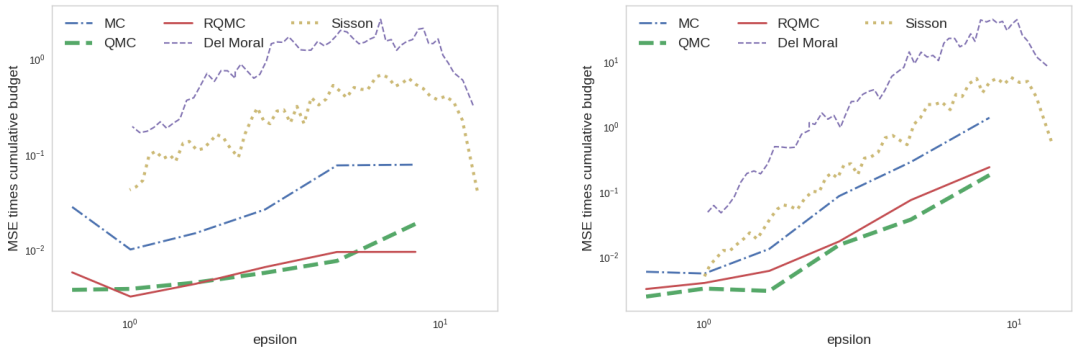
We return to the toy model of Section 5.1, taking this time $d = 3$. We compare five algorithms: three versions of Algorithm 2 with the θ_n^t's generated using, respectively, Monte Carlo, Quasi-Monte Carlo, and RQMC; the sequential ABC algorithm of Sisson et al. (2009), which (as explained previously) is essentially Algorithm 2 with a mixture proposal with N

components; and finally the algorithm of Del Moral et al. (2012). (The algorithm of Del Moral et al. (2012) generates the θ_n^t by evolving the particles resampled at the previous iteration through a Markov kernel; see the paper for more details.)

Regarding the adaptive choice of ϵ_t, we use the hybrid strategy outlined in the previous section for our MC, QMC and RQMC algorithms, we use the ESS-based strategy for Del Moral et al. (2012)'s algorithm, and we use the following strategy for Sisson et al. (2009)'s: ϵ_t is set to the median of the distances $\delta(y^\star, y_n)$ computed at the previous iteration. For all these algorithms, we set $M = 10$.

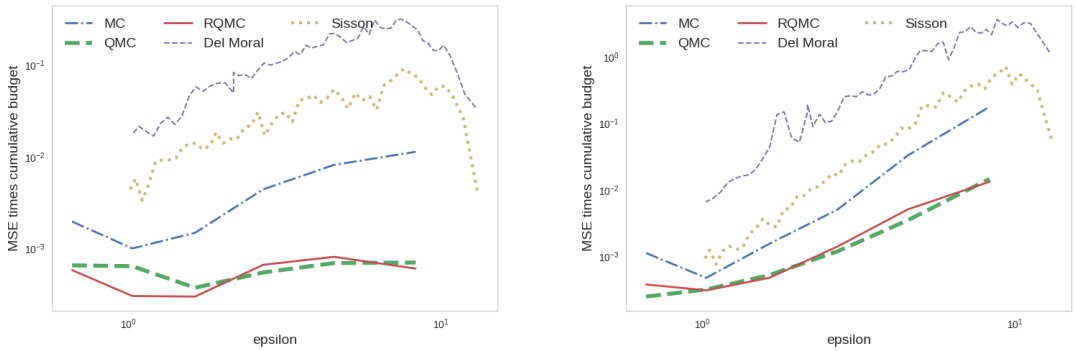
For this toy model, we simply consider the basic strategy for adapting q_t outlined in Section 6.2.1, i.e. q_t is a Gaussian fit to the previous set of particles. The five algorithms are run with either $N = 10^3$ (Figure 7) or $N = 10^4$ particles (Figure 8); in both cases the algorithms are stopped when $\epsilon_t \leq \epsilon^\star = 1$. In both figures, we plot the adjusted MSE at iteration t as a function of ϵ_t, where the adjusted MSE is the empirical MSE of a given estimate (over 50 runs) times the number of observations generated from the model up to time t. The adjusted MSE make it possible to account for the different running times of the algorithms. See also Table 2 for a direct comparison in terms of both CPU effort and MSE.

The considered estimates are the same as in Section 5.1, i.e. the ABC posterior expectation and variance of $\bar{\theta}$, the average of the components of vector θ. At least for posterior expectations, we see that the QMC and RQMC versions outperform the MC version of our algorithm, which in turn outperforms the sequential ABC algorithms of Sisson et al. (2009) and Del Moral et al. (2012).



(a) (b)

Figure 7: three-dimensional Gaussian toy example. Algorithms run with $N = 10^3$ particles. Adjusted MSE (as defined in the text) at iteration t, as function of ϵ_t, for the following posterior estimate: exceptation (left) and variance (right) of $\bar{\theta} = (\theta_1 + \theta_2 + \theta_3)/3$.



(a) (b)

Figure 8: Same as Figure 7, except algorithms are run with $N = 10^4$ particles.

Sampling method	MSE $\overline{\theta}$	MSE $\overline{\mathrm{Var}\,\theta}$	number simulated datapoints	ϵ_T
AIS-MC	0.00162	0.00037	44,980	**0.65**
AIS-QMC	**0.00039**	0.00014	**32,919**	**0.65**
AIS-RQMC	0.00049	0.00013	42,088	**0.65**
Del Moral	0.00117	0.00018	580,000	1.0
Sisson	0.00117	**0.00010**	125,928	0.95
IS-MC	0.00128	0.00513	1,000,000	0.65

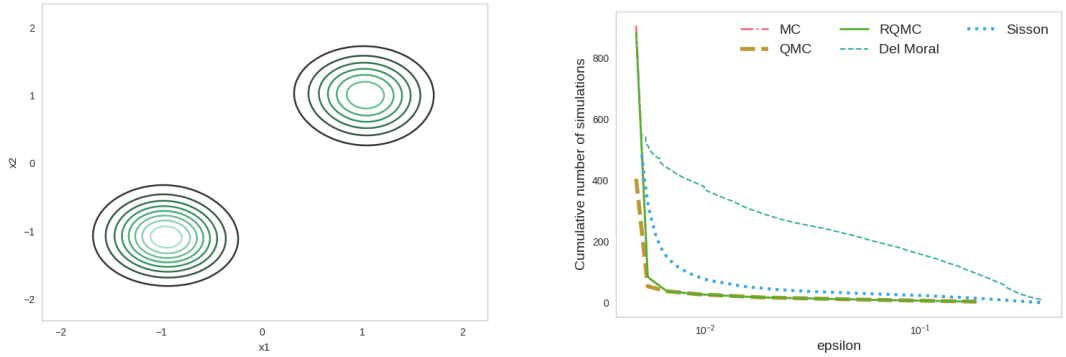
Table 2: Toy example, performance of the five considered sequential algorithms at the final iteration T, for $N = 10^3$ particles. IS-MC corresponds to the plain IS sampling without adaptation.

7.2 Bimodal Gaussian distribution

In order to illustrate the flexibility that comes from using a mixture of Gaussians for the proposal we consider a model that yields a multi-modal posterior:

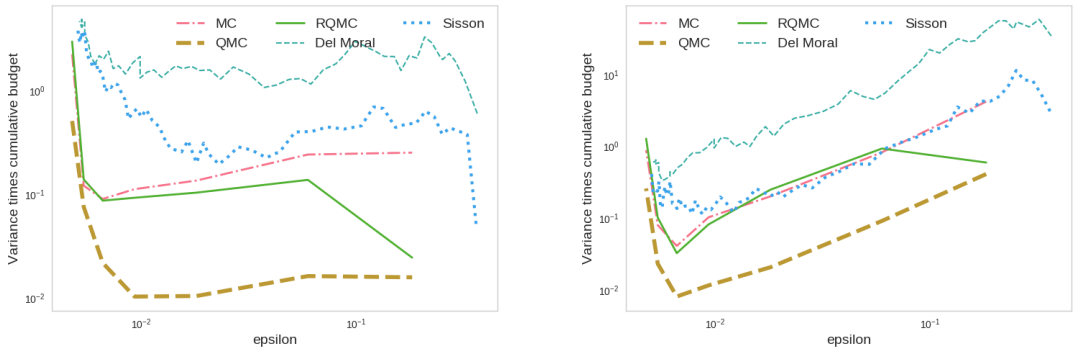
$$\begin{aligned} \theta &\sim \mathcal{U}\left([-10,10]^d\right), \\ y_i &\overset{iid}{\sim} \frac{1}{2}\mathcal{N}(\theta, I_d) + \frac{1}{2}\mathcal{N}(-\theta, I_d), \quad i = 1, \ldots, 100. \end{aligned}$$

We simulate $y^\star$ from the model. Throughout this application we set $d = 2$. The model is not identifiable and thus generates a bimodal posterior. Regarding the distance δ, we follow the idea of Bernton et al. (2017) and use the optimal transport distance between $\boldsymbol{y}$ and $\boldsymbol{y}^*$, more specifically the earth-movers-distance.



(a) (b)

Figure 9: Simulation for the bimodal distribution. Left: recovered posterior distribution. Right: average (over 50 runs) of cumulative number of simulations from the simulator across particles according to acceptance threshold; algorithms were run with $N = 10^3$ particles.



(a) (b)

Figure 10: Same plot as in Figure 7 for the bimodal example and $N = 10^3$.

We set $\epsilon^\star = 5 \times 10^{-3}$. This value has been chosen as before as a small quantile of the realized distances after 10^6 simulations from the prior and the simulator. The recovered posterior is shown in Figure 9a. Figure 9b illustrates the adaptivity in the simulation from the simulator achieved via the negative binomial approach. As the threshold becomes smaller and smaller, the number of necessary simulations start to increase severely. In the end, the number of necessary simulations of the different methods catch up with each other. Still, the approaches based on (R)QMC achieve a lower variance of the estimator as is illustrated in Figures 10a and 10b.

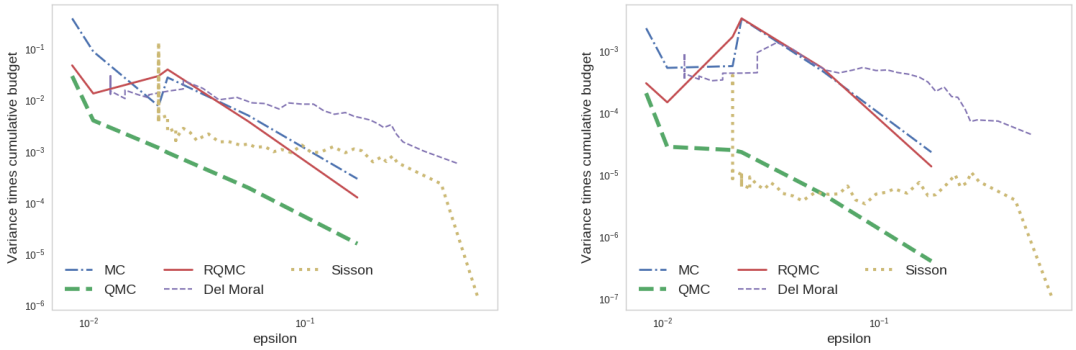
7.3 Tuberculosis mutation

We now return to the tuberculosis example presented in Section 5.3; we set the target value $\epsilon^\star = 0.01$, and restrict the CPU budget to 10^6 simulations from the model, as these simulations are computationally intensive. We see that again the QMC approach performs best in

terms of number of simulations needed and also in terms of variance times computational budget; see Figures 11a and 11b, and Table 3. The approach of Sisson et al. (2009) exceeds the total computation budget and thus does not reach the fixed threshold. Figures 11a and 11b illustrate the effect of the hybrid strategy for adapting ϵ and the number of simulations per parameter value (Section 6.3). The kink in the lines for the adaptive importance sampling approaches corresponds to the moment when the weighting is obtained via the negative binomial distribution.

Sampling method	Variance $\overline{\theta}$	Variance $\overline{\mathrm{Var}\,\theta}$	number sim. datapoints	ϵ_T
AIS-MC	0.376	5.916×10^{-6}	419,353	0.008
AIS-QMC	0.380	1.156×10^{-6}	**212,183**	**0.008**
AIS-RQMC	0.378	1.001×10^{-6}	318,196	0.008
Del Moral	**0.375**	1.065×10^{-6}	495,000	0.010
Sisson	0.393	$\mathbf{1.834 \times 10^{-7}}$	1,367,949	0.021

Table 3: Tuberculosis example, performance of the five considered sequential algorithms at the final iteration T, for $N = 500$ particles



(a) (b)

Figure 11: Same plot as in Figure 7 for the tuberculosis example and $N = 500$ particles

8 Conclusion

In this paper we introduced the use of low discrepancy sequences in approximate Bayesian computation. We found that from both a theoretical and practical perspective the use of (R)QMC in ABC can yield substantial variance reduction. However, care must be taken when using (R)QMC sequences. First, the transformation of uniform sequences to the distribution of interest must preserve the low discrepancy properties of the point set. This is of major importance for a sequential version of the ABC algorithm that is based on adaptive importance sampling. Second, the advantage of using (R)QMC tends to diminish with the dimension (of the parameter space). Fortunately, the dimension of the parameter space is often small in ABC applications. From a practical perspective we recommend to use RQMC point sets instead of QMC as these allow the assessment of the error via repeated simulation.

Another contribution of this paper is the use of the negative binomial distribution in order to adapt the CPU cost of sampling from the likelihood (for a given θ) to the considered threshold ϵ. This approach seems to reduce significantly the overall CPU cost.

Finally, If the user suspects a multimodal posterior, we recommend to estimate a mixture distribution based on the accepted samples and generate RQMC samples based on the mixture.

Acknowledgments

The research of the first author is funded by a GENES doctoral scholarship. The research of the second author is partially supported by a grant from the French National Research Agency (ANR) as part of the Investissements dAvenir program (ANR-11-LABEX-0047). We are thankful to Mathieu Gerber, two anonymous referees and the associate editor who made comments that helped us to improve the paper.

References

Agapiou, S., Papaspiliopoulos, O., Sanz-Alonso, D., and Stuart, A. M. (2015). Importance sampling: computational complexity and intrinsic dimension. *arXiv preprint 1511.06196*.

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

Barthelmé, S. and Chopin, N. (2014). Expectation propagation for likelihood-free inference. *Journal of the American Statistical Association*, 109(505):315–333.

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

Bernton, E., Jacob, P. E., Gerber, M., and Robert, C. P. (2017). Inference in generative models using the Wasserstein distance. *arXiv preprint arXiv:1701.05146*.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2016). Variational Inference: A Review for Statisticians. *arXiv preprint arXiv:1601.00670*.

Blum, M. G. (2010). Approximate Bayesian computation: A nonparametric perspective. *Journal of the American Statistical Association*, 105(491):1178–1187.

Bornn, L., Pillai, N. S., Smith, A., and Woodard, D. (2015). The use of a single pseudo-sample in approximate Bayesian computation. *Statistics and Computing*, 27(3):1–14.

Christophe, D. and Petr, S. (2015). *randtoolbox: Generating and Testing Random Numbers*. R package version 1.17.

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

Dick, J., Kuo, F. Y., and Sloan, I. H. (2013). High-dimensional integration: the quasi-Monte Carlo way. *Acta Numerica*, 22:133–288.

Fearnhead, P. and Prangle, D. (2012). Constructing summary statistics for approximate Bayesian computation: Semi-automatic approximate Bayesian computation. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 74(3):419–474.

Gerber, M. (2015). On integration methods based on scrambled nets of arbitrary size. *Journal of Complexity*, 31(6):798–816.

Gerber, M. and Chopin, N. (2015). Sequential quasi monte carlo. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 77(3):509–579.

Glasserman, P. (2013). *Monte Carlo methods in financial engineering*, volume 53. Springer Science & Business Media.

Gutmann, M. U. and Corander, J. (2016). Bayesian optimization for likelihood-free inference of simulator-based statistical models. *Journal of Machine Learning Research*, 17(125):1–47.

Hardy, G. H. (1905). On double *Fourier* series, and especially those which represent the double zeta-function with real and incommensurable parameters. *The Quarterly Journal of Pure and Applied Mathematics*, 37:53–79.

Hickernell, F. J. (2006). *Koksma–Hlawka Inequality*. American Cancer Society.

Johnson, N. L., Kemp, A. W., and Kotz, S. (2005). *Univariate discrete distributions*, volume 444. John Wiley & Sons.

Kong, A., Liu, J. S., and Wong, W. H. (1994). Sequential imputation and Bayesian missing data problems. *Journal of the American statistical association*, 89:278–288.

Kuipers, L. and Niederreiter, H. (2012). *Uniform distribution of sequences*. Courier Corporation.

L'Ecuyer, P. (2016). Randomized Quasi-Monte Carlo: An Introduction for Practitioners. In *12th International Conference on Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing (MCQMC 2016)*.

Lee, A. (2012). On the the choice of MCMC kernels for approximate Bayesian computation with SMC samplers. In *Simulation Conference (WSC), Proceedings of the 2012 Winter*, pages 1–12. IEEE.

Lee, A. and Łatuszyński, K. (2014). Variance bounding and geometric ergodicity of Markov chain Monte Carlo kernels for approximate Bayesian computation. *Biometrika*, 101(3):655–671.

Lemieux, C. (2009). *Monte Carlo and quasi-Monte Carlo sampling*. Springer Series in Statistics. Springer, New York.

Leobacher, G. and Pillichshammer, F. (2014). *Introduction to quasi-Monte Carlo integration and applications*. Springer.

Lintusaari, J., Gutmann, M. U., Dutta, R., Kaski, S., and Corander, J. (2017). Fundamentals and recent developments in approximate Bayesian computation. *Systematic biology*, 66(1):e66–e82.

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

Marin, J.-M., Raynal, L., Pudlo, P., Ribatet, M., and Robert, C. P. (2016). ABC random forests for Bayesian parameter inference. *arXiv preprint arXiv:1605.05537*.

Marjoram, P., Molitor, J., Plagnol, V., and Tavare, S. (2003). Markov chain Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences of the United States of America*, 100(26):15324–8.

Ökten, G., Tuffin, B., and Burago, V. (2006). A central limit theorem and improved error bounds for a hybrid-Monte Carlo sequence with applications in computational finance. *Journal of Complexity*, 22(4):435–458.

Owen, A. (1998). Monte Carlo extension of quasi-Monte Carlo. *1998 Winter Simulation Conference. Proceedings (Cat. No.98CH36274)*, 1(1):571–577.

Owen, A. B. (1997). Scrambled net variance for integrals of smooth functions. *The Annals of Statistics*, 25(4):1541–1562.

Owen, A. B. (2008). Local antithetic sampling with scrambled nets. *The Annals of Statistics*, 36(5):2319–2343.

Papamakarios, G. and Murray, I. (2016). Fast ε-free inference of simulation models with Bayesian conditional density estimation. In *Advances in Neural Information Processing Systems*, pages 1028–1036.

Sedki, M., Pudlo, P., Marin, J.-M., Robert, C. P., and Cornuet, J.-M. (2012). Efficient learning in ABC algorithms. *arXiv preprint 1210.1388*.

Sisson, S. A., Fan, Y., Tanaka, M. M., Rogers, A., Huang, Y., Njegic, B., Wayne, L., Gordon, M. S., Dabdub, D., Gerber, R. B., and Finlayson-pitts, B. J. (2009). Correction for Sisson et al., Sequential Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 106(39):16889–16889.

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

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

Toni, T., Welch, D., Strelkowa, N., Ipsen, A., and Stumpf, M. P. (2009). Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems. *Journal of The Royal Society Interface*, 6(31):187–202.

Wilkinson, R. D. (2014). Accelerating ABC methods using Gaussian processes. *Proceedings of the 17 th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 33:1015–1023.

9 Appendix

9.1 Proofs of main results

9.1.1 Proof of Corollary 1

For the mixed sequences we have

$$\lim_{N\to\infty}\frac{1}{N}\sum_{k=1}^{N}\sigma^2_{k,\text{qmc-mixed}} = C^2_{\text{qmc-mixed}}$$

and for the Monte Carlo estimate we have

$$\lim_{N\to\infty}\frac{1}{N}\sum_{k=1}^{N}\sigma^2_{k,\text{mc}} = C^2_{\text{mc}}$$

where

$$C^2_{\text{qmc-mixed}} = \int_{[0,1]^s} f(x)f(x)^T\mathrm{d}x - \int_{[0,1]^d}\left(\int_{[0,1]^{s-d}} f(u)\mathrm{d}X^{d+1:s}\right)\left(\int_{[0,1]^{s-d}} f(u)\mathrm{d}X^{d+1:s}\right)^T \mathrm{d}q^{1:d}$$

and

$$C^2_{\text{mc}} = \int_{[0,1]^s} f(x)f(x)^T\mathrm{d}x - \left(\int_{[0,1]^s} f(x)\mathrm{d}x\right)\left(\int_{[0,1]^s} f(x)\mathrm{d}x\right)^T.$$

We must show that

$$\left(\int_{[0,1]^s} f(x)\mathrm{d}x\right)\left(\int_{[0,1]^s} f(x)\mathrm{d}x\right)^T \preceq \int_{[0,1]^d}\left(\int_{[0,1]^{s-d}} f(u)\mathrm{d}X^{d+1:s}\right)\left(\int_{[0,1]^{s-d}} f(u)\mathrm{d}X^{d+1:s}\right)^T \mathrm{d}q^{1:d},$$

in the sense of positive definite matrices. This inequality holds in the univariate case due to the Cauchy-Schwartz inequality. In the multivariate case, let $\int_{[0,1]^{s-d}} f(u^k)\mathrm{d}X^{d+1:s} = A\left(q^{1:d}\right)$. We rewrite:

$$\int_{[0,1]^d} A\left(q^{1:d}\right)\mathrm{d}q^{1:d}\int_{[0,1]^d} A\left(q^{1:d}\right)^T\mathrm{d}q^{1:d} \preceq \int_{[0,1]^d} A\left(q^{1:d}\right)A\left(q^{1:d}\right)^T\mathrm{d}q^{1:d}.$$

In order to check the positive definiteness let $v\in\mathbb{R}^s$. We check

$$\begin{aligned}
v^T\int_{[0,1]^d} A\left(q^{1:d}\right)\mathrm{d}q^{1:d}\int_{[0,1]^d} A\left(q^{1:d}\right)^T\mathrm{d}q^{1:d}v &\leq v^T\int_{[0,1]^d} A\left(q^{1:d}\right)A\left(q^{1:d}\right)^T\mathrm{d}q^{1:d}v,\\
\int_{[0,1]^d} v^TA\left(q^{1:d}\right)\mathrm{d}q^{1:d}\int_{[0,1]^d} A\left(q^{1:d}\right)^T v\mathrm{d}q^{1:d} &\leq \int_{[0,1]^d} v^TA\left(q^{1:d}\right)A\left(q^{1:d}\right)^T v\mathrm{d}q^{1:d}.
\end{aligned}$$

While noting that $v^TA\left(q^{1:d}\right)\in\mathbb{R}$ and $A\left(q^{1:d}\right)^T v\in\mathbb{R}, \forall v\in\mathbb{R}^s$ we are back in the univariate case and the inequality holds. □

9.1.2 Proof of Theorem 3

The statement of the theorem is equivalent to $\lim_{N\to\infty} |\mathbb{P}(T_N \leq t) - \mathbb{P}(Z \leq t)| = 0$ for all $t \in \mathbb{R}^s$, $T_N = N^{1/2} S_N^{RQMC}$, and Z a random variable distributed according to the Gaussian limit.

When conditioning on the random element V in the RQMC sequence, we have that

$$\lim_{N\to\infty} \mathbb{P}(T_N \leq t | V = v) = \mathbb{P}(Z \leq t)$$

for almost all v, by Theorem 2, as a RQMC sequence is a QMC sequence with probability one. Furthermore, $|\mathbb{P}(T_N \leq t|V = v)| \leq 1$, thus the function is dominated. For all N we have

$$\begin{aligned} |\mathbb{P}(T_N \leq t) - \mathbb{P}(Z \leq t)| &= \left| \int_{\mathcal{B}} \{\mathbb{P}(T_N \leq t|V = v) - \mathbb{P}(Z \leq t)\} \, \mathrm{d}\mathbb{P}(v) \right|, \\ &\leq \int_{\mathcal{B}} |\mathbb{P}(T_N \leq t|b) - \mathbb{P}(Z \leq t)| \, \mathrm{d}\mathbb{P}(v). \end{aligned}$$

And

$$\lim_{N\to\infty} \int_{\mathcal{B}} |\mathbb{P}(T_N \leq t|V = v) - \mathbb{P}(Z \leq t)| \, d\mathbb{P}(v) = 0,$$

due to the dominated convergence theorem. Therefore

$$\lim_{N\to\infty} |\mathbb{P}(T_N \leq t) - \mathbb{P}(Z \leq t)| = 0.$$

□

9.1.3 Proof of Proposition 2

Since the θ_n's are deterministic,

$$\begin{aligned} \mathbb{E}\left[\hat{Z}_N\right] &= \frac{1}{N} \sum_{n=1}^{N} \frac{p(\theta_n)}{q(\theta_n)} \mathbb{P}_{\theta_n} \left(\delta(y, y^\star) \leq \epsilon\right) = \frac{1}{N} \sum_{n=1}^{N} f(\theta_n) \\ \mathrm{Var}[\hat{Z}_N] &= \frac{1}{MN^2} \sum_{n=1}^{N} \left\{ \frac{p(\theta_n)}{q(\theta_n)} \right\}^2 \mathbb{P}_{\theta_n} \left(\delta(y, y^\star) \leq \epsilon\right) \left\{1 - \mathbb{P}_{\theta_n} \left(\delta(y, y^\star) \leq \epsilon\right)\right\} \end{aligned}$$

and $\left|\mathbb{E}\left[\hat{Z}_N\right] - Z_\epsilon\right| = \mathcal{O}(N^{\tau-1})$ for any $\tau > 0$, by Koksma-Hlawka inequality. By the standard decomposition of the mean square error:

$$\mathbb{E}\left[(\hat{Z}_N - Z_\epsilon)^2\right] = \left(\mathbb{E}\left[\hat{Z}_N\right] - Z_\epsilon\right)^2 + \mathrm{Var}\left[\hat{Z}_N\right]$$

and since $p(\theta_n)/q(\theta_n) \leq C$, we see that that the MSE times M is $\mathcal{O}(N^{-1})$.

9.1.4 Proof of Theorem 4

One has:

$$\left(\hat{\phi}_N - \mathbb{E}_{p_\epsilon}\phi\right) = \left(\frac{\sum_{n=1}^N w_n \phi(\theta_n)}{\sum_{n=1}^N w_n} - \mathbb{E}_{p_\epsilon}\phi\right)$$
$$= \frac{N^{-1}\sum_{n=1}^N w_n \bar{\phi}(\theta_n)}{N^{-1}\sum_{n=1}^N w_n}$$

where $\bar{\phi} = \phi - \mathbb{E}_{p_\epsilon}\phi$. Since the denominator converges almost surely to Z_ϵ, and the numerator (times $N^{1/2}$) converges to a Gaussian limit (per Theorem 2), we may apply Slutsky's theorem to obtain the desired result.

More precisely, the numerator has a null expectation, and is such that

$$N^{-1/2}\sum_{n=1}^N w_n \bar{\phi}(\theta_n) \xrightarrow{\mathcal{L}} \mathcal{N}\left(0, \tau^2(\phi)\right)$$

where

$$\tau^2(\phi) = \int_\Theta \frac{p(\theta)^2}{q(\theta)} \bar{\phi}(\theta)^2 \frac{b(\theta_n)\{1 - b(\theta_n)\}}{M} \mathrm{d}\theta$$

again by direct application of Theorem 2, and using the fact that, for a fixed θ_n,

$$\mathrm{Var}_{x_n \sim q_{\theta_n}}[\hat{L}_\epsilon(x_n)] = \frac{b(\theta_n)\{1 - b(\theta_n)\}}{M}$$

with $b(\theta) = \mathbb{P}_\theta\left(\delta(y, y^\star) \leq \epsilon\right)$. □