

Appendix for Delayed Acceptance ABC-SMC

Richard G. Everitt and Paulina A. Rowińska

31st March 2020

A ABC-MCMC and ABC-SMC

This section derives the ABC-MCMC algorithm (section A.1), and the ABC-SMC of Del Moral et al. (2012) (section A.2).

A.1 ABC-MCMC and early rejection

ABC-MCMC Marjoram et al. (2003) can be derived by applying standard Metropolis-Hastings (MH) to the target

$$\pi(\theta, x | y) \propto p(\theta) l(x | \theta) P_\epsilon(y | x) \quad (1)$$

(omitting the dependence of the target on ϵ to keep the notation consistent throughout the paper). ABC-MCMC uses the proposal $q(\theta^* | \theta) l(x^* | \theta^*)$ on the pair (θ, x) , giving the acceptance probability

$$\alpha((\theta, x), (\theta^*, x^*)) = \frac{p(\theta^*) l(x^* | \theta^*) P_\epsilon(y | x^*)}{p(\theta) l(x | \theta) P_\epsilon(y | x)} \frac{q(\theta | \theta^*) l(x | \theta)}{q(\theta^* | \theta) l(x^* | \theta^*)} \quad (2)$$

$$= \frac{p(\theta^*) P_\epsilon(y | x^*) q(\theta | \theta^*)}{p(\theta) P_\epsilon(y | x) q(\theta^* | \theta)}. \quad (3)$$

The cancellation of the likelihood terms allows this algorithm to be implemented.

A.1.1 Indicator kernels and early rejection

In ABC a common choice for the kernel P_ϵ is $P_\epsilon(y | x) \propto \mathbb{I}(d(y, x) \leq \epsilon)$, where d is a distance metric. We now consider the implications of using this kernel on the ABC-MCMC acceptance probability in the preceding section. Firstly, we must consider the possibility that the denominator in the ratio is equal to 0, due to having $d(y, x) > \epsilon$. The general MH framework of Tierney (1998) (see the main paper) dictates that the acceptance probability in this case should be 0: the practical implications of this are that one must ensure that the initial value of x satisfies $d(y, x) \leq \epsilon$, or the chain will never move from its starting point (in practice instead often different values of (θ, x) are explored until $d(y, x) \leq \epsilon$ is satisfied, with this initial exploration being discarded). Therefore, we may always assume that $d(y, x) \leq \epsilon$ after the chain is initialised and hence the acceptance probability may be written as

$$\alpha((\theta, x), (\theta^*, x^*)) = \begin{cases} \frac{p(\theta^*)q(\theta|\theta^*)}{p(\theta)q(\theta^*|\theta)} & \text{if } d(y, x^*) \leq \epsilon \\ 0 & \text{otherwise} \end{cases}. \quad (4)$$

Let u be the uniformly distributed random number in $[0, 1]$ generated in order implement the accept-reject step. Picchini & Forman (2016) note that a rejection will always occur if

$$u > \frac{p(\theta^*)q(\theta|\theta^*)}{p(\theta)q(\theta^*|\theta)} \quad (5)$$

thus this condition may be checked before x^* is simulated from the likelihood. If the proposed point is not rejected after this first step, x^* is simulated, and the proposal is accepted if $d(y, x^*) \leq \epsilon$. The consequence of this idea is that for θ^* that have a small probability under the prior, we have an “early rejection” that avoids the (potentially expensive) cost of simulation. The computational savings of this approach will only be significant if the posterior is not too different from the prior.

For clarity, pseudo-code for the early rejection method is given in algorithm 1.

Algorithm 1 A single iteration of early rejection ABC-MCMC.

Inputs: Current value of θ .

Outputs: Proposed value θ^* and accept/reject decision for this proposed value.

```

 $u \sim \mathcal{U}(0, 1)$ 
 $\theta^* \sim q(\cdot \mid \theta)$ 
if  $u < 1 \wedge \frac{p(\theta^*)q(\theta \mid \theta^*)}{p(\theta)q(\theta^* \mid \theta)}$  then
     $x^* \sim l(\cdot \mid \theta^*)$ 
    if  $d(y, x^*) < \epsilon$  then
        Accept  $\theta^*$ .
    else
        Reject  $\theta^*$ .
    end if
else
    Reject  $\theta^*$ .
end if

```

A.2 ABC-SMC

Recall from the main paper the weight update used in an SMC sampler (Del Moral et al. 2006) when using a sequence of targets π_t , an MCMC move for the “move” step, and where the SMC sampler backwards kernel is the reverse of this MCMC kernel:

$$\tilde{w}_{t+1}^{(i)} = w_t^{(i)} \frac{\pi_{t+1}(\theta_t^{(i)})}{\pi_t(\theta_t^{(i)})}. \quad (6)$$

The sequence of targets used in the ABC-SMC sampler of Del Moral et al. (2012) is given by

$$\pi_t(\theta_t, x_t \mid y) \propto p(\theta_t) l(x_t \mid \theta_t) P_{\epsilon_t}(y \mid x_t), \quad (7)$$

for $t = 1 : T$. We saw in the previous section that the ABC-MCMC kernel is a valid MCMC kernel targeting $\pi_t(\theta_t, x_t \mid y)$. Choosing the SMC sampler backwards kernel to be

the reverse of this MCMC kernel we obtain

$$\tilde{w}_{t+1} = w_t \frac{p(\theta_t) l(x_t | \theta_t) P_{\epsilon_{t+1}}(y | x_t)}{p(\theta_t) l(x_t | \theta_t) P_{\epsilon_t}(y | x_t)} \quad (8)$$

$$= w_t \frac{P_{\epsilon_{t+1}}(y | x_t)}{P_{\epsilon_t}(y | x_t)}. \quad (9)$$

In the case of indicator kernels, this weight update becomes

$$\tilde{w}_{t+1} = w_t \mathbb{I}(d(y, x_t) < \epsilon_{t+1}),$$

and early rejection may be used in the ABC-MCMC move. Early rejection may provide a significant computational saving in the early stages of the SMC, since the target is likely to be close to the prior.

B Derivation of DA-ABC-MCMC and DA-ABC-SMC

B.1 DA-ABC-MCMC

We present a derivation of DA-ABC-MCMC, using the notation from the main paper. As in the previous sections, the extended state space view of ABC-MCMC is used, where the move is seen to be a Metropolis-Hastings move on the space (θ, x_1) , with θ^* proposed via $q(\cdot | \theta)$ and x_1^* via $l_1(\cdot | \theta^*)$. We may view the move as being on the space (θ, x_1, x_2) , with target proportional to

$$p(\theta^*) l_1(x_1 | \theta) l_2(x_2 | x_1, \theta) P_{\epsilon_2}(y | x_1, x_2)$$

with x_2^* being proposed via $l_2(\cdot | x_1^*, \theta^*)$. We will see that in practice this simulation does not need to be performed at the first stage (this construction is essentially the same as in Sherlock et al. (2017)). The acceptance probability at the first stage is

$$\begin{aligned} \alpha_1 &= \min \left\{ 1, \frac{p(\theta^*) l_1(x_1^* | \theta^*) P_{\epsilon_1}(y_1 | x_1^*) l_2(x_2^* | x_1^*, \theta^*)}{p(\theta) l_1(x_1 | \theta) P_{\epsilon_1}(y_1 | x_1) l_2(x_2 | x_1, \theta)} \frac{q(\theta | \theta^*) l_1(x_1 | \theta) l_2(x_2 | x_1, \theta)}{q(\theta^* | \theta) l_1(x_1^* | \theta^*) l_2(x_2^* | x_1^*, \theta^*)} \right\} \\ &= \min \left\{ 1, \frac{p(\theta^*) P_{\epsilon_1}(y_1 | x_1^*) q(\theta | \theta^*)}{p(\theta) P_{\epsilon_1}(y_1 | x_1) q(\theta^* | \theta)} \right\}. \end{aligned}$$

We observe that the marginal distribution of the target we have used is the ABC posterior with l_1 , ϵ_1 and y_1 .

Using delayed acceptance, as described in the main paper, the acceptance probability at the second stage is

$$\begin{aligned}\alpha_2 &= \min \left\{ 1, \frac{p(\theta^*) l_1(x_1^* | \theta^*) l_2(x_2^* | x_1^*, \theta^*) P_{\epsilon_2}(y | x_1^*, x_2^*)}{p(\theta) l_1(x_1 | \theta) l_2(x_2 | x_1, \theta) P_{\epsilon_2}(y | x_1, x_2)} \frac{p(\theta) l_1(x_1 | \theta) P_{\epsilon_1}(y_1 | x_1) l_2(x_2 | x_1, \theta)}{p(\theta^*) l_1(x_1^* | \theta^*) P_{\epsilon_1}(y_1 | x_1^*) l_2(x_2^* | x_1^*, \theta^*)} \right\} \\ &= \min \left\{ 1, \frac{P_{\epsilon_2}(y | x_1^*, x_2^*) P_{\epsilon_1}(y_1 | x_1)}{P_{\epsilon_2}(y | x_1, x_2) P_{\epsilon_1}(y_1 | x_1^*)} \right\}.\end{aligned}$$

B.2 DA-ABC-SMC

We now justify the weight update for the DA-ABC-SMC method described in the main paper. The target distribution used at iteration t is proportional to

$$p(\theta_t) l_1(x_{1,t} | \theta_t) l_2(x_{2,t} | x_{1,t}, \theta_t) P_{\epsilon_{2,t}}(y | x_{1,t}, x_{2,t}).$$

Using the same approach as in section A.2, we see that the weight update for each particle is given by

$$\begin{aligned}\tilde{w}_{t+1} &= w_t \frac{p(\theta_t) l_1(x_{1,t} | \theta_t) l_2(x_{2,t} | x_{1,t}, \theta_t) P_{\epsilon_{2,t+1}}(y | x_{1,t}, x_{2,t})}{p(\theta_t) l_1(x_{1,t} | \theta_t) l_2(x_{2,t} | x_{1,t}, \theta_t) P_{\epsilon_{2,t}}(y | x_{1,t}, x_{2,t})} \\ &= w_t \frac{P_{\epsilon_{2,t+1}}(y | x_{1,t}, x_{2,t})}{P_{\epsilon_{2,t}}(y | x_{1,t}, x_{2,t})}.\end{aligned}$$

B.2.1 Using indicator kernels

In this section we consider the situation when $P_{\epsilon_{1,t}}$ is chosen to be an indicator function; i.e. $P_{\epsilon_{1,t}}(y_{1,t} | x_{1,t}^*) \propto \mathbb{I}(d(y_{1,t}, x_{1,t}^*) \leq \epsilon_{1,t})$, where d is a distance metric. In this case when specifying our acceptance probabilities we need to account for our target distributions having zero density in some parts of the space. We follow the framework of Tierney (1998), in which for a target $\pi(\theta)$ and proposal q , the MH acceptance probability is written as

$$\alpha = \begin{cases} \min \left\{ 1, \frac{\pi(\theta^*) q(\theta | \theta^*)}{\pi(\theta) q(\theta^* | \theta)} \right\} & (\theta, \theta^*) \in R, \\ 0 & (\theta, \theta^*) \notin R, \end{cases}$$

where $R = \{(\theta, \theta^*) \mid \pi(\theta^*) q(\theta \mid \theta^*) > 0, \pi(\theta) q(\theta^* \mid \theta) > 0\}$. Thus, at the t -th iteration of the SMC the acceptance probability at the first stage of the delayed acceptance is

$$\alpha_{1,t} = \begin{cases} \min \left\{ 1, \frac{p(\theta_t^*) q(\theta_t \mid \theta_t^*)}{p(\theta_t) q(\theta_t^* \mid \theta_t)} \right\} & d(y_{1,t}, x_{1,t}), d(y_{1,t}, x_{1,t}^*) < \epsilon_{1,t}, \\ 0 & \text{otherwise.} \end{cases} \quad (10)$$

As in Picchini & Forman (2016), we may perform the first stage of delayed acceptance in two stages, which we will refer to as steps 1a and 1b, such that some simulations from l_1 may be avoided. At step 1a, θ_t^* is simulated from $q(\cdot \mid \theta_t)$ and an accept-reject step is performed using the acceptance probability

$$\alpha_{1a,t} = \min \left\{ 1, \frac{p(\theta_t^*) q(\theta_t \mid \theta_t^*)}{p(\theta_t) q(\theta_t^* \mid \theta_t)} \right\}.$$

At step 1b, $x_{1,t}^*$ is simulated from $l_1(\cdot \mid \theta_t^*)$ and the entire move $(\theta_t^*, x_{1,t}^*)$ is accepted (to be used in stage 2) with probability

$$\alpha_{1b,t} = \begin{cases} 1 & d(y_{1,t}, x_{1,t}), d(y_{1,t}, x_{1,t}^*) < \epsilon_{1,t} \\ 0 & \text{otherwise} \end{cases}. \quad (11)$$

Splitting the first stage into two substages could itself be seen as a form of delayed acceptance, but its acceptance rate is the same as the single step implementation since it simply uses the fact that $\mathbb{I}(d(y_{1,t}, x_{1,t}^*) < \epsilon_{1,t})$ is either 1 or 0 in order to reorganise the single step calculation in a computationally efficient way.

The acceptance probability at the second stage is

$$\alpha_{2,t} = \begin{cases} 1 & d(y_{1,t}, x_{1,t}), d(y_{1,t}, x_{1,t}^*) < \epsilon_{1,t} \text{ and } d(y, x_{2,t}), d(y, x_{2,t}^*) < \epsilon_{2,t}, \\ 0 & \text{otherwise} \end{cases},$$

which may be seen directly from the description of DA from the main paper with the appropriate choices of π_2 and K_1 . Note that $d(y, x_{2,t}) < \epsilon_{2,t}$ must be true for the particle to have non-zero weight, and $d(y_{1,t}, x_{1,t}), d(y_{1,t}, x_{1,t}^*) < \epsilon_{1,t}$ must be satisfied to reach the

second stage of DA, thus in practice we use

$$\alpha_{2,t} = \begin{cases} 1 & d(y, x_{2,t}^*) < \epsilon_{2,t}, \\ 0 & \text{otherwise.} \end{cases}$$

C Lotka-Volterra model

C.1 Full details of methods

All of our empirical results were generated using R (R Core Team 2019). We study the data “LVPerfect” in the R package `smfsb` (Wilkinson 2018) (the numerical methods for simulating from the likelihood are also taken from this package), previously studied in Wilkinson (2011). In this data the simulation starts with initial populations $X = 50$ and $Y = 100$, and has 30 time units, with the values of X and Y being recorded every 2 time units, resulting in 16 data points in each of the two time series. Our prior follows that in Wilkinson (2011), being uniform in the log domain. Specifically we use

$$p(\log(\theta)) \propto \prod_{i=1}^3 \mathcal{U}(\log(\theta_i) \mid \text{lower} = -6, \text{upper} = 2).$$

Our ABC approach follows that in Wilkinson (2011), Papamakarios & Murray (2016): as summary statistics we use a 9-dimensional vector composed of the mean, log variance and first two autocorrelations of each time series, together with the cross-correlation between them. These statistics were normalised by the standard deviation of the statistics determined by a pilot run, precisely as in Wilkinson (2011). The distance between the summary statistics used in ABC was taken to be the Euclidean norm between the normalised statistic vectors. In all our ABC algorithms we used a final tolerance $\log(\epsilon_2) = \log(0.15) \approx -1.89712$. Reducing the tolerance below this level does not appear to have a large impact on the posterior distribution.

We used DA-ABC-SMC with a variety of choices of U , A and N , and two different choices of the Euler-Maruyama step size s in the cheap simulator $s = 0.5$ and $s = 0.1$, both of which result in very rough approximations of the dynamics. We compared these

approaches with standard ABC-SMC, with $N = 200$ particles and a sequence of tolerances selected such that $U = 100$ unique particles are retained at each iteration, and also “ground truth” for the posterior expectation and standard deviation of the parameters found using a long run (10^5 iterations) of ABC-MCMC. We also compared our approach with a method based on the SMC²-style approach of Duan & Fulop (2015). This approach uses the same SMC-based likelihood estimate (with M particles) as particle MCMC (Wilkinson 2011), but embeds this within an SMC sampler rather than a pseudo-marginal MCMC chain. The sequence of distributions in the “external” SMC sampler is given by raising the likelihood estimate to a power: beginning with 0 and ending with 1 (so that the final distribution is the true posterior). The posterior targeted by this method is the same as in particle MCMC. We used the same model as in the particle MCMC of Wilkinson (2011): specifically we used a normal distribution with mean 0 and standard deviation 10 as the measurement model at each time step. Wilkinson (2011) shows that the posterior obtained using this model has a much smaller standard deviation compared to the one obtained when using ABC, therefore when comparing the SMC² approach with ABC, we only compared the posterior mean (bearing in mind that this is also slightly different between the two cases). In order that the computational cost is comparable with the ABC approaches, we use a DA move within the method of Duan & Fulop (2015). Full details of this method follow.

Algorithm 2 gives the SMC sampler of Duan & Fulop (2015). This method is adapted so that the sequence of powers to which the likelihood estimates are raised is determined adaptively, by using a bisection search to find the power such that the conditional effective sample size (CESS) (Zhou et al. 2016) is αN (where α is a proportion). The CESS is defined as

$$\text{CESS} = \frac{N \left(\sum_{i=1}^N w_{t+1}^{(i)} \omega_t^{(i)} \right)^2}{\sum_{i=1}^N w_{t+1}^{(i)} \left(\omega_t^{(i)} \right)^2}$$

where $\omega^{(i)}$ is the incremental weight for the i th particle (the factor by which we multiply $w_t^{(i)}$ by in order to obtain $\tilde{w}_{t+1}^{(i)}$).

Algorithm 2 The SMC sampler of Duan & Fulop (2015), with adaptation to choose the sequence of distributions.

Inputs: Number of particles N , the proportion α used in the adaptive approach to choosing the sequence of distributions, the proportion β used in resampling, prior p , particle filtering parameters for estimating l (including the number of particles M).

Outputs: Particles $\left\{ \left(\theta_t^{(i)}, x_t^{(i)} \right) \right\}_{i=1}^N$ and weights $\left\{ w_t^{(i)} \right\}_{i=1}^N$ for all t .

for $i = 1 : N$ **do**

$$\theta_0^{(i)} \sim p(\cdot)$$

Run a particle filter to find the likelihood estimate $\hat{l}_0^{(i)}$ at $\theta_0^{(i)}$.

$$w_0^{(i)} = 1/N$$

end for

$$\tau_0 = 0, t = 0.$$

while $\tau_t < 1$ **do**

Use bisection to choose τ_{t+1} s.t. the CESS is αN .

for $i = 1 : N$ **do**

$$\tilde{w}_{t+1}^{(i)} = w_t^{(i)} (\hat{l}_t^{(i)})^{(\tau_{t+1} - \tau_t)}$$

end for

Normalise $\{\tilde{w}_{t+1}^{(i)}\}_{i=1}^N$ to give normalised weights $\{w_{t+1}^{(i)}\}_{i=1}^N$.

Perform resampling if the ESS falls below βN .

for $i = 1 : N$ **do**

$$\theta_{t+1}^{(i)} = \theta_t^{(i)}, \hat{l}_{t+1}^{(i)} = \hat{l}_t^{(i)}$$

$$\left(\theta_{t+1}^{(i)} \right)^* \sim q \left(\cdot \mid \theta_t^{(i)} \right)$$

Run a particle filter to find the likelihood estimate $\left(\hat{l}_{t+1}^{(i)} \right)^*$ at $\left(\theta_{t+1}^{(i)} \right)^*$.

$$u \sim \mathcal{U}(0, 1)$$

if $u < 1 \wedge \frac{p \left(\left(\theta_{t+1}^{(i)} \right)^* \right) \left(\left(\hat{l}_{t+1}^{(i)} \right)^* \right)^{\tau_{t+1}} q \left(\theta_t^{(i)} \mid \left(\theta_{t+1}^{(i)} \right)^* \right)}{p \left(\theta_{t+1}^{(i)} \right) \left(\hat{l}_{t+1}^{(i)} \right)^{\tau_{t+1}} q \left(\left(\theta_{t+1}^{(i)} \right)^* \mid \theta_t^{(i)} \right)}$ **then**

$$\theta_{t+1}^{(i)} = \left(\theta_{t+1}^{(i)} \right)^*, \hat{l}_{t+1}^{(i)} = \left(\hat{l}_{t+1}^{(i)} \right)^*$$

end if

end for

$$t = t + 1$$

end while

When applying algorithm 2 to the Lotka-Volterra data, we found that in order for the SMC to avoid degeneracy (and give a posterior near to the true posterior), it required a configuration (in terms of choosing appropriate N , α and β) that resulted in a computational cost of more than an order of magnitude slower than the ABC approaches. Due to this, we used a delayed acceptance MCMC move in place of the particle MCMC move given in algorithm 2. Algorithm 3 gives the resultant algorithm. In this approach, analogous to our description of DA-ABC-SMC, $l_2 = l$ is the true likelihood to be estimated using a particle filter, and l_1 is a computationally cheap likelihood. Algorithm 3 was the method used in the main paper, with $\beta = 0.5$ in all cases, and different values of N , M and α . We note that the SMC² method of Chopin et al. (2013) was also tried, but was not found to be competitive in terms of the computational effort required to avoid degeneracy.

Algorithm 3 The SMC sampler of Duan & Fulop (2015), with adaptation to choose the sequence of distributions and a DA-MCMC move.

Inputs: Number of particles N , the proportion α used in the adaptive approach to choosing the sequence of distributions, the proportion β used in resampling, prior p , particle filtering parameters for estimating l_1 and $l_2 = l$ (including the number of particles M).

Outputs: Particles $\left\{ \left(\theta_t^{(i)}, x_t^{(i)} \right) \right\}_{i=1}^N$ and weights $\left\{ w_t^{(i)} \right\}_{i=1}^N$ for all t .

for $i = 1 : N$ **do**

$$\theta_0^{(i)} \sim p(\cdot)$$

Run particle filters to find the likelihood estimates $\hat{l}_{1,0}^{(i)}$ and $\hat{l}_{2,0}^{(i)}$ at $\theta_0^{(i)}$.

$$w_0^{(i)} = 1/N$$

end for

$$\tau_0 = 0, t = 0.$$

while $\tau_t < 1$ **do**

Use bisection to choose τ_{t+1} s.t. the CESS is αN .

for $i = 1 : N$ **do**

$$\tilde{w}_{t+1}^{(i)} = w_t^{(i)} (\hat{l}_{2,t}^{(i)})^{(\tau_{t+1} - \tau_t)}$$

end for

Normalise $\{\tilde{w}_{t+1}^{(i)}\}_{i=1}^N$ to give normalised weights $\{w_{t+1}^{(i)}\}_{i=1}^N$.

Perform resampling if the ESS falls below βN .

for $i = 1 : N$ **do**

$$\theta_{t+1}^{(i)} = \theta_t^{(i)}, \hat{l}_{t+1}^{(i)} = \hat{l}_{1,t}^{(i)}$$

$$\left(\theta_{t+1}^{(i)} \right)^* \sim q \left(\cdot \mid \theta_t^{(i)} \right)$$

Run a particle filter to find the likelihood estimate $\left(\hat{l}_{1,t+1}^{(i)} \right)^*$ at $\left(\theta_{t+1}^{(i)} \right)^*$.

$$u_1 \sim \mathcal{U}(0, 1)$$

if $u_1 < 1 \wedge \frac{p \left(\left(\theta_{t+1}^{(i)} \right)^* \right) \left(\left(\hat{l}_{1,t+1}^{(i)} \right)^* \right)^{\tau_{t+1}} q \left(\theta_t^{(i)} \mid \left(\theta_{t+1}^{(i)} \right)^* \right)}{p \left(\theta_{t+1}^{(i)} \right) \left(\hat{l}_{1,t+1}^{(i)} \right)^{\tau_{t+1}} q \left(\left(\theta_{t+1}^{(i)} \right)^* \mid \theta_t^{(i)} \right)}$ **then**

Run a particle filter to find the likelihood estimate $\left(\hat{l}_{2,t+1}^{(i)} \right)^*$ at $\left(\theta_{t+1}^{(i)} \right)^*$.

$$u_2 \sim \mathcal{U}(0, 1)$$

if $u_2 < 1 \wedge \frac{\left(\left(\hat{l}_{2,t+1}^{(i)} \right)^* \right)^{\tau_{t+1}} \left(\hat{l}_{1,t+1}^{(i)} \right)^{\tau_{t+1}}}{\left(\hat{l}_{2,t+1}^{(i)} \right)^{\tau_{t+1}} \left(\left(\hat{l}_{1,t+1}^{(i)} \right)^* \right)^{\tau_{t+1}}}$ **then**

$$\theta_{t+1}^{(i)} = \left(\theta_{t+1}^{(i)} \right)^*, \hat{l}_{1,t+1}^{(i)} = \left(\hat{l}_{1,t+1}^{(i)} \right)^*, \hat{l}_{2,t+1}^{(i)} = \left(\hat{l}_{2,t+1}^{(i)} \right)^*$$

end if

end if

end for

11

$$t = t + 1$$

end while

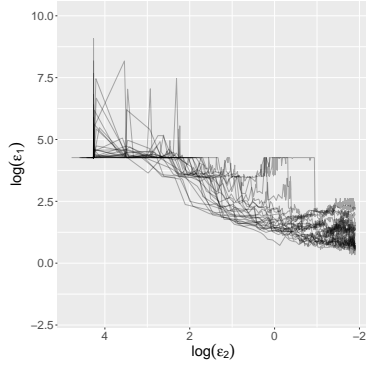
All algorithms were run 30 times, and used an expensive simulator with Euler-Maruyama step size 0.0005 (which resulted in a very accurate approximation), and included the scheme of Picchini & Forman (2016) to avoid simulations from the likelihood where they may be rejected using the prior only. In all approaches the MCMC proposal was Gaussian centred at the current point with variance given by the sample variance of the previous particles. To measure computational cost, we counted the total number of steps \mathbf{S} (taking the median $\bar{\mathbf{S}}$ over the 30 runs) simulated using Euler-Maruyama, taking into account that some simulations were cut short due to the numerical solver diverging (in the implementation in the `smfsb` package, the practical result of this is that after a certain point in the simulation, the size of the populations is assigned “NaN”). When the simulation diverged, both population sizes were assigned to be zero after the time of the divergence.

C.2 Results

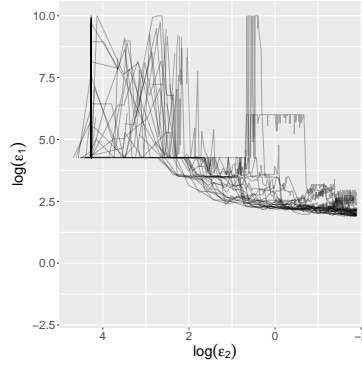
The R packages `ggplot2` (Wickham 2016), `matlab` (Roebuck 2014) and `mvtnorm` (Genz & Bretz 2009) were used when generating the results for this section. Our first observation is with the parameters $N = 200$ and $U = 100$, ABC-SMC sometimes had difficulty converging to the final tolerance. Of the 30 runs, 4 runs were not close to reducing the log tolerance to 0.15 (for some runs this was the case after more than 20,000 SMC iterations). This was not observed for any other approach. In order to present comparisons between ABC-SMC and the other approaches, we focus on median rather than mean simulation times (reported in table 1). We truncated the unfinished runs to 5,000 SMC iterations and treat them as if they had finished, but to provide a fair comparison we also present results where these runs are excluded.

Method	\bar{S}	Med. SMC iter.
DA-ABC-SMC: $N = 500, U = 100, A = 100, s = 0.1$.	1.27×10^9	508
DA-ABC-SMC: $N = 500, U = 100, A = 100, s = 0.5$.	1.08×10^9	231
DA-ABC-SMC: $N = 1000, U = 100, A = 100, s = 0.1$.	8.27×10^8	159.5
DA-ABC-SMC: $N = 1000, U = 100, A = 100, s = 0.5$.	7.04×10^8	159
DA-ABC-SMC: $N = 1000, U = 200, A = 100, s = 0.1$.	5.98×10^9	1423
DA-ABC-SMC: $N = 1000, U = 50, A = 100, s = 0.1$.	2.66×10^8	43.5
DA-ABC-SMC: $N = 5000, U = 100, A = 100, s = 0.1$.	4.78×10^8	65.5
DA-ABC-SMC: $N = 10000, U = 100, A = 100, s = 0.1$.	5.46×10^8	63
SMC ² : $N = 100, M = 100, \alpha = 0.9, s = 0.1$.	7.89×10^8	20
SMC ² : $N = 100, M = 100, \alpha = 0.99, s = 0.1$.	2.53×10^9	62.5
SMC ² : $N = 100, M = 1000, \alpha = 0.9, s = 0.1$.	6.23×10^9	17
ABC-SMC: $N = 200, U = 100$.	2.84×10^9	816.5

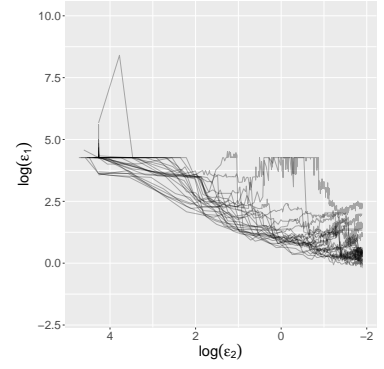
Table 1: Median number \bar{S} of Euler-Maruyama steps and median number of SMC iterations for each method.



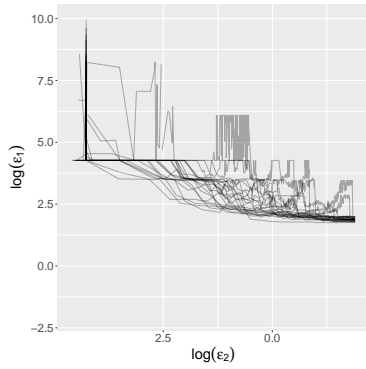
(a) DA-ABC-SMC: $N = 500, U = 100, A = 100, s = 0.1$.



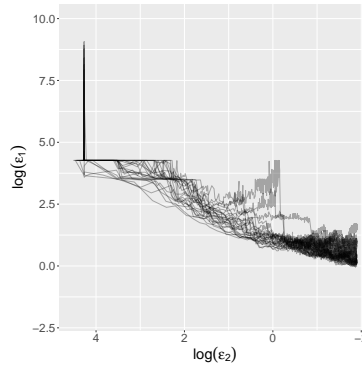
(b) DA-ABC-SMC: $N = 500, U = 100, A = 100, s = 0.5$.



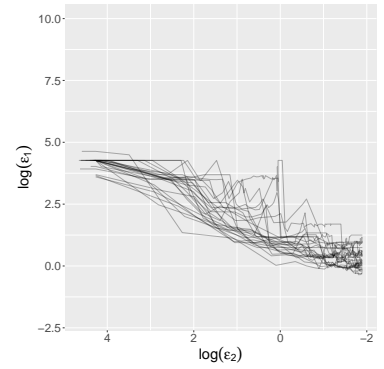
(c) DA-ABC-SMC: $N = 1000, U = 100, A = 100, s = 0.1$.



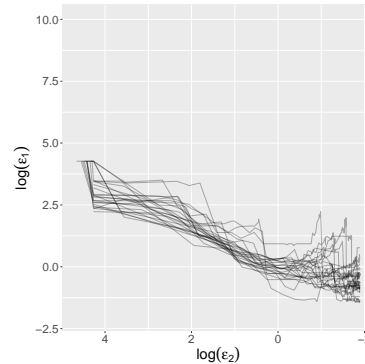
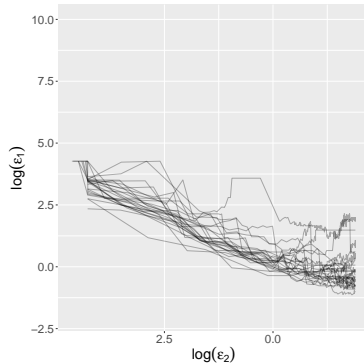
(d) DA-ABC-SMC: $N = 1000, U = 100, A = 100, s = 0.5$.

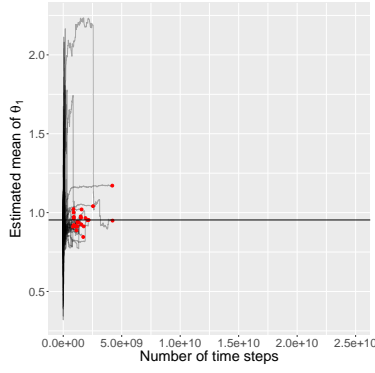


(e) DA-ABC-SMC: $N = 1000, U = 200, A = 100, s = 0.1$.

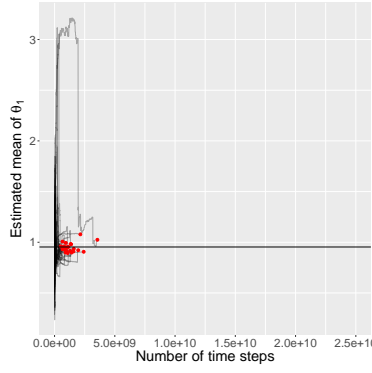


(f) DA-ABC-SMC: $N = 1000, U = 50, A = 100, s = 0.1$.

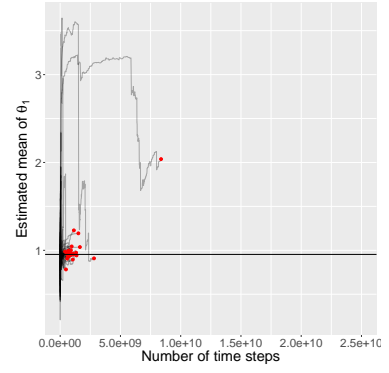




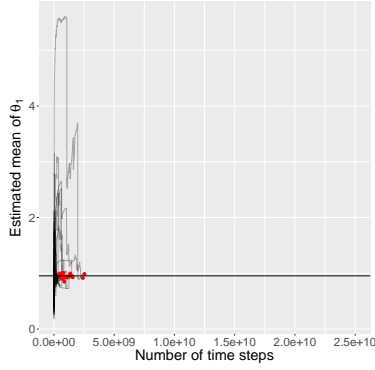
(a) DA-ABC-SMC: $N = 500$, $U = 100$, $A = 100$, $s = 0.1$.



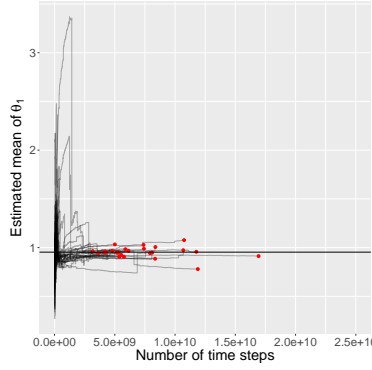
(b) DA-ABC-SMC: $N = 500$, $U = 100$, $A = 100$, $s = 0.5$.



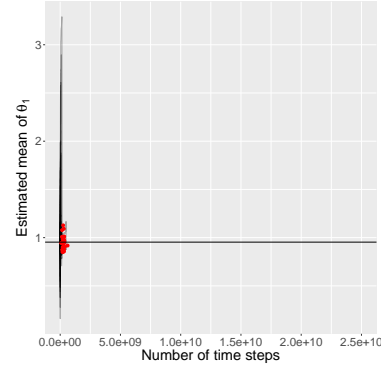
(c) DA-ABC-SMC: $N = 1000$, $U = 100$, $A = 100$, $s = 0.1$.



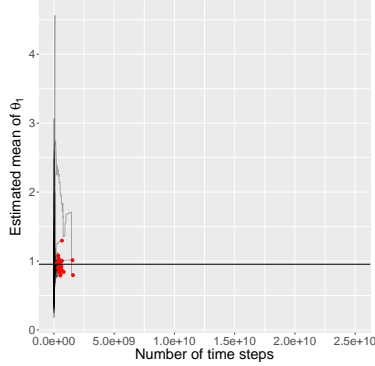
(d) DA-ABC-SMC: $N = 1000$, $U = 100$, $A = 100$, $s = 0.5$.



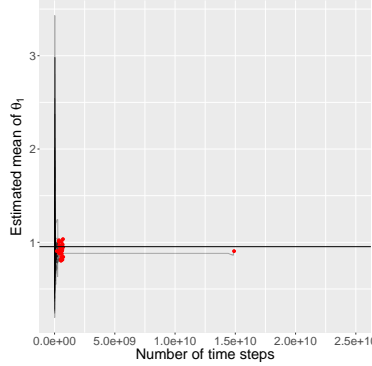
(e) DA-ABC-SMC: $N = 1000$, $U = 200$, $A = 100$, $s = 0.1$.



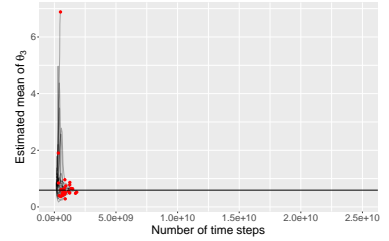
(f) DA-ABC-SMC: $N = 1000$, $U = 50$, $A = 100$, $s = 0.1$.



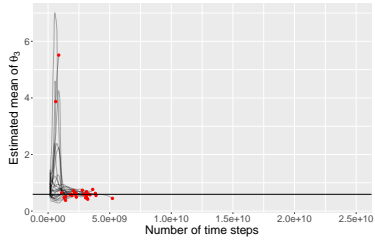
(g) DA-ABC-SMC: $N = 5000$, $U = 100$, $A = 100$, $s = 0.1$.



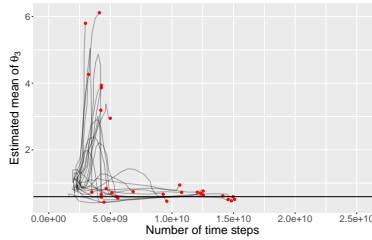
(h) DA-ABC-SMC: $N = 10000$, $U = 100$, $A = 100$, $s = 0.1$.



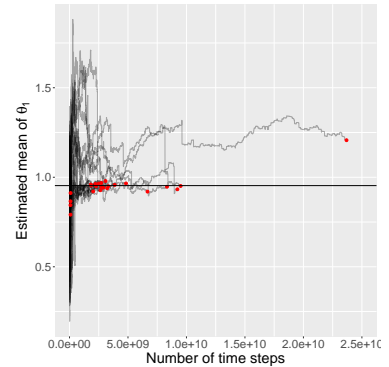
(i) SMC^2 : $N = 100$, $M = 100$, $\alpha = 0.9$, $s = 0.1$.



(j) SMC^2 : $N = 100$, $M = 100$, $\alpha = 0.99$, $s = 0.1$.



(k) SMC^2 : $N = 100$, $M = 1000$, $\alpha = 0.9$, $s = 0.1$.



(l) ABC-SMC: $N = 200$, $U = 100$.

Figure 2: The estimated posterior mean of θ_1 plotted against the total number of time steps used in Euler-Maruyama. Ground truth from ABC-MCMC is marked with a horizontal line.

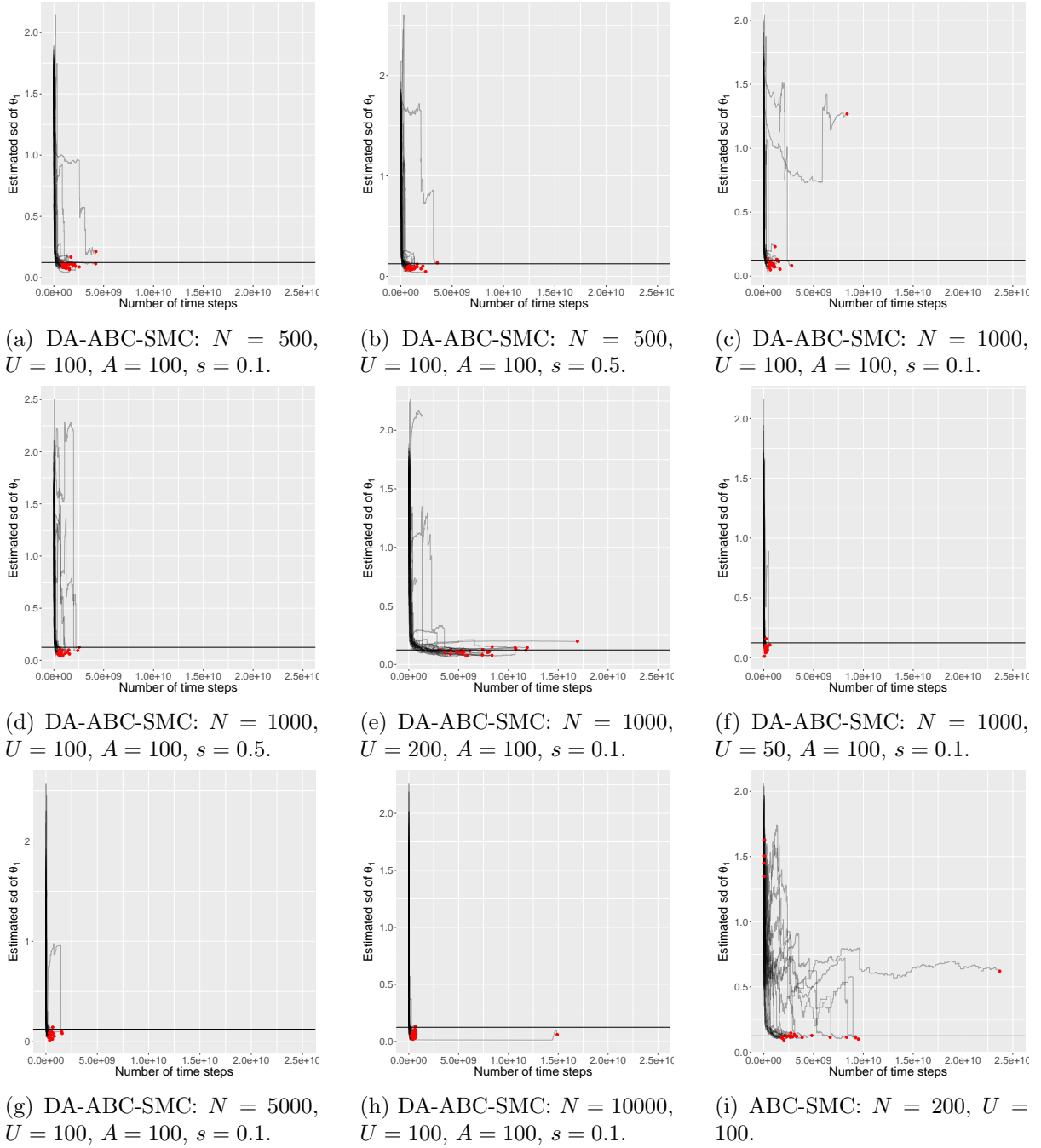
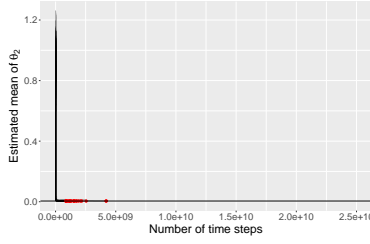
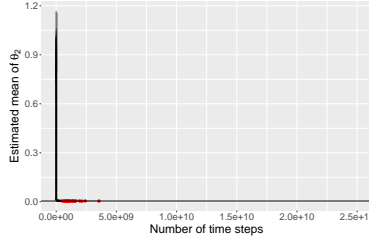


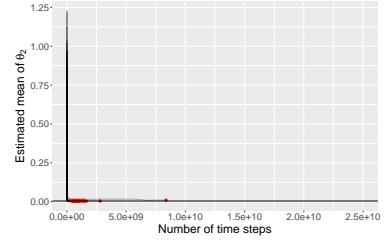
Figure 3: The estimated posterior standard deviation of θ_1 plotted against the total number of time steps used in Euler-Maruyama. Ground truth from ABC-MCMC is marked with a horizontal line.



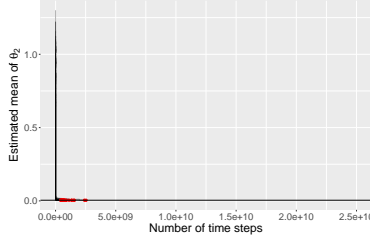
(a) DA-ABC-SMC: $N = 500$, $U = 100$, $A = 100$, $s = 0.1$.



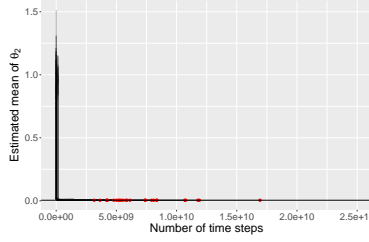
(b) DA-ABC-SMC: $N = 500$, $U = 100$, $A = 100$, $s = 0.5$.



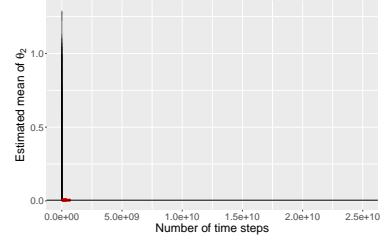
(c) DA-ABC-SMC: $N = 1000$, $U = 100$, $A = 100$, $s = 0.1$.



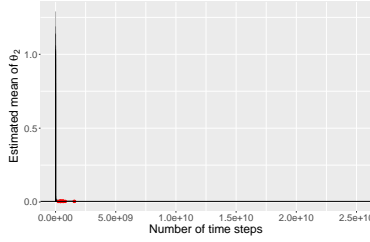
(d) DA-ABC-SMC: $N = 1000$, $U = 100$, $A = 100$, $s = 0.5$.



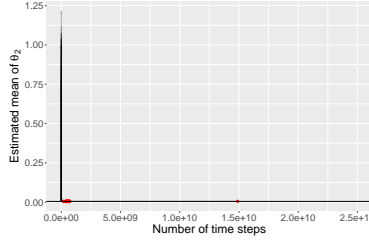
(e) DA-ABC-SMC: $N = 1000$, $U = 200$, $A = 100$, $s = 0.1$.



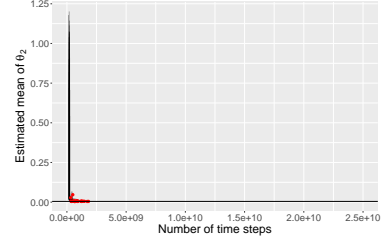
(f) DA-ABC-SMC: $N = 1000$, $U = 50$, $A = 100$, $s = 0.1$.



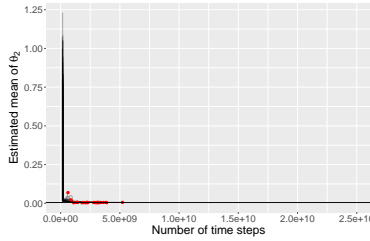
(g) DA-ABC-SMC: $N = 5000$, $U = 100$, $A = 100$, $s = 0.1$.



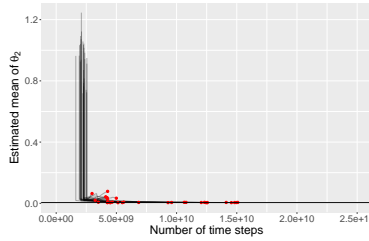
(h) DA-ABC-SMC: $N = 10000$, $U = 100$, $A = 100$, $s = 0.1$.



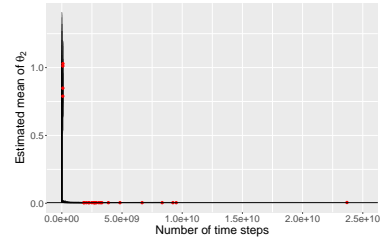
(i) SMC^2 : $N = 100$, $M = 100$, $\alpha = 0.9$, $s = 0.1$.



(j) SMC^2 : $N = 100$, $M = 100$, $\alpha = 0.99$, $s = 0.1$.

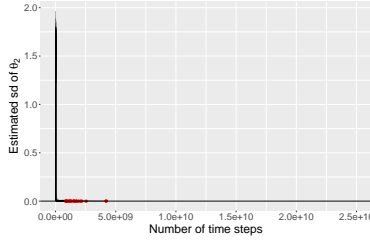


(k) SMC^2 : $N = 100$, $M = 1000$, $\alpha = 0.9$, $s = 0.1$.

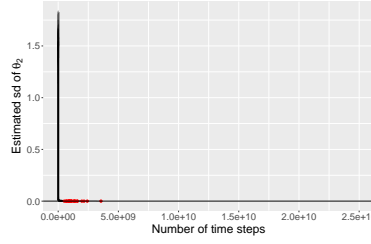


(l) ABC-SMC: $N = 200$, $U = 100$.

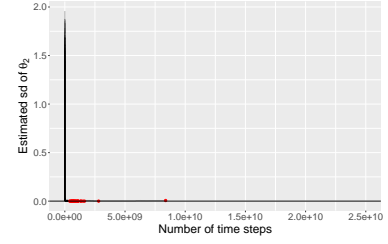
Figure 4: The estimated posterior mean of θ_2 plotted against the total number of time steps used in Euler-Maruyama. Ground truth from ABC-MCMC is marked with a horizontal line.



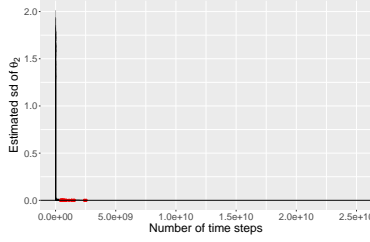
(a) DA-ABC-SMC: $N = 500$, $U = 100$, $A = 100$, $s = 0.1$.



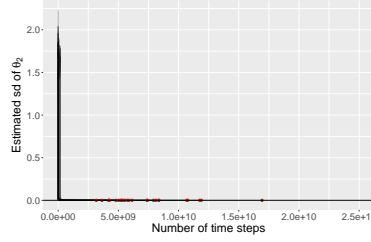
(b) DA-ABC-SMC: $N = 500$, $U = 100$, $A = 100$, $s = 0.5$.



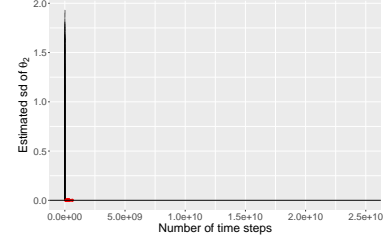
(c) DA-ABC-SMC: $N = 1000$, $U = 100$, $A = 100$, $s = 0.1$.



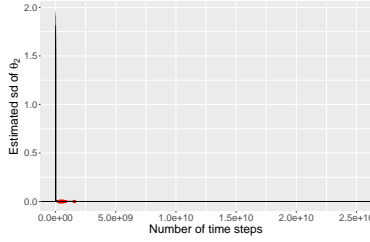
(d) DA-ABC-SMC: $N = 1000$, $U = 100$, $A = 100$, $s = 0.5$.



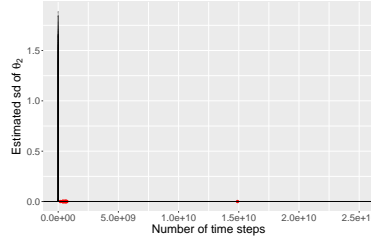
(e) DA-ABC-SMC: $N = 1000$, $U = 200$, $A = 100$, $s = 0.1$.



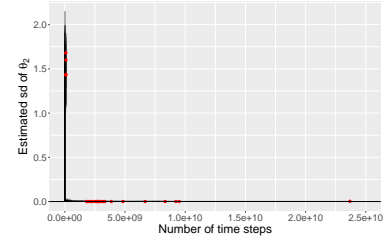
(f) DA-ABC-SMC: $N = 1000$, $U = 50$, $A = 100$, $s = 0.1$.



(g) DA-ABC-SMC: $N = 5000$, $U = 100$, $A = 100$, $s = 0.1$.

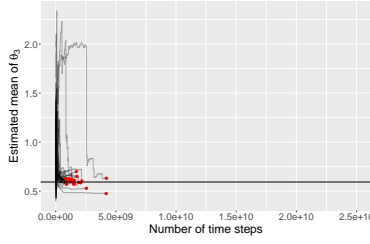


(h) DA-ABC-SMC: $N = 10000$, $U = 100$, $A = 100$, $s = 0.1$.

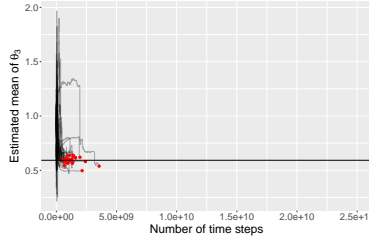


(i) ABC-SMC: $N = 200$, $U = 100$.

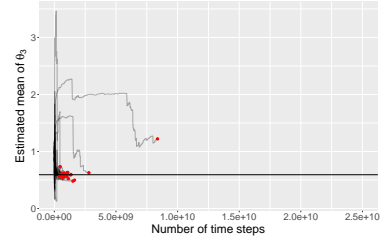
Figure 5: The estimated posterior standard deviation of θ_2 plotted against the total number of time steps used in Euler-Maruyama. Ground truth from ABC-MCMC is marked with a horizontal line.



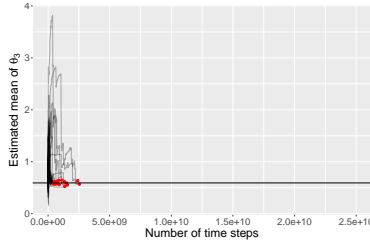
(a) DA-ABC-SMC: $N = 500$, $U = 100$, $A = 100$, $s = 0.1$.



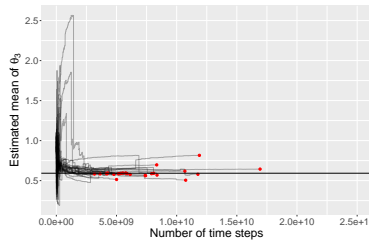
(b) DA-ABC-SMC: $N = 500$, $U = 100$, $A = 100$, $s = 0.5$.



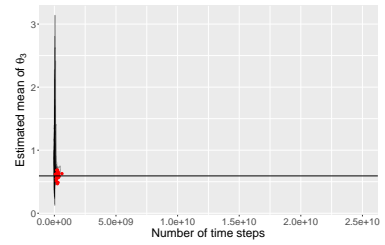
(c) DA-ABC-SMC: $N = 1000$, $U = 100$, $A = 100$, $s = 0.1$.



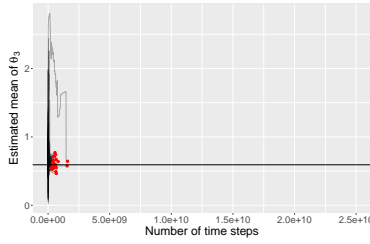
(d) DA-ABC-SMC: $N = 1000$, $U = 100$, $A = 100$, $s = 0.5$.



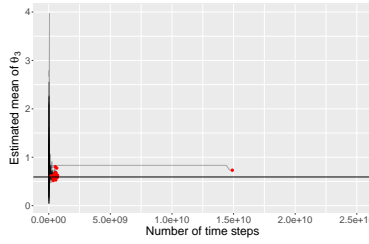
(e) DA-ABC-SMC: $N = 1000$, $U = 200$, $A = 100$, $s = 0.1$.



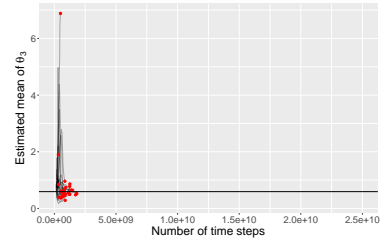
(f) DA-ABC-SMC: $N = 1000$, $U = 50$, $A = 100$, $s = 0.1$.



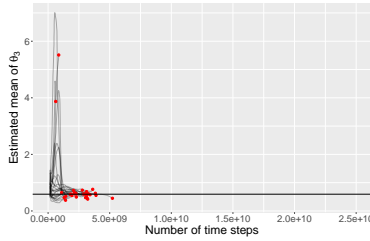
(g) DA-ABC-SMC: $N = 5000$, $U = 100$, $A = 100$, $s = 0.1$.



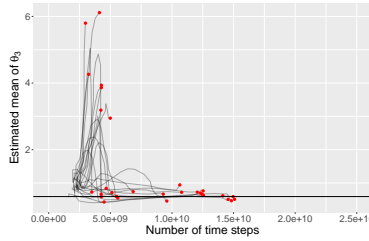
(h) DA-ABC-SMC: $N = 10000$, $U = 100$, $A = 100$, $s = 0.1$.



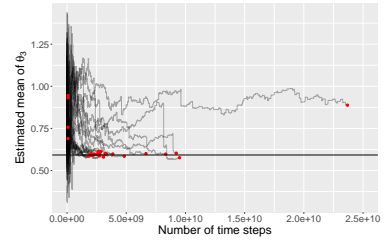
(i) SMC^2 : $N = 100$, $M = 100$, $\alpha = 0.9$, $s = 0.1$.



(j) SMC^2 : $N = 100$, $M = 100$, $\alpha = 0.99$, $s = 0.1$.



(k) SMC^2 : $N = 100$, $M = 1000$, $\alpha = 0.9$, $s = 0.1$.



(l) ABC-SMC: $N = 200$, $U = 100$.

Figure 6: The estimated posterior mean of θ_3 plotted against the total number of time steps used in Euler-Maruyama. Ground truth from ABC-MCMC is marked with a horizontal line.

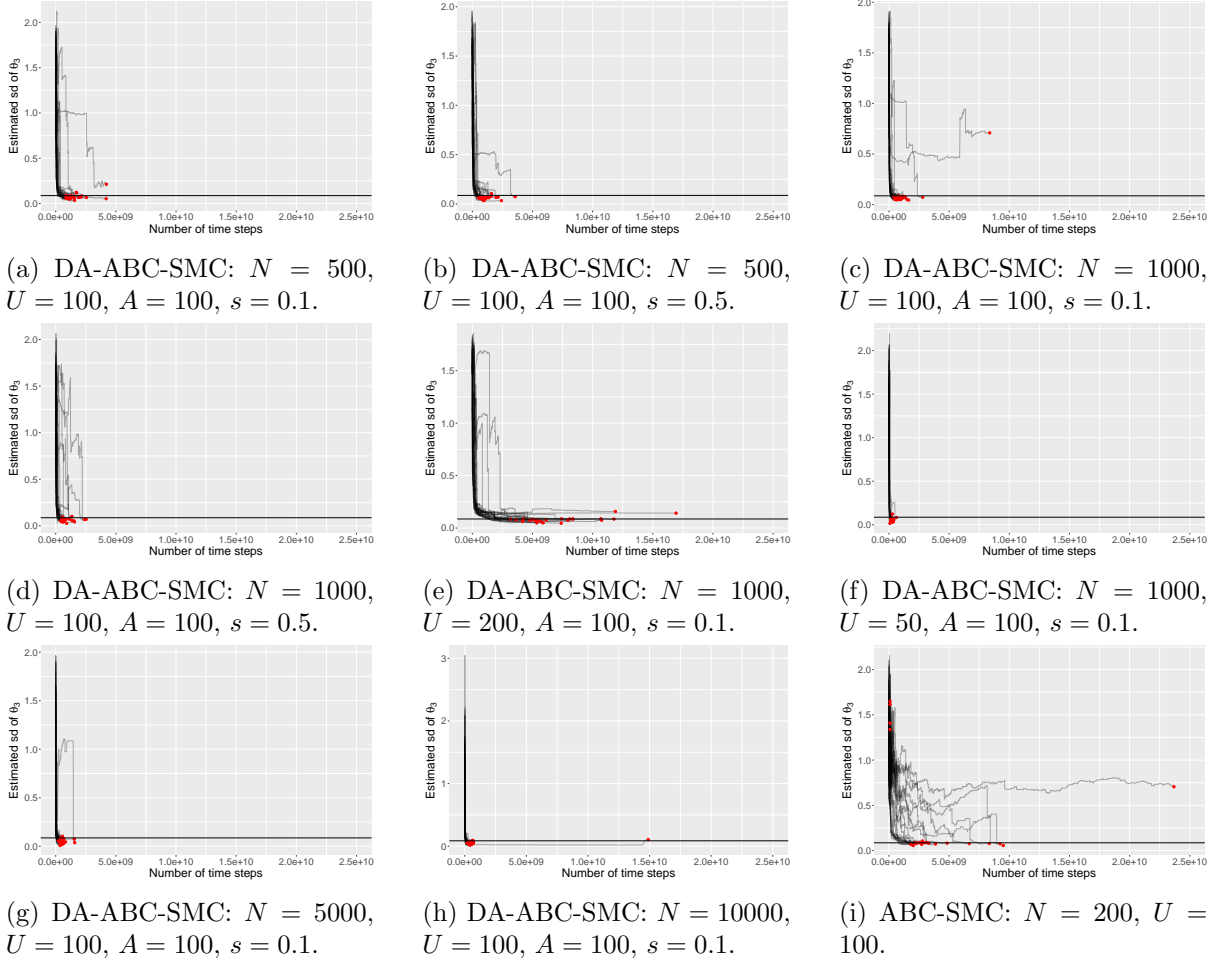


Figure 7: The estimated posterior standard deviation of θ_3 plotted against the total number of time steps used in Euler-Maruyama. Ground truth from ABC-MCMC is marked with a horizontal line.

D Latent exponential random graph model

An exponential random graph model (ERGM) is a model for network data in which the global network structure is modelled as having arisen through local interactions. In this section we consider the situation in which the network is not directly observed, thus x^h is a hidden network made up of a random variable for each edge which takes value 1 if the edge is present and 0 if it is absent, and y is a noisy observation of this network. The ERGM on x^h is

$$l(x^h | \theta_x) \propto \exp(\theta_x^T S(x^h)),$$

with an intractable normalising constant, and our noisy observations are modelled by

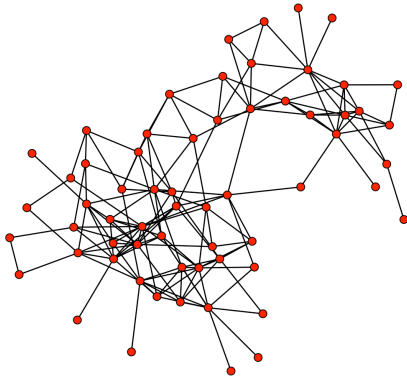
$$g(y_i | x_i^h, \theta_y) \propto \exp(\theta_y (2x_i^h - 1)(2y_i - 1))$$

where the normalising constant is tractable. We studied the Dolphin network (figure 8a) (Lusseau et al. 2003), as also analysed in Caimo & Friel (2011) where the network is treated as directly observed, and used the same summary statistics and priors as in this paper. The **igraph** package (Csardi & Nepusz 2006) was used to load this data into R. We used the statistics

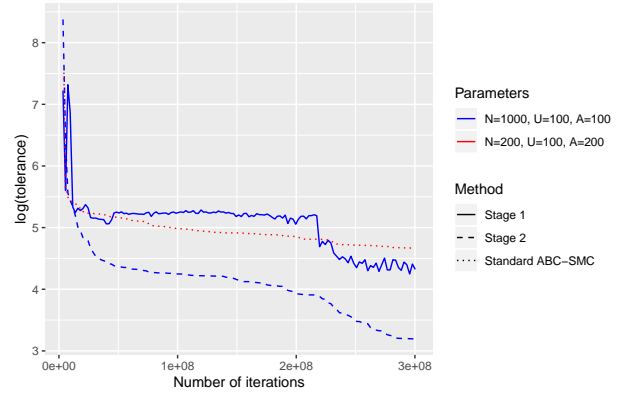
$$\begin{aligned} S_1(x^h) &= \sum_{i < j} x_{ij}^h && \text{the number of edges} \\ S_2(x^h) &= \exp(\phi_u) \sum_{i=1}^{n-1} \left\{ 1 - (1 - \exp(-\phi_u))^i \right\} D_i(x^h) && \text{geometrically weighted degree} \\ S_3(x^h) &= \exp(\phi_v) \sum_{i=1}^{n-2} \left\{ 1 - (1 - \exp(-\phi_v))^i \right\} EP_i(x^h) && \text{geometrically weighted edgewise shared partner} \end{aligned}$$

with $\phi_u = \phi_v = 0.8$, the prior on $\theta_x = (\theta_1, \theta_2, \theta_3)$ and θ_y was $(\theta_1, \theta_2, \theta_3, \theta_y) \sim \mathcal{N}(0, 30I_4)$; and we used the Euclidean distance to compare simulated with observed statistics. The **ergm** package (Hunter et al. 2008) in R was used to simulate from $l(\cdot | \theta_x)$, which uses the “tie no tie” (TNT) sampler and the expensive simulator used 15,000 iterations. Our DA-ABC-SMC algorithm used $U = A = 100$, and again the MCMC proposal was taken to be a Gaussian distribution centred at the current particle, with covariance given by the sample covariance of the particles from the previous iteration.

We ran DA-ABC-SMC for $N = 1,000$, and a cheap simulator having $B = 1,500$ (after exploratory runs suggested that this would be enough iterations to provide a useful DA proposal) and compared the results with standard ABC-SMC with the same configuration as in the previous section (both using 3×10^8 iterations of the TNT sampler). Figure 8b shows the results from the two algorithms, this time showing the sequence $\epsilon_{1,t}$ alongside the sequence $\epsilon_{2,t}$. Again we observe that the tolerance in DA-ABC-SMC reduces faster than standard ABC-SMC, and that the tolerance $\epsilon_{1,t}$ changes adaptively. This data has not previously been studied using a latent ERGM. Using particle MCMC as in Everitt



(a) The Dolphin network.



(b) ABC tolerance plotted against the number of iterations of the TNT sampler.

Figure 8: DA-ABC-SMC applied to the latent ERGM.

(2012) would require at every MCMC iteration to run an SMC sampler to integrate out the latent ERGM space, which consists of 1891 binary edge variables. We might expect that many SMC particles would be required to produce low variance marginal likelihood estimates, leading to a high computational cost. However, the acceptance rate was very low towards the end of our ABC runs, suggesting that a very large computational cost would be required to reduce $\epsilon_{2,t}$ to be close to zero.

References

- Caimo, A. & Friel, N. (2011), ‘Bayesian inference for exponential random graph models’, *Social Networks* **33**, 41–55.
- Chopin, N., Jacob, P. E. & Papaspiliopoulos, O. (2013), ‘SMC²: an efficient algorithm for sequential analysis of state space models’, *Journal of the Royal Statistical Society: Series B* **75**(3), 397–426.
- Csardi, G. & Nepusz, T. (2006), ‘The igraph software package for complex network research’, *InterJournal Complex Systems*, 1695.
URL: <http://igraph.org>
- Del Moral, P., Doucet, A. & Jasra, A. (2006), ‘Sequential Monte Carlo samplers’, *Journal of the Royal Statistical Society: Series B* **68**(3), 411–436.

- Del Moral, P., Doucet, A. & Jasra, A. (2012), ‘An adaptive sequential Monte Carlo method for approximate Bayesian computation’, *Statistics and Computing* **22**(5), 1009–1020.
- Duan, J. C. & Fulop, A. (2015), ‘Density-Tempered Marginalized Sequential Monte Carlo Samplers’, *Journal of Business and Economic Statistics* **33**(2), 192–202.
- Everitt, R. G. (2012), ‘Bayesian Parameter Estimation for Latent Markov Random Fields and Social Networks’, *Journal of Computational and Graphical Statistics* **21**(4), 940–960.
- Genz, A. & Bretz, F. (2009), *Computation of Multivariate Normal and t Probabilities*, Lecture Notes in Statistics, Springer-Verlag, Heidelberg.
- Hunter, D. R., Handcock, M. S., Butts, C. T., Goodreau, S. M. & Morris, M. (2008), ‘ergm: A package to fit, simulate and diagnose exponential-family models for networks’, *Journal of Statistical Software* **24**(3), 1–29.
- Lusseau, D., Schneider, K., Boisseau, O. J., Haase, P., Slooten, E. & Dawson, S. M. (2003), ‘The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations’, *Behavioral Ecology and Sociobiology* **54**(4), 396–405.
- Marjoram, P., Molitor, J., Plagnol, V. & Tavaré, S. (2003), ‘Markov chain monte carlo without likelihoods’, *Proceedings of the National Academy of Sciences of the United States of America* **100**(26), 15324–15328.
- Papamakarios, G. & Murray, I. (2016), ‘Fast Epsilon-Free Inference of Simulation Models with Bayesian Conditional Density Estimation’, *arXiv* (arXiv:1605.06376).
- Picchini, U. & Forman, J. L. (2016), ‘Accelerating inference for diffusions observed with measurement error and large sample sizes using Approximate Bayesian Computation: A case study’, *Journal of Statistical Computation and Simulation* **86**(1), 195–213.
- R Core Team (2019), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria.
- URL:** <https://www.R-project.org/>

- Roebuck, P. (2014), *matlab: MATLAB emulation package*. R package version 1.0.2.
URL: <https://CRAN.R-project.org/package=matlab>
- Sherlock, C., Golightly, A. & Henderson, D. A. (2017), ‘Adaptive, delayed-acceptance MCMC for targets with expensive likelihoods’, *Journal of Computational and Graphical Statistics* **26**(2), 434–444.
- Tierney, L. (1998), ‘A Note On Metropolis-Hastings Kernels For General State Spaces’, *Annals of Applied Probability* **8**(1), 1–9.
- Wickham, H. (2016), *ggplot2: Elegant Graphics for Data Analysis*, Springer-Verlag New York.
URL: <https://ggplot2.tidyverse.org>
- Wilkinson, D. (2018), *smfsb: Stochastic Modelling for Systems Biology*. R package version 1.3.
URL: <https://CRAN.R-project.org/package=smfsb>
- Wilkinson, D. J. (2011), *Stochastic Modelling for Systems Biology*, second edn, Chapman & Hall/CRC Press, Boca Raton, Florida.
- Zhou, Y., Johansen, A. M. & Aston, J. A. D. (2016), ‘Towards automatic model comparison: an adaptive sequential Monte Carlo approach’, *Journal of Computational and Graphical Statistics* **25**(3), 701–726.