



HAL
open science

Adaptive approximate Bayesian computation for complex models

Maxime Lenormand, Franck Jabot, Guillaume Deffuant

► **To cite this version:**

Maxime Lenormand, Franck Jabot, Guillaume Deffuant. Adaptive approximate Bayesian computation for complex models. 2012. hal-00638484v2

HAL Id: hal-00638484

<https://hal.science/hal-00638484v2>

Preprint submitted on 14 Mar 2012 (v2), last revised 20 May 2015 (v4)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Adaptive approximate Bayesian computation for complex models

Maxime Lenormand, Franck Jabot and Guillaume Deffuant

Received: date / Accepted: date

Abstract Approximate Bayesian computation (ABC) is a family of computational techniques in Bayesian statistics. These techniques allow to fit a model to data without relying on the computation of the model likelihood. They instead require to simulate a large number of times the model to be fitted. A number of refinements to the original rejection-based ABC scheme have been proposed, including the sequential improvement of posterior distributions. This technique allows to decrease the number of model simulations required, but it still presents several shortcomings which are particularly problematic for costly to simulate complex models. We here provide a new algorithm to perform adaptive approximate Bayesian computation, and we compare its performance with the ones of three competing algorithms. We show that our new algorithm largely outcompetes previous approaches on both a toy example and a complex social model.

Keywords Approximate Bayesian computation · Population Monte Carlo · Sequential Monte Carlo · Complex model

M. Lenormand
IRSTEA, LISC, 24 avenue des Landais, 63172 AUBIERE,
France
Tel.: +334-73-440617
E-mail: maxime.lenormand@irstea.fr

F. Jabot
IRSTEA, LISC, 24 avenue des Landais, 63172 AUBIERE,
France
Tel.: +334-73-440733
E-mail: franck.jabot@irstea.fr

G. Deffuant
IRSTEA, LISC, 24 avenue des Landais, 63172 AUBIERE,
France
Tel.: +334-73-440614
E-mail: guillaume.deffuant@irstea.fr

1 Introduction

Approximate Bayesian computation is a family of computational techniques in Bayesian statistics. These techniques allow to fit a model to data without relying on the computation of the model likelihood (Beaumont et al., 2002). Namely, if the model produces the observation $x \sim f(x|\theta)$ and $\pi(\theta)$ is the prior distribution on the parameter θ , the ABC algorithm consists in jointly simulating $\theta_i \sim \pi(\theta_i)$ and $x \sim f(x|\theta_i)$ and in accepting the simulated θ_i if, and only if, $\rho(x, y) < \epsilon$ where ρ is a distance measure between the observed data (y) and the simulated one (x), and $\epsilon > 0$ is called a tolerance level. The tolerance level thus represents the level of accuracy in the likelihood approximation. Indeed, the outputs $\{\theta_i\}_{1 \leq i \leq N}$ are distributed with density proportional to $\pi(\theta)pr_\theta\{\rho(x, y) < \epsilon\}$, where $pr_\theta\{z\}$ represents the probability distribution of z for a given parameter θ . This density approximates the model posterior distribution $\pi(\theta)pr_\theta\{f(x|\theta) = y\}$. ABC techniques require the model to be simulated a very large number of times. They hence have been mainly applied to models which can be quickly simulated (Beaumont, 2010). This approach could potentially be very broadly useful for agent-based models where the trajectory of each individual is represented. In such models, it is still difficult to use ABC techniques because of computing time limitations. Approaches to reduce and control this crucial limiting factor are thus required to apply ABC approaches to such complex models.

A number of improvements to the original ABC scheme have been proposed to speed it up. They include the use of local regressions to improve parameter inference (Beaumont et al., 2002; Blum and François, 2010), the coupling to Markov chain Monte Carlo to explore the parameter space (Marjoram et al., 2003)

and the sequential improvement of posterior distributions inspired by sequential Monte Carlo methods (Sisson et al., 2007; Toni et al., 2009; Beaumont et al., 2009). This last class of methods consist in performing the ABC simulations in several steps and in using the simulations of the previous step to build better parameter proposals for the next step. This strategy avoids the areas of low likelihood in the parameter space, by focusing the computing effort in the zones of high likelihood. Beaumont et al. (2009) proposed such an algorithm called Population Monte Carlo ABC (hereafter called PMC) which corrects previous bias in SMC-ABC implementations.

This algorithm presents two shortcomings which are particularly problematic for costly to simulate complex models. First, the sequence of tolerance levels $\{\epsilon_1, \dots, \epsilon_T\}$ has to be provided to the ABC algorithm. In practice, this implies to do preliminary simulations of the model, a step which is computationally costly for complex models. This problem has been solved by Drovandi and Pettitt (2011) and Del Moral et al. (2012) who proposed an on-line construction of the tolerance level, by examining the distance $\rho(x, y)$ of the previously performed simulations. A second shortcoming of the PMC algorithm is that it lacks a criterion to decide whether it has converged. The final tolerance level ϵ_T may be too large for the ABC approach to satisfactorily approximate the posterior distribution of the model. Inversely, a larger ϵ_T may be sufficient to obtain a good approximation of the posterior distribution, hence sparing a number of model simulations. An efficient choice of the tolerance level is particularly difficult, since this choice is case-dependent.

The approaches of Drovandi and Pettitt (2011) and Del Moral et al. (2012) use a MCMC kernel to build the parameter proposal at each algorithm step. Consequently, each retained particle (that is, a set of parameters used in a model simulation) of the previous step can be duplicated, if the MCMC jump in the parameter space is not accepted. This duplication of particles is potentially problematic, in that it decreases the effective number of particles compared to the total number of particles. This decreased number of particles can lead to a decreased quality of the posterior distribution, as we shall see below. To solve this issue, Drovandi and Pettitt (2011) proposed to perform R MCMC jump trials instead of a unique trial, while Del Moral et al. (2012) proposed to resample the particles when the effective number of particles falls below a threshold. Del Moral et al. (2012) also proposed to simulate M times the model for each particle, in order to decrease the variance of the acceptance ratio of the MCMC jump. This can also decrease the rate of particle duplication.

These solutions present the drawback of being potentially costly in terms of numbers of simulations.

In this contribution, we present a modification of the PMC algorithm that we call adaptive population Monte Carlo ABC (hereafter called APMC). In this new algorithm, the sequence of tolerance levels is determined by the algorithm itself as in Drovandi and Pettitt (2011) and Del Moral et al. (2012), and a stopping criterion is provided. Furthermore, contrary to the algorithms of Drovandi and Pettitt (2011) and Del Moral et al. (2012), our approach avoids the problem of particle duplications. We compare our new algorithm with the PMC algorithm of Beaumont et al. (2009), the replenishment SMC ABC algorithm of Drovandi and Pettitt (2011) (hereafter called RSMC) and the adaptive SMC ABC algorithm of Del Moral et al. (2012) (hereafter called SMC). This comparison of the four algorithms is performed both with a toy example and with a complex individual-based social model, the PRIMA model. Based on these two applications, we show that our approach outperforms the three other algorithms, in that it requires considerably less simulations to reach a given quality level of the posterior distribution.

2 Adaptive population Monte-Carlo approximate Bayesian computation

To solve the shortcomings of the PMC algorithm of Beaumont et al. (2009) and the problem of the duplication of particles of the approaches of Drovandi and Pettitt (2011) and Del Moral et al. (2012), we propose a modified PMC algorithm, making use of several ideas proposed by Drovandi and Pettitt (2011) and Del Moral et al. (2012). This new algorithm is presented in Box 1.

INSERT THE BOX HERE

Our new algorithm differs from the PMC algorithm of Beaumont et al. (2009) in four ways. First, the number of simulations $N - N_\alpha$ performed at each time step is controlled, whereas the algorithm of Beaumont et al. (2009) goes on until N particles satisfying the tolerance are simulated. Second, the sequence of tolerance values is determined by the algorithm as the α -quantile of the distances of the N particles to the data at each time step. This automatically generated sequence of tolerance values will be shown below to be more efficient than a sequence determined a priori. Third, the N_α closest to data particles are retained in the following step, in order to make the best use of every costly simulations. Fourth, we define a stopping criterion which evaluates whether the ensemble of N particles has sufficiently changed during the last step. To do this, we

define the proportion p_{acc} of the last step simulations which satisfy the previous tolerance. If this proportion is below an arbitrary value $p_{acc_{min}}$, our algorithm stops.

Our algorithm is also different from the one of [Drovandi and Pettitt \(2011\)](#) and [Del Moral et al. \(2012\)](#) in that it does not use a MCMC kernel. Hence, it does not present the drawback of potentially obtaining duplicated particles, but it requires a reweighting step in $O(N^2)$ instead of $O(N)$ as in [Drovandi and Pettitt \(2011\)](#). Since our goal is to apply our algorithm to complex models, the reweighting step has a negligible computing cost, so that this drawback is negligible in our case.

3 Comparison of the algorithms with a toy example

The code of the different algorithms used in this application are available at ¹.

We consider the toy example studied in [Sisson et al. \(2007\)](#) where $\pi(\theta) = \mathcal{U}_{[-10,10]}$ and $f(x|\theta) \sim \frac{1}{2}\phi(\theta, \frac{1}{100}) + \frac{1}{2}\phi(\theta, 1)$ where $\phi(\mu, \sigma^2)$ is the normal density of mean μ and variance σ^2 . In this example, we consider that $y = 0$ is observed, so that the posterior density of interest is proportional to $(\phi(0, \frac{1}{100}) + \phi(0, 1))\pi(\theta)$.

To compare our algorithm to the PMC, SMC and RSMC algorithms, we use two indicators: the number of simulations performed during the application of the algorithms, and the \mathbb{L}_2 distance between the exact posterior density and the histogram of particle values obtained with the algorithms. This \mathbb{L}_2 distance is computed on the 300-tuple obtained by dividing the support $[-10, 10]$ into 300 equally-sized bins.

In order to compare our algorithm to the PMC, the SMC and the RSMC algorithm we use an ensemble of $N = 5000$ particles and a tolerance level target equal to 0.01. For the PMC algorithm we use a decreasing sequence of tolerance level values from $\epsilon_1 = 2$ down to $\epsilon_{11} = 0.01$. For the SMC algorithm, we use 3 different values for α : $\{0.9, 0.95, 0.99\}$ and $M = 1$ as in [Del Moral et al. \(2012\)](#). For the RSMC algorithm we use $\alpha = 0.5$ as in [Drovandi and Pettitt \(2011\)](#). In our modified algorithm, we also use an ensemble of $N_\alpha = 5000$ particles. We use 9 different values for α : $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$, and 4 different values for $p_{acc_{min}}$: $\{0.01, 0.05, 0.1, 0.2\}$. In each case, we perform 50 times the algorithm, and compute the average and standard deviation of the two indicators: the total number of simulations and the \mathbb{L}_2 distance between the exact posterior density and the histogram

of particle values. We used as kernel transition a normal distribution parameterized with twice the weighted variance of the previous sample, as in [Beaumont et al. \(2009\)](#).

We report below the effects of varying α and $p_{acc_{min}}$ on the performance of our algorithm, and compare it with the PMC, SMC and RSMC algorithms.

3.1 The problem of particle duplication

The number of distinct particles progressively decreases during the course of the SMC algorithm whatever the value of α used (Fig. 1a-b), although a larger α value enables to slow down the decrease in the number of distinct particles (Fig. 1b). The waves are caused by the resampling step in the SMC algorithm (see [Del Moral et al. \(2012\)](#)), but they are not sufficient to counterbalance the overall decrease in the number of distinct particles. This decrease in the number of distinct particles decreases the posterior quality obtained with such approaches compared to methods not based on MCMC kernels, as we will see below (Fig. 2). For the RSMC algorithm, the number of distinct particles is maintained at a reasonably high level (Fig. 1c), but this has a cost in terms of the number of required simulations, as we will see below (Fig. 2). Note that the APMC and the PMC algorithms do not lead to particle duplications.

3.2 Comparison of algorithms

Whatever the value of α and $p_{acc_{min}}$ used, the APMC algorithm always give better results than the other three algorithms, in that it requires between 2 and 8 times less simulations to reach a given posterior quality \mathbb{L}_2 (Fig. 2). Furthermore, the gain in simulation number is progressive during the course of the algorithm so that good approximate posterior distributions are very quickly obtained (Fig. 2). The compromise between simulation speed and convergence level can also be illustrated using the criterion *Number of simulations* $\times \mathbb{L}_2^2$ ([Glynn and Whitt, 1992](#)). This criterion is smaller for the APMC algorithm (Fig. 3a).

3.3 Algorithm parameter study

The value of α and $p_{acc_{min}}$ influence the values of the two indicators studied. We find that smaller α and $p_{acc_{min}}$ values lead to a decrease in the \mathbb{L}_2 distance, and to an increase in the total number of simulations performed during the course of the algorithm, with $p_{acc_{min}}$ having the largest effect (Fig. 2). The value of $p_{acc_{min}}$

¹ http://motive.cemagref.fr/people/maxime.lenormand/script_r_toyex

reflects the level of convergence desired. Lower $p_{acc_{min}}$ values improve the convergence of the algorithm, but with a simulation cost. The value of α has an influence on the rate of decrease of the tolerance. Larger α values induce a smaller number of simulations and a slower decrease in tolerance at each step. But this induces an increased number of steps to reach the desired $p_{acc_{min}}$. In this toy example, our simulations show that all explored sets of $(\alpha, p_{acc_{min}})$ such that $p_{acc_{min}} < 0.1$ give good results for the criterion $Number\ of\ simulations \times \mathbb{L}_2^2$ (Fig. 3b). Large α values provide slightly better results for small $p_{acc_{min}}$ values while small α values provide slightly better results for larger $p_{acc_{min}}$ values (Fig. 3b). We therefore recommend based on this toy example to use intermediate values of α and $p_{acc_{min}}$ ($0.3 \leq \alpha \leq 0.7$ and $0.01 \leq p_{acc_{min}} \leq 0.05$), because such intermediate values present a good compromise between simulation speed and convergence levels.

4 Application to the PRIMA model

In this section, we will illustrate the applicability of our algorithm to complex models, and evaluate whether our algorithm still performs better than the PMC, the RSMC and the SMC approach with such a complex model. We will use an individual-based social model developed during the European project PRIMA². The aim of the model is to simulate the effect of a scenario of job creation (or destruction) on the evolution of the population and activities in a network of municipalities.

4.1 Model and data

The model simulates the dynamics of virtual individuals living in 7 interconnected villages in a rural area of Auvergne (a region of Central France). The dynamics include demographic change (aging, marriage, divorce, births and deaths), activity change (change of jobs, unemployment, inactivity, retirement), and movings from one municipality to another or outside of the set. The model also includes a dynamics of creation / destruction of jobs of proximity services, derived from the size of the local population. More details on the model can be found in [Huet and Deffuant \(2011\)](#). The individuals (about 3000) are initially generated using the 1990 census data of the National Institute of Statistics and Economic Studies (*INSEE*), those who work are given a job type and a location for this job (in a municipality of the set or outside), they are organised in households

living in a municipality of the set. The model dynamics is mostly data driven, but four parameters have to be estimated because they cannot be directly derived from the available data. They are noted θ_p for $1 \leq p \leq 4$, and described in Table 1.

A single run of the PRIMA model with seven rural municipalities takes about 1.4 seconds on a desktop machine (PC Intel 2.83 GHz). We use our algorithm to identify the distribution of the four parameter values for which the simulations, initialized with the 1990 census data, satisfy matching criteria with the data of the 1999 and 2006 census. The set of summary statistics $\{S_m\}_{1 \leq m \leq M}$ and the associated discrepancy measure used ρ_m are described in Table 2. We note S_m the simulated summary statistics and S'_m the observed statistics. The eight summary statistics are normalized (variance equalization) and they are combined using the infinity norm (Eq. 1):

$$\|(\rho_m(S_m, S'_m))_{1 \leq m \leq M}\|_\infty = \sup_{1 \leq m \leq M} \rho_m(S_m, S'_m) \quad (1)$$

We first generate a sample of length N from the prior $\mathcal{U}_{[a,b]}$, where $[a, b]$ is available for each parameter in Table 1, with a Latin hypercube ([Carnell, 2009](#)) and we select the best N_α particles. To move the particles, we use as kernel transition a multivariate normal distribution parameterized with twice the weighted variance-covariance matrix of the previous sample ([Filippi et al., 2011](#)).

As in the section 3, we perform a parameter study and a comparison between our algorithm, the PMC, the SMC and the RSMC algorithm. For our algorithm, we use different values of α ($\{0.3, 0.5, 0.7\}$) and $p_{acc_{min}}$ ($\{0.01, 0.05, 0.1, 0.2\}$), and a sample of $N_\alpha = 5000$ particles. For the PMC, SMC and RSMC we use an ensemble of $N = 5000$ particles and a tolerance level target equal to 1.4. The tolerance value $\epsilon = 1.4$ corresponds to the average final tolerance value we obtain with our algorithm with $p_{acc_{min}} = 0.01$. For the PMC algorithm, we use the decreasing sequence of tolerance levels $\{3, 2.5, 2, 1.7, 1.4\}$. For the SMC algorithm, we use 3 different values for the couple (α, M) : $\{(0.9, 1), (0.99, 1), (0.9, 15)\}$. For the RSMC algorithm we use $\alpha = 0.5$, as in [Drovandi and Pettitt \(2011\)](#). For each algorithm and algorithm parameter values, we perform 5 inference replicates.

Note that in this complex model, we do not know the true posterior density. We approximated this true density with the estimated posterior density obtained with the original rejection-based ABC algorithm. This estimated posterior density is represented by 7890 particles below the tolerance level $\epsilon = 1.4$. It required about 10,000,000 simulations with the original ABC algorithm.

² PRototypical policy Impacts on Multifunctional Activities in rural municipalities - EU 7th Framework Research Programme; 2008-2011; <https://prima.cemagref.fr/the-project>

To compute the \mathbb{L}_2 distance between posterior densities, we divided each parameter support into 4 equally sized bins, leading to a grid of $4^4 = 256$ cells, and we computed on this grid the sum of the squared differences between histogram values.

4.2 PRIMA model fit

Our algorithm leads to a unimodal approximate posterior distribution for the PRIMA model (Fig. 4). Interestingly, parameters θ_1 and θ_4 are slightly correlated (Fig. 4c). This is logical since they have contradictory effects on the number of child in the population. What is less straightforward is that we are able to partly tease apart these two effects with the census data available, since we get a peak in the approximate posterior distribution instead of a ridge.

4.3 Comparison of algorithms

Our algorithm again outperforms the three other algorithms, by requiring between 2 and 7 times less simulations to reach a given posterior quality \mathbb{L}_2 (Fig. 5a). Again, the gain in simulation number is progressive during the course of the algorithm. The *Number of simulations* $\times \mathbb{L}_2^2$ criterion is again smaller for the APMC algorithm (Fig. 5b).

4.4 Algorithm parameter study

As for the toy example, we find that the intermediate values of $(\alpha, p_{acc_{min}})$ that we used lead to similar results (Fig. 5c). In practice, we therefore recommend to use $\alpha = 0.5$ and $p_{acc_{min}}$ between 0.01 and 0.05 depending on the wished level of convergence.

5 Discussion

In this paper we proposed an adaptive approximate Bayesian computation scheme for complex models. This algorithm is a modified version of the population Monte-Carlo algorithm proposed by (Beaumont et al., 2009). We have modified this algorithm to circumvent some limitations in the application of the PMC for complex models. In this modified algorithm, the sequence of tolerance levels is determined by the algorithm itself as in Drovandi and Pettitt (2011) and Del Moral et al. (2012). So we no longer need to predetermine the decreasing sequence of tolerance level, which was a source of inefficiency in the previous algorithm. This algorithm

further enables us to control the number of simulations at each iteration, this number being parameterized by α .

We have also developed a stopping criterion parameterized with $p_{acc_{min}}$ and reflecting the level of convergence of the algorithm. The intuitive basis of this stopping criterion is that we consider that the algorithm has converged when there is not a large enough modification of the particles between two iterations. Indeed we stop the algorithm when the proportion of "accepted" new particles is too low. Our modified algorithm is inspired by the algorithm proposed by Drovandi and Pettitt (2011) but we do not use a MCMC kernel. Consequently, particles are guaranteed to move at each iteration, thus avoiding the problem of particle duplication. We have applied our algorithm to a toy example and to a complex social model. In both cases, our algorithm was 2 to 7 times quicker than the three other algorithms.

Our new algorithm requires to fix two algorithm variables α and $p_{acc_{min}}$. We recommend to use $\alpha = 0.5$ and $p_{acc_{min}}$ between 0.01 and 0.05 depending on the wished level of convergence.

Our adaptive algorithm has been shown to perform well on a complex model involving four parameters and a unimodal posterior distribution. It would be interesting to further evaluate this algorithm on models involving a larger number of parameters and/or multi-modal posterior distributions.

6 Tables

7 Figures

References

- Beaumont, M. A.: *Approximate Bayesian computation in evolution and ecology*. Volume 41 of *Annu. Rev. Ecol. Evol. S.* 379–406 (2010)
- Beaumont, M. A., Cornuet, J., Marin, J., Robert, C. P.: Adaptive approximate Bayesian computation. *Biometrika*. 96(4),983–990 (2009)
- Beaumont, M. A., Zhang, W., Balding, D. J.: Approximate Bayesian computation in population genetics. *Genetics*. 162(4),2025–2035 (2002)

- Blum, M. G. B. François, O.: Non-linear regression models for approximate Bayesian computation. *Stat. Comput.* 20(1),63–73 (2010)
- Carnell, R.: lhs: Latin Hypercube Samples. *R package version 0.5*. (2009)
- Drovandi, C. C., Pettitt, A. N.: Estimation of parameters for macroparasite population evolution using approximate Bayesian computation. *Biometrics* 67(1),225–233 (2011)
- Del Moral, P., Doucet, A., Jasra, A.: An adaptive sequential Monte Carlo method for approximate Bayesian computation. *Stat. Comput.* (In press) (2012)
- Filippi, S., Barnes, C., Stumpf, M. P. H.: On optimal kernel in ABC-SMC. <http://arxiv.org/pdf/1106.6280v3.pdf> (2011)
- Glynn, P.W., Whitt, W.: The asymptotic efficiency of simulation estimators. *Oper. Res.* 40(3),505–520 (1992)
- Huet, S., Deffuant, G.: Common framework for the microsimulation model in prima project. Technical report, Cemagref LISC (2011)
- Marjoram, P., Molitor, J., Plagnol, V., Tavaré, S.: Markov chain Monte Carlo without likelihoods. *P. Natl. Acad. Sci. USA* 100(26),15324–15328 (2003)
- Sisson, S. A., Fan, Y., Tanaka, M. M.: Sequential Monte Carlo without likelihoods. *P. Natl. Acad. Sci. USA* 104(6),1760–1765 (2007).
- Toni, T., Welch, D., Strelkowa, N., Ipsen, A., Stumpf, M. P. H.: Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems. *J. Roy. Soc. Interface* 6,187 (2009)

Adaptive ABC for complex models algorithm

Given α the proportion of particles to keep at each iteration, $p_{acc,min}$ the minimal acceptance rate, and N the initial number of particles,

1. Initialization:

Set $N_\alpha = \lfloor \alpha N \rfloor$

For $i = 1, \dots, N$,

Simulate $\theta_i^{(0)} \sim \pi(\theta)$ and $x \sim f(x|\theta_i^{(0)})$

Set $\rho_i^{(0)} = \rho(x, y)$

Set $w_i^{(0)} = 1$

Let $\epsilon_1 = Q_{\rho^{(0)}}(\alpha)$ the first α -quantile of $\rho^{(0)}$ where $\rho^{(0)} = \{\rho_i^{(0)}\}_{1 \leq i \leq N}$

Let $\{(\theta_i^{(1)}, w_i^{(1)}, \rho_i^{(1)})\} = \{(\theta_i^{(0)}, w_i^{(0)}, \rho_i^{(0)}) | \rho_i^{(0)} \leq \epsilon_1, 1 \leq i \leq N\}$

Take σ_1^2 as twice the weighted empirical variance of $\{(\theta_i^{(1)}, w_i^{(1)})\}_{1 \leq i \leq N_\alpha}$

Set $p_{acc} = 1$

Set $t = 2$

2. While $p_{acc} > p_{acc,min}$

For $i = N_\alpha + 1, \dots, N$,

Pick θ_i^* from $\theta_j^{(t-1)}$ with probability $\frac{w_j^{(t-1)}}{\sum_{k=1}^{N_\alpha} w_k^{(t-1)}}$, $1 \leq j \leq N_\alpha$

Generate $\theta_i^{(t-1)} | \theta_i^* \sim \mathcal{N}(\theta_i^*, \sigma_{(t-1)}^2)$ and $x \sim f(x|\theta_i^{(t-1)})$

Set $\rho_i^{(t-1)} = \rho(x, y)$

Set $w_i^{(t-1)} = \frac{\pi(\theta_i^{(t-1)})}{\sum_{j=1}^{N_\alpha} (w_j^{(t-1)} / \sum_{k=1}^{N_\alpha} w_k^{(t-1)}) \sigma_{t-1}^{-1} \varphi(\sigma_{t-1}^{-1} (\theta_i^{(t-1)} - \theta_j^{(t-1)}))}$

Set $p_{acc} = \frac{1}{N - N_\alpha} \sum_{k=N_\alpha+1}^N \mathbb{1}_{\rho_i^{(t-1)} \leq \epsilon_{t-1}}$

Let $\epsilon_t = Q_{\rho^{(t-1)}}(\alpha)$ where $\rho^{(t-1)} = \{\rho_i^{(t-1)}\}_{1 \leq i \leq N}$

Let $\{(\theta_i^{(t)}, w_i^{(t)}, \rho_i^{(t)})\} = \{(\theta_i^{(t-1)}, w_i^{(t-1)}, \rho_i^{(t-1)}) | \rho_i^{(t-1)} \leq \epsilon_t, 1 \leq i \leq N\}$

Take σ_t^2 as twice the weighted empirical variance of $\{(\theta_i^{(t)}, w_i^{(t)})\}_{1 \leq i \leq N_\alpha}$

Set $t=t+1$

Where $\forall u \in [0, 1]$ and $X = \{x_1, \dots, x_n\}$, $Q_X(u) = \inf\{x \in X | F_X(x) \geq u\}$ and $F_X(x) = \frac{1}{n} \sum_{k=1}^n \mathbb{1}_{x_k \leq x}$.

Where $\varphi(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$

Box 1: Adaptive ABC for complex models algorithm.

Table 1 PRIMA parameter descriptions

| Parameters | Description | Range |
|------------|---|------------|
| θ_1 | Average number of children per woman | $[0, 4]$ |
| θ_2 | Probability to accept a new residence for a household | $[0, 1]$ |
| θ_3 | Probability to make couple for two individuals | $[0, 1]$ |
| θ_4 | Probability to split for a couple in a year | $[0, 0.5]$ |

Table 2 Summary statistic descriptions

| Summary statistic | Description | Measure of discrepancy |
|-------------------|-------------------------------------|-------------------------|
| S_1 | Number of inhabitants in 1999 | \mathbb{L}_1 distance |
| S_2 | Age distribution in 1999 | χ^2 distance |
| S_3 | Household type distribution in 1999 | χ^2 distance |
| S_4 | Net migration in 1999 | \mathbb{L}_1 distance |
| S_5 | Number of inhabitants in 2006 | \mathbb{L}_1 distance |
| S_6 | Age distribution in 2006 | χ^2 distance |
| S_7 | Household type distribution in 2006 | χ^2 distance |
| S_8 | Net migration in 2006 | \mathbb{L}_1 distance |

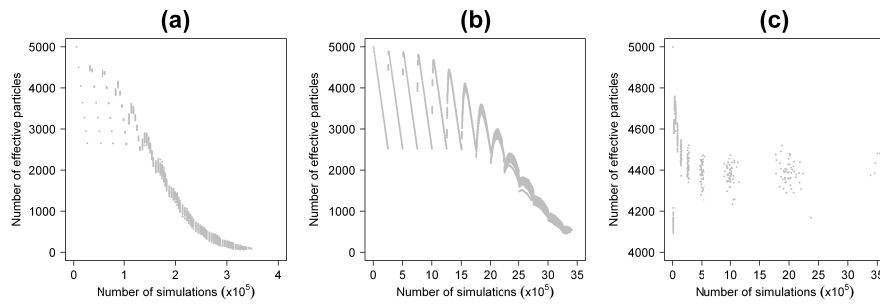


Fig. 1 Number of distinct particles in a sample of $N = 5000$ particles during the course of the SMC and RSMC algorithms applied to the toy example. (a) SMC with $\alpha = 0.9$ and $M = 1$; (b) SMC with $\alpha = 0.99$ and $M = 1$; (c) RSMC with $\alpha = 0.5$. In all three panels, the tolerance target is equal to 0.001.

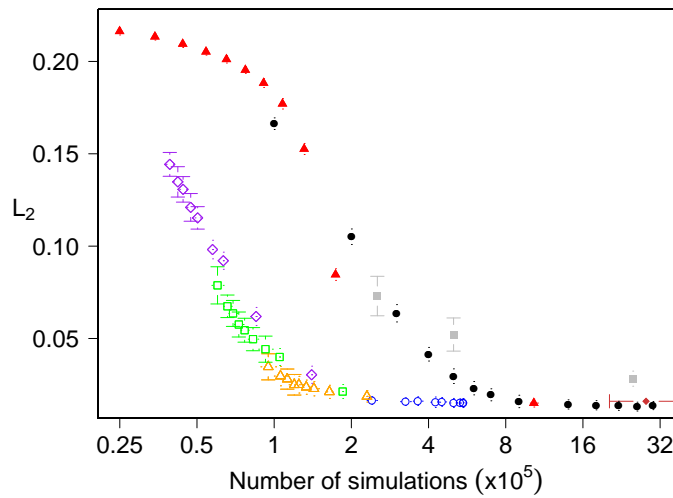


Fig. 2 Posterior quality (L_2) versus computing cost (number of simulations) averaged over 50 replicates. Vertical and horizontal bars represent the standard deviations among replicates. Algorithm parameters used for APMC: α in $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ and $p_{acc_{min}}$ in $\{0.01, 0.05, 0.1, 0.2\}$. Blue circles are used for $p_{acc_{min}} = 0.01$, orange triangles for $p_{acc_{min}} = 0.05$, green squares for $p_{acc_{min}} = 0.1$, and purple diamonds for $p_{acc_{min}} = 0.2$. PMC: red plain triangles for a sequence of tolerance levels from $\epsilon_1 = 2$ down to $\epsilon_{11} = 0.01$. SMC: grey plain square for α in $\{0.9, 0.95, 0.99\}$ (from left to right), $M = 1$ and a ϵ target equal to 0.01. RSMC: brown plain diamond for $\alpha = 0.5$ and a ϵ target equal to 0.01. Results obtained with a standard rejection-based ABC algorithm are depicted with black plain circles.

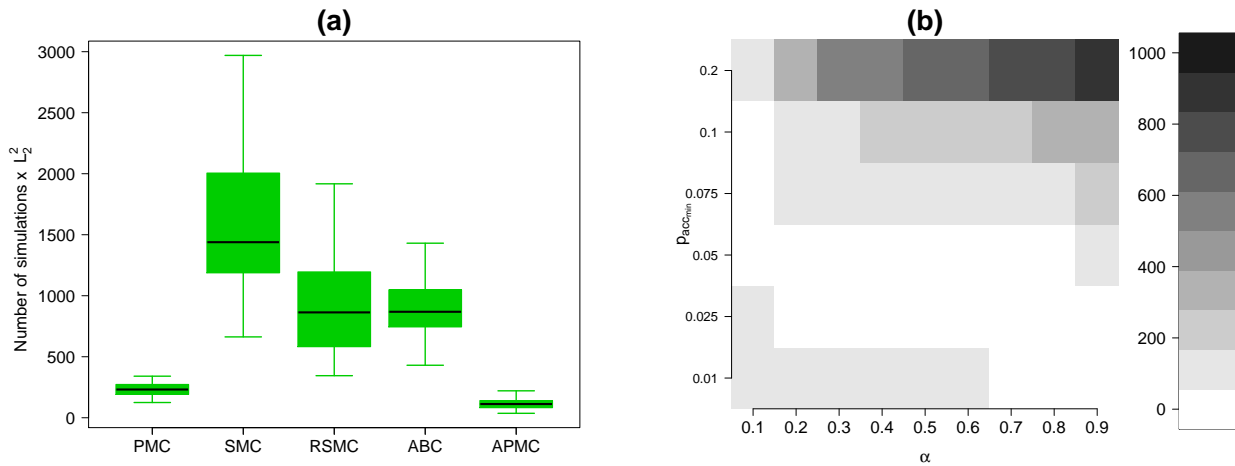


Fig. 3 (a) Boxplot of the criterion “squared \mathbb{L}_2 distance times the number of simulations” for the different ABC algorithms. APMC: for α in $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ and $p_{acc_{min}} = 0.01$; SMC: for α in $\{0.9, 0.95, 0.99\}$, $M = 1$ and a ϵ target equal to 0.01; RSMC: for $\alpha = 0.5$ and a ϵ target equal to 0.01; ABC: for a ϵ target equal to 0.01; PMC: for a sequence of tolerance levels from $\epsilon_1 = 2$ to $\epsilon_{11} = 0.01$. (b) Criterion “squared \mathbb{L}_2 distance times the number of simulations” in the APMC algorithm for the different values of α and $p_{acc_{min}}$. Each cell depicts the average of the criterion over the 50 performed replicates of the APMC.

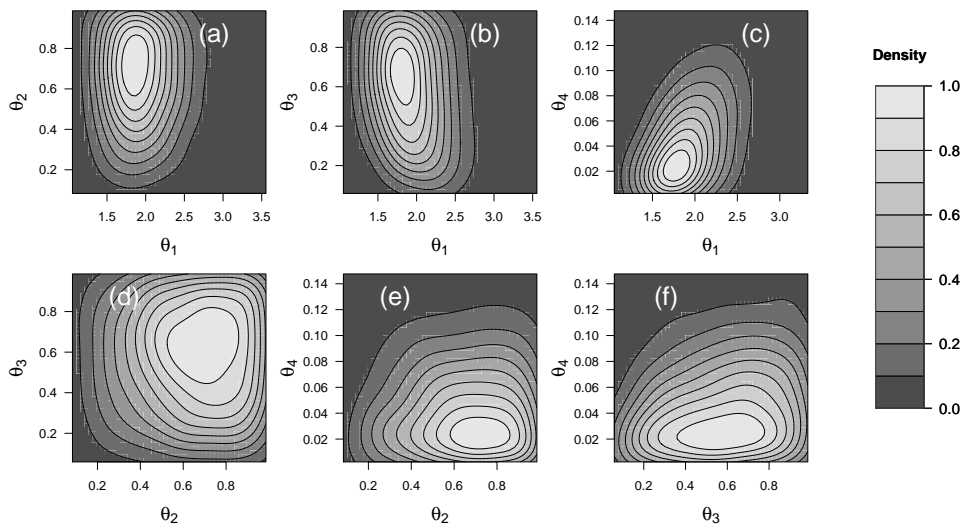


Fig. 4 Contour plot of the bivariate joint densities of θ_i and θ_j obtained with our algorithm, and with $\alpha = 0.5$ and $p_{acc_{min}} = 0.01$; (a) θ_1 and θ_2 ; (b) θ_1 and θ_3 ; (c) θ_1 and θ_4 ; (d) θ_2 and θ_3 ; (e) θ_2 and θ_4 ; (f) θ_3 and θ_4 .

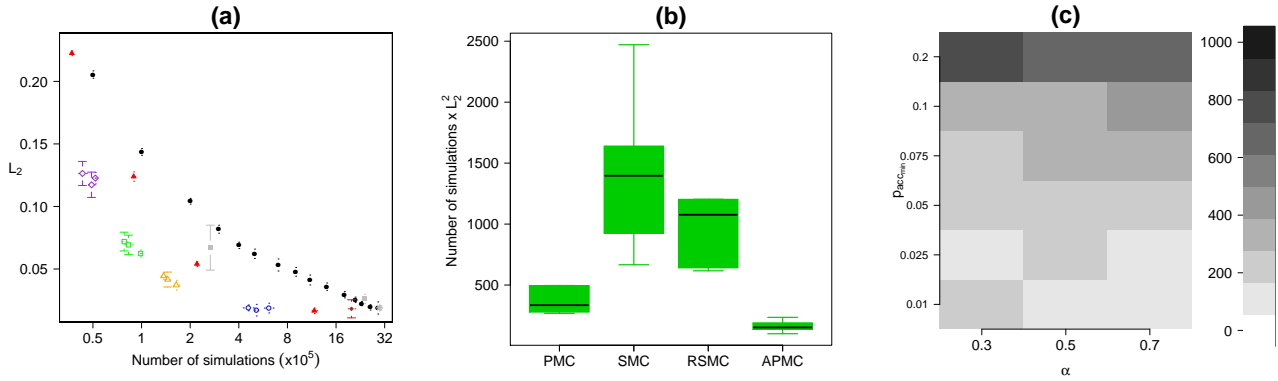


Fig. 5 (a) Posterior quality (\mathbb{L}_2) versus computing cost (number of simulations) averaged over 5 replicates. Vertical and horizontal bars represent the standard deviations among replicates. Algorithm parameters used for APMC: α in $\{0.3, 0.5, 0.7\}$ and $p_{acc_{min}}$ in $\{0.01, 0.05, 0.1, 0.2\}$. Blue circles are used for $p_{acc_{min}} = 0.01$, orange triangles for $p_{acc_{min}} = 0.05$, green squares for $p_{acc_{min}} = 0.1$, and purple diamonds for $p_{acc_{min}} = 0.2$. PMC: red plain triangles for a sequence of tolerance levels from $\epsilon_1 = 3$ to $\epsilon_5 = 1.4$. SMC: grey plain square for (α, M) in $\{(0.9, 1), (0.99, 1)\}$, grey star for $(\alpha, M) = (0.9, 15)$ and a ϵ target equal to 1.4. RSMC: brown plain diamond for $\alpha = 0.5$ and a ϵ target equal to 1.4. Results obtained with a standard rejection-based ABC algorithm are depicted with black plain circles. (b) Boxplot of the criterion “squared \mathbb{L}_2 distance times the number of simulations” for the different algorithms. APMC: for α in $\{0.3, 0.5, 0.7\}$ and $p_{acc_{min}} = 0.01$; SMC: for (α, M) in $\{(0.9, 1), (0.99, 1), (0.9, 15)\}$ and a ϵ target equal to 0.01; RSMC: for $\alpha = 0.5$ and a ϵ target equal to 0.01; ABC: for a ϵ target equal to 1.4; PMC: for a sequence of tolerance levels from $\epsilon_1 = 3$ to $\epsilon_5 = 1.4$. (c) Criterion “squared \mathbb{L}_2 distance times the number of simulations” in the APMC algorithm for the different values of α and $p_{acc_{min}}$. Each cell depicts the average of the criterion over the 5 performed replicates of the APMC.