Bayesian Analysis , Number , pp.

Supplementary Material for "Finding our Way in the Dark: Approximate MCMC for Approximate Bayesian Methods"

Evgeny Levi * and Radu V. Craiu †

1 A: Steps for implementing the ABSL algorithm

Algorithm 1 describes the steps needed to implement ABSL.

2 B: Details of the Pilot Run

In the case of ABC samplers, before running a MCMC chain, we estimate the initial and final thresholds ϵ_0 and ϵ_{15} (15 equal steps in log scale were used for all models) and the matrix A which is used to calculate the discrepancy $\delta = d(S(\mathbf{y}), s_0) = (S(\mathbf{y}) - s_0)^T A(S(\mathbf{y}) - s_0)$.
To estimate A, we use the following steps:

- Set $A = \mathbf{I}_d$
- Repeat steps I and II below J times (J=3 in our implementations)

  I Generate 500 pairs $\{\zeta_i, \mathbf{y}_i\}_{i=1}^{500}$ from $p(\zeta)f(\mathbf{y}|\zeta)$ and calculate discrepancies $\{\zeta_i, \delta_i\}_{i=1}^{500}$ with $\delta_i = d(S(\mathbf{y}_i), s_0)$

  II Let ζ^* with smallest discrepancy. Finally generate 100 pseudo-data $(\mathbf{y}_1, \ldots, \mathbf{y}_{100})$ from $f(\mathbf{y}|\zeta^*)$, compute corresponding summary statistics $(s_1, \ldots, s_{100})$ and set A to be the inverse of covariance matrix of $(s_1, \ldots, s_{100})$.

We set ϵ_0 to be the 5% quantile of the observed discrepancies. The final ϵ_{15} is obtained by implementing a Random Walk version of Algorithm 3 and decreasing ϵ_0 gradually by setting ϵ_j as the 1% quantile of discrepancies δ corresponding to accepted samples generated between adaption points a_{j-1} and a_j, for $2 \leq j \leq 15$.

The number of simulations was set to 500 and 100 just for computational convenience and is not driven by any theoretical arguments.

*Department of Statistical Sciences, University of Toronto, 700 University Ave, Toronto, M5G 1X6, Canada evgeny.levi@utoronto.ca
†Department of Statistical Sciences, University of Toronto, 700 University Ave, Toronto, M5G 1X6, Canada radu.craiu@utoronto.ca

3 C: Moving Average Model

A popular toy example to check the performances of ABC and BSL techniques is MA2 model:

$$\begin{aligned} z_i &\overset{iid}{\sim} \mathcal{N}(0,1); \quad i = \{-1, 0, 1, \cdots, n\}, \\ y_i &= z_i + \theta_1 z_{i-1} + \theta_2 z_{i-2}; \quad i = \{1, \cdots, n\}. \end{aligned} \tag{1}$$

The data are represented by the sequence $\mathbf{y} = \{y_1, \cdots, y_n\}$. It is well known that Y_i follow a stationary distribution for any θ_1, θ_2, but there are conditions required for the identifiability. Hence, we impose a uniform prior on the following set:

$$\begin{aligned} \theta_1 + \theta_2 &\geq -1, \\ \theta_1 - \theta_2 &\leq 1, \\ -2 \leq \theta_1 &\leq 2, \\ -1 \leq \theta_2 &\leq 2. \end{aligned} \tag{2}$$

It is very easy to see that the joint distribution of $\mathbf{y}$ is multivariate Gaussian with mean 0, diagonal variances $1 + \theta_1^2 + \theta_2^2$, covariance at lags 1 and 2, $\theta_1 + \theta_1\theta_2$ and θ_2 respectively and zero at other lags. In this case, the (Exact) sampling is feasible. For simulations we set $\{\theta_1 = 0.6, \theta_2 = 0.6\}$, $n = 200$ and define the summary statistics $S(\mathbf{y}) = (\hat{\gamma}_0(\mathbf{y}), \hat{\gamma}_1(\mathbf{y}), \hat{\gamma}_2(\mathbf{y}))$ as the sample variance and the covariances at lags 1 and 2. First we show the results based on one replicate. Figure 1 shows the trace plots, histograms and auto-correlation functions estimated from the posterior draws for parameters θ_1 and θ_2 for the AABC-U sampler. Note that only post burn-in samples are shown. Similarly, Figure 2 and Figure 3 display the behaviour of ABSL-U sampler and standard ABC-RW, respectively. From these plots it is apparent that the proposed AABC-U and ABSL-U have much better mixing than ABC-RW. In the interest of keeping the paper length within reasonable limits, we briefly mention that additional simulations suggest that AABC-L is similar to AABC-U and ABSL-L to ABSL-U, while ABC-IS is outperformed by ABC-RW.

In order to summarize and compare the information in the MCMC draws produced by the approximated samplers and the exact chain, we plot the estimated densities in Figure 4. The left and right side plots refer to θ_1 and θ_2, respectively. The two upper plots compare the estimated density of the exact MCMC sampler with ABC-based ones (SMC, ABC-RW and AABC-U), while the two lower plots compare the exact sampler with Synthetic Likelihood based methods (BSL-IS and ABSL-U).

The posterior distributions evaluated from AABC-U is very similar to those produced by SMC and ABC-RW, but all are distinct from the Exact one. This latter difference may be due to the loss of information incurred when the posterior is conditional on a non-sufficient statistic. Similarly, the distribution produced by ABSL-U draws is very close to that of BSL-IS. These observations hold for both components, θ_1 and θ_2.
To study the accuracy, precision and efficiency of the proposed samplers we perform a simulation study where 100 data sets are generated and all samplers are run for every

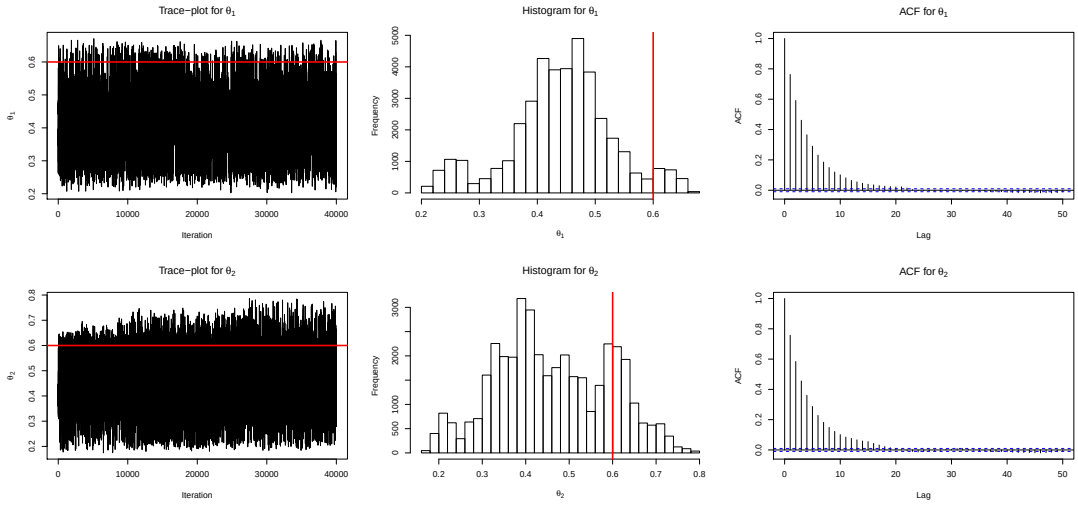

Figure 1: MA2 model: AABC-U Sampler. Each row corresponds to parameters θ_1 (top row) and θ_2 (bottom row) and shows in order from left to right: Trace-plot, Histogram and Auto-correlation function. Red lines represent true parameter values.

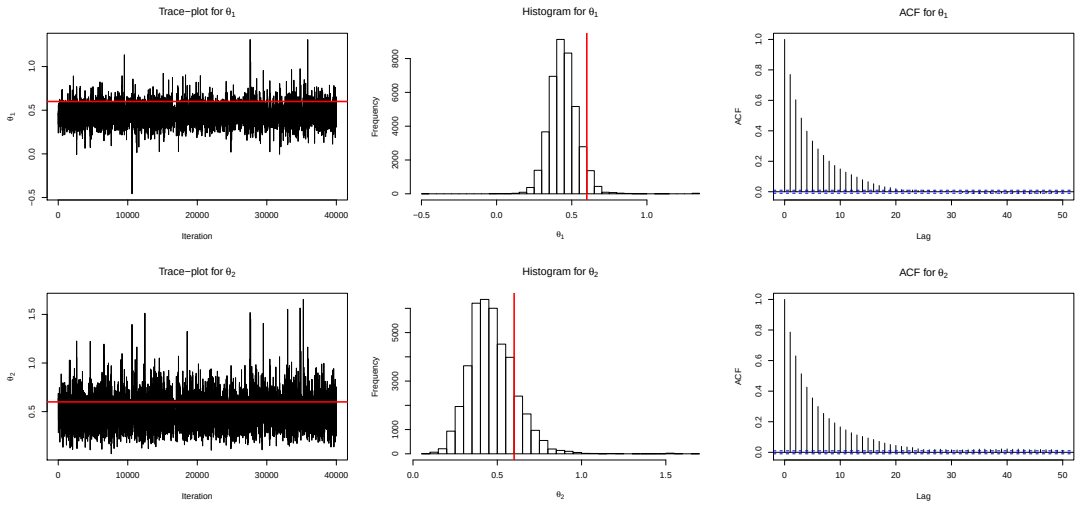

Figure 2: MA2 model: ABSL-U Sampler. Each row corresponds to parameters θ_1 (top row) and θ_2 (bottom row) and shows in order from left to right: Trace-plot, Histogram and Auto-correlation function. Red lines represent true parameter values.

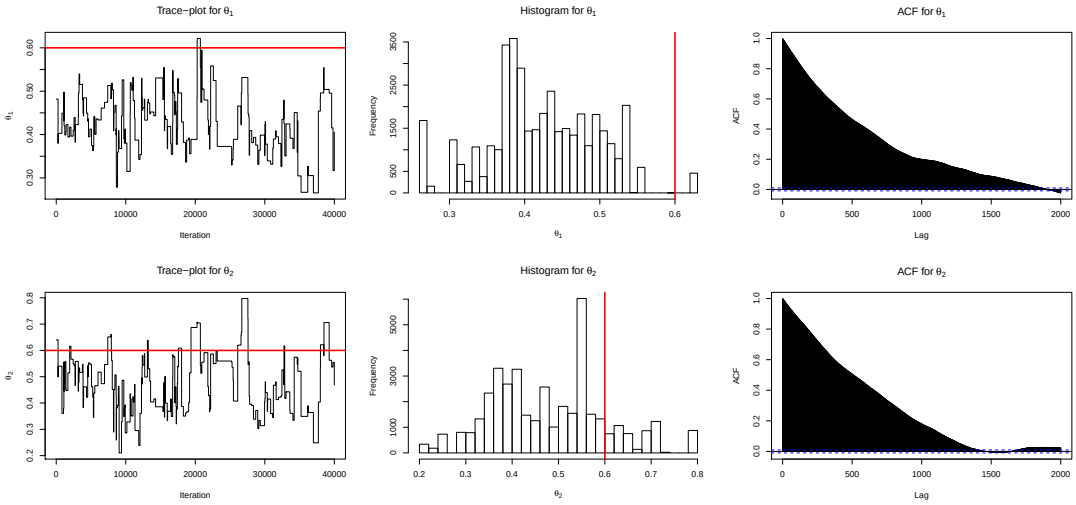

Figure 3: MA2 model: ABC-RW Sampler. Each row corresponds to parameters θ_1 (top row) and θ_2 (bottom row) and shows in order from left to right: Trace-plot, Histogram and Auto-correlation function. Red lines represent true parameter values.

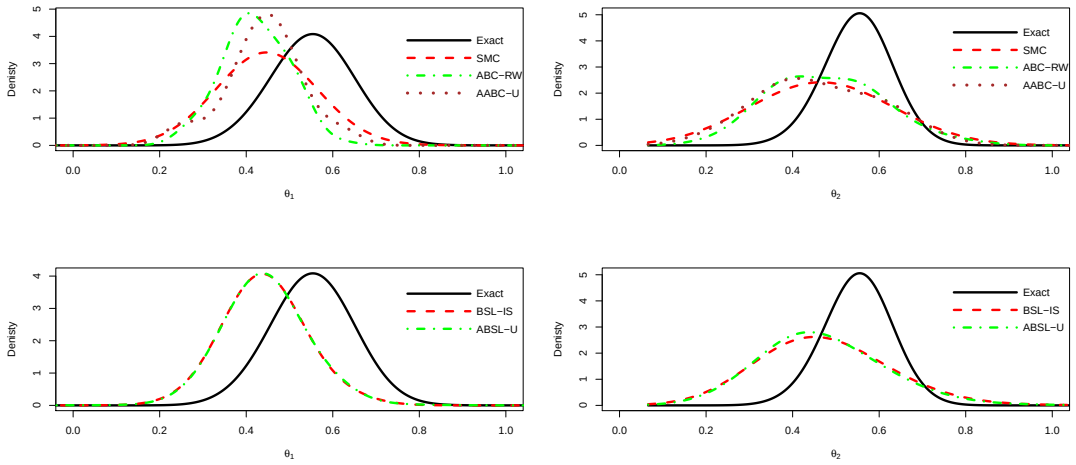

Figure 4: MA model: Estimated densities for each component. First row compares Exact, SMC, ABC-RW and AABC-U samplers. Second row compares Exact, BSL-IS and ABSL-U. Columns correspond to parameter's components, from left to right: θ_1 and θ_2.

Table 1: Simulation Results (MA model): Average Difference in Mean, Difference in Covariance, Total Variation, square roots of Bias and MSE, Effective Sample Size per CPU time, Variances of Lower/Upper Quantiles and Mean times CPU time for every sampling algorithm.

	Diff with Exact			Diff with True Par		Efficiency			
Sampler	DIM	DIC	TV	$\sqrt{\text{Bias}^2}$	$\sqrt{\text{MSE}}$	ESS/cpu	$V_{QL} * \text{cpu}$	$V_{QU} * \text{cpu}$	$V_{E} * \text{cpu}$
SMC	0.082	0.005	0.418	0.014	0.116	-	0.040	0.052	0.009
ABC-RW	0.088	0.006	0.466	0.016	0.124	0.231	0.040	0.018	0.029
ABC-IS	0.084	0.007	0.455	0.016	0.116	0.389	0.083	0.013	0.040
AABC-U	0.083	0.007	0.444	0.018	0.117	6.215	0.007	0.009	0.003
AABC-L	0.080	0.007	0.438	0.017	0.113	5.107	0.004	0.011	0.013
BSL-RW	0.082	0.007	0.438	0.015	0.115	0.282	0.092	0.492	0.046
BSL-IS	0.081	0.007	0.436	0.015	0.115	0.923	0.035	0.109	0.010
ABSL-U	0.081	0.010	0.443	0.017	0.115	5.584	0.013	0.038	0.004
ABSL-L	0.082	0.008	0.441	0.015	0.115	6.030	0.009	0.017	0.003

data set. The results are summarized in Table 3. Examining this table we immediately note that the ESS/CPU measure is much larger for the proposed algorithms than for the standard methods. The improvement is very substantial, for example ESS/cpu for AABC-U is 12 times larger than for the standard ABC procedures like ABC-RW. The proposed methods generally also outperform SMC sampler in terms of overall efficiency, i.e. when variances of the quantiles and mean CPU time are considered jointly. Similar results are shown for Bayesian Synthetic Likelihood. We also examine DIM, DIC, TV and MSE quantities that provide information about the proximity of approximate samples to the exact MCMC ones. For all these quantities the smaller the value the better is the sampler. We see that all these measures for AABC-U and AABC-L are very similar to SMC, ABC-RW and ABC-IS and frequently outperforms them. Similarly for BSL approach. Another observation is that the approximated algorithm with the uniform and linear weights generally perform very similarly.

4 D: Stochastic Volatility with Gaussian emissions

When analyzing stationary time series, it is frequently observed that there are periods of high and periods of low volatility. Such phenomenon is called *volatility clustering*, see for example (Lux and Marchesi, 2000). One way to model such a behaviour is through a Stochastic Volatility (SV) model, where variances of the observed time series depend on hidden states that themselves form a stationary time series. Consider the following model which depends on three parameters $(\theta_1, \theta_2, \theta_3)$:

$$\begin{aligned} &x_1 \sim \mathcal{N}(0, 1/(1-\theta_1^2)); \quad v_i \overset{iid}{\sim} \mathcal{N}(0,1); \quad w_i \overset{iid}{\sim} \mathcal{N}(0,1); \quad i = \{1, \cdots, n\}, \\ &x_i = \theta_1 x_{i-1} + v_i; \quad i = \{2, \cdots, n\}, \\ &y_i = \sqrt{\exp(\theta_2 + \exp(\theta_3) x_i)} w_i; \quad i = \{1, \cdots, n\}. \end{aligned} \tag{3}$$

Only $\mathbf{y} = (y_1, \cdots, y_n)$ is observed while $(x_1, \cdots, x_n)$ are hidden states. The parameter $\theta_1 \in (-1, 1)$ controls the auto-correlation of hidden states, while θ_2 and θ_3 are unrestricted and relate to the hidden states influence on the variability of the observed series. Given a hidden state, the distribution of the observed variable is normal which may not be appropriate in some examples. We introduce the following priors, independently for each parameter:

$$\begin{aligned} &\theta_1 \sim Unif[0,1], \\ &\theta_2 \sim \mathcal{N}(0, 1, -L, L), \\ &\theta_3 \sim \mathcal{N}(0, 1, -L, L). \end{aligned} \tag{4}$$

We set the true parameters to $(\theta_1 = 0.95, \theta_2 = -2, \theta_3 = -1)$ and length of the time series $n = 500$. We use the Particle MCMC (PMCMC) as the Exact sampling scheme. Since the pseudo-data sets can be easily generated for every parameter value, the SV is a good example to demonstrate the performances of the generative algorithms considered here. For summary statistics we use a 7-dimensional vector whose components are:

(C1) $\#\{i : y_i^2 > \text{quantile}(\mathbf{y}_0^2, 0.99)\}$,

(C2) Average of $\mathbf{y}^2$,

(C3) Standard deviation of $\mathbf{y}^2$,

(C4) Sum of the first 5 auto-correlations of $\mathbf{y}^2$,

(C5) Sum of the first 5 auto-correlations of $\{\mathbf{1}_{\{y_i^2 < \text{quantile}(\mathbf{y}^2, 0.1)\}}\}_{i=1}^n$,

(C6) Sum of the first 5 auto-correlations of $\{\mathbf{1}_{\{y_i^2 < \text{quantile}(\mathbf{y}^2, 0.5)\}}\}_{i=1}^n$,

(C7) Sum of the first 5 auto-correlations of $\{\mathbf{1}_{\{y_i^2 < \text{quantile}(\mathbf{y}^2, 0.9)\}}\}_{i=1}^n$.

Figures 5, 6 and 7 show trace-plots, histograms and ACF function for AABC-U, ABSL-U and ABC-RW samplers respectively for each component (red lines correspond to the

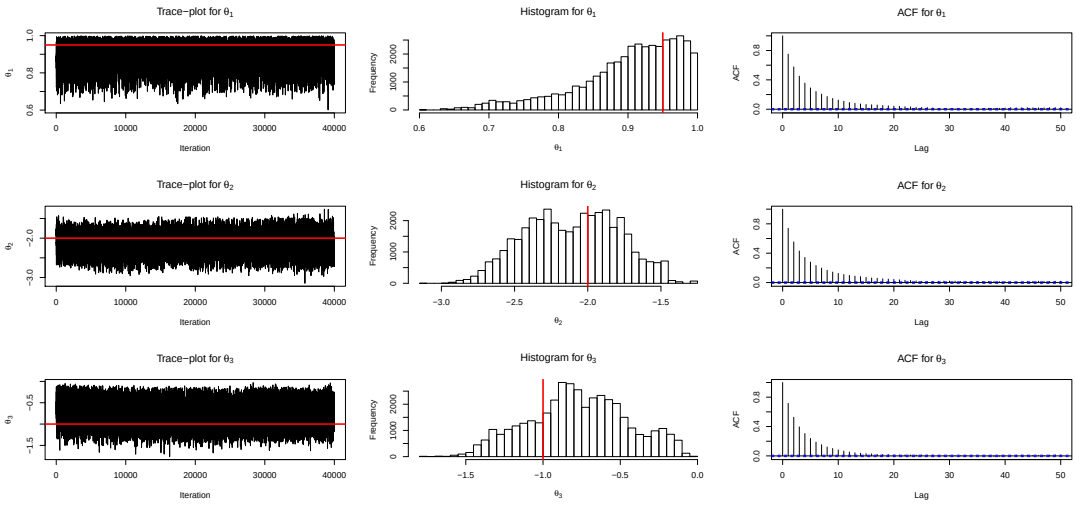

Figure 5: SV model: AABC-U Sampler. Each row corresponds to parameters θ_1 (top row), θ_2 (middle row) and θ_3 (bottom row) and shows in order from left to right: Trace-plot, Histogram and Auto-correlation function. Red lines represent true parameter values.

true parameter). The major observation is that AABC-U and ABSL-U are less sluggish than ABC-RW, exhibiting smaller auto-correlation values.

In Figure 8 we compare the sample-based kernel smoothing posterior marginal density estimates for Exact, SMC, ABC-RW and AABC-U (top row) as well as Exact, BSL-IS and ABSL-U (bottom row). We note that all samples obtained from the approximate algorithms are similar to the Exact posterior (produced using PMCMC with 100 particles). Generally all ABC-based samplers perform similarly, on the other hand ABSL-U performs worse than generic BSL-IS in this run as it is shifted away from the exact posterior for θ_1 and θ_3.

To get more general conclusions we show average results in Table 5 over 100 data replicates. We note that the proposed algorithms outperform the benchmark samplers by 8 times in ESS/cpu. These methods also significantly outperform the generic samplers if the variance of quantiles and mean (times CPU time) is considered. Moreover, AABC-U and AABC-L have very similar or smaller values for DIM, TV and MSE, which demonstrates that these samplers are much more efficient than the standard methods and at the same produce as accurate (or more accurate) parameter estimates than the generic algorithms.

ABSL-U and ABSL-L on the other hand did not perform well for this model, TV and MSE for these samplers are larger by 10% than the generic ones.

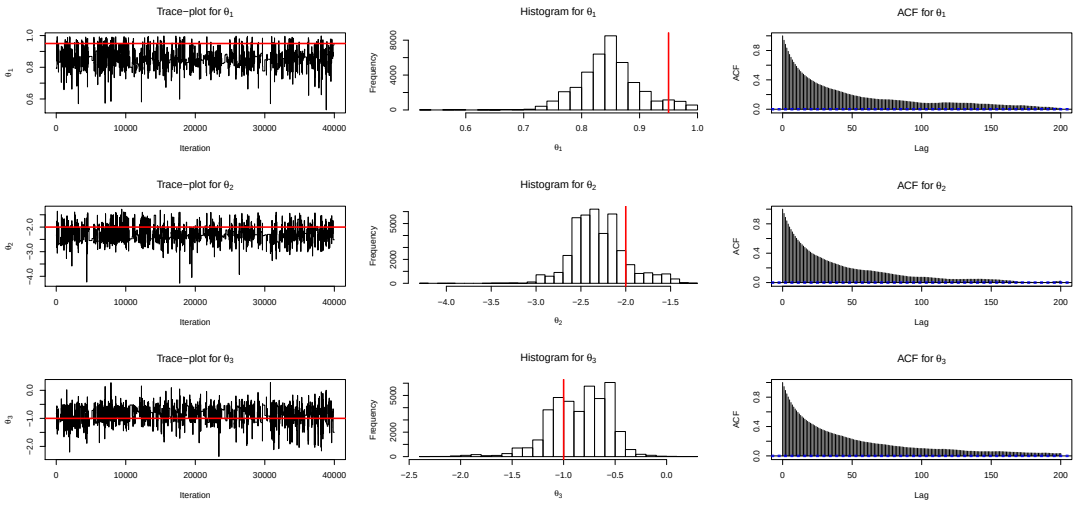

Figure 6: SV model: ABSL-U Sampler. Each row corresponds to parameters θ_1 (top row), θ_2 (middle row) and θ_3 (bottom row) and shows in order from left to right: Trace-plot, Histogram and Auto-correlation function. Red lines represent true parameter values.

Table 2: Simulation Results (SV model): Average Difference in Mean, Difference in Covariance, Total Variation, square roots of Bias and MSE, Effective Sample Size per CPU time, Variances of Lower/Upper Quantiles and Mean times CPU time for every sampling algorithm.

	Diff with Exact			Diff with True Par		Efficiency			
Sampler	DIM	DIC	TV	$\sqrt{\text{Bias}^2}$	$\sqrt{\text{MSE}}$	ESS/cpu	$\text{V}_{\text{QL}} * \text{cpu}$	$\text{V}_{\text{QU}} * \text{cpu}$	$\text{V}_{\text{E}} * \text{cpu}$
SMC	0.232	0.043	0.417	0.187	0.316	-	1.907	1.917	0.272
ABC-RW	0.210	0.040	0.459	0.228	0.342	0.097	2.785	1.200	1.678
ABC-IS	0.179	0.044	0.460	0.196	0.294	0.090	4.728	2.077	2.263
AABC-U	0.194	0.045	0.424	0.212	0.304	2.445	0.329	0.710	0.259
AABC-L	0.189	0.044	0.420	0.211	0.316	2.253	1.002	0.130	0.225
BSL-RW	0.200	0.036	0.411	0.175	0.287	0.043	12.619	10.843	2.888
BSL-IS	0.195	0.036	0.404	0.175	0.285	0.113	12.273	7.514	1.246
ABSL-U	0.229	0.042	0.551	0.184	0.303	0.822	0.945	1.766	0.326
ABSL-L	0.231	0.041	0.548	0.197	0.311	0.817	0.747	1.183	0.287

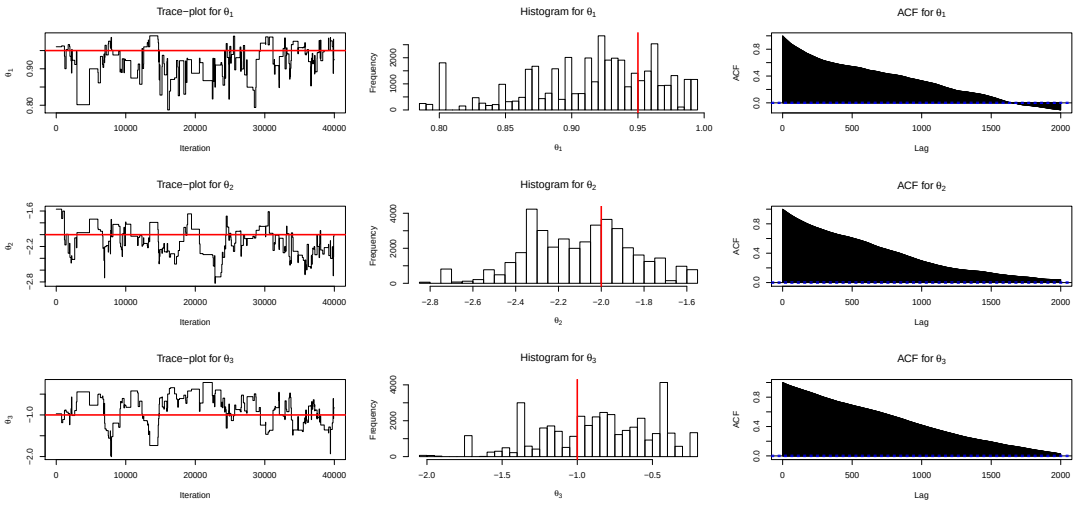

Figure 7: SV model: ABC-RW Sampler. Each row corresponds to parameters θ_1 (top row), θ_2 (middle row) and θ_3 (bottom row) and shows in order from left to right: Trace-plot, Histogram and Auto-correlation function. Red lines represent true parameter values.

5 E: Additional Plots

5.1 Additional Plots Related to Ricker's Model

Figures 9, 12 and 11 show trace-plots, histograms and ACF function for AABC-U, ABSL-U and ABC-RW samplers for each component (red lines correspond to the true parameter).

In Figure 12 we present the kernel density estimates of the marginal posteriors based on the samples obtained by different methods. First row compares Exact, SMC, ABC-RW and AABC-U samplers. Second row compares Exact, BSL-IS and ABSL-U. Columns correspond to parameter's components, from left to right: θ_1, θ_2 and θ_3.

5.2 Additional Plots Related to Stochastic Volatility with α-Stable errors

Figures 13, 15 and 14 show trace-plots, histograms and ACF function for AABC-U, ABSL-U and ABC-RW samplers respectively for each component (red lines correspond to the true parameters).

Figure 16 shows the plotted estimated densities based on the SMC, ABC-RW and AABC-U samplers.

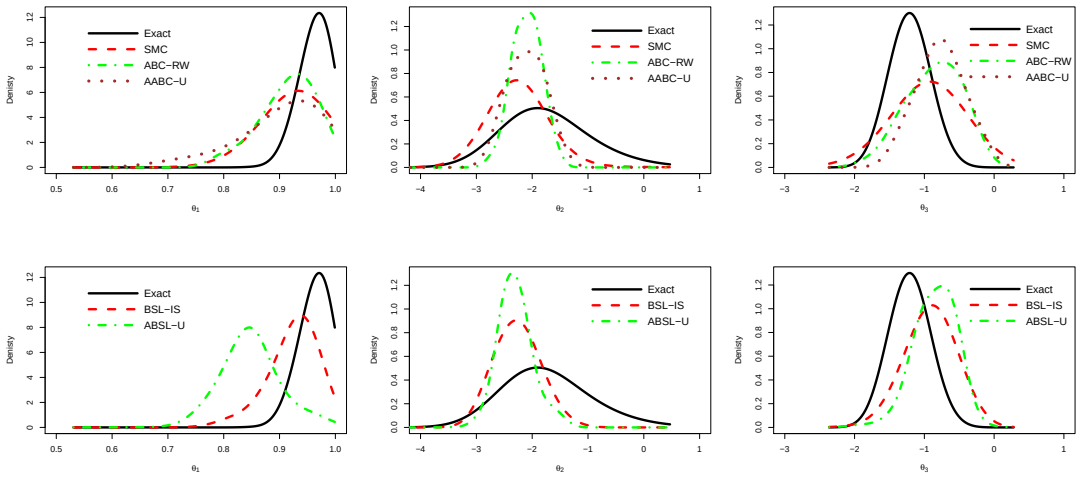

Figure 8: SV model: Estimated posterior marginal densities for each component. First row compares Exact, SMC, ABC-RW and AABC-U samplers. Second row compares Exact, BSL-IS and ABSL-U. Columns correspond to parameter's components, from left to right: θ_1, θ_2 and θ_3.

6 F: Additional Simulations using Random Walk Metropolis

In this section we perform additional simulations, on the same examples used in the main paper, but using random walk proposals. These new algorithms are no longer theoretically justifiable within the constructed framework, but it is still interesting to see how do these algorithms work in practice. Thus we compare the following algorithms:

(SMC) Standard Sequential Monte Carlo for ABC;

(ABC-RW) The modified ABC-MCMC algorithm which updates ϵ and the random walk Metropolis transition kernel during burn-in;

(ABC-IS) The modified ABC-MCMC algorithm which updates ϵ and the Independent Metropolis transition kernel during burn-in;

(BSL-RW) Modified BSL where it adapts the random walk Metropolis transition kernel during burn-in;

(BSL-IS) Modified BSL where it adapts the independent Metropolis transition kernel during burn-in;

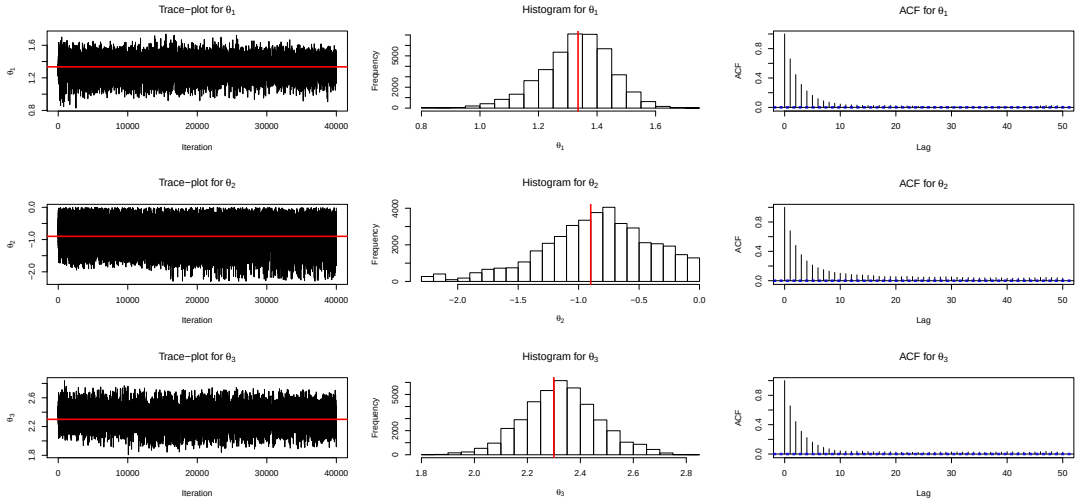

Figure 9: Ricker's model: AABC-U Sampler. Each row corresponds to parameters θ_1 (top row), θ_2 (middle row) and θ_3 (bottom row) and shows in order from left to right: Trace-plot, Histogram and Auto-correlation function. Red lines represent true parameter values.

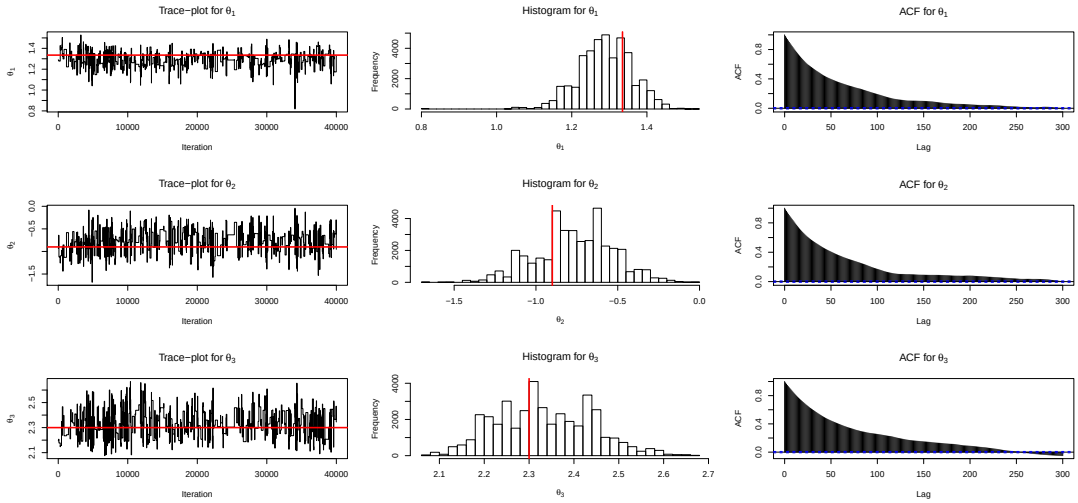

Figure 10: Ricker's model: ABSL-U Sampler. Each row corresponds to parameters θ_1 (top row), θ_2 (middle row) and θ_3 (bottom row) and shows in order from left to right: Trace-plot, Histogram and Auto-correlation function. Red lines represent true parameter values.

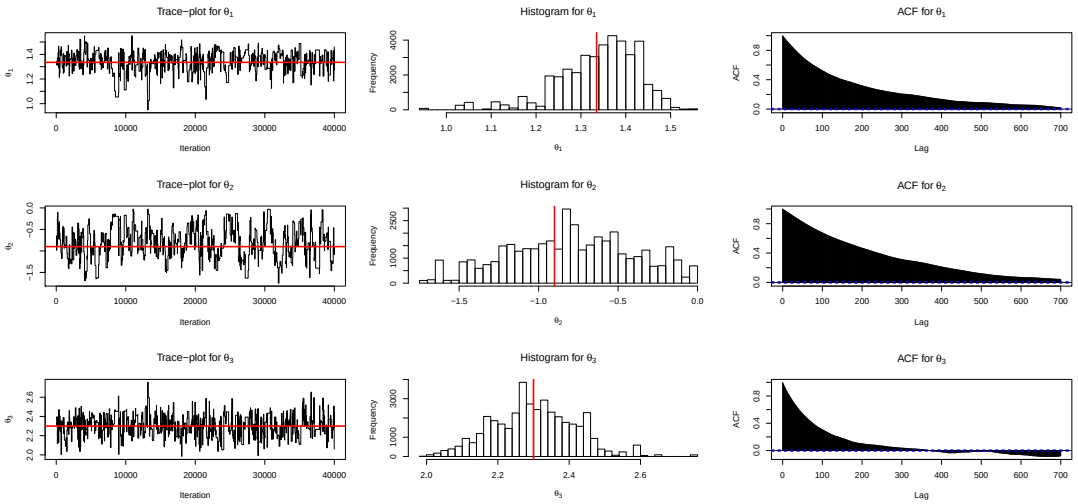

Figure 11: Ricker's model: ABC-RW Sampler. Each row corresponds to parameters θ_1 (top row), θ_2 (middle row) and θ_3 (bottom row) and shows in order from left to right: Trace-plot, Histogram and Auto-correlation function. Red lines represent true parameter values.

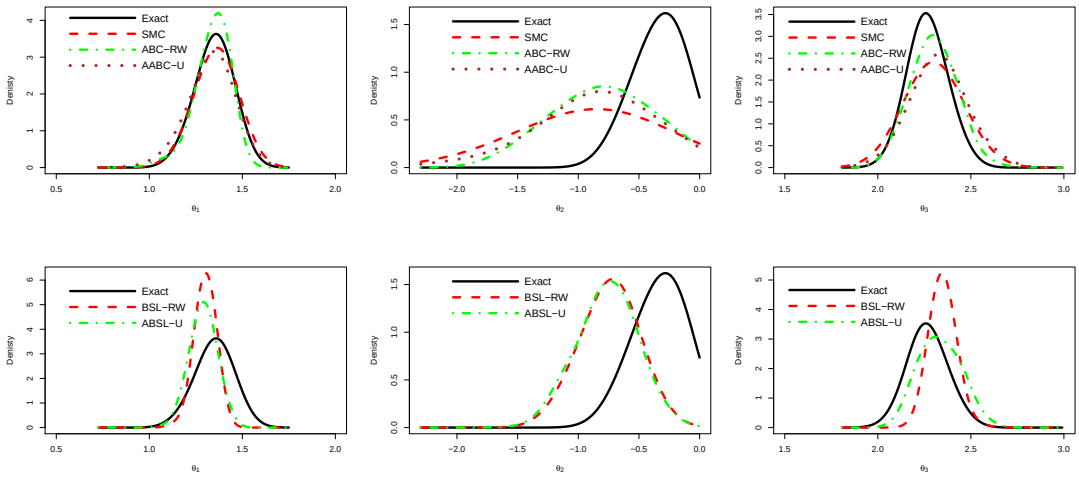

Figure 12: Ricker's model: Estimated posterior marginal densities for each component. First row compares Exact, SMC, ABC-RW and AABC-U samplers. Second row compares Exact, BSL-RW and ABSL-U. Columns correspond to parameter's components, from left to right: θ_1, θ_2 and θ_3.

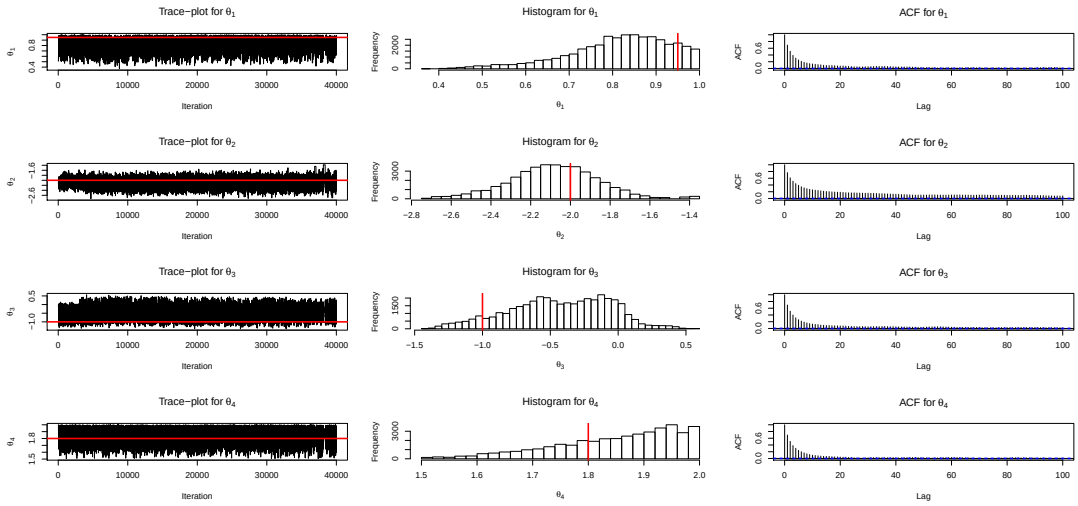

Figure 13: SV α-Stable model: AABC-U Sampler. Each row corresponds to parameters θ_1 (top row), θ_2 (second top row), θ_3 (second bottom row), θ_4 (bottom row) and shows in order from left to right: Trace-plot, Histogram and Auto-correlation function. Red lines represent true parameter values.

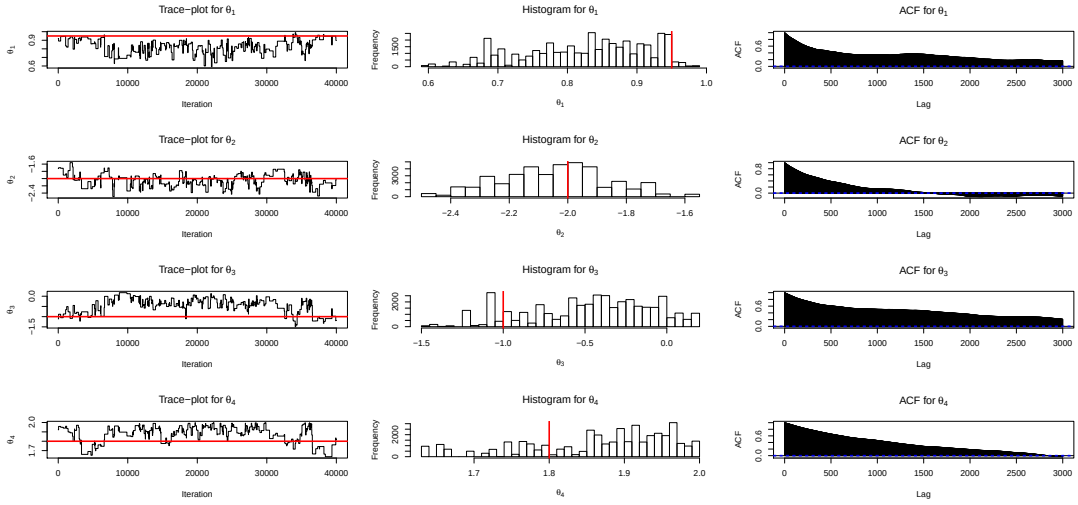

Figure 14: SV α-Stable model: ABC-RW Sampler. Each row corresponds to parameters θ_1 (top row), θ_2 (second top row), θ_3 (second bottom row), θ_4 (bottom row) and shows in order from left to right: Trace-plot, Histogram and Auto-correlation function. Red lines represent true parameter values.

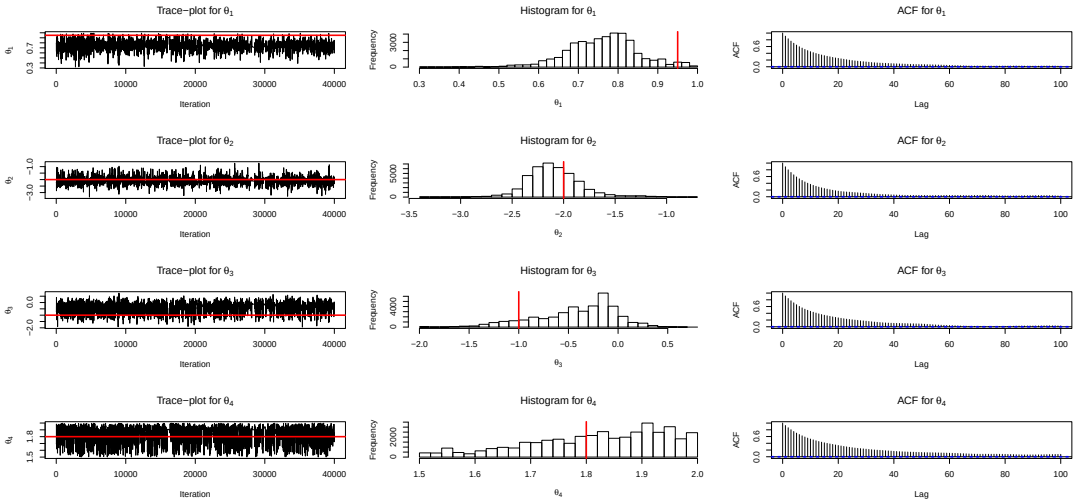

Figure 15: SV α-Stable model: ABSL-U Sampler. Each row corresponds to parameters θ_1 (top row), θ_2 (second top row), θ_3 (second bottom row), θ_4 (bottom row) and shows in order from left to right: Trace-plot, Histogram and Auto-correlation function. Red lines represent true parameter values.

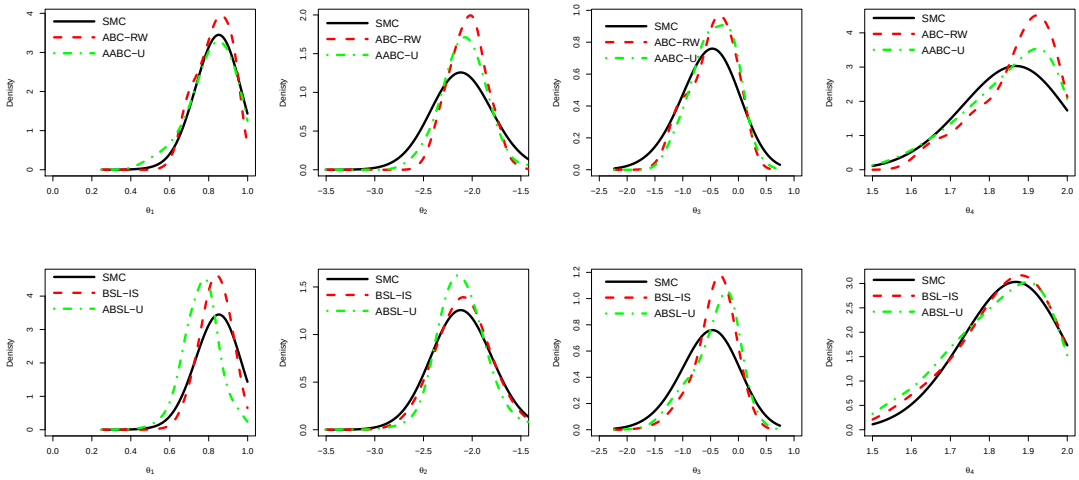

Figure 16: SV α-Stable model: Estimated densities for each component. First row compares SMC, ABC-RW and AABC-U samplers. Second row compares SMC, BSL-IS and ABSL-U. Columns correspond to parameter's components, from left to right: θ_1, θ_2, θ_3 and θ_4.

(AABC-IS-U) Approximated ABC-MCMC with independent proposals and uniform (U) weights;

(AABC-IS-L) Approximated ABC-MCMC with independent proposals and linear (L) weights;

(AABC-RW-U) Approximated ABC-MCMC with the random walk Metropolis transition kernel and uniform (U) weights;

(AABC-RW-L) Approximated ABC-MCMC with the random walk Metropolis transition kernel and linear (L) weights;

(ABSL-IS-U) Approximated BSL-MCMC with independent proposals and uniform (U) weights;

(AABC-IS-L) Approximated BSL-MCMC with independent proposals and linear (L) weights;

(ABSL-RW-U) Approximated BSL-MCMC with the random walk Metropolis transition kernel and uniform (U) weights;

(AABC-RW-L) Approximated BSL-MCMC with the random walk Metropolis transition kernel and linear (L) weights;

(Exact) Likelihood is computable and posterior samples are generated using an MCMC algorithm that is example-specific.

Note that AABC-IS-U, AABC-IS-L, ABSL-IS-U and ABSL-IS-L are the proposed algorithms described and assessed in the paper. We now add AABC-RW-U, AABC-RW-L, ABSL-RW-U and ABSL-RW-L where instead of an independent (IS), a random walk (RW) proposal is implemented (as in ABC-RW and BSL-RW) where covariance matrix is learned during the burn-in period.
Tables 3,4,5 and 6 show the performance of each sampling algorithm (averaged over 100 independent data-set replicates) for Moving Average, Ricker's, Stochastic Volatility with Gaussian and α-Stable errors models respectively.

First focusing on the ABC-based samplers we observe that for all the models the approximated posterior distributions of AABC-RW are generally similar to AABC-IS; DIM, DIC, TV and $\sqrt{\text{MSE}}$ are quite similar. The efficiency of these samplers (in terms of ESS/cpu) however shows a deterioration compared to the independent samplers, but still significantly outperform the generic samplers (ABC-RW and ABC-IS). When the efficiency is measured using variances of quantiles and mean (times CPU time) then generally AABC-RW-U performs better than AABC-RW-L. Moreover, for the Moving Average and Stochastic Volatility (with Gaussian errors) models, AABC-RW-U does not show a clear improvement over the benchmarks in terms of $\text{V}_\text{E} * \text{cpu}$. This is due to the fact that auto-correlation values (and hence the variance of the estimates) for RW samplers are generally larger than for IS.

Looking now at BSL-based algorithms, the situation is a little bit different. The proximity of the approximate posterior distribution of RW algorithms to the true posterior and true parameters is still similar to IS sampler however the efficiency changes from one model to another. For the Moving Average model, IS samplers clearly outperform the RW in terms of efficiency. It is reversed for Ricker's model where ABSL-RWs show much better ESS/cpu compared to the IS sampler. Finally for Stochastic volatility models (with Gaussian and α-stable emission probabilities) the efficiency of the algorithms with RW and IS proposals are comparable in terms of ESS/cpu, but IS is still more efficient when the variances of quantiles and mean (times CPU time) are considered. The algorithms with RW proposals still show better efficiency compared to the benchmarks (BSL-RW and BSL-IS).

To summarize, the RWM algorithms still show significant improvement in efficiency compared to the generic samplers, without sacrificing the precision of the estimated posterior. However, for some models, these samplers show a deterioration in efficiency when compared to the IS samplers. The dimension of parameter space for all the models considered does not exceed 4 and therefore the implementation of the IS is feasible. For larger parameter spaces the approximation of the target distribution with a multivariate normal and hence the use of IS may become more problematic and thus making RWM a viable alternative.

Table 3: Simulation Results (MA model): Average Difference in Mean, Difference in Covariance, Total Variation, square roots of Bias and MSE, Effective Sample Size per CPU time, Variances of Lower/Upper Quantiles and Mean times CPU time for every sampling algorithm.

	Diff with Exact			Diff with True Par		Efficiency			
Sampler	DIM	DIC	TV	$\sqrt{\text{Bias}^2}$	$\sqrt{\text{MSE}}$	ESS/cpu	$\text{V}_{\text{QL}} * \text{cpu}$	$\text{V}_{\text{QU}} * \text{cpu}$	$\text{V}_{\text{E}} * \text{cpu}$
SMC	0.081	0.004	0.389	0.014	0.115	-	0.040	0.052	0.009
ABC-RW	0.083	0.006	0.394	0.019	0.119	0.364	0.040	0.018	0.029
ABC-IS	0.082	0.007	0.390	0.015	0.116	0.634	0.083	0.013	0.040
AABC-IS-U	0.082	0.006	0.395	0.015	0.116	8.060	0.007	0.009	0.003
AABC-IS-L	0.083	0.007	0.398	0.014	0.116	6.418	0.004	0.011	0.013
AABC-RW-U	0.081	0.006	0.391	0.015	0.116	3.281	0.020	0.021	0.018
AABC-RW-L	0.081	0.006	0.392	0.014	0.116	2.632	0.023	0.025	0.016
BSL-RW	0.081	0.007	0.401	0.014	0.114	0.425	0.092	0.492	0.046
BSL-IS	0.081	0.007	0.403	0.015	0.115	1.413	0.035	0.109	0.010
ABSL-IS-U	0.080	0.009	0.401	0.017	0.115	7.428	0.013	0.038	0.004
ABSL-IS-L	0.081	0.007	0.400	0.015	0.115	8.061	0.009	0.017	0.003
ABSL-RW-U	0.080	0.013	0.409	0.023	0.116	1.741	1.087	1.553	0.045
ABSL-RW-L	0.081	0.011	0.404	0.021	0.115	2.057	0.373	0.733	0.022

Table 4: Simulation Results (Ricker's model): Average Difference in Mean, Difference in Covariance, Total Variation, square roots of Bias and MSE, Effective Sample Size per CPU time, Variances of Lower/Upper Quantiles and Mean times CPU time for every sampling algorithm.

	Diff with Exact			Diff with True Par		Efficiency			
Sampler	DIM	DIC	TV	$\sqrt{\text{Bias}^2}$	$\sqrt{\text{MSE}}$	ESS/cpu	$\text{V}_{\text{QL}} * \text{cpu}$	$\text{V}_{\text{QU}} * \text{cpu}$	$\text{V}_{\text{E}} * \text{cpu}$
SMC	0.151	0.017	0.336	0.084	0.218	-	0.655	0.180	0.065
ABC-RW	0.130	0.020	0.326	0.051	0.180	0.328	1.283	0.760	0.907
ABC-IS	0.154	0.022	0.383	0.082	0.221	0.169	0.800	1.875	0.881
AABC-IS-U	0.141	0.027	0.350	0.066	0.202	6.568	0.320	0.101	0.063
AABC-IS-L	0.142	0.026	0.339	0.070	0.200	7.818	0.505	0.065	0.245
AABC-RW-U	0.134	0.020	0.338	0.058	0.180	2.147	0.209	0.121	0.064
AABC-RW-L	0.138	0.021	0.334	0.067	0.193	1.996	11.869	0.117	3.026
BSL-RW	0.126	0.007	0.349	0.043	0.202	0.054	3.280	1.272	0.900
BSL-IS	0.111	0.008	0.343	0.015	0.182	0.011	31.815	4.890	6.892
ABSL-IS-U	0.101	0.006	0.323	0.023	0.167	0.302	1.042	0.444	0.242
ABSL-IS-L	0.109	0.006	0.331	0.014	0.181	0.241	2.280	0.664	0.267
ABSL-RW-U	0.115	0.005	0.326	0.019	0.185	0.523	0.737	0.542	0.202
ABSL-RW-L	0.113	0.005	0.331	0.021	0.184	0.365	0.954	0.498	0.332

Table 5: Simulation Results (SV model): Average Difference in Mean, Difference in Covariance, Total Variation, square roots of Bias and MSE, Effective Sample Size per CPU time, Variances of Lower/Upper Quantiles and Mean times CPU time for every sampling algorithm.

	Diff with Exact			Diff with True Par		Efficiency			
Sampler	DIM	DIC	TV	$\sqrt{\text{Bias}^2}$	$\sqrt{\text{MSE}}$	ESS/cpu	$\text{V}_{\text{QL}} * \text{cpu}$	$\text{V}_{\text{QU}} * \text{cpu}$	$\text{V}_{\text{E}} * \text{cpu}$
SMC	0.227	0.036	0.372	0.186	0.316	-	1.907	1.917	0.272
ABC-RW	0.196	0.035	0.397	0.220	0.319	0.149	2.785	1.200	1.678
ABC-IS	0.186	0.039	0.380	0.195	0.327	0.148	4.728	2.077	2.263
AABC-IS-U	0.191	0.038	0.385	0.217	0.309	4.051	0.329	0.710	0.259
AABC-IS-L	0.187	0.041	0.376	0.214	0.308	3.712	1.002	0.130	0.225
AABC-RW-U	0.220	0.036	0.409	0.230	0.402	2.298	1.052	0.330	0.401
AABC-RW-L	0.200	0.033	0.398	0.227	0.311	1.864	1.407	1.479	0.639
BSL-RW	0.195	0.031	0.374	0.180	0.289	0.078	12.619	10.843	2.888
BSL-IS	0.189	0.031	0.365	0.175	0.284	0.185	12.273	7.514	1.246
ABSL-IS-U	0.223	0.037	0.514	0.187	0.307	1.256	0.945	1.766	0.326
ABSL-IS-L	0.223	0.037	0.510	0.195	0.308	1.265	0.747	1.183	0.287
ABSL-RW-U	0.232	0.034	0.502	0.215	0.315	1.220	4.103	3.646	0.883
ABSL-RW-L	0.232	0.033	0.495	0.224	0.317	1.245	2.703	2.513	0.464

Table 6: Simulation Results (SV α-Stable model): Average Difference in Mean, Difference in Covariance, Total Variation, square roots of Bias and MSE, Effective Sample Size per CPU time, Variances of Lower/Upper Quantiles and Mean times CPU time for every sampling algorithm. In DIM, DIC and TV, samplers are compared to SMC.

	Diff with Exact			Diff with True Par		Efficiency			
Sampler	DIM	DIC	TV	$\sqrt{\text{Bias}^2}$	$\sqrt{\text{MSE}}$	ESS/cpu	$\text{V}_{\text{QL}} * \text{cpu}$	$\text{V}_{\text{QU}} * \text{cpu}$	$\text{V}_{\text{E}} * \text{cpu}$
SMC	0.000	0.000	0.000	0.224	0.305	-	1.754	0.375	0.142
ABC-RW	0.087	0.015	0.173	0.242	0.321	0.120	1.387	0.249	0.565
ABC-IS	0.077	0.016	0.185	0.225	0.306	0.120	0.799	0.871	0.472
AABC-IS-U	0.077	0.014	0.130	0.254	0.323	2.550	0.252	0.511	0.085
AABC-IS-L	0.068	0.013	0.120	0.246	0.304	2.344	0.111	0.093	0.080
AABC-RW-U	0.073	0.010	0.127	0.248	0.336	1.595	0.278	0.085	0.130
AABC-RW-L	0.061	0.009	0.110	0.257	0.315	1.399	3.464	0.772	1.250
BSL-RW	0.045	0.011	0.131	0.230	0.293	0.064	10.847	3.657	3.111
BSL-IS	0.043	0.009	0.117	0.225	0.288	0.142	12.356	5.826	1.246
ABSL-IS-U	0.065	0.014	0.210	0.229	0.297	1.088	0.706	0.495	0.196
ABSL-IS-L	0.062	0.013	0.198	0.227	0.295	1.014	1.372	0.470	0.238
ABSL-RW-U	0.059	0.012	0.196	0.266	0.322	1.057	3.520	2.274	0.399
ABSL-RW-L	0.059	0.012	0.202	0.274	0.328	1.065	3.546	1.707	0.366

7 G: Data Analysis

For real world example we consider Dow-Jones index daily log returns from January 1, 2010 until December 31, 2018. The data were downloaded from Yahoo Finance[1] website. Given a time series of prices P_i, $i = 1, \cdots, n$, log returns are calculated in the following way:

$$r_i = \log(P_i) - \log(P_{i-1}), \quad i = 2, \cdots, n.$$

The resulting time series is of length 2262. To make log returns more suitable for analysis, we standardize r_t by subtracting its mean and then multiply each return by 200, so that absolute values were not too small, Figure 17 shows transformed returns. This time series ($\mathbf{y}_0$) has mean zero by construction, and its auto-correlations and partial auto-correlations are insignificant for any lag. However, it is obvious that variances are correlated and there are alternating periods of low and high variability. This prompts us to use Stochastic Volatility model with α-Stable errors as described in the previous section. Since the likelihood does not exist for this class of models, the simulation-based methods are probably the only available tools for the inference. The evolution of time series is described by equation (5.4) and the parameter's prior is set as in equation (5.5). The skewed parameter of Stable distribution is fixed at value of -1. To estimate the posterior distribution we run AABC-U and ABLS-U samplers. The summary statistic for both methods is the same 7-dimensional vector defined in section 5.2. Each chain was run for 100 thousand iterations with last 80 thousands used for inference. Figures 18 and 19 show trace-plots and histograms for AABC-U and ABSL-U samplers respectively for each parameter. The conclusions are in agreement with the ones suggested by the

[1] https://ca.finance.yahoo.com/

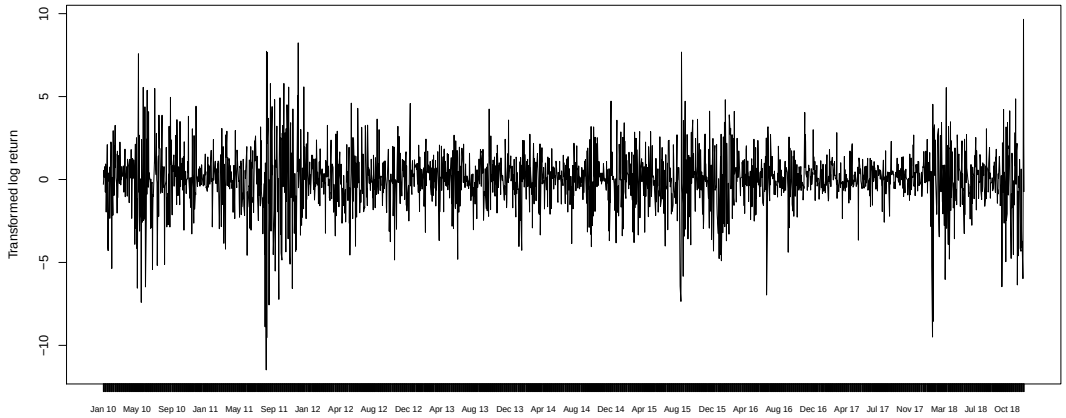

Figure 17: Dow Jones daily transformed log return for a period of Jan 2010 - Dec 2018.

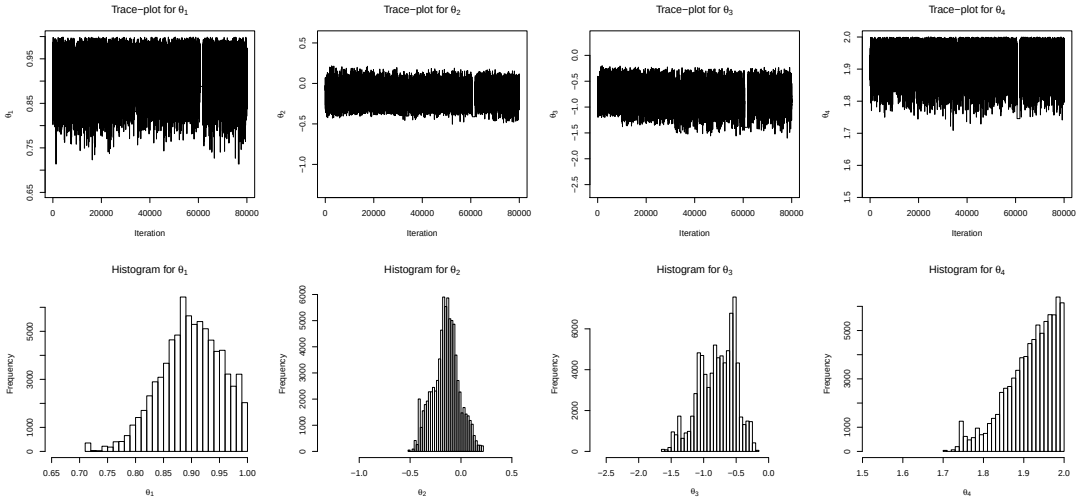

Figure 18: Dow Jones log returns: AABC-U Sampler. Every column corresponds to a particular parameter component from left to right: θ_1, θ_2, θ_3, θ_4 and shows trace-plot on top and histogram on bottom.

simulation study. The mixing of AABC-U is generally better than of ABSL-U. However, posterior draws of ABSL-U for the first 3 components are uni-modal, symmetric and

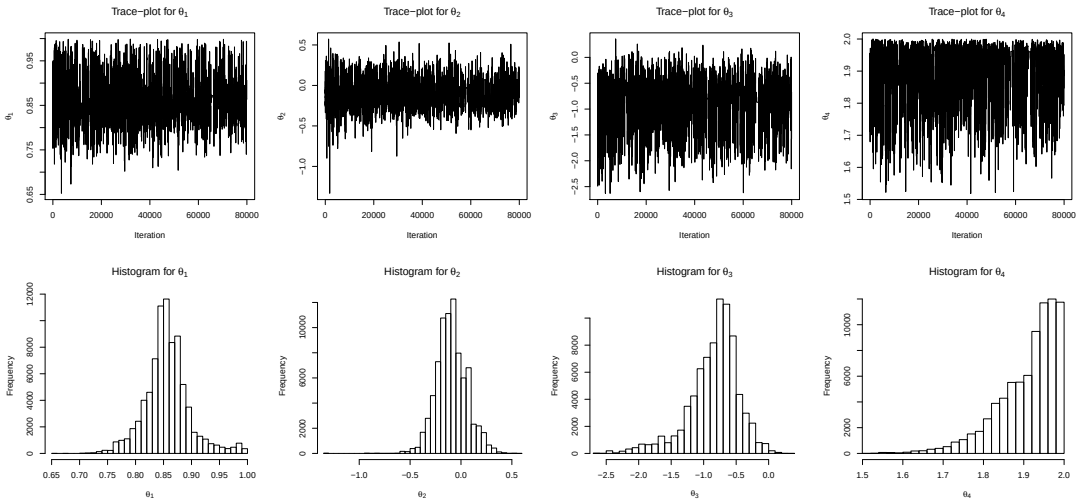

Figure 19: Dow Jones log returns: ABSL-U Sampler. Every column corresponds to a particular parameter component from left to right: θ_1, θ_2, θ_3, θ_4 and shows trace-plot on top and histogram on bottom.

bell-shaped, which is not surprising since the use of Gaussian priors within the BSL method yields Gaussian posteriors due to conjugacy. Table 7 reports posterior mean and 95% credible intervals for every parameter and for both samplers. AABC-U and

Table 7: Dow Jones log return stochastic volatility: 95% credible intervals and posterior averages for 4 parameters for two proposed samplers (AABC-U and ABSL-U).

	AABC-U			ABSL-U		
Parameter	2.5% Quantile	Average	97.5% Quantile	2.5% Quantile	Average	97.5% Quantile
θ_1	0.787	0.899	0.990	0.775	0.856	0.959
θ_2	-0.411	-0.147	0.112	-0.369	-0.092	0.222
θ_3	-1.405	-0.790	-0.304	-1.858	-0.841	-0.206
θ_4	1.758	1.916	1.997	1.721	1.909	1.996

ABSL-U produce similar results. We see that the estimated correlation between adjacent variables in the hidden layer is about 0.9 and the estimate of α-Stable emission noise is 1.91. This model can produce more extreme values than those predicted by one with standard Gaussian noise.

8 H: Theoretical Justifications

8.1 Proofs of theorems

Proof of Theorem 6.1. Note that $\sup_\theta \|\hat{h}(\theta; \mathcal{Z}_N) - h(\theta)\| \to 0$ w.p.1 implies that for all θ and ζ^* in Θ:

$$\hat{h}(\theta; \mathcal{Z}_N) \xrightarrow{p} h(\theta),$$

$$\hat{h}(\zeta^*; \mathcal{Z}_N) \xrightarrow{p} h(\zeta^*),$$

therefore by Slutsky's theorem we obtain

$$\frac{\hat{h}(\zeta^*; \mathcal{Z}_N)}{\hat{h}(\theta; \mathcal{Z}_N)} \xrightarrow{p} \frac{h(\zeta^*)}{h(\theta)},$$

for all (θ, ζ^*) in $\Theta \times \Theta$. therefore

$$\hat{\alpha}(\theta, \zeta^*; \mathcal{Z}_N) = \frac{p(\zeta^*)q(\theta)\hat{h}(\zeta^*; \mathcal{Z}_N)}{p(\theta)q(\zeta^*)\hat{h}(\theta; \mathcal{Z}_N)} \xrightarrow{p} \frac{p(\zeta^*)q(\theta)h(\zeta^*)}{p(\theta)q(\zeta^*)h(\theta)} = \alpha(\theta, \zeta^*).$$

Since $\min(1, x)$ is a continuous function, Continuous Mapping Theorem implies that

$$\hat{a}(\theta, \zeta^*; \mathcal{Z}_N) = \min(1, \hat{\alpha}(\theta, \zeta^*; \mathcal{Z}_N)) \xrightarrow{p} \min(1, \alpha(\theta, \zeta^*)) = a(\theta, \zeta^*).$$

Note that this not just a point-wise convergence, but uniform convergence in probability so that one C will work for all (θ, ζ^*). That is, for any (θ, ζ^*), $\delta > 0$ and $\epsilon > 0$ there exists C such that for all $N > C$, $P(|\hat{a}(\theta, \zeta^*; \mathcal{Z}_N) - a(\theta, \zeta^*)| > \delta) < \epsilon$.
Another important observation is that (fixing θ, ζ^* and letting $a(\theta, \zeta^*) = a$ and $\hat{a}(\theta, \zeta^*; \mathcal{Z}_N) = \hat{a}$ for convenience)

$$\begin{aligned} E_{\mathcal{Z}_N}(|\hat{a} - a|) = \int |\hat{a} - a| dF(\mathcal{Z}_N) = \int_{|\hat{a}-a|<\delta} |\hat{a} - a| dF(\mathcal{Z}_N) + \int_{|\hat{a}-a|\geq\delta} |\hat{a} - a| dF(\mathcal{Z}_N) \leq \\ \leq \delta + \int_{|\hat{a}-a|\geq\delta} dF(\mathcal{Z}_N) \leq \delta + \epsilon. \end{aligned} \tag{5}$$

Because $|\hat{a} - a| \leq 1$ and applying the definition of the convergence in probability. The above inequality shows that we can make this expected value arbitrary small by taking large enough N, moreover this result is uniform, so one N will work for all θ and ζ^*.
Next we focus on the distance between two transition kernels, this discussion is similar to the proof of Corollary 2.3 in Alquier et al. (2016). Observe that (using independent proposals):

$$\begin{aligned} P(\theta, d\zeta^*) &= q(\zeta^*)a(\theta, \zeta^*)d\zeta^* + \delta_\theta(\zeta^*)r(\theta), \\ \hat{P}_N(\theta, d\zeta^*) &= \int q(\zeta^*)\hat{a}(\theta, \zeta^*; \mathcal{Z}_N)d\zeta^* dF(\mathcal{Z}_N) + \delta_\theta(\zeta^*)\hat{r}_N(\theta), \end{aligned}$$

where $r(\theta) = 1 - \int q(\zeta^*)a(\theta, \zeta^*)d\zeta^*$ and $\hat{r}_N(\theta) = 1 - \iint q(\zeta^*)a(\theta, \zeta^*)d\zeta^* dF(\mathcal{Z}_N)$. Fix $\theta \in \Theta$, and noting that the total variation between two probability distributions that have densities is also equal to:

$$\|\pi - \hat{\pi}\|_{TV} = 0.5 \int |\pi(\theta) - \hat{\pi}(\theta)| d\theta.$$

Therefore

$$\begin{aligned} P(\theta, d\zeta^*) - \hat{P}_N(\theta, d\zeta^*) = &\int q(\zeta^*)(a(\theta, \zeta^*) - \hat{a}(\theta, \zeta^*; \mathcal{Z}_N))dF(\mathcal{Z}_N) \\ &+ \delta_\theta(d\zeta^*) \iint q(t)(a(\theta, t) - \hat{a}(\theta, t; \mathcal{Z}_N))dF(\mathcal{Z}_N)dt, \end{aligned} \tag{6}$$

and it follows that

$$\begin{aligned} \|P(\theta, d\zeta^*) - \hat{P}_N(\theta, d\zeta^*)\|_{TV} \leq &0.5 \left\{ \int \left| \int q(\zeta^*)(a(\theta, \zeta^*) - \hat{a}(\theta, \zeta^*; \mathcal{Z}_N))dF(\mathcal{Z}_N) \right| d\zeta^* \right. \\ &\left. + \left| \iint q(t)(a(\theta, t) - \hat{a}(\theta, t; \mathcal{Z}_N))dF(\mathcal{Z}_N)dt \right| \right\} \\ \leq &0.5 \left\{ \int \left| \int q(\zeta^*)(a(\theta, \zeta^*) - \hat{a}(\theta, \zeta^*; \mathcal{Z}_N))dF(\mathcal{Z}_N) \right| d\zeta^* \right. \\ &\left. + \int \left| \int q(t)(a(\theta, t) - \hat{a}(\theta, t; \mathcal{Z}_N))dF(\mathcal{Z}_N) \right| dt \right\} \\ = &\int \left| \int q(t)(a(\theta, t) - \hat{a}(\theta, t; \mathcal{Z}_N))dF(\mathcal{Z}_N) \right| dt \\ \leq &\iint q(t) \left| a(\theta, t) - \hat{a}(\theta, t; \mathcal{Z}_N) \right| dF(\mathcal{Z}_N)dt \leq \delta + \epsilon \end{aligned} \tag{7}$$

for any $\epsilon > 0$ and $\delta > 0$ and large enough N by (5). Since this result is true for any $\theta \in \Theta$ we finally get the main result:

$$\sup_\theta \|\hat{P}_N(\theta, d\zeta^*) - P(\theta, d\zeta^*)\|_{TV} \leq \delta + \epsilon \tag{8}$$

□

Proof of Theorem 6.2. We generally follow the proof of Theorem 2.4 in Johndrow et al. (2015). First observe that:

$$\nu\hat{P}_0 \cdots \hat{P}_M - \mu P^M = (\nu - \mu)P^M + \sum_{t=0}^{M-1} \nu\hat{P}_0 \cdots \hat{P}_t(\hat{P}_{t+1} - P)P^{M-t-1}.$$

By Assumptions 2 and 3, we get:

$$\|\nu\hat{P}_0 \cdots \hat{P}_t\hat{P}_{t+1} - \nu\hat{P}_0 \cdots \hat{P}_t P\|_{TV} \leq \epsilon,$$

and

$$\|\nu\hat{P}_0\cdots\hat{P}_t\hat{P}_{t+1}P^{M-t-1}-\nu\hat{P}_0\cdots\hat{P}_tPP^{M-t-1}\|_{TV}\leq\epsilon(1-\alpha)^{M-t-1}.$$

Using these results, the triangular inequality and the formula for the sum of finite geometric series we establish that:

$$\begin{aligned}\|\nu\hat{P}_0\cdots\hat{P}_M-\mu P^M\|_{TV}\leq&\|\mu P^M-\nu P^M\|_{TV}+\sum_{t=0}^{M-1}\|\nu\hat{P}_0\cdots\hat{P}_t\hat{P}_{t+1}P^{M-t-1}-\nu\hat{P}_0\cdots\hat{P}_tPP^{M-t-1}\|_{TV}\\ \leq&(1-\alpha)^M\|\mu-\nu\|_{TV}+\epsilon\sum_{t=0}^{M-1}(1-\alpha)^{M-t-1}\\ =&(1-\alpha)^M\|\mu-\nu\|_{TV}+\epsilon\frac{1-(1-\alpha)^M}{\alpha}.\end{aligned}\tag{9}$$

Finally we get the main result using that fact that μ is invariant for P (again using the sum of finite geometric series)

$$\begin{aligned}\left\|\mu-\frac{\sum_{t=0}^{M-1}\nu\hat{P}_0\cdots\hat{P}_t}{M}\right\|_{TV}=&\left\|\frac{\sum_{t=0}^{M-1}\mu P^t}{M}-\frac{\sum_{t=0}^{M-1}\nu\hat{P}_0\cdots\hat{P}_t}{M}\right\|_{TV}\\ \leq&\frac{1}{M}\sum_{t=0}^{M-1}\|\mu P^t-\nu\hat{P}_0\cdots\hat{P}_t\|_{TV}\\ \leq&\frac{1}{M}\sum_{t=0}^{M-1}\left((1-\alpha)^t\|\mu-\nu\|_{TV}+\epsilon\frac{1-(1-\alpha)^t}{\alpha}\right)\\ =&\frac{(1-(1-\alpha)^M)\|\mu-\nu\|_{TV}}{M\alpha}-\frac{\epsilon(1-(1-\alpha)^M)}{M\alpha^2}+\frac{\epsilon}{\alpha}.\end{aligned}\tag{10}$$

□

Proof of Lemma 6.1. Without loss of generality we assume that $k>j$, next define:

$$\tilde{f}(\theta^{(j)})=f(\theta^{(j)})-\mu_jf,$$

$$\tilde{g}(\theta^{(k)})=g(\theta^{(k)})-\mu_kg,$$

so that $E[\tilde{f}(\theta^{(j)})]=E[\tilde{g}(\theta^{(k)})]=0$. Then we get the following

$$\begin{aligned}cov(f(\theta^{(j)}),g(\theta^{(k)}))=&E[\tilde{f}(\theta^{(j)})\tilde{g}(\theta^{(k)})]=E[E[\tilde{f}(\theta^{(j)})\tilde{g}(\theta^{(k)})|\theta^{(j)}]]\\ =&E[\tilde{f}(\theta^{(j)})E[\tilde{g}(\theta^{(k)})|\theta^{(j)}]]=E_{\theta^{(j)}}[\tilde{f}(\theta^{(j)})\delta_{\theta^{(j)}}\hat{P}_{j+1}\cdots\hat{P}_k\tilde{g}],\end{aligned}\tag{11}$$

where δ_θ is point mass at θ and using our notation $\delta_{\theta^{(j)}}\hat{P}_{j+1}\cdots\hat{P}_k$ corresponds to the conditional distribution of $\theta^{(k)}$ given a fixed value of $\theta^{(j)}$.
Using the general observation that for any two measures ν_1 and ν_2 and any bounded function f the following inequality holds

$$|\nu_1f-\nu_2f|\leq2|f|\|\nu_1-\nu_2\|_{TV},\tag{12}$$

we find that:

$$\begin{aligned}|\delta_{\theta^{(j)}}\hat{P}_{j+1}\cdots\hat{P}_k\tilde{g}| =&|\delta_{\theta^{(j)}}\hat{P}_{j+1}\cdots\hat{P}_k\tilde{g}-0| = |\delta_{\theta^{(j)}}\hat{P}_{j+1}\cdots\hat{P}_k\tilde{g}-\mu_k\tilde{g}|\\ =&|\delta_{\theta^{(j)}}\hat{P}_{j+1}\cdots\hat{P}_k\tilde{g}-\mu_j\hat{P}_{j+1}\cdots\hat{P}_k\tilde{g}| \leq 2|\tilde{g}|\|\delta_{\theta^{(j)}}\hat{P}_{j+1}\cdots\hat{P}_k-\mu_j\hat{P}_{j+1}\cdots\hat{P}_k\|_{TV}\\ \leq&2|\tilde{g}|(1-\alpha^*)^{|k-j|}\end{aligned} \tag{13}$$

note that this result is for any $\theta^{(j)} \in \Theta$. Returning to (11) we get that:

$$cov(\tilde{f}(\theta^{(j)}),\tilde{g}(\theta^{(k)})) \leq 2|\tilde{f}||\tilde{g}|(1-\alpha^*)^{|k-j|}. \tag{14}$$

Finally by the triangular inequality $|\tilde{f}| \leq 2|f|$ for any $j = 1, 2, \cdots$ and similarly for $|\tilde{g}|$. The desired result follows immediately. □

Proof of Theorem 6.3. Using our standard notation $\nu\hat{P}_0\cdots\hat{P}_t f = E[f(\theta^{(t)})]$, Theorem 6.2, Lemma 6.1 and properties of the double sum of geometric series we get

$$\begin{aligned}&E\left[\left(\mu f-\frac{1}{M}\sum_{t=0}^{M-1}f(\theta^{(t)})\right)^2\right] = E\left[\left(\mu f-\frac{1}{M}\sum_{t=0}^{M-1}\nu\hat{P}_0\cdots\hat{P}_t f+\frac{1}{M}\sum_{t=0}^{M-1}\nu\hat{P}_0\cdots\hat{P}_t f-\frac{1}{M}\sum_{t=0}^{M-1}f(\theta^{(t)})\right)^2\right]\\ &=\left(\mu f-\frac{1}{M}\sum_{t=0}^{M-1}\nu\hat{P}_0\cdots\hat{P}_t f\right)^2+E\left[\left(\frac{1}{M}\sum_{t=0}^{M-1}\nu\hat{P}_0\cdots\hat{P}_t f-\frac{1}{M}\sum_{t=0}^{M-1}f(\theta^{(t)})\right)^2\right]\\ &\leq\left(2|f|\left(\frac{(1-(1-\alpha)^M)\|\mu-\nu\|_{TV}}{M\alpha}-\frac{\epsilon(1-(1-\alpha)^M)}{M\alpha^2}+\frac{\epsilon}{\alpha}\right)\right)^2+\frac{1}{M^2}\sum_{j=0}^{M-1}\sum_{t=0}^{M-1}cov(f(\theta^{(j)}),f(\theta^{(t)}))\\ &\leq 4|f|^2\left(\frac{(1-(1-\alpha)^M)}{M\alpha}-\frac{\epsilon(1-(1-\alpha)^M)}{M\alpha^2}+\frac{\epsilon}{\alpha}\right)^2+\frac{8|f|^2}{M^2}\sum_{j=0}^{M-1}\sum_{t=0}^{M-1}(1-\alpha^*)^{|t-j|}\\ &=4|f|^2\left(\frac{(1-(1-\alpha)^M)}{M\alpha}-\frac{\epsilon(1-(1-\alpha)^M)}{M\alpha^2}+\frac{\epsilon}{\alpha}\right)^2\\ &+8|f|^2\left(\frac{1}{M}+\frac{2}{(\alpha^*)^2}\left(\frac{(1-\alpha^*)^{M+1}-(1-\alpha^*)}{M^2}+\frac{(1-\alpha^*)-(1-\alpha^*)^2}{M}\right)\right).\end{aligned} \tag{15}$$

□

Proof of Theorem 6.6. First by **(B1)** - **(B4)**, Theorem 6.5 guarantees the uniform ergodicity of the exact chain P with $\beta = \min_{\theta\in\Theta}\frac{q(\theta)}{p(\theta)h(\theta)/c}$ where c is the normalizing constant of the posterior. Note that $\beta > 0$ since Θ is compact, the ratio is continuous and never zero. Therefore P satisfies Doeblin Condition. Next from **(B1)**, **(B4)** and **(B5)**, Theorem 6.4 implies that $\sup_{\theta\in\Theta}\|\hat{h}(\theta;\mathcal{Z}_N)-h(\theta)\| \to 0$ with probability 1. Hence by Theorem 6.1 the perturbed kernel $\hat{P}_N$ can be made arbitrary close to the exact kernel P for sufficiently large N. Note that the total variation distance between $\hat{P}_N$ and P decreases to zero as N increases. Finally, the assumptions (and therefore conclusions) of Theorems 6.2 and 6.3 follow trivially. □

Proof of Corollary 6.1. First by **(B1)**, **(B2)**, **(B3)**, **(B4)** and **(B8)**, Theorem 6.5 guarantees the uniform ergodicity of the exact chain P with $\beta = \min_{\theta\in\Theta} \frac{q(\theta)}{p(\theta)h(\theta)/c}$ where c is the normalizing constant of the posterior. Note that $\beta > 0$ since Θ is compact, the ratio is continuous and never zero. Therefore P satisfies the Doeblin Condition. Next from **(B1)**, **(B5)**, **(B6)** and **(B7)**, Theorem 6.4 implies that $\sup_{\theta\in\Theta} \|\hat{h}(\theta; \mathcal{Z}_N) - h(\theta)\| \to 0$ with probability 1. Hence by Theorem 6.1 the perturbed kernel $\hat{P}_N$ can be made arbitrary close to the exact kernel P for sufficiently large N. Note that the total variation distance between $\hat{P}_N$ and P decreases to zero as N increases. Finally, the assumptions (and therefore conclusions) of Theorems 6.2 and 6.3 follow trivially. □

References

Alquier, P., Friel, N., Everitt, R., and Boland, A. (2016). "Noisy Monte Carlo: Convergence of Markov chains with approximate transition kernels." *Statistics and Computing*, 26(1-2): 29–47. 21

Johndrow, J. E., Mattingly, J. C., Mukherjee, S., and Dunson, D. (2015). "Optimal approximating Markov chains for Bayesian inference." *arXiv preprint arXiv:1508.03387*. 22

Lux, T. and Marchesi, M. (2000). "Volatility clustering in financial markets: a microsimulation of interacting agents." *International journal of theoretical and applied finance*, 3(04): 675–702. 6

Algorithm 1 Approximated Bayesian Synthetic Likelihood (ABSL)

Given s_0, constant c, burn-in period B, J number of adaption points during burn-in , required number of samples M, initial pseudo data simulations $\mathcal{Z}_N = \{\tilde{\zeta}_n, \{\tilde{s}_n^{(j)}\}_{j=1}^m\}_{n=1}^N$ with $\tilde{\zeta}_n \sim p(\zeta)$, $\tilde{\mathbf{y}}_n^{(j)} \sim f(\mathbf{y}|\tilde{\zeta}_n)$ and $\tilde{s}_n^{(j)} = S(\tilde{\mathbf{y}}_n^{(j)})$.

Get initial $\theta^{(0)}$.

Let $\tilde{\mu}$ be expectation of prior distribution and $\tilde{\Sigma} = c\Sigma$ where Σ is covariance of the prior $p(\theta)$.

Define, $b = \lfloor (B/J) \rfloor$ and define sequence $(a_1, \cdots, a_J) = (b, 2b, \cdots, Jb)$

for $t = 1, \cdots, M$ **do**

  if $t = a_j$ for some $j = 1, \cdots, J$ **then**

    Find $\tilde{\mu}$ as mean of $\theta^{(t)}$ $t = 1, \cdots, (a_j - 1)$ and $\tilde{\Sigma} = c\Sigma$ where Σ is covariance of $\theta^{(t)}$ $t = 1, \cdots, (a_j - 1)$.

  end if

  Generate $\zeta^*, \tilde{\zeta}^* \overset{iid}{\sim} \mathcal{N}(.; \tilde{\mu}, \tilde{\Sigma})$.

  Simulate $\tilde{\mathbf{y}}^{*(j)} \sim f(\mathbf{y}|\tilde{\zeta}^*)$ and let $\tilde{s}^{*(j)} = S(\tilde{\mathbf{y}}^{*(j)})$ for $1 \leq j \leq m$.

  Add simulated parameter and statistics to the past set: $\mathcal{Z}_N = \mathcal{Z}_{N-1} \cup \{\tilde{\zeta}^*, \{\tilde{s}_n^{(j)}\}_{j=1}^m\}$ and set $N = N + 1$.

  Calculate:

$$\begin{aligned} \hat{\mu}_{\zeta^*} &= \frac{\sum_{n=1}^N [W_{Nn}(\zeta^*) \sum_{j=1}^m \tilde{s}_n^{(j)}]}{m \sum_{n=1}^N W_{Nn}(\zeta^*)} \\ \hat{\Sigma}_{\zeta^*} &= \frac{\sum_{i=1}^N [W_{Nn}(\zeta^*) \sum_{j=1}^m (\tilde{s}_n^{(j)} - \hat{\mu}_{\zeta^*})(\tilde{s}_n^{(j)} - \hat{\mu}_{\zeta^*})^T]}{m \sum_{i=1}^N W_{Nn}(\zeta^*)} \end{aligned}$$

  Calculate:

$$\begin{aligned} \hat{\mu}_{\theta^{(t)}} &= \frac{\sum_{n=1}^N [W_{Nn}(\theta^{(t)}) \sum_{j=1}^m \tilde{s}_n^{(j)}]}{m \sum_{n=1}^N W_{Nn}(\theta^{(t)})} \\ \hat{\Sigma}_{\theta^{(t)}} &= \frac{\sum_{i=1}^N [W_{Nn}(\theta^{(t)}) \sum_{j=1}^m (\tilde{s}_n^{(j)} - \hat{\mu}_{\theta^{(t)}})(\tilde{s}_n^{(j)} - \hat{\mu}_{\theta^{(t)}})^T]}{m \sum_{i=1}^N W_{Nn}(\theta^{(t)})}. \end{aligned}$$

  $\hat{h}(\zeta^*) = \mathcal{N}(s_0; \hat{\mu}_{\zeta^*}, \hat{\Sigma}_{\zeta^*})$.

  $\hat{h}(\theta^{(t)}) = \mathcal{N}(s_0; \hat{\mu}_{\theta^{(t)}}, \hat{\Sigma}_{\theta^{(t)}})$.

  Calculate $\alpha = \min\left\{1, \frac{p(\zeta^*)\hat{h}(\zeta^*)\mathcal{N}(\theta^{(t)}; \tilde{\mu}, \tilde{\Sigma})}{p(\theta^{(t)})\hat{h}(\theta^{(t)})\mathcal{N}(\zeta^*; \tilde{\mu}, \tilde{\Sigma})}\right\}$.

  Generate independent $U \sim \mathcal{U}(0, 1)$.

  if $U \leq \alpha$ **then**

    $\theta^{(t+1)} = \zeta^*$.

  else

    $\theta^{(t+1)} = \theta^{(t)}$.

  end if

end for