



**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. 2011. hal-00638484v1

**HAL Id: hal-00638484**

**<https://hal.science/hal-00638484v1>**

Preprint submitted on 4 Nov 2011 (v1), 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, which is shown to perform better on both a toy example and a complex social model.

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

---

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

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

G. Deffuant  
Cemagref, LISC, 24 avenue des Landais, 63172 AUBIERE,  
France  
Tel.: +334-73-440614  
E-mail: guillaume.deffuant@cemagref.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  $\theta$ , 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  $\theta_i$  if, and only if,  $\rho(x, y) < \epsilon$  where  $\rho$  is a distance measure between the observed data ( $y$ ) and the simulated one ( $x$ ), and  $\epsilon > 0$  is called the tolerance. The outputs  $\{\theta_i\}_{1 \leq i \leq N}$  are thus 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  $\theta$ . 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). For many complex agent-based 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 (ABC-MCMC, Marjoram et al., 2003) and the sequential improvement of posterior distributions inspired by sequential Monte Carlo methods (ABC-SMC, Sisson et al., 2007; Toni et al., 2009; Beaumont et al., 2009). This last class of meth-

ods 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 (PMC-ABC) which corrects previous bias in SMC-ABC implementations. This algorithm is presented in Appendix A.

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. Furthermore, a badly chosen sequence of tolerance levels may inflate the number of simulations required to reach a given precision as we will see below. A second shortcoming of the PMC-ABC algorithm is that it lacks a criterion to decide whether it has converged. The final tolerance level  $\epsilon_T$  may be too large for the ABC approach to satisfactorily approximate the posterior distribution of the model. Inversely, a larger  $\epsilon_T$  may be sufficient to obtain a good approximation of the posterior distribution, hence sparing a number of model simulations.

In this contribution, we present a modification of the PMC-ABC algorithm, where the sequence of tolerance levels is determined by the algorithm itself, and where a stopping criterion is provided. We compare this new algorithm to the PMC-ABC algorithm with a toy example. Finally, we apply our new algorithm to a complex individual-based social model, the PRIMA model.

## 2 Modified population Monte-Carlo approximate Bayesian computation

To solve the shortcomings of the PMC-ABC algorithm of Beaumont et al. (2009), we propose a modified algorithm, making use of several ideas proposed by Drovandi and Pettitt (2011). This new algorithm is presented in the box 1.

INSERT THE BOX HERE

Our new algorithm differs from the PMC-ABC 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  $\alpha$ -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  $N_\alpha$  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) in that it does not use a MCMC kernel. Hence, it does not present the drawback of potentially obtaining duplicated particles, but 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, making our algorithm more appropriate in this case.

## 3 Comparison of our modified algorithm to the original PMC-ABC algorithm with a toy example

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  $\mu$  and variance  $\sigma^2$ . 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-ABC one, 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.

We use an ensemble of  $N = 5000$  particles for the PMC-ABC algorithm and a decreasing sequence of tolerance values from  $\epsilon_1 = 2$  down to  $\epsilon_{11} = 0.01$ . In our modified algorithm, we also use an ensemble of  $N_\alpha = 5000$  particles. We use 9 different values for  $\alpha$ :  $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ , 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 report below the effects of varying  $\alpha$  and  $p_{acc_{min}}$  on the performance of our algorithm, and

compare it with the original PMC-ABC algorithm of Beaumont et al. (2009).

### 3.1 Algorithm parameter study

The value of  $\alpha$  and  $p_{acc_{min}}$  have great influences on the two indicators studied. We find that smaller  $\alpha$  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. 1). The value of  $p_{acc_{min}}$  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  $\alpha$  has an influence on the rate of decrease of the tolerance. Larger  $\alpha$  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 there is no optimal set of  $(\alpha, p_{acc_{min}})$  which would decrease the number of simulations required to reach a given level of convergence  $\mathbb{L}_2$ . Indeed, we can observe that all investigated couples are almost on the same decreasing curve, except for the large value  $p_{acc_{min}} = 0.2$  which leads to slightly larger  $\mathbb{L}_2$  values for the same number of simulations (Fig. 1). However, this toy example suggests that intermediate values of  $\alpha$  and  $p_{acc_{min}}$  ( $0.3 \leq \alpha \leq 0.7$  and  $0.01 \leq p_{acc_{min}} \leq 0.05$ ) offer the advantage of leading quickly to good convergence levels (see Fig. 2 for a visualization of the good match of the approximate posterior distributions).

### 3.2 Comparison with the original PMC-ABC algorithm

To compare the  $\mathbb{L}_2$  distances obtained with our modified algorithm and with the original PMC-ABC algorithm, we performed a Student's T-test for each  $\alpha$  value and  $p_{acc_{min}} = 0.01$ . Variance equality and normality of the  $\mathbb{L}_2$  distances were verified (data not shown). For all the  $\alpha$  values, our modified algorithm performed as well as the original PMC-ABC algorithm in terms of  $\mathbb{L}_2$  distance to the true posterior density (Table 1). But to get to the same result quality, our modified algorithm uses between 2 and 4 times less simulations, depending on the value of  $\alpha$  (Table 1). Furthermore, this gain in simulation number is progressive during the course of the algorithm so that good approximate posterior distributions are very quickly obtained (Fig. 1).

## 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 original PMC-ABC approach with such a complex model. We will use an individual-based social model developed during the European project PRIMA<sup>1</sup>. 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)<sup>2</sup>. 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  $\theta_p$  for  $1 \leq p \leq 4$ , and described in Table 2.

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). A run of 1,000,000 simulations, as is customary in standard ABC studies, would then cost around 20 days of computing time. 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  $\rho_m$  are described in Table 3. We note  $S_m$  the simulated summary statistics and  $S'_m$  the observed

<sup>1</sup> PRototypical policy Impacts on Multifunctional Activities in rural municipalities - EU 7th Framework Research Programme; 2008-2011; <https://prima.cemagref.fr/the-project>

<sup>2</sup> Huet, S., Deffuant, G.: Common framework for the microsimulation model in prima project. Technical report, Cemagref LISC (2011)

statistics. The eight summary statistics are normalized using the standard score, and they are combined using the infinity norm (Eq. 1):

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

At the first step we generate a sample of length  $N$  from the prior  $\mathcal{U}_{[a,b]}$ , where  $[a, b]$  is available for each parameter in Table 2, with a latin hypercube (Rob Carnell [rcarnellr@battelle.org](mailto:rcarnellr@battelle.org) (2009). lhs: Latin Hypercube Samples. R package version 0.5.) and we select the best  $N_\alpha$  particles. To move the particles, we use as kernel transition a multivariate normal distribution parametrized with twice the variance-covariance matrix of the previous sample (Filippi et al., 2011).<sup>3</sup>

As in the section 3, we perform a parameter study and a comparison between our algorithm and the PMC-ABC algorithm. For our algorithm, we use different values of  $\alpha$  ( $\{0.3, 0.5, 0.7\}$ ) and  $p_{acc_{min}}$  ( $\{0.01, 0.05, 0.1, 0.2\}$ ), and a sample of  $N_\alpha = 5000$  particles. For the PMC-ABC algorithm we use an ensemble of  $N = 5000$  particles and the decreasing sequence of tolerance values  $\{0.5, 0, -0.5, -1, -1.5, -1.7\}$ . The tolerance value  $\epsilon = -1.5$  corresponds to the one we obtain with our algorithm with  $p_{acc_{min}} = 0.01$ . We added the lower value  $\epsilon = -1.7$  in the sequence of tolerances to evaluate the effect of using a slightly lower final tolerance on the number of simulations performed and on the  $\mathbb{L}_2$  distance. 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 by the average of 20 approximate posterior density;  $3 \times 5 = 15$  approximate posterior densities computed with our algorithm for  $\alpha = \{0.3, 0.5, 0.7\}$  and  $p_{acc_{min}} = 0.01$ ; 5 approximate posterior densities computed with the PMC-ABC algorithm for a final tolerance threshold equal to  $-1.5$ . Thus, what we call in the following the  $\mathbb{L}_2$  distance is the distance to this averaged approximate posterior density. In order to show that there is not a significant difference between the approximated posterior densities obtained in the 4 cases, we have performed 4 ANOVAs. First, we chose a cluster of 5 densities for reference (for example, the 5 approximated posterior densities obtained with the 5 replicates of our algorithm with  $\alpha = 0.5$ ). Second, for each of the three remaining groups ( $\alpha = 0.3$ ,  $\alpha = 0.7$  and the PMC-ABC algorithm) we compute the 25  $\mathbb{L}_2$  distance between the 5 posterior densities of that group and the reference group. Third, we perform an ANOVA

in order to compare the 3 samples. We repeat this process with the 4 reference clusters. For the 4 ANOVAs, there was no statistically significant differences between group means at the 0.05 level.

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 absolute differences between histogram values.

## 4.2 PRIMA model fit

Our algorithm leads to a unimodal approximate posterior distribution for the PRIMA model (Fig. 3). Interestingly, parameters  $\theta_3$  and  $\theta_4$  are correlated (Fig. 3f). This is logical since they have contradictory effects on the number of couples 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 Algorithm parameter study

In this section, we study the influence of  $\alpha$  and  $p_{acc_{min}}$  on the algorithm results. As for the toy example, we have plotted on (Fig. 4) the  $\mathbb{L}_2$  distance against the number of simulations for different values of  $\alpha$  and  $p_{acc_{min}}$ . Again, there is no optimal choice for the couple  $(\alpha, p_{acc_{min}})$ . Larger  $\alpha$  values lead to a slower decrease of the tolerance threshold and of the acceptance rate (Fig. 5). Inversely, the number of step increases with increasing  $\alpha$  values. To illustrate the influence of  $\alpha$  on the posterior density, we plotted on Fig. 6, the average estimated posterior density of the parameter  $\theta_1$  for the 3 different  $\alpha$  values, we obtained almost the same density in each case.

It is important to note that in Fig. 6, the standard deviation among replicates is very small. All replicates of the algorithm thus converge to the same approximate posterior density.

## 4.4 Comparison with the original PMC-ABC algorithm

In the case where for the PMC-ABC algorithm we choose a final tolerance threshold of  $-1.5$  (the one obtained with our algorithm and  $p_{acc_{min}} = 0.01$ ), we obtain similar  $\mathbb{L}_2$  distances with both algorithms but with 3 times less simulations on average for our algorithm compared

<sup>3</sup> Filippi, S., Barnes, C., Stumpf, M. P. H.: On optimal kernel in ABC-SMC (2011) [http://arxiv.org/PS\\_cache/arxiv/pdf/1106/1106.6280v2.pdf](http://arxiv.org/PS_cache/arxiv/pdf/1106/1106.6280v2.pdf)

to the original PMC-ABC approach (Fig. 4). A second advantage of our algorithm is to automatically control the tolerance level. To illustrate this, imagine that an operator had used a larger final tolerance threshold equal to  $-1$  with the original PMC-ABC procedure. This would have led to a relatively bad level of convergence (Fig. 4). More problematically, imagine that an operator had used a slightly smaller final tolerance threshold equal to  $-1.7$ , then he would have had to wait a very long time so that the algorithm stops. Indeed, we stopped the PMC-ABC algorithm after 3,000,000 simulations, and it still had not reached this tolerance level.

Note that we also plotted on Fig.4 the results obtained with the original ABC algorithm. In this case, we note that the PMC-ABC algorithm improves significantly the results obtained with the original ABC algorithm but much less than our algorithm.

## 5 Discussion

In this paper we proposed an adaptive approximate Bayesian computation 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-ABC for complex models. In this modified algorithm, the sequence of tolerance levels is determined by the algorithm itself. So we no longer need to predetermine the decreasing sequence of tolerance threshold, which was a source of inefficiency in the previous algorithm, as illustrated with the PRIMA model. This algorithm further enables us to control the number of simulations at each iteration, this number being parametrized by  $\alpha$ .

We have also developed a stopping criterion parametrized 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 4 times quicker than the original PMC-ABC algorithm.

Our new algorithm requires to fix two algorithm variables  $\alpha$  and  $p_{acc_{min}}$ . We have seen that there is not

an optimal value for this set, although intermediate values ( $0.3 \leq \alpha \leq 0.7$  and  $0.01 \leq p_{acc_{min}} \leq 0.05$ ) provide an interesting trade-off between computational cost and 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.* (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)
- Drovandi, C. C., Pettitt, A. N.: Estimation of parameters for macroparasite population evolution using approximate bayesian computation. *Biometrics*. 67(1),225–233 (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  $\alpha$  the proportion of particles to keep at each iteration and  $p_{acc_{min}}$  the minimal acceptance rate,

## 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  $\alpha$ -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  $\sigma_1^2$  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  $\theta_i^*$  from  $\theta_j^{(t-1)}$  with probability  $w_j^{(t-1)}$ ,  $1 \leq j \leq N_\alpha$

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

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  $\sigma_t^2$  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** Number of simulations, average  $\mathbb{L}_2$  distance (standard deviation in brackets), p-value of the T-test for 50 replicates for different values of  $N$  and  $p_{acc_{min}} = 0.01$ .

$\alpha$	N	Number of Simulations	$\mathbb{L}_2$ distance	T-test p-value
0.9	5555	240320	0.01641(0.00220)	0.12
0.8	6250	325000	0.01581(0.00190)	0.22
0.7	7142	362880	0.01608(0.00278)	0.31
0.6	8333	424958	0.01550(0.00208)	0.32
0.5	10000	450000	0.01565(0.00259)	0.21
0.4	12500	500000	0.01508(0.00203)	0.25
0.3	16666	529970	0.01520(0.00173)	0.56
0.2	25000	545000	0.01502(0.00228)	0.97
0.1	50000	545000	0.01513(0.00297)	0.12

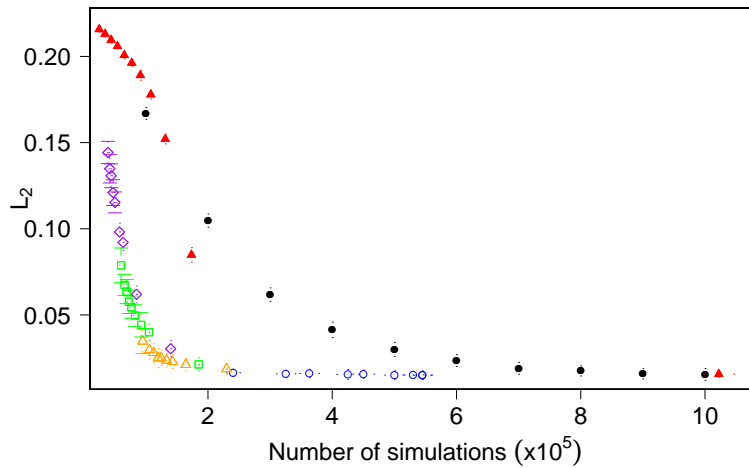
For 50 runs of the PMC-ABC algorithm, we obtained in average 1022195.3 simulations (with a standard deviation of 23650.8) and an average  $\mathbb{L}_2$  distance equal to 0.01566 (with a standard deviation of 0.00189).

**Table 2** PRIMA parameter descriptions

Parameters	Description	Range
$\theta_1$	Average number of children per woman	[0, 4]
$\theta_2$	Probability to accept a new residence for an household	[0, 1]
$\theta_3$	Probability to make couple for two individuals	[0, 1]
$\theta_4$	Probability to split for a couple in a year	[0, 0.5]

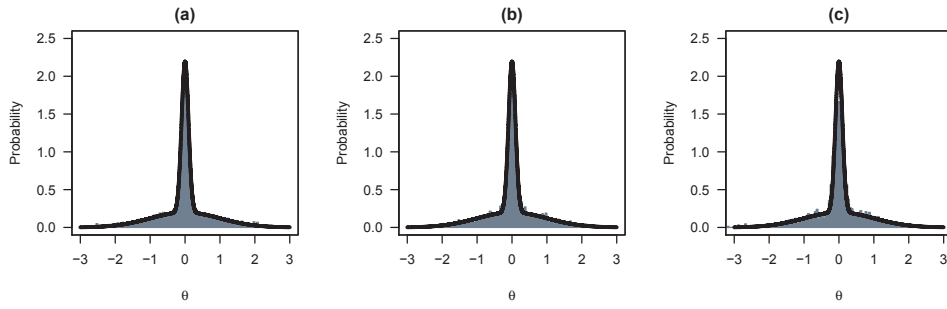
**Table 3** Summary statistic descriptions

Summary statistic	Description	Measure of discrepancy
$S_1$	Number of inhabitants in 1999	$\mathbb{L}_1$ distance
$S_2$	Age distribution in 1999	$\chi^2$ distance
$S_3$	Household type distribution in 1999	$\chi^2$ 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	$\chi^2$ distance
$S_7$	Household type distribution in 2006	$\chi^2$ distance
$S_8$	Net migration in 2006	$\mathbb{L}_1$ distance

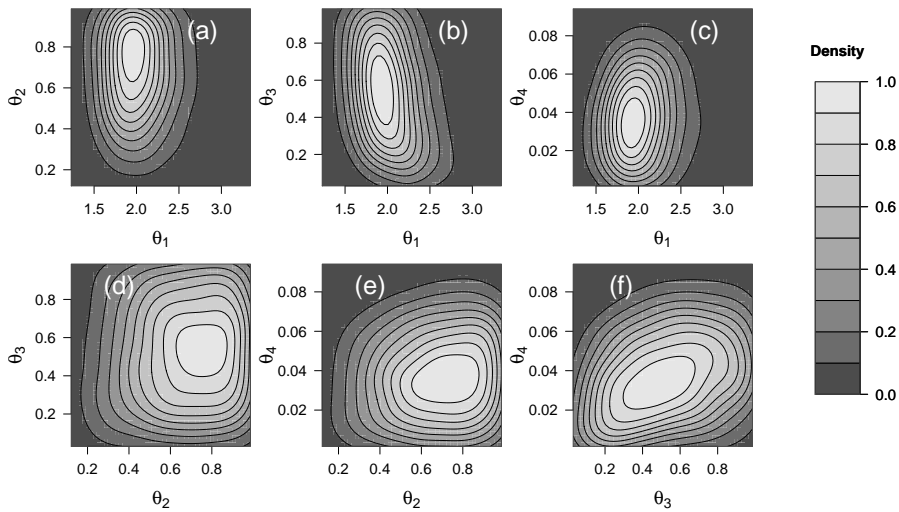


**Fig. 1** Average  $\mathbb{L}_2$  distance as a function of the average number of simulations over 50 replicates for  $\alpha$  in  $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$  and  $p_{acc_{min}}$  in  $\{0.01, 0.05, 0.1, 0.2\}$ . Blue circles are used for  $p_{acc_{min}} = 0.01$ , orange triangles for  $p_{acc_{min}} = 0.05$ , green squares for  $p_{acc_{min}} = 0.1$ , and purple diamonds for  $p_{acc_{min}} = 0.2$ . Results obtained with the original PMC-ABC algorithm are depicted with red plain triangles for a sequence of tolerance thresholds from  $\epsilon_1 = 2$  to  $\epsilon_{11} = 0.01$ . Results obtained with a standard ABC algorithm are depicted with black plain circles. Bars represent the standard deviations among replicates.

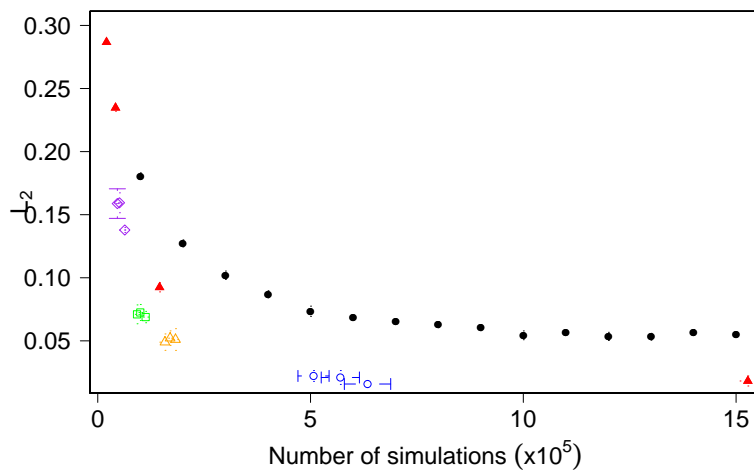




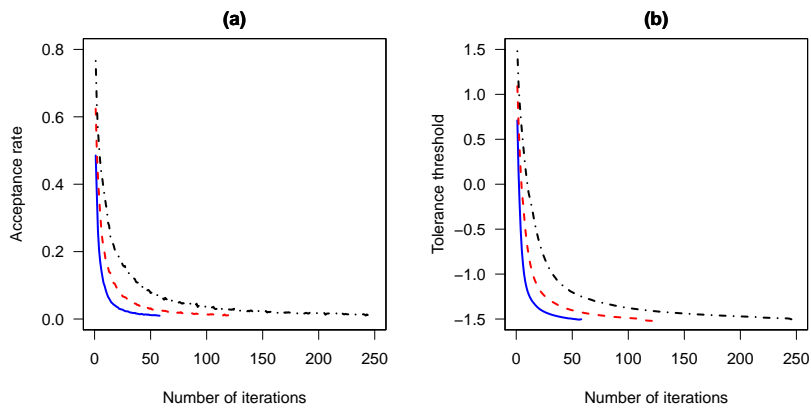
**Fig. 2** Histograms of the samples obtained with the modified PMC-ABC algorithm for  $N_\alpha = 5000$  and  $\alpha = 0.5$ . Columns correspond to different values of  $p_{acc_{min}}$ : (a)  $p_{acc_{min}} = 0.01$ ; (b)  $p_{acc_{min}} = 0.05$ ; (c)  $p_{acc_{min}} = 0.1$ . The exact posterior density is plotted as a full curve.



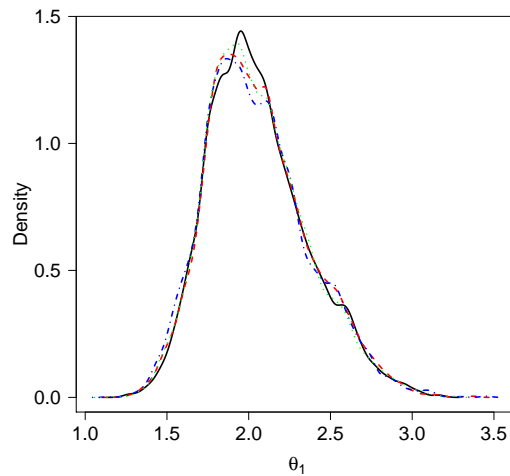
**Fig. 3** Contour plot of the bivariate joint densities of  $\theta_i$  and  $\theta_j$  obtained with our algorithm, and with  $\alpha = 0.5$  and  $p_{acc_{min}} = 0.5$ ; (a)  $\theta_1$  and  $\theta_2$ ; (b)  $\theta_1$  and  $\theta_3$ ; (c)  $\theta_1$  and  $\theta_4$ ; (d)  $\theta_2$  and  $\theta_3$ ; (e)  $\theta_2$  and  $\theta_4$ ; (f)  $\theta_3$  and  $\theta_4$ .



**Fig. 4** Average  $\mathbb{L}_2$  distance as a function of the average number of simulations over 5 replicates for  $\alpha$  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  $p_{acc_{min}} = 0.2$ . Results obtained with the original PMC-ABC algorithm are depicted with red plain triangles for a sequence of tolerance thresholds from  $\epsilon_1 = 0.5$  to  $\epsilon_5 = -1.5$ . Results obtained with a standard ABC algorithm are depicted with black plain circles.



**Fig. 5** Evolution of the average acceptance rate (a) and of the average tolerance threshold (b) as a function of the number of iterations for 5 replicates and  $p_{acc_{min}} = 0.01$ . Results obtained for  $\alpha = 0.3$  are depicted with a blue solid line, for  $\alpha = 0.5$  with a red dashed line, and for  $\alpha = 0.7$  are depicted with a black dotted-dashed line. In each case, standard deviations are too small to be seen.



**Fig. 6** Average approximate posterior density of  $\theta_1$  for 5 replicates of the algorithm for different values of  $\alpha$  and  $p_{acc_{min}} = 0.01$ , and for the original PMC-ABC algorithm. Results obtained for  $\alpha = 0.3$  are depicted with a black solid line, for  $\alpha = 0.5$  with a red dashed line, and for  $\alpha = 0.7$  are depicted with a blue dotted-dashed line. Results obtained with the original PMC-ABC algorithm, with the same final tolerance threshold as with our algorithm, are depicted with a green dotted line. In each case, standard deviations are too small to be seen.

## 8 Appendix A

### PMC-ABC Algorithm

Given a decreasing sequence of tolerance levels  $\epsilon_1 \geq \dots \geq \epsilon_T$ ,

1. At iteration  $t = 1$ ,

For  $i = 1, \dots, N$

Simulate  $\theta_i^{(1)} \sim \pi(\theta)$  and  $x \sim f(x|\theta_i^{(1)})$  until  $\rho(x, y) < \epsilon_1$

Set  $w_i^{(1)} = \frac{1}{N}$

Take  $\sigma_2^2$  as twice the weighted empirical variance of the  $\theta_i^{(1)}$ 's

2. At iteration  $2 \leq t \leq T$ ,

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

Pick  $\theta_i^*$  from  $\theta_j^{(t-1)}$  with probability  $w_j^{(t-1)}$

Generate  $\theta_i^{(t)}|\theta_i^* \sim \mathcal{N}(\theta_i^*, \sigma_t^2)$  and  $x \sim f(x|\theta_i^{(t)})$  until  $\rho(x, y) < \epsilon_t$

Set  $w_i^{(t)} \propto \frac{\pi(\theta_i^{(t)})}{\sum_{j=1}^N w_j^{(t-1)} \sigma_t^{-1} \varphi(\sigma_t^{-1}(\theta_i^{(t)} - \theta_j^{(t-1)}))}$

Take  $\sigma_{t+1}^2$  as twice the weighted empirical variance of the  $\theta_i^{(t)}$ 's

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