Multilevel rejection sampling for approximate Bayesian computation

David J. Warne [a],[*], Ruth E. Baker [b], Matthew J. Simpson [a]

[a] *School of Mathematical Sciences, Queensland University of Technology, Brisbane, Australia*
[b] *Mathematical Institute, University of Oxford, Oxford, United Kingdom*

ARTICLE INFO

ABSTRACT

Likelihood-free methods, such as approximate Bayesian computation, are powerful tools for practical inference problems with intractable likelihood functions. Markov chain Monte Carlo and sequential Monte Carlo variants of approximate Bayesian computation can be effective techniques for sampling posterior distributions in an approximate Bayesian computation setting. However, without careful consideration of convergence criteria and selection of proposal kernels, such methods can lead to very biased inference or computationally inefficient sampling. In contrast, rejection sampling for approximate Bayesian computation, despite being computationally intensive, results in independent, identically distributed samples from the approximated posterior. An alternative method is proposed for the acceleration of likelihood-free Bayesian inference that applies multilevel Monte Carlo variance reduction techniques directly to rejection sampling. The resulting method retains the accuracy advantages of rejection sampling while significantly improving the computational efficiency.

© 2018 Elsevier B.V. All rights reserved.

1. Introduction

Statistical inference is of fundamental importance to all areas of science. Inference enables the testing of theoretical models against observations, and provides a rational means of quantifying uncertainty in existing models. Modern approaches to statistical inference, based on Monte Carlo sampling techniques, provide insight into many complex phenomena (Beaumont et al., 2002; Pooley et al., 2015; Ross et al., 2017; Stumpf, 2014; Sunnåker et al., 2013; Tavaré et al., 1997; Thorne and Stumpf, 2012; Vo et al., 2015).

Suppose we have: a set of observations, $\mathcal{D}$; a method of determining the likelihood of these observations, $\mathcal{L}(\boldsymbol{\theta}; \mathcal{D})$, under the assumption of some model characterised by parameter vector $\boldsymbol{\theta} \in \Theta$; and a prior probability density, $p(\boldsymbol{\theta})$. The posterior probability density, $p(\boldsymbol{\theta} \mid \mathcal{D})$, can be computed using Bayes' Theorem,

$$p(\boldsymbol{\theta} \mid \mathcal{D}) = \frac{\mathcal{L}(\boldsymbol{\theta}; \mathcal{D})p(\boldsymbol{\theta})}{\int_{\Theta} \mathcal{L}(\boldsymbol{\theta}; \mathcal{D})p(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}}. \tag{1}$$

---

* Correspondence to: 2 George St, Brisbane, Queensland, 4000, Australia.
E-mail address: david.warne@qut.edu.au (D.J. Warne).

Explicit expressions for likelihood functions are rarely available (Tavaré et al., 1997; Warne et al., 2017; Wilkinson, 2009); motivating the development of likelihood-free methods, such as approximate Bayesian computation (ABC) (Stumpf, 2014; Sunnåker et al., 2013). ABC methods approximate the likelihood through evaluating the discrepancy between data generated by a simulation of the model of interest and the observations, yielding an approximate posterior,

$$p(\boldsymbol{\theta} \mid d(\mathcal{D}, \mathcal{D}_s) < \epsilon) \propto \mathbb{P}(d(\mathcal{D}, \mathcal{D}_s) < \epsilon \mid \boldsymbol{\theta})p(\boldsymbol{\theta}). \tag{2}$$

Here, $\mathcal{D}_s \sim f(\mathcal{D} \mid \boldsymbol{\theta})$ is data generated by the model simulation process, $f(\mathcal{D} \mid \boldsymbol{\theta})$, d is a discrepancy metric, and $\epsilon > 0$ is the acceptance threshold. Due to this approximation, Monte Carlo estimators based on Eq. (2) are biased (Barber et al., 2015). In spite of this bias, however, ABC methods have proven to be very powerful tools for practical inference applications in many scientific areas, including evolutionary biology (Beaumont et al., 2002; Tavaré et al., 1997; Thorne and Stumpf, 2012), ecology (Stumpf, 2014), cell biology (Ross et al., 2017; Johnston et al., 2014; Vo et al., 2015) and systems biology (Wilkinson, 2009).

1.1. Sampling algorithms for ABC

The most elementary implementation of ABC is ABC rejection sampling (Pritchard et al., 1999; Sunnåker et al., 2013), see Algorithm 1. This method generates N independent and identically distributed samples $\boldsymbol{\theta}^1, \ldots, \boldsymbol{\theta}^N$ from the posterior distribution by accepting proposals, $\boldsymbol{\theta}^* \sim p(\boldsymbol{\theta})$, when the data generated by the model simulation process $f(\mathcal{D} \mid \boldsymbol{\theta}^*)$ is within ϵ of the observed data, $\mathcal{D}$, under the discrepancy metric $d(\mathcal{D}, \cdot)$. ABC rejection sampling is simple to implement, and samples are independent and identically distributed. Therefore ABC rejection sampling is widely used in many applications (Browning et al., 2018; Grelaud et al., 2009; Navascués et al., 2017; Ross et al., 2017; Vo et al., 2015). However, ABC rejection sampling can be computationally prohibitive in practice (Barber et al., 2015; Fearnhead and Prangle, 2012). This is especially true when the prior density is highly diffuse compared with the target posterior density (Marin et al., 2012), as most proposals are rejected.

Algorithm 1 ABC rejection sampler

1: **for** $i = 1, \ldots, N$ **do**
2:     **repeat**
3:         Sample prior, $\boldsymbol{\theta}^* \sim p(\boldsymbol{\theta})$.
4:         Generate data, $\mathcal{D}_s \sim f(\mathcal{D} \mid \boldsymbol{\theta}^*)$.
5:     **until** $d(\mathcal{D}, \mathcal{D}_s) \leq \epsilon$
6:     Set $\boldsymbol{\theta}^i \leftarrow \boldsymbol{\theta}^*$.
7: **end for**

To improve the efficiency of ABC rejection sampling, one can consider a likelihood-free modification of Markov chain Monte Carlo (MCMC) (Beaumont et al., 2002; Marjoram et al., 2003; Tanaka et al., 2006) in which a Markov chain is constructed with a stationary distribution identical to the desired posterior. Given the Markov chain is in state $\boldsymbol{\theta}^i$, a state transition is proposed via a proposal kernel, $K(\boldsymbol{\theta} \mid \boldsymbol{\theta}^i)$.

The Metropolis–Hastings (Hastings, 1970; Metropolis et al., 1953) state transition probability, h, can be modified within an ABC framework to yield

$$h = \begin{cases} \min\left(\dfrac{p(\boldsymbol{\theta}^*)K(\boldsymbol{\theta}^i \mid \boldsymbol{\theta}^*)}{p(\boldsymbol{\theta}^i)K(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}^i)}, 1\right) & \text{if } d(\mathcal{D}, \mathcal{D}_s) \leq \epsilon, \\ 0 & \text{otherwise.} \end{cases}$$

The stationary distribution of such a Markov chain is the desired approximate posterior (Marjoram et al., 2003). Algorithm 2 provides a method for computing N_T iterations of this Markov chain.

While MCMC-ABC sampling can be highly efficient, the samples in the sequence, $\boldsymbol{\theta}^1, \ldots, \boldsymbol{\theta}^{N_T}$, are not independent. This can be problematic as it is possible for the Markov chain to take long excursions into regions of low posterior probability. This incurs additional bias that is potentially significant (Sisson et al., 2007). A poor choice of proposal kernel can also have considerable impact upon the efficiency of MCMC-ABC (Green et al., 2015). The question of how to choose the proposal kernel is non-trivial. Typically proposal kernels are determined heuristically. However, automatic and adaptive schemes are available to assist in obtaining near optimal proposals in some cases (Cabras et al., 2015; Roberts and Rosenthal, 2009). Another additional complication is that of determining when the Markov Chain has converged; this is a challenging problem to solve in practice (Roberts and Rosenthal, 2004).

Sequential Monte Carlo (SMC) sampling was introduced to address these potential inefficiencies (Del Moral et al., 2006) and later extended within an ABC context (Sisson et al., 2007; Drovandi and Pettitt, 2011; Toni et al., 2009). A set of samples, referred to as particles, is evolved through a sequence of ABC posteriors defined through a sequence of T acceptance thresholds, $\epsilon_1, \ldots, \epsilon_T$ (Sisson et al., 2007; Beaumont et al., 2009). At each step, $t \in [0, T]$, the current ABC posterior, $p(\boldsymbol{\theta} \mid d(\mathcal{D}, \mathcal{D}_s) < \epsilon_t)$, is approximated by a discrete distribution constructed from a set of N_P particles $\boldsymbol{\theta}_t^1, \ldots, \boldsymbol{\theta}_t^{N_P}$ with

---

Algorithm 2 MCMC-ABC

1: Given initial sample $\boldsymbol{\theta}^1 \sim p(\boldsymbol{\theta} \mid d(\mathcal{D}, \mathcal{D}_s) < \epsilon)$.
2: **for** $i = 2, \ldots, N_T$ **do**
3:    Sample transition kernel, $\boldsymbol{\theta}^* \sim K(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{i-1})$.
4:    Generate data, $\mathcal{D}_s \sim f(\mathcal{D} \mid \boldsymbol{\theta}^*)$.
5:    **if** $d(\mathcal{D}, \mathcal{D}_s) \leq \epsilon$ **then**
6:       Set $h \leftarrow \min \left(p(\boldsymbol{\theta}^*)K(\boldsymbol{\theta}^{i-1} \mid \boldsymbol{\theta}^*)/p(\boldsymbol{\theta}^{i-1})K(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}^{i-1}), 1 \right)$.
7:       Sample uniform distribution, $u \sim \mathcal{U}(0, 1)$.
8:       **if** $u \leq h$ **then**
9:          Set $\boldsymbol{\theta}^i \leftarrow \boldsymbol{\theta}^*$.
10:       **else**
11:          Set $\boldsymbol{\theta}^i \leftarrow \boldsymbol{\theta}^{i-1}$.
12:       **end if**
13:    **else**
14:       Set $\boldsymbol{\theta}^i \leftarrow \boldsymbol{\theta}^{i-1}$.
15:    **end if**
16: **end for**

---

importance weights $W_t^1, \ldots, W_t^{N_P}$ that is, $\mathbb{P}(\boldsymbol{\theta} = \boldsymbol{\theta}_t^i) = W_t^i$ for $i = 1, \ldots, N_P$. The collection is updated to represent the ABC posterior of the next step, $p(\boldsymbol{\theta} \mid d(\mathcal{D}, \mathcal{D}_s) < \epsilon_{t+1})$, through application of rejection sampling on particles perturbed by a proposal kernel, $K(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{i-1})$. If $p(\boldsymbol{\theta} \mid d(\mathcal{D}, \mathcal{D}_s) < \epsilon_t)$ is similar to $p(\boldsymbol{\theta} \mid d(\mathcal{D}, \mathcal{D}_s) < \epsilon_{t+1})$, the acceptance rate should be high. The importance weights of the new family of particles are updated using an optimal backwards kernel (Sisson et al., 2007; Beaumont et al., 2009). Algorithm 3 outlines the process.

Algorithm 3 SMC-ABC

1: Initialise $\boldsymbol{\theta}_1^i \sim p(\boldsymbol{\theta})$ and $W_1^i = 1/N_P$, for $i = 1, \ldots, N_P$.
2: **for** $t = 2, \ldots, T$ **do**
3:    **for** $i = 1, \ldots, N_P$ **do**
4:       **repeat**
5:          Set $\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta}_{t-1}^j$ with probability W_{t-1}^j.
6:          Sample transition kernel, $\boldsymbol{\theta}^{**} \sim K(\boldsymbol{\theta} \mid \boldsymbol{\theta}^*)$.
7:          Generate data, $\mathcal{D}_s \sim f(\mathcal{D} \mid \boldsymbol{\theta}^{**})$.
8:       **until** $d(\mathcal{D}, \mathcal{D}_s) \leq \epsilon_t$
9:       Set $\boldsymbol{\theta}_t^i \leftarrow \boldsymbol{\theta}^{**}$.
10:       Set $W_t^i \leftarrow p(\boldsymbol{\theta}_t^i)/\sum_{j=1}^{N_P} W_{t-1}^j K(\boldsymbol{\theta}_t^i \mid \boldsymbol{\theta}_{t-1}^j)$.
11:    **end for**
12:    Normalise weights so that $\sum_{i=1}^{N_P} W_t^i = 1$.
13: **end for**

Through the use of independent weighted particles, SMC-ABC avoids long excursions into the distribution tails that are possible with MCMC-ABC. However, SMC-based methods can be affected by particle degeneracy, and the efficiency of each step is still dependent on the choice of proposal kernel (Green et al., 2015; Filippi et al., 2013). Just as with MCMC-ABC, adaptive schemes are available to guide proposal kernel selection (Beaumont et al., 2009; Del Moral et al., 2012). The choice of the sequence of acceptance thresholds is also important for the efficiency of SMC-ABC. However, there are good solutions to generate these sequences in an adaptive manner (Drovandi and Pettitt, 2011; Silk et al., 2013).

1.2. Multilevel Monte Carlo

Multilevel Monte Carlo (MLMC) is a recent development that can significantly reduce the computational burden in the estimation of expectations (Giles, 2008). To demonstrate the basic idea of MLMC, consider computing the expectation of a continuous-time stochastic process X_t at time T. Let Z_t^τ denote a discrete-time approximation to X_t with time step τ: the expectations of X_T and Z_T^τ are related according to

$$\mathbb{E}[X_T] = \mathbb{E}\left[Z_T^\tau\right] + \mathbb{E}\left[X_T - Z_T^\tau\right].$$

That is, an estimate of $\mathbb{E}\left[Z_T^\tau\right]$ can be treated as a biased estimate of $\mathbb{E}[X_T]$. By taking a sequence of time steps $\tau_1 > \cdots > \tau_L$, the indices of which are referred to as *levels*, we can arrive at a telescoping sum,

$$\mathbb{E}\left[Z_T^{\tau_L}\right] = \mathbb{E}\left[Z_T^{\tau_1}\right] + \sum_{\ell=2}^{L} \mathbb{E}\left[Z_T^{\tau_\ell} - Z_T^{\tau_{\ell-1}}\right]. \tag{3}$$

Computing this form of the expectation returns the same bias as that returned when computing $\mathbb{E}\left[Z_T^{\tau_L}\right]$ directly. However, Giles (2008) demonstrates that a Monte Carlo estimator for the telescoping sum can be computed more efficiently than directly estimating $\mathbb{E}\left[Z_T^{\tau_L}\right]$ in the context of stochastic differential equations (SDEs). This efficiency comes from exploiting the fact that the bias correction terms, $\mathbb{E}\left[Z_T^{\tau_\ell} - Z_T^{\tau_{\ell-1}}\right]$, measure the expected difference between the estimates on levels ℓ and $\ell - 1$. Therefore, sample paths of $Z_T^{\tau_{\ell-1}}$ need not be independent of sample paths of $Z_T^{\tau_\ell}$. In the case of SDEs, samples may be generated in pairs driven by the same underlying Brownian motion, that is, the pair is coupled. By the strong convergence properties of numerical schemes for SDEs, Giles (2008) shows that this coupling is sufficient to reduce the variance of the Monte Carlo estimator. This reduction in variance is achieved through optimally trading off statistical error and computational cost across all levels. Through this trade-off, an estimator is obtained with the same accuracy in *mean-square* to standard Monte Carlo, but at a reduced computational cost. This saving of computational cost is achieved since fewer samples of the most accurate discretisation are required.

1.3. Related work

Recently, several examples of MLMC applications to Bayesian inference problems have appeared in the literature. One of the biggest challenges in the application of MLMC to inverse problems is the introduction of a strong coupling between levels. That is, the construction of a coupling mechanism that reduces the variances of the bias correction terms enough to enable the MLMC estimator to be computed more efficiently than standard Monte Carlo. Dodwell et al. (2015) demonstrate a MLMC scheme for MCMC sampling applicable to high-dimensional Bayesian inverse problems with closed-form likelihood expressions. The coupling of Dodwell et al. (2015) is based on correlating Markov chains defined on a hierarchy in parameter space. A similar approach is also employed by Efendiev et al. (2015). A multilevel method for ensemble particle filtering is proposed by Gregory et al. (2016) that employs an optimal transport problem to correlate a sequence of particle filters of different levels of accuracy. Due to the computational cost of the transport problem, a local approximation scheme is introduced for multivariate parameters (Gregory et al., 2016). Beskos et al. (2017) look more generally at the case of applying MLMC methods when independent and identically distributed sampling of the distributions on some levels is infeasible, the result is an extension of MLMC in which coupling is replaced with sequential importance sampling, that is, a multilevel variant of SMC (MLSMC).

MLMC has also recently been considered in an ABC context. Guha and Tan (2017) extend the work of Efendiev et al. (2015) by replacing the Metropolis–Hasting acceptance probability in a similar way to the MCMC-ABC method (Marjoram et al., 2003). The MLSMC method (Beskos et al., 2017) is exploited to achieve coupling in an ABC context by Jasra et al. (2017).

1.4. Aims and contribution

ABC samplers based on MCMC and SMC are generally more computationally efficient than ABC rejection sampling (Marjoram et al., 2003; Sisson et al., 2007; Toni et al., 2009). However, there are many advantages to using ABC rejection sampling. Specific advantages are: (i) its simple implementation; (ii) it produces truly independent and identically distributed samples, that is, there is no need to re-weight samples; and (iii) there are no algorithm parameters that affect the computational efficiency. In particular, no proposal kernels need be heuristically defined for ABC rejection sampling.

The aim of this work is to design a new algorithm for ABC inference that retains the aforementioned advantages of ABC rejection sampling, while still being efficient and accurate. The aim is not to develop a method that is always superior to MCMC and SMC methods, but rather provide a method that requires little user-defined configuration, and is still computationally reasonable. To this end, we investigate MLMC methods applied directly to ABC rejection sampling.

Various applications of MLMC to ABC inference have been demonstrated very recently in the literature (Guha and Tan, 2017; Jasra et al., 2017), with each implementation contributing different ideas for constructing effective control variates in an ABC context. It is not yet clear when one approach will be superior to another. Thus the question of how best to apply MLMC in an ABC context remains an open problem. Since there are no examples of an application of MLMC methods to the most fundamental of ABC samplers, that is rejection sampling, such an application is a significant contribution to this field.

We describe a new algorithm for ABC rejection sampling, based on MLMC. Our new algorithm, called MLMC-ABC, is as general as standard ABC rejection sampling but is more computationally efficient through the use of variance reduction techniques that employ a novel construction for the coupling problem. We describe the algorithm, its implementation, and validate the method using a tractable Bayesian inverse problem. We also compare the performance of the new method against MCMC-ABC and SMC-ABC using a standard benchmark problem from epidemiology.

Our method benefits from the simplicity of ABC rejection sampling. We require only the discrepancy metric and a sequence of acceptance thresholds to be defined for a given inverse problem. Our approach is also efficient, and achieves comparable or superior performance to MCMC-ABC and SMC-ABC methods, at least for the examples considered in this paper. Therefore, we demonstrate that our algorithm is a promising method that could be extended to design viable alternatives to current state-of-the-art approaches. This work, along with that of Guha and Tan (2017) and Jasra et al. (2017), provides an additional set of computational tools to further enhance the utility of ABC methods in practice.

2. Methods

In this section, we demonstrate our application of MLMC ideas to the likelihood-free inference problem given in Eq. (2). The initial aim is to compute an accurate approximation to the joint posterior cumulative distribution function (CDF) of $\boldsymbol{\theta}$, using as few data generation steps as possible. We define a data generation step to be a simulation of the model of interest given a proposed parameter vector. While the initial presentation is given in the context of estimating the joint CDF, the method naturally extends to expectations of other functions (see Appendix). However, the joint CDF is considered first for clarity in the introduction of the MLMC coupling strategy.

We apply MLMC methods to likelihood-free Bayesian inference by reposing the problem of computing the posterior CDF as an expectation calculation. This allows the MLMC telescoping sum idea, as in Eq. (3), to be applied. In this context, the levels of the MLMC estimator are parameterised by a sequence of L acceptance thresholds $\epsilon_1, \ldots, \epsilon_L$ with $\epsilon_\ell > \epsilon_{\ell+1}$ for all $\ell \in [1, L]$. The efficiency of MLMC relies upon computing the terms of the telescoping sum with low variance. Variance reduction is achieved through exploiting features of the telescoping sum for CDF approximation, and further computational gains are achieved by using properties of nested ABC rejection samplers.

2.1. ABC posterior as an expectation

We first reformulate the Bayesian inference problem (Eq. (1)) as an expectation. To this end, note that, given a k-dimensional parameter space, Θ, and the random parameter vector, $\boldsymbol{\theta}$, if the joint posterior CDF, $F(\mathbf{s}) = \mathbb{P}(\theta_1 \leq s_1, \ldots, \theta_k \leq s_k)$ is differentiable, that is, its probability density function (PDF) exists, then

$$\mathbb{P}(\theta_1 \leq s_1, \ldots, \theta_k \leq s_k) = \int_{-\infty}^{s_1} \cdots \int_{-\infty}^{s_k} p(\boldsymbol{\theta} \mid \mathcal{D}) \, \mathrm{d}\theta_k \ldots \mathrm{d}\theta_1$$

$$= \int_{A_{\mathbf{s}}} p(\boldsymbol{\theta} \mid \mathcal{D}) \, \mathrm{d}\theta_k \ldots \mathrm{d}\theta_1,$$

where $A_{\mathbf{s}} = \{\boldsymbol{\theta} \in \Theta : \theta_1 \leq s_1, \ldots, \theta_k \leq s_k\}$. This can be expressed as an expectation by noting

$$\int_{A_{\mathbf{s}}} p(\boldsymbol{\theta} \mid \mathcal{D}) \, \mathrm{d}\theta_k \ldots \mathrm{d}\theta_1 = \int_{\Theta} \mathbb{1}_{A_{\mathbf{s}}}(\boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \mathcal{D}) \, \mathrm{d}\theta_k \ldots \mathrm{d}\theta_1$$

$$= \mathbb{E}\left[\mathbb{1}_{A_{\mathbf{s}}}(\boldsymbol{\theta}) \mid \mathcal{D}\right],$$

where $\mathbb{1}_{A_{\mathbf{s}}}(\boldsymbol{\theta})$ is the indicator function: $\mathbb{1}_{A_{\mathbf{s}}}(\boldsymbol{\theta}) = 1$ whenever $\boldsymbol{\theta} \in A_{\mathbf{s}}$ and $\mathbb{1}_{A_{\mathbf{s}}}(\boldsymbol{\theta}) = 0$ otherwise. Now, consider the ABC approximation given in Eq. (2) with discrepancy metric $d(\mathcal{D}, \mathcal{D}_s)$ and acceptance threshold ϵ. The ABC posterior CDF, denoted by $F_\epsilon(\mathbf{s})$, will be

$$F_\epsilon(\mathbf{s}) = \int_{A_{\mathbf{s}}} p(\boldsymbol{\theta} \mid d(\mathcal{D}, \mathcal{D}_s) < \epsilon) \, \mathrm{d}\theta_k \ldots \mathrm{d}\theta_1$$

$$= \mathbb{E}\left[\mathbb{1}_{A_{\mathbf{s}}}(\boldsymbol{\theta}) \mid d(\mathcal{D}, \mathcal{D}_s) < \epsilon\right]. \tag{4}$$

The marginal ABC posterior CDFs, $F_{\epsilon,1}(s), \ldots, F_{\epsilon,k}(s)$, are

$$F_{\epsilon,j}(s) = \lim_{s_{i \neq j} \to \infty} F_\epsilon(\mathbf{s}).$$

2.2. Multilevel estimator formulation

We now introduce some notation that will simplify further derivations. We define $\boldsymbol{\theta}_\epsilon$ to be a random vector distributed according to the ABC posterior CDF, $F_\epsilon(\mathbf{s})$, with acceptance threshold, ϵ, as given in Eq. (4). This provides us with simplification in notation for the ABC posterior PDF, $p(\boldsymbol{\theta}_\epsilon) = p(\boldsymbol{\theta} \mid d(\mathcal{D}, \mathcal{D}_s) < \epsilon)$, and the conditional expectation, $\mathbb{E}\left[\mathbb{1}_{A_{\mathbf{s}}}(\boldsymbol{\theta}_\epsilon)\right] = \mathbb{E}\left[\mathbb{1}_{A_{\mathbf{s}}}(\boldsymbol{\theta}) \mid d(\mathcal{D}, \mathcal{D}_s) < \epsilon\right]$. For any expectation, P, we use $\hat{P}^N$ to denote the Monte Carlo estimate of this expectation using N samples.

The standard Monte Carlo integration approach is to generate N samples $\boldsymbol{\theta}_\epsilon^1, \ldots, \boldsymbol{\theta}_\epsilon^N$ from the ABC posterior, $p(\boldsymbol{\theta}_\epsilon)$, then evaluate the empirical CDF (eCDF),

$$\hat{F}_\epsilon^N(\mathbf{s}) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_{A_{\mathbf{s}}}(\boldsymbol{\theta}_\epsilon^i), \tag{5}$$

for $\mathbf{s} \in \mathcal{S}$, where $\mathcal{S}$ is a discretisation of the parameter space Θ. For simplicity, we will consider $\mathcal{S}$ to be a k-dimensional regular lattice. In general, a regular lattice will not be well suited for high dimensional problems. However, since this is the first time that this MLMC approach has been presented, it is most natural to begin the exposition with a regular lattice, and then discuss other more computationally efficient approaches later. More attention is given to other possibilities in Section 4.

The eCDF is not, however, the only Monte Carlo approximation to the CDF one may consider. In particular, Giles et al., (2015) demonstrate the application of MLMC to a univariate CDF approximation. We now present a multivariate equivalent of the MLMC CDF of Giles et al. (2015) in the context of ABC posterior CDF estimation. Consider a sequence of L acceptance thresholds, $\{\epsilon_\ell\}_{\ell=1}^{\ell=L}$, that is strictly decreasing, that is, $\epsilon_\ell > \epsilon_{\ell+1}$. In this work, the problem of constructing optimal sequences is not considered, as the focus is the initial development of the method. More discussion around this problem is given in Section 4. Given such a sequence, $\{\epsilon_\ell\}_{\ell=1}^{\ell=L}$, we can represent the CDF (Eq. (4)) using the telescoping sum

$$F_{\epsilon_L}(\mathbf{s}) = \mathbb{E}\left[\mathbb{1}_{A_\mathbf{s}}(\boldsymbol{\theta}_{\epsilon_L})\right] = \sum_{\ell=1}^{L} Y_\ell(\mathbf{s}), \tag{6}$$

where

$$Y_\ell(\mathbf{s}) = \begin{cases} \mathbb{E}\left[\mathbb{1}_{A_\mathbf{s}}(\boldsymbol{\theta}_{\epsilon_1})\right] & \text{if } \ell = 1, \\ \mathbb{E}\left[\mathbb{1}_{A_\mathbf{s}}(\boldsymbol{\theta}_{\epsilon_\ell}) - \mathbb{1}_{A_\mathbf{s}}(\boldsymbol{\theta}_{\epsilon_{\ell-1}})\right] & \text{if } \ell > 1. \end{cases} \tag{7}$$

Using our notation, the MLMC estimator for Eqs. (6) and (7) is given by

$$\hat{F}_{\epsilon_L}^{N_1,\ldots,N_L}(\mathbf{s}) = \sum_{\ell=1}^{L} \hat{Y}_\ell^{N_\ell}(\mathbf{s}), \tag{8}$$

where

$$\hat{Y}_\ell^{N_\ell}(\mathbf{s}) = \begin{cases} \dfrac{1}{N_1} \sum_{i=1}^{N_1} g_\mathbf{s}(\boldsymbol{\theta}_{\epsilon_1}^i) & \text{if } \ell = 1, \\ \dfrac{1}{N_\ell} \sum_{i=1}^{N_\ell} \left[g_\mathbf{s}(\boldsymbol{\theta}_{\epsilon_\ell}^i) - g_\mathbf{s}(\boldsymbol{\theta}_{\epsilon_{\ell-1}}^i)\right] & \text{if } \ell > 1, \end{cases} \tag{9}$$

and $g_\mathbf{s}(\boldsymbol{\theta})$ is a Lipschitz continuous approximation to the indicator function; this approximation is computed using a tensor product of cubic polynomials,

$$g_\mathbf{s}(\boldsymbol{\theta}) = \prod_{j=1}^{k} \xi\left(\frac{s_j - \theta_j}{\delta_j}\right),$$

where δ_j is the lattice spacing in the jth dimension, and $\xi(x)$ is a piece-wise continuous polynomial,

$$\xi(x) = \begin{cases} 1 & x \le -1, \\ \dfrac{5}{8}x^3 - \dfrac{9}{8}x + \dfrac{1}{2} & -1 < x < 1, \\ 0 & x \ge 1. \end{cases}$$

This expression is based on the rigorous treatment of smoothing required in the univariate case given by Giles et al. (2015). While other polynomials that satisfy certain conditions are possible (Giles et al., 2015; Reiss, 1981), here we restrict ourselves to this relatively simple form. Application of the smoothing function improves the quality of the final CDF estimate, just as using smoothing kernels improves the quality of PDF estimators (Silverman, 1986). Such a smoothing is also necessary to avoid convergence issues with MLMC caused by the discontinuity of the indicator function (Avikainen, 2009; Giles et al., 2015).

To compute the $\hat{Y}_1^{N_1}(\mathbf{s})$ term (Eq. (9)), we generate N_1 samples $\boldsymbol{\theta}_{\epsilon_1}^1, \ldots, \boldsymbol{\theta}_{\epsilon_1}^{N_1}$ from $p(\boldsymbol{\theta}_{\epsilon_1})$; this represents a biased estimate for $F_{\epsilon_L}(\mathbf{s})$. To compensate for this bias, correction terms, $\hat{Y}_\ell^{N_\ell}(\mathbf{s})$, are evaluated for $\ell > 2$, each requiring the generation of N_ℓ samples $\boldsymbol{\theta}_{\epsilon_\ell}^1, \ldots, \boldsymbol{\theta}_{\epsilon_\ell}^{N_\ell}$ from $p(\boldsymbol{\theta}_{\epsilon_\ell})$ and N_ℓ samples $\boldsymbol{\theta}_{\epsilon_{\ell-1}}^1, \ldots, \boldsymbol{\theta}_{\epsilon_{\ell-1}}^{N_\ell}$ from $p(\boldsymbol{\theta}_{\epsilon_{\ell-1}})$, as given in Eq. (9). It is important to note that the samples, $\boldsymbol{\theta}_{\epsilon_{\ell-1}}^1, \ldots, \boldsymbol{\theta}_{\epsilon_{\ell-1}}^{N_\ell}$, used to compute $\hat{Y}_\ell^{N_\ell}(\mathbf{s})$ are independent of the samples, $\boldsymbol{\theta}_{\epsilon_{\ell-1}}^1, \ldots, \boldsymbol{\theta}_{\epsilon_{\ell-1}}^{N_{\ell-1}}$, used to compute $\hat{Y}_{\ell-1}^{N_{\ell-1}}(\mathbf{s})$.

2.3. Variance reduction

The goal is to introduce a coupling between levels that controls the variance of the bias correction terms. With an effective coupling, the result is an estimator with lower variance, hence the number of samples required to obtain an accurate estimate is reduced. Denote v_ℓ as the variance of the estimator $\hat{Y}_\ell^{N_\ell}(\mathbf{s})$. For $\ell \ge 2$ this can be expressed as

$$\begin{aligned} v_\ell &= \mathbb{V}\left[g_\mathbf{s}(\boldsymbol{\theta}_{\epsilon_\ell}) - g_\mathbf{s}(\boldsymbol{\theta}_{\epsilon_{\ell-1}})\right] \\ &= \mathbb{V}\left[g_\mathbf{s}(\boldsymbol{\theta}_{\epsilon_\ell})\right] + \mathbb{V}\left[g_\mathbf{s}(\boldsymbol{\theta}_{\epsilon_{\ell-1}})\right] - 2 \cdot \mathbb{C}\left[g_\mathbf{s}(\boldsymbol{\theta}_{\epsilon_\ell}), g_\mathbf{s}(\boldsymbol{\theta}_{\epsilon_{\ell-1}})\right], \end{aligned}$$

where $\mathbb{V}[\cdot]$ and $\mathbb{C}[\cdot, \cdot]$ denote the variance and covariance, respectively. Introducing a positive correlation between the random variables $\boldsymbol{\theta}_{\epsilon_\ell}$ and $\boldsymbol{\theta}_{\epsilon_{\ell-1}}$ will have the desired effect of reducing the variance of $\hat{Y}_\ell^{N_\ell}(\mathbf{s})$.

In many applications of MLMC, a positive correlation is introduced through driving samplers at both the ℓ and $\ell - 1$ level with the same *randomness*. Properties of Brownian motion or Poisson processes are typically used for the estimation of expectations involving SDEs or Markov processes (Giles, 2008; Anderson and Higham, 2012; Lester et al., 2016). In the context of ABC methods, however, simulation of the quantity of interest is necessarily based on rejection sampling. The reliance on rejection sampling makes a strong coupling, in the true sense of MLMC, a difficult, if not impossible task. Rather, here we introduce a weaker form of coupling through exploiting the fact that our MLMC estimator is performing the task of computing an estimate of the ABC posterior CDF. We combine this with a property of nested ABC rejection samplers to arrive at an efficient algorithm for computing $\hat{F}_{\epsilon_L}(\mathbf{s})$.

We proceed to establish a correlation between levels as follows. Assume we have computed, for some $\ell < L$, the terms $\hat{Y}_1^{N_1}(\mathbf{s}), \ldots, \hat{Y}_\ell^{N_\ell}(\mathbf{s})$ in Eq. (8). That is, we have an estimator to the CDF at level ℓ by taking the sum

$$\hat{F}_{\epsilon_\ell}^{N_1, \ldots, N_\ell}(\mathbf{s}) = \sum_{m=1}^{\ell} \hat{Y}_m^{N_m}(\mathbf{s}),$$

with marginal distributions $\hat{F}_{\epsilon_\ell, j}^{N_1, \ldots, N_\ell}(s_j)$ for $j = 1, \ldots, k$. We can use this to determine a coupling based on matching marginal probabilities when computing $\hat{Y}_{\ell+1}(\mathbf{s})$. After generating $N_{\ell+1}$ samples $\boldsymbol{\theta}_{\epsilon_{\ell+1}}^1, \ldots, \boldsymbol{\theta}_{\epsilon_{\ell+1}}^{N_{\ell+1}}$ from $p(\boldsymbol{\theta}_{\epsilon_{\ell+1}})$, we compute the eCDF, $\hat{F}_{\epsilon_{\ell+1}}^{N_{\ell+1}}(\mathbf{s})$, using Eq. (5) and obtain the marginal distributions, $\hat{F}_{\epsilon_{\ell+1}, j}^{N_{\ell+1}}(s_j)$ for $j = 1, \ldots, k$. We can thus generate $N_{\ell+1}$ coupled pairs $\{\boldsymbol{\theta}_{\epsilon_{\ell+1}}^i, \boldsymbol{\theta}_{\epsilon_\ell}^i\}$ by choosing the $\boldsymbol{\theta}_{\epsilon_\ell}^i$ with the same marginal probabilities as the empirical probability of $\boldsymbol{\theta}_{\epsilon_{\ell+1}}^i$. That is, the jth component of $\boldsymbol{\theta}_{\epsilon_\ell}^i$ is given by

$$\theta_{\epsilon_\ell, j}^i = \hat{G}_{\epsilon_\ell}^{N_1, \ldots, N_\ell} \left(\hat{F}_{\epsilon_{\ell+1}, j}^{N_{\ell+1}}(\theta_{\epsilon_{\ell+1}, j}^i) \right),$$

where $\theta_{\epsilon_{\ell+1}}^i$ is the jth component of $\boldsymbol{\theta}_{\epsilon_{\ell+1}}^i$ and $\hat{G}_{\epsilon_\ell, j}^{N_1, \ldots, N_\ell}(s)$ is the inverse of the jth marginal distribution of $\hat{F}_{\epsilon_\ell}^{N_1, \ldots, N_\ell}(\mathbf{s})$. This introduces a positive correlation between the sample pairs, $\{\boldsymbol{\theta}_{\epsilon_{\ell+1}}^i, \boldsymbol{\theta}_{\epsilon_\ell}^i\}$, since an increase in any of the components of $\boldsymbol{\theta}_{\epsilon_{\ell+1}}^i$ will cause an increase in the same component $\boldsymbol{\theta}_{\epsilon_\ell}^i$. This correlation reduces the variance in the bias correction estimator $\hat{Y}_{\epsilon_{\ell+1}}^{N_{\ell+1}}(\mathbf{s})$ computed according to Eq. (9). We can then update the MLMC CDF to get an improved estimator by using

$$\hat{F}_{\epsilon_{\ell+1}}^{N_1, \ldots, N_{\ell+1}}(\mathbf{s}) = \hat{F}_{\epsilon_\ell}^{N_1, \ldots, N_\ell}(\mathbf{s}) + \hat{Y}_{\epsilon_{\ell+1}}^{N_{\ell+1}}(\mathbf{s}),$$

and apply an adjustment that ensures monotonicity. We continue this process iteratively to obtain $\hat{F}_{\epsilon_L}^{N_1, \ldots, N_L}(\mathbf{s})$.

It must be noted here that this coupling mechanism introduces an approximation for the general inference problem; therefore some additional bias can be introduced. This is made clear when one considers the process in terms of the copula distributions of $\boldsymbol{\theta}_{\epsilon_{\ell+1}}$ and $\boldsymbol{\theta}_{\epsilon_\ell}$. If these copula distributions are the same, then the coupling is exact and there is no additional bias. The coupling is also exact for the univariate case ($k = 1$). Therefore, under the assumption that the correlation structure does not change significantly between levels, then the bias should be low. In practice, this requirement affects the choice of the acceptance threshold sequence, $\epsilon_1, \ldots, \epsilon_L$; we discuss this in more detail in Section 4. In Section 3.1, we demonstrate that for sensible choices of this sequence, the introduced bias is small compared with the bias that is inherent in the ABC approximation.

2.4. Optimal sample sizes

We now require the sample numbers $N_1, \ldots, N_L$ that are the optimal trade-off between accuracy and efficiency. Denote d_ℓ as the number of data generation steps required during the computation of $\hat{Y}_\ell^{N_\ell}(\mathbf{s})$ and let $c_\ell = d_\ell/N_\ell$ be the average number of data generation steps per accepted ABC posterior sample using acceptance threshold ϵ_ℓ.

Given $v_1 = \mathbb{V}\left[g_{\mathbf{s}}(\boldsymbol{\theta}_{\epsilon_1})\right]$ and $v_\ell = \mathbb{V}\left[g_{\mathbf{s}}(\boldsymbol{\theta}_{\epsilon_\ell}) - g_{\mathbf{s}}(\boldsymbol{\theta}_{\epsilon_{\ell-1}})\right]$, for $\ell > 1$, one can construct the optimal N_ℓ under the constraint $\mathbb{V}\left[\hat{F}_{\epsilon_L}^{N_1, \ldots, N_L}(\mathbf{s})\right] = \mathcal{O}(h^2)$, where h^2 is the target variance of the MLMC CDF estimator. As shown by Giles (2008), using a Lagrange multiplier method, the optimal $N_1, \ldots, N_L$ are given by

$$N_\ell = \mathcal{O}\left(h^{-2}\right) \sqrt{\frac{v_\ell}{c_\ell}} \sum_{m=1}^{L} \sqrt{v_m c_m}, \quad \ell = 1, \ldots, L. \tag{10}$$

In practice, the values for $v_1, \ldots, v_L$ and $c_1, \ldots, c_L$ will not have analytic expressions available; rather, we perform a low accuracy trial simulation with all $N_1 = \cdots = N_L = c$, for some comparatively small constant, c, to obtain the relative scaling of variances and data generation requirements.

2.5. Improving acceptance rates

A MLMC method based on the estimator in Eq. (8) and the variance reduction strategy given in Section 2.3 would depend on standard ABC rejection sampling (Algorithm 1) for the final bias correction term $\hat{Y}_{\epsilon_L}^{N_L}$. For many ABC applications of interest, the computation of this final term will dominate the computational costs. Therefore, the potential computational gains depend entirely on the size of N_L compared to the number of samples, N, required for the equivalent standard Monte Carlo approach (Eq. (5)). While this approach is often an improvement over rejection sampling, we can achieve further computational gains by exploiting the iterative computation of the bias correction terms.

Let $\text{supp}(f(x))$ denote the support of a function $f(x)$, and note that, for any $\ell \in [2, L]$, $\text{supp}(p(\boldsymbol{\theta}_{\epsilon_\ell})) \subseteq \text{supp}(p(\boldsymbol{\theta}_{\epsilon_{\ell-1}}))$. This follows from the fact that if, for any $\boldsymbol{\theta}$, $\mathbb{P}(d(\mathcal{D}, \mathcal{D}_s) < \epsilon_\ell \mid \boldsymbol{\theta}) > 0$, then $\mathbb{P}(d(\mathcal{D}, \mathcal{D}_s) < \epsilon_{\ell-1} \mid \boldsymbol{\theta}) > 0$ since $\epsilon_\ell < \epsilon_{\ell-1}$. That is, any simulated data generated using parameter values taken from outside $\text{supp}(p(\boldsymbol{\theta}_{\epsilon_{\ell-1}}))$ cannot be accepted on level ℓ since $d(\mathcal{D}, \mathcal{D}_s) > \epsilon_{\ell-1}$ almost surely. Therefore, we can truncate the prior to the support of $p(\boldsymbol{\theta}_{\epsilon_{\ell-1}})$ when computing $\hat{Y}_{\ell}^{N_\ell}(\mathbf{s})$, thus increasing the acceptance rate of level ℓ samples. In practice, we need to approximate the support regions through sampling. For simplicity, in this work we restrict sampling of the prior at level ℓ to within the bounding box that contains all the samples generated at level $\ell - 1$. However, more sophisticated approaches could be considered, and may result in further computational improvements.

2.6. The algorithm

We now have all the components to construct our MLMC-ABC algorithm. We compute the MLMC CDF estimator (Eq. (8)) using the coupling technique for the bias correction terms (Section 2.3) and prior truncation for improved acceptance rates (Section 2.5).

Optimal $N_1, \ldots, N_L$ are estimated as per Eq. (10) and Section 2.4. Once $N_1, \ldots, N_L$ have been estimated, computation of the MLMC-ABC posterior CDF $\hat{F}_{\epsilon_L}(\mathbf{s})$ proceeds according to Algorithm 4.

Algorithm 4 MLMC-ABC

1: Initialise $\epsilon_1, \ldots, \epsilon_L$, $N_1, \ldots, N_L$ and prior $p(\boldsymbol{\theta})$.
2: Set $p(\boldsymbol{\theta}_{\epsilon_0}) \leftarrow p(\boldsymbol{\theta})$.
3: **for** $\ell = 1, \ldots, L$ **do**
4:     **for** $i = 1, \ldots, N_\ell$ **do**
5:         **repeat**
6:             Sample $\boldsymbol{\theta}^* \sim p(\boldsymbol{\theta})$ restricted to $\text{supp}(p(\boldsymbol{\theta}_{\epsilon_{\ell-1}}))$.
7:             Generate data, $\mathcal{D}_s \sim f(\mathcal{D} \mid \boldsymbol{\theta}^*)$.
8:         **until** $d(\mathcal{D}, \mathcal{D}_s) \leq \epsilon_\ell$.
9:         Set $\boldsymbol{\theta}_{\epsilon_\ell}^i \leftarrow \boldsymbol{\theta}^*$.
10:     **end for**
11:     **for** $\mathbf{s} \in \mathcal{S}$ **do**
12:         Set $\hat{F}_{\epsilon_\ell}^{N_\ell}(\mathbf{s}) \leftarrow \sum_{i=1}^{N_\ell} g_{\mathbf{s}}(\boldsymbol{\theta}_{\epsilon_\ell}^i)/N_\ell$.
13:     **end for**
14:     **if** $\ell > 1$ **then**
15:         **for** $i = 1, \ldots, N_\ell$ **do**
16:             **for** $j = 1, \ldots, k$ **do**
17:                 Set $\theta_{\epsilon_{\ell-1},j}^i \leftarrow \hat{G}_{\epsilon_{\ell-1},j}^{N_1,\ldots,N_{\ell-1}} \left(\hat{F}_{\epsilon_\ell,j}^{N_\ell} \left(\theta_{\epsilon_\ell,j}^i \right) \right)$.
18:             **end for**
19:         **end for**
20:         **for** $\mathbf{s} \in \mathcal{S}$ **do**
21:             Set $\hat{Y}_{\epsilon_\ell}^{N_\ell}(\mathbf{s}) \leftarrow \sum_{i=1}^{N_\ell} \left[g_{\mathbf{s}}(\boldsymbol{\theta}_{\epsilon_\ell}^i) - g_{\mathbf{s}}(\boldsymbol{\theta}_{\epsilon_{\ell-1}}^i) \right]/N_\ell$.
22:             Set $\hat{F}_{\epsilon_\ell}^{N_1,\ldots,N_\ell}(\mathbf{s}) \leftarrow \hat{F}_{\epsilon_{\ell-1}}^{N_1,\ldots,N_{\ell-1}}(\mathbf{s}) + \hat{Y}_{\epsilon_\ell}^{N_\ell}(\mathbf{s})$.
23:         **end for**
24:     **end if**
25: **end for**

The computational complexity of MLMC-ABC (Algorithm 4) is roughly $\mathcal{O}(c_S + N_L(c_L + c_G))$ where c_L is the expected cost of generating a single sample of the ABC posterior with threshold ϵ_L, c_S is the cost of updating the CDF estimate and c_G is the cost of the coupling which involves the marginal CDF inverses. From Algorithm 4, we have $c_S = \mathcal{O}(N_1|\mathcal{S}|)$ and from Barber et al. (2015) we have $c_L = \mathcal{O}(\epsilon^{-n})$ where n is the dimensionality of $\mathcal{D}$. The marginal inverse operations require only two steps:

1. find $\mathbf{s} \in \mathcal{S}$ such that, $s_j \leq \theta_{\epsilon_\ell,j}^i < s_j + \delta_j$. Such an operation is, at most, $\mathcal{O}(\log_k|\mathcal{S}|)$;
2. invert the interpolating cubic spline, which can be done in $\mathcal{O}(1)$.

It follows that $c_G = \mathcal{O}(\log_k |\mathcal{S}|)$. For any practical application, ϵ_L will need to be sufficiently small, that is, $(N_1 |\mathcal{S}| / N_L - \log_k |\mathcal{S}|) \ll c_L$, in order for the cost of generating posterior samples at level L dominates the cost of the marginal CDF inverse operations and the lattice updating. Computational gains over ABC rejection sampling are achieved through decreasing N_L and c_L, via variance reduction and prior truncation.

Our primary focus in Algorithm 4 is on using MLMC to estimate the posterior CDF. The coupling mechanism is more readily communicated in this case. However, the MLMC-ABC method is more general and can be used to estimate $\mathbb{E}\left[U(\boldsymbol{\theta}_{\epsilon_\ell})\right]$ where $U(\boldsymbol{\theta}_{\epsilon_\ell})$ is any Lipschitz continuous function. In this more general case, only the marginal CDFs need be accumulated to facilitate the coupling mechanism. For more details see Appendix.

3. Results

In this section, we provide numerical results to demonstrate the validity, accuracy and performance of our MLMC-ABC method using some common models from epidemiology. In the first instance, we consider a tractable compartmental model, the stochastic SIS (Susceptible–Infected–Susceptible) model (Weiss and Dishon, 1971), to confirm the convergence and accuracy of MLMC-ABC. We then consider the Tuberculosis transmission model introduced by Tanaka et al. (2006) as a benchmark to compare our method with MCMC-ABC (Algorithm 2) and SMC-ABC (Algorithm 3). While we have chosen to perform our evaluation using an epidemiological model due to the particular prevalence of ABC methods in the life sciences, the techniques outlined in this manuscript are completely general and applicable to many areas of science.

3.1. A tractable example

The SIS model is a common model from epidemiology that describes the spread of a disease or infection for which no significant immunity is obtained after recovery; the common cold, for example. The model is given by

$$S + I \xrightarrow{\beta} 2I,$$
$$I \xrightarrow{\gamma} S,$$

with parameters $\boldsymbol{\theta} = \{\beta, \gamma\}$ and hazard functions for infection and recovery given by

$$h_I(S, I) = \beta S I \text{ and } h_R(S, I) = \gamma I,$$

respectively. This process defines a discrete-state continuous-time Markov process with a forward transitional density function that is computationally feasible to evaluate exactly for small populations $N_{pop} = S + I$. That is, for $t > s$, the probability of $S(t) = x$ given $S(s) = y$, with density denoted by $p(x, t \mid y, s; \beta, \gamma)$, has a solution obtained through the xth element of the vector

$$P(y, \beta, \gamma) = \exp(Q(\beta, \gamma)(t - s))\mathbf{y}, \tag{11}$$

where the yth element of column vector, $\mathbf{y}$, is one and all other elements zero, $\exp(\cdot)$ denotes the matrix exponential and $Q(\beta, \gamma)$ is the infinitesimal generator matrix of the Markov process; for the SIS model, $Q(\beta, \gamma)$ is a tri-diagonal matrix dependent only on the parameters of the model.

Let $S_{obs}(t)$ be an observation at time t of the number of susceptible individuals in the population. We generate observations using a single realisation of the SIS model with parameters $\beta = 0.003$ and $\gamma = 0.1$, population size $N_{pop} = 101$, and initial conditions $S(0) = 100$ and $I(0) = 1$; Observations are taken at times $t_1 = 4, t_2 = 8, \ldots, t_{10} = 40$. Using the analytic solution to the SIS transitional density (using Eq. (11)), we arrive at the likelihood function

$$\mathcal{L}(\beta, \gamma; \mathcal{D}) = \prod_{i=1}^{10} p(S_{obs}(t_i), t_i \mid S_{obs}(t_{i-1}), t_{i-1}; \beta, \gamma), \tag{12}$$

where $S_{obs}(t_0) = s_0$ almost surely. Hence, we can obtain an exact solution to the SIS posterior $p(\beta, \gamma \mid \mathcal{D})$ given the priors $\beta \sim \mathcal{U}(0, 0.06)$ and $\gamma \sim \mathcal{U}(0, 2)$. Given this exact posterior, quadrature can be applied to compute the posterior CDF, given by

$$F(s_1, s_2) = \int_{-\infty}^{s_1} \int_{-\infty}^{s_2} p(\beta, \gamma \mid \mathcal{D}) \, d\beta \, d\gamma.$$

For the ABC approximation we use a discrepancy metric based on the sum of squared errors,

$$d(\mathcal{D}, \mathcal{D}_s) = \left[\sum_{i=1}^{10} \left(S_{obs}(t_i) - S^*(t_i) \right)^2 \right]^{1/2},$$

where $S^*(t)$ is a realisation of the model generated using the Gillespie algorithm (Gillespie, 1977) for a given set of parameters, and $\mathcal{D}_s = [S^*(t_1), S^*(t_2), \ldots, S^*(t_{10})]$. An appropriate acceptance threshold sequence for this metric is $\epsilon_1, \ldots, \epsilon_L$, with $\epsilon_\ell = \epsilon_1 m^{1-\ell}$, $m = 2$ and $\epsilon_1 = 75$ (Toni et al., 2009).
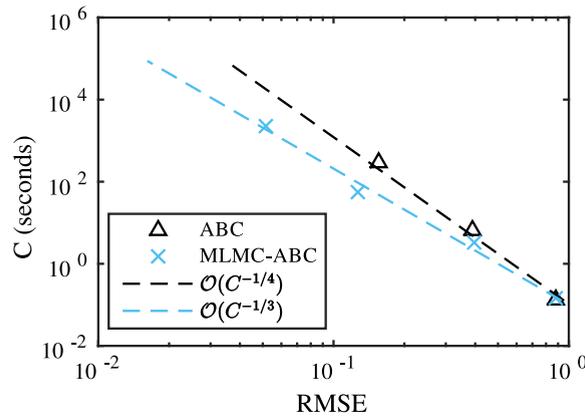
Fig. 1. RMSE convergence for MLMC-ABC compared with ABC rejection sampling. The RMSE is computed using the exact solution to the posterior density of the SIS model.

We can use the exact likelihood (Eq. (12)) to evaluate the convergence properties of our MLMC-ABC method. Based on the analysis of ABC rejection sampling by Barber et al. (2015), the rate of decay of the root mean-squared error (RMSE) as the computational cost is increased is slower for ABC than standard Monte Carlo. We find experimentally for ABC rejection sampling, that the decay of RMSE of the SIS model is approximately $\mathcal{O}(C^{-1/4})$, where C is the expected computational cost of generating N ABC posterior samples. The 95% confidence interval of this experimental rate is $[-0.27, -0.23]$, which is consistent with the expected theoretical result of -0.25 (Barber et al., 2015). This is slower than the expected $\mathcal{O}(C^{-1/2})$ decay typically achieved with standard Monte Carlo. We compare the ABC rejection sampler decay rate with that achieved by our MLMC-ABC methods.

We use the L_∞ norm for the RMSE, that is,

$$\text{RMSE} = \sqrt{\mathbb{E}\left[\left\| F - \hat{F}_{\epsilon_L} \right\|_\infty^2\right]},$$

where F is the exact posterior CDF, evaluated directly from the likelihood (Eq. (12)), and $\hat{F}_{\epsilon_L}$ is the Monte Carlo estimate of the CDF. A regular lattice that consists of 300×300 points is used for this estimate. The computation required for Gillespie simulations completely dominates the added computation of updating the lattice. The RMSE is computed using 20 independent MLMC-ABC CDF estimations for $L = 1, \ldots, 3$. The sequence of sample numbers, $N_1, \ldots, N_L$, is computed using Eq. (10) with target variance of $\mathcal{O}(\epsilon^2)$ and 100 trial samples. Fig. 1 demonstrates the improved convergence rate over the ABC rejection sampler convergence rate. Based on the least-squares fit, the RMSE decay is approximately $\mathcal{O}(C^{-1/3})$. The 95% confidence interval for the rate is $[-0.26, -0.34]$ which is consistent with theory from Giles et al. (2015) for the univariate situation. For smaller target RMSE, this results in an order of magnitude reduction in the computational cost.

We also consider the effect of the additional bias introduced through the coupling mechanism as presented in Section 2.3. We set the sample numbers at all levels to be 10^4 to ensure the Monte Carlo error is negligible and compare the bias for different values of acceptance threshold scaling factor m. The bias is computed according to the L_∞ norm, that is,

$$\text{Bias} = \mathbb{E}\left[\left\| \hat{F}_{\epsilon_L}^c - \hat{F}_{\epsilon_L}^u \right\|_\infty\right],$$

where $\hat{F}_{\epsilon_L}^c$ denotes the estimator computed according to Algorithm 4 and $\hat{F}_{\epsilon_L}^u$ denotes the estimator computed without any coupling. That is, $\hat{F}_{\epsilon_L}^u$ is computed using standard Monte Carlo to evaluate each term in the MLMC telescoping sum (Eq. (6)). Note that, computationally, $\hat{F}_{\epsilon_L}^u$ will always be inferior to a standard Monte Carlo estimate. Fig. 2 shows that, not only does the bias decay as m decreases, but also the additional bias is well within the order of bias expected from the ABC approximation. This result, along with Fig. 1, demonstrates that the reduction in estimator variance can dominate the increase in bias. Thus, compared with standard Monte Carlo, a significantly lower RMSE for the same computational effort is achieved with MLMC using this coupling strategy.

3.2. Performance evaluation

We now evaluate the performance of MLMC-ABC using the model developed by Tanaka et al. (2006) in the study of tuberculosis transmission rate parameters using DNA fingerprint data (Small et al., 1994). This model has been selected due to the availability of published comparative performance evaluations of MCMC-ABC and SMC-ABC (Tanaka et al., 2006; Sisson et al., 2007).
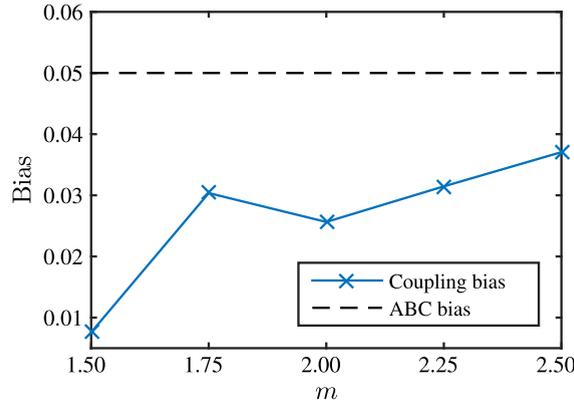
Fig. 2. Convergence of coupling bias as a function of m.

The model proposed by Tanaka et al. (2006) describes the occurrence of tuberculosis infections over time and the mutation of the bacterium responsible, *Mycobacterium tuberculosis*. The number of infections, I, at time t is

$$I_t = \sum_{i=1}^{G_t} X_{i,t},$$

where G_t is the number of distinct genotypes and $X_{i,t}$ is the number of infections caused by the ith genotype at time t. For each genotype, new infections occur with rate α, infections terminate with rate δ, and mutation occurs with rate μ; causing an increase in the number of genotypes. This process, as with the SIS model, can be described by a discrete-state continuous-time Markov process. In this case, however, the likelihood is intractable, but the model can still be simulated using the Gillespie algorithm (Gillespie, 1977). After a realisation of the model is completed, either by extinction or when a maximum infection count is reached, a sub-sample of 473 cases is collected and compared against the IS6110 DNA fingerprint data of tuberculosis bacteria samples (Small et al., 1994). The dataset consists of 326 distinct genotypes; the infection cases are clustered according to the genotype responsible for the infection. The collection of clusters can be summarised succinctly as $30^1\,23^1\,15^1\,10^1\,8^1\,5^2\,4^4\,3^{13}\,2^{20}\,1^{282}$, where n^j denotes there are j clusters of size n. The discrepancy metric used is

$$d(\mathcal{D}, \mathcal{D}_s) = \frac{1}{n}\,|g(\mathcal{D}) - g(\mathcal{D}_s)| + |H(\mathcal{D}) - H(\mathcal{D}_s)|, \tag{13}$$

where n is the size of the population sub-sample (e.g., $n = 473$), $g(\mathcal{D})$ denotes the number of distinct genotypes in the dataset (e.g., $g(\mathcal{D}) = 326$) and the genetic diversity is

$$H(\mathcal{D}) = 1 - \frac{1}{n^2} \sum_{i=1}^{g(\mathcal{D})} n_i(\mathcal{D})^2,$$

where $n_i(\mathcal{D})$ is the cluster size of the ith genotype in the dataset.

We perform likelihood-free inference on the tuberculosis model for the parameters $\boldsymbol{\theta} = \{\alpha, \delta, \mu\}$ with the goal of evaluating the efficiencies of MLMC-ABC, MCMC-ABC and SMC-ABC. We use a target posterior distribution of $p(\boldsymbol{\theta} \mid d(\mathcal{D}, \mathcal{D}_s) < \epsilon)$ with $d(\mathcal{D}, \mathcal{D}_s)$ as defined in Eq. (13) and $\epsilon = 0.0025$. The acceptance threshold sequence, $\epsilon_1, \ldots, \epsilon_{10}$, used for both SMC-ABC and MLMC-ABC is $\epsilon_i = \epsilon_{10} + (\epsilon_{i-1} - \epsilon_{10})/2$ with $\epsilon_1 = 1$ and $\epsilon_{10} = 0.0025$. The improper prior is given by $\alpha \sim \mathcal{U}(0, 5)$, $\delta \sim \mathcal{U}(0, \alpha)$ and $\mu \sim \mathcal{N}(0.198, 0.06735^2)$ (Sisson et al., 2007; Tanaka et al., 2006). For the MCMC-ABC and SMC-ABC algorithms we apply a typical Gaussian proposal kernel,

$$K(\boldsymbol{\theta}^{(i)} \mid \boldsymbol{\theta}^{(i-1)}) = \mathcal{N}\left(\boldsymbol{\theta}^{(i-1)}, \boldsymbol{\Sigma}\right),$$

with covariance matrix

$$\boldsymbol{\Sigma} = \begin{bmatrix} 0.75^2 & 0 & 0 \\ 0 & 0.75^2 & 0 \\ 0 & 0 & 0.03^2 \end{bmatrix}. \tag{14}$$

Such a proposal kernel is reasonable to characterise the initial explorations of an ABC posterior as no correlations between parameters are assumed.

While MLMC-ABC does not require a proposal kernel function, some knowledge of the variance of each bias correction term is needed to determine the optimal sample numbers, $N_1, \ldots, N_L$. This is achieved using 100 trial samples of each level.
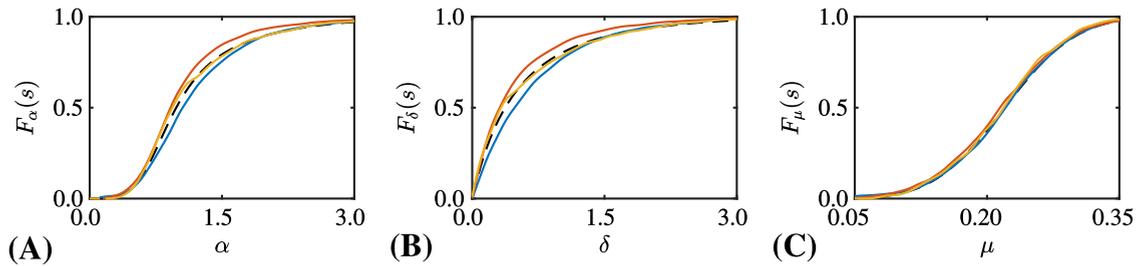
Fig. 3. Estimated marginal CDFsi – (**A**) α, (**B**) δ and (**C**) μ – for the tuberculosis transmission stochastic model. Estimate computed using MLMC-ABC with $N_L = 800$ (solid yellow), MCMC-ABC over 1.2×10^6 iterations (solid blue), SMC-ABC with 3200 particles (solid red) and high precision solution (dashed black). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 1
Comparison of MLMC-ABC against MCMC-ABC and SMC-ABC using a naive proposal kernel.

MLMC-ABC			MCMC-ABC			SMC-ABC		
N_L	N_s	RMSE	N_T	N_s	RMSE	N_P	N_s	RMSE
100	102,246	0.1362	160,000	163,800	0.2434	400	347,281	0.1071
200	255,443	0.1136	320,000	323,841	0.1828	800	701,100	0.0954
400	577,312	0.1067	640,000	643,930	0.1832	1600	1,385,790	0.0858
800	1,040,140	0.0861	1,280,000	1,283,590	0.1345	3200	2,792,510	0.0784

Table 2
Comparison of MLMC-ABC against MCMC-ABC and SMC-ABC using heuristically chosen proposal densities.

MLMC-ABC			MCMC-ABC			SMC-ABC		
N_L	N_s	RMSE	N_T	N_s	RMSE	N_P	N_s	RMSE
100	102,246	0.1362	160,000	161,604	0.1718	400	102,902	0.1270
200	255,443	0.1136	320,000	322,962	0.1254	800	198,803	0.0767
400	577,312	0.1067	640,000	642,412	0.1127	1600	402,334	0.0652
800	1,040,140	0.0861	1,280,000	1,305,340	0.0934	3200	797,893	0.0560

The number of data generation steps is also recorded to compute $N_1, \ldots, N_L$, as per Eq. (10). The resulting sample numbers are then scaled such that N_L is a user prescribed value.

The efficiency metric we use is the root mean-squared error (RMSE) of the CDF estimate versus the total number of data generation steps, N_s. The mean-squared error (MSE) is taken under the L_∞ norm, that is, MSE $= \mathbb{E}\left[\|F_{\epsilon_\ell} - \hat{F}_{\epsilon_\ell}\|_\infty^2 \right]$ where F_{ϵ_ℓ} is the exact ABC posterior CDF and $\hat{F}_{\epsilon_\ell}$ is the Monte Carlo estimate using a regular lattice with $100 \times 100 \times 100$ points. To compute the RMSE, a high precision solution is computed using 10^6 ABC rejection samples of the target ABC posterior. This is computed over a period of 48 h using 500 processor cores.

Table 1 presents the RMSE for MCMC-ABC, SMC-ABC and MLMC-ABC for different configurations. The RMSE values are computed using 10 independent estimator calculations. The algorithm parameter varied is the number of particles, N_P, for SMC-ABC, the number of iterations, N_T, for MCMC-ABC and the level L sample number, N_L, for MLMC-ABC. Using the proposal kernel provided in Eq. (14), SMC-ABC requires almost 30% more data generation steps than MLMC-ABC to obtain the same RMSE. MLMC-ABC obtains nearly double the accuracy of MCMC-ABC for the same number of data generation steps. Fig. 3 shows an example of the high precision marginal posterior CDFs, $F_\alpha(s)$, $F_\delta(s)$ and $F_\mu(s)$, compared with the numerical solutions computed using the three methods.

We note that these results represent a typical scenario when solving this problem with a standard choice of proposal densities for MCMC-ABC and SMC-ABC. However, obtaining a good proposal kernel is a difficult open problem, and infeasible to do heuristically for high-dimensional parameter spaces. Therefore, efficient proposal kernels are almost never obtained without significant manual adjustment or additional algorithmic modifications such as adaptive schemes (Beaumont et al., 2009; Del Moral et al., 2012; Roberts and Rosenthal, 2009). Nevertheless, we demonstrate in Table 2 the increased efficiency for MCMC-ABC and SMC-ABC when using a highly configured Gaussian proposal kernel with covariance matrix as determined by Tanaka et al. (2006)

$$\Sigma = \begin{bmatrix} 0.5^2 & 0.225 & 0 \\ 0.225 & 0.5^2 & 0 \\ 0 & 0 & 0.015^2 \end{bmatrix}. \tag{15}$$

We note that Sisson et al. (2007) use a similar proposal kernel, however they do not explicitly state the difference in the covariance matrix Σ; we therefore assume that Eq. (15) represents a proposal kernel that is heuristically optimal to the

target posterior density. We emphasise that it would be incredibly rare to arrive at an optimal proposal kernel without any additional experimentation. Even in this unlikely case, MLMC-ABC is still comparable with MCMC-ABC. However, MLMC-ABC is clearly not as efficient as SMC-ABC for this heuristically optimised scenario. The scenario is intentionally biased toward MCMC-ABC and SMC-ABC, so this result is not unexpected. Future research could consider a comparison of the methods when the extra computational burden of determining the optimal Σ, such as implementation of an adaptive scheme, is taken into account.

4. Discussion

Our results indicate that, while SMC-ABC and MCMC-ABC can be heuristically optimised to be highly efficient, an accurate estimate of the parameter posterior can be obtained using the MLMC-ABC method presented here in a relatively automatic fashion. Furthermore, the efficiency of MLMC-ABC is comparable or improved over MCMC-ABC and SMC-ABC, even in the case when efficient proposal kernels are employed.

The need to estimate the variances of each bias correction term could be considered a limitation of the MLMC-ABC approach. However, we find in practice that these need not be computed to high accuracy and can often be estimated with a relatively small number of samples. There could be examples of Bayesian inference problems where MLMC-ABC is inefficient on account of the variance estimation inaccuracy. We have so far, however, failed to find an example for which 100 samples of each bias correction term is insufficient to obtain a good MLMC-ABC estimator.

There are many modifications one could consider to further improve MLMC-ABC. In this work, we explicitly specify the sequence of acceptance thresholds in advance for both MLMC-ABC and SMC-ABC. As this is the initial presentation of the method, it is appropriate to consider this idealised case. However, it is unclear if an optimal sequence of thresholds for SMC-ABC will also be optimal for MLMC-ABC and vice versa. Furthermore, as mentioned in Section 1, practical applications of SMC-ABC often determine such sequences adaptively (Drovandi and Pettitt, 2011; Silk et al., 2013). A modification of MLMC-ABC to allow for adaptive acceptance thresholds would make MLMC-ABC even more practical as it could be used to minimise the coupling bias. The exact mechanisms to achieve this could be based on similar ideas to adaptive SMC-ABC (Drovandi and Pettitt, 2011). Given the solution at level $\ell - 1$, the next level ℓ could be determined through: (i) sampling a fixed set of *prior* samples; (ii) sorting these samples based on the discrepancy metric; (iii) selecting a discrepancy threshold, ϵ_ℓ, that optimises coupling bias and the variance of the bias correction term. Future work should address these open problems.

Other improvements could focus on the discretisation used for the eCDF calculations. The MLMC-ABC method has no requirement of a regular lattice, and alternative choices would likely enable MLMC-ABC to scale to much higher dimensional parameter spaces. Adaptive grids that refine with each level is one option that could be considered; however, unstructured grids or kernel based methods also have potential.

Improvements to the coupling scheme are also possible avenues for future consideration. The coupling approach we have considered depends only on the computation of the marginal posterior CDFs and assumes nothing about the underlying model. It may be possible to take advantage of certain model specific features to improve variance reduction. There may also be cases where the rejection sampling scheme is prohibitive for the more accurate acceptance levels. The combination of our coupling scheme with the MLSMC scheme recently proposed by Beskos et al. (2017) and Jasra et al. (2017) is a promising possibility to mitigate these issues. Future work should investigate, compare and contrast the variety of available coupling strategies in the growing body of literature on MLMC for ABC and Bayesian inverse problems.

We have shown, in a practical way, how MLMC techniques can be applied to ABC inference. We also demonstrate that variance reduction strategies, even when applied to simple methods such as rejection sampling, can achieve performance improvements comparable, and in some cases superior, to modern advanced ABC methods based on MCMC and SMC methods. Therefore, the MLMC framework is a promising area for designing improved samplers for complex statistical problems with intractable likelihoods.

Acknowledgements

Appendix. MLMC for ABC with general Lipschitz functions

Minor modifications of the MLMC-ABC method (Algorithm 4) are possible to enable the computation of expectations of the form

$$\mathbb{E}\left[U(\boldsymbol{\theta}) \mid d(\mathcal{D}, \mathcal{D}_s) < \epsilon\right] = \int_{\Theta} U(\boldsymbol{\theta}) p(\boldsymbol{\theta} \mid d(\mathcal{D}, \mathcal{D}_s) < \epsilon) \, d\theta_k \ldots d\theta_1,$$

where $U(\boldsymbol{\theta})$ is any Lipschitz continuous function.

Using the same notation as defined in Section 2.2, the MLMC telescoping sum may be formed for a given sequence of acceptance thresholds, $\epsilon_1 > \cdots > \epsilon_L$, to compute the expectation $E_{\epsilon_L} = \mathbb{E}\left[U(\boldsymbol{\theta}_{\epsilon_L})\right]$. That is,

$$E_{\epsilon_L} = \sum_{\ell=1}^{L} P_{\epsilon_\ell}, \quad P_{\epsilon_\ell} = \begin{cases} \mathbb{E}\left[U(\boldsymbol{\theta}_{\epsilon_1})\right], & \ell = 1, \\ \mathbb{E}\left[U(\boldsymbol{\theta}_{\epsilon_\ell}) - U(\boldsymbol{\theta}_{\epsilon_{\ell-1}})\right], & \ell > 1. \end{cases} \tag{A.1}$$

From Eq. (A.1), it is straightforward to obtain equivalent expressions to Eqs. (8) and (9).

Algorithm 5 Modified MLMC-ABC

1: Initialise $\epsilon_1, \ldots, \epsilon_L, N_1, \ldots, N_L$ and prior $p(\boldsymbol{\theta})$.
2: Set $p(\boldsymbol{\theta}_{\epsilon_0}) \leftarrow p(\boldsymbol{\theta})$.
3: **for** $\ell = 1, \ldots, L$ **do**
4:      **for** $i = 1, \ldots, N_\ell$ **do**
5:          **repeat**
6:              Sample $\boldsymbol{\theta}^* \sim p(\boldsymbol{\theta})$ restricted to supp($p(\boldsymbol{\theta}_{\epsilon_{\ell-1}})$).
7:              Generate data, $\mathcal{D}_s \sim f(\mathcal{D} \mid \boldsymbol{\theta}^*)$.
8:          **until** $d(\mathcal{D}, \mathcal{D}_s) \leq \epsilon_\ell$.
9:          Set $\boldsymbol{\theta}_{\epsilon_\ell}^i \leftarrow \boldsymbol{\theta}^*$.
10:      **end for**
11:      **for** $j = 1, \ldots, k$ **do**
12:          **for** $s \in \mathcal{S}_j$ **do**
13:              Set $\hat{F}_{\epsilon_\ell,j}^{N_\ell}(s) \leftarrow \sum_{i=1}^{N_\ell} \xi\left((s_j - \theta_{\epsilon_\ell,j}^i)/\delta_j\right)/N_\ell$.
14:          **end for**
15:      **end for**
16:      **if** $\ell = 1$ **then**
17:          $\hat{E}_{\epsilon_\ell}^{N_\ell} \leftarrow \sum_{i=1}^{N_\ell} U(\boldsymbol{\theta}_{\epsilon_\ell}^i)/N_\ell$
18:      **else**
19:          **for** $i = 1, \ldots, N_\ell$ **do**
20:              **for** $j = 1, \ldots, k$ **do**
21:                  Set $\theta_{\epsilon_{\ell-1},j}^i \leftarrow \hat{G}_{\epsilon_{\ell-1},j}^{N_1,\ldots,N_{\ell-1}}\left(\hat{F}_{\epsilon_\ell,j}^{N_\ell}\left(\theta_{\epsilon_\ell,j}^i\right)\right)$.
22:              **end for**
23:          **end for**
24:          **for** $j = 1, \ldots, k$ **do**
25:              **for** $s \in \mathcal{S}_j$ **do**
26:                  Set $\hat{Y}_{\epsilon_\ell,j}^{N_\ell}(s) \leftarrow \sum_{i=1}^{N_\ell}\left[\xi\left((s_j - \theta_{\epsilon_\ell,j}^i)/\delta_j\right) - \xi\left((s_j - \theta_{\epsilon_{\ell-1},j}^i)/\delta_j\right)\right]/N_\ell$.
27:                  Set $\hat{F}_{\epsilon_\ell,j}^{N_1,\ldots,N_\ell}(s) \leftarrow \hat{F}_{\epsilon_{\ell-1},j}^{N_1,\ldots,N_{\ell-1}}(s) + \hat{Y}_{\epsilon_\ell,j}^{N_\ell}(s)$.
28:              **end for**
29:          **end for**
30:          $\hat{P}_{\epsilon_\ell}^{N_\ell} \leftarrow \sum_{i=1}^{N_\ell}\left[U(\boldsymbol{\theta}_{\epsilon_\ell}^i) - U(\boldsymbol{\theta}_{\epsilon_{\ell-1}}^i))\right]/N_\ell$
31:          $\hat{E}_{\epsilon_\ell}^{N_1,\ldots,N_\ell} \leftarrow \hat{E}_{\epsilon_{\ell-1}}^{N_1,\ldots,N_{\ell-1}} + \hat{P}_{\epsilon_\ell}^{N_\ell}$.
32:      **end if**
33: **end for**

The modified MLMC-ABC algorithm proceeds in a very similar fashion to Algorithm 4, however, there is no need to hold a complete discretisation of the parameter space, Θ. This is because only the k marginal CDFs, $F_{\epsilon_\ell,1}(s_1), \ldots, F_{\epsilon_\ell,k}(s_k)$, are required to form the coupling strategy in Section 2.3. Thus, we denote $\mathcal{S}_j$ to be a discretisation of the jth coordinate axis. This significantly reduces the computational burden of the lattice in higher dimensions since $|\mathcal{S}_j| = \mathcal{O}(\log_k|\mathcal{S}|)$. The resulting algorithm is given in Algorithm 5.

References

Anderson, D.F., Higham, D.J., 2012. Multilevel Monte Carlo for continuous time Markov chains, with applications in biochemical kinetics. Multiscale Model. Simul. 10 (1), 146–179. http://dx.doi.org/10.1137/110840546.

Avikainen, R., 2009. On irregular functionals of SDEs and the Euler scheme. Financ. Stoch. 13 (3), 381–401. http://dx.doi.org/10.1007/s00780-009-0099-7.

Barber, S., Voss, J., Webster, M., 2015. The rate of convergence for approximate Bayesian computation. Electron. J. Stat. 9, 80–105. http://dx.doi.org/10.1214/15-EJS988.

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

Beaumont, M.A., Zhang, W., Balding, D.J., 2002. Approximate Bayesian computation in population genetics. Genetics 162 (4), 2025–2035.

Beskos, A., Jasra, A., Law, K., Tempone, R., Zhou, Y., 2017. Multilevel sequential Monte Carlo samplers. Stochastic Process. Appl. 127 (5), 1417–1440. http://dx.doi.org/10.1016/j.spa.2016.08.004.

Browning, A.P., McCue, S.W., Binny, R.N., Plank, M.J., Shah, E.T., Simpson, M.J., 2018. Inferring parameters for a lattice-free model of cell migration and proliferation using experimental data. J. Theoret. Biol. 437, 251–260. http://dx.doi.org/10.1016/j.jtbi.2017.10.032.

Cabras, S., Castellanos Neuda, M.E., Ruli, E., 2015. Approximate Bayesian computation by modelling summary statistics in a quasi-likelihood framework. Bayesian Anal. 10 (2), 411–439. http://dx.doi.org/10.1214/14-BA921.

Del Moral, P., Doucet, A., Jasra, A., 2006. Sequential Monte Carlo samplers. J. R. Stat. Soc. Ser. B Stat. Methodol. 68 (3), 411–436. http://dx.doi.org/10.1111/j.1467-9868.2006.00553.x.

Del Moral, P., Doucet, A., Jasra, A., 2012. An adaptive sequential Monte Carlo method for approximate Bayesian computation. Stat. Comput. 22 (5), 1009–1020. http://dx.doi.org/10.1007/s11222-011-9271-y.

Dodwell, T.J., Ketelsen, C., Scheichl, R., Teckentrup, A.L., 2015. A hierarchical multilevel Markov chain Monte Carlo algorithm with applications to uncertainty quantification in subsurface flow. SIAM/ASA J. Uncertain. Quantif. 3 (1), 1075–1108. http://dx.doi.org/10.1137/130915005.

Drovandi, C.C., Pettitt, A.N., 2011. Estimation of parameters for macroparasite population evolution using approximate Bayesian computation. Biometrics 67 (1), 225–233. http://dx.doi.org/10.1111/j.1541-0420.2010.01410.x.

Efendiev, Y., Jin, B., Michael, P., Tan, X., 2015. Multilevel Markov chain Monte Carlo method for high-contrast single-phase flow problems. Commun. Comput. Phys. 17 (1), 259–286. http://dx.doi.org/10.4208/cicp.021013.260614a.

Fearnhead, P., Prangle, D., 2012. Constructing summary statistics for approximate Bayesian computation: semi-automatic approximate Bayesian computation. J. R. Stat. Soc. Ser. B Stat. Methodol. 74 (3), 419–474. http://dx.doi.org/10.1111/j.1467-9868.2011.01010.x.

Filippi, S., Barnes, C.P., Cornebise, J., Stumpf, M.P.H., 2013. On optimality of kernels for approximate Bayesian computation using sequential Monte Carlo. Stat. Appl. Genet. Mol. Biol. 12 (1), 87–107. http://dx.doi.org/10.1515/sagmb-2012-0069.

Giles, M.B., 2008. Multilevel Monte Carlo path simulation. Oper. Res. 56 (3), 607–617. http://dx.doi.org/10.1287/opre.1070.0496.

Giles, M.B., Nagapetyan, T., Ritter, K., 2015. Multilevel Monte Carlo approximation of cumulative distribution function and probability densities. SIAM/ASA J. Uncertain. Quantif. 3 (1), 267–295. http://dx.doi.org/10.1137/140960086.

Gillespie, D.T., 1977. Exact stochastic simulation of coupled chemical reactions. J. Phys. Chem. 81 (25), 2340–2361. http://dx.doi.org/10.1021/j100540a008.

Green, P.J., Łatuszyński, K., Pereyra, M., Robert, C.P., 2015. Bayesian computation: a summary of the current state, and samples backwards and forwards. Stat. Comput. 25 (4), 835–862. http://dx.doi.org/10.1007/s11222-015-9574-5.

Gregory, A., Cotter, C.J., Reich, S., 2016. Multilevel ensemble transform particle filtering. SIAM J. Sci. Comput. 38 (3), A1317–A1338. http://dx.doi.org/10.1137/15M1038232.

Grelaud, A., Marin, J.-M., Robert, C.P., Rodolphe, F., Tallay, J.-F., 2009. ABC likelihood-free methods for model choice in Gibbs random fields. Bayesian Anal. 4 (2), 317–335. http://dx.doi.org/10.1214/09-BA412.

Guha, N., Tan, X., 2017. Multilevel approximate Bayesian approaches for flows in highly heterogeneous porous media and their applications. J. Comput. Appl. Math. 317, 700–717. http://dx.doi.org/10.1016/j.cam.2016.10.008.

Hastings, W.K., 1970. Monte Carlo sampling methods using Markov chains and their applications. Biometrika 57 (1), 97–109.

Jasra, A., Jo, S., Nott, D., Shoemaker, C., Tempone, R., 2017. Multilevel Monte Carlo in approximate Bayesian computation. ArXiv e-prints arXiv:1702.03628.

Johnston, S.T., Simpson, M.J., McElwain, D.L.S., Binder, B.J., Ross, J.V., 2014. Interpreting scratch assays using pair density dynamics and approximate Bayesian computation. Open Biol. 4 (9), 140097. http://dx.doi.org/10.1098/rsob.140097.

Lester, C., Baker, R.E., Giles, M.B., Yates, C.A., 2016. Extending the multi-level method for the simulation of stochastic biological systems. Bull. Math. Biol. 78 (8), 1640–1677. http://dx.doi.org/10.1007/s11538-016-0178-9.

Marin, J.-M., Pudlo, P., Robert, C.P., Ryder, R.J., 2012. Approximate Bayesian computational methods. Stat. Comput. 22 (6), 1167–1180. http://dx.doi.org/10.1007/s11222-011-9288-2.

Marjoram, P., Molitor, J., Plagnol, V., Tavaré, S., 2003. Markov chain Monte Carlo without likelihoods. Proc. Natl. Acad. Sci. USA 100 (26), 15 324–15, 328. http://dx.doi.org/10.1073/pnas.0306899100.

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E., 1953. Equation of state calculations by fast computing machines. J. Chem. Phys. 21 (6), 1087–1092. http://dx.doi.org/10.1063/1.1699114.

Navascués, M., Leblois, R., Burgarella, C., 2017. Demographic inference through approximate-Bayesian-computation skyline plots. PeerJ 5, e3530. http://dx.doi.org/10.7717/peerj.3530.

Pooley, C.M., Bishop, S.C., Marion, G., 2015. Using model-based proposals for fast parameter inference on discrete state space, continuous-time Markov processes. J. R. Soc. Interface 12 (107), 20150225. http://dx.doi.org/10.1098/rsif.2015.0225.

Pritchard, J.K, Seielstad, M.T., Perez-Lezaun, A., Feldman, M.W., 1999. Population growth of human Y chromosomes: a study of Y chromosome microsatellites. Mol. Biol. Evol. 16 (12), 1791–1798. http://dx.doi.org/10.1093/oxfordjournals.molbev.a026091.

Reiss, R.-D., 1981. Nonparametric estimation of smooth distribution functions. Scand. J. Stat. 8, 116–119.

Roberts, G.O., Rosenthal, J.S., 2004. General state space Markov chains and MCMC algorithms. Prob. Surv. 1, 20–71. http://dx.doi.org/10.1214/154957804100000024.

Roberts, G.O., Rosenthal, J.S., 2009. Examples of adaptive MCMC. J. Comput. Graph. Stat. 18 (2), 349–367. http://dx.doi.org/10.1198/jcgs.2009.06134.

Ross, R.J.H., Baker, R.E., Parker, A., Ford, M.J., Mort, R.L., Yates, C.A., 2017. Using approximate Bayesian computation to quantify cell–cell adhesion parameters in cell migratory process. Npj Syst. Biol. Appl. 3 (1), 9. http://dx.doi.org/10.1038/s41540-017-0010-7.

Silk, D., Filippi, S., Stumpf, M.P.H., 2013. Optimizing threshold-schedules for sequential approximate Bayesian computation: applications to molecular systems. Stat. Appl. Genet. Mol. Biol. 12 (3), 603–618. http://dx.doi.org/10.1515/sagmb-2012-004.

Silverman, B.W., 1986. Density Estimation for Statistics and Data Analysis. In: Monographs on Statistics and Applied Probability, Chapman & Hall/CRC, Boca Ranton.

Sisson, S.A., Fan, Y., Tanaka, M.M., 2007. Sequential Monte Carlo without likelihoods. Proc. Natl. Acad. Sci. USA 104 (6), 1760–1765. http://dx.doi.org/10.1073/pnas.0607208104.

Small, P.M., Hopewell, P.C., Singh, S.P., Paz, A., Parsonnet, J., Ruston, D.C., Schecter, G.F., Daley, C.L., Schoolnik, G.K., 1994. The epidemiology of tuberculosis in San Francisco –a population-based study using conventional and molecular methods. New Engl. J. Med. 330 (24), 1703–1709. http://dx.doi.org/10.1056/NEJM199406163302402.

Stumpf, M.P.H., 2014. Approximate Bayesian inference for complex ecosystems. F1000Prime Rep. 6, 60. http://dx.doi.org/10.12703/P6-60.

Sunnåker, M., Busetto, A.G., Numminen, E., Corander, J., Foll, M., Dessimoz, C., 2013. Approximate Bayesian computation. PLOS Comput. Biol. 9 (1), e1002803. http://dx.doi.org/10.1371/journal.pcbi.1002803.

Tanaka, M.M., Francis, A.R., Luciani, F., Sisson, S.A., 2006. Using approximate Bayesian computation to estimate tuberculosis transmission parameter from genotype data. Genetics 173 (3), 1511–1520. http://dx.doi.org/10.1534/genetics.106.055574.

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

Thorne, T., Stumpf, M.P.H., 2012. Graph spectral analysis of protein interaction network evolution. J. R. Soc. Interface 9 (75), 2653–2666. http://dx.doi.org/10.1098/rsif.2012.0220.

Toni, T., Welch, D., Strelkowa, N., Ipsen, A., Stumpf, M.P.H., 2009. Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems. J. R. Soc. Interface 6, 187–202. http://dx.doi.org/10.1098/rsif.2008.0172.

Vo, B.N., Drovandi, C.C., Pettit, A.N., Simpson, M.J., 2015. Quantifying uncertainty in parameter estimates for stochastic models of collective cell spreading using approximate Bayesian computation. Math. Biosci. 263, 133–142. http://dx.doi.org/10.1016/j.mbs.2015.02.010.

Warne, D.J., Baker, R.E., Simpson, M.J., 2017. Optimal quantification of contact inhibition in cell populations. Biophys. J. 113 (9), 1920–1924. http://dx.doi.org/10.1016/j.bpj.2017.09.016.

Weiss, G.H., Dishon, M., 1971. On the asymptotic behavior of the stochastic and deterministic models of an epidemic. Math. Biosci. 11 (3), 261–265. http://dx.doi.org/10.1016/0025-5564(71)90087-3.

Wilkinson, D.J., 2009. Stochastic modelling for quantitative description of heterogeneous biological systems. Nat. Rev. Genet. 10, 122–133. http://dx.doi.org/10.1038/nrg2509.