Monte Carlo methods for intractable and doubly intractable density estimation

Ivis Kerama

A thesis submitted for the degree of
Doctor of Philosophy

Department of Mathematics and Statistics,
University of Reading
November 2022

Declaration of Authorship

I confirm that this is my own work and the use of all material from other sources has been properly and fully acknowledged.

Ivis Kerama

Acnowledgements

First and foremost I would like to thank Richard Everitt for his patience and guidance throughout by PhD. His calm demeanour as well as insight into the many topics of our discussions made my journey through the last 4 years greatly enjoyable. I would also like to thank Tom Thorne for coming aboard as an advisor on my 3rd year and for providing constant guidance throughout the last chapter of the thesis as well as extremely useful comments and discussion about applications and the algorithm itself. For my journey through graduate level mathematics and research I would like say an immense thank you to Horatio Boedihardjo for supervising me on my MRes, indirectly teaching me how to read and write at a graduate and research level in pure mathematics, as well as how to be rigorous and perform at the highest level.

I would also like to thank the members of Richard's lab at Reading for the very enjoyable and useful reading groups and discussions. Special thanks to Felipe for our many discussions and collaboration on the 3rd chapter of this thesis. Additionally, I would like to thank the staff of Mathematics of Planet Earth CDT for the substantial support throughout the years of my doctoral studies, as well as my fellow students for contributing to a great working environment. Lastly, I would like to thank my friends for always supporting me and for our endless discussions that encompass the entire world. The last and most certainly greatest thank you, would have to be to my parents. For their constant support, infinite love and for allowing me to be the very best version of myself.

This work would not be possible without the financial support of the Engineering and Physical Sciences Research Council Centre for Doctoral Training in the Mathematics of Planet Earth.

Abstract

This thesis is concerned with Monte Carlo methods for intractable and doubly intractable density estimation. The primary focus is on the likelihood free method of Approximate Bayesian inference, where the presence of an intractable likelihood term necessitates the need for various approximation procedures. We propose a novel Sequential Monte Carlo based algorithm and demonstrate the significant efficiency (computational and statistical) improvements compared to the widely used SMC-ABC, in numerical experiments for a simple Gaussian model and a more realistic random network model. Further, we investigate a recently proposed algorithm, called SAMC-ABC, an adaptive MCMC algorithm where we also demonstrate some advantages over ABC-MCMC; primarily in the reduction of variance of the estimated means although at a cost of increased bias for which we propose a potential correction. In addition, we provide theoretical guarantees of ergodicity and convergence of another newly proposed algorithm termed Adaptive Noisy Exchange, that is aimed at problems of intractable normalising constants where regular MCMC cannot be employed. Finally, we propose potential improvements and future research directions for all of the considered algorithms.

Contents

List of Figures

List of Algorithms

Chapter 1

Introduction

1.1 Introduction and main contributions of this work

A great deal of problems in the real world amount to processing some kind of information that we (usually) record from a number of different processes or occurrences. In statistics we would like to be able to infer something from that data that would allow us to formulate some pattern or even explain why it is so and not some other way. We build a priori models in order to make sense of the data, and of course we modify or completely reconstruct those models when they don't agree with what we are observing. Statistical science has through the centuries tried to make sense of the combination of data and models through rigorous mathematical formulations and theory. A branch which is uniquely equipped to deal with the introduction of new information is that of Bayesian analysis. In that framework, we treat both the data, but also the unknowns of the model specification as variables and try to perform inference on those variables in order to generate a model which supports our data. Equipped with such a model we can then even try to predict future occurrences given past information (generated or observed), assuming of course our model is close enough to reality. All these methodologies emanate from the assumptions that we can explicitly calculate certain integrals of interest, that

naturally come up as ingredients and desired quantities of our models, or at the very least approximate them. It is exactly this procedure which will occupy us for the entirety of this thesis. Namely, the approximation of so called posterior distributions: statistical distributions of the parameters of interest given our data. There is, nevertheless, already an explicit assumption in all of this; we have already specified the form and what kind of distribution our model should have. We are subsequently trying to infer the parameters that define it and perhaps even try to compare different models. In many scenarios such an explicit specification is not possible. There is no real functional form for which we can define our functions and integrals explicitly or such calculation is be prohibitively expensive in computational terms. It is these scenarios that we will explore, propose and validate novel algorithms for which we hope will allow practitioners to utilise in all of the scientific cases they might encounter for which these algorithms might be of use.

1. In chapter 2 we investigate the recently proposed algorithm by Richards and Karagiannis [2020] and perform numerical experiments demonstrating its advantages over standard ABC-MCMC algorithms as well as suggest improvements based on post-correcting the acquired MCMC samples.

2. In chapter 3 we give an overview of auxiliary space methods and their noisy variants that are utilised in cases where one has intractable normalising constants and regular MCMC methods cannot be employed. Further, we investigate theoretically a recent proposal for a novel algorithm: Adaptive noisy exchange algorithm by Friel and Drovandi [2019] (and communicated personally through the supervisory team) where we prove the convergence of the algorithm under very mild assumptions to the correct "noisy" target, thus validating its use in practice.

3. In chapter 4 we introduce the novel algorithm termed Rare event ABC-SMC^2. We formulate its structure, perform numerical experiments suggesting a significant improvement over standard ABC SMC and propose modifications as well as potential expansions on its constituent parts.

1.2 Motivational examples and purpose of statistical inference

In the Bayesian framework of statistics we model the observed data as well as any unknowns as random variables. Taking this approach, we, through some function $f \in \mathcal{F}$, specify the distribution $f(\mathbf{y}|\boldsymbol{\theta})$ for the observed data $\mathbf{y} = (y_1, ..., y_n) \in \mathcal{Y} \subseteq \mathbb{R}^n$, with dimension $n \in \mathbb{N}$, given a vector of unknown parameters $\boldsymbol{\theta} = (\theta_1, ..., \theta_k) \in \boldsymbol{\Theta} \subseteq \mathbb{R}^d$ of some dimension $d \in \mathbb{N}$. By this statistical model we define the Law [1]. We also assume that θ is a random quantity assigned a prior distribution $\pi(\boldsymbol{\theta}|\boldsymbol{\eta})$, where $\boldsymbol{\eta} \in \mathbf{E}$ is a vector of hyperparameters. The goal of our inquiry here is the "posterior" distribution, where it is defined as a conditional distribution given the observations and calculated through Bayes's theorem as follows:

$$p(\boldsymbol{\theta}|\mathbf{y}, \boldsymbol{\eta}) = \frac{p(\mathbf{y}, \boldsymbol{\theta}|\boldsymbol{\eta})}{p(\mathbf{y}|\boldsymbol{\eta})} = \frac{p(\mathbf{y}, \boldsymbol{\theta}|\boldsymbol{\eta})}{\int p(\mathbf{y}, \boldsymbol{\theta}|\boldsymbol{\eta}) d\boldsymbol{\theta}} = \frac{f(\mathbf{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}|\boldsymbol{\eta})}{\int f(\mathbf{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}|\boldsymbol{\eta}) d\boldsymbol{\theta}}^{2} \tag{1.1}$$

For a number of models in various fields the likelihood $f(\cdot)$ is intractable in the sense that one cannot evaluate $f(\cdot) \approx l(\mathbf{y}|\boldsymbol{\theta})$. For example the majority of stochastic differential equation models such as the (stochastic) Lotka-Volterra model Wilkinson [2013a], most individual based models (IBMs) Grimm and Railsback [2005], and many other models defined through computer simulations. Approximate Bayesian computation (ABC) (Tavaré et al. [1997], Pritchard et al. [1999], Beaumont [2003]) is a class of likelihood free inference methods that is utilised to perform approximate inference for the parameters of such models, as indicated for example in (Toni et al. [2009]). An excellent review of the ABC in Ecology and Evolution is given by Beaumont [2010], and its use for IBMs has recently been explored in (van der Vaart et al. [2015], van der Vaart et al. [2016].

[1] we will hereafter assume that this always admits a density with respect to some reference measure dy of the random variables $(Y_n)_{n=1}^N$, which we usually call the likelihood

[2] a technical requirement here will be that of the finiteness of the denominator, in order for this posterior to be well-defined

The power of Bayesian inference was not embraced immediately for a number of reasons. It would, nevertheless, be mandatory to say that the true resuscitation of Bayesianism as a philosophy of inference came from the rigorous work by de Finetti de Finetti [1974], L. J. Savage Savage [1954], D. V. Lindley Lindley [1965] and George Box Box and Tiao [1973] between the 50s and 80s, yet it remained more or less impractical. The advent of powerful computers and specifically the application of Monte Carlo methods combined with the well developed theory of Markov chains to certain classes of models and the work by the brilliant team at Los Alamos, Metropolis et al. [1953b] kick-started the entire field of computational statistics (although not for quite some time afterwards-it was the exponential improvement of personal computers/workstations and specific software suites that made the usage of these methods extremely widespread).

Since Metropolis et al. [1953b] original paper and especially the more formal treatment and introduction of the Hastings correction by Hastings [1970] in 1970 there has been a rapid expansion of the volume of work on the field of computational statistics. An important milestone for the generality and breadth of applicability of the methods was the introduction of reversible jump MCMC Green [1995], where one could transverse different dimensionality models and do inference on a varying-dimension model space. It is therefore no wonder that Markov Chain Monte Carlo methods are the most successful methods in Bayesian practice and inference today. One could argue that this is due to their (theoretical) ability in enabling inference from arbitrary complex distributions of correspondingly arbitrary large dimensionality. In the same spirit as regular Monte Carlo methods MCMC produces correlated samples from a distribution of interest rather than a calculation of the its closed integral form [3]. An important question therefore arises given that we are producing a finite sample estimate (as we will soon explain) of some arbitrary integral; do we have some notion of approximation in quantitative terms ? In other words, can we state in some sense exactly how many samples are needed in order to achieve some arbitrary degree of accuracy for our approximation ? Assume for example that we would like to estimate some posterior π and through the

[3] a calculation for which in many if not most real-world scenarios and the integrals of interest would otherwise be impossible if otherwise completely computationally infeasible; for example with quadrature methods

MCMC procedure, which we shall explain in detail in section 1.3.4 and 1.3.5. we apply some Markov kernel P, n times. We would like to calculate how close our approximation is to our intended posterior measure in total variation distance [4]:

$$\|P_x^n - \pi\|_{\mathrm{TV}} = \frac{1}{2}\sum_y |P^n(x,y) - \pi(y)| = \max_{A\subseteq\mathcal{X}} |P^n(x,A) - \pi(A)| \tag{1.2}$$

Which yields the following problem: Given P, π, x and $\epsilon > 0$, how large n so

$$\|P_x^n - \pi\|_{\mathrm{TV}} < \epsilon \tag{1.3}$$

Which for the practitioner equates to the questions of how long should the algorithm run for and how many samples are sufficient (a thing which itself needs specification).

Can we formulate a function about the degree for which this approximation is appropriate/close to the analytical solution based on the number of samples we have ? There exists a great deal of work on the asymptotics of these methods, the rate of convergence, dependence on dimensionality and optimal rates for acceptance, how one can reduce the variance of the estimator systematically, what is the optimum of that (if it exists) etc. Theory tells us that increasing the number of samples, say N, makes the approximation more accurate (the Monte Carlo error decreases as$\frac{1}{\sqrt{N}}$ with N the number of samples or steps of the algorithm - if a Markov Chain Central Limit theorem exists with the usual result being Birkhoff's Ergodic Theorem Geyer [2005]. Despite getting exact asymptotic results a lot of things are essentially hidden in that error rate: see Jones [2004], Flegal et al. [2008]Flegal and Jones [2010],Jones et al. [2006], Vats et al. [2019]). The issue of metastability[5] of Markov chains Beltrán and Landim [2015], Landim [2019] and that of knowing where to stop the algorithm since in a very rough sense it has sample "**all** the important

[4]Given some measurable space $(\Omega, \mathcal{F})$ and probability measures P and Q defined on $(\Omega, \mathcal{F})$. The total variation distance between P and Q is defined as $||(P,Q)||_{TV} = \sup_{A\in\mathcal{F}} |P(A) - Q(A)|$

[5]essentially the late long time behaviour of the chain and specifically the repeated passage from areas of low to high probability, for technical definitions albeit applied to continuous time chains and the substantially developed theory there see Olivieri and Vares [2005]

areas of density" in the characteristic set sense remains elusive. For a survey of theoretical results and outstanding issues see Freidlin and Koralov [2017], Olivieri and Vares [2005], Bovier et al. [2001],
Bovier et al. [2002], Cassandro et al. [1984].

Unfortunately, by construction and the very nature of Markov chain theory and contrary to regular Monte Carlo methods, given the iterative nature of the algorithm the produced samples are correlated. The method is extremely well studied and very mature. The existence of the invariant density of the chain and the convergence of the produced ergodic averages to this density can be shown to exist for a gigantic range of posterior classes (see Roberts and Rosenthal [2004] for a general overview, and Roberts and Rosenthal [2016] for adaptive cases as well as Meyn and Tweedie [2009], especially the concluding chapters [13-19] for an in depth look at the underlying theory and detailed commmentary on general convergence) and in that sense it offers a satisfactory explanation of the method's popularity. Alas, the issue of convergence remains due the simple fact alluded above: the user has to choose the number of iterations of the algorithm and perform relevant statistics on the output; an area of research which considerable effort has been put into. Additionally, the finite -computational resource constrained- samples make the user wary of the quality/accuracy of any given output of the chain since they are not independent and identically distributed samples from the posterior, but rather correlated ones. Sometimes called the variance estimation problem, this issue is of considerable practical importance since we would also like to estimate the Monte Carlo variances (or equivalent standard errors) associated with the MCMC generated posterior estimates. Nevertheless, at this point it would be prudent to give a brief tour of how and why Monte Carlo methods have culminated, at least in one of their branches, in this extremely powerful class we call Markov Chain Monte Carlo. Nevertheless, let us take a step back and see how these methods came about and go through various levels of sophistication as far as approximations of integrals are concerned.

1.3 Monte Carlo

1.3.1 General approximations of integrals

One could consider, very roughly speaking, the idea of Monte Carlo as that of the approximation of arbitrary integrals. The integrals considered would be impossible to calculate otherwise both due to the dimensionality but also due to the functional form and in most cases due to a mixture of both. We are therefore left with the issue of how one can go about approximating those integrals that escape analytical close-form solutions. Let us then see where Monte Carlo methods are situated compared to others and why it is in many ways preferable if not essential. For example see Heinrich and Novak [2002] for an overview of some interesting results regarding the optimality of deterministic, randomized and quantum algorithms (the last of which wont be of concern in the present work) from Hölder or Sobolev[6] spaces (see definition in Adams and Fournier [2003]). Some of the first results were derived in Bakhvalov [1959], Bakhvalov [1962]. An important result from these papers is the fact that randomized algorithms perform better than their deterministic counterparts and more importantly a near optimal convergence rate is exhibited by Monte Carlo methods when then dimension is large or when the integrand exhibits low smoothness (as defined in Heinrich and Novak [2002]). Despite such results, Monte Carlo methods are not always appropriate. For example using an MCMC or SMC method wont give us independent samples (in distribution) from our intended target. Furthermore the approximation will become worse as the dimensionality increases for a fixed sample size. Further, consider the fact that the Monte Carlo method error rate is[7] $1/\sqrt{N}$, and the dimension d usually appears in a constant of proportionality that impacts the actual implementation and performance of the algorithms in all real world applications. The theoretical rates derived in the work above become less useful or relevant. Results closer to the practical usage of these methods include for example Roberts and Rosenthal [2001], Neal and Roberts [2006], Yang et al. [2020] for

[6] Complete normed (as a combination of L^p norms of the functions) vector spaces of functions with weak derivatives (in the sense of no assumption on the availability of differentiable functions but only integrable)

[7] always

the Metropolis-Hastings algorithm and its behaviour as the dimension of the state space increases and recipes of how the algorithm should be configured while taking that into account. Also for the pseudo-marginal algorithm Andrieu and Roberts [2009] which we will explain shortly Sherlock et al. [2015]. For SMC method see Beskos et al. [2014a] and Beskos et al. [2014b], and in particular for importance sampling which will be directly relevant to one of the proposed algorithms in this thesis see Agapiou et al. [2017].

1.3.2 Monte Carlo methods

Suppose we would like to compute an integral of the form:

$$I_g = \int_{\mathcal{Y}} h(\mathbf{y}) d\mathbf{y} \tag{1.4}$$

with h some general function on some space $\mathcal{Y}$, with the condition that the above integral is finite. Consider a probability density function g and a function f such that

$$\mathbb{E}_g[f(\mathbf{y})] = \int_{\mathcal{Y}} f(\mathbf{y}) g(\mathbf{y}) d\mathbf{y} = I_g \tag{1.5}$$

Assume that we can also obtain samples $Y_1, ..Y_N$ from g, then a Monte Carlo estimator is the sum defined as :

$$\frac{1}{N} \sum_{i=1}^{N} f\left(Y_i\right) \tag{1.6}$$

and by the Law of Large numbers one could get that

$$\frac{1}{N} \sum_{i=1}^{N} f\left(Y_i\right) \underset{N \to \infty}{\overset{g}{\longrightarrow}} I_g$$

assuming the integral exists. Assuming further that $\sigma^2 = \mathbb{V}_g[f(\mathbf{y})] = \mathbb{E}_g[f^2(\mathbf{y})] - I_g^2$ is finite, then the Central Limit theorem gives and even stronger result:

$$\sqrt{N}\left(\sum_{i=1}^{N} f\left(Y_i\right) - I_g\right) \xrightarrow[N\to\infty]{\mathcal{L}} \mathcal{N}\left(0, \sigma^2\right) \tag{1.7}$$

where we also know that the estimator above converges to I_g at a rate of $\mathcal{O}(N^{-1/2})$

There are various methods one could use to generate the required samples: transformation of random variables, rejection sampling, importance sampling, Markov Chain Monte Carlo, Sequential Monte Carlo etc.

1.3.3 Importance sampling

Importance sampling aims to generate samples from a distribution of interest g by rewriting the expectation of 1.5 by introducing an auxiliary distribution q and rewriting as :

$$\mathbb{E}_G[f(\mathbf{y})] = \mathbb{E}_Q[f(\mathbf{y})w(\mathbf{y})] \tag{1.8}$$

with q absolutely continuous with respect to g and where we call $w(\mathbf{x}) = \frac{dg}{dq}(\mathbf{x})$ the importance weight (which can be easily seen to be a Radon-Nikodym derivative). By the LLN the integral I_g can be approximated by

$$\frac{1}{N}\sum_{i=1}^{N} f\left(Y_i\right) w(Y_i), \tag{1.9}$$

where $Y_i \sim q$. The advantage of such approach is the fact that we can calculate the weights up to multiplicative constant and estimate them by

$$\frac{1}{N}\sum_{i=1}^{N} w(Y_i) \tag{1.10}$$

and additionally show that the normalised estimator

$$\frac{\sum_{i=1}^{N} f\left(X_i\right) w\left(Y_i\right)}{\sum_{i=1}^{N} w\left(Y_i\right)} \tag{1.11}$$

converges to $\mathbb{E}_g$ The trick in importance sampling is to notice that once you can re-express the expectation

$$\mathbb{E}_g[f(y)] = \frac{\int_{\mathcal{Y}} f(y)\frac{g(y)}{q(y)}q(y)\mu(\mathrm{d}y)}{\int_{\mathcal{Y}} \frac{g(y)}{q(y)}q(y)\mu(\mathrm{d}y)} = \frac{\frac{\int_{\mathcal{Y}} f(y)\frac{g(y)}{q(y)}q(y)\mu(\mathrm{d}y)}{\int_{\mathcal{Y}} q(y)\mu(\mathrm{d}y)}}{\frac{\int_{\mathcal{Y}} \frac{g(y)}{q(y)}q(y)\mu(\mathrm{d}y)}{\int_{\mathcal{Y}} q(y)\mu(\mathrm{d}y)}} = \frac{\mathbb{E}_q\left[\frac{g(y)}{q(y)}f(y)\right]}{\mathbb{E}_q\left[\frac{g(y)}{q(y)}\right]} \tag{1.12}$$

with potentially unnormalised g and q. The expression above demonstrates that by generating samples $Y_1, ..Y_N$ from some procedure (Monte Carlo simulation or MCMC) we obtain

$$\widehat{\mathbb{E}_g\left[f(Y)\right]} = \frac{\widehat{\mathbb{E}\left[\frac{g(Y)}{q(Y)}f(Y)\right]}}{\mathbb{E}_q\left[\frac{g(Y)}{q(Y}\right]} = \frac{\sum_i^N w_i f(Y_i)}{\sum_i^N w_i} = \frac{\frac{\sum_i^N \frac{g(Y_i)}{q(Y_i)}f(Y_i)}{N}}{\frac{\sum_i^N \frac{g(Y_i)}{q(Y_i)}}{N}} \tag{1.13}$$

The ratios $w_i = \frac{G(y_i)}{Q(y_i)}, i = 1, \ldots, N$, are called importance weights.

1.3.4 Markov chain Monte Carlo

Markov chain Monte Carlo methods have a long history Metropolis et al. [1953a], Smith and Gelfand [1992], Tierney [1998], Roberts et al. [1998], Roberts and Rosenthal [2004] , and the general idea was more or less developed at the same time as the first Monte Carlo methods. The method works by specifying a target density measure μ (appropriately defined distribution that is absolutely continuous with respect to some measure ν) on some space $\mathcal{S}$. The target one is interested in is usually rather complex, and not really amenable to the simple Monte Carlo methods. The complexity comes in various forms, often simultaneously. It can be that the integral we would like to evaluate is high dimensional and it cannot be solved by quadrature methods for example

given their exponential scaling. It can also be the case that the size of the data that one considers or the dimension of the inference target/integral and general space one is working is particularly high. Finally, a limited computational budget allows only methods that run within some time frame. It is therefore obvious that a more efficient methodology than simple Monte Carlo simulation is needed if we are to tackle all these issues. MCMC methods work by simulating a Markov Chain that samples more efficiently from the target space of interest than regular MC methods. The extremely interesting main idea of the method works by defining an appropriate Markov process that has as invariant measure given by the target density of interest and specifically for discrete times, for which our defined Markov chain, say Y_n has as its stationary distribution. Then given certain conditions such as *irreducibility* and *aperiodicity*[8] , Meyn and Tweedie [2009] a limiting distribution exists for that process, it is unique and is its stationary distribution. Therefore, samples from that process are asymptotically distributed according to the measure of interest, and hence we can construct ergodic averages given the number of time steps or samples , by running the chain for a long period of time. The basic Law of Large Numbers for the MCMC algorithm informs us about those averages in the following way:

Theorem 1. *(Ergodic Theorem (Robert and Casella [2004])) If $(Y_n)_{n\geq 0}$ is a positive Harris recurrent Markov chain with invariant measure P, then for every $h \in L_1(P)$, we have*

$$\frac{1}{N}\sum_{i=1}^{N} h\left(Y_i\right) \underset{N\to\infty}{\longrightarrow} \int h(y)P(dy) \tag{1.14}$$

as well as a corresponding Central Limit Theorem:

[8]aperiodicity is not strictly speaking necessary for the existence of ergodic averages Roberts and Rosenthal [2004], corollary 6

Theorem 2. *(Markov Functional Central Limit Theorem (Robert and Casella [2004])) If* $(Y_n)_{n\geq 0}$ *is a positive Harris recurrent and irreducible Markov chain, geometrically ergodic with invariant measure* P*, and if the function* h *satisfies* $\mathbb{E}_P[h(Y)] = 0$ *and* $\mathbb{E}_P\left[|h(Y)|^{2+\epsilon}\right] < \infty$ *for some* $\epsilon > 0$*, the we have*

$$\frac{1}{N}\sum_{i=1}^{N} h\left(Y_i\right) \xrightarrow[N\to\infty]{\mathcal{L}} \mathcal{N}\left(0, \sigma_h^2\right)$$

for some finite $\sigma_h^2 = \mathbb{E}_P\left[h\left(Y_0\right)^2\right] + 2\sum_{k=1}^{\infty}\mathbb{E}_P\left[h\left(Y_0\right)h\left(Y_k\right)\right] < \infty$.

Various other conditions such as *drift* Meyn and Tweedie [2009] also exist for which if one verifies their presence one could also make statements about the speed (geometrically fast, although there are other rates of convergence as well: polynomial, uniform etc Meyn and Tweedie [2009]) at which the chain converges in a certain sense to the target density of interest Mengersen and Tweedie [1996], and more importantly also establish CLT such as the one in Roberts and Rosenthal [2004].

1.3.5 The Metropolis-Hastings algorithm

The Metropolis-Hastings method is an algorithmic implementation of the theoretical idea described above first appearing in Metropolis et al. [1953b] where the implementation was with a symmetrical proposal distribution (we will see shortly what that means) and later extended to the more general case by Hastings [1970]. The basic algorithm proceeds as follows: Given a current point in our sample space Y_t a new point gets picked according to some arbitrary proposal distribution $q(y'|y_t)$ (that defines a Markov Kernel), which results in

a Markov Chain $\{Y_i\}_{i=1}^{\infty}$ that is the result of the following algorithm:

Algorithm 1: Metropolis-Hastings algorithm

Input: starting point Y_0, proposal distribution q and number of iterations t

for *t=1:N* **do**
- Sample $Y' \sim q(\cdot \mid Y_{n-1})$
- Calculate the acceptance probability $\alpha(Y_{n-1}, Y')$, where

$$\alpha(Y_{n-1}, Y') = \min\left\{1, \frac{p(Y')\, q(Y_{n-1} \mid Y')}{p(Y_{n-1})\, q(Y' \mid Y_{n-1})}\right\}$$

- Sample $U \sim \mathcal{U}[0,1]$
- **if** $U < \alpha(Y_{n-1}, Y')$ **then**
  - then $Y_n \leftarrow Y'$
- **else**
  - $Y_n \leftarrow Y_{n-1}$
- **end**

end

One should notice the importance of the accept-reject step in the algorithm above, as given appropriate proposals such chains are irreducible (a positive probability of visiting every point in the space in some finite number of iteration) and aperiodic (there is no cyclicity in the moves of the sample space) and as we mentioned those two conditions result in the chain converging to the stationary distribution irrespective of the starting point, and furthermore given also an appropriate function of the acceptance probability (for example MH or barker's acceptance Barker [1965]) the chains are reversible with respect to the target density p. It is also worth mentioning that usually such conditions need to be checked on an individual case basis. It is important to also note that by definition the algorithm does not produce independent samples given its Markovian structure, although there are ways one considers to what extent the samples that are acquired can be considered i.i.d. from the target of interest. A very large number of iterations might be necessary to escape the initial point's neighbourhood and actually sample from the main mass of the target measure. Usually practitioners refer to this as a burn-in period where those initial samples (usually chosen to be some percentage of the total number of samples) are discarded from the final ones. Lastly it is worth pointing out that there is quite a substantial variability of the performance characteristics of this general class of algorithms dependent on the choice of proposal q, the dimensionality of the sample space, the characteristics of the

target distribution (multimodality, shape) and so on.

1.3.6 Sequential Monte Carlo

Another importance class of algorithms is that of sequential Monte Carlo methods. As the name implies they were initially conceived for sampling from a dynamic distribution where new values are added sequentially in the target distribution, as is for example the case of some stochastic process (for example measurement of the position of some object in motion) that is evolving in real time. MCMC algorithms are by design appropriate for sampling from static distributions, as is the posterior of some parameter vector for example $p(\theta \mid y_{1:n})$ with no new $y_{n+1,\dots}$ updated/included later on, and furthermore are also rather costly as at every iteration one requires to incorporate all the available observations. We therefore need some sequential algorithm that incorporates observations as they come in and is able to produce ergodic averages while targeting the correct target distribution. This is the case with the issue of filtering, where one faces in a number of different scenarios such as the need to track some moving object Brasnett et al. [2005], Gustafsson et al. [2002], Doucet and Johansen [2011], inferring stock price movements, epidemic tracking etc. The main idea of a particle filter is one where we could approximate the target at every observation time as a collection of point masses, called particles, weighted appropriately, such that by a process of mutation, correction and resampling one could take into account new observations and deal with the sequential nature of the target distribution. These general class of methods, can therefore allows us to construct the corresponding ergodic averages by Monte Carlo approximations of the expectations of interest defined with respect to the target measures one wishes to consider, by a ratio of unbiased estimators much in the same spirit as importance sampling. In fact as we shall see at every iteration of these algorithms the weights of these particles is constructed as such ratios Doucet et al. [2001], Cappe et al. [2006], Doucet and Johansen [2011], Del Moral [2004]. It is also exceedingly useful in order to be able to get a sense for the correctness of the samples we have that some form of CLT is established. Under certain assumptions and conditions on the filtering model and the mutation step as well as the resampling step the ergodic

averages do indeed satisfy a form of CLT as provided in Del Moral [2004], Del Moral and Miclo [2000], Del Moral and Guionnet [1999], Chopin [2004].

Let us now describe how these algorithms operate. Assume that one has some process Y_t generated through some Markovian dynamics. Additionally assume those have some *transition* density $f(y_{t+1}|y_t)$, and we additionally have some initial draw from some prior $y \sim f_0$, for a continuous density f_0. Further, suppose we have some noisy observations (and perhaps partial) of this chain denoted by $X_{1:\infty}$ which are independent given $Y_{1:\infty}$, and that admit some *observation* density $g(x_t|y_t)$. We would now like to infer at any time (or prior to that) instance t the posterior of those observations. An immediate calculation via Bayes theorem gives us the following:

$$p_t(y_t) = \frac{g_t\left(x_t \mid y_t\right) \int_{\mathbb{R}^d} f_t\left(y_t \mid y_{t-1}\right) p_{t-1}\left(y_{t-1}\right) \mathrm{d}y_{t-1}}{\iint_{\mathbb{R}^{2d}} g_t\left(x_t \mid y_t\right) f_t\left(y_t \mid y_{t-1}\right) p_{t-1}\left(y_{t-1}\right) \mathrm{d}y_{t-1}\ \mathrm{d}y_t} \tag{1.15}$$

and[9] therefore we now want to approximate some expectation:

$$p_t[z] = \int_{\mathbb{R}^d} z(y_t) p_t(y_t) dy_t \tag{1.16}$$

in an online manner through updates in observations (at the time they arrive).

Particle Filters

We therefore want to approximate integrals like the one shown in 1.16, satisfying some conditions and having certain properties. Now one can observe that we can state equation 1.15 equivalently as

[9] note here we have suppressed the conditioning on the data $x_{1:t}$

$$p_t(y_t) = \frac{1}{\beta_t(\mathbb{R}^{dt})} \int_{\mathbb{R}^{d(t-1)}} f_0(y_0) \prod_{s=1}^{t} f_s(y_s \mid y_{s-1}) g_s(x_s \mid y_s) \, dy_{0:t-1} \tag{1.17}$$

with the expectation $p_t[z]$ written alternatively as:

$$p_t[z] = \frac{1}{\beta(\mathbb{R}^{dt})} \int_{\mathbb{R}^{dt}} z(y_t) \prod_{s=1}^{t} f_s(y_s \mid y_{s-1}) g_s(x_s \mid y_s) \, \mathrm{d}x_{0:t} \tag{1.18}$$

with $\beta()$ a normalizing measure. We can now view this as a natural dynamic procedure of the (normalized) importance sampling estimator of the previous section. Here we have some population of particles $y_t^{1:N}$ that evolve according to some dynamics f and where their weights $\hat{w}_t^{1:N}\left(y_t^{1:N}\right)$ are updated sequentially in order to accommodate the evolution of the sequence of target densities p_t

Sequential Importance Sampling

One way of constructing an estimator for $p_t[z]$ is to construct a normalized importance sampling estimator of it. Let $q_t(y_{0:t})$ be some proposal density that can be dependent on the number of observations $x_{1:\infty}$ (although not necessarily) or any subset of those. At any time t, one can calculate the importance weights as :

$$w_t(y_{0:t}) := \frac{\beta_t(y_{0:t})}{q_t(y_{0:t})} = \frac{g_t(x_t \mid y_t) f_t(y_t \mid y_{t-1}) q_{t-1}(y_{0:t-1})}{q_t(y_{0:t})} w_{t-1}(y_{0:t-1}) . \tag{1.19}$$

although one notices that using that as a general proposal is fairly inefficient from a computational standpoint since simulating the path $y_{0:T}$ and calculating its likelihood through $q()$ will impose a heavy computational burden. Therefore it would be much preferred if we could decompose $q(\cdot)$ (and we can given the assumption of Markov Property) and do so sequentially:

$$q_t\left(y_{0:t}\right)^{10} = f_0\left(y_0\right)\prod_{s=1}^{t} f_s\left(y_s \mid y_{0:s-1}\right) \tag{1.20}$$

as we can then observe that the importance weights w now satisfy a recursion of the form:

$$w_t\left(y_{0:t}\right) = \frac{g_t\left(x_t \mid y_t\right) f_t\left(y_t \mid y_{t-1}\right)}{p_t\left(y_t \mid y_{0:t-1}\right)} w_{t-1}\left(y_{0:t-1}\right) \tag{1.21}$$

We can therefore see that the normalised importance weights

$$\sum_{i=1}^{N} \hat{w}_t^{(i)}\left(Y_{0:t}^{(1:N)}\right) z\left(Y_t^{(i)}\right) \tag{1.22}$$

and the (normalized) importance sampling estimator

$$\hat{w}_t^{(i)}\left(Y_{0:t}^{(1:N)}\right) = \frac{w_t\left(Y_{0:t}^{(i)}\right)}{\sum_{j=1}^{N} w_t\left(Y_{0:t}^{(j)}\right)} \tag{1.23}$$

can be iteratively updated. Assuming one has $\operatorname{supp}\left(\beta_t\right) \subseteq \operatorname{supp}\left(q_t\right)$, then this importance sampling estimator has the same properties as the normalized importance sampling one. Of course there is a certain obvious price that we have to pay for the benefit such algorithm brings: it is evident that the products of 1.21 will result in a rapid increase in variance as one could calculate by taking the variance of the weights at time t and at time $t+1$ and therefore estimators of this nature will exhibit large variance. This is commonly referred to in the literature as a path degeneracy problem. In order to address this issue Gordon et al. [1993], Kitagawa [1987] proposed the sequential importance resampling algorithm and in a more general sense the sequential Monte Carlo algorithm class as described in the algorithm 2 of the next page. It is essentially a combination of SIS and resampling. In this algorithmic improvement one resamples the particles according to their normalised weights $\hat{w}_t^{1:N}$ resulting in an alleviation of the degeneracy issue, by throwing away particles with small weights and replicating those with large weights (relative to each other). One could summarise the improvement here by noting that we essentially propose new particles through some proposal q by incorporating the information provided

[10] this is one possible choice of the proposal function $q(\cdot)$. One could of course use the prior as the proposal $q_1 = \kappa(y_1), q(y_t \mid y_{t-1}) = f(y_t \mid y_{t-1})$ which would result in the simplified weights being equal to $g(x_t \mid y_t)$. This is nonetheless a sub-optimal choice.

by the empirical measure of the approximation of our target at time t. In other words we propose from $\widehat{\gamma}_1(y_1)\, q_2(y_2 \mid y_1)$, rather than from $q_1(y_1)\, q_2(y_2 \mid y_1)$ as in sequential importance sampling, with $\widehat{\beta}$ the approximation of β_t at time t in algorithm 2. It is worth mentioning that this, nevertheless, does not eliminate the issue of particle degeneracy since no original genealogies (or paths) of the particles are created, just replicated ones. Resampling usually comes in different flavours, such as stratified, systematic, residual and multinomial. Residual resampling exhibits lower variance than multinomial Liu and Chen [1998]. Further, Carpenter J et al. [1999] show that stratified resampling has minimum variance as an unbiased resampling technique. Any choice of the resampling techniques mentioned above *will* immediately increase the variance of the estimator employed, yet since the goal is to decrease the path degeneracy it can be said that for larger time horizons it will reduce the variance in

the estimator overall.

Algorithm 2: Sequential Monte Carlo (general) Algorithm

if *t=1* **then**
  for $i \in \{1:N\}$ **do**
    sample $Y_1^i \sim q(\cdot)$, the compute weights [11]:

$$w_1^i = \frac{\gamma\left(y_1^i\right)}{q\left(y_1^i\right)}.$$

    sample the ancestral (at time $t-1$) indices a_t^i of the resampled particles from $h(|\cdot) \in \{1:N\}$ with j^{th} probability $w_1^j \propto w_1^j$, resulting in $\left\{a_1^{1:N}\right\}$. Set normalised weights $= 1/N$
  end
  $t = t+1$
  for $i \in \{1:N\}$ **do**
    sample $Y_n^i \mid x_{1:n-1}^{a(i)} \sim q\left(\cdot \mid y_{1:n-1}^{a(i)}\right)$ and compute the weights:

$$w_n^i = \frac{\gamma\left(y_n^i \mid y_{1:n-1}^{a(i)}\right)}{q\left(y_n^i \mid y_{1:n-1}^{a(i)}\right)}$$

    sample indices a_t^i of the resampled particles from $h(|\cdot) \in \{1:N\}$ with j^{th} probability $w_n^j \propto \hat{w}_n^j$. Set normalised weights $= 1/N$, and set $n := n+1$. Return to the start of Step 2
  end
  .
else
end

An added benefit of the SMC sampling scheme is the fact that it provides estimates of the normalising constants of γ_t, which are in fact unbiased:

$$\hat{\beta}_{,1:t} = \prod_{t=1}^{T}\left[\frac{1}{N}\sum_{i=1}^{N}\hat{w}_t^i\left(x_{1:t}^i\right)\right] \tag{1.24}$$

1.3.7 Sequential Monte Carlo Samplers

Given the motivation behind Sequential Monte Carlo, it would perhaps seems strange to wonder whether one could in fact use SMC like algorithms but not

[11] unormalised

for the explicit purpose of sequential inference, but rather for a more general inference procedure on a rather general space of interest. The authors in Del Moral et al. [2006], Del Moral and Jasra [2007], Del Moral et al. [2007] propose one such methodology called Sequential Monte Carlo Samplers. The intention there is to have some sort of technique that would enable one to perform inference on some generic sequence of target measures in arbitrary spaces that have some relation. In general we would like to perform inference on some general complex target of some form $\omega(dx)$. One could try to implement the algorithms described in the previous section but once the algorithm moves to the next step no update can be performed retrospectively for the previous states, only to new updates (and states). In addition, it is obvious that given the updating nature of the algorithm and incorporation of new observations it operates on an increasing state space (strictly so) and enjoys the conditional independence properties that are very much an integral part of its procedure. It is therefore a critical issue of defining a proper state space where one such sampling procedure might be effectively carried out. The novel idea of the aforementioned works is that of defining a sequence of synthetic distributions that exhibit the required properties.

What we essentially want to do is given our intended target $\omega(dy)$ on some measurable space $(A, \mathcal{A})$, we define a sequence of targets $\omega_{1:n}(dy)$ so that $\omega_n(dy) = \omega(dy)$. An advantage of that is that we have the common measure space $(A, \mathcal{A})$, where those targets are defined upon, instead of a sequence of nested spaces
$\{(E_n, \mathcal{E}_n)\,; E_{n-1} \subseteq E_n\}_{n=1}^m$ as in the regular approach of the previous sections. Assume for now that we would like to have $\omega_{1:n}(dy)$ as our target with a SMC scheme. At some iteration t the particle population $Y_t^{1:N}$ is distributed with importance density $\gamma_{t-1}(dy_{t-1})$ and additionally perturbed by some Markov Kernel $K_t(y_{t-1}, dy_t)$ that has as its density $k_n(y_n|y_{n-1})$; we then have the particle population be distributed (marginally) following the proposal distribution:

$$\gamma_t\left(dy_t\right) = \int_A \gamma_{t-1}\left(dy_{t-1}\right) K_t\left(y_{t-1}, dy_t\right) \tag{1.25}$$

and assume that we can calculate the density $\gamma_t(y_t)$ pointwise. It is the easy to see that one can calculate the importance weights and furthermore the ex-

pections one needs with immediate Importance sampling calculation as before. Nevertheless, reconsider the integral above in an expanded form:

$$\gamma_t(y_t) = \int_{A^{t-1}} \gamma_1(y_1) \left(\prod_{j=2}^{t} k_j(y_j \mid y_{j-1}) \right) dy_{1:t-1} \tag{1.26}$$

It seems like it is impossible to calculate not to mention its high dimensionality. Alternatively, we attempt toe estimate the integral above (1.26) with the following expression:

$$\gamma_{t-1}^N k_t(y_t) := \frac{1}{N} \sum_{i=1}^{N} k_t\left(x_t \mid Y_{t-1}^{(i)}\right) \tag{1.27}$$

The issue with that of course is that the cost of that procedure would scale as $\mathcal{O}(N^2)$ making the cost prohibitive (since the density $\gamma_t(Y_t^j)$ would have to be approximated for all $j = 1 : T$. Additionally, we cannot always calculate $k_t(y_t|y_{t-1})$ pointwise and therefore even if we could tolerate the computation burden, this makes it completely unfeasible otherwise. It is therefore this setting that allows the alternative in the form of a SMC sampler thereby bypassing the need to compute 1.26. The authors then have the idea of an augmented state space that is being facilitated by the existence of a forward and backwards Markov Kernel $\mathcal{K}$ and $\mathcal{L}$ respectively, although the ingenious idea here is that of the backward kernel taking into account the weakness we mentioned before about allowing the past states to be incorporated into the algorithmic setup retrospectively. Del Moral et al. [2006] then use an importance density $\gamma_t(y_{1:t})$ in order to obtain a sample from the augmented joint density

$$\hat{\omega}_t(y_{1:t}) = \frac{\tilde{\eta}_t(y_{1:t})}{Z_t} \stackrel{def}{=} \frac{\eta_t(y_t) \prod_{k=2}^{t} l_{k-1}(y_{k-1} \mid y_k)}{Z_t} \tag{1.28}$$

with marginal $\omega_t(y_t)$ and $\tilde{\eta}_t$ some artificial distribution, while η_t the artificial joint distribution at time t when using the reverse Markov kernels l_t as explained in Del Moral et al. [2006]. We can now see that every particle path $\left\{Y_{1:t-1}^{(i)}\right\}$ at iteration t is carried forward with the Markov kernel $K_t(y_{t-1}, dy_t)$ and consequently given weights $W_y^{(i)} \propto w\left(Y_{1:y}^{(i)}\right)$, with the function $w(y_{1:y})$,

that measures the discrepancy of $\hat{\eta}_t(x_{1:t})$ and $\gamma_t(x_{1:t})$:

$$\begin{aligned} w(y_{1:t}) &= \frac{\tilde{\eta}_t(y_{1:t})}{\gamma_t(y_{1:t})} \\ &= \frac{\tilde{\eta}_{t-1}(y_{1:t-1})\,\eta_t(y_t)\,l_{t-1}(y_{t-1} \mid y_t)}{\eta_{t-1}(y_{t-1})} \cdot \frac{1}{\eta_{t-1}(y_{1:t-1})\,k_t(y_t \mid y_{t-1})} \\ &= w(y_{1:t-1}) \cdot \frac{\eta_t(y_t)\,l_{t-1}(y_{t-1} \mid y_t)}{\eta_{t-1}(y_{t-1})\,k_t(y_t \mid y_{t-1})}. \end{aligned} \tag{1.29}$$

Now, given that $\hat{\omega}_t(y_{1:t})$ has $\omega_t(y_t)$ as its marginal distribution, the final weighted sample $\left\{\left(Y_t^{(i)}, W_t^{(i)}\right)\right\}$ then is indeed by construction an approximation of the target density we started with. Given the weight update described above we can now see that the algorithm for the SMC sampler 3 would be the one given below:

Algorithm 3: Sequential Monte Carlo Sampler

if $t = 1$ **then**
  for *i=1:N* **do**
    $Y_1^{(i)} \sim \mu_1$ with $\mu_1(\cdot)$ an instrumental distribution
    Set weights to: $W_1^{(i)} \propto \frac{\mathrm{d}\eta 1}{\mathrm{d}\mu_1}\left(Y_1^{(i)}\right)$
  end
else
  $t \leftarrow t+1$
  for $i = 1 : N$ **do**
    $Y_t^{(i)} \sim K_t\left(Y_{t-1}^{(i)}, \cdot\right)$ and
    $W_t^{(i)} \propto W_{t-1}^{(i)} \frac{\eta_t\left(Y_t^{(2)}\right) L_{t-1}\left(Y_t^{(2)}, Y_{t-1}^{(i)}\right)}{\eta_{t-1}\left(Y_{t-1}^{(i)}\right) K_t\left(Y_{t-1}^{(i)}, Y_t^{(i)}\right)}$
  end
  Resampling can be performed at this step by sampling ancestral indices $a \sim$ Categorical distribution $O(W_t^1, ., W_t^N)$ Additonal rejuvenation can be conducted at this stage by allowing the particles to move by a Markov kernel of invariant distribution ω_t.
end

It is worth pointing that since the unknown normalising constants are proportional to the weights we update the weights with η_t instead of ω_t as indicated in the algorithm with the proportional sign. Furthermore, given that we get an estimate of the normalising constant from SMC type algorithms as in (1.24) we can see that we will also be able to estimate in this case the ratio of them, i.e. η_t and η_0 respectively as follows:

$$\widehat{\frac{\eta_t}{\eta_0}}^N = \prod_{k=1}^{t} \sum_{i}^{N} w_k(y_{1:k}^{(i)}) \tag{1.30}$$

1.3.8 The pseudo-marginal approach to MCMC

It is often the case that in real world applications the density of interest π is often intractable (in more than one way). Work done for example in Beaumont [2003], Andrieu and Roberts [2009] inspired by issues in population genetics has tried to address those issues, with the main assumption that an unbiased estimate of a point evaluation of the target is in fact available. Explicitly this means that for $\forall x \in X$ we have available the estimates $\hat{\pi}(x)$ (which are non-negative) with the unbiasedness condition $\mathbb{E}[\hat{\pi}(x)] = \pi(x)$ and with the expectation over all the random variables used (implicitly) to generate the computed estimate $\hat{\pi}(x) = \hat{\pi}(x; \psi)$.

We could of course independently sample $N \in \mathbb{N}^*$ estimates $\left\{\hat{\pi}(x)^{(i)}\right\}_{i=1}^{N}$ for every proposal x - taking advantage of the Law of Large Numbers- and substitute the approximated quantity into the acceptance ratio of the marginal algorithm by using the estimate $N^{-1}\Sigma_i \hat{\pi}(x)^{(i)} \approx \pi(x)$. Subsequently we could, by increasing N, effectively (and mathematically) decrease the discrepancy between this estimate and the true value to an arbitrary precision. Therefore by increasing our number of estimates N we could imagine that we are in some sense running an "averaged" algorithm that intuitively approaches the exact one. In fact as shown in Andrieu and Roberts [2009] this intuition is not only correct but we are in fact despite running a "noisy" sectional (with respect to the acceptance ratio) version of some "exact" algorithm we are marginally targeting the true posterior (hence the term pseudo-marginal). To see this consider the case $N = 1$ without loss of generality.

The analysis of this type of algorithmic framework is given in Andrieu and Roberts [2009], Andrieu and Vihola [2015]. We will follow their notation for the time being. Writing the unbiased estimates as their true value multiplied by some noise, i.e. $\hat{\pi}(x) = W_x \pi(x)$, with $W_x \sim Q_x(\cdot) \geq 0$ and $\mathbb{E}[W_x] = 1, \forall x \in X$.

In order to be precise take the collection of probability measures $\{Q_x\}_{x\in X}$ be defined on some measurable space $(\mathbb{R}_+, B(\mathbb{R}_+))$. Now consider the extended distribution on a product space such as $(\mathrm{X}\times\mathbb{R}_+, \mathcal{B}(\mathrm{X})\times\mathcal{B}(\mathbb{R}_+))$, given by $\tilde{\pi}(\mathrm{d}x,\ \mathrm{d}\omega) := \pi(\mathrm{d}x)\pi_x(\mathrm{d}\omega)$ where $\pi_x(\mathrm{d}\omega) := Q_x(\mathrm{d}\omega)\omega$. Clearly we get $\tilde{\pi}$ having as marginal distribution π because $\int \tilde{\pi}(x,\ \mathrm{d}\omega) = \pi(x)$.

Consider now the regular MH algorithm having as target $\tilde{\pi}$ with proposal $\tilde{q}(x,w;\mathrm{d}y,\ \mathrm{d}u) := q(x,\ \mathrm{d}y)Q_y(\ \mathrm{d}u)$. The acceptance ratio ends up being:

$$\begin{aligned}\tilde{\alpha}(x,\omega;y,u) &= \min\left\{1, \frac{\tilde{\pi}(\mathrm{d}y,\ \mathrm{d}u)\tilde{q}(y,u;\mathrm{d}x,\ \mathrm{d}\omega)}{\tilde{\pi}(\mathrm{d}x,\ \mathrm{d}\omega)\tilde{q}(x,\omega;\mathrm{d}y,\ \mathrm{d}u)}\right\}\\ &= \min\left\{1, \frac{\pi(\mathrm{d}y)Q_y(\ \mathrm{d}u)uq(y,\ \mathrm{d}x)Q_x(\ \mathrm{d}\omega)}{\pi(\mathrm{d}x)Q_x(\ \mathrm{d}\omega)\omega q(x,\ \mathrm{d}y)Q_y(\ \mathrm{d}u)}\right\}\\ &= \min\left\{1, \frac{\pi(\mathrm{d}y)q(y,\ \mathrm{d}x)}{\pi(\mathrm{d}x)q(x,\ \mathrm{d}y)}\frac{u}{\omega}\right\}\\ &= \min\left\{1, r(x,y)\frac{u}{\omega}\right\}\end{aligned} \tag{1.31}$$

By simplifying the acceptance ratio we can see that we have the "original ratio" $r(x,y)$ times some factor which translates into a variant of the marginal algorithm characterised as "noisy" (since we have altered the target in the numerator and denominator). We call the resulting algorithm pseudo-marginal. The name refers rather to the idea and intuition behind the construction of this extended space but we as one can see by careful consideration of the quantities involved we are in fact making use of an "exact" algorithm, since we are after all still targeting π. As a last step consider now the Markov kernel used in the algorithm:

$$\tilde{P}(x,\omega;\mathrm{d}y,\ \mathrm{d}u) := \tilde{\alpha}(x,\omega;y,u)\tilde{q}(x,\omega;\mathrm{d}y,\ \mathrm{d}u) + \delta_{x,\omega}(\ \mathrm{d}y,\ \mathrm{d}u)\tilde{\rho}(x,\omega) \tag{1.32}$$

and where, analogously to the marginal algorithm, we reject with probability

$$\tilde{\rho}(x,\omega) := 1 - \iint \tilde{\alpha}(x,\omega;y,u)\tilde{q}(x,\omega;\mathrm{d}y,\ \mathrm{d}u) \tag{1.33}$$

Some complications arise if we wish to extend the argument to the average of a multiple unbiased estimators. Assume that we average N of those (unbiased) estimates at each step, thereby obtaining an estimate of our target $\forall x \in \mathrm{X}$, of $N^{-1}\sum_i W_i \pi(x)$ where $W_{1:N} := (W_1, \ldots, W_N) \sim Q_x^N$, with Q_x^N being some probability measure on $\mathbb{R}_+^N$..

In analogous fashion to the $N = 1$ scenario, we can think of the pseudo-marginal algorithm as utilising these averages as a MH algorithm targeting $\tilde{\pi}^N(\mathrm{d}x, \mathrm{d}w_{1:N}) := \pi(\mathrm{d}x)\pi_x^N(\mathrm{d}w_{1:N})$ where

$$\pi_x^N(\mathrm{d}\omega_{1:N}) := Q_x^N(\mathrm{d}\omega_1, \ldots, \mathrm{d}w_N)\, N^{-1}\Sigma_i \omega_i \tag{1.34}$$

which admits π as its marginal distribution. Taking (x, ω) to be the current state, for some $x \in X$ and $w \in \mathcal{R}^+$, consider the following algorithm.

Algorithm 4: General Pseudo-Marginal algorithm

begin
  Input: state at current time: (x, w)
  get a sample $Y \sim q(x, \cdot)$.
  get a sample $U \sim Q_Y^N(\cdot)$.
  With probability $\tilde{\alpha}(x, w; Y, U)$ as given in 1.31 :

$$\text{set } (x', w') \leftarrow (Y, U) \tag{1.35}$$

  otherwise set

$$\text{set } (x', w') \leftarrow (x, w) \tag{1.36}$$

  Output: new state (x', w')
end

It is worth mentioning that the acceptance rate of an "exact" type marginal (within the appropriate extended space) algorithm is always greater than that of the pseudo-marginal. The authors in Andrieu and Vihola [2015] do in fact demonstrate that the asymptotic variance of the pseudo-marginal algorithm is always greater than that of the exact marginal (see Thorem 7 in Andrieu and Vihola [2015]).

1.3.9 Particle MCMC

Given that we now have a framework through which if we could somehow obtain unbiased estimators of some kind of likelihood, the next obvious question is through what means, and under which algorithmic framework might we do so. One of the most important applications of pseudo-marginal framework one could argue is the work of Andrieu et al. [2010].

In that framework we would like to have some augmented space and corresponding measure, such that given our target density ω, we will be able to construct an extended density ω^N, that has our target as its marginal. The main idea here would be to use SMC type algorithms like those of the previous section. Assume we have a target density denoted $\omega\left(\theta, y_{1:n}\right) = \omega(\theta)\omega_\theta\left(y_{1:n}\right)$. We can sample from $\omega_\theta\left(y_{1:n}\right)$ using some SMC method. In Andrieu et al. [2010] , the authors propose using an embedded SMC algorithm to generate the proposal inside an MCMC algorithm that has as invariant target density an (extended) version of $\omega(\theta, y_{1:n})$. Let us follow the same steps as the paper in developing the idea of the PMCMC methodology. Let us begin by constructing the extended target ω^N. Consider the joint density of $(\mathbf{y}, \mathbf{a})$ up to time n (of the SMC) is

$$\psi_\theta\left(y_{1:n}^{1:m}, a_{1:n-1}\right) = \left(\prod_{i=1}^{M} G_\theta\left(y_1^i\right)\right)\left(\prod_{j=2}^{n} r\left(a_{j-1} \mid w_{j-1}\right)\prod_{i=1}^{M} G_\theta\left(y_j^i \mid y_{1:j-1}^{a(i)}\right)\right) \tag{1.37}$$

without the terminal resampling. Here $r\left(a_{j-1} \mid w_{j-1}\right) = \prod_{i=1}^{M} w_{j-1}^{a(i)}$. Here a_i, and for most of the notation, denotes the ancenstral index the ith particle sampled according to the particle weights w^i from some distribution r (for example multinomial). Given θ, the density of $\left\{y_{1:n}^{1:M}, a_{1:n-1}^{1:M}\right\}$ conditioned on $\left(Y_{1:n}^k = y_{1:n}^k, A_{1:n-1}^k = a_{1:n-1}^k\right)$ will be

$$\frac{\psi_\theta\left(y_{1:n}^{1:m}, a_{1:n-1}\right)}{G_\theta\left(y_1^k\right)\left(\prod_{j=2}^{n} w_{j-1}^{a(k)} G_\theta\left(y_j^k \mid y_{1:j-1}^{a(k)}\right)\right)} \tag{1.38}$$

One can use 1.37 and the expression above to formulate an extended target as

$$\omega^N\left(k, \theta, y_{1:n}, a_{1:n-1}\right) = \frac{\omega\left(\theta, y_{1:n}^k\right) \psi_\theta\left(y_{1:n}, a_{1:n-1}\right)}{M^n G_\theta\left(y_1^k\right)\left(\prod_{j=2}^n w_{j-1}^{a(k)} G_\theta\left(y_j^k \mid y_{1:j-1}^{a(k)}\right)\right)} \tag{1.39}$$

with

$$\frac{\omega\left(\theta, y_{1:n}^k\right)}{M^n} = \omega^M\left(\theta, y_{1:n}^k, a_{1:n-1}^k \mid (\text{samples from 1.38})\right) \tag{1.40}$$

Where by k we denote the RV that represents the index of one sample $\left(y_{1:n}^k, a_{1:n-1}^k\right)$ being resampled from $\left\{y_{1:n}^{1:M}, a_{1:n-1}^{1:M}\right\}$. Therefore, $\left(y_{1:n}^k, a_{1:n-1}^k\right)$ and θ can be accordingly sampled from the marginal $\omega\left(\theta, y_{1:n}\right)$ if one indeed has $\{k, \theta, y_{1:n}, a_{1:n-1}\}$ from 1.39. Finally, sampling $\{k, \theta, y_{1:n}, a_{1:n-1}\}$, we can target 1.39 with the (PMMH) algorithm below 5, that has the following proposal density:

$$G^N\left(k, \theta, y_{1:n}, a_{1:n-1}\right) = G(\theta \mid \theta^*) \psi_\theta\left(y_{1:n}, a_{1:n-1}\right) w_n^k \tag{1.41}$$

where $\theta \sim G(\cdot \mid \theta^*)$ proposes a new value in the parameter space $\theta^* \in \Theta$ conditional on the current (accepted) one θ and w_n^k the probability of resampling the path $\left(y_{1:n}^k, a_{1:n-1}^k\right)$. Also notice that if $\omega\left(\theta, y_{1:n}\right) = \omega\left(\theta, y_{1:n} \mid x_{1:n}\right)$ and $\gamma\left(\theta, y_{1:n}\right) = \omega\left(\theta, x_{1:n}, y_{1:n}\right)$, then we have $Z_{\theta,1:n} = p_\theta\left(y_{1:n}\right)$ as the normalising constant of $\omega_\theta\left(y_{1:n}\right) = \omega_\theta\left(y_{1:n} \mid x_{1:n}\right)$. In such a situation, the acceptance probability of 5 suggests that the target of Particle Marginal Metropolis-Hastings is $\omega\left(\theta \mid x_{1:n}\right) \propto \omega(\theta) p_\theta\left(x_{1:n}\right)$, that is of course in turn the marginal density of

$\omega(\theta, x_{1:n} \mid y_{1:n})$.

Algorithm 5: Particle Marginal MC algorithm

begin
  Input: $\theta^t \sim \pi$, sample the rest of the variables through 1.39
  :
  if $t = 1$ **then**
    $y^t_{1:n}, a^t_{1:n-1} \mid \cdots \sim \psi_{\theta^t}(\cdot)$ from running the SMC in algorithm 2 , without terminal resampling step (at iteration t_{last}).
    pick $k^t \propto W^{t,k^t}_n$.
    Calculate the estimate, $\hat{Z}^t_{\theta^t,1:n}$, by 1.24
  else
    $t = t + 1$, Set Sample $\theta^* \sim q\left(\cdot \mid \theta^{t-1}\right)$. similarly to step 1 :
    sample $y^*_{1:n}, a^*_{1:n-1} \mid \cdots \sim \psi_{\theta^*}(\cdot)$ by using algorithm 2 with a terminal resampling .
    Choose $k^* \propto W^{*,k^*}_n$. Finally, calculate the marginal likelihood estimate, $\hat{Z}^*_{\theta^*,1:n}$, by 1.24
    With acceptance probability

$$1 \wedge \frac{\omega^N\left(k^*,\theta^*,y^*_{1:n},a^*_{1:n-1}\right)}{\omega^N\left(k^{t-1},\theta^{t-1},y^{t-1}_{1:n},a^{t-1}_{1:n-1}\right)} \cdot \frac{G^N\left(k^{t-1},\theta^{t-1},y^{t-1}_{1:n},a^{t-1}_{1:n-1}\right)}{G^N\left(k^*,\theta^*,y^*_{1:n},a^*_{1:n-1}\right)} = 1 \wedge \frac{\omega(\theta^*)}{\omega(\theta^{t-1})} \frac{G\left(\theta^{t-1}|\theta^*\right)}{G(\theta^*|\theta^{t-1})} \frac{\hat{Z}^*_{\theta^*,1:n}}{\hat{Z}^{t-1}_{\theta^{t-1},1:n}} \tag{1.42}$$

    set $k^l = k^*, \theta^l = \theta^*, y^l_{1:n} = y^*_{1:n}$, and $a^l_{1:n-1} = a^*_{1:n-1}$. Otherwise, set $k^l = k^{l-1}, = \theta^{l-1}, y^l_{1:n} = y^{l-1}_{1:n}$, and $a^l_{1:n-1} = a^{l-1}_{1:n-1}$ go to step 1
  end
end

1.3.10 SMC^2 algorithm

The SMC^2 algorithm Chopin et al. [2013] is an application of the idea behind PMCMC for inference of Hidden Markov Models with the aim to sample from $\pi(\theta, x_{1:n} \mid y_{1:n})$. More generally it is a noisy version (in the pseudo marginal sense) of a general SMC algorithm; the IBIS algorithm of Chopin [2002]. In the IBIS algorithm we have an idealised weight updating since it is often the case that for the model and problem we are interested in the ratio is not computable and are instead replaced by an auxiliary SMC sampler that provides an unbiased estimate of the desired measure. In order to justify and prove the consistency of the algorithm's estimates (for particle number $N_\theta \to N$ for some large N), one could think of it as a reformulation of the SMC algorithm

on an extended space $\mathcal{X} \times \mathbf{\Theta}$ as the density in the beginning implies. We could also think of the algorithm as a particle filter type such as the ones described in Vergé et al. [2015].

Algorithm 6: IBIS algorithm

for *t=1:T* **do**
  Sample $\theta_i, i \in 1 : N_\theta$
  compute the weights

$$\hat{\omega}_t\left(\theta^m\right) = p\left(y_t \mid y_{1:t-1}, \theta^m\right), \quad L_t = \frac{1}{\sum_{m=1}^{N_\theta} \omega^m} \times \sum_{m=1}^{N_\theta} \omega^m \hat{\omega}_t\left(\theta^m\right) \tag{1.43}$$

  with $p\left(y_1 \mid y_{1:0}, \theta\right) = p\left(y_1 \mid \theta\right)$ when $t = 1$.
  Update the weights,

$$\omega^m \leftarrow \omega^m \hat{\omega}_t\left(\theta^m\right) \tag{1.44}$$

  given some degeneracy condition (in the ESS [12]sense) , sample $\tilde{\theta}^m$ independently from the mixture

$$\frac{1}{\sum_{m=1}^{N_\theta} \omega^m} \sum_{m=1}^{N_\theta} \omega^m K_t\left(\theta^m, \cdot\right) \tag{1.45}$$

  replace the current weighted particle system, with the new unweighted particles:

$$\left(\theta^m, \omega^m\right) \leftarrow \left(\tilde{\theta}^m, 1\right) \tag{1.46}$$

  normalise weights and update
end

The author Chopin [2004] proves that one can estimate

$$\mathbb{E}\left[\psi(\theta) \mid y_{1:t}\right] = \int \psi(\theta) p\left(\theta \mid y_{1:t}\right) d\theta \tag{1.47}$$

consistently and in the asymptotic regime $N_\theta \to \infty$ with

$$\frac{\sum_{i=1}^{N_\theta} \omega^i \psi\left(\theta^m\right)}{\sum_{i=1}^{N_\theta} \omega^i} \tag{1.48}$$

for all integrable functions ψ defined appropriately. The Markov kernel K denotes an MCMC procedure with which we maintain the invariance of $p(\theta|y_{1:t})$

[12]Abbreviation for Effective Sample Size defined as $\text{ESS} = \frac{\left(\sum_{m=1}^{N_\theta} \omega^m\right)^2}{\sum_{m=1}^{N_\theta} (\omega^m)^2}$. The degeneracy condition is usually something of the form $ESS < \alpha N_\theta$ with $a \in [0, 1]$

They also show that the L calculated in 1.44 is a consistent estimator of $p(y_t|y_{1:t-1})$, as well as asymptotically consistent. A crucial point of the algorithm nonetheless is the realisation that the likelihood steps $p(y_t|y_{1:t-1,\theta})$ used throughout (in the SMC but also in the MCMC step) are intractable (in general state space model, but also as we will see in chapter 5 in the context of ABC). The authors then in the seminal paper Chopin et al. [2013] propose a way to estimate that with a SMC type algorithm. We will revisit SMC^2 in the fourth chapter of this thesis as it will form the basis for our novel algorithm.

1.3.11 Approximate Bayesian Computation

The first idea of an ABC-like algorithm was demonstrated in Rubin [1987]; a "conceptual experiment" of sort. One could sample different values of θ from some prior distribution $\pi(\theta)$ and then under some model $p(\cdot, \theta)$ a new set of data $\mathbf{y}$ [13] would be simulated. If the observed data points (in a general sense) were equal to the simulated ones then Rubin argued that the set of the drawn parameters is a sample from the true posterior $\pi(\theta|\mathbf{y})$. The author went to point out that this would imply that for continuous high dimensional data one would need to have an infinite number of steps to obtain just one sample from the desired posterior. Almost 15 years later and in Pritchard et al. [1999] the proposed algorithm was implemented, while addressing the infinite iteration issue and proposing a finite run time resolution. The idea required that the observed data were not exact matches of the simulated one, but rather "close" to them for some pre-defined metric and associated distance ϵ.

A brief overview of approximate Bayesian computation is at hand since it forms the underlying basis of this project and despite not being part of the completed novel work presented at the end, it is still used in the stochastic approximation Markov chain Monte Carlo for which we have some preliminary results.

[13] we will denote from here on and for the rest of this section with bold $\mathbf{y}$, the potentially multidimensional data. Similarly parameter(s) θ can also be assumed to be of an arbitrary dimension without any change in our elaboration or description of the operation of the algorithms unless explicitly stated

The ABC process is as follows:

1. generate parameter values θ from some distribution $g(\cdot)$ (usually the prior)
2. generate data y from the likelihood $p(y|\theta)$ conditional on those parameter values θ
3. accept the proposed θ if $||\mathbf{y} - \mathbf{y}_{obs}|| \leq \epsilon$, notice the equivalence between this and drawing a sample $(\theta, \mathbf{y})$ from some joint distribution proportional to $\mathbb{I}(||\mathbf{y} - \mathbf{y}_{obs}|| \leq \epsilon)p(\mathbf{y}|\theta)g(\theta)$ with $\mathbb{I}$ the indicator function, and $\mathbb{I}(S) = 1$ if S is true or $\mathbb{I}(S) = 0$ if S is false

If our sample is accepted with probability proportional to $\pi(\theta)/g(\theta)$ this means that our likelihood-free rejection algorithm is sampling from the joint distribution proportional to:

$$\mathbb{I}(||\mathbf{y} - \mathbf{y}_{obs}|| \leq \epsilon)p(\mathbf{y}|\theta)g(\theta)\frac{\pi(\theta)}{g(\theta)} = \mathbb{I}(||\mathbf{y} - \mathbf{y}_{obs}|| \leq \epsilon)p(\mathbf{y}|\theta)\pi(\theta) \tag{1.49}$$

and hence if we have $\epsilon = 0$ the marginal of θ of 1.49 equals the true posterior since

$$\lim_{\epsilon \to 0} \int \mathbb{I}(||\mathbf{y} - \mathbf{y}_{obs}|| \leq \epsilon)p(\mathbf{y}|\theta)\pi(\theta)d\mathbf{y} = \int \delta_{y_{obs}}(\mathbf{y})p(\mathbf{y}|\theta)\pi(\theta)d\mathbf{y} = p(\mathbf{y}_{obs}|\theta)\pi(\theta) \tag{1.50}$$

For $\epsilon \to 0$, the rejection algorithm produces samples, $(\theta, \mathbf{y})$ and by marginalising over y, one has the distribution of the target posterior, $\pi(\theta|\mathbf{y}_{obs})$. (The marginal of this auxiliary dataset $\mathbf{y}$ is essentially a point mass at $\{\mathbf{y} = \mathbf{y}_{obs}\}$)

The above approach, is termed rejection-ABC due to the fact that if the simulated observations are not "close" enough to the real data we reject the assumption that they came from a likelihood with parameter θ close to the true parameter. It becomes evident that an algorithm incorporating such method can be easily implemented by any practitioner of any field (as it has been) with some success. Of course the reality is that if one is to have any hope in

simulating data that are close enough to the observation one would need to simulate for millions and billions of parameters values and most importantly compare every single observation to every simulated one (in the output vector of arbitrary dimensionality and size) and taking a simple unweighted) euclidean distance between them will somehow output a good comparison down to a single number, ϵ. This seems like a rather poor approximation to the true posterior target. First, we always reject observation that are further than some arbitrary chosen ϵ, and in practice this is usually set up after a very large number of simulations have been performed and we decide we are going to keep a certain (small) percentage of them. One improvement to this is the fact that we can set up our process with a different kernel (rather than the uniform) that will probabilistically accept/reject simulated values depending on how far/close the simulation data are to observations. Secondly, although not an "improvement" in a theoretical sense but rather a very sensible computational consideration, is that we can resort to using not the full simulated output and observations in the comparison, but rather some summary statistics of those. The reason is that it is very hard to try and get very high dimensional objects "close" in some defined metric. The probability of that happening rapidly decreases as the dimensionality grows. We therefore have two improvements to rejection ABC: (i) instead of the uniform kernel use some other smooth kernel that will return a more continuous scaling from 0 to 1 for data that are "close" and "far" from the observation and which makes more sense than an arbitrary span of the uniform kernel which by construction does not differentiate between samples that are for example exactly equal to the observed ones and those that are the furthest away (i.e. the distance being exactly ϵ). (ii) arguably the first thing a practitioner encounters even before the choice of kernel is that of the comparison between observations themselves. As we mentioned an arbitrary distance metric between **all** observation points will hardly have any given parameter value be close enough to our true one since they **all** have to be close to each other. Suffice to say that this seems very much unlikely as the dimensionality and size of data grows, until a certain points for which it truly does not make much sense any more due to high dimensional spaces and their inherent characteristics not to mention the computational effort. In order for that to happen we then need to increase the scale parameter ϵ which then by construction makes the approximation to the posterior worse. We can therefore resort to summary statistics for our data and compare those. The

statistics reside in a much lower dimensional space than the full data obviously, which allows us to turn the scale parameter ϵ down in order to achieve a better approximate posterior.

ABC-MCMC

We will for simplicity of presentation assume it is reasonable to use the uniform kernel and the full data (although in any subsequent notation changing the pseudo-dataset $\mathbf{y}$ with some summary statistics $S(\mathbf{y})$ does not alter our exposition. We should note that for the remainder of the section we will switch our notation from y to x and from y_{obs} to y for ease of notation ABC uses the posterior distribution

$$\pi_\epsilon(\theta, x \mid y) \propto p(\theta) f(x \mid \theta) g_\epsilon(y \mid x, \theta),$$

where g_ϵ is the "ABC kernel", which we choose to be the usual uniform one around x, and tends to the Dirac $\delta_x(y)$ as $\epsilon \to 0$, such that, roughly speaking, $\pi_\epsilon(\theta, x \mid y) \to \pi(\theta \mid y)$ as $\epsilon \to 0$.

The reason this posterior distribution is used is that f is intractable in the sense that it cannot be evaluated pointwise at θ. The idea that is exploited is then that one can set up a Monte Carlo algorithm to sample from $\pi_\epsilon(\theta, x \mid y)$ by making use of $f(x \mid \theta)$ as a proposal for x. For example, in a step in an MCMC algorithm, where $\theta^* \sim q(\cdot \mid \theta)$ and $x^* \sim f(\cdot \mid \theta^*)$, we obtain an acceptance probability of

$$\begin{aligned}\alpha((\theta^*, x^*) \mid (\theta, x)) &= 1 \wedge \frac{p(\theta^*) f(x^* \mid \theta^*) g_\epsilon(y \mid x^*, \theta^*)}{p(\theta) f(x \mid \theta) g_\epsilon(y \mid x, \theta)} \frac{q(\theta \mid \theta^*) f(x \mid \theta)}{q(\theta^* \mid \theta) f(x^* \mid \theta^*)} \\ &= 1 \wedge \frac{p(\theta^*) g_\epsilon(y \mid x^*, \theta^*)}{p(\theta) g_\epsilon(y \mid x, \theta)} \frac{q(\theta \mid \theta^*)}{q(\theta^* \mid \theta)}.\end{aligned}$$

This means that we can implement the algorithm without ever evaluating f at θ. This is called ABC-MCMC. With this view, we can see that this algorithm might not explore the space very efficiently - because the proposal for x is not likely to be very good in a number of cases.

We can see that this is a pseudo-marginal type algorithm Andrieu and Roberts [2009] (section 1.5) : $f(x \mid \theta)\, g_\epsilon(y \mid x, \theta)$ is used as a crude (but unbiased) estimate of
$\int_x f(x \mid \theta)\, g_\epsilon(y \mid x, \theta)\, dx$, see for example Fearnhead and Prangle [2012]. Note that we could use a number of x samples for each θ, and obtain a more efficient MCMC (but at increased computational cost).

1.3.12 SMC-ABC

One of the application of SMC sampler is to likelihood free inference problems and specifically in Approximate Bayesian Computation. They were first introduced in Sisson et al. [2007] with the idea being that we define similarly to SMC sampler a sequence of targets, that here are a sequence of ABC posteriors with decreasing tolerance levels $\epsilon_t \leq \epsilon_{t-1}$, for $t = 1, \ldots, T$. The initial tolerance is usually chosen so that samples are drawn from the prior distribution. We choose the final tolerance level to be some desired ϵ. Here we will use a variation of the idea by Del Moral et al. [2012], with the sequence of unormalized targets being $\pi_{\epsilon_t}(y \mid x) f(x \mid \theta) p(\theta)$ for $t = 1, \ldots, T$, with initial distribution $f(x \mid \theta)$. To fix notation we denote the values of the particles in the SMC sampler to have a (p) superscript to distinguish them from random variables/vectors. Here we use a form of ABC-SMC that utilises the likelihood estimates of points drawn from the likelihood given θ: take the n^{th} point to be the m^{th} particle in θ-space, denoted by $x_{t,\theta}^{n,m}$. Initialise the algorithm by sampling each $\theta_0^{(m)} \sim p$ setting its unnormalised weight $\omega_0^{(m)} = 1$ and for each m, simulate $x_{0,\theta}^{n,m} \sim f_{\theta_0^{(m)}}(\cdot)$ for $1 \leq n \leq N_x$. These simulations in x-space conditional on θ will be used to estimate the ABC likelihood at iteration t with $l_t(y \mid \theta) = \int_x \pi_{\epsilon_t}(y \mid H(x, \theta))\phi(x)dx$ by using the estimator

$$\hat{l}(y \mid \theta) = \frac{1}{N_x} \sum_{n=1}^{N_x} \pi_{\epsilon_t}(y \mid H(x^n, \theta))\phi(x^n)). \tag{1.51}$$

Then the ABC-SMC sampler is as follows.

Algorithm 7: SMC-ABC algorithm

for $t = 1 : T$ **do**
  for *each* $1 \leq m \leq N_x$ **do**
    Compute the estimate of the likelihood $l_t\left(y \mid \theta^m\right)$ of the ABC likelihood when using ϵ_t

$$\hat{l}_t\left(y \mid \theta^m\right) = \frac{1}{N_x} \sum_{n=1}^{N_x} \pi_{\epsilon_t}\left(y \mid x_{t,\theta}^{n,m}\right).$$

    - Update the importance weights. If $t = 1$

$$\omega^m \leftarrow \omega^m \hat{l}_1\left(y \mid \theta^m\right)$$

    else if $t > 1$

$$\omega^m \leftarrow \omega^m \frac{\hat{l}_t\left(y \mid \theta^m\right)}{\hat{l}_{t-1}\left(y \mid \theta^m\right)}$$

    - If some degeneracy condition is fulfilled (e.g. the effective sample size), sample $\left(\bar{\theta}^m, \bar{x}_t^{1:N_x,m}\right)$ independently from the mixture distribution

$$\frac{\omega^m}{\sum_{j=1}^{N_\theta} \omega^j} K_t\left\{\left(\theta^m, x_t^{1:N_x,m}\right), \cdot\right\}$$

    where K_t is an ABC-MCMC kernel with respect to target t in the SMC.
  end
end

Here we have described a different version of SMC-ABC as given in Del Moral et al. [2012]. In our case MCMC moves are only utilised given certain degeneracy criteria. We also write the resampling step and an MCMC moves as a sampling procedure from a mixture distribution to compare directly to the SMC^2 of Chopin et al. [2013] which will be the base for the novel algorithm in chapter 4.

Chapter 2

Stochastic Approximation Monte Carlo ABC

2.1 Wang-Landau algorithm

An important problem of inference on general target measure ω on some measurable space $(\mathcal{Y}, \mathcal{K}, \mu)$ is that of multimodality. Putting aside the issue with distributions that can have particularly problematic shapes (e.g. banana shape, funnel etc) the issue of multiple modes presents unique challenges that are different in a number of ways than those presented by unusually shaped posterior distributions. Primarily that difficulty lies in the fact that even very well pre-conditioned and adaptive algorithms can have issues transversing the very lower probability regions between nodes, especially so in high dimensional spaces. A number of approaches have been proposed in order to overcome such issues, such as parallel tempering Geyer [1991], with an adaptive version in Miasojedow et al. [2013], the Wang-Landau algorithm Wang and Landau [2001a], Wang and Landau [2001b], the approach of simulated tempering by Marinari and Parisi [1992], and tempered transition of Neal [1996]. More recently there have been interesting developments such as the stochastic approximation Monte Carlo method of Liang [2009],and for an overview Liang [2014], based on earlier work Liang [2005], with extensions Liang [2007], as

well as more recent work based on the idea of mode jumping in Pompe et al. [2020]. In this chapter we will focus our attention and provide some encouraging experiments for a derivative work of the stochastic approximation Monte Carlo method applied to likelihood free inference problems as proposed first in Richards and Karagiannis [2020] and communicated privately by the second author to us. First, let us give a brief overview of the basis for stochastic approximation methods in general: the Wang-Landau algorithm.

Assume a partition of the state space of interest $\mathcal{Y}$ into disjoint sets $\mathcal{Y}_1, \mathcal{Y}_2, ...\mathcal{Y}_n$:

$$\mathcal{Y} = \bigcup_{i=1}^{n} \mathcal{Y}_i \tag{2.1}$$

and assume we can obtain independent and identically distributed samples (by some procedure, although here the formulation will be for MCMC algorithms), $Y_1, ...Y_T$. Then we have that for any $j \in [1, n]$

$$\frac{1}{T}\sum_{k=1}^{T} \mathbb{I}_{\mathcal{Y}_n}(Y_k) \xrightarrow[T\to\infty]{\mathbb{P}} \int_{\mathcal{Y}_j} \omega(x)dx =: \psi_j \tag{2.2}$$

with $\mathbb{I}_{\mathcal{Y}_n}$ denoting the indicator function being equal to 1 when $y \in \mathcal{Y}_n$ and 0 otherwise. If we can generate samples $Y_1, ...Y_T$ from some ergodic chain, for example one constructed by a Metropolis-Hastings algorithm we then have convergence in the sense defined on chapter 1, section 2.

In the Wang-Landau algorithm we aim to acquire samples so that any subsample

$$\{Y_k \text{ for } k \in [1, T] \text{ s.t. } Y_k \in \mathcal{Y}_j\} \tag{2.3}$$

for any $i \in [1, m]$ will have distribution based on the restriction of ω to $\mathcal{Y}_i$ and consequently for any $i \in [1, m]$

$$\frac{1}{T}\sum_{k=1}^{T} \mathbb{I}_{\mathcal{Y}_i}(Y_k) \xrightarrow[T\to\infty]{\mathrm{P}} \xi_i \tag{2.4}$$

with $\xi_= (\xi_1, \ldots, \xi_n)$ some choice of vector (which shall play the role of weights

or frequency of visit to the different "levels" that can be chosen to "slice" our target distribution, and could be any vector in $[0,1]^k$. such that $\sum_{i=1}^{n} \xi_i = 1$).

The WL algorithm and its applications and improvements/expansions over the years have as one of the primary goals the exploration of difficult multimodal distributions and overcoming the difficulty of moving over areas of low probability mass. They do so by altering the "weight" or frequencies these disjoint n-dimensional (for n-dimensional distribution) sets have in the sampling rate of the algorithm. One can think of these as a pseudo-geometric way of modifying the target distribution into some smoother version of it dynamically thereby making the inter-modal jumps much easier (with respect to the acceptance probability).

The algorithm proposed in Liang [2005],Liang [2009], allows to forcefully alter the sampling from the target ω by giving areas of low probability a higher proportion of visits (than otherwise) ξ_i from the n-dimensional set of the target. We can set these frequencies or weights according to how we view the difficulty of sampling from a given target (or how aggressive we want the weighting to be). This approach present some issues. First, the "splitting" of the posterior mass of ω over the subsets of $\mathcal{Y}$ results in masses, say ψ_i in 2.2 that are unknown and consequently there is no automatic way of increasing or decreasing the weight of each subregion in order to get the sampling frequency ξ_k that we want. One can think of the weights ξ as the frequency that we would like each subset of our state space to be sampled from. The original approach of the algorithm is to introduce some vector of ratios at each green iteration t of the algorithm defined as $z_t = (z_t(1), \ldots, z_t(n))$, that we update at every iteration and which are essentially estimates or approximations of $\psi_1/\xi_1, \ldots, \psi_n/\xi_n$. Taking into account 2.2 and 2.4, we have that for the posterior of interest ω and the vector of corrections $z_t = (z_t(1), \ldots, z_t(n))$, the "corrected" distribution of interest can be defined as :

$$\omega_h(y) \propto \omega(y) \times \sum_{i=1}^{n} \frac{\mathbb{I}_{\mathcal{Y}_i}(y)}{z(i)} \tag{2.5}$$

In the algorithm one usually defines some function $S : \mathcal{Y} \mapsto \{1, .., n\}$ that

maps values of the state space $y \in \mathcal{Y}$ and gives the index i of the subspace $\mathcal{Y}_i$ such that $y \in \mathcal{Y}_i$, allowing one to re-write 2.5 as $\omega_h(y) \propto \omega(x)/h(S(y))$. One should think about this process as follows: The functions purpose is to find where in the subspace of the sliced posterior our proposed new value of the MCMC chain lives, and accordingly modify the sampling frequency of that region for future visits, thereby biasing the sampling procedure, and in essence changing the posterior itself at every iteration. The algorithm switches between a sample generating step target ω_h under some kernel K_h (of the MCMC algorithm) and an update of the vector z using the sample generated previously. We can see that the algorithm behaves in an adaptive fashion since the samples generated previously $t-1$ are used in the update of the kernel at iteration t. This process can be imagined as one where an auxiliary chain (h_t) is created producing a collection of samples that are not drawn from our intended posterior. It is not immediately obvious what kind of correction is needed in order to obtain samples from the true posterior since these ones are clearly generated by an adaptive procedure which itself approximates the true target (in addition to the MC approximation.

The algorithm has seen considerable usage in the physics world Malakis et al. [2006], Cunha Netto et al. [2006]. Silva et al. [2006]. It is often used with a 'flat histogram criterion'. The convergence properties of this procedure are nevertheless not fully understood (some results indicate that the aforementioned criterion to be reached in finite time in Jacob and Ryder [2014]). There exist several variations of the base algorithm with various modification pandering to the needs of the application at hand and the needs of the practitioner. An important component of the algorithm is that of whether to have a deterministic or stochastic schedule in updating the z we have defined previously. Lets define some elementary conditions for this to make sense in the context of the algorithm.

Define $(\zeta_t)_{t\in\mathbb{N}}$ to be some sequence in $\mathbb{R}^+$ satisfying the following conditions:

$$\begin{cases} \sum_{t\geq 0} \zeta_t & = \infty \\ \sum_{t\geq 0} \zeta_t^2 & < \infty \end{cases} \tag{2.6}$$

Essentially implying convergence of a certain order. One would use a schedule such as $\zeta_t := t^{-\alpha}$ with $\alpha \in [0.5, 1]$ We see that such schedule deterministi-

cally decreases and its essential role is to modify how much the adaptation and weighting of each subset of our space changes over time (and essentially asymptotically being zero at the end for convergence to the true/intended posterior). We give the pseudo-code for the general Wang-Landau below in Algorithm 8. In this form, the schedule γ_t will iteratively decrease, and we thus call it "deterministic".

Algorithm 8: Wang-Landau algorithm with deterministic schedule

```
begin
    Initialize ∀i ∈ {1, ..., n} set h_0(i) ← 1/n
    Initialize Y_0 ∈ 𝒴
    for t = 1 : T do
        Sample Y_t trom K_{z_{t-1}}(Y_{t-1}, ·), MH kernel targeting ω_{h_{t-1}}.
        Update the penalties: log h_t(i) ← log h_{t-1}(i) + g(𝕀_{𝒴_i}(Y_t), ξ_i, ζ_t)
    end
end
```

In the last step of the algorithm we update h_{t-1} to h_t, its value depending on whether the subspace has been visited before, and therefore increasing the value, while if it has not been visited, decreasing it. An obvious first question would be what are the potential choices of the updating function g. We are only constrained by the fact that the function should be positive when $Y_t \in \mathcal{Y}_i$ and that sufficient conditions are met when it is close to 0 such that the sequence of ζ_t decreases, thus hopefully ensuring that the penalties do converge in the appropriate topology (which for all intents and purposes here it will be the same throughout). In the literature authors/users (for example see Liang [2005], and the review in Liang [2014]) seem to perform the step with either of the following two cases:

$$\log h_t(i) \leftarrow \log h_{t-1}(i) + z_t \left(\mathbb{I}_{\mathcal{Y}_i}(Y_t) - \xi_i\right) \tag{2.7}$$

or

$$\log h_t(i) \leftarrow \log h_{t-1}(i) + \log\left[1 + z_t \left(\mathbb{I}_{\mathcal{Y}_i}(Y_t) - \xi_i\right)\right] \tag{2.8}$$

We know that if z_t converges to 0 when t increases, and with the first being the first-order Taylor expansion of the second, we would expect the results to look similar. According to the authors in Jacob and Ryder [2014]

this doesn't seem to be universally the case. There are various convergence results for the case of the updating schedule and its form shown above. This deterministic schedule makes sure that h_t changes in a diminishing fashion as the number of iterations grows and therefore the defined kernels K_{h_t} do so accordingly. Studies of adaptive algorithms such as the ones in Andrieu and Atchadé [2007] Atchadé et al. [2011] Fort et al. [2014] where a condition known as diminishing adaptation holds include cases of this algorithm but with an important difference. It is the target itself that changes in every iteration (recall the form of the posterior in the previous page) and not -usually- the proposal distribution which is the case for these kinds of adaptive algorithms. The interested reader can also consult Andrieu and Moulines [2006]. Finally, let us briefly also mention an improvement of the algorithm where the schedule decreases at random times only (by using the flat histogram criterion) and is in fact the version widely used in the physics literature (without a particular theoretical underpinning of its validity nonetheless), yet since in this chapter we will not be using that version we only passingly mention it for completeness.

2.1.1 Metropolis-Hasting ABC posterior

Here let us reintroduce the ABC-MCMCM algorithm and recall the invariant target of the algorithm. It will be useful as a backdrop when we think about what the newly proposed algorithm is in trying to achieve: ABC uses the posterior distribution $\pi_\epsilon(\theta, x \mid y) \propto p(\theta) f(x \mid \theta) g_\epsilon(y \mid x, \theta)$ where g_ϵ is the "ABC kernel", which is usually symmetric around x, and tends to the Dirac $\delta_x(y)$ as $\epsilon \to 0$, such that, roughly speaking, $\pi_\epsilon(\theta, x \mid y) \to \pi(\theta \mid y)$ as $\epsilon \to 0$. The most widely used choice is to take g_θ to be a uniform, $\mathcal{U}(y \mid x - \epsilon, x + \epsilon)$ for all θ. However, a more sensible choice in many situations might be to use $\mathcal{N}(y \mid x, \sqrt{\epsilon})$, with $\sigma := \sqrt{\epsilon}$ which as we will see is in fact almost necessary in the context of this algorithm. The reason this posterior distribution is used is that f is intractable in the sense that it cannot be evaluated pointwise at θ. The idea being exploited here is that one can then set up a Monte Carlo algorithm to sample from $\pi_\epsilon(\theta, x \mid y)$ by making use of $f(x \mid \theta)$ as a proposal for x. For example, in a step in an MCMC algorithm, where $\theta^* \sim q(\cdot \mid \theta)$ and

$x^* \sim f(\cdot \mid \theta^*)$, we obtain an acceptance probability of

$$\begin{aligned}\alpha\left((\theta^*, x^*) \mid (\theta, x)\right) &= 1 \wedge \frac{p(\theta^*) f(x^* \mid \theta^*) g_\epsilon(y \mid x^*, \theta^*)}{p(\theta) f(x \mid \theta) g_\epsilon(y \mid x, \theta)} \frac{q(\theta \mid \theta^*) f(x \mid \theta)}{q(\theta^* \mid \theta) f(x^* \mid \theta^*)} \\ &= 1 \wedge \frac{p(\theta^*) g_\epsilon(y \mid x^*, \theta^*)}{p(\theta) g_\epsilon(y \mid x, \theta)} \frac{q(\theta \mid \theta^*)}{q(\theta^* \mid \theta)}\end{aligned} \tag{2.9}$$

Thus we implement the algorithm without ever evaluating f at θ. With this view, we can see that this algorithm might not explore the space very efficiently - because the proposal for x is unlikely to be particularly close to areas of high mass concentration. It is also worth remembering that in this algorithmic setup we must make the critical choice of ϵ from the very beginning thus immediately restricting what the acceptance rate can be, despite the use of different kernels which have only a probabilistic effect on this. Furthermore, it is perhaps worth noting that as we briefly discussed in the introduction this can be seen as a pseudo-marginal algorithm of the type in Andrieu and Roberts [2009] : $f(x \mid \theta) g_\epsilon(y \mid x, \theta)$ is used as a crude (but unbiased) estimate of $\int_x f(x \mid \theta) g_\epsilon(y \mid x, \theta) dx$.. We should mention here that one could use a number of x samples for each θ, and obtain a more efficient MCMC (but at increased computational cost). Consider now the target defined above as:

$$\pi_\epsilon(\theta, x \mid y) = \frac{g_\epsilon(y \mid x, \theta) f(x \mid \theta) \pi(\theta)}{\int_{\Theta \times \mathcal{Y}} g_\epsilon(y \mid x, \theta) f(\mathrm{d}x \mid \theta) \pi(\mathrm{d}\theta)} \tag{2.10}$$

The MH-ABC algorithm targeting $\pi_\epsilon(\mathrm{d}\theta, \mathrm{d}x \mid y)$ with proposal distribution $q(\cdot \mid \theta)$ has the form

Algorithm 9: Approximate Bayesian Computation Metropolis-Hastings algorithm

begin
  draw θ' from $q(\cdot \mid \theta)$
  draw x' from $f(x \mid \theta')$
  accept θ' with probability $a_{\mathrm{MHABC}} = \min(1, R_{\mathrm{MHABC}})$ with

$$R_{\mathrm{MHABC}} = \frac{\pi_\epsilon(\theta', x' \mid y) q(\theta \mid \theta') f(x \mid \theta)}{\pi_\epsilon(\theta, x \mid y) q(\theta' \mid \theta) f(x' \mid \theta')} = \frac{g_\epsilon(y \mid x', \theta') \pi(\theta') q(\theta \mid \theta')}{g_\epsilon(y \mid x, \theta) \pi(\theta) q(\theta' \mid \theta)} \tag{2.11}$$

end

2.1.2 Issues with ABC-MCMC and an idea for the augmentation of space by ϵ levels

In the introduction to the Wang-Landau algorithm we saw that one of the main benefits and motivation for its derivation was the idea that by splitting the target posterior mass over arbitrary levels of the space of interest, and setting penalties for the visitation of those levels one can overcome very low probability areas or "wells" where the chain can get stuck (i.e. tails) in order to visit complex multimodal target densities. A way to see why our proposal here might have substantial benefits in the context of ABC-MCMC is to imagine the Wang-Landau, and more specifically algorithms such as the SAMC of Liang [2005], Liang [2007], Liang [2010], as algorithms that alter the geometry of the target space by "smoothing" out modes, thus making the probability landscape significantly smoother, thereby allowing the chain to move much more freely around the parameter space. Here, instead the idea of Richards and Karagiannis [2020] is for the levels to be the different ϵ_i of some arbitrary target thus augmenting our space. Hence the chain can in fact "jump" to higher and lower levels of epsilon at will, allowing the serious limitation of ABC-MCMC where low values of epsilon cause it to get stuck for very long periods. Of course one can set a very small epsilon, thus causing the MCMC algorithm to have a vanishingly small acceptance probability and extremely low efficiency, or set up a larger than desired value, thus making the estimator of the ABC posterior worse. In the proposed approach one sets energy "levels" over ϵ (one could also jointly set (ϵ, θ), since multimodality of the posterior over the parameters of interest might be the case in addition to the issue being addressed here) within a certain epsilon range $\epsilon_1, .., \epsilon_k$ over k different levels. The user chooses the smallest and largest value as well as how many different partitions there can be in the algorithm (for both ϵ or jointly with the θ space). Wang-Landau is an algorithm that samples from a modified posterior distribution. Suppose that the marginal is $\omega(\theta \mid y)$. Then WL uses the target

$$\bar{\pi}(\theta \mid y) = \omega(\theta \mid y) \frac{1}{d} \sum_{i=1}^{d} \frac{\mathcal{I}_{\Theta_i}(\theta)}{\psi(i)}$$

where $\psi(i) = \int_{\Theta_i} \omega(\theta \mid y) \, d\theta$. Drawing with equal probability from each of the Θ_i regions, should assist us in exploring the space. However, we are not

drawing from the true posterior, and we need to perform a correction if we wish to obtain points from the posterior. Combining the ABC posterior and the WL one, we use the target

$$\bar{\pi}_\epsilon(\theta, x \mid y) \propto p(\theta) f(x \mid \theta) g_\epsilon(y \mid x, \theta) \frac{1}{d} \sum_{i=1}^{d} \frac{\mathcal{I}_{E_i}(\theta, x)}{\psi(i)} \tag{2.12}$$

where the E_i are regions defined by stratifying the $\mathcal{Y}$ -space by the distance from the observed data y, i.e.

$$E_i = \{(\theta, x) \mid \delta_{i-1} \leq d(y, x) < \delta_i\} \tag{2.13}$$

with $0 = \delta_0 < \delta_1 < \ldots < \delta_d = \infty$ and where

$$\begin{aligned} \psi(i) &= \int_{E_i} p(\theta) f(x \mid \theta) g_\epsilon(y \mid x, \theta) d\theta dx \\ &= \int_\theta p(\theta) \left[\int_{\delta_{i-1} \leq d(y,x) < \delta_i} f(x \mid \theta) g_\epsilon(y \mid x, \theta) dx \right] d\theta \end{aligned} \tag{2.14}$$

Therefore running the Wang-Landau algorithm our MCMC chain will target the density 2.12 (asymptotically), and we are in fact, at each iteration, altering the target (or more precisely targeting a different one at iteration t). The implication of this is that we aim to to spend an equal amount of time in each stratum (although it is the case here that since we are not considering the stratums to be a splicing of the parameter space, rather the epsilon, which itself defines a different ABC pseudo-posterior, we might want to have a variably, perhaps biased sampling rate for smaller epsilon levels since this will reward us with a better estimator [1]).

It is also worth pointing out that there is a clear trade-off here. We want just enough flexibility to avoid the sticky behaviour due to ϵ but not so much as to force the algorithm to spend time on large values of epsilon thus sampling

[1] The main distinction here from the original SAMC algorithm is that instead of partitioning the parameter space we partition the epsilon range into different levels. In the case of the parameter space we want to spent time that is proportional to the distribution mass of those parameters so as to not bias the sampling, whereas in the epsilon case we want to spend time mostly in lower ones (lower epsilon → better posterior approximation), and therefore we donât want proportional sampling in those partitions but rather biased towards smaller epsilons

(and contributing to the mixture) of "bad" ABC posterior approximations.

Let us consider the role and interpretation of $\psi(i)$ in our implementation of the Wang-Landau algorithm within the ABC context:

$$\begin{aligned}
\psi(i) &= \int_\theta p(\theta) \left[\int_{\delta_{i-1} \leq d(y,x) < \delta_i} f(x \mid \theta) g_\epsilon(y \mid x, \theta) dx \right] d\theta \\
&= \int_\theta p(\theta) \left[\int_{d(y,x) < \delta_i} f(x \mid \theta) g_\epsilon(y \mid x, \theta) dx - \int_{d(y,x) < \delta_{i-1}} f(x \mid \theta) g_\epsilon(y \mid x, \theta) dx \right] d\theta \\
&= \int_\theta p(\theta) \left[\int_{d(y,x) < \delta_i} f(x \mid \theta) g_\epsilon(y \mid x, \theta) dx \right] d\theta \\
&- \int_\theta p(\theta) \left[\int_{d(y,x) < \delta_{i-1}} f(x \mid \theta) g_\epsilon(y \mid x, \theta) dx \right] d\theta \\
&\triangleq \bar{\psi}(i) - \bar{\psi}(i-1)
\end{aligned} \tag{2.15}$$

with $\bar{\psi}$ defined by this equation. This implies that $\bar{\psi}(d) = \sum_{i=1}^{d} \psi(i)$ is equal to the ABC marginal likelihood. Furthermore, if we choose g_ϵ to be uniform, then the $\bar{\psi}(i), \{i = 1, .., d\}$ are related to the ABC marginal likelihood with the uniform kernel for different values of ϵ. This means that by running ABC WL, we also get an estimate of the marginal likelihood, although in the present work we will not be utilising this added benefit.

An important point is to notice that the adaptation introduces a bias to the target, and in effect we are not really sampling from the true target. We have points generated from our chain from the biased posterior and if we treat them as independent points form this posterior we can use importance sampling to get the points from the true ABC posterior. We could treat the MCMC points from this "biased" posterior as independent points from this posterior, and use importance sampling to get points from the true ABC posterior. The importance weight for a point from region with index i would be

$$\begin{aligned}
w &= \frac{p(\theta) f(x \mid \theta) g_\epsilon(y \mid x, \theta)}{p(\theta) f(x \mid \theta) g_\epsilon(y \mid x, \theta) \frac{1}{d} \sum_{i=1}^{d} \frac{\mathcal{I}_{E_i}(\theta, x)}{\psi(i)}} \\
&= d\psi(i)
\end{aligned} \tag{2.16}$$

The weight would be $w = \psi(i)/\omega(i)$, if we give stratum i desired weight $\omega(i)$, rather than simply $1/d$. Suppose that we used a different tolerance ϵ_{WL} in

the WL, compared to the tolerance ϵ for which we wanted the posterior. We would have the weight as being

$$\begin{aligned} w &= \frac{p(\theta)f(x \mid \theta)g_{\epsilon}(y \mid x, \theta)}{p(\theta)f(x \mid \theta)g_{\epsilon_{WL}}(y \mid x, \theta)\sum_{i=1}^{d}\frac{\omega(i)\mathcal{I}_{E_i}(\theta,x)}{\psi(i)}} \\ &\propto \frac{g_{\epsilon}(y \mid x, \theta)}{g_{\epsilon_{WL}}(y \mid x, \theta)}\frac{\psi(i)}{\omega(i)} \end{aligned} \tag{2.17}$$

For the uniform kernel, this would boil down to $w \propto \mathcal{I}(d(x,y) \leq \epsilon)\psi(i)/\omega(i)$

The deterministic schedule is defined as :

$$\gamma_t = \frac{1}{t^b} \tag{2.18}$$

with b some user defined value and t the time index of the Markov Chain iteration. A value of t_0 is defined initially such that

$$\gamma_t = \left\{ \begin{array}{ll} 1, & \text{if } t_0 \geq \frac{1}{t^b} \\ \frac{1}{t^b}, & \text{otherwise} \end{array} \right\} \tag{2.19}$$

2.2 The (SAMC-ABC) algorithm

We can now define the SAMC-ABC algorithm with the setup being as follows:
Let $\mathcal{E} = \{E_j; j = 1, \ldots, m+1\}$ be a partition of the sampling space $\mathcal{Y}$ with subregions
$E_1 = ((\theta, x) \in \Theta \times \mathcal{Y} : \epsilon_0 < U(\Theta \times \mathcal{Y}, x) \leq \epsilon_1), \ldots,$
$E_j = ((\Theta \times \mathcal{Y}, x) \in \Theta \times \mathcal{Y} : \epsilon_{j-1} < U(\Theta \times \mathcal{Y}, x) \leq \epsilon_j), \ldots,$
$E_{m+1} = ((\Theta \times \mathcal{Y}, x) \in \Theta \times \mathcal{Y} : \epsilon_m < U(\Theta \times \mathcal{Y}, x) < \epsilon_{m+1})$, with grid $\{\epsilon_j; \epsilon_j \in \mathbb{R}, j = 1 : m\}$, for $m > 0$, and $\epsilon_0 = 0$ and $\epsilon_{m+1} = +\infty$. Here, $j(\theta, x)$ indicates the label of the sub-region of the partition $\mathcal{E}$ that corresponds to the value (θ, x). Let $\omega := (\omega_j; j = 1, \ldots, m+1)$, such that $\phi_j = \Pr((\theta, x) \in E_j), \omega_j > 0$ and $\sum_{j=1}^{m+1} \phi_j = 1$, denote the vector of desired sampling frequencies of the m subregions $\{E_j\}$. Define the SAMC-ABC posterior distribution as previously

$\omega_{\epsilon,\psi}(\mathrm{d}\cdot \mid y) := \omega\left(\mathrm{d}\cdot \mid y, g_\epsilon, \mathcal{E}, \omega\right)$ with density

$$\omega_{\epsilon,\psi}(\theta, x \mid y) = \sum_{j=1}^{m+1} \phi_j \frac{1}{\psi_j} \omega_\epsilon(\theta, x \mid y) \delta_{E_j}(\theta, x) \tag{2.20}$$

at where $\psi := (\psi_j; j = 1:m), \psi_j = \int_{E_j} \omega_\epsilon(\mathrm{d}\theta, \mathrm{d}x \mid y) < \infty$ are the bias weights. with, ψ unknown. Let $z \in \mathbb{R}^{m+1}$ be vectors such that

$$\omega_{\epsilon,z}(\theta, x \mid y) \propto \sum_{j=1}^{m+1} \phi_j \omega_\epsilon(\theta, x \mid y) \exp\left(-h_j\right) \delta_{E_j}(\theta, x) \tag{2.21}$$

Note that $\exp\left(h_j\right) \propto \psi_j/\phi_j$, for $j = 1, \ldots, m+1$. Also note that $\psi = \phi_j \exp\left(h_j\right)$ iff $h \leftarrow h - C$ and $C = \log\left(\sum_{j=1}^{m+1} \phi_j \exp\left(z_j\right)\right)$, for $j = 1, \ldots, m+1$. Note that the reason for those choices are the fact that in Liang et al. [2007] prove that h in fact converges to the $C - log(\psi/\omega) - log(\phi + \mu)$ with μ equal to $\sum_{j \in \{i:E_j = \emptyset\}} \phi_j / \left(m - m_0\right)$, m_0 is the number of empty subregions. The SAMC-ABC algorithm targeting $\omega_{\epsilon,\psi}(\theta, x \mid y)$ with proposal distribution $q(\cdot \mid \theta)$ has

the following form:

Algorithm 10: The Stochastic Approximation Monte Carlo ABC algorithm SAMC-ABC

begin

Sampling step:

Draw

$$\theta_{t+1} \sim P_{\mathrm{MH}}\left(\cdot \mid \theta_t; \mathcal{E}, \phi, \psi\right) \tag{2.22}$$

as

(a) draw θ' from $q(\mathrm{d}\cdot \mid \theta)$

(b) draw x' from $f(x \mid \theta)$

(c) accept θ' with probability $a_{\mathrm{SAMCABC}} = \min\left(1, R_{\mathrm{SAMCABC}}\right)$ with

$$\begin{aligned} R_{\mathrm{SAMCABC}} &= \frac{\omega_{\epsilon,\psi}\left(\theta', x' \mid y\right) q\left(\theta \mid \theta'\right) f\left(x \mid \theta'\right)}{\omega_{\epsilon,\psi}(\theta, x \mid y) q\left(\theta' \mid \theta\right) f\left(x' \mid \theta'\right)} \\ &= \frac{g_\epsilon\left(y \mid x', \theta'\right) \omega\left(\theta'\right) \exp\left(h_{j(\theta', x')}\right) q\left(\theta \mid \theta'\right)}{g_\epsilon(y \mid x, \theta) \omega(\theta) \exp\left(h_{j(\theta, x)}\right) q\left(\theta' \mid \theta\right)} \end{aligned} \tag{2.23}$$

Update step

begin

Compute

$$h_{t+1} = h_t + \gamma_{t+1}\left(p_{t+1} - \phi\right)$$

$$\text{where } p_{t+1} := p_{t+1}\left(\theta_{t+1}, x_{t+1}\right), \text{ and } \left[p_{t+1}\right]_j = \left\{ \begin{array}{ll} 1 & \text{if } \left(\theta_{t+1}, x_{t+1}\right) \in E_j \\ 0 & \text{, if } \left(\theta_{t+1}, x_{t+1}\right) \notin E_j \end{array} \right\} \tag{2.24}$$

end

end

2.2.1 Lotka-Voltera

The Lotka-Voltera model is defined as a Markov jump process where the number of individuals in a population of animals (prey and predator) is modelled. It is often used in the ABC literature due to the fact that the likelihood is unavailable point wise, yet we can have exact simulations from the model by making use of the Gillespie algorithm Gillespie [1977]. The stochastic version of the model describes the time evolution of two species, say Y_1 (prey) and Y_2 (predator) through the following reaction equations:

$$\begin{aligned} Y_1 &\xrightarrow{r_1} 2Y_1 \\ Y_1 + Y_2 &\xrightarrow{r_2} 2Y_2 \\ Y_2 &\xrightarrow{r_3} \emptyset \end{aligned} \tag{2.25}$$

with r_1, r_3 first order and r_2 second order reactions. Here r in general represents a rate constant with rdt being the probability that the population species Y_1 doubles (or a predator dies with rate r_2 or a prey dies with rate r_3 during the time interval $[t, t + dt)$. The observations out of the model are some $y = (y_1, \ldots, y_n)$ composed of vectors $y_i = (y_{i,1}, y_{i,2}) \in \mathbb{N}^2$, corresponding to population levels at integer times. We usually let $\theta = (\log r_1), \log r_2, \log r_3$. This model is Markovian since $p(y_i \mid y_{1:i-1}, \theta) = p(y_i \mid y_{i-1}, \theta)$. The model is therefore a good test case for ABC as indicated in Toni et al. [2009]. The authors in Boys et al. [2008] demonstrate that an MCMC algorithm is feasible for this model, although as noted in Holestein [2009](chapter 4) the proposed schemes can be inefficient.

2.2.2 Numerical experiments

The target density of interest here is that of $\pi_\epsilon(\theta \mid y)$, with $\theta = (\theta_1, \theta_2, \theta_3)$ the 3 parameters of the Lotka-Voltera model specified in the previous section. We set up the priors for these parameters to be $[\log\theta_1, \log\theta_2, \log\theta_2] \sim U([-6, 2]^3)$. These range of priors seems to have a good coverage of the posterior of interest Golightly and Wilkinson [2005],

Golightly et al. [2015], while our summary statistics are based on the observations of the model output at discrete times and by taking the autocorrelations of the $\overline{X}_k = X_{5k}$ with lag 2 and the mean and variances of $\overline{Y}_k = Y_{5k}$ and $\overline{X}_k = X_{5k}$ with initial values of the populations as $X_1 = 50, X_2 = 100$. We run the R packages $smfsb$ from Wilkinson [2011] which is a collection of tools for building/simulating stochastic kinetic models, and the associated dataset (LV-data, which consists of observations at integer times), . In order to establish a baseline we run initially a long chain of particle MCMC as provided in the package in order to estimate the marginal log-likelihood and calculate through MCMC the posterior of the parameters. Furthermore, we run again, a long chain of ABC-MCMC in order to have a baseline for which we can compare the proposed algorithm against.

The values that generated the output that serves as the observations for the comparison through the ABC kernel are $\boldsymbol{\theta} = (1.0, 0.005, 0.6)$, while the empirical mean values out of the PMCMC run are $\boldsymbol{\theta} = (0.958[0.032], 0.00486[0.000014],$ $0.613[0.018)$ with the standard deviations in brackets. For the distance metric used within ABC we use the Euclidean distance between those summary statistics given above, normalised by the standard deviations of each obtained by a very large number $\sim 6 * 10^8$of samples from the prior (and associated distances). For the ABC-MCMC algorithm we run both the standard Uniform kernel $\mathbb{I}(||S(y_{obs}) - S(y_{generated})|| < \epsilon)$ and the Gaussian $\mathcal{N}(||S(y_{obs}) - S(y_{generated})|||0, \sqrt(\epsilon))$, although in this case it made little actual difference so we used the uniform kernel throughout instead. The reference level of epsilon under which acceptance occurs for all the runs was $\epsilon = 0.2$. The subset of the SAMC-ABC algorithm were split into various different sizes with a minimum value of $\epsilon_1 = 0$ and a maximum value of $\epsilon_{N_{grid}} = 1.0$. We chose this arrangement in order to have a reasonable comparison between ABC-MCMC and SAMC-ABC given that the motivation behind this ABC variation of the SAMC algorithm is to allow a more efficient exploration of the target space, therefore a somewhat wide enough range of values is chosen in order for the algorithm to not get stuck in trying to propose values that generate a very small epsilons, yet not unreasonably wide such that the sample is too biased towards values that are far from the true ones. The reason for this as we will see is that given the construction of the algorithm as seen in the previous section we are in fact at every time sampling from a different target density and

in fact the final sample can be though of as a sample from a mixture of ABC posterior approximations with different tolerances (ϵ). All the chains were run for 20000 iterations, and each replicated 40 times in order to get an acceptable variance of the estimated empirical means. Initially we see that for an initial $t_0 = 10$, in figure 2.1 the effects of the new algorithm are immediate. We get a good decrease in variance for all three parameters. Interestingly an increase in the grid size, which essentially gives a finer or coarser stratification of the epsilon levels seems to always improve the estimates for parameters θ_1 and θ_3 with the average moving closer to the true value. The exception here being the θ_2 which interestingly seems to become worse as one increases the grid size. The same behaviour is exhibited when one increases the SAMC schedule $t_0 = 50$ as seen in 2.2, and for $t_0 = 100$ although slightly diminished.

The behaviour can be seen at large values of $t_0 = 100$ as well where the effect is slightly more prominent and it seems that the grid size has a diminished effect compared to values of $t_0 = 10$. Given equation 2.18, we can see that t_0 is controlling *when* the algorithm will start to diminish the adaptation rate and in essence larger values (or very large values) depend on the value of the parameter b and can therefore allow the algorithm to always have a stochastic schedule coefficient γ_t that is equal to 1 and thus modify the probability that a given value will be accepted or rejected fully. The reason for the decreasing schedule as we explained in the introductory section is that of convergence of the algorithm as show for general MCMC algorithms in Roberts and Rosenthal [2007]. In our case, the effect can be explained by the fact that the correction factors in the SAMC MH ratio keep altering the acceptance rate of regions that are not visited (by increasing the chance they acceptances are made there) as well as those that are visited (by decreasing the acceptance rates), therefore allowing the MCMC to accepts parameter values from larger epsilon levels, thereby corresponding to worse parameters and therefore resulting in worse estimates. Of course that is the case for this very particular model. Different models and scenarios will probably require different approaches in how much the adaptation needs to keep going. It might be the case that the chain exhibits particularly sticky behaviour and a longer period of adaptation with factor of $\gamma_t = 1$ needs to be applied in order to escape problematic areas and overcome regions of very low probability. One should also notice that this is inextricably linked with the number of partitions. The more partitions, the finer the grid

and the "slicing" over possible energy levels. That granularity allows more fine tuning of partition updating (of the h)and hence allowable parameter values by (in this case) permiting only slightly larger values of epsilon to be acceptable (and to allow parameters that generated sets within those epsilon levels).

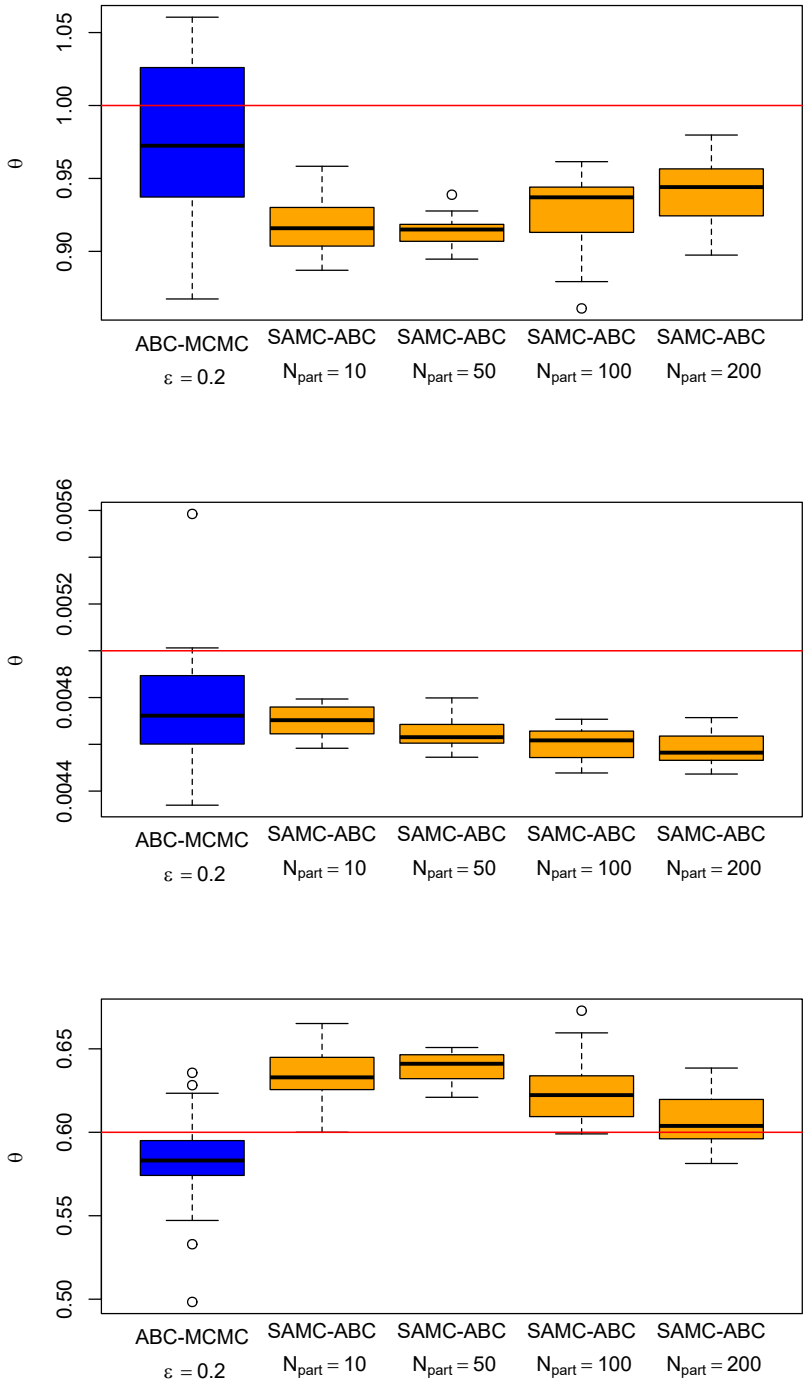

Figure 2.1: Comparison of empirical means of ABC-MCMC and SAMC-ABC for various grid sizes. The deterministic schedule here has a value of $t_0 = 10$ and $b = 0.7$. The red line indicates the true values.Top figure is parameter θ_1, middle is θ_2, and bottom is θ_3. The chains were run for 20000 iterations and for 40 replications each.

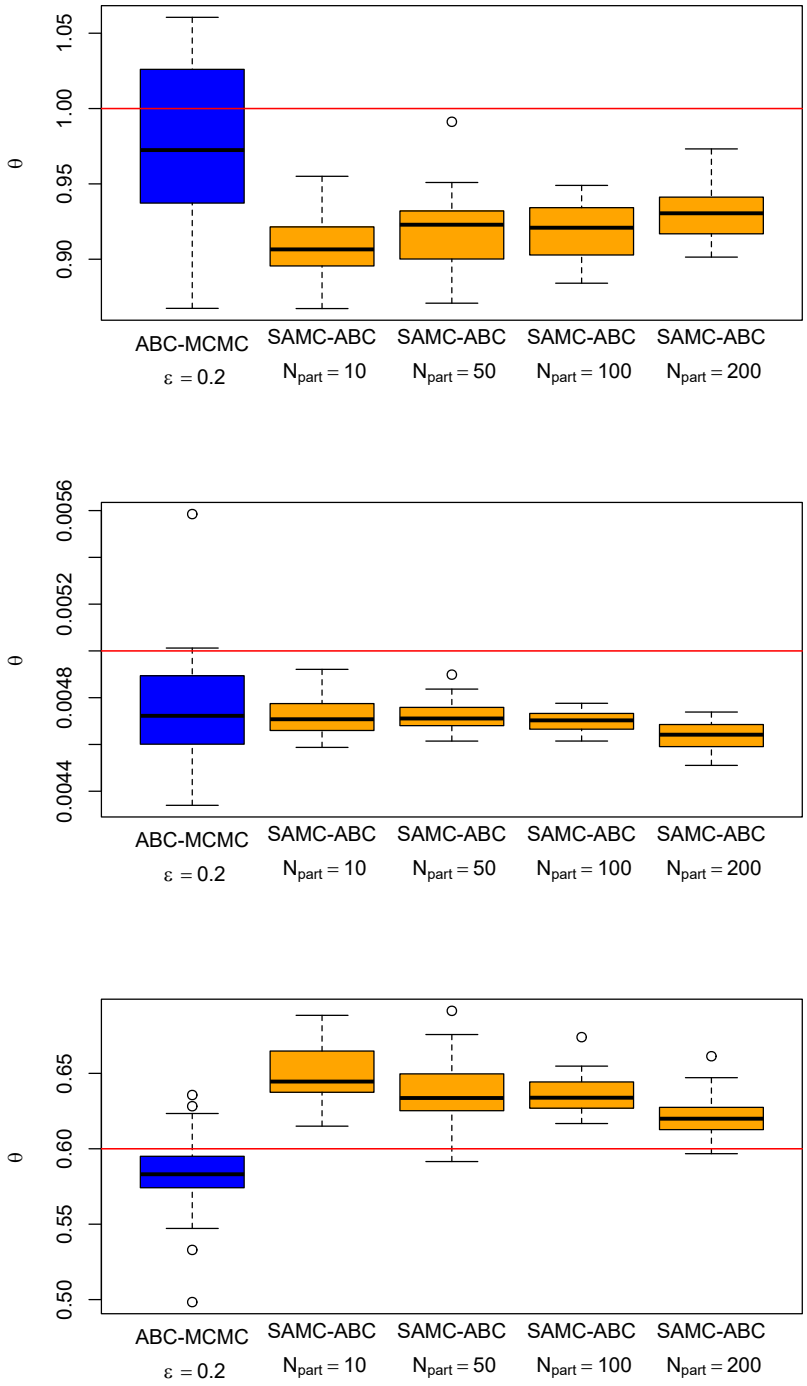

Figure 2.2: Comparison of empirical means of ABC-MCMC and SAMC-ABC for various grid sizes. The deterministic schedule here has a value of $t_0 = 50$ and $b = 0.7$. The red line indicates the true values.Top figure is parameter θ_1, middle is θ_2, and bottom is θ_3. The chains were run for 20000 iterations and for 40 replications each.

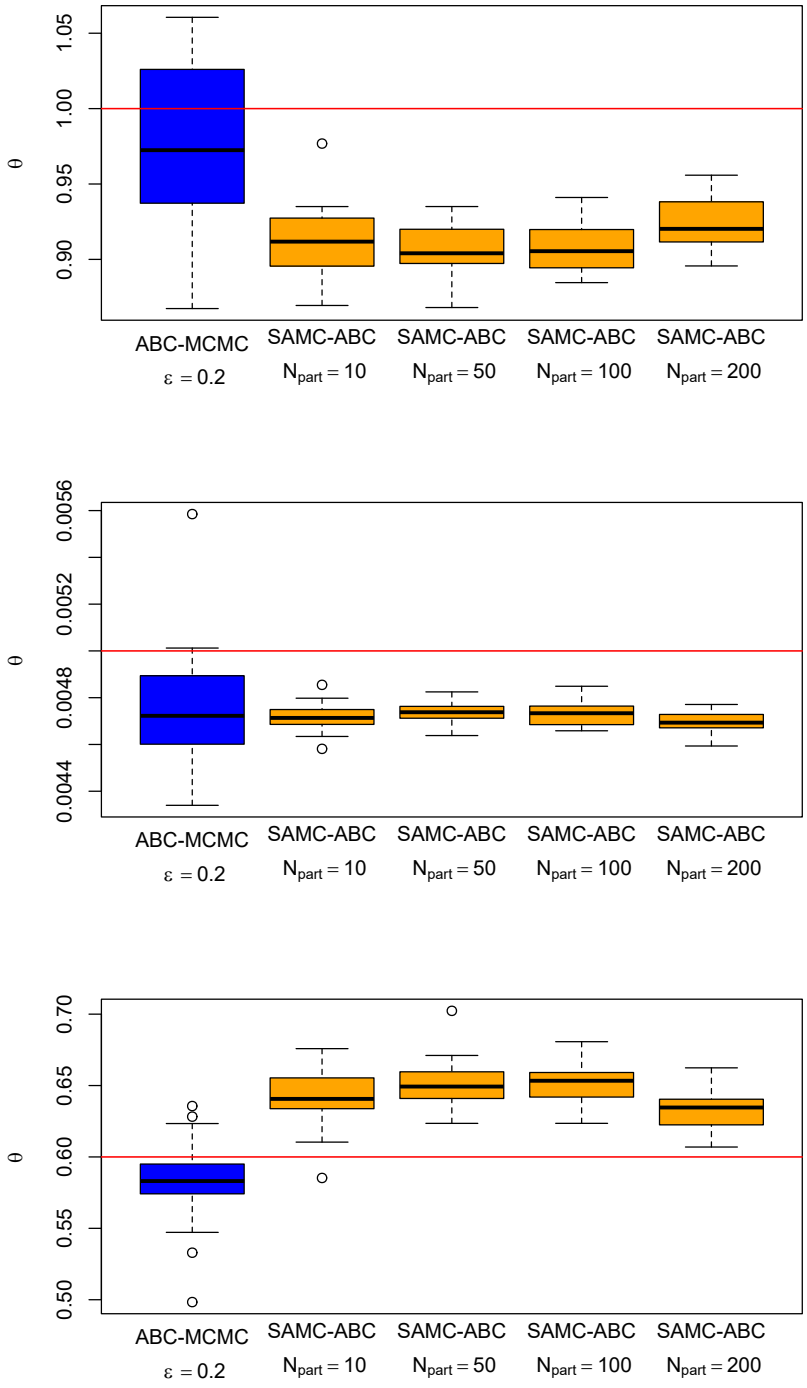

Figure 2.3: Comparison of empirical means of ABC-MCMC and SAMC-ABC for various grid sizes. The deterministic schedule here has a value of $t_0 = 100$ and $b = 0.7$. The red line indicates the true values.Top figure is parameter θ_1, middle is θ_2, and bottom is θ_3. The chains were run for 20000 iterations and for 40 replications each.

We see in figure 2.4 the effect of the parameter b is quite prominent. From equation 2.18 we can see that a very small value (comparatively speaking) of the parameter results in an increase in bias for two of the parameters θ_1, θ_2 while still decreasing the relative variance. The opposite is true for larger values. The choice here was deliberate. Given the form of equation 2.18 we can easily guess what the effect on the schedule would be. Take for example a chain with 20000 iterations, a small b, say 0.2, and $t_0 = 50$. In this case after the first 50 steps, γ_{50} will be 1 since $t_0 > 50^{0.2} = 2.186$. We would need more than 3.410^8 steps for the condition to be true thereby guaranteeing that the chain will never stop adapting for practical purposes. On the other hand for the same value of t_0 but for $b = 1.5$ it would mean that the chain will experience rapid decrease in the levels of adaptation after the 13^{th} time step and after a few more time steps practically stop adapting. We therefore directly observe the important effect our deterministic schedule has on the estimates. A continuing adaptation procedure that does not diminish as the algorithm progresses in any significant way, results in estimates that are biased although albeit with a significantly reduced variance. The tradeoff is clear: the price to pay for adaptation is that of bias. The algorithm allows larger values of epsilon to be accepted, thereby biasing the sample towards parameter values that are further from the true posterior values. Of course there are ways one might proceed in order to try and account for that bias.

As we described in the introductory sections, an importance sampling step can be made as a post-processing step in order to try and reduce the bias. Nevertheless is not exactly clear despite the simple form of equation 2.17 as we are aiming to estimate the ψs and those estimates come in the form of h as defined in the end of the previous section. One of the outputs of the algorithm is a vector of these values h, indicating a value for each sub-region for which we have sliced our space (here of epsilon). Assuming we want an equal sampling frequency for each partition we could by calculate these bias weights and then resample our values form the theta chain with the weights proportional. Of course that would assume that we would like an equal visit frequency in each partition. That would be the case if we wanted to partition only the parameter space θ and hence induce bias by that, or partition depending on log-likelihood and bias by that or even jointly. In these cases one could see why it would make sense to have an equal number of samples from each partition. In our

case the partitioning is on ϵ space (conditional on θ) and therefore an equal visit frequency would imply that we want samples from small and larger values of ϵ to be treated equally which is unreasonable. In the ABC context (assuming the correct model) we want the smallest value of epsilon possible for a given computational effort and therefore we would want more samples to be taken from the smallest values. We should remember that the entire motivation of the algorithm is to form such partition in order to aid exploration.

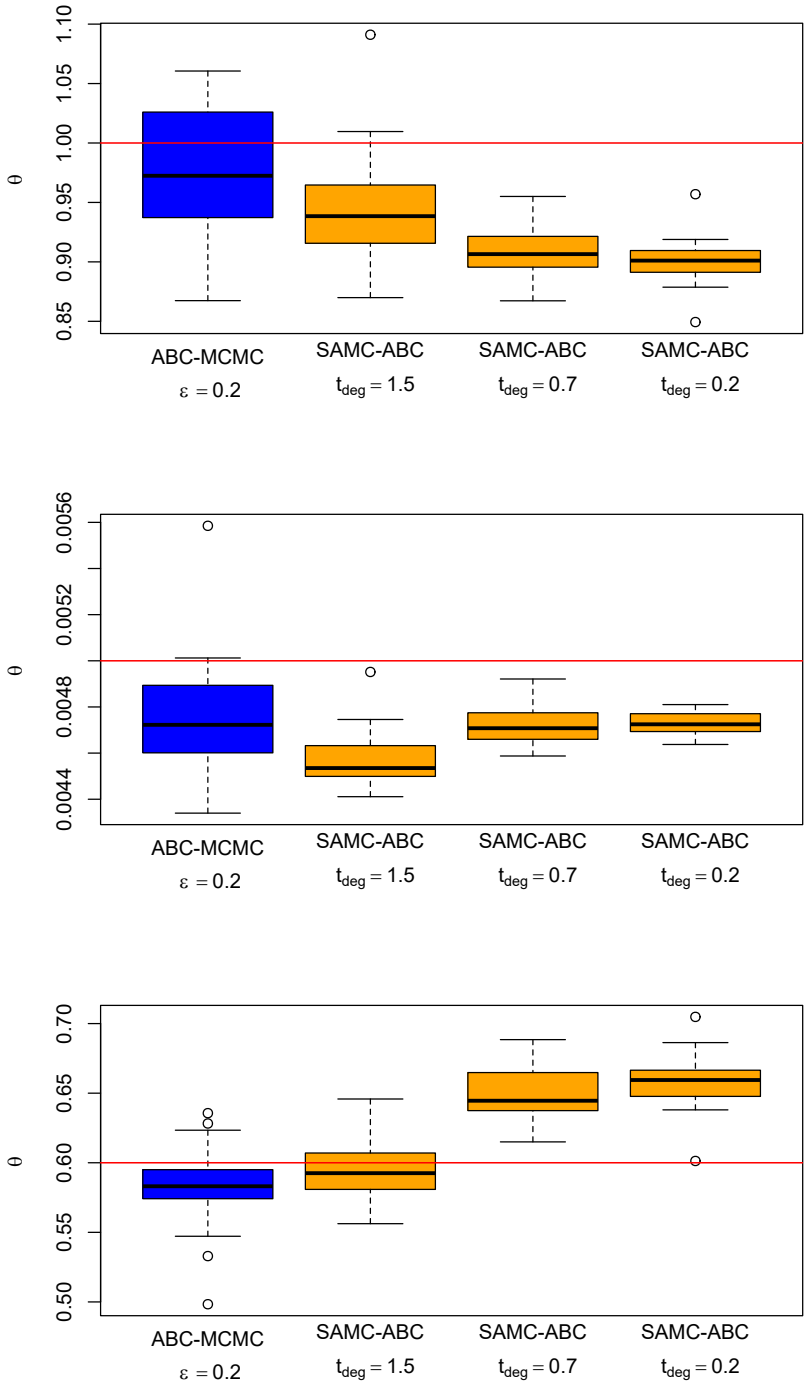

Figure 2.4: Comparison of empirical means of ABC-MCMC and SAMC-ABC for various time schedules. The grid size is fixed at $N_{grid} = 100$ while the decay factor of the schedule is fixed at $b = 0.7$. The red line indicates the true values.Top figure is parameter θ_1, middle is θ_2, and bottom is θ_3. The chains were run for 20000 iterations and for 40 replications each.

2.2.3 Conclusions

We have performed a comparison between standard ABC-MCMC and the SAMC-ABC algorithm for a different set of parameters controlling the SAMC-ABC algorithm. We found that there is a marked reduction in variance compared to ABC-MCMC as the exploration of the space is made more efficient due to the adaptive nature of the algorithm. Nevertheless, a bias is induced due to the nature of the algorithm which can be made worse by allowing the algorithm to adapt for prolonged periods of time. The partition size exhibits the most significant improvement on the quality of estimates of the parameters, as a relatively modest increase from 10 to 200 partitions significantly improves accuracy, and decreases the bias. The aforementioned effect can nonetheless, get again diminished if the adaptation continues for too long. Given that theoretically predicted and experimentally shown the samples are indeed biased and we propose a importance sampling correction, although it is not clear at this point in time exactly how this correction should be performed as it depends on the desired sampling frequency for each partition. The estimates of the bias weights are an output of the algorithm and therefore specifying a simple $1/d$ sampling frequency should give us the desired weights, yet that does not result in improved and reduced biased estimates. It is clear that a more sophisticated method of post correction should be performed given the very promising results of the method as the variance reduction has already demonstrated.

Chapter 3

Adaptive noisy exchange algorithm

3.1 Intractable and doubly intractable densities

A considerable amount of effort has been invested in trying to infer parameters of models (in a Bayesian formalism), where the posterior is intractable; meaning one cannot evaluate

$$\pi(\boldsymbol{\theta}|\mathbf{y}) \approx p(\boldsymbol{\theta})g(\mathbf{y}|\boldsymbol{\theta}) \tag{3.1}$$

pointwise (we have suppressed the dependence on the prior hyperparameters like those appearing in 1.1 of section 1.2 for the sake of notational convenience, but in a great number of instances sufficiently complex phenomena to be modelled will require more complex formulation). The intractability of the likelihood term $g(\mathbf{y}|\boldsymbol{\theta})$ can be due to a variety of reasons. For instance, it might be the case that the data we have available make that likelihood term completely impractical to calculate as the computational resources to do so would be extremely large (keeping in mind that the calculation would need for example to be performed N times during a simple MCMC algorithm of N

steps). Another issue would be that $g(\mathbf{y}|\boldsymbol{\theta}) = \int_{\mathcal{X}} g(\mathbf{y}, \mathbf{x}|\boldsymbol{\theta}) d\mathbf{x}$ due to a very large number of latent variables $\mathbf{x}$ being present, renders it a very high dimensional integral, as for example in state space models, where additionally the formulated transition densities are also unknown. Some other examples would be the cases where $g(\mathbf{y}|\boldsymbol{\theta})$ cannot be evaluated, but we can draw samples from it, as is the case where $\{\mathbf{y}_{obs}|\boldsymbol{\theta}\}$ is given by a complex stochastic (computer) model, where any value of the parameter $\boldsymbol{\theta}$ serves as input in a forward simulator where the output is $\mathbf{y}_{out}$, yet the likelihood function itself is not known in an explicitly functional form. Last, but not least we could have $g(\mathbf{y}|\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})}\gamma(\mathbf{y}|\boldsymbol{\theta})$, with $Z(\theta)$ an intractable normalising constant (but the term $\gamma(\mathbf{y}|\theta)$ tractable). This for example occurs in formulation of the Ising model or a Markov random field.

The last case is where the focus of this chapter lies. In fact we will be concerned with the case of doubly intractable distributions and the proof of convergence of one such algorithm for generating draws from $\pi(\boldsymbol{\theta}|\mathbf{y})$. But first, let us take a brief tour of the development of algorithms that aimed to tackle such issues.

Consider for example a standard MCMC algorithm targeting some posterior distribution $\pi(\boldsymbol{\theta}|\mathbf{y}) = \frac{\gamma(\mathbf{y}|\boldsymbol{\theta}^*)}{Z(\theta)}p(\theta)$. In this case the MCMC acceptance probability is equal to

$$\alpha(\boldsymbol{\theta}, \boldsymbol{\theta}^*) = \min\left\{1, \frac{\gamma(\mathbf{y}|\boldsymbol{\theta}^*)p(\boldsymbol{\theta}^*)q(\boldsymbol{\theta}|\boldsymbol{\theta}^*)}{\gamma(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})q(\boldsymbol{\theta}^*|\boldsymbol{\theta})}\frac{Z(\boldsymbol{\theta})}{Z(\boldsymbol{\theta}^*)}\right\}. \tag{3.2}$$

with $q(\cdot)$ some arbitrary proposal distribution. This ideal algorithm cannot be implemented since the ratio $\frac{Z(\boldsymbol{\theta})}{Z(\boldsymbol{\theta}^*)}$ is intractable (since the numerator/denominator are). Notice here that is because the normalizing constant is a function of θ that this intractability arises as this would not be the case in regular MCMC where their evaluation is not needed. The doubly intractable term is coined in Murray et al. [2006] due the fact that MCMC algorithms (as estimators) are approximations (unless one has an infinitely long chain) and each step requires an infeasible computation. In order to overcome such issues a number of approaches have been proposed. We will showcase a few since they provide the history and explain the natural evolution of such approaches

towards the current problem discussed here.

3.2 Doubly intractable likelihoods and adaptive noisy exchange

The authors in Møller et al. [2004], construct a pseudo-marginal MCMC algorithm through the use of an unbiased importance sampling estimator

$$\frac{1}{Z(\boldsymbol{\theta})} \approx \frac{q_u(\mathbf{x}|\boldsymbol{\theta}, \mathbf{y})}{\gamma(\mathbf{x}|\boldsymbol{\theta})} \tag{3.3}$$

where $\boldsymbol{x} \sim g(\cdot|\boldsymbol{\theta})$, for the numerator in 3.2 (and the analogous in the denominator, and where q_x any normalized distribution, but we often choose one such as $q_x(\mathbf{x}|\boldsymbol{\theta}^*, \mathbf{y}) = g\big(\mathbf{x}|\hat{\boldsymbol{\theta}}\big)$, and taking $\hat{\boldsymbol{\theta}}$ by point estimate; for example maximum pseudo-likelihood estimate etc. The procedure above is best seen as a single auxiliary variable MCMC, while the estimator can be improved (and remain unbiased of course), by the usage of a number of importance points, M, and/or using approaches such as annealed importance sampling Neal [2001] with intermediate target sequence instead of standard IS. For example let:

$$f_i(\cdot|\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}, \mathbf{y}) \approx \gamma_i(\cdot|\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}, \mathbf{y}) = \gamma(\cdot|\boldsymbol{\theta})^{(i-1)/(a-1)} \gamma(\hat{\boldsymbol{\theta}})^{(a-i)/(a-1)} \tag{3.4}$$

be that sequence of intermediate targets, and K_i some MCMC kernel with target f_i. Then for each i, $x_i \sim K_i(\cdot|x_{i+1})$ the improved estimator is :

$$\frac{Z(\hat{\boldsymbol{\theta}})}{Z(\boldsymbol{\theta})} \approx \frac{1}{M} \sum_{m=1}^{M} \prod_{i=2}^{a} \frac{\gamma_{i-1}(x_i^{(m)}|\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}, \mathbf{y})}{\gamma_i(x_i^{(m)}|\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}, \mathbf{y})} \tag{3.5}$$

with the added benefit of the additional $Z(\hat{\boldsymbol{\theta}})$ term cancelling in the acceptance ratio. This modification (usage of AIS instead of plain IS) was suggested by Murray et al. [2006], with the approach named "multiple auxillary variable" (MAV) method. Building upon the previously mentioned method Murray et al. [2006] propose an exact approximate MCMC algorithm by making use of an

unbiased importance sampling estimator

$$\frac{Z(\boldsymbol{\theta})}{Z(\boldsymbol{\theta}^*)} \approx \frac{\gamma(\mathbf{x}|\boldsymbol{\theta})}{\gamma(\mathbf{x}|\boldsymbol{\theta}^*)} \tag{3.6}$$

where $\boldsymbol{x} \sim g(\cdot|\boldsymbol{\theta}^*)$, which one notices, it directly approximates the ratio in equation 3.2 (the equivalent **IS** approach would not be possible). The aforementioned estimator can arguably be shown to exhibit improved (inferential) performance using AIS, by using multiple importance points resulting in an inexact ("noisy") algorithm Alquier et al. [2016].

All of these approaches of course are based on the requirement of being able to simulate from the likelihood $\boldsymbol{x} \sim g(\cdot|\boldsymbol{\theta}^*)$ and for most problems with doubly intractable g it's impossible to perform that action exactly. Caimo and Friel [2011] instead propose the idea of using a long MCMC run and taking the last point to be x, whereas Everitt [2012] demonstrate that the incurred bias goes to zero as the length of the MCMC chain goes to infinity. Authors refer to this method as applied to the exchange algorithm as "approximate exchange algorithm". Interestingly Liang [2010], suggest the possibility of using only one MCMC step for generating x, while also being an approximate exchange method.

3.3 Intractable normalising constants and the augmented space idea

In order to now sample from the target given in 3.1 and approximate the ratio appearing in 3.2 the approach of Murray et al. [2006] that we have only outlined so far can be used. The authors introduce an auxiliary variable[1] $x \in \mathrm{X}$ and consider the following augmented target density

$$p\left(\theta^{*}, x, \theta \mid y\right) \propto g(y \mid \theta) p(\theta) q\left(\theta^{*} \mid \theta\right) p\left(x \mid \theta^{*}\right) \tag{3.7}$$

which has $g(\theta \mid y)$ as its θ-marginal distribution. An MCMC algorithm with $p\left(\theta^{*}, x, \theta \mid y\right)$ as its target density has the following acceptance probability

$$\alpha\left(\theta, \theta^{*}, x\right) = \frac{\gamma\left(y \mid \theta^{*}\right) p\left(\theta^{*}\right) q\left(\theta \mid \theta^{*}\right) \gamma(x \mid \theta)}{\gamma(y \mid \theta) p(\theta) q\left(\theta^{*} \mid \theta\right) \gamma\left(x \mid \theta^{*}\right)} \frac{Z(\theta) Z\left(\theta^{*}\right)}{Z\left(\theta^{*}\right) Z(\theta)} \tag{3.8}$$

with the intractable normalising constants at the end canceling out. An important point here is that it is often infeasible to sample perfectly from $p(x|\theta)$, as is the case for example in exponential random graph models. It is nonetheless, possible to use the last sample from an MCMC chain as an approximate sample from $p(x|\theta)$. This has been named the approximate exchange algorithm and theoretical guarantees for it are provided in Everitt [2012]. Despite the fact that the exchange algorithm overcomes the intractable nature of the standard MH acceptance ratio, there is no guarantee that good mixing is to be expected. The authors in Alquier et al. [2016] propose a noisy version of the algorithm with the aim of improving the relative efficiency. Let us give some details of this algorithm.

The authors in Alquier et al. [2016] view the exchange algorithm as a replacement of the ratio of normalizing constants in 3.2 with $\gamma(x \mid \theta)/\gamma(x \mid \theta^{*})$ present in 3.8. One can show that $\gamma(x \mid \theta)/\gamma(x \mid \theta^{*})$ is an unbiased estimator

[1] in the previous section we indicated with $\mathbf{x}$ the potential vector character of x; henceforth we will simply omit the boldface notation while still assuming both data and parameters can be of arbitrary dimensionality. The majority of arguments carry over to the vectorial case albeit with more cumbersome calculations. Where that is needed it will be indicated so.

of the ratio of normalising constants

$$\mathrm{E}_{p(x|\theta^*)}\left[\frac{\gamma(x \mid \theta)}{\gamma\left(x \mid \theta^*\right)}\right] = \frac{Z(\theta)}{Z\left(\theta^*\right)} \tag{3.9}$$

A behaviour that occurs in pseudo-marginal style methods is that of poor mixing due to the simple sample nature of the estimator Andrieu and Roberts [2009]. It is therefore possible that the relatively poor mixing occuring in the exchange algorithm is due to the same phenomenon; i.e the single sampler estimator of the ratio of normalising constant may exhibit high variance. The authors in Doucet et al. [2015] show that if the pseudo-marginal chain does not exhibit any stickiness then estimator must have low enough variance. Due to this reason Alquier et al. [2016] develop the noisy exchange algorithm where multiple copies of the auxiliary variable are considered, and averaged:

$$\frac{1}{N}\sum_{n=1}^{N}\frac{\gamma\left(x_n \mid \theta\right)}{\gamma\left(x_n \mid \theta^*\right)} \approx \frac{Z(\theta)}{Z\left(\theta^*\right)} \tag{3.10}$$

Despite the fact that there exists an improvement in mixing of the chain, the intended target $p(\theta \mid y)$, is no longer the invariant measure of that chain. Nonetheless, Alquier et al. [2016] prove that the considered algorithm does in fact converge to the true posterior for increasing N, and additionally provided bounds on the total variance distance between the approximate and true posteriors (for a geometrically ergodic Markov chain). An additional potential benefit of the proposed algorithms is when the N simulations one needs for the noisy exchange version (and assuming one can have perfect sampling of $p(x \mid \theta)$), can be generated independently on multi-core machines. Furthermore, an additional approach can be that of the approximate noisy exchange algorithm where the final N samples from the MCMC procedure can be used after some burn-in period (in the case where the auxiliary chain is required for approximate simulation). Of course one could argue that it is always a case of trade-offs: whether the exchange or the noisy exchange version of the algorithm are better for the task at hand depends on whether one can run a chain with better mixing characteristics, while dealing with the computational overhead of generating N model simulations per iteration. It is therefore the case

that despite the benefits of improved mixing, that is indirectly reducing the overall computational effort by producing a better approximation, the number of model simulations one requires is very large (in typical scenarios). This fact is especially bothersome since a lot of doubly intractable model, some of which we have mentioned, such as the exponential random graphs and the Ising model as applied to large images are rather expensive to simulate. It is therefore the case that algorithms that will in some way reduce the required model simulation are very much desirable. The adaptive noisy exchange method aims to fill that gap.

A first approach in reducing the required number of simulations is that instead of estimating the ratio $Z(\theta)/Z(\theta^*)$, we estimate the ratio

$$\frac{\widehat{Z\left(\theta^*\right)}}{Z(\theta)} = \frac{1}{N}\sum_{n=1}^{N}\frac{\gamma\left(x_n \mid \theta^*\right)}{\gamma\left(x_n \mid \theta\right)} \tag{3.11}$$

where $x_1, \ldots, x_N \sim p(x \mid \theta)$. The main thing to notice here is that we can reuse the values $x_1, \ldots, x_N$ generated at current step θ_t of the chain. Therefore we only need to generate new values only when a proposal is accepted. The Metropolis-Hastings acceptance probability for this approach is given by

$$\alpha\left(\theta, \theta^*\right) = \frac{\gamma\left(y \mid \theta^*\right) p\left(\theta^*\right) q\left(\theta \mid \theta^*\right)}{\gamma(y \mid \theta) p(\theta) q\left(\theta^* \mid \theta\right)} \left(\frac{\widehat{Z\left(\theta^*\right)}}{Z(\theta)}\right)^{-1} \tag{3.12}$$

While Friel and Drovandi [2019] have observed that this approach reduces significantly the number of required model simulations, it has also been the case that a tendency to give conservative estimates of the posterior distribution is observed. Specifically, the posterior variances seem to be inflated. Therefore we required an approach that can reduce simulation frequency and give posterior estimates closer to the noisy exchange algorithm, at the same time. The authors in Boland et al. [2018] propose an idea of pre-computation in order to accelerate the noisy exchange algorithm. Initially, they start by calculating an estimate of the initial point and an estimate of the Hessian of the log posterior at the same point. Subsequently a grid of θ values is created over the parameter space and they then generate N simulations at every one

of those points. Let us denote the collection of precomputing information as $\left\{\theta_m, \{x_m^j\}_{j=1}^N\right\}_{m=1}^M$ for a total of M grid points. Define θ_m as support points. During a run of the noisy exchange algorithm, we obtain for each combination of θ and θ^*, the ratio $Z(\theta)/Z(\theta^*)$ can be estimated by initially identifying a path of support points that connects θ and θ^* through the parameter space. Denoting these points as $\left(\theta_{(1)}, \theta_{(2)}, \ldots, \theta_{(C)}\right)$ where $\theta_{(1)}$ and $\theta_{(C)}$ are the closest support points to θ and θ^*, respectively. Then, the ratio of normalising constants can be estimated by noting that

$$\frac{Z(\theta)}{Z(\theta^*)} = \frac{Z(\theta)}{Z\left(\theta_{(1)}\right)} \times \frac{Z\left(\theta_{(1)}\right)}{Z\left(\theta_{(2)}\right)} \times \cdots \times \frac{Z\left(\theta_{(C-1)}\right)}{Z\left(\theta_{(C)}\right)} \times \left(\frac{Z(\theta^*)}{Z\left(\theta_{(C)}\right)}\right)^{-1} \tag{3.13}$$

We can therefore estimate the first C terms based on 3.10 using the pre-generated samples at the support points corresponding to those in 3.13. In general the $(n+1)^{th}$ term inside the parenthesis can be estimated by the pre-generated samples at the n^{th} support point. We can therefore see that after all the pre-computed samples have been generated the noisy exchange algorithm does not use any model simulation. The authors in Boland et al. [2018], consider different approaches of constructing such a path, with one approach being directly linked points (i.e. only 2 support points), the ones closest to the points θ and θ^*. Another approach is to take the average of estimates over multiple paths. Friel and Drovandi [2019] focus instead on the direct path. Finally, one important issue would be how would one know a priori where should the grid be placed in order to provide proper coverage of the posterior support, and consequently points may be placed where there is negligible posterior support.

3.4 Adaptive noisy exchange algorithm and proof of convergence

Let us reiterate the issue at hand: a characteristic of doubly-intractable problems is that the likelihood function associated to the model cannot be evaluated pointwise due to an intractable normalising constant. Given a set of

parameters $\theta \in \Theta \subseteq \mathbb{R}^d$, the likelihood function is given by

$$f_\theta(y) = \frac{g_\theta(y)}{Z_\theta},$$

where g_θ [2]is the unnormalised density and Z_θ is the intractable constant . Therefore, given a prior distribution p_0, the posterior for θ given some data y is given by

$$\pi(\theta \mid y) \propto f_\theta(y)\, p_0(\theta).$$

If we were able to compute Z_θ, a standard Metropolis-Hastings algorithm would propose moves according to q and would accept such moves according to the following acceptance probability

$$\alpha(\theta, \vartheta) := \min\left\{1, \frac{g_\vartheta(y)\, p_0(\vartheta)\, q(\vartheta, \theta) Z_\theta}{g_\theta(y)\, p_0(\theta)\, q(\theta, \vartheta) Z_\vartheta}\right\}. \tag{3.14}$$

However, since such a method is not feasible a commonly used alternative is to estimate the ratio $R(\theta, \vartheta) := Z_\theta / Z_\vartheta$ via

$$\widehat{R}_N(\theta, \vartheta) \equiv \widehat{R}_{U_\vartheta^{1:N}}(\theta, \vartheta) := \frac{1}{N}\sum_{i=1}^{N} \frac{g_\theta}{g_\vartheta}\left(U_\vartheta^{(i)}\right), \quad \text{where } U_\vartheta^{1:N} := \left\{U_\vartheta^{(i)}\right\}_{i=1}^{N} \overset{iid}{\sim} f_\vartheta(\cdot),$$

as in 3.11. Appealing properties of the previous estimator are the following: for any $(\theta, \vartheta) \in \Theta^2$

$$\mathbb{E}\left[\widehat{R}_N(\theta, \vartheta)\right] = R(\theta, \vartheta),$$
$$\text{and} \quad \lim_{N\to\infty} \widehat{R}_N(\theta, \vartheta) \overset{as}{=} R(\theta, \vartheta).$$

Notice that the above estimate requires the ability to draw auxiliary samples from the likelihood f_ϑ; this will only be possible in specific scenarios, e.g. using the coupling from the past algorithm Propp and Wilson [1996]. Alternatively,

[2]we denote $g_\theta(y) = g(y|\theta)$ and similarly for the rest of the formal treatment, we also drop the bold vector notation we have used so far for visual ease; it is nonetheless assumed that all of the parameters and observations can of course be part of a multidimensional space with arbitrary large vector components

one can run an MCMC algorithm targeting f_ϑ (which doesn't require the computation of any Z_θ) and then use the resulting values after a "sufficiently long" run.

However, the downside of plugging in the estimator $\widehat{R}_N$ into a Metropolis-Hastings algorithm is that this usually results in a noisy method in the sense that it does not target the desired posterior distribution. An exception to this is when $N = 1$ resulting in the exchange algorithm Murray et al. [2006], which is an exact method. Nevertheless, the variance of the estimator $\widehat{R}_{N=1}$ is usually very large, which means the exchange algorithm might be inefficient at exploring the desired posterior. For this reason, increasing N is not necessarily a bad idea provided we can control the introduced bias.

Our focus is at reducing the computational cost of implementing a standard noisy method since at each iteration we would need to simulate the set of auxiliary variables $U_\vartheta^{1:N}$, which are immediately forgotten after accepting or rejecting a move. Instead, we aim at reusing (as much as possible) the generated variables $V_{S,N} := \left\{U_{\bar{\theta}}^{1:N}\right\}_{\bar{\theta}\in S}$, where S denotes the set of values $\bar{\theta} \in \Theta$ for which we have simulated a set of auxiliary variables $U_{\bar{\theta}}^{1:N}$. Reusing these variables can be done by finding a path $\mathcal{P}_{\theta,\vartheta} = \left\{\bar{\theta}_1, \cdots, \bar{\theta}_{m_{\theta,\vartheta}}\right\} \subseteq S$, where $m_{\theta,\vartheta} \in \mathbb{N}_0 := \mathbb{N} \cup \{0\}$, between the current θ and the proposed ϑ; we then estimate $R(\theta, \vartheta)$ using

$$\widehat{R}_{S,N}^{(1)}(\theta, \vartheta) \equiv \widehat{R}_{\mathcal{P}_{\theta,\vartheta},V_{S,N}}(\theta, \vartheta) := \prod_{i=1}^{m_{\theta,\vartheta}+1} \widehat{R}_N\left(\bar{\theta}_{i-1}, \bar{\theta}_i\right),$$

where $\bar{\theta}_0 = \theta$ and $\bar{\theta}_{m_{\theta,\vartheta}+1} = \vartheta$. This estimator is once again unbiased and consistent with respect to $R(\theta, \vartheta)$ for any $(\theta, \vartheta) \in \Theta^2$, and depending on the chosen path $\mathcal{P}_{\theta,\vartheta}^{(m)}$, the variance of $\widehat{R}_{S,N}^{(1)}$ can be reduced to only a fraction of the variance of $\widehat{R}_N$ (Neal [2001], section 4). Finally, notice that in order to compute the last term in the product of $\widehat{R}_{m,N}$ we need the set of variables $U_\vartheta^{1:N}$, implying we still need to simulate auxiliary variables whenever we propose a state ϑ that has not been visited before. This can be wasteful if the proposed value ϑ is "close" to some $\bar{\theta} \in S$. Therefore, we will only generate $U_\vartheta^{1:N}$ whenever the $d_{\vartheta,S} := \min_{\bar{\theta}\in S} \left\|\vartheta - \bar{\theta}\right\| > \varepsilon$, for some metric $\|\cdot\|$ on Θ and some $\varepsilon > 0$. Thus, if $d_{\vartheta,S} < \varepsilon$ we could estimate $R(\theta, \vartheta)$ using

$$\widehat{R}^{(2)}_{S,N}(\theta,\vartheta) := \frac{\widehat{R}_{\mathcal{P}_{\theta,\vartheta},V_{S,N}}\left(\theta,\bar{\theta}_{m_{\theta,\vartheta}}\right)}{\widehat{R}_N\left(\vartheta,\bar{\theta}_{m_{\theta,\vartheta}}\right)}.$$

Note that the numerator $\widehat{R}_{\mathcal{P}_{\theta,\vartheta},V_{S,N}}\left(\theta,\bar{\theta}_{m_{\theta,\vartheta}}\right)$ is very similar to $\widehat{R}^{(1)}_{S,N}$, but not precisely the same: it uses the same path as $\widehat{R}^{(1)}_{S,N}$, but taking terms in the product up to the point before the end point ϑ in the path. Despite this estimator being biased, it remains consistent as $N \to \infty$ with respect to $R(\theta,\vartheta)$ for any $(\theta,\vartheta) \in \Theta^2$, ratios of unbiased estimators for this type of models have been explored before as in Boland et al. [2018].

We now present the resulting algorithm and the corresponding probability kernel when using the previous strategies by computing either $\widehat{R}^{(1)}_{S,N}$ or $\widehat{R}^{(2)}_{S,N}$. We also present comparisons to the ideal algorithm, i.e. the one using the intractable $\alpha(\theta,\vartheta)$.

Algorithm 11: Adaptive Noisy Exchange

Input: θ_0, S_0, $V_{S_0,N} = \left\{U_{\bar{\theta}}^{1:N} \mid \bar{\theta} \in S_0 \text{ and } U_{\bar{\theta}}^{1:N} \overset{iid}{\sim} f_{\bar{\theta}}(\cdot)\right\}$.

begin
- Sample $\vartheta \sim q(\cdot \mid \theta_0)$.
- Choose a path $\mathcal{P}_{\theta_0,\vartheta} = \left\{\bar{\theta}_1, \cdots, \bar{\theta}_{m_{\theta_0,\vartheta}}\right\} \subseteq S_0$ connecting θ_0 and ϑ.
- **if** $d_{\vartheta,S_0} = \min_{\bar{\theta}\in S_0} \|\vartheta - \bar{\theta}\| \geq \varepsilon$ **then**
  - Draw $U_{\vartheta}^{1:N} \overset{iid}{\sim} f_{\vartheta}(\cdot)$ and compute $\widehat{R}^{(1)}_{S_0,N}(\theta_0,\vartheta)$ using $\mathcal{P}_{\theta_0,\vartheta}$ and $V_{S_0,N}$; Define $S_1 := S_0 \bigcup \{\vartheta\}$, $V_{S_1,N} := V_{S_0,N} \bigcup \left\{U_{\vartheta}^{1:N}\right\}$. Set $\theta_1 = \vartheta$ with probability $\tilde{\alpha}^{(1)}_{S_1,N}(\theta_0,\vartheta)$, otherwise set $\theta_1 = \theta_0$;
- **else**
  - Compute $\widehat{R}^{(2)}_{S_0,N}(\theta_0,\vartheta)$ using $\mathcal{P}_{\theta_0,\vartheta}$ and $V_{S_0,N}$; Define $S_1 = S_0$, $V_{S_1,N} := V_{S_0,N}$. Set $\theta_1 = \vartheta$ with probability $\tilde{\alpha}^{(2)}_{S_1,N}(\theta_0,\vartheta)$, otherwise set $\theta_1 = \theta_0$
- **end**

end

Output: θ_1, S_1, $V_{S_1,N}$

3.5 Algorithm and kernel

Based on the description above, we now define the algorithms to be studied in this section. Firstly we define the following approximate acceptance probabil-

ities, for $j \in \{1, 2\}$

$$\tilde{\alpha}_{S,N}^{(j)}(\theta, \vartheta) = \min\left\{1, \frac{g_\vartheta(y)\, p_0(\vartheta)\, q(\vartheta, \theta)}{g_\theta(y)\, p_0(\theta)\, q(\theta, \vartheta)} \widehat{R}_{S,N}^{(j)}(\theta, \vartheta)\right\}.$$

It is worth noticing that a) we obtain the ideal metropolis-hastings kernel by replacing $\tilde{\alpha}_{S,N}^{(1)}$ and $\tilde{\alpha}_{S,N}^{(2)}$ with α. Furthermore, the algorithm defines a Markov chain for the variables $\{(\theta_t, S_t, V_{S_t,N})\}_{t\geq 0}$, with the transition kernel given by:

$$\begin{aligned}
&\tilde{P}(\theta_0, S_0, V_{S_0,N}; \theta_1, S_1, V_{S_1,N}) := \\
&\quad q(\theta_0, \theta_1)\, \mathbb{1}(d_{\theta_1,S_0} > \varepsilon)\, f_{\theta_1}^{\otimes N}\left(U_{\theta_1}^{1:N}\right) \tilde{\alpha}_{S_1,N}^{(1)}(\theta_0, \theta_1)\, \delta_{S_0\cup\{\theta_1\}, V_{S_0,N}\bigcup\left\{U_{\theta_1}^{1:N}\right\}}(S_1, V_{S_1,N}) \\
&\quad + \delta_{\theta_0}(\theta_1) \int_{\left\{\vartheta | d_{\vartheta,S_0} > \varepsilon\right\}} f_\vartheta^{\otimes N}\left(U_\vartheta^{1:N}\right) \left(1 - \tilde{\alpha}_{S_1,N}^{(1)}(\theta_0, \vartheta)\right) \delta_{S_0\cup\{\vartheta\}, V_{S_0,N}\bigcup\left\{U_\vartheta^{1:N}\right\}}(S_1, V_{S_1,N})\, q(\theta_0, d\vartheta) \\
&\quad + q(\theta_0, \theta_1)\, \mathbb{1}(d_{\theta_1,S_0} \leq \varepsilon)\, \tilde{\alpha}_{S_1,N}^{(2)}(\theta_0, \theta_1)\, \delta_{S_0, V_{S_0,N}}(S_1, V_{S_1,N}) \\
&\quad + \delta_{\theta_0,S_0,V_{S_0,N}}(\theta_1, S_1, V_{S_1,N}) \int_{\left\{\vartheta | d_{\vartheta,S_0} \leq \varepsilon\right\}} \left(1 - \tilde{\alpha}_{S_1,N}^{(2)}(\theta_0, \vartheta)\right) q(\theta_0, d\vartheta).
\end{aligned} \tag{3.15}$$

The four lines on the right hand side of the expression above correspond, respectively, to the following four cases:

- ϑ is outside of the currently defined region (the current union of balls around previously visited points), and the ϑ is accepted;
- ϑ is outside of the currently defined region, and the ϑ is rejected (in this line there is no integral over θ-space, as there usually is in a rejection, since we still need to keep track of ϑ through it becoming part of the set of saved points);
- ϑ is inside of the currently defined region, and the ϑ is accepted;
- ϑ is outside of the currently defined region, and the ϑ is rejected.

We will study the marginal (on θ-space) kernel of (3.15) , given below. This kernel is no longer a Markov kernel (unless we use a fixed S and V).

$$\begin{aligned}\tilde{P}_{S_n,V_{S_n,N}}(\theta_n;\theta_{n+1}) = {} & q(\theta_n,\theta_{n+1})\,\mathbb{1}\left(d_{\theta_{n+1},S_n}>\varepsilon\right)\mathbb{E}_{U^{1:N}_{\theta_{n+1}}\sim f^{\otimes N}_{\theta_{n+1}}}\left[\tilde{\alpha}^{(1)}_{S_n\cup\{\theta_{n+1}\},N}(\theta_n,\theta_{n+1})\right] \\ & + \delta_{\theta_n}(\theta_{n+1})\int_{\left\{\vartheta|d_{\vartheta,S_n}>\varepsilon\right\}}\mathbb{E}_{U^{1:N}_{\vartheta}\sim f^{\otimes N}_{\vartheta}}\left[1-\tilde{\alpha}^{(1)}_{S_n\cup\{\vartheta\},N}(\theta_n,\vartheta)\right]q(\theta_n,d\vartheta) \\ & + q(\theta_n,\theta_{n+1})\,\mathbb{1}\left(d_{\theta_{n+1},S_n}\leq\varepsilon\right)\tilde{\alpha}^{(2)}_{S_n,N}(\theta_n,\theta_{n+1}) \\ & + \delta_{\theta_n}(\theta_{n+1})\int_{\left\{\vartheta|d_{\vartheta,S_n}\leq\varepsilon\right\}}\left(1-\tilde{\alpha}^{(2)}_{S_n,N}(\theta_n,\vartheta)\right)q(\theta_n,\vartheta)\,d\vartheta\end{aligned} \tag{3.16}$$

Let $\gamma_{n,N} := \{S_n, V_{S_n,N}\}$, we want to show that for any starting point θ_0 and arbitrary $\delta > 0$ the following holds for large enough n and N

$$\left\|\mathbb{P}\left[\theta_n\in\cdot\mid\theta_0\right]-\pi\right\|_{TV}<\delta,$$

where $\mathbb{P}\left[\theta_n\in\cdot\mid\theta_0\right]$ denotes the conditional distribution of $\theta_n \mid \theta_0$. This will be done in two steps:

- First, we guarantee the existence of a finite stopping time $\tau\in\mathbb{N}$ such that $\gamma_n=\gamma_\tau$ for any $n\geq\tau$. This will imply, for $n\geq\tau$

$$\mathbb{P}\left[\theta_n\in A\mid\theta_0\right]=\mathbb{E}_{\tau,\theta_\tau,\gamma_{\tau,N}|\theta_0}\left[\tilde{P}^{n-\tau}_{\gamma_{\tau,N}}(\theta_\tau,A)\right]. \tag{3.17}$$

- Secondly, we show the existence of $n(\tau,S_\tau,\delta)\geq\tau$ and $N_0(n)$ large enough such that for $N\geq N_0$

$$\sup_{\theta,\gamma}\left\|\tilde{P}^{n-\tau}_{\gamma,\tau}(\theta,\cdot)-\pi\right\|_{TV}<\delta. \tag{3.18}$$

The desired result will be obtained by the triangle inequality and appli-

cation of Jensen's inequality:

$$\left\|\mathbb{P}\left[\theta_n \in \cdot \mid \theta_0, \gamma_{0,N}\right] - \pi\right\|_{TV} = \left\|\mathbb{E}_{\tau,\theta_\tau,\gamma_{\tau,N}|\theta_0,\gamma_{0,N}}\left[\tilde{P}^{n-\tau}_{\gamma_{\tau,N}}\left(\theta_\tau, \cdot\right)\right] - \pi\right\|_{TV} \tag{3.19}$$

$$\leq \mathbb{E}_{\tau,\theta_\tau,\gamma_{\tau,N}|\theta_0,\gamma_{0,N}}\left\|\tilde{P}^{n-\tau}_{\gamma_{\tau,N}}\left(\theta_\tau, \cdot\right) - \pi\right\|_{TV} \tag{3.20}$$

$$< \delta. \tag{3.21}$$

3.5.1 Adaptation

In order to conclude that equation (3.17) is indeed satisfied we will require the result below:

Proposition 1. *If Θ is compact, and for any starting point $\theta_0 \in \Theta$, there exists an a.s. finite random time τ at which the noisy adaptive chain stops adapting.*

Proof. Without loss of generality assume $\Theta \subseteq \mathbb{R}^d$ is a hyper-cube of length L, i.e. $vol\left(\Theta\right) = L^d$. Divide Θ into smaller contiguous cubes of volume $\left(\varepsilon/2\right)^d$, implying that Θ is made of $H = \left(2L/\varepsilon\right)^d$ smaller cubes.

Suppose that at time t_n the chain is at state $\theta_{t_n} = x$, and that the grid S_{t_n}, composed of points that are at least distance ε apart, is made of n points, i.e. $S_{t_n} = \{y_1, \ldots, y_n = x\}$, where $t_n \geq n$. This implies that at time t_n there exist at most $H - n$ cubes that are not fully covered by the set spanned by S_{t_n} given by

$$\bar{S}_{t_n,\varepsilon} = \bigcup_{i=1}^{n} \left\{z \in \Theta \mid d\left(x, y_i\right) \leq \epsilon\right\}.$$

Now, denote these $H - n$ remaining cubes by the set $\{C_r\}_{r=1}^{H-n}$, this means that for each C_r there exists a set $A_r \subseteq C_r$ such that $A_r \bigcap \bar{S}_{t_n,\varepsilon} = \emptyset$. Define a new chain $\left(\tilde{\theta}_s\right)_{s\geq 0}$ evolving exactly as the original chain (θ_t) from time t_n, but without further adaptations, i.e. it evolves using grid S_{t_n} and starting point

$\tilde{\theta}_0 = \theta_{t_n} = x$; this new chain is Markovian since the chain $(\theta_t)_t$ is Markovian. Define the first time such Markov chain hits the set $\bar{A}_{t_n} := \bigcup_{r=1}^{H-n} A_r$ as follows

$$\tau_{n+1} := \inf \left\{ s \geq 0 \mid \tilde{\theta}_s \in \bar{A}_{t_n} \right\}.$$

By construction $\tau_{n+1} \geq 1$ a.s. and since eachA_r has a positive Lebesgue measure, assuming the Markov chain is Harris recurrent

$$\mathbb{P}\left[\tau_{n+1} < \infty | \tilde{\theta}_0 = x, S_{t_n}\right] = 1.$$

Finally, let $t_{n+1} := t_n + \tau_{n+1}$ and notice that $\theta_{t_n+s} = \tilde{\theta}_s$ for all $s \in \{0, \ldots, \tau_{n+1}\}$. This implies that $\theta_{t_{n+1}} \in \bar{A}_{t_n}$ for the first time and the state at that time (say y_{n+1}) is added to the new grid $S_{t_{n+1}} := S_{t_n} \bigcup \{y_{n+1}\}$. Therefore, at time t_{n+1} there are at most $H - n - 1$ squares that are not fully covered by $\bar{S}_{t_{n+1},\varepsilon}$. An induction argument completes the proof noting that $\tau \leq \tau_1 + \cdots + \tau_H < \infty$ a.s. □

Remark 1. The previous result also implies that $|S_\tau| \leq H = (2L/\varepsilon)^d$.

3.6 Convergence

Let P be the Markov kernel associated to the ideal MH algorithm, i.e. the exact algorithm accepting moves according to $\alpha(\theta, \vartheta)$ in (3.14), such kernel is given

$$P(\theta_0, \theta_1) := q(\theta_0, \theta_1)\, \alpha(\theta_0, \theta_1) + \delta_{\theta_0}(\theta_1) \left(1 - \int \alpha(\theta_0, \vartheta)\, q(\theta_0, d\vartheta)\right).$$

We will restrict to the case where Θ is compact for which, under mild conditions, the kernel P is uniformly ergodic Roberts and Rosenthal [2004]. Uniform ergodicity guarantees the existence of $C < \infty$, $\rho \in (0, 1)$ such that

$$\sup_{\theta \in \Theta} \|P^n(\theta, \cdot) - \pi\|_{TV} \leq C\rho^n.$$

Now, fixing τ, γ_τ, θ and whenever $n \geq \tau$ we have

$$\begin{aligned}\left\|\tilde{P}_{\gamma_{\tau,N}}^{n-\tau}(\theta,\cdot)-\pi\right\|_{TV} &\leq \left\|\tilde{P}_{\gamma_{\tau,N}}^{n-\tau}(\theta,\cdot)-P^{n-\tau}(\theta,\cdot)\right\|_{TV}+\left\|P^{n-\tau}(\theta,\cdot)-\pi\right\|_{TV}\\ &\leq (n-\tau)\left\|\tilde{P}_{\gamma_{\tau,N}}(\theta,\cdot)-P(\theta,\cdot)\right\|_{TV}+C\rho^{n-\tau}. \qquad (3.22)\end{aligned}$$

Notice that the algorithm stops adapting at time τ, this which implies that any proposed move $\vartheta \sim q(\cdot \mid \theta)$ will lie within ε from some point in S_τ. Hence, if $n \geq \tau$ the kernel $\tilde{P}_{\gamma_{n,N}}$ as in 3.16 simplifies to

$$\begin{aligned}\tilde{P}_{\gamma_\tau,N}(\theta_n,\theta_{n+1}) &= q(\theta_{n+1}\mid\theta_n)\,\tilde{\alpha}^{(2)}_{S_\tau,N}(\theta_n,\theta_{n+1})\\ &\quad+\delta_{\theta_n}(\theta_{n+1})\int_\Theta\left(1-\tilde{\alpha}^{(2)}_{S_\tau,N}(\theta_n,\vartheta)\right)q(\vartheta\mid\theta_n)\,d\vartheta.\end{aligned}$$

We will also need a few mild assumptions and lemmas in order to prove the theorem below. The main result can be now stated: We show that the limiting distribution for the noisy adaptive chain approaches the desired target π, in terms of the total variation distance, and assuming that the chain is run long enough and that N is large.

Assumption 1. *The unnormalised likelihood g_θ is continuous and differentiable for $\theta \in \Theta$ and satisfies:*

1. $\sup_{\vartheta,\theta\in\Theta}\sup_{u\in\mathcal{U}}\left|\log\left(\frac{g_\theta}{g_\vartheta}(u)\right)\right| < \infty$, which implies there exists $K < \infty$ such that

$$\sup_{\vartheta,\theta}\sup_u\frac{g_\theta}{g_\vartheta}(u)\leq\exp(K) \quad \text{and} \quad \inf_{\vartheta,\theta}\inf_u\frac{g_\theta}{g_\vartheta}(u)\geq\exp(-K);$$

2. $\sup_{\theta\in\Theta}\sup_{u\in\mathcal{U}}\left|\frac{\partial}{\partial\theta}g_\theta(u)\right| < \infty$.

Theorem 3. *Suppose Assumption 1 holds and assume Θ is compact. For any starting point $\theta_0 \in \Theta$ and any $\delta > 0$, there exists $n(\delta,\theta_0)$ sufficiently large such that*

$$\lim_{N\to\infty}\left\|\mathbb{P}[\theta_n\in\cdot\mid\theta_0]-\pi\right\|_{TV}\leq\delta.$$

The first lemma shows how one can bound the total variation distance between the noisy and exact kernels in terms of their acceptance probabilities.

Lemma 1. *For fixed τ, N and $\gamma_{\tau,N} = \{S_\tau, V_{S_\tau,N}\}$ the kernels $\tilde{P}_{\gamma_{\tau,N}}$ and P satisfy for any $\theta \in \Theta$*

$$\left\| \tilde{P}_{\gamma_{\tau,N}}(\theta,\cdot) - P(\theta,\cdot) \right\|_{TV} \leq 2 \sup_{\vartheta\in\Theta} \left| \tilde{\alpha}^{(2)}_{S_\tau,N}(\theta,\vartheta) - \alpha(\theta,\vartheta) \right|. \tag{3.23}$$

Lemma 1 proof:

Proof. Recalling that $\|\mu\|_{TV} = \frac{1}{2}\sup_{f\in B_1} |\mu(f)| = \sup_{A\in\mathcal{B}(\Theta)} |\mu(A)|$, we have

$$\begin{aligned}
\left| \tilde{P}_{\gamma_{\tau,N}}(\theta,A) - P(\theta,A) \right| &\leq \left| \int_A \left(\tilde{\alpha}^{(2)}_{S_\tau,N}(\theta,\vartheta) - \alpha(\theta,\vartheta) \right) q(\theta,d\vartheta) \right| \\
&\quad + \left| \int_\Theta \left(\tilde{\alpha}^{(2)}_{S_\tau,N}(\theta,\vartheta) - \alpha(\theta,\vartheta) \right) q(\theta,d\vartheta) \right| \\
&\leq 2 \int_\Theta \left| \tilde{\alpha}^{(2)}_{S_\tau,N}(\theta,\vartheta) - \alpha(\theta,\vartheta) \right| q(\theta,d\vartheta) \\
&\leq 2 \sup_{\vartheta\in\Theta} \left| \tilde{\alpha}^{(2)}_{S_\tau,N}(\theta,\vartheta) - \alpha(\theta,\vartheta) \right| \int_\Theta q(\theta,d\vartheta).
\end{aligned}$$

Taking the supremum over $A \in \mathcal{B}(\Theta)$ on both sides finishes the proof. □

Remark 2. The supremum in the previous lemma is random, which could be problematic in terms of measurability. However, since Θ is some subset of $\mathbb{R}^d$ the resulting supremum is the same as the supremum over a countable set on $\mathbb{Q}^d$.

We now show that the acceptance probabilities $\tilde{\alpha}^{(2)}_{S,N}$ and α can be bounded in terms of the estimators $\widehat{R}_N$.

Lemma 2. *For fixed N and $\gamma_N = \{S, V_{S,N}\}$ the acceptance probabilities $\tilde{\alpha}^{(2)}_{S,N}$ and α satisfy*

$$\left| \tilde{\alpha}^{(2)}_{S,N}(\theta,\vartheta) - \alpha(\theta,\vartheta) \right| \leq (|S|+1) \sup_{\theta\in\Theta,\vartheta\in S} \left| \log\left(\frac{\widehat{R}_N(\theta,\vartheta)}{R(\theta,\vartheta)} \right) \right|.$$

Lemma 2 proof:

Proof. Notice that the function $x \longmapsto \min\{1, \exp(x)\}$ is Lipschitz with coefficient 1. Therefore

$$\left|\tilde{\alpha}^{(2)}_{S,N}(\theta,\vartheta) - \alpha(\theta,\vartheta)\right| \leq \left|\log\left(\widehat{R}^{(2)}_{S,N}(\theta,\vartheta)\right) - \log\left(R(\theta,\vartheta)\right)\right|,$$

and by repetitively applying the triangle inequality

$$\begin{aligned}
\left|\log\left(\widehat{R}^{(2)}_{S,N}(\theta,\vartheta)\right) - \log\left(R(\theta,\vartheta)\right)\right| &= \left|\log\left(\frac{\widehat{R}_{\mathcal{P}_{\theta,\vartheta},V_{S,N}}\left(\theta,\bar{\theta}_{m_{\theta,\vartheta}}\right)}{\widehat{R}_N\left(\vartheta,\bar{\theta}_{m_{\theta,\vartheta}}\right)}\right) - \log\left(\frac{R\left(\theta,\bar{\theta}_{m_{\theta,\vartheta}}\right)}{R\left(\vartheta,\bar{\theta}_{m_{\theta,\vartheta}}\right)}\right)\right| \\
&\leq \left|\log\left(\widehat{R}_{\mathcal{P}_{\theta,\vartheta},V_{S,N}}\left(\theta,\bar{\theta}_{m_{\theta,\vartheta}}\right)\right) - \log\left(R\left(\theta,\bar{\theta}_{m_{\theta,\vartheta}}\right)\right)\right| \\
&\quad + \left|\log\left(\widehat{R}_N\left(\vartheta,\bar{\theta}_{m_{\theta,\vartheta}}\right)\right) - \log\left(R\left(\vartheta,\bar{\theta}_{m_{\theta,\vartheta}}\right)\right)\right| \\
&\leq \left|\log\left(\widehat{R}_N\left(\theta,\bar{\theta}_1\right)\right) - \log\left(R\left(\theta,\bar{\theta}_1\right)\right)\right| \\
&\quad + \sum_{i=2}^{m_{\theta,\vartheta}} \left|\log\left(\widehat{R}_N\left(\bar{\theta}_{i-1},\bar{\theta}_i\right)\right) - \log\left(R\left(\bar{\theta}_{i-1},\bar{\theta}_i\right)\right)\right| \\
&\quad + \left|\log\left(\widehat{R}_N\left(\vartheta,\bar{\theta}_{m_{\theta,\vartheta}}\right)\right) - \log\left(R\left(\vartheta,\bar{\theta}_{m_{\theta,\vartheta}}\right)\right)\right|.
\end{aligned}$$

Since $\bar{\theta}_i \in S$ for each $i \in \{1,\ldots,m_{\theta,\vartheta}\}$ each term in the middle sum can be bounded by

$$\left|\log\left(\widehat{R}_N\left(\bar{\theta}_{i-1},\bar{\theta}_i\right)\right) - \log\left(R\left(\bar{\theta}_{i-1},\bar{\theta}_i\right)\right)\right| \leq \sup_{\theta,\vartheta\in S}\left|\log\left(\frac{\widehat{R}_{U^{1:N}_{\vartheta}}(\theta,\vartheta)}{R(\theta,\vartheta)}\right)\right|,$$

whereas the two remaining terms are each bounded by $\sup_{\theta\in\Theta,\vartheta\in S}\left|\log\left(\frac{\widehat{R}_N(\theta,\vartheta)}{R(\theta,\vartheta)}\right)\right|$. Therefore

$$\begin{aligned}
\left|\tilde{\alpha}^{(2)}_{S,N}(\theta,\vartheta) - \alpha(\theta,\vartheta)\right| &\leq 2\sup_{\theta\in\Theta,\vartheta\in S}\left|\log\left(\frac{\widehat{R}_N(\theta,\vartheta)}{R(\theta,\vartheta)}\right)\right| + (m_{\theta,\vartheta,S}-1)\sup_{\theta,\vartheta\in S}\left|\log\left(\frac{\widehat{R}_N(\theta,\vartheta)}{R(\theta,\vartheta)}\right)\right| \\
&\leq (m_{\theta,\vartheta}+1)\sup_{\theta\in\Theta,\vartheta\in S}\left|\log\left(\frac{\widehat{R}_N(\theta,\vartheta)}{R(\theta,\vartheta)}\right)\right|.
\end{aligned}$$

Finally, recall that the path $\mathcal{P}_{\theta,\vartheta} \subseteq S$ which implies $m_{\theta,\vartheta} \leq |S|$, the result then follows. □

Assumption 1 allows us to show a uniform convergence result for $\widehat{R}_N(\theta, \vartheta)$ towards the intractable $R(\theta, \vartheta)$ as $N \to \infty$.

Lemma 3. *Under Assumption 1, if Θ is compact and for fixed S*

$$\sup_{\theta \in \Theta, \vartheta \in S} \left| \widehat{R}_{U_\vartheta^{1:N}}(\theta, \vartheta) - R(\theta, \vartheta) \right| \xrightarrow{p} 0, \qquad \text{as } N \to \infty. \tag{3.24}$$

This in turn implies

$$\sup_{\theta \in \Theta, \vartheta \in S} \left| \log \left(\frac{\widehat{R}_N(\theta, \vartheta)}{R(\theta, \vartheta)} \right) \right| \xrightarrow{p} 0, \qquad \text{as } N \to \infty. \tag{3.25}$$

Lemma 3 proof:

Proof. Let

$$\begin{aligned} G_{\vartheta,N}(\theta) \equiv G_{\vartheta, U_\vartheta^{1:N}}(\theta) &:= \widehat{R}_{U_\vartheta^{1:N}}(\theta, \vartheta) - R(\theta, \vartheta) \\ &=: \widehat{Q}_{\vartheta,N}(\theta) - Q_\vartheta(\theta) \end{aligned}$$

we first show that for any $\vartheta \in S$

$$\sup_{\theta \in \Theta} |G_{\vartheta,N}| \xrightarrow{p} 0, \qquad \text{as } N \to \infty.$$

To show this type of uniform convergence we follow Andrews [1992] (Theorem 1) for obtaining the result we need:

1. Θ to be bounded, which is assumed throughout;
2. $G_{\vartheta,N}(\theta) \xrightarrow{p} 0$ as $N \to \infty$ for all $\theta \in \Theta$, which follows directly from the Weak Law of Large Numbers;
3. $\{G_{\vartheta,N}(\theta)\}_{N \geq 1}$ to be stochastically equicontinuous on Θ.

This last condition is easily shown using Andrews [1992](Lemma 1) by proving $\theta \longmapsto \widehat{Q}_{\vartheta,N}(\theta)$ is Lipschitz on Θ with bounded random Lipschitz coefficient.

Indeed, $\widehat{Q}_{\vartheta,N}(\theta)$ is continuous and differentiable on θ from Assumption 1, implying there exists a constant $K < \infty$ such that for any N, and pair $(\theta, \vartheta) \in \Theta^2$ and any $U_\vartheta^{1:N} \in \mathcal{U}^N$

$$\begin{aligned}\left|\frac{\partial}{\partial\theta}\widehat{Q}_{\vartheta,N}(\theta)\right| &\leq \frac{1}{N}\sum_{i=1}^{N}\left|\frac{\partial}{\partial\theta}g_\theta\left(U_\vartheta^{(i)}\right)\right| \\ &< K.\end{aligned}$$

Finally,

$$\begin{aligned}\mathbb{P}\left[\sup_{\theta\in\Theta,\vartheta\in S}\left|\widehat{R}_{U_\vartheta^{1:N}}(\theta,\vartheta) - R(\theta,\vartheta)\right| > \delta \mid S\right] &= \mathbb{P}\left[\bigcup_{\vartheta\in S}\left\{\sup_{\theta\in\Theta}\left|\widehat{R}_{U_\vartheta^{1:N}}(\theta,\vartheta) - R(\theta,\vartheta)\right| > \delta\right\} \mid S\right] \\ &\leq \sum_{\vartheta\in S}\mathbb{P}\left[\sup_{\theta\in\Theta}\left|\widehat{R}_{U_\vartheta^{1:N}}(\theta,\vartheta) - R(\theta,\vartheta)\right| > \delta \mid S\right],\end{aligned}$$

which leads to the first convergence in 3.24 since S is a finite set.

The second convergence 3.25 is easily shown using the first claim and the fact that $|\log(x)| \leq x^{-1/2}|x-1|$ for any $x > 0$. □

Remark 3. The first inequality in 3.19 is true since for a signed measure ν_x, probability distribution μ and $f \in B_1 := \{g : \Theta \to \mathbb{R} \mid |g| \leq 1\}$

$$\begin{aligned}\left|\mathbb{E}_{X\sim\mu}\nu_X(f)\right| &\leq \mathbb{E}\left|\nu_X(f)\right| \\ &\leq \mathbb{E}\left(\sup_{f\in B_1}\left|\nu_X(f)\right|\right) \\ &= 2\mathbb{E}\left\|\nu_X\right\|_{TV}.\end{aligned}$$

Therefore $\left\|\mathbb{E}\nu_X\right\|_{TV} = \frac{1}{2}\sup_{f\in B_1}\left|\mathbb{E}\left(\nu_X(f)\right)\right| \leq \mathbb{E}\left\|\nu_X\right\|_{TV}$.

Remark 4. The previous result also implies that $|S_\tau| \leq H = (2L/\varepsilon)^d$.

We are now able to show that the limiting behaviour of the noisy chain approaches to the desired target π, provided that it stops adapting after some fixed time τ.

Proposition 2. *Under Assumption 1 and for any $\delta > 0$ and fixed τ, S_τ there exists $n_0(\tau, \delta) > \tau$ such that for any $n \geq n_0$*

$$\lim_{N\to\infty} \mathbb{P}\left[\sup_{\theta\in\Theta} \left\| \tilde{P}^{n-\tau}_{\gamma_{\tau,N}}(\theta_\tau,\cdot) - \pi \right\|_{TV} > \delta \mid \tau, S_\tau\right] = 0.$$

Proposition 2 proof:

Proof. Using the previous lemmas and Remark 4 we have for any $n > \tau$

$$\begin{aligned}\sup_{\theta\in\Theta} \left\| \tilde{P}^{n-\tau}_{\gamma_{\tau,N}}(\theta,\cdot) - \pi \right\|_{TV} &\leq 2(n-\tau)(|S_\tau|+1) \sup_{\theta\in\Theta,\vartheta\in S_\tau} \left|\log\left(\frac{\widehat{R}_N(\theta,\vartheta)}{R(\theta,\vartheta)}\right)\right| + C\rho^{n-\tau} \\ &\leq 2n\left(\left(\frac{2L}{\varepsilon}\right)^d + 1\right) \sup_{\theta\in\Theta,\vartheta\in S_\tau} \left|\log\left(\frac{\widehat{R}_N(\theta,\vartheta)}{R(\theta,\vartheta)}\right)\right| + C\rho^{n-\tau}.\end{aligned}$$

Take $n_0 := \tau + \left\lceil \frac{\log(2C)-\log(\delta)}{\log(\rho^{-1})} \right\rceil$. Then $C\rho^{n-\tau} < \delta/2$ if $n \geq n_0$, and

$$\begin{aligned}&\mathbb{P}\left[\sup_{\theta\in\Theta} \left\| \tilde{P}^{n-\tau}_{\gamma_{\tau,N}}(\theta_\tau,\cdot) - \pi \right\|_{TV} > \delta \mid \tau, S_\tau\right] \\ &\leq \mathbb{P}\left[2n\left(\left(\frac{2L}{\varepsilon}\right)^d + 1\right) \sup_{\theta\in\Theta,\vartheta\in S_\tau} \left|\log\left(\frac{\widehat{R}_N(\theta,\vartheta)}{R(\theta,\vartheta)}\right)\right| + \frac{\delta}{2} > \delta \mid \tau, S_\tau\right] \\ &= \mathbb{P}\left[\sup_{\theta\in\Theta,\vartheta\in S_\tau} \left|\log\left(\frac{\widehat{R}_N(\theta,\vartheta)}{R(\theta,\vartheta)}\right)\right| > \frac{\delta}{4n\left(\left(\frac{2L}{\varepsilon}\right)^d + 1\right)} \mid \tau, S_\tau\right].\end{aligned}$$

Using Lemma 3 the result is obtained. □

Remark 5. Notice that the previous result relies on the fact that τ and S_τ are fixed. However, we have not been entirely explicit about the dependence of these variables with the index N. This means that the distribution of τ and S_τ is likely to be affected for different values of N. Nevertheless, the proposition will ensure convergence to the true target provided that an initial and fixed value for N is used up to the time τ. After that, for fixed S_τ one can increase the number of auxiliary variables used for computing the ratios $\widehat{R}_N$ in order to guarantee convergence.

We now address the case in which the distributions of τ and S_τ change

as N increases. The following technical result ensures convergence in mean of the term $\sup_{\theta\in\Theta,\vartheta\in S}\left|\log\left(\frac{\widehat{R}_N(\theta,\vartheta)}{R(\theta,\vartheta)}\right)\right|$ using the uniform convergence from the previous lemma.

Lemma 4. *Let $Z_N(S) := \sup_{\theta\in\Theta,\vartheta\in S}\left|\log\left(\frac{\widehat{R}_N(\theta,\vartheta)}{R(\theta,\vartheta)}\right)\right|$. Then, under Assumption 1 and if Θ is compact, the following holds for any finite set $S\in\mathcal{P}(\Theta)$ for some*

$$\lim_{N\to\infty}\sup_{S\in\mathcal{P}(\Theta)}\mathbb{E}\left[Z_N(S)\right]=0.$$

Proof. Let $Y_N(\vartheta) := \sup_{\theta\in\Theta}\left|\widehat{R}_N(\theta,\vartheta)-R(\theta,\vartheta)\right|$, using the fact that $|\log(x)|\leq x^{-1/2}|x-1|$ for any $x>0$, we have for some $C>0$

$$\begin{aligned}Z_N(S) &\leq \frac{\sup_{\theta\in\Theta,\vartheta\in S}\left|\widehat{R}_N(\theta,\vartheta)-R(\theta,\vartheta)\right|}{\inf_{\theta,\vartheta\in\Theta}\sqrt{R(\theta,\vartheta)\,\widehat{R}_N(\theta,\vartheta)}}\\ &\leq C\sup_{\vartheta\in S}Y_N(\vartheta).\end{aligned}$$

Hence,

$$\begin{aligned}\mathbb{E}\left[Z_N(S)\right] &\leq C\mathbb{E}\sum_{\vartheta\in S}Y_N(\vartheta)\\ &\leq C\mathbb{E}\left[\sum_{\vartheta\in S}\mathbb{E}\left[Y_N(\vartheta)\mid S\right]\right]\\ &\leq C\mathbb{E}\left[|S|\sup_{\vartheta\in\Theta}\mathbb{E}\left[Y_N(\vartheta)\right]\right]\\ &\leq C\left(\frac{2L}{\varepsilon}\right)^d\sup_{\vartheta\in\Theta}\mathbb{E}\left[Y_N(\vartheta)\right],\end{aligned}$$

leading to $\sup_S\mathbb{E}\left[Z_N(S)\right]\leq C\sup_{\vartheta\in\Theta}\mathbb{E}\left[Y_N(\vartheta)\right]$ for some $C>0$.

What is left to show is $\lim_{N\to\infty}\sup_{\vartheta\in\Theta}\mathbb{E}\left[Y_N(\vartheta)\right]=0$. This is easily shown by Dini's theorem since the function $f_N(\vartheta)=\mathbb{E}\left[Y_N(\vartheta)\right]$ is defined on the compact space Θ, is continuous, converges pointwise to zero, and satisfies

$$f_{N+1}\leq f_N.$$

This last claim is shown by noting that for a collection of i.i.d. random variables $\left\{X^{(i)}\right\}_{i\geq 1}$ with mean zero, the sums $\left\{\Sigma_N := \sum_{i=1}^N X^{(i)}\right\}_{N\geq 1}$ and partial sums $\left\{\Sigma_N^{(-k)} := \sum_{i=1}^N X^{(i)} - X^{(k)}\right\}_{N\geq 1, k\leq N}$ satisfy

$$\begin{aligned}\frac{1}{N+1}\mathbb{E}\left|\Sigma_{N+1}\right| &= \mathbb{E}\left[\frac{1}{N+1}\left|\frac{\Sigma_N^{(-(N-1))} + \Sigma_N^{(-N)} + ... + \Sigma_N^{(-1)}}{N}\right|\right] \\ &\leq \frac{1}{N+1}\left(\mathbb{E}\left|\frac{\Sigma_N^{(-(N-1))}}{N}\right| + ... + \mathbb{E}\left|\frac{\Sigma_N^{(-1)}}{N}\right|\right) \\ &= \frac{1}{N}\mathbb{E}\left|\Sigma_N\right|.\end{aligned}$$

□

The final intermediate result concerns the convergence in probability of the stopping time τ as $N \to \infty$. As shown, this variable converges to the corresponding stopping time τ^* that arises from running the ideal MH chain, but keeping track of the proposed states (as in the adaptive algorithm) in order to generate a similar grid of points $S^*_{\tau^*}$.

Lemma 5. *Let* $\left(X_n^{(N)} := \left(\theta_n^{(N)}, S_n^{(N)}, V_{S_n,N}^{(N)}\right)\right)_{n\geq 0}$ *and* $\left(X_n^* = \left(\theta_n^*, S_n^*, V_{S_n^*,N}^*\right)\right)_{n\geq 0}$ *denote the approximate and ideal Markov chains, respectively. Denote also, respectively by* $\tau^{(N)}$ *and* τ^* *the corresponding times at which the chains have created a finite cover of* Θ*. For any starting point* x_0 *such that* $X_0^{(N)} = X_0^* = x_0$*, as* $N \to \infty$

$$\tau\left(N\right) \xrightarrow{p} \tau^*.$$

Lemma 5 proof:

Proof. We have that

$$\begin{aligned}
\mathbb{P}\left[\tau^{(N)} \neq \tau^* \mid X_0^{(N)} = X_0^* = x_0\right] &\leq \mathbb{P}\left[X_{\tau^*}^{(N)} \neq X_{\tau^*}^* \mid X_0^{(N)} = X_0^* = x_0\right] \\
&\leq \mathbb{P}\left[X_t^{(N)} \neq X_t^*, \tau^* \leq t \mid X_0^{(N)} = X_0^* = x_0\right] + \mathbb{P}\left[X_{\tau^*}^{(N)} \neq X_{\tau^*}^*, \tau^* > t \mid X_0^{(N)} = X_0^* = x_0\right] \\
&\leq \sum_{n=0}^{t-1} \mathbb{P}\left[X_{n+1}^{(N)} \neq X_{n+1}^*, X_n^{(N)} = X_n^* \mid X_0^{(N)} = X_0^* = x_0\right] + \mathbb{P}\left[\tau^* > t \mid X_0^{(N)} = X_0^* = x_0\right] \\
&\leq \sum_{n=0}^{t-1} \mathbb{P}\left[X_{n+1}^{(N)} \neq X_{n+1}^* \mid X_n^{(N)} = X_n^*, X_0^{(N)} = X_0^* = x_0\right] + \mathbb{P}\left[\tau^* > t \mid X_0^{(N)} = X_0^* = x_0\right] \\
&\leq \sum_{n=0}^{t-1} \int \mathbb{P}\left[X_{n+1}^{(N)} \neq X_{n+1}^* \mid X_n^{(N)} = X_n^* = x_n\right] P^n\left(dx_n \mid x_0\right) + \mathbb{P}\left[\tau^* > t \mid X_0^{(N)} = X_0^* = x_0\right].
\end{aligned}$$

For any choice of x_n, we have that $\mathbb{P}\left[X_{n+1}^{(N)} \neq X_{n+1} \mid X_n^{(N)} = X_n^* = x_n\right] \to 0$ as $N \to \infty$. Using the bounded Convergence Theorem and the fact that $\tau^* < \infty$ a.s. for any starting point we conclude that $\tau^{(N)} \xrightarrow{p} \tau^*$ as $N \to \infty$. □

We are now able to prove the main result.

Proof of Theorem 3. Let $d_{TV}\left(n; \tau, \gamma_{\tau,N}\right) = \sup_{\theta \in \Theta} \left\| \tilde{P}_{\gamma_{\tau,N}}^{n-\tau}\left(\theta, \cdot\right) - \pi \right\|_{TV}$. Consider $n \geq \tau$, for some $C > 0$ we have that

$$\begin{aligned}
\mathbb{E}\left[d_{TV}\left(n; \tau, \gamma_{\tau,N}\right)\right] &= \mathbb{E}\left[d_{TV}\left(n; \tau, \gamma_{\tau,N}\right) \mathbb{1}\left(\tau \leq \tau^*\right)\right] + \mathbb{P}\left[d_{TV}\left(n; \tau, \gamma_{\tau,N}\right) \mathbb{1}\left(\tau > \tau^*\right)\right] \\
&\leq Cn\mathbb{E}\left[\sup_{\theta \in \Theta, \vartheta \in S_\tau} \left| \log\left(\frac{\widehat{R}_N\left(\theta, \vartheta\right)}{R\left(\theta, \vartheta\right)}\right)\right|\right] + C\rho^n \mathbb{E}\left[\rho^{-\tau^*}\right] + \mathbb{P}\left[\mathbb{1}\left(\tau > \tau^*\right)\right],
\end{aligned}$$

where the last inequality follows from (3.22), Lemmas 1 and 2, and the fact that $\mathbb{E}\left[\rho^{-\tau(N)}\right] \to \mathbb{E}\left[\rho^{-\tau^*}\right]$ since $\tau\left(N\right) \xrightarrow{p} \tau^*$ and $\mathbb{E}\rho^{-\tau(N)} < \infty$ for any $N \in \mathbb{N}$

Now, let $n_0\left(\delta\right) = \frac{\log(\delta^{-1}) + \log\left(C\mathbb{E}\left[\rho^{-\tau^*}\right]\right)}{\log(\rho^{-1})}$, then for any $n \geq \max\left\{n_0, \tau^*\left(\theta_0\right)\right\}$

$$\mathbb{E}\left[d_{TV}\left(n; \tau, \gamma_{\tau,N}\right)\right] \leq Cn \sup_{S \in \mathcal{P}(\Theta)} \mathbb{E}\left[\sup_{\theta \in \Theta, \vartheta \in S} \left| \log\left(\frac{\widehat{R}_N\left(\theta, \vartheta\right)}{R\left(\theta, \vartheta\right)}\right)\right|\right] + \delta + \mathbb{P}\left[\mathbb{1}\left(\tau > \tau^*\right)\right].$$

Using Lemma 4, taking $N \to \infty$ guarantees that, for fixed $n \geq \max\left\{n_0, \tau^*\right\}$, the first and third terms on the right hand side of the above inequality go to

zero. Therefore, since δ was picked arbitrarily, for any $n \geq \max\{n_0, \tau^*\}$ we have that

$$\lim_{N\to\infty} \mathbb{E}\left[d_{TV}\left(n; \tau, \gamma_{\tau,N}\right)\right] = 0.$$

Finally, the result is obtained noting that

$$\begin{aligned}\left\|\mathbb{P}\left[\theta_n \in \cdot \mid \theta_0\right] - \pi\right\|_{TV} &\leq \mathbb{E}\left[\left\|\tilde{P}^{n-\tau}_{\gamma_{\tau,N}}\left(\theta_\tau, \cdot\right) - \pi\right\|_{TV}\right] \\ &\leq \mathbb{E}\left[d_{TV}\left(n; \tau, \gamma_{\tau,N}\right)\right].\end{aligned}$$

□

3.7 Conclusions

In this chapter we have proved the convergence of adaptive noisy exchange algorithm. The main assumption that one could reasonably expect to generalise is that of the compact space considered here. The main reason for the compactness is the space coverage by n-balls. An unbounded space, will result in a possibly non-finite time in order to cover, depending on the stochastic process being defined and the integral of the r^d being infinite, since otherwise the space is not almost surely covered (for $\mathbb{R}^d$) Biermé and Estrade [2012] . An additional issue of importance here is the effect of the size of n-balls of radius ϵ. For all intents and purposes while theoretical guarantees that the algorithm will stop adapting in finite time, thus converging to the desired target, practically speaking the adaptation will be ongoing for the run of the algorithm even for a very large number of iterations. In simple inference problems with low dimensionality of the parameter space and large values of ϵ the space will have been covered by regions where all the proposals within few iterations will almost always fall inside and hence have already the precomputed points calculated due to the grid, and therefore no further adaptation might take place in the regions of high posterior concentration. On unimodal low dimensional

targets that will most likely be the case. On the other hand, high dimensional, and multimodal targets (not necessarily at the same time) will leave numerous areas uncovered by the union of balls of radius ϵ, thus leaving regions where the algorithm still needs to recalculate the newly proposed points and simulate from the likelihood, and therefore making the probability that the algorithm will propose on those points increase as the dimensionality grows. Despite the finite time occurrence of the stopping of adaptation the reality is that as $d \to \infty$ the number of iterations for that stopping time will increase exponentially since the scaling of the cubes to cover the space as in Proposition 1 is itself exponential in dimension) thus making the algorithm always be in the adaptation phase for all practical applications on high dimensional problems. It is interesting to note that we obviously have not provided rates of convergence of the algorithm or a kind of Law of Large numbers (usually weak forms), with the former perhaps not being exactly feasible.

Chapter 4

Rare event ABC-SMC^2 algorithm

4.1 Introduction

In chapter **1** of this thesis we highlighted a number of issues one encounters when dealing with intractable likelihoods, in whichever form they appear and with regards to how that intractability emerged. In this chapter we will be dealing with intractability pertaining to the impossibility or impracticality of pointwise likelihood evaluation (numerically), yet for which direct simulation from the model of interest is possible (given various parameters). We are therefore in the setting of approximate Bayesian computation.

More specifically we are interested in tackling one of the issues that arises in ABC due to the various layers of approximation one resorts to when using these types of algorithms. It is perhaps instructive to reiterate here that in the ABC setting we are dealing not only with the regular Monte Carlo error, but also with the error due to the non-zero tolerance or bandwidth of the kernel between real and simulated data, as well as with the error due to the summary statistics (the latter of which can have in certain cases really drastic effects on the inference Robert et al. [2011]) one uses -dependent on the task

at hand- due to the information loss incurred by this approximation). In this work of particular interest is the issue of summary statistics and specifically the exponential dimensionality scaling with regards to how many samples one needs to maintain a certain level of error in the approximation, and a potential amelioration of that issue (or perhaps even a complete avoidance of the usage of such summary statistics, when possible). In ABC we are dealing with two distinct dimensionality issues: First, one in which the parameter space is of some fixed dimensionality, say d, for parameters $\theta \in \mathcal{G} \subseteq \mathcal{R}^d$ in the continuous case for simple RV, and for which our arbitrary model, say $\mathcal{M}$, takes as input, i.e. $\mathcal{M}(f(\theta), x, ...)$. For example, it can be some likelihood $\mathcal{M} = f(x|\theta)$ in the Bayesian setting or if we are using some complex simulator, $\mathcal{M} = \boxed{\square}$, for some black box $\boxed{\square}$ process, which can be a climate model, a genetic process/model, a process expressed as some system of stochastic partial differential equations etc. Secondly, the state space dimension $x \in \mathcal{X}^k$ for some different dimension k, that is generated and labeled as simulated data and is required to be compared under some kernel $\mathcal{K}$ of choice to the real observation/data (of the same dimension). Our contributions are aimed towards the second issue; i.e. that of the state space dimensionality and its effects on the final inference with respect to the variance of the estimators and the accuracy given specific tolerance parameters in the ABC context.

Given what we have described so far and taking into account the general ABC setting, it would be very instructive to view the method as a nearest neighbours algorithm Cucala et al. [2009], Biau et al. [2015] , with the initially motivation and nonparametric flavor in Loftsgaarden and Quesenberry [1965], Cover [1968], Fix and Hodges [1989] . It is well understood that that approach becomes less effective as the dimension grows, commonly (perhaps overly so) referred to as "curse of dimensionality". It is easy to see why: for some high dimensional output from the model, we require for a large number of random components to be matched to an arbitrary precision for some distance metric/kernel to those components of the real data. It is evident that the probability of that happening as the number of components increases, decreases rapidly and at various rates. It is this very issue we are trying to address here. One of the first steps in this direction, given the aforementioned setting, were made in Prangle et al. [2018]. The authors suggest using auxiliary variables u in the pseudo-marginal spirit to estimate the intractable likelihood term

Andrieu et al. . The explicit assumption is made that the model we will be dealing with can be written as some form of deterministic transformation (although not necessarily deterministic, see below) of these variables u and the parameters of interest $H(u, \theta)$. It is easy to see that this then forces the u variables (since the vector of θs are fixed for each transformation) to encode all the randomness in the simulation of the model/process. This is of course an old idea that has appeared in various places over the years (for one instance see the discussion by Andrieu et al in Fearnhead and Prangle [2012]). The approach of Prangle et al. [2018] is then to for fixed θ values to use rare event methods to estimates probabilities of the form:

$$\mathbb{P}(u \in A; ||H(u, \theta) - y_{obs}||_{\mathcal{D}} \leq \epsilon|\theta) \tag{4.1}$$

We have seen in the ABC introductory section (section 1.3.11) that this is in fact the approximate likelihood of θ used in the ABC methods introduced previously. The authors proposed the usage of the Rare Event SMC algorithm by Cérou et al. [2012] for the estimation of these probabilities. It is argued that the resulting estimates (as shown in the paper) are unbiased or low bias (in the adaptive version of the algorithm) and can thus be used by various other inference methods. One of which and the main framework for this construction is the pseudo-marginal methods.

Before we return to those methods in more detail let us give a more detailed account of why one would want to use these rare event probabilities in the framework discussed in these works to estimate the likelihood (or ratio of likelihood in our case) of interest. Intuitively, thinking about the probability above one sees that we are potentially and in most practical cases almost always, dealing with the calculation of very small probabilities, hence the name rare events. It is highly unlikely that we will draw u (given some vector θ such that they generate simulated data that fall within some metric distance close to those of the real data). The relative error of the estimate of $\Pr(y(\theta, x) \approx y_{\text{obs}})$ has high variance when these probabilities are small since we require that the simulated data are arbitrarily close to the real ones. The rare event idea is to split this very small probability event into many events of significantly higher

probability. This can be done by considering nested sets of events for which the probability of falling within the next nested (smaller) set given the next largest one is fairly high. In detail consider splitting our event into sets of latent variables $A_1 \supset A_2 \supset A_3 \supset .. \supset A_t$ representing a cascade of higher probabilities of hitting the innermost set. We then need to estimate probabilities of the form $\Pr(A_1), \Pr(A_2 \mid A_1), \Pr(A_3 \mid A_2), \ldots$ and since by construction this product is our original small event probability we are done. Finally, if the probabilities indicated above are indeed large (relative to each other), then the variance of the final estimator's relative error is smaller Prangle et al. [2018] than using a single state Monte Carlo (Cérou et al. [2012], L'Ecuyer et al. [2007]).

Now of course the fact that these events will have sufficiently higher probability is not a given, but rather our intention, since it would then make it much easier to simulate from since we will be trying to sample going from a larger set to a slightly smaller one $A_t \to A_{t-1}$ and not directly to the smallest one $A_1 \to A_n$ thus having a higher probability of hitting it). It is also easy to see that one can indeed increase the probability of these events by increasing the number of nested sets, although a balance is needed since we are always bound by computational resources and we cannot arbitrarily break down the problem into so many subparts that we end up with a higher average cost than the one we originally intended to reduce.

We can therefore estimate $P(A_1)$ using N Monte Carlo samples, (which essentially amounts to running the RE-SMC sampler of Cérou et al. [2012] as used in Prangle et al. [2018] and given in algorithm 12), then reuse the samples $u \in A_1$, by sampling randomly from that set and to avoid duplicates perturb them appropriately. The authors in Prangle et al. [2018] used for example a slice sampler to perform essentially inference on u space. Thus, the resulting MC sample is used to estimate $P(A_2|A_1)$ and so on for all the required conditional probabilities. The authors also note that for this approach to perform well the mappings H must of such nature such that small perturbations in the random variables u produce small perturbations in the output of the mappings $y = H(u, \theta)$. In essence we could argue from a dynamical systems point of view that two very close initial points in some d dimensional space, which would usually be the seed random variables that go into any kind of simulator would not produce some nonlinear behaviour when it comes to the output,

our state space of interest $\mathbf{y}$. In other words sufficiently close points in the u-random variable space produce sufficiently close points (in the metric that is appropriate for the application of interest) in the state space. Furthermore, we should passingly mention the issue of non-identifiability and in general a small comment on the vast literature of inverse problems:i.e. extremely similar outputs could well be produced by very dissimilar inputs. It is obvious that if that is the case of the model being considered, any method based on distance similarity approach in the ABC context will fail.

At this point it would be pertinent to address the issue of using the SMC estimator for these rare events Cérou et al. [2012], instead of just a simple importance sampling one. We can see (as for example in Prangle et al. [2018]) that in order to control the variance of an importance sampling estimator of the ABC likelihood we need a number of points that is exponential in the dimension dAgapiou et al. [2017]. Furthermore, note that when ϵ is small we need an order of $\mathcal{O}(\epsilon^d)$ points Prangle et al. [2018] to get an acceptance. It is nevertheless the case that if we use an SMC estimator instead of IS the required number of points needed to control the variance is quadratic instead of exponential in the number of dimensions $\mathcal{O}\left([\log\epsilon^{-d}]^2\right)$. (Beskos et al. [2014a], Agapiou et al. [2017]) .

Furthermore one should consider the fact that for a fixed number of Monte Carlo samples N, the choice of kernel scale parameter or bandwidth is representative of a typical variance-bias balancing act; assume we have a large bandwidth $h = \mathcal{K}(y_{obs} - y_{sim})$, which represents the desired metric distance between real and simulated data. We then draw a greater number of samples M from the ABC posterior π_{ABC}, thus reducing the variance but yet we have a poorer ABC approximation. On the other hand assuming we decide on a small h, the posterior approximation is improved but Monte Carlo variance is increased (as the samples within that bandwidth given fixed total N of samples are fewer). Furthermore assuming the statistics $s(y)$ are not sufficient[1], we are in fact approximating a different posterior than the one using the full data (which in real applications is most likely to be the case). We have thus reduced the variance, yet increased the bias. Similarly, in the extreme case

[1]Some statistic $\Gamma = f(y_1, y_2, \cdots, y_n)$ is sufficient if for each f, the conditional distribution of $y_1, y_2, \cdots, y_n$ given $F = f$ and θ does not depend on θ.

where $h = 0$ the ABC likelihood has resulted in the true posterior although the variance of this likelihood estimator is maximal. On the other hand for any $h > 0$ chosen in a way to control the variance of ABC MCMC estimates for example, we are again biasing the final estimate.

Given what we have described so far one naturally wonders what is the justification or benefit for the usage of auxiliary variables and the transformation from u-space to that of y-space. It would be beneficial to introduce the issue when one wants to approximate the ABC likelihood by using importance sampling. Consider for example a simple importance sampling algorithm where the target is the usual full ABC posterior: $\pi_{ABC}(\theta, s \mid s_{obs})$.

Since we are interested in sampling from θ-space we can propose points by using the proposal $g(\theta, s) = f(s|\theta)g(\theta)$. Explicitly in this proposal we have used the "true" ABC likelihood $p(s|\theta) = \int_x K_h(s, s_{obs}) f_\theta(s) dx$ which is by the very setting of ABC intractable as we showed in the introduction on chapter 1. Nonetheless consider the importance sampling estimator and the resulting (unnormalized) weights: $\mathcal{D}(s, s_{obs})$

$$\frac{\pi_{ABC}(\theta, s \mid s_{obs})}{g(\theta, s)} \propto \frac{K_h(s, s_{obs}) f(s \mid \theta)\pi(\theta)}{f(s \mid \theta)g(\theta)} = \frac{K_h(s, s_{obs})\pi(\theta)}{g(\theta)} := \tilde{w}(\theta) \quad (4.2)$$

with h the chosen target distance (which depends on the application domain and specific model).

We therefore see that the intractable term f has vanished. Similarly consider the implementation of ABC MCMC with the same target but proposal distribution defined as:

$$g[(\theta, s), (\theta', s')] = g(\theta, \theta') f(s' \mid \theta') \quad (4.3)$$

and the upper index $\theta^{(i)}$ denoting the current time state of the Markov Chain. The acceptance probability of the proposed move from $\left(\theta^{(i)}, s^{(i)}\right)$ to $(\theta', s') \sim g\left[\left(\theta^{(i)}, s^{(i)}\right), (\theta', s')\right]$ becomes $a\left[\left(\theta^{(i)}, s^{(i)}\right), (\theta', s')\right] = \min\left\{1, \alpha\left[\left(\theta^{(i)}, s^{(i)}\right), (\theta', s')\right]\right\}$,

where

$$\begin{aligned}\alpha\left[\left(\theta^{(i)},s^{(i)}\right),(\theta',s')\right] &= \frac{\pi_{ABC}\left(\theta',s'\mid s_{obs}\right)g\left[\left(\theta',s'\right),\left(\theta^{(i)},s^{(i)}\right)\right]}{\pi_{ABC}\left(\theta^{(i)},s^{(i)}\mid s_{obs}\right)g\left[\left(\theta^{(i)},s^{(i)}\right),\left(\theta',s'\right)\right]} \\ &= \frac{K_h\left(s,s_{obs}\right)f\left(s'\mid\theta'\right)\pi\left(\theta'\right)}{K_h\left(s,s_{obs}\right)f\left(s^{(i)}\mid\theta^{(i)}\right)\pi\left(\theta^{(i)}\right)}\frac{g\left(\theta',\theta^{(i)}\right)f\left(s^{(i)}\mid\theta^{(i)}\right)}{g\left(\theta^{(i)},\theta'\right)f\left(s'\mid\theta'\right)} \\ &= \frac{K_h\left(s,s_{obs}\right)\pi\left(\theta'\right)}{K_h\left(s,s_{obs}\right)\pi\left(\theta^{(i)}\right)}\frac{g\left(\theta',\theta^{(i)}\right)}{g\left(\theta^{(i)},\theta'\right)}\end{aligned} \tag{4.4}$$

Again, we see that neatly the intractable likelihood term f vanishes. As we saw in the expository examples of these well known and used algorithms in the case of a joint space exploration (θ, s) things become much simpler since the intractable term vanishes. Assume, for example, that one wants to use a different proposal than the likelihood f , we see that in this case the terms will not cancel in the importance weights and therefore approaches like importance sampling or methods based upon Sequential importance sampling become only theoretical in nature, since implementation is impossible: the weights cannot be computed due to the appearance of the intractable term f. On the other hand assume that we do in fact use the likelihood as the proposal and now consider for example an MCMC update step that would be used inside an SMC sampler or annealed importance sampling.

We define the sequence of targets for the Rare Event SMC ABC algorithm as:

$$\pi_t(x) \propto K_{\epsilon_t}(y_t \mid x_t) f(x_t|\theta) \tag{4.5}$$

with the sequence of $\epsilon_t, t \in \mathcal{T} \subset \mathbb{N}$ being the bandwidth h at every iteration, $i \in \mathcal{I} \subset \mathbb{N}$ denoting the index of the particle. We included the parameter θ here for notational convenience but remember for now that the parameter here is fixed for each run of the "internal" RE-SMC algorithm which we will soon describe.

We see that ABC-MCMC is necessarily (due to the choice of proposal) using an IS estimator of the likelihood and we would therefore like to do better

than that, yet in order to do so we would require a different proposal. One could also see that the difference here between this target and the one in the ABC-MCMC space with the chain targeting the joint θ, y posterior is that we are dealing with a fixed θ and performing inference on the state space y by simulating auxiliary variables u. Hence our proposal here would be just the intractable likelihood itself f, as the only way to propose new x points. Let us not forget we are trying to estimate the likelihood here by running an SMC sampler. We can now see that in trying to perform an MCMC move on the space of y or $s(y)$ would require us to explicitly use the only way we can propose samples in that space i.e f the intractable likelihood. This thusly, renders our algorithm impossible to implement. It is worth noticing exactly why the "trick" of the cancellation of the intractable term f vanishes in the ABC MCMC acceptance probability: the MH ratio represents the target at the proposed parameter value multiplied by a proposal to that value, divided by the target at the current parameter value multiplied by a proposal with the inverse move. Therefore the intractable term given this algorithmic symmetry $(\theta, y) \leftrightarrow (\theta', y')$ appears both in the numerator and denominator and conversely so in the proposal side of things. Thus the terms cancel neatly. This is not the case here: the absence of the joint space which includes the parameter of interest doesn't allow the intractable term to show in some inverse fashion allowing us to cancel it entirely. Here instead we are proposing new points y (which we can only do from the model simulations) from a fixed θ thus the -intractable- likelihood appears in numerator and denominator (which is of course a proxy from simulator draws, based on the same parameter vector). Consider, nevertheless, a rewriting of the ABC "likelihood"

$$\tilde{p}\left(y' \mid \theta\right) = \int_{\mathcal{Y}} K\left(y, y'\right) p(y \mid \theta) \mathrm{d}y \tag{4.6}$$

Assume now that we can instead decompose the simulator either through some transformation we can compute analytically or some "black box" process; here let us focus on the former. Rewriting this proxy likelihood by using some transformation H we have:

$$\tilde{p}\left(y' \mid \theta\right) = \int_{\mathcal{U}} K\left\{H(\theta, u), y'\right\} D(u) \mathrm{d}u \tag{4.7}$$

where $D(u)$ would be some distribution on u-space which makes our posterior distribution of interest have the form

$$\tilde{p}\left(\theta \mid y'\right) \propto \int_{\mathcal{U}} K\left\{H(\theta, u), y'\right\} D(u) p(\theta) \mathrm{d}u \tag{4.8}$$

which in turn means we can now write the intractable MH acceptance probability as :

$$\frac{K_{\epsilon_t}\left(y \mid H\left(u^*, \theta\right)\right) D\left(u^* \mid \theta\right) q\left(u_{t,\theta} \mid \cdot\right) p(\theta) q(\theta|\theta^*)}{K_{\epsilon_t}\left(y \mid H\left(u_{t,\theta}, \theta\right)\right) D\left(u_{t,\theta} \mid \theta\right) q\left(u^* \mid \cdot\right) p(\theta) q(\theta^*|\theta)} \tag{4.9}$$

We now see that provided the distribution of the latent variables u is tractable we have our usual latent variable scenario [2] . We can now calculate everything and thus can run some SMC algorithm to evaluate the ratio of ABC likelihoods, which here will be the Rare Event SMC. This idea was first used in Prangle et al. [2018] as mentioned, where the authors incoprorate the Rare Event SMC estimator of Cérou et al. [2012] in a pseudo-marginal setting of ABC.

[2] in some latent variables models the ABC posterior is not approximate; communicated with emphasis on Wilkinson [2013b], and exploited in Fearnhead and Prangle [2012], Dean et al. [2014]

4.2 Rare Event estimation and SMC

4.2.1 Estimating the ABC likelihood

As we have seen from the introductory chapter of this thesis, and just reiterated in a few expository paragraphs above the likelihood estimate at a point θ is simply $\pi_\epsilon(y \mid x)$, where $x \sim f_\theta(\cdot)$. This is a Monte Carlo estimate of what we call the "true" ABC likelihood

$$l(y \mid \theta) = \int_x \pi_\epsilon(y \mid x) f(x|\theta) dx \tag{4.10}$$

Let is provide a detailed account on how the RE SMC will be used to estimate this likelihood and be used within another SMC sampler to estimate the parameter vector of interest θ with π some Kernel of choice (having previously used the notation K_ϵ for it). As indicated in Del Moral et al. [2012] the estimated ABC likelihood $\pi_\epsilon(y \mid x)$ is a very high variance estimate of the true ABC likelihood since it uses only a single Monte Carlo point, and in certain circumstances it is perhaps more efficient to take the sample average of $\pi_\varepsilon(y \mid \cdot)$ for several simulations from $f_\theta(\cdot)$. While that might seem as imposing an additional computational burden, the decreased variance of the estimator might lead to a decrease in variance of estimates of another algorithm, within which it might be embedded. In this case, for N_x points simulated from $f_\theta(\cdot)$, the estimated likelihood is:

$$\hat{l}(y \mid \theta) = \frac{1}{N_x} \sum_{n=1}^{N_z} \pi_\varepsilon \left(y \mid x^n\right) \tag{4.11}$$

To aid our understanding let us view our Monte Carlo estimator as an importance sampling estimator of the normalising constant $\int_{\mathcal{X}} \pi_e(y \mid x) f_\theta(x) dx$ of the unnormalised target distribution $\pi_\epsilon(y \mid x) f_\theta(x)$ when using proposal $f_\theta(x)$. This importance sampling estimator is unbiased, and its variance depends on the distance between the proposal and the target Agapiou et al. [2017] . In Andrieu and Roberts [2009] we learn that the unbiasedness of the estimated

likelihood will result in ABC-MCMC having the same invariant distribution as if we had used the true ABC likelihood, thus providing a solid theoretical background as to why using such an estimated quantity in an MCMC acceptance probability makes sense. We are therefore in the setting of the pseudo-marginal algorithm of section 1.3.8. One can observe that a likelihood estimator with a higher variance usually results in a less efficient MCMC algorithm (not always the case, see for example Andrieu and Vihola [2016]). For the estimated ABC likelihood, we have that the distance between the unnormalised target $\pi_e(y \mid x)f_\theta(x)$ and proposal $f_\theta(x)$ (and hence the variance of the estimator), will tend to be larger when the dimension of y is higher and when ϵ is smaller. Of particular interest, and the main impetus of this work, as originally envisioned in the MCMC scenario in Prangle et al. [2018] is dealing with the cases of large dimensionality in y-space. The variance of the estimator increases exponentially with the dimension Agapiou et al. [2017] and consequently the full -raw- dataset is rarely used in practice. It is, therefore, common practice to reduce that dimension significantly with one resorting to summary statistics, thereby reducing variance as we previously saw, but nonetheless introducing bias thus changing our target. For example consider the decomposition into the MC error + bias

$$\mathbb{E}\left[\left|\sum_{i=1}^{N} w_i \varrho\left(\theta_i\right) - \pi\left(\varrho \mid \mathbf{s}_{\text{obs}}\right)\right|^p\right]^{1/p} \leqslant \mathbb{E}\left[\left|\sum_{i=1}^{N} w_i \varrho\left(\theta_i\right) - \pi_\epsilon\left(\varrho \mid \mathbf{s}_{\text{obs}}\right)\right|^p\right]^{1/p} + \left|\pi_\epsilon\left(\varrho \mid \mathbf{s}_{\text{obs}}\right) - \pi\left(\varrho \mid \mathbf{s}_{\text{obs}}\right)\right| \tag{4.12}$$

suggested by A.Jasra in Fearnhead et al. [2010][3], see also the proposal of the error form in Marin et al. [2014]. Additionally and perhaps most importantly a low dimensional y space (or summary statistics $s(y)$) due to aforementioned constraints, limits the available θ-space we can attempt to draw inferences on since we must have $d_y > k_\theta$ for $\mathbf{y} \in \mathcal{Y}^d$, $\theta \in \mathcal{G}^k$. In trying to decrease ϵ we see that a similar trade-off is considered. It is important to remember that the ABC "likelihood" will only result in the true posterior when $\epsilon = 0$, and consequently the variance of the estimator will be maximal. It is readily apparent that in practice some value $\epsilon > 0$ is used in order to reduce in some

[3] for some test function $\varrho : \Theta \to \mathbb{R}$, w normalised weights (i.e. for MCMC they would be $1/N$
and $\pi\left(\varrho \mid \mathbf{s}_{\text{obs}}\right) := \int_\Theta \varrho(\theta)\pi\left(\theta \mid \mathbf{s}_{\text{obs}}\right) \mathrm{d}\theta$

sense the variance, trying to avoid particularly high values. This of course results in the introduction of bias as we mentioned. It is therefore this very issue that we are trying to address here by trying to remove the need for the introduction of summary statistics and hence the reduction in the dimension of the state space, while at the same time avoiding a high variance likelihood estimator. We achieve this by adopting the approach of Prangle et al. [2018], thereby introducing an SMC algorithm for computing this estimator, the main advantage being the quadratic instead of exponential scaling as we explained in the introductory section.

4.2.2 Decomposition of the simulator into tractable terms and rare-event SMC

We have already described the reason for the need for such a transformation of the state space in order to circumvent the intractable likelihood in an MCMC update inside the rare event SMC algorithm. Let us see exactly how the rare event algorithm fits into this picture and how it will being utilized for our purposes. It might be worth mentioning that a certain number of approaches operate under the assumption that the parametric or non-parametric nature of a given model for the joint or conditional distribution of θ, y aiming to reduce the variance whilst trying to avoid the introduction of bias.

Instead here we make use of the idea of decomposing the simulator as we outlined previously, replacing the intractable likelihood term f with some tractable function , D such that we can draw samples from it, and a transformation of those draws, say u to our state space of interest y through H. We can in principle of course make those u drawn from $\phi(\cdot|\theta)$ depend on the vector of θ, in order to achieve sufficient generality.

We will see examples where we operate under two scenarios: one where both u and the transformation H can depend on θ and one where the u are drawn from some simple initial distribution with predefined parameters. We are therefore at the situation of which we aimed at the beginning; namely that we can now move around (u, θ)-space jointly. An example of this approach has

been explored in Graham and Storkey [2017] with the usage of Hamiltonian MCMC where the idea is similar in the sense that we try to now perform inference jointly in the input space u here, as well as θ. The motivation is perhaps instructive: we obtain a high variance ABC likelihood since draws from u space are picked independently at each iteration from some distribution that does not depend on y. It is therefore natural to try and introduce some kind of dependence structure (through the transformation and the inference algorithm on that space) such that we can fine-tune these u to specific y and θ. We do so by choosing the random vector u such that the likelihood simulations conditional on θ are sufficiently close to y, thereby increasing the efficiency of our scheme by successively drawing better values of u at each iteration (in the sense of minimizing the distance as mentioned).

We are therefore almost forced to adopt the rare event SMC algorithm since we need a process of simulating from the conditional distribution $[u \mid \theta, y]$, with the aim of tailoring the u to θ and y. The reason for that, is that this fine-tuning approach of tailoring the values of u to those of θ, y results in a decreasing cascade of possible "good" values, thus motivating further the use of a rare event algorithm given the inherent nature of ABC and the explicit dependance on some sequence of decreasing tolerance levels, which obviously imposes a decreasing chance of accepting such appropriate values for u. This results in an estimator of the ABC likelihood $l(y \mid \theta) = \int_u \pi_\epsilon(y \mid H(u, \theta))D(u)du$, as was our original intention, benefiting us additionally with a lower variance estimator than the standard approach, previously mentioned as importance sampling. Given what we just described we are exactly in the scenario of the marginal particle MCMC algorithm of Andrieu et al. [2010], albeit adapted to the ABC framework.

4.3 Algorithmic setup

Let us now describe the rare event SMC ABC estimator[4]:

Assume a sequence of $\tau \in \mathbb{N}$ targets with the final one being $\pi_\epsilon(y \mid H(u,\theta))\phi(u)$. The "0th target" (the proposal) is given by $\phi(u)$, and the t^{th} target (for $1 \leq t \leq \tau$ is $\pi_{\epsilon_t}(y \mid H(u,\theta))\phi(u)$, where $\infty > \epsilon_1 > \ldots > \epsilon_\tau = \epsilon$.

1. Draw N_u points from the proposal.
2. For $t = 1 : T$ re-weight each point by multiplying its current weight by

$$\frac{\pi_{\epsilon_t}(y \mid H(u,\theta))}{\pi_{\epsilon_{t-1}}(y \mid H(u,\theta))} \tag{4.13}$$

3. resample and execute an MCMC move with target $\pi_{\epsilon_t}(y \mid H(u,\theta))\phi(u)$.

the ABC likelihood can then be estimated by taking the average of the weights at each step, and subsequently calculating the product of these averages over SMC iterations. The normalising constant $l_t(y \mid \theta) = \int_u \pi_{\epsilon_t}(y \mid H(u,\theta))\phi(u)du$ is the ABC likelihood with tolerance ϵ_t. When updating to the target at iteration $t+1$ the weights can be used to estimate $l_{t+1}(y \mid \theta)/l_t(y \mid \theta)$ as we just described (this is the term needed in the weight update in the external SMC as we will soon see). Using a tolerance sequence of $\epsilon_1 > \ldots > \epsilon_T$ we now are ready to define the rare-event SMC ABC algorithm.

For our method, of course to be efficient we must be careful in how we actually propose points in u-space and thus explore that space. For our first toy model where we demonstrate the superiority of our algorithm we adopt the same algorithm used in Prangle et al. [2018], which is slice sampler that is observed to be quite efficient in moving around $u - space$ for this particular problem. It should be noted that it can be quite tedious to design appropriate MCMC moves on that space, and a bad algorithm design at this level of our

[4]when one uses the uniform kernel, the work of Cérou et al. [2012], explores such an example of an SMC algorithm, which can be clearly seen to be a sub-case of more general rare-event style SMC algorithms

construction can seriously hinder the performance of the entire edifice, thus possibly erasing the computational gains we started with at a theoretical level and for which is the main reason for proposing this approach. The reason for that hindrance is the fact that the changing epsilon which gets passed down from the external algorithms to the innermost u-space sampling one results in a dramatic reduction of the scale of the posterior, and therefore an automatic or at least semi-automatic way of constructing the MCMC move is needed in order to have some a degree of efficiency (in an MCMC moves sense), since very different MCMC moves will be efficient to different epsilons. One can think of this as somewhat equivalent to the issue of different scales of the dimension on some general sampling problem, and for which approaches such as HMC Neal [2011], or MALA-type algorithms Caimo and Friel [2011] come into play (with their benefits -high dimensionality performance- and issues, such as robustness to tuning etc).

To get a clear understanding of why the subsequent rare-event ABC SMC algorithm has the form that it does let us first note the algorithm proposed in Cérou et al. [2012]:

Algorithm 12: Rare event SMC algorithm, with adaptive ϵ sequence Cérou et al. [2012]

Data: Parameters θ, number of particles N, target number to accept N_{acc}, map H

Output: $\hat{P}$, approximation to ABC likelihood

for $t = 1 : N$ **do**
  Let ϵ_t be the maximum of (a) the N_{acc} such that the smallest $H_\theta(u_{t-1}^2)$ value and (b) ϵ ; `// A bisection routine between` ϵ_{t-1} `and` ϵ `is run here to calculate the maximum` ϵ_t `such that at least` N_{acc} `particles will be accepted`
  Calculate $I_t = \left\{ i \mid H\left(u_{t-1}^{(i)}\right) \leq \epsilon_t \right\}$ and $\hat{P}_t = |I_t| / N$; `// Here` $\hat{P}_t$ `the small probability estimate`
  for $i = 1 : N$ **do**
    sample $u_t^{(i)}$ by drawing j uniformly from I_t and applying a Markov kernel
    to $u_{t-1}^{(j)}$ with invariant density $\pi\left(u \mid \theta, H_\theta\left(u\right)\right) \leq \epsilon_{t-1})$ (taking $\epsilon_0 = \infty$).
    If $\epsilon_t = \epsilon$ break loop and go to step 8, setting $T = t$ **return** $\hat{P} = \prod_{t=1}^{T} \hat{P}_t$
  end
end

In algorithm 15, we have indicated a generic Markov kernel as originally defined in Cérou et al. [2012] but we will use initially a slice sampling update as in Prangle et al. [2018] due to the benefits discussed therein.

The algorithm for the slice sampler is given below:

Algorithm 13: Slice sampling update for rare event SMC Prangle et al. [2018]

Data: current state x of dimension p, map $H(x)$, threshold ϵ, initial search width w. It's assumed that $K\{H_\theta(u)\} \leq \epsilon$

Output: $\hat{P}$, approximation to ABC likelihood

while *true* **do**

- Sample $v \sim N(0, I_p)$
- Sample $u \sim$ Uniform $(0, w)$. Let $a = -u, b = w - u$
- Sample $z \sim$ Uniform (a, b)
- Define a vector x' by $x'_i = r(x_i + zv_i)$ using the reflection function:

$$r(y) = \begin{cases} m & m < 1 \\ 2 - m & m \geq 1 \end{cases} \tag{4.14}$$

- where m is the remainder of y modulo 2 .
- If $\Phi(x') \leq \epsilon$ then return x'
- If $z < 0$ let $a = z$, otherwise let $b = z$

end

We can now formulate the Rare-Event SMC for our case in the ABC

setting. The algorithm is given below:

Algorithm 14: Rare event SMC algorithm

Data: parameter θ, thresholds $\epsilon_{1:t}$

if $t = 1$ **then**

  for $1 \leq m \leq N_u$, **do**

    Sample u_0^n from $\phi(u \mid \theta)$.

    compute and normalise internal weights

$$\tilde{w}_{1,\theta}\left(u_{0,\theta}^m\right) = \frac{\pi_{\epsilon_1}\left(y|H\left(u_{0,\theta}^m,\theta\right)\right)\phi\left(u_{0,\theta}^m|\theta\right)}{\phi\left(u_{0,\theta}^m|\theta\right)} = \pi_{\epsilon_1}\left(y \mid H\left(u_{0,\theta}^m,\theta\right)\right)$$
$$w_{1,\theta}^m = \frac{\tilde{w}_{1,\theta}\left(u_{0,\theta}^m\right)}{\sum_{i=1}^{N_u}\tilde{w}_{1,\theta}\left(u_{0,\theta}^n\right)} \tag{4.15}$$

    – sample $a_{1,\theta}^n$ from $\mathcal{M}\left(w_{1,\theta}^{1:N_u}\right)$ (the multinomial distribution with parameters $\left(w_{1,\theta}^{1:N_u}\right)$, or use another unbiased resampling method).

    – use an MCMC like the one given below in algorithm 15 move on θ giving result $u_{1,\theta}^n$ with invariant distribution $\pi_{\epsilon_1}(y \mid H(u,\theta))\phi(u \mid \theta)$.

  end

else

  for $1 \leq m \leq N_u$, **do**

    compute and normalise internal weights

$$\tilde{w}_{t,\theta}^m\left(u_{t-1,\theta}^{a_{t-1,\theta}^m}\right) = \frac{\pi_{\epsilon_t}\left(y \mid H\left(u_{t-1,\theta}^{a_{t-1,\theta}^m},\theta\right)\right)\phi\left(u_{t-1,\theta}^{a_{t-1,\theta}^m} \mid \theta\right)}{\pi_{\epsilon_{t-1}}\left(y \mid H\left(u_{t-1,\theta}^{a_{t-1,\theta}^m,\theta)}\right)\phi\left(u_{t-1,\theta}^{a_{t-1}^m} \mid \theta\right)} = \frac{\pi_{\epsilon_t}\left(y \mid H\left(u_{t-1,\theta}^{a_{t-1,\theta}^m},\theta\right)\right)}{\pi_{\epsilon_{t-1}}\left(y \mid H\left(u_{t-1,\theta}^{a_{t-1,\theta}},\theta\right)\right)} \tag{4.16}$$

$$w_{t,\theta}^m = \frac{\tilde{w}_{t,\theta}\left(u_{t-1,\theta}^{a_{t-1,\theta}^n}\right)}{\sum_{i=1}^{N_u}\tilde{w}_{t,\theta}\left(u_{t-1,\theta}^{a_{t-1,\theta}^n}\right)} \tag{4.17}$$

    Sample $a_{t,\theta}^n$ from $\mathcal{M}\left(w_{t,\theta}^{1:N_u}\right)$ (or using another unbiased resampling method).

    For each $1 \leq n \leq N_u$, use an MCMC move given below in algorithm 15 on $u_{t-1,\theta}^{a_{t,\theta}^n}$ giving result $u_{t,\theta}^n$ with invariant distribution $\pi_{\epsilon_t}(y \mid H(u,\theta))\phi(u \mid \theta)$

  end

end

Algorithm 15: MCMC moves for Rare-Event ABC SMC

for $1 \leq m \leq N_u$, **do**

- sample $u^* \sim q_t\left(\cdot \mid u_{t,\theta}^m\right)$ - let $u_{t,\theta}^m = u^*$ with probability

$$\frac{\pi_{\epsilon_t}\left(y \mid H\left(u^*, \theta\right)\right) \phi\left(u^* \mid \theta\right) q\left(u_{t-1,\theta}^{a_{t,\theta}^m} \mid u^*\right)}{\pi_{\epsilon_t}\left(y \mid H\left(u_{t-1,\theta}^{a_{t,\theta}^m}, \theta\right)\right) \phi\left(u_{t-1,\theta}^{a_{t,\theta}^m} \mid \theta\right) q\left(u^* \mid u_{t-1,\theta}^{a_{t,\theta}^m}\right)} \tag{4.18}$$

- otherwise let $u_{t.\theta}^m = u_{t-1,\theta}^{a_{t,\theta}^n}$.

end

For one of our numerical experiments we will make use of the slice sampler updates as experiments show it performs well given that $u \in [0,1]^d$, and a general MCMC update depending on the model we are interested in as we will see in our experiments, since the u-space in each case can be quite different. The output of algorithm 14 we get an estimate of the ABC 'likelihood'

$$\bar{l}\left(y \mid \theta\right) = \prod_{t=1}^{T} \sum_{n=1}^{N_u} \tilde{w}_t^n. \tag{4.19}$$

.

Given that we now have an estimator of the ratio of likelihoods 4.13 we can finally incorporate that into the external parameter space exploration algorithm like in the SMC^2 of Chopin et al. [2013], by using rare event ABC to estimate likelihood ratios. SMC^2 is designed for a state space model setting: $y_{1:t}$ are noisy observations of the latent time series $x_{1:t}$. Our generative model for this situation is specified in two parts: $f_\theta\left(x_{1:t}\right)$, which models the dynamics of the latent time series, and $g_\theta\left(y_{1:t} \mid x_{1:t}\right)$. which models the distribution of the observations. SMC^2 may be used to estimate the posterior distribution on both θ and $x_{1:t}$. It is set up using an "external" SMC on θ-space, and an "internal" SMC on x-space conditional on θ. The internal SMC has target $f_\theta\left(x_{1:t}\right) g_\theta\left(y_{1:t} \mid x_{1:t}\right)$ at iteration t. When updating to $f_\theta\left(x_{1:t+1}\right) g_\theta\left(y_{1:t+1} \mid x_{1:t+1}\right)$ at iteration $t+1$, we have added in the terms $f_\theta\left(x_{t+1} \mid x_{1:t}\right) g_\theta\left(y_{t+1} \mid x_{1:t+1}, y_{1:t}\right)$, so that the weights at iteration t can be used to estimate $p\left(y_{t+1} \mid y_{1:t}, \theta\right) = p\left(y_{1:t+1} \mid \theta\right) / p\left(y_{1:t} \mid \theta\right)$ (this being the term needed in the weight update in the external SMC).

For comparison and ease of understanding given how we have described the SMC^2 algorithm in the introductory chapters, we follow as closely as possible, the notation in the SMC^2 paper, with exceptions pertaining to clarity within this thesis and overall notation usage, as well as references to specific papers; we will indicate and alter notation accordingly when the danger for obfuscation arises. First, note that $\tilde{w}_{t,\theta}\left(u_{t-1,\theta}^{a_{t-1,\theta}^n}\right)$ does not depend on $u_{t,\theta}^n$ (this is always the case when we use an MCMC move within our SMC). As in that paper, we can let, for $t = 1$, $\psi_{1,\theta}\left(u_1^{1:N_u}, a_1^{1:N_u}\right)$ and, for $t = 2 : T, \psi_{t,\theta}\left(u_{1:t}^{1:N_u}, a_{1:t}^{1:N_u}\right)$, be the joint distribution of all of the random variables generated up to time t [5]. The "internal" SMC algorithm is the rare event SMC method introduced

[5] note the notation alteration of sampling a_t at iteration t

above. The "external" SMC algorithm is given below:

Algorithm 16: Rare event ABC-SMC^2 algorithm

for $t = 1 : T$ **do**
  for $1 \leq m \leq N_\theta$ **do**
    sample θ^m from $p(\theta)$, and set $\omega^m \leftarrow 1$.
    if $t = 1$, **then**
      sample $\left(u_1^{1:N_u,m}, a_1^{1:N_u,m}\right)$ from ψ_{1,θ^m}, (i.e. run algorithm 14 for $t = 1$) and compute the estimate of the ABC likelihood l_1 when using ϵ_1

$$l_1\left(\widehat{y \mid \theta_0^m}\right) = \frac{1}{N_u}\sum_{n=1}^{N_u} \tilde{w}_{1,\theta}\left(u_{1,\theta}^{n,m}\right) \tag{4.20}$$

      and update the importance weights

$$\omega^m \leftarrow \omega_0^m l_1\left(\widehat{y \mid \theta_0^m}\right) \tag{4.21}$$

    else
      sample $\left(u_t^{1:N_u,m}, a_t^{1:N_u,m}\right)$ from ψ_{t,θ^m} conditional on $\left(u_{1:t-1}^{1:N_u,m}, a_{1:t-1}^{1:N_u,m}\right)$ (i.e. run t algorithm 14 for the t_{th} step)) and compute the estimate of ratio l_t/l_{t-1} of the ABC likelihoods when using ϵ_t and ϵ_{t-1}

$$\frac{l_t\left(\widehat{y \mid \theta_{t-1}^m}\right)}{l_{t-1}\left(y \mid \theta_{t-1}^m\right)} = \frac{1}{N_u}\sum_{n=1}^{N_u} \tilde{w}_{t,\theta_{t-1}}\left(u_{t-1,\theta_{t-1}}^{a_{t-1,\theta_{t-1}}^n}\right) \tag{4.22}$$

      and update the importance weights

$$\omega^m \leftarrow \omega_{t-1}^m \frac{l_t\left(\widehat{y \mid \theta_{t-1}^m}\right)}{l_{t-1}\left(y \mid \theta_{t-1}^m\right)} \tag{4.23}$$

    end
  end
  $\{\omega_t^m\}_{m=1}^{N_\theta} \leftarrow$ normalise $\left(\{\tilde{\omega}_t^m\}_{m=1}^{N_\theta}\right)$;
  if *some degeneracy condition is met* **then** `// resample and move`
    for $m = 1 : N_\theta$ **do**
      Simulate $\left(\theta_t^m, u_{1:t}^{1:N_u,m}, a_{1:t-1}^{1:N_u,m}\right)$ from the mixture distribution

$$\sum_{i=1}^{N_\theta} \omega_t^i K_t\left\{\cdot \mid \left(\theta_{t-1}^i, u_t^{1:N_u,i}, a_t^{1:N_u,i}\right)\right\},$$

      where K_t is the MCMC move from Prangle et al. [2018], i.e.: $i^* \sim \mathcal{M}\left(\{\omega_t^i\}_{i=1}^{N_\theta}\right)$, then $\theta^* \sim q_t\left(\cdot \mid \theta_{t-1}^{i^*}\right)$, then run algorithm 14 up to ϵ_t conditional on θ^*.
      Set $\theta_t^m = \theta^*$ and $u_{1:t}^{n,m}, a_{1:t-1}^{n,m}$ and $\tilde{w}_{1:t}^{n,m}$ to be the variables and unnormalised weights generated when running algorithm 14 with probability

$$1 \wedge \frac{p\left(\theta^*\right)}{p\left(\theta_{t-1}^{i^*}\right)} \frac{q\left(\theta_{t-1}^{i^*} \mid \theta^*\right)}{q\left(\theta^* \mid \theta_{t-1}^{i^*}\right)} \frac{\bar{l}\left(y \mid \theta^*\right)}{\prod_{t=1}^{T}\sum_{n=1}^{N_u} \tilde{w}_t^{n,*}},$$

      where $\bar{l}$ is defined in equation 4.19;
      Else set $\theta_t^m = \theta_{t-1}^{i^*}$, $\tilde{w}_{1:t}^{n,m} = \tilde{w}_{1:t}^{n,i^*}$, $u_{1:t}^{n,m} = u_{1:t}^{n,i^*}$ and $a_{1:t-1}^{n,m} = a_{1:t-1}^{n,i^*}$.
    end
    $\omega_t^m = 1/N_\theta$ for $m = 1 : N_\theta$;
  end
end

4.3.1 Adapting the sequence of tolerances

For the new rare event algorithm, as in ABC-SMC, we can adaptively choose the sequence of tolerances ϵ_t. In order to do that we will need to be careful about the way we calculate the correct new ϵ level and the technique used to do that. In Del Moral et al. [2012] the selection of the new tolerance level ϵ is done such that the following condition holds: $\text{ESS}(\{W_n^i, \epsilon_n\} = \alpha\text{ESS}(\{W_{n-1}^i, \epsilon_{n-1}\}$ for some $\alpha \in (0,1)$ with the weights being the approximation of the ratio of likelihoods as defined in the algorithm above and alpha a percentage of particles that we want to survive onto the next iteration. Here, let $W_{t-1}^{(i)}$ denote the normalised weight of particle i at time $t-1$, and let $w_t^{(i)}$ denote the unnormalised incremental weight of particle i at iteration t. Then, the ESS is calculated by using the current weight of each particle as follows:

$$\text{ESS}_t = \left[\sum_{j=1}^{N}\left(\frac{W_{t-1}^{(j)}w_t^{(j)}}{\sum_{k=1}^{N}W_{t-1}^{(k)}w_t^{(k)}}\right)^2\right]^{-1} = \frac{\left(\sum_{j=1}^{N}W_{t-1}^{(j)}w_t^{(j)}\right)^2}{\sum_{k=1}^{N}\left(W_{t-1}^{(k)}\right)^2\left(w_t^{(k)}\right)^2} \tag{4.24}$$

The calculated ESS of the weights at some time t relays the information of the accumulated mismatch between the proposal distribution and the target (when one thinks about it as an extended space with the full trajectory of the sample paths being in it), since the last resampling phase. The authors in Zhou et al. [2016] note that by either fixing the relative or absolute reduction in the calculated ESS between the successive distribution of the SMC algorithm does not result in a common discrepancy measure between them, with the exception of the case where resampling is performed at every iteration. Therefore the authors argue it is preferable and does result in the desired common measure if one instead uses the conditional ESS as defined in their work and given below:

$$CESS_t = \left[\sum_{j=1}^{N}NW_{t-1}^{(j)}\left(\frac{w_t^{(j)}}{\sum_{k=1}^{N}NW_{t-1}^{(k)}w_t^{(k)}}\right)^2\right]^{-1} = \frac{N\left(\sum_{j=1}^{N}W_{t-1}^{(j)}w_t^{(j)}\right)^2}{\sum_{k=1}^{N}W_{t-1}^{(k)}\left(w_t^{(k)}\right)^2} \tag{4.25}$$

which in our case becomes:

$$CESS_t = \frac{N_\theta \left(\sum_{m=1}^{N_\theta} w_{t-1}^m \frac{\widehat{l_t\left(y|\theta_{t-1}^m\right)}}{l_{t-1}\left(y|\theta_{t-1}^m\right)} \right)^2}{\sum_{m=1}^{N_\theta} w_{t-1}^m \left(\frac{\widehat{l_t\left(y|\theta_{t-1}^m\right)}}{l_{t-1}\left(y|\theta_{t-1}^m\right)} \right)^2}, \tag{4.26}$$

4.4 Numerical experiments

4.4.1 High dimensional Gaussian toy model

For our first experiments we consider a high d-dimensional, truncated at zero, Gaussian example where we draw $\{x_i\}_{i=1:d} \sim \mathcal{N}_i(\mu_i, \sigma_i)$ with $x_i \in \mathbb{R}^+$ samples of dimension d and we would like to infer the variance with the mean known, i.e. $\mu = 0$ for all components. We will not consider any dependence between components and therefore the "multivariate" Normal has zeros in the covariance matrix everywhere except the diagonal and hence we are in a scenario with i.i.d draws from a product of 1-D Gaussians where we want to infer the σ (which is the same) for all dimensions. These would be our "observations" x_i for which we will simulate against.

A few implementation details that are important in the context of inference here. In early experiments simulating from a Normal distribution resulted in a subsequence of tolerances for which higher weights were assigned to lower values (than the known one) of the parameter of interest, resulting in the sequence of posteriors being skewed initially towards that area of the parameter space and thus inhibiting the move of particles towards the true parameter value[6] (the reason being that thinking of the samples as coming from a symmetric distribution, here the Normal one, we get that for a large tolerance level the average euclidean distance between all the x_{obs} and the simulated ones x_i^m for each particle m gives a certain distribution of those distances. The algorithm at every level chooses a cutoff such that certain percentage of the particles survive and hence this distance distribution gets truncated at that level. Initially smaller values of θ which means a Gaussian with smaller variance constitute most of that initial distribution of distances (despite not giving the smallest distances) and hence those particles have a higher chance of propagating forward. But as the algorithm progress and a smaller tolerances are imposed eventually the algorithm only accepts those proposals of θ that generate the data with the closest distance to the observed ones). In the re-

[6] which nonetheless did not affect overall the algorithm but the model exhibited pathological behaviour and therefore a more appropriate modification was needed in order to ensure that the algorithm is not inhibited by factors other than its intrinsic ones

freshment MCMC step of the algorithm we propose new parameter values with a normal distribution that has variance equal to the variance of the particle population and centered around the current value (for each particle) , since the samples become progressively more concentrated and thus some notion of dynamic scaling of proposal is needed in order to not have most of the samples rejected. It is worth re-iterating the mechanism of the algorithm: a new initialisation of the RE-SMC-ABC algorithm is run for each proposed θ, for the entire adaptively chosen schedule up to current time $\epsilon_{1:t}$, since the internal algorithms assumes a given theta for which the u are combined with and through the transformation give new pseudo-data y_i and thus new distances.

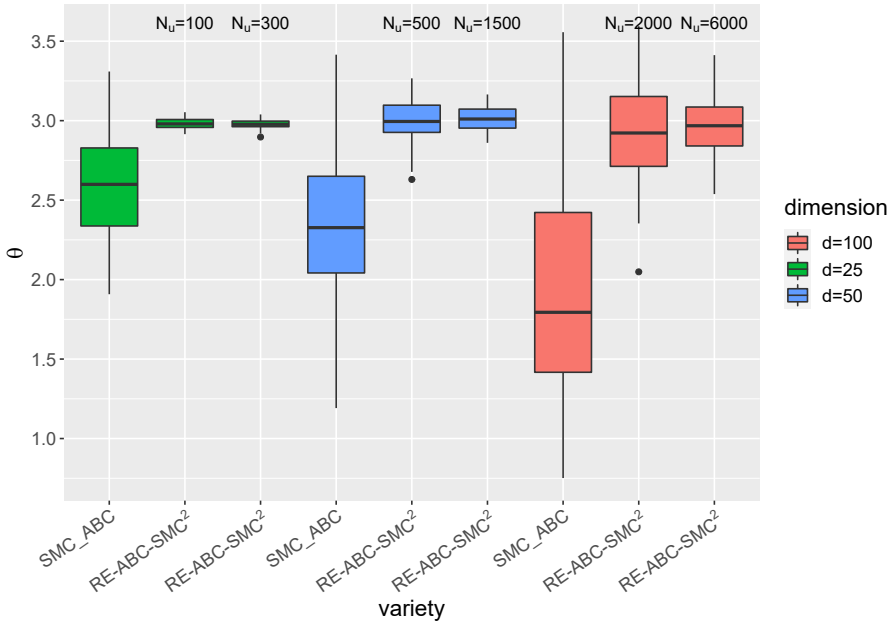

Figure 4.1: Comparison of empirical means between ABC-SMC and RE-ABC SMC^2 for different dimensions for the Gaussian model, over 50 replications of each run. The true value of the parameter is $\theta = \mathbf{3.0}$. The ABC-SMC algorithm was run for a similar time frame as the RE-ABC SMC^2 in order to provide an accurate representation of inference quality given computational resources available. Both algorithms adaptively choose the number of the MCMC refreshment steps after the resampling scheme. The resampling takes place when ESS drops below half of the current number of particles (on theta space). The number of internal particles for RE-ABC SMC^2 is indicated in the figure.

In our numerical experiments in order to properly compare the SMC-

ABC and $RE-ABC-SMC^2$ algorithms we need to equate in some sense the computational effort required by both since it is obvious that the nested structure of our algorithm induces a significantly higher computational cost. As such one approach would be to equate the number of likelihood draws in total between them since in likelihood free methods such as ABC it is often if not always the case that the simulator is by far dominating the computational budget and other operations within the algorithm one uses are orders of magnitude less demanding of the total budget (comparatively speaking; with the exception perhaps of concurrency issues and communication in largely parallel algorithms). For example in the SMC-ABC case we have a cost of approximately $[N]t + [N] \cdot k_{MCMC} \cdot [r]$ whereas in the $RE-ABC-SMC^2$ we have a cost of $N_\theta \cdot N_u \cdot + [r] \cdot N_\theta \cdot N_u \cdot k$, with $N =:$ number of particles, t the random number of iterations of the adaptive algorithms, r the resampling times and k the number of MCMC refreshment steps. Given the adaptive nature of the algorithms it makes such comparison hard although one could calculate some minimum values, which is what we did here. For example in order to equate the $RE-ABC-SMC^2$ sampler of $N_\theta = 250$ and $N_u = 250$ particles, we are comparing to an SMC-ABC sampler of $N_\theta = 600000$ particles, since the internal RE-SMC of our algorithm needs to perform all the time steps up until the time t_r again since a new θ has been proposed. The same is true for the MCMC moves. In our experiments this heuristic approach is more or less consistent with the overall computational time measured in CPU seconds, which after all is the only real metric that a practitioner will be interested in when deciding which algorithm to use, since in the framework of ABC we are interested in obtaining the smaller epsilon possible for the computational budget available (assuming the model is correct) and therefore a better approximation to the ABC posterior (and consequently the "true" posterior) with a lower variance and minimum bias. An important addition to the algorithm that will assist in ameliorating the issue that persists in all ABC algorithms (even if it performs much better as in our case) no matter how efficient the acceptance probability of MCMC moves, is the refreshment stage after resampling. It is evident that for very small epsilons, the acceptance probability will decrease rapidly, and such was the case here. Perhaps it is indicative of the limits of our algorithm given that even with such fine control of the source of randomness as here through the RE-SMC, there can still be issues with the algorithm proposing parameters that result in observations being extremely close to the data. In

order to overcome this issue to a certain degree, all algorithms were run with an adaptive schedule for the number of MCMC refreshment moves. The modification is based on South et al. [2019], where the authors propose the following expression to calculate the number of MCMC steps N.

$$N_t = \left\lceil \frac{\log(c)}{\log\left(1 - \hat{p}^t_{\text{acc}}\right)} \right\rceil \tag{4.27}$$

with $\lceil \cdot \tau \rceil$ denoting the ceiling function and $\hat{p}^t_{\text{acc}} = \frac{1}{N} \sum_{i=1}^{N} \alpha\left(\theta^i_t, \theta^{i,*}_t\right)$ is the acceptance rate based on the first MCMC iteration on the N particles. The idea is to have a theoretical probability $1 - c$ that a particle is moved at least once. At this point we should mention that the reason for stopping at given epsilon levels for each dimensionality, is that the SMC would indeed collapse for very small values, and it would be clear by monitoring the acceptance rate of the MCMC refreshment steps that it would do so as 1 out of N external particles would be accepted at those very low levels. Consequently the above adaptive procedure would result in a very large number of MCMC moves in order to propose particles that would be accepted and since every resampling and MCMC step at the external SMC level induces the internal RE-SMC to re-run (since it has sampled a new θ) from the initial ϵ_0 up to ϵ_t until that time of resampling t the computational burden would be correspondingly very large. As such a decision has to be made on the number of maximum MCMC moves one is allowed to perform, or dynamically save the particle population and actively monitor the acceptance rate and proposed N of MCMC moves. If those seem to have plateau for a prolonged period of time, it is best to terminate the algorithm. This is the procedure we followed here as well. It is important of course to not let the population be very close to collapsing and terminate prior to that event.

Given the tuning choices outlined above we can see from our first and perhaps most important graph for this toy model that the numerical experiments do verify the theoretical motivation and justification for the algorithm. For a given computational budget we can see that in figure 4.1 the RE-ABC-SMC^2 significantly outperforms SMC-ABC since the latter obtains a worse estimate of the ABC posterior, by not being able to get to a lower tolerance level. The discrepancy is moreover increasing as the dimensionality increases

which is what we expected given the theoretical considerations in the introduction. An increase in state space dimensionality means that the estimate of the ABC likelihood is carried out by a importance sampling step which as we know scales exponentially in the number of samples and this requires an extremely large number of particles to keep up with (or as we previously mentioned) more than 1 sample for each estimate which would be equivalent to a multiplication of particles by that amount, say M (although with a lower cost since each particle needs to go through a series of steps externally and thus overall contribute more to the cost than individual exactly similar operations within particles).

Furthermore, the increase in the number of internal particles N_u for which the estimate of the ration of ABC likelihoods is carried out, results in a decrease in variance, consistent with theoretical results.

The point above is demonstrated further by looking at figures 4.2,4.3, 4.4, for state space dimensions $d = 25, 50, 100$ respectively, where for a given computational budget the RE-ABC-SMC^2 gets down to a much smaller epsilons, and hence to a lower variance more accurate approximation of the posterior.

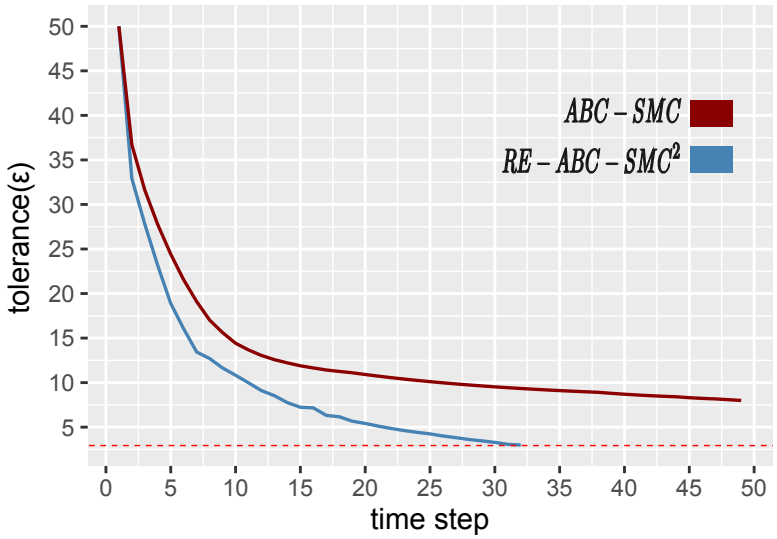

Figure 4.2: Comparison of adaptive schedules for ABC-SMC and RE-ABC SMC^2 in dimension $\mathbf{d} = 25$ The algorithm parameters are the same as in figure 4.1. The total run time for both algorithms is dictated by the time it takes for RE-ABC-SMC^2 to reach a pre-defined threshold of epsilon. Here $\epsilon = 3$.

.

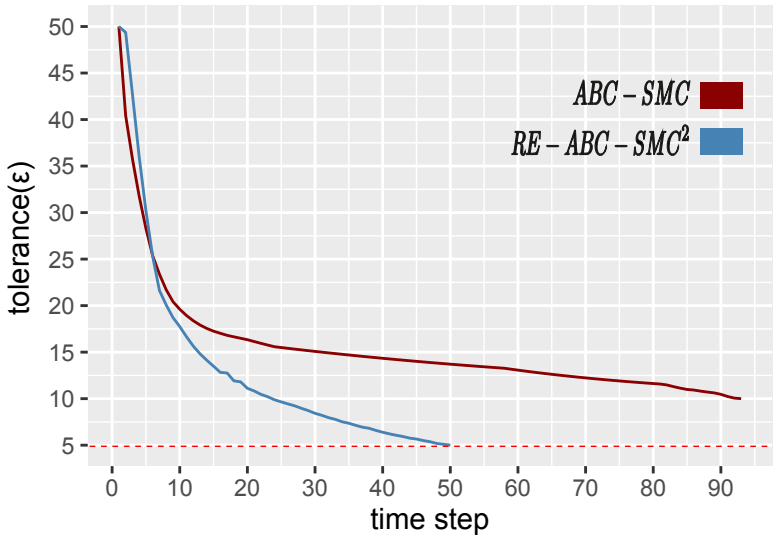

Figure 4.3: Comparison of adaptive schedules for ABC-SMC and RE-ABC SMC^2 in dimension $\mathbf{d} = 50$ The algorithm parameters are the same as in figure 4.1. The total run time for both algorithms is dictated by the time it takes for RE-ABC-SMC^2 to reach a pre-defined threshold of epsilon. Here $\epsilon = 5$.

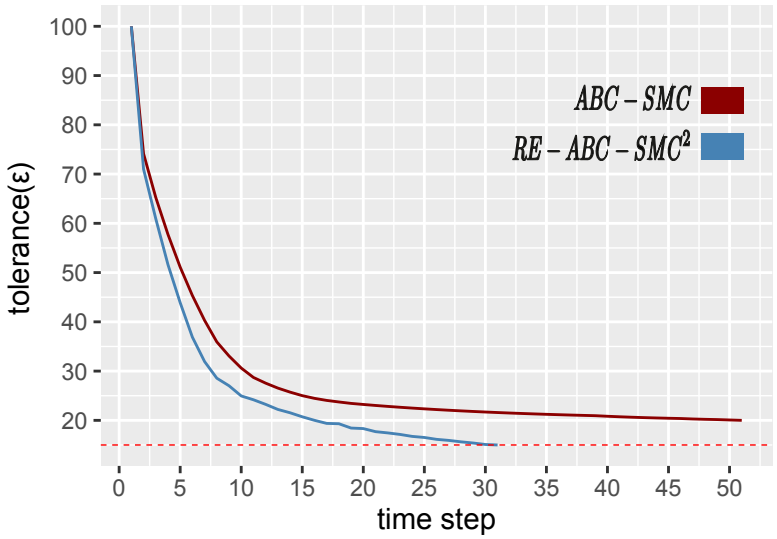

Figure 4.4: Comparison of adaptive schedules for ABC-SMC and RE-ABC SMC^2 in dimension $\mathbf{d} = 100$ The algorithm parameters are the same as in figure 4.1. The total run time for both algorithms is dictated by the time it takes for RE-ABC-SMC^2 to reach a pre-defined threshold of epsilon. Here $\epsilon = 10$.

4.4.2 Duplication divergence random graph model

Here we will be dealing with an interesting example of a biologically inspired network growth model, aptly named duplication-divergence model. The model has two parameters, call them p and r, with the first representing the probability that edges are retained when a node is duplicated while the second the probability that the duplicated node forms an edge to the new node. The process repeats by adding nodes to the network, one at a time, and duplicating an existing node (and potentially its edges).

Take an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ represented as a set of nodes $\mathcal{N}$ and edges $\mathcal{E}$, at each iteration of the process, we select a node $x_i \in \mathcal{N}$ uniformly at random to duplicate. To carry out the procedure, we first create a new node in the network x^*, with no edges. Then we take all nodes $x_j \mid (x_i, x_j) \in \mathcal{E}$ neighbouring x_i, and attach them to the new node x^* forming new edges (x_j, x^*), each with probability p. Finally the new node x^* is connected to the node that was duplicated, x_i, forming an edge (x_i, x^*), with probability r. This process is carried out until the desired number of nodes in the network is attained. In practice a seed network is regularly used as an initial structure from which steps of the duplication-divergence process are carried out. In our test case we will be using an Erdős-Rényi random graph, which is constructed by forming each of the $N(N-1)/2$ possible edges in an undirected graph of N nodes with probability a.

To see how this fits into our RE ABC-SMC sampler consider the variables $u \in \mathcal{U} \subset [0,1]^d$ representing the random draws generated when simulating the model. In certain such models, it is not possible to directly derive an MCMC algorithm that will draw samples from u-space. In the duplication divergence model, some of the u will correspond to the binary choice of adding or keeping the existing number of edges, between a newly created node x^* and one of the neighbours x_j of the existing node x_i that was chosen to be duplicated. It is however the case that the number of such u will depend on the number of edges x_i has, which in turn could depend on previous values of u used were formed when node x_i was created. Consequently, the dimensionality d of $\mathbf{u}^d$ will change dependent on its value in an arbitrary (non-a priori obvious) way. In order to

overcome this issue, we propose to partition u into two sets, u^s and u^r. We require that the dimension of u^s is fixed, and that there possibly a scheme to perform MCMC moves on u^s with invariant distribution $\pi_{\epsilon_t}(y \mid H(u,\theta))\phi(u \mid \theta)$ where $u = (u^s, u^r)$. The remaining -of random variable length- u^r are sampled from $\phi(u^r \mid u^s, \theta)$ which corresponds to the transformation H described in section 4.2.2 and their dimensionality thus remains fixed as the output of the simulator (given the samples from the rare event algorithm u^r is set from the user; in our case we set it to d=100 as described below.

In the context of the duplication-divergence model, we take u^s to be the set of Bernoulli random variables used to construct the seed graph. For each pair of nodes in the seed graph of size N, an edge is added with probability a. For a seed graph of size N, there are $M = N(N-1)/2$ possible edges, and hence $u^s = u^s_1, ..., u^s_M$. Each of these variables encodes the presence ($u^s_i = 1$) or absence ($u^s_i = 0$) of an edge in the seed network (with the index denoting the node number in some arbitrary ordering. We then obviously have (given the Bernoulli assumption) that the distribution of this RV is $\phi(u^s_i|a) = a^{u^s_i}(1-a)^{1-u^s_i}$.

To construct an MCMC kernel $\mathcal{P}$ on u^s, we apply a Metropolis-Hastings sampler with a proposal that either adds or deletes an edge in the seed network with equal probability $q_{\text{add}} = q_{\text{del}} = 0.5$. When an edge addition proposal is chosen, one of the $N(N-1)/2 - |\mathcal{E}|$ possible pairs of unconnected nodes is selected uniformly at random, and an edge added between them. For an edge deletion proposal, one of the $|\mathcal{E}|$ edges in the seed network is chosen uniformly at random and deleted.

To calculate the distance between two unlabelled and undirected graphs $\mathcal{G}_1$ and $\mathcal{G}_2$, we employ an approximation to the edit distance between them. This is defined as the smallest number of edges that would need to either be added to or deleted from $\mathcal{G}_1$ or $\mathcal{G}_2$ for the two graphs to become isomorphic. The edit distance is prohibitively expensive form a computational standpoint but can be in fact approximated, following Thorne and Stumpf [2012] using the ordered eigenvalues $\alpha_1, \ldots, \alpha_N$ and $\beta_1, \ldots, \beta_N$ of the adjacency matrices

of $\mathcal{G}_1$ and $\mathcal{G}_2$ respectively as

$$(\mathcal{G}_1, \mathcal{G}_2) \approx \sum_i (\alpha_i - \beta_i)^2 \tag{4.28}$$

The influence of the seed network has a strong impact on the structure of the final one and therefore it is of considerable interest to infer "good" seed networks.

Numerical experiments

We are constructing our seed network by using $N = 20$ number of nodes and $\alpha = 0.5$. The inference of the RE-SMC sampler is performed on the seed-network space, and in particular we are trying to infer what are "good" networks. The seed networks are then fed into the duplication divergence overall model where the final network has node size of $N = 100$. Given the parameters here, and taking into account the number of possible edges for the seed graph we are in fact dealing with a dimensionality of $d = 190$ for the $u^i_{seed}, i \in [1 : d]$. The sample space of u is a discrete space as it can only take values in $\{0, 1\}$ and therefore $u \in \{0, 1\}^d$. The numerical experiments here demonstrate again, the substantially better scaling of the RE-ABC-SMC^2 algorithm against SMC-ABC. Here, we run the adaptive version of both algorithms setting a fixed computational budget and terminating them when that has been reached. In figures 4.5 and 4.6 , we demonstrate the variance over 50 runs of each algorithm for parameters p and r respectively, where we see the significantly reduced variance of RE-ABC-SMC^2. It is the case again, as in the toy model of the previous section that for a fixed computational effort the SMC-ABC can get only get to a higher epsilon/tolerance as indicated in 4.7, the resulting posterior is a markedly worse approximation with a larger variance. It is worth noting that increasing the number of internal particles in RE-ABC-SMC^2 we achieve another important reduction in the variance of the empirical means.

Additionally, in the case of parameter r, we observe an important improvement. It is usually the case that for both the simplest ABC rejection and also to some extent ABC-SMC that the variance of the r parameter is much larger and in fact the mean is far from the true value of 0.2 in or case. Here the internal SMC sampler is providing much better estimates of the likelihood. The reason for that is twofold. Before we get into that let us reiterate and important distinction between this example and the Gaussian one. In the Gaussian case we did inference on the entirety of u-space, whereas here due to how the model is constructed and is arguably to be the case for most real world examples, only a limited number of random seed u-variables are defined and thus we are able to design a sampler on that space (here a simple RW

Metropolis). It is therefore the case that we cannot expect the optimal theoretical performance of our algorithm in such case since we are in essence going from the "perfect" RE-ABC-SMC^2 algorithm to a degenerate case, which of course if we were to use only 1 time step of the internal SMC sampler would be almost equivalent to SMC-ABC. In conclusion we expect to have an entire spectrum of performance advantages dependant on the complexity of the model/simulator we are looking at and to what degree we can define the model appropriately such that we can write down the model or the subset of u-space.

It is important here to note the changing slope of the sequence of epsilons, which implies what we theoretically assumed (and hoped) would happen: Given the fixed number of particles the SMC-ABC algorithm quickly (compared to our method) start producing a sequence of epsilon of closer and closer values, indicating an inability to effectively propose and move the samples that would generate simulations closer to the observations. In fact what we found in all runs is that if we tried to set a target epsilon similar to that of the RE-ABC-SMC^2 the population of particles in SMC-ABC would always collapse at a significantly higher epsilon. Therefore, as it is usually done in practice, the user would either use samples form a worse approximation to the target of interest, or need to substantially increase the computational effort (practically embedded in the number of particles or MCMC refreshment steps) to get a better tolerance level.

We should of course also state that there is no magic bullet. RE-ABC-SMC^2 will itself also collapse if pushed into very small tolerance levels.

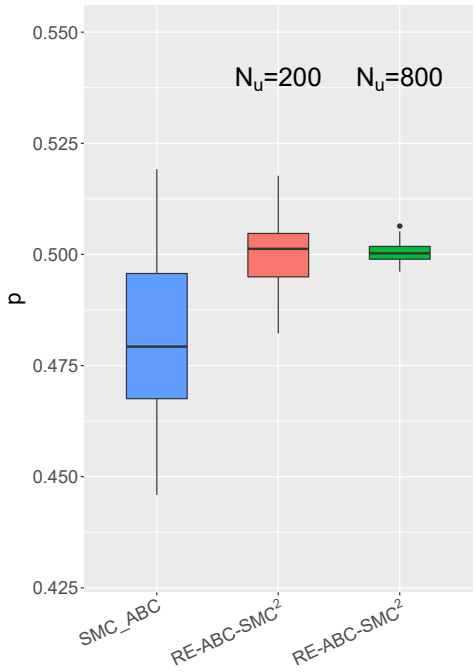

Figure 4.5: Empirical means of parameter **p** over 50 replications of each algorithm for the Duplication random graph model, and comparison between different number of internal N_u particles. The true parameter value is **0.5**. Both algorithms perform 2 steps of the MCMC refreshment step after the resampling scheme. The resampling takes place when ESS drops below half of the current number of particles (on theta space).

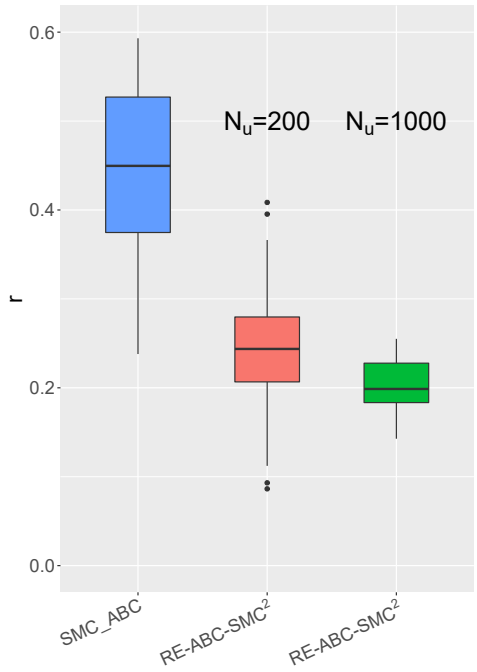

Figure 4.6: Empirical means of parameter **r** over 50 replications of each algorithm for the Duplication random graph model, and comparison between different number of internal N_u particles. The true parameter value is **0.2**. Both algorithms perform 2 steps of the MCMC refreshment step after the resampling scheme. The resampling takes place when ESS drops below half of the current number of particles (on theta space).

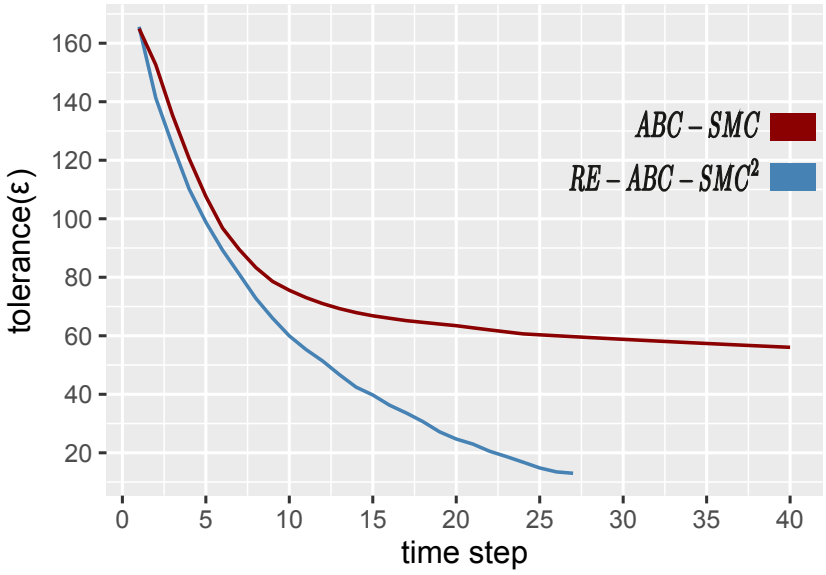

Figure 4.7: Duplication divergence random graph model tolerance over time for ABC-SMC and RE-ABC-SMC^2. The algorithms were run for a similar CPU time with the approximate number of likelihood calls being equal in order to have a computationally normalised comparison.

4.5 Conclusions

We have combined the nice properties of the rare event approach for simulating $u \mid \theta$ with the use of an SMC method for exploring θ -space, which adaptively chooses the tolerance ϵ. Compared to ABC-SMC our algorithm allows us to deal with considerably higher dimensional data (since the u-space rare event SMC algorithms tackles that directly but exploring the optimal distribution of u variables that for a given θ give the lowest ϵ. Furthermore, a longstanding issue with ABC methods is that very high state space dimensions make any ABC algorithm prohibitively expensive given the exponential scaling of the importance sampling estimator used, which in turns inhibits efficient exploration of high dimensional parameter spaces. It is of course worth keeping in mind that either in the setting of particle MCMC as in Andrieu et al. [2010], and the rare-event variant in Prangle et al. [2018] or SMC^2 Chopin et al. [2013] and correspondingly our case, the high dimensionality of the parameter space is still being tackled by the 'external' algorithm (MCMC or SMC) and thus inherits the strengths and weaknesses of each. While the 'internal' rare event sampler as explained before provides a much more efficient estimate (and bet-

ter given a fixed computational effort) of the ABC 'likelihood' to plug in. The entire benefit lies in the way the external SMC or MCMC algorithm uses this better estimate of the 'likelihood' due to a better exploration of the u-space and therefore in turn performing a better or more efficient estimate in the parameter space. Lastly, it is also obvious that poor parameter space estimates and proposals will in turn inhibit u-space exploration and reduce the benefit seen otherwise. We have shown here that due to the nature of the rare event algorithm which scales quadratically instead of exponentially Agapiou et al. [2017], Prangle et al. [2018], exploration of extremely high dimensional spaces becomes tractable. The main difficulty here lies in the fact that as we saw for example in the duplication divergence model it is up to the user to design and come up with a sampler for the internal u space. Critically, it is important that one is able to effectively define how the u variables are related to the generated observations, which is model dependent or if they are only able to be defined as a subset of the total random seeds, due to complex dependencies such as the one that inhibited how many of the total u variables we are able to infer in the network experiments. Moreover, the method has the clear benefit over RE-ABC-MCMC, inheriting all the usual benefits of SMC-ABC over MCMC-ABC. Specifically, we can adaptively choose ϵ and thus for a set tolerance level target which we set, perform the inference much more efficiently with the added advantages of Sequential Monte Carlo algorithms compared to MCMC ones. We should also add that the embarrassingly parallel nature of the internal and external SMC loops brings a significant computational efficiency benefit on top of the dimensional scaling benefits, therefore further increase the overall benefit over other methods, and last but certainly not least allow us to estimate the marginal likelihood for the model we are dealing with; an issue of critical importance for many real world applications.

Chapter 5

Conclusion and future Work

In this thesis we proposed and formulated a novel algorithm called RE-ABC-SMC^2, thereby combing the RE-MCMC of Prangle et al. [2018] and the SMC^2 of
Chopin et al. [2013] and observing that it significantly outperforms the current state of the art SMC-ABC algorithm both in adaptive and non-adaptive variants. It is clear that for the toy model and for a slice of a world application the algorithm would allow the tolerance of the ABC approximation to go lower for a given computational budget or perform the inference significantly faster for the same tolerance level. Overall, it is our hope that it would allow practitioners to perform much more efficiently their inference procedures for the models which this can be applied to. Furthermore, we validated the correctness of the proposed adaptive noisy exchange algorithm as suggested in Drovandi and Frial [2017] thus validating its use and correctness in practice, and hopefully allowing practitioners to use in any real world applications where the normalising constant cannot be evaluated as we have described in detail in the introduction to chapter 3. Last but not least, we discovered some promising results concerning the proposed SAMC-ABC algorithm by Richards and Karagiannis [2020] that we believe given more work in the future could be used as an alternative method to ABC-MCMC with significantly greater flexibility in the choice of tolerance and the robustness with respect to "sticky" behaviour of the chain inherent in the ABC-MCMC algorithm for low levels of tolerance in applications of interest.

Overall we saw how the inability to have an explicit functional form of the posterior of interest, either entirely or for the normalising constant, leads to methods for which a considerable degree of sophistication is required to not only perform the inference a practitioner would like, but also at a level where the computational budget is reasonable. From the 2 chapters devoted to the ABC methods, it is clear that while the basic original algorithm is very simple and used with success, the requirement of many real world models and scenarios require far more computational power than it would be feasible to any individual or team of researchers. As such through the years a number of improvements and integration of different methods have been proposed to more efficiently approximate the required posteriors. It would not be unreasonable to expect such significant improvements, as the ones demonstrated here for the RE-ABC-SMC^2 algorithm for example, to be made in the future. While, depending on the application the improvement may be measured in orders of magnitude, real world scenarios would still require weeks and months of runtime for this algorithm, let alone for older ones. In fact, such improvements would indeed be mandatory if better or more sophisticated models are to be employed and used in the future, in any field. Currently, models on the scale of meteorological or climatological models are completely computationally intractable if one wants to run the full model for every iteration of one of those algorithms for example. Model reduction techniques and emulators have made great improvements and would allow Monte Carlo techniques to be used efficiently but there is a price to pay for that reduction. Finally, we could argue in the same spirit for targets where the normalising constants are intractable either in the functional/approximating or computational sense. As we saw in chapter 3, a substantial computational burden is imposed in order to get a good estimate with as low variance as possible, while always taking into account the induced bias given the "noisy" nature of the algorithm. As such the proposed methodology aims to substantially reduce the computational load by adapting the number of points needed instead of calculating a large grid of them initially with no real guideline as to what density one would require for an efficient algorithm. The proof solidifies the validity and trustworthiness of the experimental data so far in Friel and Drovandi [2019] while pointing to an interesting compromise between adaptation speed and dimensionality of the parameter space.

5.1 Stochastic approximation ABC-MCMC

There is a great deal of work that can be done in order to extend all of the proposed algorithms. In the SAMC-ABC case, the obvious first improvement as alluded in the end of chapter 2 is to implement a range of post-processing in order to reduce the induced bias due to the algorithm. The importance sampling suggestion is the simplest one, yet even then not exactly clear how one would perform it in a principled way. Moreover, we can instead use a non-deterministic schedule for the adaptation, as for example in Wang and Landau [2001b],Wang and Landau [2001a], that instead decrease only when some criterion is met. This might provide a balance between bias and adaptation for the purposes of more efficient exploration. In addition, a very important and arguably significant effect on inference would be that of joint stratification of not only ϵ-space but that of the (θ, ϵ). The benefits of the WL algorithm in the cases of multimodality and strange shapes of target densities are well documented and therefore the combination of both approaches would perhaps be most beneficial. An obvious hindrance in all of these improvements, and reflecting back upon the original algorithm is the theoretical guarantees for the convergence of it to the correct target. We do know that the samples are biased, but what happens as adaptation decreases and as $N \to \infty$ is poorly understood. Some work such as Jacob and Ryder [2014] is known about convergence of the stochastic adaptation schedule is known but the assumptions and the settings are somewhat simple, such that the applications of the theoretical results in real world examples are dubious.

5.2 Adaptive Noisy Exchange

From a theoretical standpoint, we would like to be able to additionally provide rates of convergence to the degree that is possible for the algorithm as well as some perhaps weak form of the Law of Large numbers. That would provide a better grasp of the performance of the algorithm to users. This is rather important given the fact that the adaptive nature of the algorithm and its specific adaptation may result in estimates of the posterior that are perhaps

poor since for example an extremely slow convergence to equilibrium might be observed and one should remember that the phases of the algorithm as utilised in the proof are those of the adaptive noisy chain and the noisy exact chain. It might be that specific adaptation policies result in a large number of iterations where the samples acquired are from a very noisy approximation and thus should be discarded in a more principled way rather than the heuristics users often employ in standard MCMC algorithms. Furthermore it would be interesting to perhaps prove certain bounds and note the effect of epsilon on the adaptation rates and perhaps suggest both optimal epsilons as well as acceptance rates if at all possible in this context. The latter of which has seen been worked on by a number of researches on more standard MCMC algorithms.

5.3 Rare event ABC-SMC^2

A number of improvements can be made to the algorithm. First it would be very interesting to see to what extent a more principled way to propose MCMC moves in the external SMC, could be developed. As we mentioned in the numerical experiments section of chapter 4, a more robust MCMC kernel as $\epsilon \to 0$ is needed if we are to avoid collapse of the SMC algorithm. The idea proposed in Lee [2012] of some r-hit MCMC kernel, are promising as the proposal is that the simulations of *both* proposal and current values of the parameter are done according to the likelihood until an acceptance, or "hit" is observed. The author proposes variations that auxiliary data associated with the parameters are generated until either 1-hit or r-hits are observed. Given that those kernels satisfy detailed balance we think that it would be reasonable to expect that to also be satisfied in the case of the SMC^2 algorithm thus proving a solid theoretical justification for the validity of the approach. Secondly, we that the way one performs inference on u-space is important in the performance of the overall algorithm. It would be rather interesting to also compare more complicated models were the u-variables have more complicated dependencies and possibly design more efficient methods of sampling on that space. Admittedly, that would perhaps increase the computational burden substantially if the dimension is large. We envision running perhaps another

PMCMC algorithm (inside SMC^2) that targets the joint posterior of (u, θ) in a Metropolis-Gibbs scenario (and particle Gibbs as in Andrieu et al. [2010]). Furthermore, it would be extremely interesting to compare to what extent, using all the random seed variables as in the Gaussian case, versus using only a subset due to intricacies and dependencies of the considered model (as in the duplication-divergence), this affects the performance of the overall algorithm. It stands to reason that using all the u variables, is taking full advantage of the RE-SMC part of the algorithm and therefore is providing the maximum effectiveness overall. Yet, as we saw in the second example of the network model a substantial improvement of SMC is still observed, thus one could imagine that incorporating and an even greater number of seed variables would perhaps result in even further improvements with regards to variance of empirical estimates, decrease in epsilon speed and overall true posterior accuracy.

Bibliography

Robert A Adams and John Fournier. *Sobolev spaces*. Academic Press, Boston, MA, 2nd edition, 2003. ISBN 978-0-12-044143-3.

S Agapiou, O Papaspiliopoulos, D. Sanz-Alonso, and A M Stuart. Importance Sampling: Computational Complexity and Intrinsic Dimension. *Statistical Science*, 32(3):405–431, 2017. URL https://www.jstor.org/stable/26408299.

P. Alquier, N. Friel, R. Everitt, and A. Boland. Noisy Monte Carlo: convergence of Markov chains with approximate transition kernels. *Statistics and Computing*, 26(1-2):29–47, 2016. ISSN 15731375. doi:10.1007/s11222-014-9521-x.

Donald W.K. Andrews. Generic uniform convergence. *Econometric Theory*, 8 (2):241–257, 1992. ISSN 14694360. doi:10.1017/S0266466600012780.

C. Andrieu, A Doucet, and A Lee. Contribution to the discussion of "Constructing summary statistics for approximate Bayesian computation: semi-automatic approximate Bayesian computation" by Fearnhead and Prangle. *Journal of the Royal Statistical Society Series B*, (74(3)):451–452.

Christophe Andrieu and Yves F. Atchadé. On the efficiency of adaptive MCMC algorithms. *Electronic Communications in Probability*, 12:336–349, 2007. ISSN 1083589X. doi:10.1214/ECP.v12-1320.

Christophe Andrieu and Éric Moulines. On the ergodicity properties of some adaptive MCMC algorithms. *Annals of Applied Probability*, 16(3):1462–1505, 2006. ISSN 10505164. doi:10.1214/105051606000000286.

Christophe Andrieu and Gareth O. Roberts. The pseudo-marginal approach for efficient Monte Carlo computations. *Annals of Statistics*, 37(2):697–725, 2009. ISSN 00905364. doi:10.1214/07-AOS574.

Christophe Andrieu and Matti Vihola. Convergence properties of pseudo-marginal Markov chain Monte Carlo algorithms. *Annals of Applied Probability*, 25(2):1030–1077, 2015. ISSN 10505164. doi:10.1214/14-AAP1022.

Christophe Andrieu and Matti Vihola. Establishing some order amongst exact approximations of MCMCs. *Annals of Applied Probability*, 2016. ISSN 10505164. doi:10.1214/15-AAP1158.

Christophe Andrieu, Arnaud Doucet, and Roman Holenstein. Particle Markov Chain Monte Carlo. *Journal of the Royal Statistical Society Series B*, 72(2): 1–33, 2010.

Yves Atchadé, Gersende Fort, Eric Moulines, and Pierre Priouret. Adaptive Markov chain Monte Carlo: Theory and methods. In *Cambridge University Press*. 2011. doi:10.1017/CBO9780511984679.003.

N.S. Bakhvalov. On approximate computation of integrals. *Vestnik Moskov. Gos. Univ. Ser. Math. Mech. Astron. Phys. Chem*, 4:3–18, 1959.

N.S. Bakhvalov. On the rate of convergence of indeterministic integration processes within the functional classes W. *Theory Prob. Applications,*, 7: 227, 1962. doi:https://doi.org/10.1049/ip-rsn:19990255.

AA Barker. Monte Carlo Calculations of the Radial Distribution Functions for a Proton-Electron Plasma. *Australian Journal of Physics*, 1965. ISSN 0004-9506. doi:10.1071/ph650119.

Mark A. Beaumont. Estimation of population growth or decline in genetically monitored populations. *Genetics*, 164(3):1139–1160, 2003. ISSN 00166731.

Mark A. Beaumont. Approximate Bayesian Computation in Evolution and Ecology. *Annual Review of Ecology, Evolution, and Systematics*, 41:379–406, 2010. ISSN 1543-592X. doi:10.1146/annurev-ecolsys-102209-144621.

J. Beltrán and C. Landim. A Martingale approach to metastability. *Probability Theory and Related Fields*, 161:267–307, 2015. ISSN 14322064. doi:10.1007/s00440-014-0549-9.

Alexandros Beskos, Dan Crisan, and Ajay Jasra. On the stability of sequential Monte Carlo methods in high dimensions. *Annals of Applied Probability*, 24 (4):1396–1445, 2014a. ISSN 10505164. doi:10.1214/13-AAP951.

Alexandros Beskos, Dan O. Crisan, Ajay Jasra, and Nick Whiteley. Error bounds and normalising constants for sequential Monte Carlo samplers in high dimensions. *Advances in Applied Probability*, 46(1):279–306, 2014b. ISSN 00018678. doi:10.1239/aap/1396360114.

Gérard Biau, Frédéric Cérou, and Arnaud Guyader. New insights into approximate Bayesian computation. *Annales de l'institut Henri Poincare (B) Probability and Statistics*, 2015. ISSN 02460203. doi:10.1214/13-AIHP590.

Hermine Biermé and Anne Estrade. Covering the whole space with poisson random balls. *Lat. Am. J. Probab. Math. Stat*, 9:213–229, 2012. ISSN 19800436.

Aidan Boland, Nial Friel, and Florian Maire. Efficient MCMC for gibbs random fields using pre-computation. *Electronic Journal of Statistics*, 12(2):4138–4179, 2018. ISSN 19357524. doi:10.1214/18-EJS1504.

Anton Bovier, Michael Eckhoff, Véronique Gayrard, and Markus Klein. Metastability in stochastic dynamics of disordered mean-field models. *Probability Theory and Related Fields*, 119:99–161, 2001. ISSN 01788051. doi:10.1007/PL00012740.

Anton Bovier, Michael Eckhoff, Véronique Gayrard, and Markus Klein. Metastability and low lying spectra in reversible Markov chains. *Communications in Mathematical Physics*, 228:219–255, 2002. ISSN 00103616. doi:10.1007/s002200200609.

George E. P. Box and George C Tiao. *Bayesian Inference in Statistical Analysis*. Addison Wiley Pub Co, 1973.

R. J. Boys, D. J. Wilkinson, and T. B.L. Kirkwood. Bayesian inference for a discretely observed stochastic kinetic model. *Statistics and Computing*, 18: 125–135, 2008. ISSN 09603174. doi:10.1007/s11222-007-9043-x.

Paul A. Brasnett, Lyudmila Mihaylova, Nishan Canagarajah, and David Bull. Particle filtering with multiple cues for object tracking in video se-

quences. In *Image and Video Communications and Processing 2005*, 2005. doi:10.1117/12.585882.

Alberto Caimo and Nial Friel. Bayesian inference for exponential random graph models. *Social Networks*, 33(1):41–55, 2011. ISSN 03788733. doi:10.1016/j.socnet.2010.09.004.

O Cappe, E Moulines, and T Ryden. *Inference in Hidden Markov Models*. Springer New York, New York, 2006. ISBN 9780387289823. doi:https://doi.org/10.1007/0-387-28982-8.

Carpenter J, P Clifford, and P Fearnhead. An improved particle filter for nonlinear problems. *EEE Proceedings on Radar, Sonar and Navigation*, 146: 2–7, 1999. doi:https://doi.org/10.1049/ip-rsn:19990255.

Marzio Cassandro, Antonio Galves, Enzo Olivieri, and Maria Eulália Vares. Metastable behavior of stochastic dynamics: A pathwise approach. *Journal of Statistical Physics*, 35:603–634, 1984. ISSN 00224715. doi:10.1007/BF01010826.

F. Cérou, P. Del Moral, T. Furon, and A. Guyader. Sequential Monte Carlo for rare event estimation. *Statistics and Computing*, 22:795–808, 2012. ISSN 09603174. doi:10.1007/s11222-011-9231-6.

N. Chopin, P. E. Jacob, and O. Papaspiliopoulos. SMC2: An efficient algorithm for sequential analysis of state space models. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 75(3):397–426, 2013. ISSN 13697412. doi:10.1111/j.1467-9868.2012.01046.x.

Nicolas Chopin. A sequential particle filter method for static models. *Biometrika*, 89(3):539–551, 2002. ISSN 00063444. doi:10.1093/biomet/89.3.539.

Nicolas Chopin. Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference. *Annals of Statistics*, 32(6):2385–2411, 2004. ISSN 00905364. doi:10.1214/009053604000000698.

Thomas M. Cover. Estimation by the Nearest Neighbor Rule. *IEEE Transactions on Information Theory*, 14(1):50–55, 1968. ISSN 15579654. doi:10.1109/TIT.1968.1054098.

Lionel Cucala, Jean Michel Marin, Christian P. Robert, and D. M. Titterington. A Bayesian reassessment of nearest-neighbor classification. *Journal of the American Statistical Association*, 104(485):263–273, 2009. ISSN 01621459. doi:10.1198/jasa.2009.0125.

A. G. Cunha Netto, C. J. Silva, A. A. Caparica, and R. Dickman. Wang-Landau sampling in three-dimensional polymers. *Brazilian Journal of Physics*, 36(3A):619–622, 2006. ISSN 01039733. doi:10.1590/S0103-97332006000500005.

Bruno de Finetti. Theory of Probability. Wiley, New York:, 1974.

Thomas A. Dean, Sumeetpal S. Singh, Ajay Jasra, and Gareth W. Peters. Parameter estimation for hidden Markov models with intractable likelihoods. *Scandinavian Journal of Statistics*, 41(4):970–987, 2014. ISSN 14679469. doi:10.1111/sjos.12077.

P. Del Moral and A. Guionnet. Central limit theorem for nonlinear filtering and interacting particle systems. *Annals of Applied Probability*, 9(2):275–297, 1999. ISSN 10505164. doi:10.1214/aoap/1029962742.

P. Del Moral and L. Miclo. Branching and interacting particle systems approximations of Feynman-Kac formulae with applications to non-linear filtering. In J. Azéma, M Ledoux, M Émery, and M Yor, editors, *Séminaire de Probabilités XXXIV. Lecture Notes in Mathematics, vol 1729*, chapter 1, pages 1–145. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000. doi:10.1007/bfb0103798.

P. Del Moral, A. Doucet, and G. W. Peters. Sharp propagation of chaos estimates for Feynman-Kac particle models. *Theory of Probability and its Applications*, 128(1):332–353, 2007. ISSN 0040585X. doi:10.1137/S0040585X97982529.

Pierre Del Moral. *Feynman-Kac formulae: genealogical and interacting particle systems with applications*. Springer Verlag, New York, 2004. doi:https://doi.org/10.1007/978-1-4684-9393-1.

Pierre Del Moral and Ajay Jasra. Sequential Monte Carlo for Bayesian Computation. *Bayesian Statistics*, 8:1–34, 2007.

Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 68(3):411–436, 2006. ISSN 13697412. doi:10.1111/j.1467-9868.2006.00553.x.

Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. An adaptive sequential Monte Carlo method for approximate Bayesian computation. *Statistics and Computing*, 22(5):1009–1020, 2012. ISSN 09603174. doi:10.1007/s11222-011-9271-y.

A. Doucet, M. K. Pitt, G. Deligiannidis, and R. Kohn. Efficient implementation of Markov chain Monte Carlo when using an unbiased likelihood estimator. *Biometrika*, 102(2):295–313, 2015. ISSN 14643510. doi:10.1093/biomet/asu075.

Arnaud Doucet and Adam Johansen. A tutorial on particle filtering and smoothing: fifteen years later. In *OXFORD Handb. NONLINEAR Filter.*, pages 656–704, 2011. URL citeulike-article-id:9086845{%}0Ahttp://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.157.772.

Arnaud D Doucet, N de Freitas, and N Gordon. *Sequential Monte Carlo Methods in Practice*. Springer Verlag, 2001. ISBN 978-0-387-95146-1. doi:https://doi.org/10.1007/978-1-4757-3437-9.

C Christopher Drovandi and Nial Frial. An Adaptive Noisy MCMC Algorithm for Doubly Intractable Models (personal communication). 2017.

Richard G. Everitt. Bayesian parameter estimation for latent markov random fields and social networks. *Journal of Computational and Graphical Statistics*, 21(4):940–960, 2012. ISSN 15372715. doi:10.1080/10618600.2012.687493.

Paul Fearnhead and Dennis Prangle. Constructing summary statistics for approximate Bayesian computation: Semi-automatic approximate Bayesian computation. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 74(3):419–474, 2012. ISSN 13697412. doi:10.1111/j.1467-9868.2011.01010.x.

Paul Fearnhead, Omiros Papaspiliopoulos, Gareth O. Roberts, and Andrew Stuart. Random-weight particle filtering of continuous time processes. *Jour-*

nal of the Royal Statistical Society. Series B: Statistical Methodology, 72(4): 497–512, 2010. ISSN 13697412. doi:10.1111/j.1467-9868.2010.00744.x.

Evelyn Fix and J. L. Hodges. Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties. *International Statistical Review / Revue Internationale de Statistique*, 1989. ISSN 03067734. doi:10.2307/1403797.

James M. Flegal and Galin L. Jones. Batch means and spectral variance estimators in Markov chain Monte Carlo. *Annals of Statistics*, 38(2):1034–1070, 2010. ISSN 00905364. doi:10.1214/09-AOS735.

James M. Flegal, Murali Haran, and Galin L. Jones. Markov Chain Monte Carlo: Can We Trust the Third Significant Figure? *Statistical Science*, 23 (2):250–260, 2008. ISSN 0883-4237. doi:10.1214/08-sts257.

G. Fort, E. Moulines, P. Priouret, and P. Vandekerkhove. A central limit theorem for adaptive and interacting Markov chains. *Bernoulli*, 20(2):457–485, 2014. ISSN 13507265. doi:10.3150/12-BEJ493.

M. Freidlin and L. Koralov. Metastable Distributions of Markov Chains with Rare Transitions. *Journal of Statistical Physics*, 167:pages 1355–1375, 2017. ISSN 00224715. doi:10.1007/s10955-017-1777-z.

Nial Friel and C Christopher Drovandi. Adaptive noisy exchange algorithm. Technical report, 2019.

Charles J Geyer. Markov Chain Monte Carlo Maximum Likelihood, Computing Science and Statistics. *Proc. of the 23rd Symposium Interface, 1991*, 1991. URL https://hdl.handle.net/11299/58440.

Charles J Geyer. Markov Chain Monte Carlo Lecture Notes, 2005. URL https://www.stat.umn.edu/geyer/f05/8931/n1998.pdf.

Daniel T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *Journal of Physical Chemistry*, 81(25):2340–2361, 1977. ISSN 00223654. doi:10.1021/j100540a008.

A. Golightly and D. J. Wilkinson. Bayesian inference for stochastic kinetic models using a diffusion approximation. *Biometrics*, 61(3):781–788, 2005. ISSN 0006341X. doi:10.1111/j.1541-0420.2005.00345.x.

Andrew Golightly, Daniel A. Henderson, and Chris Sherlock. Delayed acceptance particle MCMC for exact inference in stochastic kinetic models. *Statistics and Computing*, 2015. ISSN 15731375. doi:10.1007/s11222-014-9469-x.

N. J. Gordon, D. J. Salmond, and A. F.M. Smith. Novel approach to nonlinear/non-gaussian Bayesian state estimation. *IEE Proceedings, Part F: Radar and Signal Processing*, 140(2):107–113, 1993. ISSN 0956375X. doi:10.1049/ip-f-2.1993.0015.

Matthew M. Graham and Amos J. Storkey. Asymptotically exact inference in differentiable generative models. *Electronic Journal of Statistics*, 11(2): 5105–5164, 2017. ISSN 19357524. doi:10.1214/17-EJS1340SI.

Peter J. Green. Reversible jump Markov chain monte carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1995. ISSN 00063444. doi:10.1093/biomet/82.4.711.

Volker Grimm and Steven F. Railsback. *Individual-based Modeling and Ecology*. Princeton University Press, 2005. ISBN 069109666X. doi:10.1111/j.1467-2979.2008.00286.x. URL http://doi.wiley.com/10.1111/j.1467-2979.2008.00286.x.

Fredrik Gustafsson, Fredrik Gunnarsson, Niclas Bergman, Urban Forssell, Jonas Jansson, Rickard Karlsson, and Per Johan Nordlund. Particle filters for positioning, navigation, and tracking. *IEEE Transactions on Signal Processing*, 50(2):425–437, 2002. ISSN 1053587X. doi:10.1109/78.978396.

W.K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Source: Biometrika Biometrika*, 57(1):97–109, 1970. ISSN 0006-3444. doi:10.1093/biomet/57.1.97. URL http://www.jstor.org/stable/2334940{%}5Cnhttp://www.jstor.org/page/info/about/policies/terms.jsp{%}5Cnhttp://www.jstor.org.

Stefan Heinrich and Erich Novak. Optimal Summation and Integration by Deterministic, Randomized, and Quantum Algorithms. In KT. Fang, H. Niederreiter, and F.J. Hickernell, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2000*. Springer, Berlin, Heidelberg, 2002. doi:10.1007/978-3-642-56046-0_4.

R Holestein. *Particle Markov Chain Monte Carlo*. Phd, University of British Columbia, 2009.

Pierre E. Jacob and Robin J. Ryder. The Wang-Landau algorithm reaches the flat histogram criterion in finite time. *Annals of Applied Probability*, 24(1): 34–53, 2014. ISSN 10505164. doi:10.1214/12-AAP913.

Galin L. Jones. On the Markov chain central limit theorem. *Probability Surveys*, 1(0):299–320, 2004. ISSN 1549-5787. doi:10.1214/154957804100000051. URL http://projecteuclid.org/euclid.ps/1104335301.

Galin L. Jones, Murali Haran, Brian S. Caffo, and Ronald Neath. Fixed-width output analysis for Markov chain Monte Carlo. *Journal of the American Statistical Association*, 101(476):1537–1547, 2006. ISSN 01621459. doi:10.1198/016214506000000492.

Genshiro Kitagawa. Non-gaussian stateâspace modeling of nonstationary time series. *Journal of the American Statistical Association*, 82(400):1032–1041, 1987. ISSN 1537274X. doi:10.1080/01621459.1987.10478534.

Claudio Landim. Metastable Markov chains. *Probability Surveys*, 16:143–227, 2019. ISSN 15495787. doi:10.1214/18-PS310.

Pierre L'Ecuyer, Valérie Demers, and Bruno Tuffin. Rare events, splitting, and quasi-Monte Carlo. *ACM Transactions on Modeling and Computer Simulation*, 17(2):9–52, 2007. ISSN 10493301. doi:10.1145/1225275.1225280.

Anthony Lee. On the choice of MCMC kernels for approximate Bayesian computation with SMC samplers. In *Proceedings - Winter Simulation Conference*. Institute of Electrical and Electronics Engineers, 2012. ISBN 9781467347792. doi:10.1109/WSC.2012.6465212.

Faming Liang. A generalized Wang-Landau algorithm for Monte Carlo computation. *Journal of the American Statistical Association*, 100(472):1311–1327, 2005. ISSN 01621459. doi:10.1198/016214505000000259.

Faming Liang. Annealing stochastic approximation Monte Carlo algorithm for neural network training. *Machine Learning*, 68(3):201–233, 2007. ISSN 08856125. doi:10.1007/s10994-007-5017-7.

Faming Liang. On the use of stochastic approximation Monte Carlo for Monte Carlo integration. *Statistics and Probability Letters*, 79(5):581–587, 2009. ISSN 01677152. doi:10.1016/j.spl.2008.10.007.

Faming Liang. A double Metropolis-Hastings sampler for spatial models with intractable normalizing constants. *Journal of Statistical Computation and Simulation*, 80(9):1007–1022, 2010. ISSN 00949655. doi:10.1080/00949650902882162.

Faming Liang. An Overview of Stochastic Approximation Monte Carlo. *Wiley Interdisciplinary Reviews: Computational Statistics*, 6(4):240–254, 2014. ISSN 19390068. doi:10.1002/wics.1305.

Faming Liang, Chuanhai Liu Liu, and Raymond J. Carroll. Stochastic approximation in Monte Carlo computation. *Journal of the American Statistical Association*, 102(477):305–320, 2007. ISSN 01621459. doi:10.1198/016214506000001202.

Dennis Lindley. *Introduction to Probability and Statistics from a Bayesian Viewpoint*. Cambridge University Press, 1965. ISBN 9780511662973. doi:https://doi.org/10.1017/CBO9780511662973.

Jun S. Liu and Rong Chen. Sequential monte carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93(443):1032–1044, 1998. ISSN 1537274X. doi:10.1080/01621459.1998.10473765.

D. O. Loftsgaarden and C. P. Quesenberry. A Nonparametric Estimate of a Multivariate Density Function. *The Annals of Mathematical Statistics*, 36 (3):1049–1051, 1965. ISSN 0003-4851. doi:10.1214/aoms/1177700079.

A. Malakis, P. Kalozoumis, and N. Tyraskis. Monte Carlo studies of the square Ising model with next-nearest-neighbor interactions. *European Physical Journal B*, 50:63–67, 2006. ISSN 14346028. doi:10.1140/epjb/e2006-00032-2.

Jean Michel Marin, Natesh S. Pillai, Christian P. Robert, and Judith Rousseau. Relevant statistics for Bayesian model choice. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 76(5):833–859, 2014. ISSN 14679868. doi:10.1111/rssb.12056.

E. Marinari and G. Parisi. Simulated tempering: A New Monte Carlo Scheme. *EPL*, 19:451, 1992. ISSN 12864854. doi:10.1209/0295-5075/19/6/002.

K. L. Mengersen and R. L. Tweedie. Rates of convergence of the Hastings and Metropolis algorithms. *Annals of Statistics*, 24(1):101–121, 1996. ISSN 00905364. doi:10.1214/aos/1033066201.

N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953a. ISSN 0021-9606. doi:10.1063/1.1699114. URL http://aip.scitation.org/doi/10.1063/1.1699114.

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953b. ISSN 00219606. doi:10.1063/1.1699114.

S Meyn and R. L. Tweedie. *Markov Chains and Stochastic Stability*. Cambridge University Press,, New York, NY, USA, 2nd edition, 2009. doi:https://doi.org/10.1017/CBO9780511626630.

Błazej Miasojedow, Eric Moulines, and Matti Vihola. An adaptive parallel tempering algorithm. *Journal of Computational and Graphical Statistics*, 22 (3):649–664, 2013. ISSN 10618600. doi:10.1080/10618600.2013.778779.

Jesper Møller, Anthony N Pettitt, Kasper K Berthelsen, and Robert W Reeves. An efficient Markov chain Monte Carlo method for distributions with intractable normalising constants. *Biometrika*, 93(2):451–458, 2004. URL https://watermark.silverchair.com/932451.pdf?token=AQECAHi208BE49Ooan9kkhW{_}Ercy7Dm3ZL{_}9Cf3qfKAc485ysgAAAckwggHFBgkqhkiG9w0BBwa

Ian Murray, Zoubin Ghahramani, and David J. C. MacKay. MCMC for doubly-intractable distributions. *UAI*, pages 359–366, 2006.

Peter Neal and Gareth Roberts. Optimal scaling for partially updating MCMC algorithms. *Annals of Applied Probability*, 16(2):475–515, 2006. ISSN 10505164. doi:10.1214/105051605000000791.

Radford M. Neal. Sampling from multimodal distributions using tempered transitions. *Statistics and Computing*, 6:353–366, 1996. ISSN 09603174. doi:10.1007/BF00143556.

Radford M. Neal. Annealed importance sampling. *Statistics and Computing*, 11(2):125–139, 2001. ISSN 09603174. doi:10.1023/A:1008923215028.

Radford M. Neal. MCMC using hamiltonian dynamics. In *Handbook of Markov Chain Monte Carlo*, page 50. Chapman and Hall/CRC, 1st edition, 2011. ISBN 9781420079425. doi:10.1201/b10905-6.

E Olivieri and M.E Vares. *Large deviations and metastability*. Cambridge University Press, Cambridge, 2005. doi:https://doi.org/10.1017/CBO9780511543272.

Emilia Pompe, Chris Holmes, and Krzysztof Łatuszyński. A framework for adaptive mcmc targeting multimodal distributions. *Annals of Statistics*, 48 (5):2930 – 2952, 2020. ISSN 21688966. doi:10.1214/19-AOS1916.

Dennis Prangle, Richard G. Everitt, and Theodore Kypraios. A rare event approach to high-dimensional approximate Bayesian computation. *Statistics and Computing*, 28(4):819–834, 2018. ISSN 15731375. doi:10.1007/s11222-017-9764-4.

Jonathan K. Pritchard, Mark T. Seielstad, Anna Perez-Lezaun, and Marcus W. Feldman. Population growth of human Y chromosomes: A study of y chromosome microsatellites. *Molecular Biology and Evolution*, 16(12):1791–1798, 1999. ISSN 07374038. doi:10.1093/oxfordjournals.molbev.a026091.

James Gary Propp and David Bruce Wilson. Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures and Algorithms*, 9(1):223–252, 1996. ISSN 10429832. doi:10.1002/(SICI)1098-2418(199608/09)9:1/2<223::AID-RSA14>3.0.CO;2-O.

Kieran Richards and Georgios Karagiannis. Likelihood Free Stochastic Approximation Monte Carlo. Technical report, Department of Mathematical Sciences, Durham University, 2020.

Christian P. Robert and George Casella. *Monte Carlo Statistical Methods*. Springer Science+Business Media, 2 edition, 2004. doi:https://doi.org/10.1007/978-1-4757-4145-2.

Christian P. Robert, Jean Marie Cornuet, Jean Michel Marin, and Natesh S. Pillai. Lack of confidence in approximate Bayesian computation model choice. *Proceedings of the National Academy of Sciences of the United States of America*, 108(37):15112–15117, 2011. ISSN 00278424. doi:10.1073/pnas.1102900108.

Gareth O. Roberts and Jeffrey S. Rosenthal. Optimal Scaling for Various Metropolis-Hastings Algorithms. *Statistical Science*, 16(4):351–367, 2001. ISSN 08834237. doi:10.1214/ss/1015346320.

Gareth O. Roberts and Jeffrey S. Rosenthal. General state space Markov chains and MCMC algorithms. *Probability Surveys*, 1(0):20–71, 2004. ISSN 1549-5787. doi:10.1214/154957804100000024. URL http://projecteuclid.org/euclid.ps/1099928648.

Gareth O Roberts and Jeffrey S Rosenthal. Coupling and Ergodicity of Adaptive MCMC. *Journal of Applied Probability*, 44(2):458–475, 2007.

Gareth O Roberts and Jeffrey S Rosenthal. Examples of Adaptive MCMC Examples of Adaptive MCMC. *Journal of Computational and Graphical Statistics*, 86(March):349–367, 2016. ISSN 1537-2715. doi:10.1198/jcgs.2009.06134. URL http://www.tandfonline.com/action/journalInformation?journalCode=ucgs20.

Gareth O. Roberts, Jeffrey S. Rosenthal, and Peter O. Schwartz. Convergence properties of perturbed Markov chains. *Journal of Applied Probability*, 35 (1):1–11, 1998. ISSN 00219002. doi:10.1239/jap/1032192546.

Donald B. Rubin. The Calculation of Posterior Distributions by Data Augmentation: Comment: A Noniterative Sampling/Importance Resampling Alternative to the Data Augmentation Algorithm for Creating a Few Imputations When Fractions of Missing Information Are Modest: The SIR. *Journal of the American Statistical Association*, 82(398):528–540, 1987. ISSN 01621459. doi:10.2307/2289460.

Leonard J. Savage. *The foundations of statistics*. John Wiley & Sons, Inc., 1954.

Chris Sherlock, Alexandre H. Thiery, Gareth O. Roberts, and Jeffrey S. Rosenthal. On the efficiency of pseudo-marginal random walk metropolis algorithms. *Annals of Statistics*, 43(1):238–275, 2015. ISSN 00905364. doi:10.1214/14-AOS1278.

C. J. Silva, A. A. Caparica, and J. A. Plascak. Wang-Landau Monte Carlo simulation of the Blume-Capel model. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 73(3):036702, 2006. ISSN 15393755. doi:10.1103/PhysRevE.73.036702.

S A Sisson, Y Fan, and Mark M Tanaka. Sequential Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences of the United States of America*, 104(6):1760–5, 2007. ISSN 0027-8424. doi:10.1073/pnas.0607208104. URL http://www.ncbi.nlm.nih.gov/pubmed/17264216{%}5Cnhttp://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC1794282.

a F M Smith and a E Gelfand. Bayesian Statistics without Tears. *The American Statistician*, 46(2):84–88, 1992. ISSN 00031305. doi:10.1080/00031305.1992.10475856. URL http://www.jstor.org/stable/2684170.

L. F. South, A. N. Pettitt, and C. C. Drovandi. Sequential Monte Carlo samplers with independent Markov chain Monte Carlo proposals. *Bayesian Analysis*, 14(3):753–776, 2019. ISSN 19316690. doi:10.1214/18-BA1129.

Simon Tavaré, David J. Balding, R. C. Griffiths, and Peter Donnelly. Inferring coalescence times from DNA sequence data. *Genetics*, 145(2):505–518, 1997. ISSN 00166731. doi:https://doi.org/10.1093/genetics/145.2.505.

Thomas Thorne and Michael P.H. Stumpf. Graph spectral analysis of protein interaction network evolution. *Journal of the Royal Society Interface*, 9(75): 2653–2666, 2012. ISSN 17425662. doi:10.1098/rsif.2012.0220.

Luke Tierney. A note on Metropolis-Hastings kernels for general state spaces. *Annals of Applied Probability*, 8(1):1–9, 1998. ISSN 10505164. doi:10.1214/aoap/1027961031.

Tina Toni, David Welch, Natalja Strelkowa, Andreas Ipsen, and Michael P.H. Stumpf. Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems. *Journal of the Royal Society Interface*, 6(31):187–202, 2009. ISSN 17425662. doi:10.1098/rsif.2008.0172.

Elske van der Vaart, Mark A. Beaumont, Alice S A Johnston, and Richard M. Sibly. Calibration and evaluation of individual-based models using Approximate Bayesian Computation. *Ecological Modelling*, 312:182–190, 2015. ISSN 03043800. doi:10.1016/j.ecolmodel.2015.05.020.

Elske van der Vaart, Alice S.A. Johnston, and Richard M. Sibly. Predicting how many animals will be where: How to build, calibrate and evaluate individual-based models. *Ecological Modelling*, 326:113–123, 2016. ISSN 03043800. doi:10.1016/j.ecolmodel.2015.08.012.

Dootika Vats, James M Flegal, and Galin L Jones. Multivariate output analysis for Markov chain Monte Carlo. *Biometrika*, 106(2):321–337, 2019. ISSN 0006-3444. doi:10.1093/biomet/asz002. URL `https://academic.oup.com/biomet/article/106/2/321/5426969`.

Christelle Vergé, Cyrille Dubarry, Pierre Del Moral, and Eric Moulines. On parallel implementation of sequential Monte Carlo methods: the island particle model. *Statistics and Computing*, 25:243–260, 2015. ISSN 15731375. doi:10.1007/s11222-013-9429-x.

Fugao Wang and D. P. Landau. Efficient, multiple-range random walk algorithm to calculate the density of states. *Physical Review Letters*, 86(10): 2050–2053, 2001a. ISSN 00319007. doi:10.1103/PhysRevLett.86.2050.

Fugao Wang and D. P. Landau. Determining the density of states for classical statistical models: A random walk algorithm to produce a flat histogram. *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, 64(5):16, 2001b. ISSN 1063651X. doi:10.1103/PhysRevE.64.056101.

D. J. Wilkinson. *Stochastic Modelling for Systems Biology*. Chapman & Hall/CRC Press., 2011.

Richard David Wilkinson. Approximate Bayesian computation (ABC) gives exact results under the assumption of model error. *Statistical Applications*

in Genetics and Molecular Biology, 12(2):129–141, 2013a. ISSN 15446115. doi:10.1515/sagmb-2013-0010.

Richard David Wilkinson. Approximate Bayesian computation (ABC) gives exact results under the assumption of model error. *Statistical Applications in Genetics and Molecular Biology*, 12(2):129–141, 2013b. ISSN 15446115. doi:10.1515/sagmb-2013-0010.

Jun Yang, Gareth O. Roberts, and Jeffrey S. Rosenthal. Optimal scaling of random-walk metropolis algorithms on general target distributions. *Stochastic Processes and their Applications*, 130(10):6094–6132, 2020. ISSN 03044149. doi:10.1016/j.spa.2020.05.004.

Yan Zhou, Adam M. Johansen, and John A.D. Aston. Toward Automatic Model Comparison: An Adaptive Sequential Monte Carlo Approach. *Journal of Computational and Graphical Statistics*, 25(3):701–726, 2016. ISSN 15372715. doi:10.1080/10618600.2015.1060885.