Some New Results in Sequential Monte Carlo

A thesis presented for the degree of

Doctor of Philosophy of Imperial College

by

James Stewart Martin

Department of Mathematics

Imperial College

180 Queen's Gate, London SW7 2BZ

JULY 13, 2012

I certify that this thesis, and the research to which it refers, are the product of my own work, and that any ideas or quotations from the work of other people, published or otherwise, are fully acknowledged in accordance with the standard referencing practices of the discipline.

Signed:

Copyright

Abstract

Sequential Monte Carlo (SMC) methods have been well studied within the context of performing inference with respect to partially observed Markov processes, and their use in this context relies upon the ability to evaluate or estimate the likelihood of a set of observed data, given the state of the latent process. In many real-world applications such as the study of population genetics and econometrics, however, this likelihood can neither be analytically evaluated nor replaced by an unbiased estimator, and so the application of exact SMC methods to these problems may be infeasible, or even impossible. The models in many of these applications are complex, yet realistic, and so development of techniques that can deal with problems of likelihood intractability can help us to perform inference for many important yet otherwise inaccessible problems; this motivates the research presented within this thesis.

The main focus of this work is the application of approximate Bayesian computation (ABC) methodology to state-space models (SSMs) and the development of SMC methods in the context of these ABC SSMs for filtering and smoothing of the latent process. The introduction of ABC here avoids the need to evaluate the likelihood, at the cost of introducing a bias into the resulting filtering and smoothing estimators; this bias is explored theoretically and through simulation studies. An alternative SMC procedure, incorporating an additional rejection step, is also considered and the novel application of this rejection-based SMC procedure to the ABC approximation of the SSM is considered.

This thesis will also consider the application of MCMC and SMC methods to a class of partially observed point process (PP) models. We investigate the problem of performing sequential inference for these models and note that current methods often fail. We present a new approach to smoothing in this context, using SMC samplers (Del Moral et al., 2006). This approach is illustrated, with some theoretical discussion, on a doubly stochastic PP applied in the context of finance.

Acknowledgements

First and foremost, I would like to express my sincere thanks to my supervisors, Dr. Emma McCoy and Dr. Ajay Jasra, for their provision of time, expertise and advice. I am also grateful to both the Engineering and Physical Sciences Research Council (EPSRC), whose funding has made this work possible, and the Department of Statistics and Applied Probability at the National University of Singapore, who provided me with office space and resources whilst I visited Dr. Jasra in January 2012.

Thanks also go to the many faces at IC who have made eight years worthwhile: primarily to Jov and Roxy, and to Richie, for countless Thursday evenings and the inevitable Friday mornings; to everyone in Stats, in particular to Ed, Brian and Anna; and to everyone in ICURFC and the '22 Club for providing those aspects of university life that complement the academic side so well.

Finally, my greatest thanks go to my family, for their unending support and encouragement, as well as their unwavering belief that I can achieve whatever I set out to do; this completed thesis is truly a testament to their support.

James S. Martin

Table of contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Preamble

Sequential Monte Carlo (SMC) methods are a broad class of simulation-based techniques for performing statistical inference with respect to a sequence of related probability distributions. The study of SMC methodology has become one of the most important areas of research in the fields of statistics and engineering, and it remains a fertile area of study today. Wide ranging applications have increased the attention that these methods have received, across a variety of disciplines; examples of areas of application include signal processing (e.g. Vermaak et al., 2002) and econometrics (e.g. Del Moral et al., 2007).

This thesis will consider the application of SMC and Markov chain Monte Carlo (MCMC) methods to the class of discrete-time state space models (SSMs) specified by the joint Markov chain $\{X_t, Y_t\}_{t \geq 1}$, in which the state process $\{X_t\}_{t \geq 1}$ can only be observed indirectly through the dependent data process $\{Y_t\}_{t \geq 1}$. The majority of this thesis will focus on a generalised specification of the SSM, however we will also consider a specific class of SSMs, in which both $\{X_t\}_{t \geq 1}$ and $\{Y_t\}_{t \geq 1}$ are point processes. SSMs are also known as hidden Markov models (HMMs), and they provide a versatile framework in which to study many problems of interest, such as single- or multi-target tracking (e.g Gordon et al., 1993; Bar-Shalom et al., 2001) and stochastic volatility estimation (e.g. Barndorff-Nielsen and Shephard, 2001; Andrieu et al., 2010).

We describe first the generalised SSM, which is specified completely through the evolution of $\{X_t\}_{t\geq 1}$ and $\{Y_t\}_{t\geq 1}$. This specification is made through the conditional densities $q_t(x_t \,|x_{t-1})$ and $g_t(y_t \,|x_t)$, which satisfy the following relations, for $t \geq 1$:

$$\begin{aligned} \mathbb{P}\left(X_t \in A|x_{t-1}\right) &= \int_A q_t(x_t \,|x_{t-1})dx_t, && (1.1) \\ \mathbb{P}\left(Y_t \in A|x_t\right) &= \int_A g_t(y_t \,|x_t)dy_t, && (1.2) \end{aligned}$$

where $\mathbb{P}$ denotes probability and $X_0 = x_0$ is assumed known. We refer to q_t and g_t as the transition and observation densities, respectively. In particular, these relations specify that the hidden state process $\{X_t\}_{t\geq 1}$ is a Markov chain, and the data at time t, Y_t, is conditionally independent of the historical data $\{Y_1, \ldots, Y_{t-1}\}$ given X_t.

This thesis will consider the problem of performing inference with respect to the conditional distribution of the hidden state $\{X_t\}_{t\geq 1}$, given observation of the observed data $\{Y_t\}_{t\geq 1}$. In particular, we will be interested in:

- the distribution of the hidden state at time t, X_t, conditional on the observed data $\{Y_1, \ldots, Y_t\} = \{y_1, \ldots, y_t\}$; and
- the distribution of the hidden state subprocess $\{X_{t_1}, \ldots, X_{t_2}\}$, conditional on the observed data $\{Y_1, \ldots, Y_T\} = \{y_1, \ldots, y_T\}$, where $1 \leq t_1 \leq t_2 \leq T$.

These two problems are referred to as the filtering and smoothing problems, respectively, and this thesis will focus primarily on these problems within the context of the generalised SSM. In particular, we will seek to develop SMC and MCMC methods that avoid the need to either explicitly evaluate the observation density $g(y_t|x_t)$ or replace it with an unbiased estimate. This is a problem of interest in many real world applications, such as population genetics and econometrics, where the observation density has no explicit analytical form and cannot be estimated without bias.

We will also consider a specific class of SSMs, in which both the hidden and observed processes are point processes (PPs); the observed data process is a PP, whose time-inhomogeneous intensity is the hidden state process on which we wish to perform inference. The filtering and smoothing problems, when defined with respect to this specific class of SSMs, are of particular interest in the analysis of ultra-high-frequency (UHF) financial data.

In the next section, we will see how the Bayesian approach to statistical inference is naturally suited to analysing the filtering and smoothing problems. We therefore proceed by giving a recapitulation of the Bayesian paradigm before presenting the filtering and smoothing problems in detail.

1.2 The Problem: Bayesian Filtering and Smoothing

We present the Bayesian framework in the context of performing inference on a random variable (RV) $X \in \mathbb{R}^{d_x}$, $d_x \geq 1$, given the observation of a dependent RV $Y \in \mathbb{R}^{d_y}$, $d_y \geq 1$, which we refer to as the data. The Bayesian paradigm is the concept of updating prior knowledge of X, given the observation of the data. Prior to observing $Y = y$, we represent any existing knowledge of the distribution of X with a prior density, $p(x)$. Upon observation of the data y, we define the *likelihood function* to be the conditional density $p(y|x)$, and this can be used to update our knowledge of X via Bayes' Theorem:

$$p(x|y) = \frac{p(y|x)\, p(x)}{p(y)} \tag{1.3}$$

where $p(y) = \displaystyle\int_{\mathbb{R}^{d_y}} p(y|x)\, p(x)\, dy$ is a normalizing density, typically referred to as the *marginal likelihood* of the data. $p(x|y)$ is known as the *posterior density* of x

and in many situations, it suffices to consider the proportional relationship

$$p\,(x|y) \propto p\,(y|x)\,p\,(x)\,. \tag{1.4}$$

The filtering and smoothing problems of interest are now formalised. A thorough review of the notation adopted in this thesis will be presented in Section 1.5, however we introduce some relevant notation and conventions here. The hidden state and observation processes are denoted $\{X_t\}_{t\geq 1}$ and $\{Y_t\}_{t\geq 1}$, as before, and the distribution of these processes may depend upon a parameter vector $\theta \in \Theta \subseteq \mathbb{R}^{d_\theta}$, which may be treated as either known or unknown. In the case where inference is being performed conditional upon an unknown parameter vector θ, dependence of a given density upon θ will be expressed through a subscript, i.e. we denote the transition and likelihood densities $q_{t,\theta}\,(x_t|x_{t-1})$ and $g_{t,\theta}\,(y_t|x_t)$, respectively. Where the parameter vector is treated as known, the subscript θ is suppressed. We denote finite sub-processes entirely through their subscripts, e.g. for $0 \leq t_1 < t_2 \leq \infty$, $X_{t_1:t_2} := \{X_{t_1}, \ldots, X_{t_2}\}$. Finally, throughout this thesis, $X_0 = x_0$ is assumed known.

The filtering problem addresses the estimation, at time t, of the class of expectations of the form

$$\begin{aligned} \mathbb{E}_{\pi_{t,\theta}}\left[h(X_t)|y_{1:t}\right] &= \int_{\mathbb{R}^{d_x}} h\,(x_t)\,\pi_\theta\,(x_t\,|y_{1:t}\,)\,dx_t & (1.5)\\ \pi_\theta\,(x_t|y_{1:t}) &\propto \int_{\mathbb{R}^{(t-1)d_x}} \left[\prod_{i=1}^{t} g_{i,\theta}\,(y_i\,|x_i\,)\,q_{i,\theta}\,(x_i\,|x_{i-1}\,)\right] dx_{1:t-1}, & (1.6)\end{aligned}$$

for some $\pi_\theta\,(x_t|y_{1:t})$-integrable function h. The expectation in (1.5) is the filtered expectation of the function $h(x_t)$ and $\pi_\theta\,(x_t\,|y_{1:t}\,)$ is the filtering density.

Smoothing addresses a similar estimation problem, where one is interested in expectations with respect to the distribution of the entire hidden process up to the current time point; the filtering density in (1.5) is replaced with the smoothing density:

$$\pi_\theta\left(x_{1:t}|y_{1:t}\right) \propto \prod_{i=1}^{t} g_{i,\theta}\left(y_i\left|x_i\right.\right) q_{i,\theta}\left(x_i\left|x_{i-1}\right.\right), \tag{1.7}$$

and we similarly consider the estimation of smoothed expectations of the form

$$\mathbb{E}_{\pi_{1:t,\theta}}\left[h(X_{1:t})|y_{1:t}\right] \;=\; \int_{\mathbb{R}^{td_x}} h\left(x_{1:t}\right) \pi_\theta\left(x_{1:t}\left|y_{1:t}\right.\right) dx_{1:t},$$

for any $\pi_{1:t,\theta}$-integrable function $h : \mathbb{R}^{td_x} \to \mathbb{R}$.

It is also valid to talk about smoothing when one is interested in the distribution of a particular part of the latent process; for instance, one can compute expectations with respect to the marginal smoothing density,

$$\pi_\theta\left(x_n|y_{1:t}\right) \propto \int_{\mathbb{R}^{(t-1)d_x}} \left[\prod_{i=1}^{t} g_{i,\theta}\left(y_i\left|x_i\right.\right) q_{i,\theta}\left(x_i\left|x_{i-1}\right.\right)\right] dx_{1:n-1} dx_{n+1:t}, \tag{1.8}$$

for any $0 < n < t$. To avoid ambiguity, (1.7) is referred to as the joint smoothing density. Note that the filtering density is simply a particular marginalisation of the joint smoothing density.

From (1.7), it is clear that the joint smoothing density can be calculated recursively. A recursive relationship is also available for the filtering density (1.6), in the form of the so-called filtering recursions:

$$\textit{Predict:} \quad p_\theta\left(x_t\left|y_{1:t-1}\right.\right) = \int_{\mathbb{R}^{d_{x_{t-1}}}} q_{t,\theta}\left(x_t\left|x_{t-1}\right.\right) \pi_\theta\left(x_{t-1}\left|y_{1:t-1}\right.\right) dx_{t-1}, \tag{1.9}$$

$$\textit{Update:} \quad \pi_\theta\left(x_t\left|y_{1:t}\right.\right) = \frac{g_{t,\theta}\left(y_t|x_t\right) p_\theta\left(x_t|y_{1:t-1}\right)}{\int_{\mathbb{R}^{d_{x_t}}} g_{t,\theta}\left(y_t|x_t\right) p_\theta\left(x_t|y_{1:t-1}\right) dx_t}. \tag{1.10}$$

It is clear to see that $p_\theta(x_t|y_{1:t-1})$, which is known as the predictive filter, corresponds to a prior density for X_t, i.e. prior to the observation of a new data point y_t. It follows that the update step above corresponds to Bayes theorem (1.3); this motivates our choice to perform inference in a Bayesian framework.

Consider the discrete-time SSM that can be written in the form

$$X_t = Q_t(X_{t-1}, V_t), \quad t \geq 1, \tag{1.11}$$

$$Y_t = G_t(X_t, Z_t), \quad t \geq 1, \tag{1.12}$$

where Q_t and G_t are linear functions, referred to as the transition and observation functions, respectively, and $V_t \in \mathbb{R}^{d_x}$ and $Z_t \in \mathbb{R}^{d_y}$ are independent Gaussian RVs; this is referred to as a linear, Gaussian SSM. It was shown by Kalman (1960) that, in this scenario, if the filter $\pi_\theta(x_t|y_{1:t})$ is Gaussian, then the filter at time $t+1$ will also be Gaussian. Furthermore, Kalman showed that the first and second moments of the filtering distribution can be calculated exactly through the filtering recursions. Thus, exact inference with respect to a linear Gaussian SSM may be performed deterministically by projecting the first and second moments of the distribution through the filtering recursion, and this is referred to as the Kalman filter. These results were extended to the continuous time, linear, Gaussian SSM by Kalman and Bucy (1961).

The inevitable next step in the development of the filtering problem was the generalisation to the scenario where both the transition and observation functions, respectively Q_t and G_t above, are allowed to be nonlinear. Approximate numerical solutions to the nonlinear problem follow from the Kalman filter, with the extended Kalman filter (EKF; see Sorenson and Stubberud, 1968), the unscented Kalman filter (UKF; see Julier and Uhlmann, 1997) and the ensemble Kalman filter (EnKF; see Evensen, 1994) being important examples. The Kalman filter and its derivatives are all based on the same two-step update procedure described above, and are developed under the assumption that the latent and observed processes are driven by Gaussian noise. Although these variants can often be applied to SSMs with non-Gaussian dynamics, the accuracy of the resulting filtered estimates will typically deteriorate as we depart from Gaussianity. Attention therefore turns to alternative solutions to the nonlinear filtering problem.

Sequential Monte Carlo methods, which are the focus of this thesis, are a broad class of numerical methods aimed at approximating a sequence of related distributions, facilitating the estimation of expectations of the form (1.5) through Monte Carlo integration. Since the application of SMC methods is subject neither to linearity nor distributional constraints, they are considered the preferred approach to the filtering and smoothing problems. The fundamentals of SMC methodology and its application to these problems are addressed in greater detail in Chapter 2.

1.3 Objectives of the Thesis

This thesis presents some new developments in the application of SMC methodology to the filtering and smoothing problems, building upon current state-of-the-art approaches and facilitating inference for problems of interest that cannot be addressed using standard methods. As will be seen, the point-wise evaluation of the likelihood is central to the execution of SMC-based filtering and smoothing methods, yet in many real-world applications, such as the study of population genetics and stochastic volatility modelling, the likelihood is often either analytically unavailable with no unbiased estimate, or too complex to be calculated with a reasonable computational budget. For the remainder of this thesis, an *intractable* likelihood will be one that cannot be pointwise evaluated or replaced with an unbiased estimator, and we remark that the issue of intractability is kept separate from any issues of computational complexity in the evaluation of the likelihood.

The main focus of this work is the development of approximate Bayesian computation (ABC) methodology for state-space models (SSMs), as well as the development of SMC and MCMC methods in the context of these ABC SSMs for filtering and smoothing of the latent process. The introduction of ABC here avoids the need to evaluate the likelihood, at the cost of introducing a bias into the resulting filtering and smoothing estimators; this bias will be explored theoretically and through simulation studies. Alternative approaches to the filtering problem that address the problem of likelihood intractability include variational Bayesian filtering (Šmídl and Quinn, 2008) and convolution particle filtering (Campillo and Rossi, 2009); the latter will be discussed briefly in Chapter 3.

We will also consider an alternative SMC procedure for filtering and smoothing, which incorporates an additional rejection step. The use of Rejection SMC in the reported form is not widespread, yet it is theoretically justified and has the potential to improve performance for filtering and smoothing, particularly in the ABC framework. Justification for the presented procedure is given and the novel application of our procedure to the ABC approximation of the SSM is considered.

This thesis will also consider the application of MCMC and SMC methods to a class of partially observed point process (PP) models. We investigate the problem of performing sequential inference for these models and note that current methods often fail. We present a new approach to smoothing in this context, using SMC samplers (Del Moral et al., 2006). This approach is illustrated, with some theoretical discussion, on a doubly stochastic PP applied in the context of finance.

1.4 Contributions of the Thesis

The work presented in Chapter 3 provides some of the first empirical results concerning the accuracy of performing particle filtering with respect to an ABC approximation of the generalised SSM. Chapter 4 presents the first theoretical results concerning the deterministic bias associated with performing smoothing-based inference with respect to the ABC approximation of the SSM; this chapter also presents the first empirical results concerning the accuracy of ABC smoothing procedures. In both Chapters 3 and 4, we present the first empirical evidence that the use of a rejection-based SMC procedure can reduce the variance of the resulting estimates. In Chapter 4, we propose a variant of the state-of-the-art PMMH procedure of Andrieu et al. (2010) that employs a forward-smoothing procedure (Del Moral et al.,

2009) in the update scheme, and this is empirically observed to outperform the PMMH procedure of Andrieu et al. (2010) when applied to the SSM and its ABC approximation. Finally, in Chapter 5, we propose two different variants of an SMC sampler algorithm (Del Moral et al., 2006), which allow, for the first time, an SMC-based approach to performing inference with respect to an SSM in which both the latent and observed processes are point processes.

Some of the work in Chapter 3 is also presented in joint work with Dr. Jasra, Dr. S.S. Singh (University of Cambridge) and Dr. McCoy (Jasra et al., 2010). The work in Chapter 5 is also presented in a joint paper with Dr. Jasra and Dr. McCoy (Martin et al., 2012a). The work in Chapter 4 is the subject of an Imperial College Technical Report (Martin et al., 2012b), which is joint work with all of the above authors and Dr. N. Whiteley (University of Bristol).

1.5 Notation and Conventions

In this section, we detail notation that will be used throughout this thesis. Wherever possible, the notation provided here will remain consistent throughout, and any deviations in the notation will be highlighted when they are encountered.

1.5.1 Measure Theory, Random Variables and Processes

Given the sample subspace $E \subseteq \Omega$, with corresponding σ-algebra $\mathcal{E}$, the function $\mu : \mathcal{E} \to \mathbb{R}$ is a measure if it is nonnegative, i.e. $\mu(A) \geq 0 \ \forall A \in \mathcal{E}$, zero when evaluated on the null set, i.e. $\mu(\emptyset) = 0$, and countably additive:

$$\mu\left(\bigcup_{i=1}^{n} E_i\right) = \sum_{i=1}^{n} \mu\left(E_i\right),$$

for pairwise disjoint subspaces $E_i \subseteq E$, $i = 1, \ldots, n$. The pairing $(E, \mathcal{E})$ is referred to as a measurable space and we say that the triple $(E, \mathcal{E}, \mu)$ is a measure space if μ is a measure defined on that measurable space.

A measure μ is a probability measure if it is restricted to the unit interval, i.e. $\mu : \mathcal{E} \to [0,1]$, and it further satisfies $\mu(E) = 1$. The triple $(E, \mathcal{E}, \mu)$ is a probability space if and only if μ is a probability measure defined on $(E, \mathcal{E})$.

Throughout, random variables (RVs) will be denoted in the uppercase, suppressing reference to the corresponding event in the sample space, with their values being represented by the corresponding lowercase character. For example, $X : \Omega \to E$ is an $(E, \mathcal{E})$-valued RV, and we may write $X \in E$.

For a particular point $x \in E$, $dx \in \mathcal{E}$ will refer to the infinitesimal subspace of E containing x. We will often denote measures e.g. $\mu(dx)$, in order to reflect the dependence of the measure on the value of the random variable. If $\mu(dx)$ is a probability measure, then we may talk about the probability *distribution* of the RV X, say. This refers to the set of values that $\mu(dx)$ can take for all $x \in E$ and, indeed, a probability distribution is defined by the specification of the probability measure $\mu(dx)$. We may say that an RV X is *distributed* according to a distribution $\mu(dx)$, and this is denoted $X \sim \mu(dx)$. Where we are interested in $N \geq 1$ RVs that are identically distributed according to $\mu(dx)$, and not part of the same time-indexed process, we use the following notation: $X^{(i)} \sim \mu(dx)$, $i = 1, \ldots, N$.

This thesis will be concerned with performing inference with respect to various probability measures. In particular, we will often be interested in the integral

$$\int_E h(x)\mu(dx),$$

where h is referred to as being μ-integrable. When $\mu(dx)$ is a probability measure, the above integral is the expectation of $h(X)$ with respect to μ.

For two measures $\mu(dx)$, $\lambda(dx)$, defined upon a common measurable space $(E, \mathcal{E})$, we say that μ is *dominated* by λ if $\lambda(A) = 0 \Rightarrow \mu(A) = 0$, $\forall A \in \mathcal{E}$. This is alternatively written $\mu \ll \lambda$ and we may also say that μ is *absolutely continuous* with respect to λ. In such a scenario, the Radon-Nikodym Theorem states that there exists a λ-measurable function, referred to as the Radon-Nikodym derivative of μ with respect to (w.r.t.) λ, and denoted $d\mu/d\lambda$, such that

$$\mu(dx) = \int_E \frac{d\mu}{d\lambda}(x)\lambda(dx).$$

If μ is a probability measure, dominated by λ, then the probability density function admitted by μ w.r.t. λ is defined as the Radon-Nikodym derivative $d\mu/d\lambda$.

For two measures μ_1, μ_2, defined upon $(E, \mathcal{E})$, define the *total variation distance* between the two measures as

$$\|\mu_1 - \mu_2\|_{TV} = \sup_{A \in \mathcal{E}} |\mu_1(A) - \mu_2(A)|\,.$$

We will deal almost exclusively with time-indexed processes. For the sequence of $(E, \mathcal{E})$-valued RVs $\{X_t\}_{t\geq 0}$, the notation $X_{t_1:t_2}$ will denote: the finite sub-process $\{X_{t_1}, \dots, X_{t_2}\}$, for $0 \leq t_1 < t_2 < \infty$; the RV $X_{t_1} \in E$ for $t_2 = t_1$; and the null set $\emptyset$ for $t_1 > t_2$. Indeed, throughout, we adopt the conventions that $\sum_\emptyset = 0$ and $\prod_\emptyset = 1$.

In the SSM setup, we will be concerned with problems in which $E = \mathbb{R}^d$, for some dimension $d \geq 1$, and $\mathcal{E}$ is generated as the Borel σ-algebra of $\mathbb{R}^d$. In particular, the processes of interest will be the hidden state process $\{X_t\}_{t\geq 1}$ and the observation process $\{Y_t\}_{t\geq 1}$ such that $X_t \in \mathbb{R}^{d_x}$ and $Y_t \in \mathbb{R}^{d_y}$ $\forall t \geq 1$. Where a process is denoted without specifying the range of the time index, we will assume $t \geq 1$

1.5.2 Filtering and Smoothing Densities

Attention in this thesis lies principally in performing inference with respect to filtering and smoothing distributions, as described in Section 1.2. These distributions will typically evolve over time; we use time indices, either in the subscript of the argument or in the subscript of the measure when the measures are referred to without argument. Thus, we have the following notation for the filtering and smoothing distributions:

- $\pi_t = \pi(dx_t|y_{1:t})$, the filtering distribution at time t,
- $\pi_{1:t} = \pi(dx_{1:t}|y_{1:t})$, the joint smoothing distribution at time t,
- $\pi_{s:t,T} = \pi\left(dx_{s:t}\left|y_{1:T}\right.\right)$, the marginal smoothing distribution over $[s,t]$, given observation of the data up to time T.

Whenever an unknown parameter vector θ is included in the subscript of the measure, the time parameter takes precedence. Throughout, the initial state x_0 will be treated as known, and the dependence of the filtering and smoothing distributions on this known parameter is suppressed.

We also note here that wherever a generic density notation is required, we use $p(\cdot)$.

1.5.3 Markov Chains

This thesis will use Markov chains extensively, and these are introduced in Section 2.2.2. We provide here a summary of some key notation.

A Markov chain is a time indexed process $\{X_t\}_{t\geq 0}$ whose value at time t is dependent only on its value at $t-1$. Assuming that the chain evolves on the measurable space $(E,\mathcal{E})$, we have that the distribution of the chain at time t is defined through the probability measure $Q_t : E \times \mathcal{E} \longrightarrow [0,1]$, which is known as a Markov transition kernel and denoted $Q_t(x_{t-1}, dx_t)$. Assuming a suitable dominating measure, we denote the corresponding Markov transition density as $q_t(x_t|x_{t-1})$. Occasionally, we will wish to consider transition kernels and densities without referring to a particular time index; we will use the notation $Q(x, dx^*)$ and $q(x^*|x)$ for these respective quantities.

A key concept in defining Markov chains is that of the *invariant* distribution. For the Markov chain above, this is defined in Section 2.2.2 to be the probability measure $\pi(dx)$ such that, at time t,

$$\int_E \pi(dx_{t-1})Q_t(x_{t-1}, dx_t) = \pi(dx_t).$$

We will be interested in the simulation of Markov chains with a specified invariant distribution. The techniques that we shall use to do so will involve the use of an alternative probability measure $K_t : E \times \mathcal{E} \to [0, 1]$ to propose candidate values for the chain at time t. This probability measure is referred to as a proposal kernel and is denoted $K_t(x_{t-1}, dx_t)$. Assuming a suitable dominating measure here, we can also denote the corresponding proposal density as $k_t(x_t|x_{t-1})$. As with the transition kernel and density, we may also wish to denote the proposal kernel and density without reference to the time parameter; the notation follows the same conventions as above.

When dealing with particle MCMC methods in Section 2.4, we will be interested in constructing Markov chains for which the RV at each iteration is itself the time indexed hidden state process in an SSM. We will therefore have two 'time' indices to keep track of. In this scenario, we move the Markov chain time index out of the subscript, and denote the Markov chain of interest $\{X_{1:T}(n)\}_{n\geq 1}$.

1.5.4 Sequential Monte Carlo

SMC methods are designed to generate $N \geq 1$ interacting time-indexed processes, which are referred to in this context as *particles*. We denote the ensemble of particles up to time t as $\left\{X_{1:t}^{(i)}\right\}_{i=1}^{N}$, and we have that, at each time $t \geq 1$, $\left\{X_{1:t}^{(i)}\right\}_{i=1}^{N}$ forms a joint Markov chain. The remaining SMC notation will be introduced in the exposition of the methodology.

We establish a convention in the terminology. An algorithm is said to be *sequential* if it is able to process data as it arrives over time, and an algorithm is said to be *on-line* if it is sequential and has a fixed computational cost per iteration/time-step.

1.5.5 Approximate Bayesian Computation

This thesis will consider the use of ABC methodology, and we summarise here some of the notation that will be used. This notation is presented out of context and without any definitions, and so should serve principally as a reference point for the reader.

Central to the use of ABC is the definition of the auxiliary RV $U \in \mathbb{R}^{d_y}$, known as the pseudo-data, which is defined on the same space as the RV that corresponds to a set of observed data, $Y \in \mathbb{R}^{d_y}$. The pseudo-data and the observed data are then used to form an approximation of the likelihood function, using the ABC kernel function, $G^{\epsilon}(u, y)$. In general, this is dependent on Y and U through an ABC tolerance parameter ϵ and a distance metric $\rho(s(u), s(y))$, where $s(\cdot)$ is a summary statistic.

The ABC kernel, described above, can be used to form approximations of several densities of interest; the ABC approximations of the densities are denoted through use of an ϵ in the superscript, e.g. the ABC approximation of the filtering density $\pi_{\theta}^{\epsilon}(x_t|y_{1:t})$ corresponds to the exact filtering density $\pi_{\theta}(x_t|y_{1:t})$.

1.5.6 Point Processes

In Chapter 5, we will consider the HMM in which both the hidden state and observed data processes are point processes (PPs). We summarise here the notation that is introduced there, and note that this is once again presented with little context, but may serve as a useful reference point.

The latent PP is defined on the interval $[0, t_n]$ by the variables $(k_{t_n}, \phi_{1:k_{t_n}}, \zeta_{1:k_{t_n}})$ where $\phi_{1:k_{t_n}}$ are the ordered event times (constrained to $[0, t_n]$), with k_{t_n} the number of events up to t_n, and $\zeta_{1:k_{t_n}}$ are the values of the process at each event time.

The observed PP is defined similarly, and consists of the variables $(r_{t_n}, \omega_{1:r_{t_n}}, \xi_{1:r_{t_n}})$ where $\omega_{1:r_{t_n}}$ are the event times and $\xi_{1:r_{t_n}}$ are the observed values of the process at each event time. In Chapter 5, the observed PP is interpreted as a set of financial data; in this context, $\omega_{1:r_{t_n}}$ are the trading times and $\xi_{1:r_{t_n}}$ are the log-returns on the r_{t_n} observed financial transactions.

Additional notation will be provided in the exposition of the methodology in Chapter 5.

1.6 Structure of the Thesis

In Chapter 2, we provide an in-depth review of MCMC and SMC methods, as well as giving an introduction to particle MCMC (PMCMC) methods and approximate Bayesian computation. We also present a rejection-based SMC procedure, referred to here as RSMC, which was introduced by Del Moral (2004).

In Chapter 3, we introduce the ABC approximation of the generalized SSM, and we specify the ABC approximations of the joint smoothing and filtering densities. We discuss some theoretical results concerning the deterministic bias of the ABC approximation of the filtering density, before providing details of the implementation of particle filtering in the ABC framework. We also present a detailed numerical analysis of the accuracy of the ABC particle filter, applying it first to a linear Gaussian model, and then to a challenging nonlinear Gaussian model from the SMC literature. We also consider the use of RSMC and provide the first empirical results concerning the performance of this procedure in a filtering context.

In Chapter 4, we consider the problem of performing inference with respect to the ABC approximation of the joint smoothing distribution. We present a detailed theoretical analysis of the deterministic bias that is induced on a class of expectations that are defined with respect to the ABC approximation of the joint smoothing distribution. We present two SMC-based procedures for performing sequential inference with respect to the ABC approximation of the joint smoothing density, and we present a novel variant of the PMMH algorithm of Andrieu et al. (2010), which can be used to perform batch inference with respect to both the exact joint smoothing distribution and its ABC approximation. In this chapter, we also consider the use of RSMC in the context of smoothing.

In Chapter 5, we consider the application of SMC methods to an SSM comprised of a pair of point processes. We consider two existing approaches to performing inference with respect to these point process models, including an SMC sampler approach that will be developed throughout the chapter. We demonstrate the poor performance of a straightforward application of an SMC sampler to this class of models, and we propose two alternative modifications to the sampler. The improved performance of these proposed samplers is demonstrated on both simulated and real financial data.

In Chapter 6, we offer a final summary of the contributions of this thesis, and present some ideas for future development of the work presented here.

1.7 List of Acronyms

In Table 1.1, we provide a list of acronyms used in this thesis.

Acronym	Meaning
ABC	Approximate Bayesian computation
CLT	Central Limit Theorem
(E/T)RSMC	(Empirical/Theoretical) Rejection SMC
ESS	Effective sample size
IMH	Independent MH
IS	Importance Sampling
MC	Monte Carlo
MCMC	Markov chain Monte Carlo
MH	Metropolis-Hastings
MMH	Marginal MH
MPDABC	Multiple pseudo-data ABC
PIMH	Particle IMH
PMCMC	Particle MCMC
PMMH	Particle MMH
PRC	Partial rejection control
RJMCMC	Reversible jump MCMC
SIR	Sequential importance resampling
SIS	Sequential IS
SLLN	Strong Law of Large Numbers
SMC	Sequential Monte Carlo
SSM	State space model

Table 1.1: The acronyms used in this thesis.

Chapter 2

Monte Carlo Methods: A Review

2.1 Introduction

This chapter presents a review of Markov chain Monte Carlo (MCMC; Section 2.2) and sequential Monte Carlo (SMC; Section 2.3) methods, as well as an introduction to particle MCMC (PMCMC; Section 2.4). An introduction to approximate Bayesian computation (ABC) is also provided in Section 2.5.

2.2 Markov Chain Monte Carlo

Markov chain Monte Carlo (MCMC) methods are a well established class of simulation-based techniques, which can be used for approximating the expectation of a given function with respect to a complex probability distribution. Loosely speaking, this is achieved through generating a large number of statistically dependent samples from the distribution of interest, referred to as the target distribution, and using Monte Carlo estimation, defined below, to approximate the expectation of interest. MCMC methods are widely applicable, as their use requires only that the density of the target distribution is either known pointwise up to a normalizing constant or can be replaced by an unbiased estimator.

The Metropolis-Hastings algorithm and the Gibbs sampler are two procedures that fall under the MCMC umbrella, and they can be used either independently or in conjunction with each other to provide efficient and accurate simulation methods for exploring a given target distribution. In a Bayesian setting, many problems of interest can be approached using MCMC methods, and we will see that the Metropolis-Hastings and Gibbs procedures provide the basis for the more advanced methods that can be used to tackle the Bayesian filtering and smoothing problems that are of interest here. A necessary precursor to MCMC methods is Monte Carlo estimation, and so we provide a brief introduction to this concept before introducing MCMC.

2.2.1 Monte Carlo Estimation

Consider the probability space $(E, \mathcal{E}, \pi)$, and suppose we wish to evaluate the expected value of a π-integrable function f,

$$\mathbb{E}_\pi \left[f(X)\right] = \int_E f(x)\pi(dx).$$

Typically, we find that the probability distribution $\pi(dx)$ is too complex for standard integration methods and that the dimension of the space E is too large for deterministic numerical approaches such as quadrature. We therefore turn to Monte Carlo (MC) methods, which provide a simulation-based numerical approach that can deal with both complex integrands and high-dimensional spaces.

If the distribution of interest, $\pi(dx)$, may be sampled from straightforwardly, then an unbiased estimator for the expectation of interest is provided by the Monte Carlo estimate, which is defined as

$$\hat{f}_\pi^N(X) = \frac{1}{N}\sum_{i=1}^{N} f\left(X^{(i)}\right), \tag{2.1}$$

where $X^{(1)}, \ldots, X^{(N)} \sim \pi(dx)$ are independently drawn samples.

Whilst there are techniques for sampling from well-known distributions (e.g. Gaussian, gamma, e.t.c.), the direct sampling of independent, identically-distributed (i.i.d.) random variables from many distributions of interest is a non-trivial problem, and this is where Markov chains prove useful. Relatively straightforward algorithms exist for efficiently generating Markov chains with a specified stationary, or invariant distribution, and so a natural alternative to direct sampling is to generate a Markov chain with invariant distribution $\pi(dx)$. As we shall see in the next section, the generated chains must satisfy only weak conditions in order to be considered suitable samples from $\pi(dx)$.

2.2.2 Markov Chain Theory

We now detail some Markov chain theory, much of which can be found in Roberts and Rosenthal (2004) and Tierney (1998). For concise work on techniques for Markov chain simulation, Tierney (1994) was also consulted.

Consider the discrete-time stochastic process $\{X_t\}_{t\geq 0}$, allowed to evolve on the measurable space $(E, \mathcal{E})$; we say that the process is Markovian if the distribution of its future state is dependent only upon its current state and not upon any past state. This basic assumption is defined more rigorously by the following: for $t \geq 0$, the distribution of the $(E, \mathcal{E})$-valued RV X_{t+1}, conditional upon $X_{0:t} = x_{0:t}$, is defined by the probability measure $Q_t : E \times \mathcal{E} \to [0, 1]$, and is denoted $Q_t(x_t, dx_{t+1})$. We assume that the distribution of X_0 is known, and without loss of generality, we will assume that this is the point mass at x_0, i.e. $X_0 = x_0$ is known.

The joint law of the Markov chain is the measure defining the finite-dimensional distribution induced by the subprocess $X_{0:t}$ on the product measure space $(E^{t+1}, \mathcal{E}^{t+1})$, and will be denoted $\mathcal{L}\,(dx_0, \ldots, dx_t)$. We can also talk about the marginal law of the Markov chain at a particular time t, which we shall denote $\mathcal{L}(dx_t)$. This is the measure defining the distribution induced by the process on the random variable X_t. According to the Ionescu Tulcea Extension Theorem (see, e.g. Shiryaev, 1996, p.249), this can be iteratively defined as the pushforward measure composed of the marginal law of the chain at $t-1$ and the Markov transition kernel $Q_t\left(x_{t-1}, dx_t\right)$:

$$\mathcal{L}(dx_t) = \int_E \mathcal{L}\left(dx_{t-1}\right) Q_t\left(x_{t-1}, dx_t\right), \quad t \geq 2,$$

where the marginal law at $t=1$ is given by $\int_E Q_1\left(x_0, dx_1\right)$. As a result, the law of the Markov chain is fully specified by its initial value and its transition kernel.

The motivation behind using Markov chains lies in the ability to sample a chain with a prespecified invariant (or stationary) distribution. In practice, the chain is constructed such that its invariant distribution is equal to the target distribution $\pi(dx)$.

Specification of the Invariant Distribution

For a Markov chain with transition kernel $Q\left(x, dx^*\right)$, we define an invariant distribution $\pi(dx)$ on the measurable space $(E, \mathcal{E})$ to be such that,

$$\int_E \pi(dx) Q(x, dx^*) = \pi(dx^*).$$

As noted by Roberts and Rosenthal (2004), in order to specify an invariant distribution $\pi(dx)$ for a chain $\{X_t\}$, it is a sufficient condition for the chain to be *reversible* with respect to $\pi(dx)$. That is, the transition kernel Q and the distribution $\pi(dx)$ satisfy the detailed balance condition:

$$Q(x, dx^*)\pi(dx) = Q(x^*, dx)\pi(dx^*). \tag{2.2}$$

It is, however, insufficient simply for there to exist an invariant distribution; it is indeed possible for a chain to have an invariant distribution to which it does not converge. MCMC algorithms must therefore be designed such that convergence to the target distribution can be guaranteed.

Convergence to the Invariant Distribution

We detail here the two conditions on the chain that, between them, guarantee convergence to the target distribution $\pi(dx)$, with respect to the total variation distance $\|\mathcal{L}_t - \pi\|_{TV}$. We give these conditions before presenting the convergence result.

ϕ-irreducibility Define $\{X_t\}$ to be ϕ-irreducible if there exists a non-zero σ-finite measure ϕ on $(E, \mathcal{E})$ such that for all sets $A \in \mathcal{E}$ with $\phi(A) > 0$, there exists $1 \leq k < \infty$ such that $\forall x_t \in E$,

$$Pr\left(X_{t+k} \in A \middle| x_t\right) > 0.$$

Aperiodicity A Markov chain $\{X_t\}$ is defined here to be periodic if there exist $d \geq 2$ disjoint subsets $A_1, \ldots, A_d \in \mathcal{E}$ such that

$$\begin{aligned} Q(x, A_{i+1}) &= 1 \quad \forall x \in A_i, \quad i = 1, \ldots, d-1, \\ Q(x, A_1) &= 1 \quad \forall x \in A_d. \end{aligned}$$

We say that $\{X_t\}$ has a period d and that $A_1, \ldots, A_d$ is the periodic decomposition. If there does not exist $d \geq 2$ such that we can construct a periodic decomposition, the chain is aperiodic.

If the Markov chain $\{X_t\}$ with invariant distribution equal to our target distribution $\pi\left(dx\right)$ is both ϕ-irreducible and aperiodic then, by Theorem 4 of Roberts and Rosenthal (2004), this is also the limiting distribution for $\{X_t\}$ as $t \rightarrow \infty$; for $\pi - a.e.\ x_0 \in E$,

$$\lim_{t \rightarrow \infty} \|\mathcal{L}_t - \pi\| = 0.$$

If a Markov chain converges to its invariant distribution, then this is also referred to as its equilibrium distribution.

MCMC algorithms proceed by constructing a Markov chain with equilibrium distribution equal to the target sampling distribution. Once a Markov chain of sufficient length has been created, a pre-defined burn-in period, in which the chain is considered not to have reached its equilibrium, is discarded and the remainder of the chain provides a statistically dependent sample from the target distribution. Since Monte Carlo estimation requires an i.i.d. sample from the distribution of interest, an immediate concern would be the effect that the Markov dependence in the chain would have on the accuracy of the subsequent MC-type estimate.

It is shown by Meyn and Tweedie (2009) that for a ϕ-irreducible, aperiodic (i.e. ergodic) M-length Markov chain $X_{1:M}$ with invariant distribution $\pi(dx)$, the resulting MC estimator $\hat{f}_\pi^M(X)$, calculated as in (2.1), satisfies the Strong Law of Large Numbers (SLLN) for any test function f such that $\int_E |f(x)|\, \pi(dx) < \infty$:

$$\lim_{M\to\infty} \hat{f}_\pi^M(X) = \mathbb{E}_\pi\left[f(X)\right] \quad w.p.1. \tag{2.3}$$

Thus, the conditions of ϕ-irreducibility and aperiodicity are sufficient to guarantee convergence of the MCMC estimate to the expectation of any test function that is absolutely integrable with respect to the target distribution.

2.2.3 Metropolis-Hastings and the Gibbs Sampler

For a brief review of MCMC development see Andrieu et al. (2004); more thorough expositions can be found in the books by Liu (2001) and Robert and Casella (2004). Here, we shall concentrate on the two most widely used MCMC algorithms; the Metropolis-Hastings algorithm and the Gibbs sampler.

Metropolis-Hastings

Introduced by Metropolis et al. (1953) and generalised by Hastings (1970), the principle behind the Metropolis-Hastings algorithm is to iteratively generate a Markov chain by starting at an arbitrary value and proposing and subsequently accepting or rejecting candidate values for the remainder of the chain. Suppose we wish to construct a Markov chain, $\{X_t\}$; the proposal of a candidate value x^* is generated according to a proposal kernel $K\left(x, dx^*\right)$ with corresponding density $k\left(x^*|x\right)$, defined with respect to some dominating measure. The acceptance of this candidate value is then determined by an acceptance probability $\alpha\left(x, x^*\right)$. Thus the transition kernel for the resulting Markov chain may be written

$$\begin{aligned} Q\left(x, A\right) &= \int_A K\left(x, dx^*\right) \alpha\left(x, x^*\right) + \left(1 - \int_A K\left(x, dx^*\right) \alpha\left(x, x^*\right)\right) \mathbb{I}_A(x) \\ &= \int_A \alpha\left(x, x^*\right) k\left(x^*|x\right) dx^* + \left(1 - \int_A \alpha\left(x, x^*\right) k\left(x^*|x\right) dx^*\right) \mathbb{I}_A(x). \end{aligned}$$

By considering this representation of the transition kernel and requiring the detailed balance equation in (2.2) to hold for reversibility to be guaranteed, the form of $\alpha\left(x, x^*\right)$ may be determined to be (Tierney, 1998)

$$\alpha\left(x, x^*\right) = \begin{cases} r\left(x, x^*\right) \wedge 1 & x, x^* \in R \\ 0 & x, x^* \in R^c, \end{cases} \tag{2.4}$$

where $R \in \mathcal{E} \otimes \mathcal{E}$ is the symmetric set on which the measures $\pi(dx)K\left(x, dx^*\right)$ and $\pi(dx^*)K\left(x^*, dx\right)$ are equivalent, i.e. mutually absolutely continuous. $r(x, x^*)$ will be referred to as the acceptance ratio, and is defined as the ratio of Radon-Nikodym derivatives of $\pi(dx^*)K\left(x^*, dx\right)$ and $\pi(dx)K\left(x, dx^*\right)$, taken with respect to some common dominating measure. This common dominating measure is guaranteed by their mutual absolute continuity; the fact that share a common dominating measure also allows us to write the acceptance ratio as

$$r\left(x, x^*\right) = \frac{\pi\left(x^*\right) k\left(x|x^*\right)}{\pi\left(x\right) k\left(x^*|x\right)},$$

```
Parameters: $M \geq 1$, $x_0 \in E$;
Result: A statistically dependent Markov chain, $X_{0:M}$, with limiting
        distribution equal to the target distribution $\pi(dx)$
Initialization: Set $X_0 = x_0$.
for $t = 1 \ldots M$ do
    • Given $x_{t-1} = x$, generate a candidate value $x^*$ according to the proposal
      kernel $K(x, dx^*)$.
    • Define $\alpha(x, x^*)$ as in (2.4).
    • Set $\begin{cases} X_t = x^* & \text{with probability } \alpha \\ X_t = x & \text{otherwise} \end{cases}$
end
```

Algorithm 1: The Metropolis-Hastings algorithm

and specify the symmetric set R as

$$R = \{(x, x^*) : \pi(x^*) \, k(x|x^*) > 0 \text{ and } \pi(x) \, k(x^*|x) > 0\}.$$

The MH procedure is detailed in Algorithm 1.

The original procedure proposed by Metropolis et al. (1953) uses a symmetric proposal density $k(x^*|x) = k(x|x^*)$, further simplifying the acceptance ratio:

$$r(x, x^*) = \frac{\pi(x^*)}{\pi(x)}.$$

In high dimensions, this simplification has the potential to save a significant amount of computational effort. This must be caveated by saying that the computational efficiency of the procedure is still dominated by the quality of the proposal density $k(x^*|x)$. The quality of a proposal is judged largely in terms of its resulting rejection rate, that is the proportion of candidate values it generates that are subsequently rejected. If the choice of proposal results in a high rejection rate, then exploration of the sample space will require more iterations of the algorithm, increasing computational effort. However, if the rejection rate is low, then it will take longer for the distribution of the chain to converge to the target distribution. Conventionally, an acceptance rate of approximately 23% is usually sought (see Roberts et al., 1997).

The Metropolis-Hastings algorithm produces, by construction, a Markov chain that is reversible with respect to the target distribution. Verification of the condition of ϕ-irreducibility via a suitable choice of ϕ is straightforward in most cases, with it usually sufficing for ϕ to be the Lebesgue measure (Roberts and Rosenthal, 2004). Finally, we note that a sufficient condition for aperiodicity in the general Markov chain with transition kernel $Q\left(x, dx^{*}\right)$ is (Tierney, 1994)

$$Q\left(x, dx^{*}\right) > 0, \quad \forall x \in dx^{*},$$

and that the extension of this condition to the Metropolis-Hastings setting is to condition on the probability of the rejection of the candidate state being positive (Robert and Casella, 2004):

$$\Pr\left[r\left(x, x^{*}\right) < 1\right] > 0.$$

which, in the case of common dominating measures for proposal and target densities, can be written

$$\Pr\left[\pi\left(x\right) k\left(x^{*}|x\right) > \pi\left(x^{*}\right) k\left(x|x^{*}\right)\right] > 0.$$

This condition is generally satisfied in practice, as it would only be contravened in the case where the proposal kernel K is the transition kernel for a reversible Markov chain. This can be seen by considering the following: since x and x^{*} are interchangeable, the above is contradicted if and only if

$$\pi(x)k(x^{*}|x) = \pi(x^{*})k(x|x^{*}),$$

which is simply the detailed balance condition for the Markov chain with transition kernel $K(x, dx^*)$ and invariant distribution $\pi(dx)$.

Independent Metropolis-Hastings and Marginal Metropolis-Hastings

In MH procedures, specific choices for the proposal kernel $K(x, dx^*)$ result in variations on the basic algorithm that will be of interest when considering PMCMC methods in Section 2.4; these are briefly introduced here.

Consider the proposal kernel $K(x, dx^*) = K(dx^*)$, that is the kernel that proposes candidate values independent of the current value of the Markov chain. The candidate value is accepted or rejected according to the acceptance probability $\alpha(x, x^*)$ as in (2.4), where $r(x, x^*)$ is now defined as the Radon-Nikodym derivative of $\pi(dx^*)K(dx)$ with respect to $\pi(dx)K(dx^*)$. The resulting MH procedure is referred to as Independent MH (IMH).

Suppose that interest lies in performing inference with respect to the joint distribution $\pi(dx, d\theta)$, where $\theta \in \Theta$ is an unknown parameter, say. We use an MH procedure to target the joint distribution $\pi(dx, d\theta) = \pi_\theta(dx)\pi(d\theta)$, and we make the key assumption that it is possible to sample from the conditional distribution $\pi_\theta(dx)$. Under this assumption, we adopt the proposal kernel $K(\{x, \theta\}, \{dx^*, d\theta^*\}) = \pi_{\theta^*}(dx^*)K(\theta, d\theta^*)$, i.e. we propose a candidate parameter θ^* and then subsequently sample X^* from the conditional distribution $\pi_{\theta^*}(dx^*)$. The candidate values are subsequently accepted or rejected, based upon the MH acceptance probability, which, assuming a common dominating measure for the target density and proposal kernel, can be shown to take the form

$$\alpha(\{x, \theta\}, \{x^*, \theta^*\}) = 1 \wedge \frac{\pi(\theta^*)k(\theta|\theta^*)}{\pi(\theta)k(\theta^*|\theta)},$$

which we note is simply the acceptance probability for the MH procedure targeting the marginal distribution $\pi(d\theta)$. This approach to sampling from the joint distribution $\pi(dx, d\theta)$ is thus referred to as marginal MH (MMH).

Metropolis-Hastings algorithms provide generic procedures for generating ergodic Markov chains that converge to a given invariant distribution; little information about the probabilistic structure of the target distribution is required in order to implement them. Given knowledge of the target distribution's structure, however, allows a procedure for generating Markov chains that does not require an accept/reject step: the Gibbs sampler, introduced by Geman and Geman (1984).

The Gibbs Sampler

The Gibbs sampler can be well suited to generating multidimensional chains given knowledge of a conditional structure of the target distribution. A thorough introduction to the Gibbs sampler and its variants, including coverage of their convergence properties, is provided by Robert and Casella (2004).

The sampler begins by dividing the variable into a number of components of smaller dimension; for notation's sake, we will here assume that the variable is split into components of single dimension. The sampler proceeds by cycling through the individual components, sampling a new value for each according to the target distribution conditioned upon the other $d-1$ components. In all instances, we use the most up-to-date information available on the remaining $d-1$ components, such that at each iteration, the conditional distributions from which we sample X_n change as we cycle through the d components, changing to reflect those components that have already been updated. The d conditional distributions are referred to in the literature as the full conditionals, and they are denoted using subscripts; at each

```
**Parameters**: $M$, $x_0$.
**Result**: A statistically dependent Markov chain $X_{0:M}$, where
        $X_t = (X_{t,1}, \ldots, X_{t,d})$, with limiting distribution equal to the target
        distribution $\pi(dx)$.
Initialization: Set $X_0 = x_0$.
for $t = 1, \ldots, M$ do
    Generate a permutation $\sigma_1, \ldots, \sigma_d$ of the indices $1, \ldots, d$.
    for $j$ = 1,...,$d$ do
        • Define $X_{t,-\sigma_j} = \left(X_{t,\sigma_1}, \ldots, X_{t,\sigma_{j-1}}, X_{t-1,\sigma_{j+1}}, \ldots, X_{t-1,\sigma_d}\right)$
        • Sample a value for $X_{t,\sigma_j}$ from the full conditional $\pi_{\sigma_j}\left(dx_{\sigma_j} \middle| X_{t,-\sigma_j}\right)$.
    end
end
```

Algorithm 2: The random-scan Gibbs sampler

iteration, for $j = 1, \ldots, d$, the components of X^* are sampled according to

$$
\begin{aligned}
X_1^* &\sim \pi_1\left(\,\cdot\,\middle|X_2, X_3 \ldots, X_d\right), \\
X_2^* &\sim \pi_2\left(\,\cdot\,\middle|X_1^*, X_2 \ldots, X_d\right), \\
X_3^* &\sim \pi_3\left(\,\cdot\,\middle|X_1^*, X_2^*, \ldots, X_d\right), \\
&\vdots \\
X_d^* &\sim \pi_d\left(\,\cdot\,\middle|X_1^*, X_2^*, \ldots, X_{d-1}^*\right).
\end{aligned}
$$

There are two commonly referenced variations of the Gibbs sampler: the deterministic scan and random scan samplers. The deterministic scan Gibbs sampler cycles through the components of the chain sequentially at each iteration, updating component j before moving on to component $j+1$, for $j = 1, \ldots, d-1$. In contrast, the random scan Gibbs sampler cycles through the d components in a random order at each iteration. The random scan Gibbs sampler is detailed in Algorithm 2; the deterministic scan version is a special case of this. It is remarked that the original procedure introduced by Geman and Geman (1984) was a version of the random-scan Gibbs sampler, where one component of the chain, chosen at random, was updated at each iteration.

2.2.4 Reversible Jump MCMC

Consider the case where the dimension of the space on which the target distribution $\pi(x)$ is defined, is not known prior to simulation. In this situation, the design of efficient proposals for use in MH-type procedures is nontrivial. We require a proposal kernel that can move the chain between spaces of differing dimension, and this can be achieved by treating the dimension of the space as a random variable. This is the approach adopted by the reversible-jump MCMC (RJMCMC) procedure, introduced by Green (1995).

Consider the case where we wish to target the distribution $\pi(dx)$ defined on the space E_d, where the dimension d is unknown. The RJMCMC procedure achieves this by sampling from the countable union of the dimension $d \geq 0$ and all possible spaces of dimension d:

$$\bar{E} = \bigcup_{d \geq 1} \left(\{d\} \times E_d \right),$$

with corresponding σ-algebra $\bar{\mathcal{E}}$.

The RJMCMC algorithm has a similar basic structure to the MH algorithm. At each iteration, one proposes a candidate X^* according to some proposal kernel $K\left(x, dx^*\right)$, and subsequently accepts the proposed value with probability given by an MH-type acceptance probability; otherwise, the proposed value is rejected. At each iteration, a proposal is made to move the chain from the state $x \in E_d$ to a state $x^* \in E_{d^*}$. The complication caused by the movement between spaces of differing dimension not only lies in the specification of the proposal kernel K, but in the subsequent careful calculation of the acceptance probability.

Specification of a Proposal Kernel

We begin by detailing the proposal mechanism. Suppose that, at the current iteration, our chain is such that $x \in E_d$, and we wish to move to a point $x^* \in E_{d^*} \subset \bar{E}$. The proposal kernel $K : E_d \times \bar{\mathcal{E}} \to [0, 1]$ can be represented as the countably infinite sum

$$K\left(x, dx^*\right) = \sum_{d^* \geq 1} c_{d^*} K_{d^*}\left(x, dx^*\right),$$

with $K_{d^*} : E_d \times \mathcal{E}_{d^*} \to [0, 1]$ the proposal kernel relating to the move from a d-dimensional space to a d^*-dimensional space and c_{d^*} a mixture coefficient, defined such that $\int_E K\left(x, dx^*\right) = 1$. Note that c_{d^*} could depend upon $x \in E_d$. Green (1995) considers the generalised case where the proposal kernel at each iteration is a probability submeasure, $\int_E K\left(x, dx^*\right) \leq 1$, allowing for the possibility that at any iteration, the procedure may propose no move, i.e. the chain stays in its current state. Here, we shall forego this generalisation, specifying K as a probability measure. This will also be the case considered in the application of this procedure to point processes in Chapter 5.

When proposing candidates according to K, one typically samples a candidate dimension and then subsequently a candidate state from the corresponding space, i.e. we rely on the decomposition of the proposal densities:

$$k_{d^*}\left(x^*|x\right) = p\left(d^*, x^*\left|d, x\right.\right) = p\left(d^*\left|d\right.\right) p\left(x^*\left|x, d^*\right.\right).$$

The Accept/Reject Step

The acceptance ratio is defined as before, as the Radon-Nikodym derivative of the probability measure $\pi(dx^*)K(x^*, dx)$ with respect to some dominating measure, divided by the Radon-Nikodym derivative of the probability measure $\pi(dx)K(x, dx^*)$ with respect to another dominating measure. Note that, for the MH procedures detailed in Section 2.2.3, these dominating measures were common, allowing us to immediately express the acceptance ratio in terms of the relevant target and proposal densities.

In general, $\pi(dx^*)K(x^*, dx)$ and $\pi(dx)K(x, dx^*)$ will not share a common dominating measure, due to the differing dimensions on which they are defined. In order to write the ratio in terms of the relevant densities, we include a Jacobian term, which accounts for this difference in dominating measures:

$$\alpha\left(\left\{d, x\right\}, \left\{d^*, x^*\right\}\right) = \frac{\pi\left(d^*, x^*\right)}{\pi\left(d, x\right)} \frac{p\left(d \left| d^* \right.\right) p\left(x \left| x^*, d \right.\right)}{p\left(d^* \left| d \right.\right) p\left(x^* \left| x, d^* \right.\right)} \left| \frac{\partial x^*}{\partial x} \right| \wedge 1.$$

A common problem that requires the use of RJMCMC considers the case where one is interested in simulating from a family of point processes, where the number of changepoints in the process up to a given time point is Poisson distributed. This problem will be studied in greater detail in Chapter 5, where we combine RJMCMC methods with sequential Monte Carlo (SMC) methods; the latter are the subject of the next section.

2.3 Sequential Monte Carlo

Sequential Monte Carlo (SMC) methods are a broad class of Monte Carlo procedures that build upon sequential importance sampling and resampling ideas and can be used to target a sequence of related probability measures $\pi_1(dx), \ldots, \pi_n(dx)$ of increasing dimension.

SMC methods can be described as algorithms that iteratively propagate a sample of N variables (referred to as *particles*) forward in time using a series of proposal kernels, whilst correcting for the discrepancy between the empirical distribution of the propagated sample and the target distribution through the use of importance weights. In general, SMC procedures also consider the possibility of redistributing the sample at each time step, according to these importance weights, in order to improve the quality of the empirical distribution at later time steps. The development of these ideas is explored in further detail throughout this section.

SMC algorithms were initially developed as methods for performing inference with respect to the sequence of target distributions that arise naturally in the filtering and smoothing problems; see, e.g. Doucet et al. (2000). SMC approaches to these problems are commonly referred to as particle filtering and particle smoothing.

2.3.1 Particle Filtering & Smoothing

The particulars of particle filtering and smoothing are discussed in this subsection, however as a precursor, we consider the Monte Carlo integration technique of importance sampling.

Importance Sampling

Once again, we consider the problem of evaluating expectations with respect to some non-trivial distribution $\pi(dx)$, which is assumed to admit the density $\pi(x)$ with respect to some dominating measure. In Subsection 2.2.1, we described the Monte Carlo estimator, constructed using a sample from the target distribution. When one is unable to obtain this sample directly from the target distribution, it was seen that Markov chain Monte Carlo methods could adequately facilitate MC integration. Importance sampling (IS) provides an alternative approach, and requires only the ability to pointwise evaluate $\pi(x)$ up to a normalizing constant. Furthermore, as we shall see below, in cases where direct sampling is possible, it can be shown that importance sampling has the potential to offer improvements in accuracy when performing sample-based inference. A concise introduction to this technique is given by Liu (2001) and Doucet et al. (2001).

Suppose one can sample from an auxiliary proposal distribution $\eta(dx)$ that dominates $\pi(dx)$, i.e. $\pi \ll \eta$. This auxiliary distribution is referred to as the importance distribution and we assume that it admits the density $\eta(x)$ with respect to the same reference measure as above. Under these assumptions, we can write the expectation of interest as

$$\mathbb{E}_\pi \left[f(X)\right] = \int_E f(x) \frac{\pi(x)}{\eta(x)} \eta(x) dx. \tag{2.5}$$

From this relation, it is clear that

$$\mathbb{E}_\pi \left[f(X)\right] = \mathbb{E}_\eta \left[\frac{\pi(X)}{\eta(X)} f(X)\right],$$

and so the expectation of interest can be approximated by the following sample mean:

$$\tilde{f}_{IS}^N(X) = \frac{1}{N} \sum_{i=1}^N \frac{\pi\left(X^{(i)}\right)}{\eta\left(X^{(i)}\right)} f\left(X^{(i)}\right), \tag{2.6}$$

for $X^{(1)}, \ldots, X^{(N)} \sim \eta(dx)$. This is an unbiased estimator, and its variance, taken with respect to the importance distribution, is defined as:

$$\begin{aligned} \mathrm{Var}_\eta \left[\tilde{f}_{IS}^N(X)\right] &= \frac{1}{N} \mathrm{Var}_\eta \left[\frac{\pi(X)}{\eta(X)} f(X)\right] \\ &= \frac{1}{N} \int \frac{\pi(x) f^2(x)}{\eta(x)} \pi(x) dx - \frac{1}{N} \mathbb{E}_\pi^2 \left[f(X)\right]. \end{aligned} \tag{2.7}$$

Consider the Monte Carlo estimator (2.1). This is clearly also unbiased, with variance given by

$$\mathrm{Var}_\pi \left[\hat{f}_\pi^N(X)\right] = \frac{1}{N} \int f^2(x) \pi(x) dx - \frac{1}{N} \mathbb{E}_\pi^2 \left[f(X)\right].$$

Thus, the difference between the variance of these two estimators is

$$\mathrm{Var}_\pi \left[\hat{f}_\pi^N(X)\right] - \mathrm{Var}_\eta \left[\tilde{f}_{IS}^N(X)\right] = \frac{1}{N} \int \left(1 - \frac{\pi(x)}{\eta(x)}\right) f^2(x) \pi(x) dx$$

and so an unbiased IS estimator can be designed with variance lower than that of the MC estimator. On the other hand, however, the presence of the ratio of densities in the integrand of (2.7) indicates that a poor choice of importance distribution, for example a distribution with lighter tails than the target distribution, will result in an unbounded variance (see, e.g. Robert and Casella, 2004).

An Alternative Importance Sampling Estimator

A requirement for the use of the unbiased IS estimator above is that the target density can be calculated exactly. Suppose instead that the target density is known only up to a normalizing constant:

$$\pi(x) = \frac{\gamma(x)}{Z}$$

In order to proceed with importance sampling in this scenario, one must use a sensible estimate for this normalizing constant. Motivated by (2.5), we see that Z can be represented as follows:

$$Z := \int_E \gamma(dx) = \int_E \frac{\gamma(x)}{\eta(x)} \eta(dx) = \mathbb{E}_\eta \left[\frac{\gamma(X)}{\eta(X)} \right].$$

Thus, a sensible estimator for the normalizing constant is simply the sample mean of the evaluated ratio of densities $\gamma(x)/\eta(x)$. This leads to an alternative importance sampling estimator, the self-normalized weighted sample mean. Define the unnormalized weight function to be

$$w(x) = \frac{\gamma(x)}{\eta(x)} \tag{2.8}$$

and, for a sample drawn from the importance distribution, $X^{(1)}, \ldots, X^{(N)} \sim \eta(dx)$, estimate the expectation of interest as the weighted sample mean, using the self-normalized weights $W^{(i)} \propto w\left(X^{(i)}\right)$, $\sum_{i=1}^N W^{(i)} = 1$:

$$\hat{f}_{IS}^N(X) = \sum_{i=1}^N W^{(i)} f\left(X^{(i)}\right) = \frac{N^{-1} \sum_{i=1}^N w\left(X^{(i)}\right) f\left(X^{(i)}\right)}{N^{-1} \sum_{i=1}^N w\left(X^{(i)}\right)}. \tag{2.9}$$

This alternative IS estimator is biased. According to the SLLN, however, the numerator will converge almost surely to the integral

$$\int_E f(x)\gamma(x)dx,$$

and the denominator will converge almost surely to the normalizing constant Z, also by the SLLN. Almost-sure convergence to the expectation of interest is therefore guaranteed for (2.9). Furthermore, some choices of $\eta(dx)$ lead to the mean squared error (MSE) of this estimator being less than that of $\tilde{f}_{IS}^N(X)$. To see this, consider the bias and variance of this biased IS estimator. Using a second-order Taylor series expansion of (2.9), about the expected values of the numerator and denominator and with respect to the importance distribution, yields the following result for the bias:

$$\mathbb{E}_\eta\left[\hat{f}_{IS}^N(X)\right] - \mathbb{E}_\pi\left[f(X)\right] = \frac{1}{N}\Big\{\mathbb{E}_\pi\left[f(X)\right]\mathrm{Var}_\eta\left[w(X)\right] - \mathrm{Cov}_\eta\left[w(X)f(X), w(X)\right]\Big\} \\ + O\left(N^{-2}\right),$$

and so the estimator (2.9) is asymptotically unbiased. One can also obtain an asymptotic result for the estimator variance. The common approach is to use the delta method (e.g. Casella and Berger, 2002), which consists of employing a Taylor series expansion as above, truncated to first-order terms.

$$\mathrm{Var}_\eta\left[\hat{f}_{IS}^N(X)\right] = \frac{1}{N}\Big\{\mathrm{Var}_\eta\left[w(X)f(X)\right] + \mathbb{E}_\pi^2\left[f(X)\right]\mathrm{Var}_\eta\left[w(X)\right] \\ - 2\mathbb{E}_\pi\left[f(X)\right]\mathrm{Cov}_\eta\left[w(X)f(X), w(X)\right]\Big\} + O\left(N^{-2}\right).$$

Combining the above results with the corresponding results for $\tilde{f}_{IS}^N(X)$, we have the following difference between the mean squared errors of the estimators:

$$MSE(\hat{f}_{IS}^{N}) = MSE(\tilde{f}_{IS}^{N}) + \frac{1}{N}\Big\{\mathbb{E}_{\pi}^{2}\left[f(X)\right]\mathrm{Var}_{\eta}\left[w(X)\right] \\ - 2\mathbb{E}_{\pi}\left[f(X)\right]\mathrm{Cov}_{\eta}\left[w(X)f(X), w(X)\right]\Big\} + O\left(N^{-2}\right),$$

and so, as noted by Liu (2001), the biased self-normalized IS estimator $\hat{f}_{IS}^{N}(X)$ has a lower MSE than its unbiased counterpart $\tilde{f}_{IS}^{N}(X)$ when $w(X)f(X)$ and $w(X)$ are highly correlated.

Choosing a Suitable Importance Sampling Estimator

As we have seen above, even though the self-normalized IS estimator $\hat{f}_{IS}^{N}(X)$ is biased, it can be designed such that its MSE is lower than that of the unbiased estimator $\tilde{f}_{IS}^{N}(X)$. This is achieved when the quantities $w(X)f(X)$ and $f(X)$ are highly correlated. The use of $\hat{f}_{IS}^{N}(X)$ has the added practical advantage that the importance distribution need only be known up to some normalizing constant. It is therefore preferable to use this biased IS estimator, choosing an importance distribution such that $w(X)f(X)$ and $f(X)$ are highly correlated. Henceforth, $\hat{f}_{IS}^{N}(X)$ will be referred to simply as the IS estimator.

Liu (1996, 2001) provides a 'rule of thumb' for choosing such importance distributions, and it is detailed here as it will motivate the definition of the *effective sample size* (ESS; Kong et al., 1994; Liu, 1996), which will be useful in designing efficient SMC methods. Consider the ratio of the variance of the IS estimator $\hat{f}_{IS}^{N}(X)$, taken with respect to the importance distribution, relative to the variance of the Monte Carlo estimator $\hat{f}_{\pi}^{N}(X)$, taken with respect to the target distribution. This gives a measure of the efficiency of the IS procedure relative to the direct sampling method. As shown by Liu (1996), the variance of $\hat{f}_{IS}^{N}(X)$ can be reduced to

$$\mathrm{Var}_{\eta}\left[\hat{f}_{IS}^{N}(X)\right] = \frac{1}{N}\mathrm{Var}_{\pi}\left[f(X)\right]\left\{1 + \mathrm{Var}_{\eta}\left[\frac{\gamma(X)}{\eta(X)}\right]\right\} + O\left(N^{-2}\right)$$

Thus, the required ratio takes the form

$$\frac{\mathrm{Var}_{\pi}\left[\hat{f}_{\pi}^{N}(X)\right]}{\mathrm{Var}_{\eta}\left[\hat{f}_{IS}^{N}(X)\right]} \approx \frac{\mathrm{Var}_{\pi}\left[f(X)\right]}{\mathrm{Var}_{\pi}\left[f(X)\right]\left\{1+\mathrm{Var}_{\eta}\left[\frac{\gamma(X)}{\eta(X)}\right]\right\}} \tag{2.10}$$

where the approximation is due to the negligence of the second-order terms in $\mathrm{Var}_{\eta}\left[\hat{f}_{IS}^{N}(X)\right]$. As noted by Liu (1996), the negligence of these higher order terms may not always be justified, however a large advantage to this efficiency measurement is the fact that it is independent of the test function f, and thus is a useful tool for comparing the efficiency of different IS procedures.

The above ratio indicates the size of the sample required for direct sampling, relative to that of the sample used for importance sampling, in order to achieve the same level of estimator variance. Thus, for an importance sampling procedure using the importance distribution η, the effective sample size (ESS) is defined as

$$ESS = \frac{N}{1+\mathrm{Var}_{\eta}\left[\frac{\gamma(X)}{\eta(X)}\right]}, \tag{2.11}$$

where N is the sample size used for IS. The ESS therefore takes a value in the closed interval $[1, N]$ and, in practice, an importance distribution for which the ESS is close to N may be sought. The ESS will prove useful in an SMC context, and this is detailed further below.

We consider now the sequential implementation of the IS method described above, with the aim being to target a sequence of related distributions $\pi_1(dx), \ldots, \pi_t(dx)$.

Sequential Importance Sampling for State Space Models

We now return to the SSM framework, given in (1.1)-(1.2), and present sequential importance sampling (SIS) in more detail. In the SSM context, our sequence of target distributions is the family of smoothing distributions, $\{\pi\left(dx_{1:t}|y_{1:t}\right)\}_{t\geq 1}$, such that $\pi(dx_{1:t}|y_{1:t})$ admits the joint smoothing density $\pi(x_{1:t}|y_{1:t})$ with respect to some common dominating measure.

Suppose that, at time t, one has access to the sample $\left\{X_{1:t-1}^{(i)}\right\}_{i=1}^{N}$, which is distributed according to the importance distribution $\eta\left(dx_{1:t-1}|y_{1:t-1}\right)$. Our aim is to construct a sample that is distributed according to $\pi(dx_{1:t}|y_{1:t})$, and we do so using the following SIS steps.

Using a proposal kernel $K_t(x_{t-1}, dx_t)$, we initially augment each of the existing particles $X_{1:t-1}^{(i)}$ with the additional point $X_t^{(i)} \sim K_t(X_{t-1}^{(i)}, \ \cdot \ )$. This defines the importance distribution at time t as

$$\eta(dx_{1:t}|y_{1:t}) := \eta(dx_{1:t-1}|y_{1:t-1})K_t(x_{t-1}, dx_t),$$

and we make the assumption that $\pi(dx_{1:t}|y_{1:t}) \ll \eta(dx_{1:t}|y_{1:t})$. We assume also that $\pi(dx_{1:t}|y_{1:t})$ and $\eta(dx_{1:t}|y_{1:t})$ admit their respective densities, $\pi(x_{1:t}|y_{1:t})$ and $\eta(x_{1:t}|y_{1:t})$ with respect to a common dominating measure. Note that the proposal kernel $K_t(x_{t-1}, dx_t)$ may, in general, also be dependent upon $x_{1:t-2}$ and $y_{1:t}$; for notational convenience, and without loss of generality, we denote the proposal kernel as above, reflecting only dependence upon the state at the previous time point.

Subsequent to propagation of the samples, weights can be calculated so as to reflect the extent to which each sample chain $X_{1:t}^{(i)}$ can be considered to be from the joint smoothing distribution:

$$W_t^{(i)} \propto \frac{\pi\left(X_{1:t}^{(i)}\,|y_{1:t}\right)}{\eta\left(X_{1:t}^{(i)}\,|y_{1:t}\right)}, \quad \sum_{i=1}^{N} W_t^{(i)} = 1. \tag{2.12}$$

Using these importance weights, referred to as the SIS weights, it is possible to approximate the smoothing distribution using a Monte Carlo estimator. The SMC approximation of the joint smoothing distribution is defined here as

$$\hat{\pi}\left(dx_{1:t}\,|y_{1:t}\right) = \sum_{i=1}^{N} W_t^{(i)} \delta_{X_{1:t}^{(i)}}\left(dx_{1:t}\right). \tag{2.13}$$

This approximation immediately allows us to define the SMC estimator for an expectation taken with respect to the joint smoothing density:

$$\hat{f}^N_{\pi_{1:t}}(X_{1:t}) := \hat{\mathbb{E}}_{\pi_{1:t}}\left[f(X_{1:t})\right] = \sum_{i=1}^{N} W_t^{(i)} f\left(X_{1:t}^{(i)}\right).$$

Similarly, the SMC estimator for an expectation taken with respect to the filtering density is defined as

$$\hat{f}^N_{\pi_t}(X_t) := \hat{\mathbb{E}}_{\pi_t}\left[f(X_t)\right] = \sum_{i=1}^{N} W_t^{(i)} f\left(X_t^{(i)}\right). \tag{2.14}$$

Recursive Calculation of the SIS Weights

As noted by Doucet et al. (2001), the SIS weights (2.12) can be calculated under the assumption that the joint importance density satisfies the relationship

$$\eta(x_{1:t}|y_{1:t}) = \eta(x_{1:t-1}|y_{1:t-1})k_t(x_t|x_{t-1}, y_t).$$

We also make use of the following identity concerning the joint distribution of the state process and observation record:

$$\pi\left(x_{1:t}, y_{1:t}\right) = \pi\left(x_{1:t-1}, y_{1:t-1}\right) q_t\left(x_t\left|x_{t-1}\right.\right) g_t\left(y_t\left|x_t\right.\right),$$

with q_t the prior transition density and g_t the likelihood, and we define

$$\gamma(x_{1:t}, y_{1:t}) = \pi(x_{1:t}, y_{1:t}),$$

such that the joint smoothing density may be written $\pi(x_{1:t}|y_{1:t}) \propto \gamma(x_{1:t}, y_{1:t})$. The resulting normalizing constant is the joint marginal density for the observed data, $Z_t = \pi\left(y_{1:t}\right)$.

The unnormalized weight functions $w\left(x_{1:t}\right)$ can now be defined as in (2.8):

$$\begin{aligned} w(x_{1:t}) = \frac{\gamma(x_{1:t}, y_{1:t})}{\eta(x_{1:t}|y_{1:t})} &= \frac{\gamma(x_{1:t-1}, y_{1:t-1})}{\eta(x_{1:t-1}|y_{1:t-1})} \cdot \frac{q_t(x_t|x_{t-1})g_t(y_t|x_t)}{k_t(x_t|x_{t-1}, y_t)}, \\ &= w(x_{1:t-1}) \cdot \tilde{w}(x_{t-1}, x_t), \end{aligned} \tag{2.15}$$

where $\tilde{w}\left(x_{t-1}, x_t\right)$ is referred to as the incremental weight function.

Using (2.15), we can recursively calculate the weights $\left\{W_t^{(i)}\right\}_{i=1}^{N}$, as

$$W_t^{(i)} \propto W_{t-1}^{(i)} \cdot \tilde{w}\left(X_{t-1}^{(i)}, X_t^{(i)}\right), \quad i = 1, \ldots, N.$$

Thus, at each time step t, one only requires access to $\left\{X_{t-1:t}^{(i)}, W_{t-1}^{(i)}\right\}_{i=1}^{N}$. This has the important practical consequence of allowing sequential estimation via the SMC estimators (2.13)-(2.14), without requiring the storage of $\left\{X_{1:t-2}^{(i)}, W_{1:t-2}^{(i)}\right\}_{i=1}^{N}$.

Similarly to before, the normalizing density $\pi\left(y_{1:t}\right)$ can be estimated using the weights. At time t, given the normalized sample weights at the previous iteration and the sampled particles at the previous and current iterations, $\left\{X_{t-1:t}^{(i)}, W_{t-1}^{(i)}\right\}_{i=1}^{N}$, the ratio of consecutive normalizing densities $\pi(y_{1:t})/\pi(y_{1:t-1})$ can be estimated by

$$\frac{1}{N}\sum_{i=1}^{N} W_{t-1}^{(i)} \tilde{w}\left(X_{t-1}^{(i)}, X_t^{(i)}\right).$$

This estimator enables recursive estimation of the marginal likelihood of the data, which will be of great use in the implementation of PMCMC methods (see Section 2.4). The resulting estimator for the marginal likelihood normalizing constant $\pi(y_{1:t})$ is as follows, and it is noted that this is an unbiased estimator:

$$\hat{\pi}(y_{1:t}) = \prod_{n=1}^{t} \frac{1}{N} \sum_{i=1}^{N} W_{n-1}^{(i)} \tilde{w}\left(X_{n-1}^{(i)}, X_n^{(i)}\right).$$

The Weight Degeneracy Issue

In batch inference scenarios, where MCMC methods may also be used, it may be argued that the sequential importance sampling approach is preferable; the generated samples here are independent and identically distributed, resulting in potentially more efficient estimators. The major disadvantage of the SIS approach, however, is that the variance of the sample weights, conditional on the observed data, follows an increasing trend with respect to the time parameter. This is due to the following result (Kong et al., 1994):

$$\mathbb{E}_{\pi(y_{1:t})}\Big[\mathrm{Var}_{\pi_{1:t}}\big[w(x_{1:t})|y_{1:t}\big]\Big] \geq \mathbb{E}_{\pi(y_{1:t-1})}\Big[\mathrm{Var}_{\pi_{1:t-1}}\big[w(x_{1:t-1})|y_{1:t-1}\big]\Big].$$

This was first reported in the context of Bayesian filtering by Doucet et al. (2000), and has the effect of reducing the accuracy of the filtering estimates (2.14).

In practice we find that, due to the recursive nature of the relation in (2.15), any area of concentration in the distribution of the weights over the sample is compounded at each time step. Combined with the increasing variance of the sample weights, the effect is that after no more than a few time steps, there are typically very few states with non-zero weighting.

Resampling

A major breakthrough for SMC methods was introduced by Gordon et al. (1993). The most significant contribution of this paper was to introduce a resampling step to the SIS procedure, inspired by a result from Smith and Gelfand (1992). Based on the bootstrap methodology of Efron (1982), the principle is that, once a sample has been collected from a distribution, one obtains a more favourable sample by assuming this to be an empirical representation of the distribution and sampling from it, with replacement. The original sample is then discarded and replaced with the bootstrap sample. This has the effect of replicating desirable elements of the sample, at the acceptable cost of discarding the most unsuitable samples.

If we are resampling at every step, then the new procedure at time t, referred to as sequential importance resampling (SIR), is as follows:

- Given $\left\{X_{1:t-1}^{(i)}\right\}_{i=1}^{N}$, generate a sample $\left\{X_{t}^{(i)}\right\}_{i=1}^{N}$,
- For $i = 1, \ldots, N$, set $W_{t-1}^{(i)} = \frac{1}{N}$ and calculate $W_t^{(i)}$ according to (2.15),
- For $j = 1, \ldots, N$, choose index i with probability $W_t^{(i)}$ and set $X_{1:t}^{*(j)} = X_{1:t}^{(i)}$,
- Re-define $\left\{X_{1:t}^{(i)}\right\}_{i=1}^{N} = \left\{X_{1:t}^{*(i)}\right\}_{i=1}^{N}$, discarding the original values of $\left\{X_{1:t}^{(i)}\right\}_{i=1}^{N}$.

Since the weights are reset at each step, and are calculated only up to a constant of proportionality, we can discard $W_{t-1}^{(i)}$ from relation (2.15), when using it in the above procedure. Note that this is only true when resampling occurs at every time step.

The resampling mechanism detailed above is known as multinomial resampling, and is the resampling scheme with which the first particle filters were proposed (Gordon et al., 1993). Other resampling schemes are available, such as stratified resampling (Kitagawa, 1996) and residual resampling (Liu and Chen, 1998). Both of these alternative methods draw particles from the empirical distribution $\hat{\pi}(dx_{1:t})$ (2.13), however both are designed such that the number of replicates of a particular particle $X_{1:t}^{(i)}$ after resampling is approximately equal to $\left[NW_t^{(i)}\right]$, where $[x]$ is the integer part of x. It is noted that the use of either stratified or residual resampling can be shown to offer improvements over multinomial resampling in the variance of the resulting SMC estimators (see e.g. Chopin, 2004; Douc et al., 2005). We will concentrate on the use of a multinomial resampling scheme within the SMC procedures presented here, in order to facilitate a more principled comparison with the use of RSMC methods, which are introduced in Section 2.3.2; this will be explained further in that section.

In resetting the weights, we also discard any information already collected about the distribution of the weights, and so degeneracy of the sample weights is no longer as much of an issue. With the introduction of resampling, however, comes another problem: particle path degeneracy.

The Particle Path Degeneracy Issue

As we are resampling with replacement according to the weights, at each resampling step the least informative particles may be discarded and the more informative particles replicated. Note that it is the entire history of each particle $X_{1:t}$ that is subject to possible replacement. As a result, the number of distinct paths in the particle system's history diminishes with each resampling step; the cumulative effect of this is that the majority of the particles in the system share the same history, even after only a modest number of iterations. This is well documented as the particle path degeneracy issue (see e.g. Del Moral et al., 2009; Doucet and Johansen, 2011) and after a sufficient time t, there will be an integer k such that the paths $x_{1:t-k}^{(i)}$ will be the same for all $i \in \{1, \ldots, N\}$. Figure 2.1 illustrates this degeneracy for the first 30 time steps of a SIR procedure targeting the unobserved state in a linear Gaussian SSM. It is noted that as a result of particle path degeneracy, it is not possible to accurately approximate the distribution of $X_{1:t}$ for increasing length t. It is possible at time t to approximate the distribution of $X_{t-k:t}$, however we must have fixed length k of small magnitude.

The convergence properties of various sequential importance sampling based methods, both with and without resampling, are examined by both Del Moral (2004) and Chopin (2004), and the reader is directed to these references for further discussion on these convergence properties.

Resampling allows us, to an extent, to deal with the issue of weight degeneracy. From this alone, a practitioner might deduce that resampling is preferable in particle filtering. On the other hand, however, we have seen above that resampling leads to particle path degeneracy. Furthermore, resampling adds to the computational expense of the SMC procedure, and so it makes sense to avoid excessive resampling.

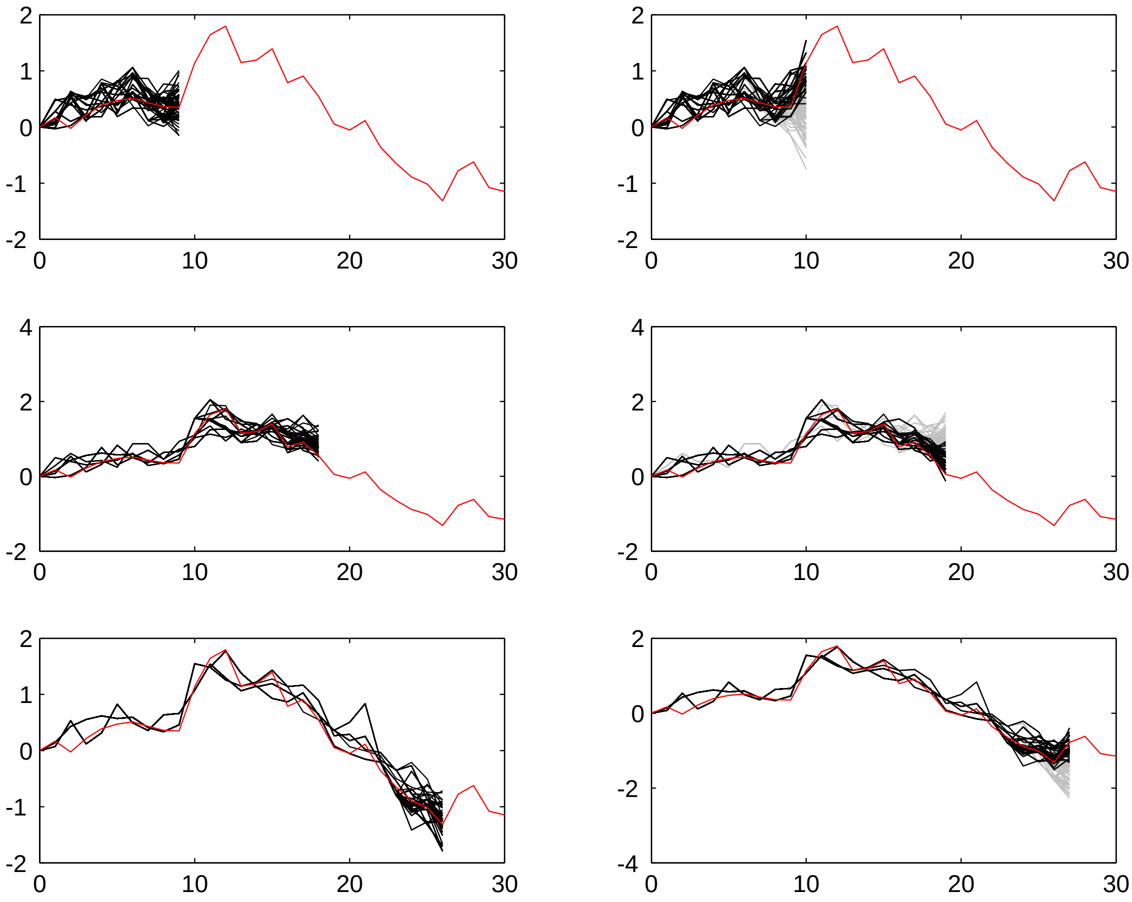


Figure 2.1: Particle path degeneracy for a SIR procedure. In each row, the left hand plot shows the particles in existence before propagation via a proposal kernel; the right hand plot shows the augmented particles, with the particles in grey subsequently being discarded upon resampling. The true state being estimated is given by the red line.

Thus, we have a trade off that we must manage. In order to effectively do so, we turn to the effective sample size (2.11). At each time step, after assigning importance weights to the propagated sample, the ESS is estimated according to

$$\widehat{ESS}\left(\left\{W_t^{(i)}\right\}_{i=1}^{N}\right) = \left(\sum_{i=1}^{N}\left(W_t^{(i)}\right)^2\right)^{-1}, \tag{2.16}$$

and, conditional on $\widehat{ESS}$ being less than some pre-determined threshold, resampling is performed according to the chosen resampling scheme.

Adding this adaptive resampling step to the SIS steps described above provides

Data: $y_{1:T}$.
Parameters: x_0, T, N.
Result: The SMC estimates of $\mathbb{E}_{\pi_t}\left[f(X_t)\right]$, for $t = 1, \ldots, T$.

1. Set $t = 1$. For $i = 1, \ldots, N$, sample $X_t^{(i)} \sim K_t\left(x_0, dx_t\right)$, and compute

$$W_t^{(i)} \propto \frac{g_t\left(y_t \left| X_t^{(i)}\right.\right) q_t\left(X_t^{(i)} | x_0\right)}{k_t\left(X_t^{(i)} | x_0\right)}, \quad \sum_{i=1}^{N} W_t^{(i)} = 1.$$

2. If $\widehat{ESS}\left(\left\{W_t^{(i)}\right\}_{i=1}^{N}\right) < \frac{N}{2}$, then, for $i = 1, \ldots, N$, resample $X_t^{(i)}$ independently from the discrete distribution

$$\hat{\pi}\left(dx_t\right) = \sum_{i=1}^{N} W_t^{(i)} \delta_{X_{1:t}^{(i)}}\left(dx_t\right)$$

and set $W_t^{(i)} = \frac{1}{N}$.

3. Set $t = t + 1$. For $i = 1, \ldots, N$, sample $X_t^{(i)} \sim K_t\left(X_{t-1}^{(i)}, dx_t\right)$ and calculate the filtering weights

$$W_t^{(i)} \propto W_{t-1}^{(i)} \cdot \frac{g_t\left(y_t \left| X_t^{(i)}\right.\right) q_t\left(X_t^{(i)} \middle| X_{t-1}^{(i)}\right)}{k_t\left(X_t^{(i)} \middle| X_{t-1}^{(i)}\right)}, \quad \sum_{i=1}^{N} W_t^{(i)} = 1.$$

4. Calculate the particle filtering estimate $\hat{f}_{\pi_t}^{N}(X_t)$ according to (2.14).
   If $t = T$, stop; otherwise, return to Step 2.

Algorithm 3: An SMC Filtering Algorithm

the SMC procedure that will be used here for particle filtering and particle smoothing. The full particle filtering procedure is detailed in Algorithm 3, where it is noted that the SMC filtered estimate of the expected value of a given test function is calculated prior to resampling at each step.

Choice of Proposal Density

As seen above, one can use any proposal distribution to generate particles, as long as its positive support encompasses that of the target distribution and one is able to pointwise evaluate it up to some normalizing constant. Of course, for computational efficiency, an easy-to-sample proposal may be chosen, however, this inevitably involves a trade-off with how well it mirrors the target distribution. A large discrepancy between proposal and target will result in large variance of the importance weights. As shown by Doucet et al. (2000), the proposal density that minimizes the variance of the importance weights is

$$k_t(x_t|x_{t-1}, y_t) = \frac{q_t(x_t|x_{t-1})g_t(y_t|x_t)}{\int q_t(x_t|x_{t-1})g_t(y_t|x_t)dx_t}.$$

Use of this proposal density within an SIS framework requires both the ability to sample according to $k_t(x_t|x_{t-1}, y_t)$, i.e. with knowledge of the data at the current time point, and the ability to evaluate or estimate the normalizing constant $\int q_t(x_t|x_{t-1})g_t(y_t|x_t)dx_t$. This normalizing constant will not have an analytical form in the general case, e.g. where the transition or observation noise in the SSM are non-Gaussian. Furthermore, when considering filtering and smoothing in this thesis, interest will lie principally in scenarios in which the likelihood $g_t(y_t|x_t)$ can neither be evaluated nor replaced with an unbiased estimator and so use of the conditionally optimal proposal will therefore not be suitable here. For this reason, all SMC filtering and smoothing procedures considered here will use the transition density $q_t(x_t|x_{t-1})$ as the proposal density. It is noted that this is a sub-optimal proposal mechanism, as particles are proposed with no knowledge of the observed data at the current time point. However, since the incremental weight functions under this proposal are reduced to the likelihood function, this allows us to focus on the role of the likelihood in the accuracy of the resulting filtering and smoothing estimates, and the effect of estimating it through the use of ABC methodology, which will be introduced in Section 2.5. The use of this proposal, and the subsequent simplification of the importance weights, has the added advantage of reducing the computational budget of all SMC algorithms considered.

2.3.2 Rejection SMC

We consider here a modification to the SMC methodology that incorporates a rejection-based approach to resampling at each time step of the SMC procedure. This approach to SMC was introduced by Del Moral (2004), and it is noted that, since it is independent of the target distribution, it can be applied to both the filtering procedure and the smoothing procedures considered in Chapter 4. This rejection-based approach to SMC will henceforth be referred to as Rejection SMC (RSMC).

The RSMC procedure is initialised as before, and at each time step t, after calculating the incremental importance weights, the following acceptance probabilities are assigned to the particles: for $i = 1, \ldots, N$,

$$\beta_t^{(i)} = \beta_t \left(X_{t-1:t}^{(i)} | y_t \right) := \frac{\tilde{w} \left(X_{t-1}^{(i)}, X_t^{(i)} \right)}{\sup\limits_{x_t \in \mathbb{R}^{d_x}} \left\{ \tilde{w} \left(X_{t-1}^{(i)}, x_t \right) \right\}} \leq 1, \tag{2.17}$$

under the assumption that the supremum in the denominator exists for any fixed y_t. With probability $\beta_t^{(i)}$, the particle $X_{1:t}^{(i)}$ is not resampled, otherwise it is resampled according to the particle approximation of the smoothing distribution, $\hat{\pi}\left(dx_{1:t}\right)$, given in (2.13). The weights of all particles are subsequently set to $1/N$. This procedure is detailed in full in Algorithm 4.

This procedure is motivated through the use of the rejection kernel of Del Moral

```
Data: $y_{1:T}$.
Parameters: $x_0$, $T$, $N$.
Result: The TRSMC estimates of $\mathbb{E}_{\pi_t}\left[f(X_t)\right]$, for $t = 1, \ldots, T$.

1. Set $t = 1$. For $i = 1, \ldots, N$, sample $X_t^{(i)} \sim K_t\left(x_0, dx_t\right)$, and compute
       $W_t^{(i)} \propto \frac{g_t\left(y_t \middle| X_t^{(i)}\right) q_t\left(X_t^{(i)} | x_0\right)}{k_t\left(X_t^{(i)} | x_0\right)}, \quad \sum_{i=1}^{N} W_t^{(i)} = 1.$
2. For $i = 1, \ldots, N$, calculate $\beta_t^{(i)}$ using (2.17), and with probability $\beta_t^{(i)}$ do not
   resample, otherwise resample $X_t^{(i)}$ according to the discrete distribution
       $\hat{\pi}\left(dx_{1:t}\right) = \sum_{i=1}^{N} W_t^{(i)} \delta_{X_{1:t}^{i}}\left(dx_{1:t}\right).$
3. For $i = 1, \ldots, N$, set $W_t^{(i)} = \frac{1}{N}$.
4. Set $t = t + 1$. For $i = 1, \ldots, N$, sample $X_t^{(i)} \sim K_t(X_{t-1}^{(i)}, dx_t)$ and calculate
   the filtering weights
       $W_t^{(i)} \propto W_{t-1}^{(i)} \cdot \frac{g_t\left(y_t \middle| X_t^{(i)}\right) q_t\left(X_t^{(i)} \middle| X_{t-1}^{(i)}\right)}{k_t\left(X_t^{(i)} \middle| X_{t-1}^{(i)}\right)}, \quad \sum_{i=1}^{N} W_t^{(i)} = 1.$
5. Calculate the particle filtering estimate $\hat{f}_{\pi_t}^{N}(X_t)$ according to (2.14).
   If $t = T$, stop; otherwise, return to Step 2.
```

Algorithm 4: A Theoretical Rejection SMC Filtering Algorithm

(2004), which is defined here using current notation as

$$H_t\left(\left(\{x_{t-1}^{(i)}\}_{i=1}^{N}, y_{t-1}\right), dx_t\right) = \beta_t\left(x_t^{(i)} | y_t\right) K_t(x_{t-1}^{(i)}, dx_t) + \left[1 - \beta_t\left(x_t^{(i)} | y_t\right)\right] \sum_{j=1}^{N} \frac{\tilde{w}(x_{t-1}^{(j)}, x_t^{(j)})}{\sum_{l=1}^{N} \tilde{w}(x_{t-1}^{(l)}, x_t^{(l)})} K_t(x_{t-1}^{(j)}, dx_t). \tag{2.18}$$

It has been shown by Del Moral (2004) that the asymptotic (as $N \to \infty$) variance in the CLT associated with estimates resulting from the SMC approximation of the joint smoothing distribution (2.13) is smaller when the approximation is obtained using an RSMC procedure than when using a SIR algorithm, i.e. an SMC procedure that resamples at every time step. It will be of interest here to see if this advantage is realised for finite N, in particular when comparing against an SMC procedure that uses dynamic resampling. The motivating result of Del Moral (2004) corresponds to an SIR procedure that uses a multinomial resampling scheme; we therefore also use multinomial resampling for our dynamic resampling SMC procedures.

Consider the definition of the acceptance probabilities in (2.17). This definition is made under the assumption that the supremum in the denominator is available and finite, an assumption that will not always be satisfied. We therefore consider also a version of RSMC that is robust to this assumption. We use the same procedure as described above, with acceptance probabilities defined instead using the empirical maximum incremental weight calculated over all particles, in place of the theoretical supremum over all x_t; that is

$$\hat{\beta}_t^{(i)} = \hat{\beta}_t\left(X_t^{(i)}|y_t\right) := \frac{\tilde{w}\left(X_{t-1}^{(i)}, X_t^{(i)}\right)}{\max_i\left\{\tilde{w}\left(X_{t-1}^{(i)}, X_t^{(i)}\right)\right\}} \leq 1. \tag{2.19}$$

It is stressed that $\hat{\beta}_t^{(i)}$ is not intended as an estimator for $\beta_t^{(i)}$, simply an empirical alternative. Moreover, note that neither $\beta_t^{(i)}$ nor $\hat{\beta}_t^{(i)}$ require online estimation.

Henceforth, in order to distinguish between the different forms of the acceptance probability being used, we refer to RSMC using $\beta_t^{(i)}$ as Theoretical Rejection SMC (TRSMC) and to RSMC using $\hat{\beta}_t^{(i)}$ as Empirical Rejection SMC (ERSMC). It is remarked that, as N increases, one would expect the standard errors of the estimates produced by the ERSMC procedures to approach the standard errors for the corresponding estimates produced by the TRSMC procedures. This is due to the fact that, as $N \to \infty$, the empirical maximum of the observed incremental weights at time t approaches the theoretical supremum over all proposed states x_t, thus

$\hat{\beta}_t^{(i)} \to \beta_t^{(i)}$ as $N \to \infty$. It is also remarked that the use of RSMC avoids the need to choose either the resampling criterion or threshold, which are associated with the dynamic resampling SMC procedure. Since the use of the ESS is based upon the approximation (2.10), which is not always justified, the use of an automated RSMC procedure may be preferable.

2.3.3 Sequential Monte Carlo Samplers

Sequential Monte Carlo methods may also be used for performing inference with respect to complex distributions outside of the SSM framework. Consider the distribution $\pi(dx)$ defined upon the measurable space $(E, \mathcal{E})$. SMC samplers, formalised by Del Moral et al. (2006), can be used to target this probability measure by approximating a sequence of intermediary measures $\{\pi_n\}_{n=1}^m$ such that $\pi_m(dx) = \pi(dx)$. The advantage of SMC samplers over alternative SMC methodology is that they are geared towards targeting a sequence of measures $\{\pi_n\}_{n=1}^m$ defined upon the common space $(E, \mathcal{E})$, as opposed to targeting measures defined upon a sequence of nested spaces, $\{(E_n, \mathcal{E}_n) \ ; \ E_{n-1} \subseteq E_n\}_{n=1}^m$, the latter being the approach of particle filters. This common-space approach will be advantageous, for example, in performing inference with respect to latent point process models, as will be discussed later. We provide a review of the framework associated to SMC samplers.

Proceeding as in Section 2.3.1, we could target $\{\pi_n\}_{n=1}^m$ via a sequential importance sampling scheme. At iteration n, given N particles $\left\{X_{n-1}^{(i)}\right\}_{i=1}^N$ assumed to be distributed according to $\eta_{n-1}(dx_{n-1})$, each is perturbed according to the Markov kernel $K_n\left(x_{n-1}, dx_n\right)$ with corresponding density $k_n\left(x_n \left| x_{n-1}\right.\right)$. The resulting particles are subsequently marginally distributed according to the proposal distribution

$$\eta_n\left(dx_n\right) = \int_E \eta_{n-1}\left(dx_{n-1}\right) K_n\left(x_{n-1}, dx_n\right).$$

Under the assumption that the density $\eta_n(x_n)$ is available to compute pointwise, importance weights and expectations of interest may be calculated via straightforward IS calculations. However, in reality, the integral

$$\eta_n(x_n) = \int_{E^{n-1}} \eta_1(x_1) \left(\prod_{j=2}^{n} k_j(x_j \,| x_{j-1}) \right) dx_{1:n-1}. \tag{2.20}$$

is typically high-dimensional and impossible to compute. One potential solution is to approximate $\eta_n(x_n)$ via

$$\eta_{n-1}^{N} k_n(x_n) := \frac{1}{N} \sum_{i=1}^{N} k_n\left(x_n \left| X_{n-1}^{(i)}\right.\right),$$

however the computational complexity of the resulting algorithm would be $O(N^2)$, as the evaluated importance density $\eta_n\left(X_n^{(j)}\right)$ would need to be approximated by the above, for $j = 1, \ldots, N$; this computational budget is prohibitive. In addition, it is not always possible to pointwise evaluate $k_n(x_n \,| x_{n-1})$ and so this approximation might not be possible.

SMC samplers provide an alternate approach to this problem that maintains the attractive feature of the SIS framework, without having to evaluate (2.20). Del Moral et al. (2006) propose an auxiliary variable technique, introducing a so-called *backward kernel* $L_{n-1}(x_n, dx_{n-1})$ to mirror the effect of the Markov kernel K_n; denote the corresponding density $l_{n-1}(x_{n-1} \,| x_n)$.

Del Moral et al. then perform sequential importance sampling using the joint importance density $\eta_n(x_{1:n})$ to obtain a sample from the (auxiliary) joint target density

$$\tilde{\pi}_n(x_{1:n}) = \frac{\tilde{\gamma}_t(x_{1:n})}{Z_n} \tag{2.21}$$

where we define

$$\tilde{\gamma}_n(x_{1:n}) = \gamma_n(x_n) \prod_{k=2}^{n} l_{k-1}(x_{k-1} \,| x_k). \tag{2.22}$$

At iteration n, each of the particle paths $\left\{X_{1:n-1}^{(i)}\right\}$ is propagated according to the

Markov kernel $K_n(x_{n-1}, dx_n)$ and weighted according to $W_n^{(i)} \propto w\left(X_{1:n}^{(i)}\right)$, using the weighting function $w(x_{1:n})$, which measures the discrepancy between $\tilde{\gamma}_n(x_{1:n})$ and $\eta_n(x_{1:n})$:

$$\begin{aligned} w(x_{1:n}) &= \frac{\tilde{\gamma}_n(x_{1:n})}{\eta_n(x_{1:n})} \\ &= \frac{\tilde{\gamma}_{n-1}(x_{1:n-1})\,\gamma_n(x_n)\,l_{n-1}(x_{n-1}\,|x_n)}{\gamma_{n-1}(x_{n-1})} \cdot \frac{1}{\eta_{n-1}(x_{1:n-1})\,k_n(x_n\,|x_{n-1})} \\ &= w(x_{1:n-1}) \cdot \frac{\gamma_n(x_n)\,l_{n-1}(x_{n-1}\,|x_n)}{\gamma_{n-1}(x_{n-1})\,k_n(x_n\,|x_{n-1})}. \end{aligned} \tag{2.23}$$

Finally, since $\tilde{\pi}_n(x_{1:n})$ admits $\pi_n(x_n)$ as a marginal, the final weighted sample $\left\{\left(X_n^{(i)}, W_n^{(i)}\right)\right\}$ is an approximation of the original target density by construction.

This method shifts focus from directly sampling from the marginal densities of interest to sampling from the sequence of joint densities $\{\tilde{\pi}_n\}_{n=1}^m$, with respective nested supports $\{E_n\}_{n=1}^m$; it is clear that this method does not operate entirely within a common measurable space at each time step. It is worth noting, however, that the intention throughout is to perform inference with respect to the marginal density π_n, which *does* remain defined upon a common space at each time step. Also, since the weights may be calculated recursively, with the incremental weight dependent upon the (unnormalized) marginal densities, and not the full joint densities, it is intuitive that SMC samplers are, in general, of fixed computational complexity. Indeed, the standard SMC sampler algorithm, given by Del Moral et al. (2006), has computational complexity $O(N)$.

2.4 Particle MCMC

This section provides a review of particle MCMC (PMCMC) methodology. PMCMC methods were introduced by Andrieu et al. (2010) as a set of techniques for performing batch inference that use elements of SMC methodology to address obstacles faced in the application of standard MCMC methods to complex problems.

We detail here the application of PMCMC methods to the filtering and smoothing problems, but note that the methodology is much more general and can be applied to a wide variety of problems outside the SSM framework.

2.4.1 MCMC for State Space Models

Consider the application of MCMC methods to the generalised state space model, with transition and observation densities (1.1)-(1.2) parameterised by $\theta \in \Theta$, and note that interest here lies in performing batch inference on the joint smoothing density $\pi_\theta(x_{1:T}|y_{1:T})$. The PMCMC methods used in this work are based upon the MH procedure, as detailed in Section 2.2.3. It is noted that PMCMC methods can also be constructed using the Gibbs sampler, however these are not considered here; details on the particle Gibbs procedure can be found in Andrieu et al. (2010).

Typically, the length of the state, T, is too large for straightforward MH procedures to be efficient: high-dimensional proposals are generally too cumbersome to sample from and dimension-reduced alternatives such as local or single-site update schemes result in chains that suffer from slow mixing properties, especially when there is a strong dependence structure within $(x_{1:T}, \theta)$. Further difficulty lies in the situation where point-wise evaluation of the joint smoothing density up to a constant is impossible, as the calculation of the acceptance probability is infeasible in this case.

PMCMC helps to deal with issues of difficulty associated with both high-dimensionality and intractability of the target density. The basic premise is to use an SMC procedure as the proposal mechanism in an MCMC algorithm, in order to propose suitable candidates from the joint smoothing distribution. The candidate value proposed by the SMC scheme is then accepted or rejected according to an MH acceptance probability, which may also be estimated as a by-product of the SMC procedure. Andrieu et al. (2010) use a particle filter, as defined in Section 2.3.1, drawing candidate states $X^*_{1:T}$ from the SMC approximation of the joint smoothing distribution given in (2.13). Of course, the particle path degeneracy issue renders this a poor approximation of the joint smoothing distribution. As is also noted by Andrieu et al. (2010), however, the PMCMC procedure does not require the candidate resulting from the SMC procedure to be an accurate representation of the smoothed state, merely a single sampled path from the target smoothing distribution.

This forms the basis of a Particle IMH (PIMH) procedure, which we now detail further. To clarify the discussion, we develop the notation. Previously, the proposal densities for SMC procedures and MCMC procedures have both been denoted $k\left(x^{*}|x\right)$; for clarity, we will discriminate through use of subscripts, using k_{smc} and k_{mh} respectively.

2.4.2 Particle Independent Metropolis-Hastings (PIMH)

The procedure described above corresponds to an IMH algorithm, since the particle filter used to generate the candidate state $x^{*}_{1:T}$ is independent of the current value of the Markov chain. Hence, this is referred to as a Particle IMH (PIMH) procedure. We now give the details of the procedure, specifically the proposal density and the resulting MH acceptance probability.

PIMH Target Density and Proposal Density

Framing the problem as one of sampling from the collection of realised particles $\left\{X^{(i)}_{1:T}\right\}_{i=1}^{N}$ results in a proposal density that does not typically have a simple analytical form. As an alternative approach, Andrieu et al. (2010) frame the problem as one of sampling from the extended space traversed by all variables generated by the SMC procedure, i.e. the N candidate values for the particles generated at each time step and the indices of the particles chosen to propagate at each resampling step. As we will explain below, this allows a closed form representation of the proposal density, and thus a tractable acceptance probability.

As PIMH is not used explicitly within this thesis, full technical details of the target and proposal densities are omitted, however we do give a brief description of the steps taken to arrive at these densities, as well as the basic form of the densities and how they contribute to the resulting MH ratio; for full details, see Andrieu et al. (2010). To clarify the discussion below, we require a round of notation for the variables generated in the SMC procedure. Denote the N-vector of state variables generated at each time point t by $\bar{X}_t = \left(\bar{X}_t^1, \ldots, \bar{X}_t^N\right) \in \mathbb{R}^{Nd_x}$, and the indices of those particles chosen during resampling at t by $\bar{a}_t = \left(\bar{a}_t^1, \ldots, \bar{a}_t^N\right) \in \{1, \ldots, N\}^N$. We further denote the joint density of all variables generated by the SMC procedure as $\psi\left(\bar{x}_{1:T}, \bar{a}_{1:T-1}\right)$.

Andrieu et al. (2010) show that, by sampling and storing $\bar{X}_{1:T}$ and $\bar{a}_{1:T}$, one can subsequently, *deterministically*, reconstruct a particular particle given its final index. As a result, subsequent to running the SMC procedure at each MH iteration, the problem of sampling from the collection of realised particles $\left\{X_{1:T}^{(i)}\right\}_{i=1}^N$ can be reduced to sampling a final index, j, from the empirical distribution provided by the final weights. Thus, the proposal density is given by the product

$$k_{\text{mh}}(\bar{X}_{1:T}, \bar{a}_{1:T-1}, j) = \psi\left(\bar{X}_{1:T}, \bar{a}_{1:T-1}\right) \cdot W_T^{(j)}.$$

The target density is also constructed using the joint density $\psi\left(\bar{x}_{1:T}, \bar{a}_{1:T-1}\right)$. To identify this target, we note that, at each MH iteration, given all the variables generated by the SMC procedure, *each* of the possible reconstructed particles is distributed according to the joint smoothing distribution $\pi_\theta(x_{1:T}|y_{1:T})$. Thus, the PIMH target density can be constructed as the product of

- the joint density of the variables generated by the SMC procedure, conditioned upon the union of the chosen particle index j and the values of the variables in $\left\{\bar{X}_{1:T}, \bar{a}_{1:T-1}\right\}$ that contribute to the realised particle $X_{1:T}^{(j)}$,
- the density targeted by the SMC procedure, evaluated at the generated state values in $\bar{X}_{1:T}$ that contribute to $X_{1:T}^{(j)}$, i.e. the joint smoothing density evalu-

ated at the realised particle, $\pi\left(X_{1:T}^{(j)}|y_{1:T}\right)$,

- the probability of generating the indices in $\bar{a}_{1:T-1}$ that contribute to $X^{(j)}$, i.e. $N^{-(T-1)}$,
- and the probability of choosing the final particle index j, i.e. N^{-1}.

PIMH Acceptance Ratio

Andrieu et al. (2010) show that, when using the above forms of the target and proposal densities, the ratio of these quantities is equal to $\hat{\pi}_\theta(y_{1:T})/\pi_\theta(y_{1:T})$. Thus the acceptance ratio can be reduced to the simple form

$$\alpha\left(x_{1:T}, x_{1:T}^*\right) = \frac{\hat{\pi}_\theta^*(y_{1:T})}{\hat{\pi}_\theta(y_{1:T})} \wedge 1,$$

where $\hat{\pi}_\theta(y_{1:T})$ is the current estimate of the marginal likelihood and $\hat{\pi}_\theta^*(y_{1:T})$ is the estimate provided by the SMC proposal scheme used to generate $X_{1:T}^*$.

The full PIMH procedure is given in Algorithm 5 .

2.4.3 Particle Marginal Metropolis-Hastings (PMMH)

We consider the problem of sampling from the joint posterior

$$\pi\left(x_{1:T}, \theta\,|y_{1:T}\right) = \pi_\theta\left(x_{1:T}\,|y_{1:T}\right)\pi\left(\theta\,|y_{1:T}\right),$$

and we recall the MMH procedure described in Section 2.2.3. Under the assumption that we can sample from $\pi_\theta\left(dx_{1:T}\,|y_{1:T}\right)$, we use the proposal

$$k_{\text{mh}}\left(x_{1:T}^*, \theta^*\,|x_{1:T}, \theta\right) = \pi_{\theta^*}\left(x_{1:T}^*\,|y_{1:T}\right)k\left(\theta^*\,|\theta\right),$$

```
Data: $y_{1:T}$, $x_0$.
Parameters: $M$,$N$,$T$.
Result: A statistically dependent Markov chain, $\{x_{1:T}(n)\}_{n=0}^{M}$, with limiting
        distribution equal to the target smoothing distribution
        $\pi\left(dx_{1:T}\left|y_{1:T}\right.\right)$.

  1. Set $n = 0$. Using an SMC procedure, generate a particle approximation of
     the target distribution,
         $\hat{\pi}\left(dx_{1:T}\right) = \sum_{i=1}^{N} W_T^{(i)} \delta_{X_{1:T}^{(i)}}\left(dx_{1:T}\right),$
     defining $\hat{\pi}\left(y_{1:T}\right) = \prod_{i=1}^{T} \sum_{i=1}^{N} w\left(X_{1:T}^{(i)}\right)$.
  2. Sample $X_{1:T}(0)$ from $\hat{\pi}\left(dx_{1:T}\right)$
  3. Set $n = n + 1$. Using Algorithm 3, generate a particle approximation of the
     target distribution,
         $\hat{\pi}\left(dx_{1:T}\right) = \sum_{i=1}^{N} W_T^{(i)} \delta_{X_{1:T}^{(i)}}\left(dx_{1:T}\right),$
     defining $\hat{\pi}^{*}\left(y_{1:T}\right) = \prod_{i=1}^{T} \sum_{i=1}^{N} w\left(X_{1:T}^{(i)}\right)$.
  4. Sample a candidate value $X^{*}$ from $\hat{\pi}\left(dx_{1:T}\right)$
  5. Define $\alpha\left(x, x^{*}\right) = 1 \wedge \frac{\hat{\pi}^{*}\left(y_{1:T}\right)}{\hat{\pi}\left(y_{1:T}\right)}$
  6. Set $\begin{cases} X(n) = X^{*}, \hat{\pi}\left(y_{1:T}\right) = \hat{\pi}^{*}\left(y_{1:T}\right) & \text{with probability } \alpha \\ X(n) = X(n-1) & \text{otherwise.} \end{cases}$
     If $n = M$, stop, else return to Step 3.
```

Algorithm 5: A Particle Independent Metropolis-Hastings Algorithm

and this leads to the following MH acceptance probability:

$$\alpha\left(\left\{x_{1:T}, \theta\right\}, \left\{x_{1:T}^{*}, \theta^{*}\right\}\right) \quad = \quad 1 \wedge \frac{\pi_{\theta^{*}}\left(y_{1:T}\right) \pi\left(\theta^{*}\right)}{\pi_{\theta}\left(y_{1:T}\right) \pi\left(\theta\right)} \frac{k\left(\theta\left|\theta^{*}\right.\right)}{k\left(\theta^{*}\left|\theta\right.\right)}, \tag{2.24}$$

where $\pi\left(\theta\right)$ is the marginal target density for θ.

Andrieu et al. (2010) propose the use of an SMC procedure, which targets the joint smoothing distribution $\pi_\theta(x_{1:T}|y_{1:T})$ by sampling from $\pi_\theta(x_{1:T}, y_{1:T})$, and estimating the normalizing density via the unbiased estimate $\hat{\pi}_\theta(y_{1:T})$. The fact that this estimate is unbiased means that the SMC procedure will provide a sample from the true target density $\pi(x_{1:T}|y_{1:T})$, and not an approximation thereof. This validates the use of the SMC procedure to propose candidate states $X^*_{1:T}$ within the MMH procedure.

Framing the sampling problem as before, we have that the density of all variables generated by the SMC procedure is denoted $\psi_\theta(\bar{x}_{1:T}, \bar{a}_{1:T-1})$. The PMMH procedure therefore targets, at each iteration, the density that is formed from the product of:

- the joint density $\psi_\theta(\bar{x}_{1:T}, \bar{a}_{1:T-1})$, conditioned upon the union of the chosen particle index j, the values of the variables in $\left\{\bar{X}_{1:T}, \bar{a}_{1:T-1}\right\}$ that contribute to the realised particle $X^{(j)}_{1:T}$, and the value of θ,

- the density targeted by the SMC procedure, evaluated at the generated state values in $\bar{X}_{1:T}$ that contribute to $X^{(j)}_{1:T}$, i.e. $\pi_\theta\left(X^{(j)}_{1:T}, y_{1:T}\right)/\hat{\pi}_\theta(y_{1:T})$,

- the marginal target density for θ, i.e. $\pi(\theta)$,

- the probability of generating the indices in $\bar{a}_{1:T-1}$ that contribute to $X^{(j)}$, i.e. $N^{-(T-1)}$,

- and the probability of choosing the final particle index j, i.e. N^{-1}.

In practice, the MH acceptance probability in (2.24) is calculated at each iteration, using the unbiased estimates of the marginal likelihood, $\hat{\pi}_\theta(y_{1:T})$.

We finally remark that, as with the PIMH procedure, the estimates of the marginal likelihoods can be calculated recursively during the SMC update scheme, and thus the filtering weights need not be stored. The full procedure for PMMH is given in Algorithm 6.

Data: $y_{1:T}$, x_0.
Parameters: M,N,T.
Result: Two statistically dependent Markov chains, $\{x_{1:T}(n)\}_{n=0}^{M}$, with limiting distribution equal to the target smoothing distribution $\pi_\theta\left(dx_{1:T}\left|y_{1:T}\right.\right)$, and $\{\theta(n)\}_{n=0}^{M}$.

1. Set $n = 0$. Generate $\theta(0)$ according to some prior $p(\theta)$ and, using an SMC procedure, generate a particle approximation of the target distribution,
$$\hat{\pi}_{\theta(0)}\left(dx_{1:T}\right) = \sum_{i=1}^{N} W_T^{(i)} \delta_{X_{1:T}^{(i)}}\left(dx_{1:T}\right),$$
defining $\hat{\pi}_{\theta(0)}\left(y_{1:T}\right) = \prod_{i=1}^{T}\sum_{i=1}^{N} w\left(x_{1:T}^{(i)}\right)$.
2. Sample $x_{1:T}(0)$ from $\hat{\pi}_{\theta(0)}\left(dx_{1:T}\right)$
3. Set $n = n + 1$. Set $\theta = \theta(n-1)$ and generate a candidate value θ^* according to a proposal kernel $K\left(\theta, d\theta^*\right)$
4. Using Algorithm 3, generate a particle approximation of the target distribution,
$$\hat{\pi}_{\theta^*}\left(dx_{1:T}\right) = \sum_{i=1}^{N} W_T^{(i)} \delta_{X_{1:T}^{(i)}}\left(dx_{1:T}\right),$$
defining $\hat{\pi}_{\theta^*}\left(y_{1:T}\right) = \prod_{i=1}^{T}\sum_{i=1}^{N} w\left(x_{1:T}^{(i)}\right)$.
5. Sample a candidate value $x_{1:T}^*$ from $\hat{\pi}_{\theta^*}\left(dx_{1:T}\right)$
6. Calculate $\alpha\left(x, x^*\right)$ according to (2.24), estimating the marginal likelihoods under θ and θ^* by $\hat{\pi}_\theta\left(y_{1:T}\right)$ and $\hat{\pi}_{\theta^*}\left(y_{1:T}\right)$ respectively.
7. Set
$$\begin{cases} \theta(n) = \theta^*, & x(n) = x^*, & \hat{\pi}_{\theta(n)}\left(y_{1:t}\right) = \hat{\pi}_{\theta^*}\left(y_{1:T}\right) & \text{w.p. } \alpha \\ \theta(n) = \theta(n-1), & x(n) = x(n-1), & \hat{\pi}_{\theta(n)}\left(y_{1:t}\right) = \hat{\pi}_{\theta(n-1)}\left(y_{1:T}\right) & \text{otherwise} \end{cases}$$
If $n = M$ stop, else return to Step 3

Algorithm 6: A Particle Marginal Metropolis-Hastings Algorithm

2.5 Approximate Bayesian Computation

This section provides a brief introduction to approximate Bayesian computation (ABC) methods. Bayesian inference provides a useful approach to performing inference in a wide range of scenarios. When performing Bayesian inference in a computational setting, however, the explicit evaluation of the likelihood can prove problematic for many real world problems. In many areas of interest, such as the study of population genetics (e.g. Tavaré et al., 1997; Pritchard et al., 1999) or in financial applications (e.g. Jasra et al., 2010), the likelihood can neither be evaluated exactly through a closed form expression, nor replaced by an unbiased estimate. In such scenarios, ABC can provide an alternative method for performing inference that avoids the evaluation of troublesome likelihood functions.

In Chapters 3 and 4, we introduce the application of ABC methods to the dynamic problems of filtering and smoothing, respectively. In particular, we will use ABC to introduce, in Chapter 3, an approximation of the generalised SSM (1.1)-(1.2). We provide here an introduction to ABC methods in the context of their application to static models.

2.5.1 ABC for Static Models

Suppose we have a set of data $y \in \mathbb{R}^{d_y}$, assumed to be drawn from the parametric distribution with likelihood $g\left(y\,|\theta\right)$, and that our aim is to perform simulation-based Bayesian inference on the unknown parameter vector $\theta \in \Theta$, to which we assign a prior $\pi(\theta)$. If the likelihood is intractable, then a standard Bayesian approach to simulating from the posterior distribution of interest is impossible. Several methods have been proposed for circumventing the evaluation of the likelihood, in an effort to complete a Bayesian-type approach to problems of this nature, and these have led to formalisation in the form of approximate Bayesian computation (ABC). A recent review of ABC methods is provided by Marin et al. (2011).

The first proposed methods for circumventing the evaluation of the likelihood were introduced in the genetics literature by Tavaré et al. (1997) and Pritchard et al. (1999). These papers propose rejection-based procedures that are similar to each other, motivated by the lack of a tractable relationship between a set of population parameters and their observed DNA data. In both cases, and in the current notation, the procedures iteratively execute the following steps: candidate parameters are proposed according to the prior $\pi(\theta)$; a set of artificial data, or pseudo-data, $u \in \mathbb{R}^{d_y}$ is simulated according to the likelihood $u \sim g(\cdot|\theta)$; and the candidate parameter is accepted if the pseudo-data u are, in some sense, close to the observed data. We omit precise details of these approaches, noting that they are simply implementations of a rejection-sampling procedure that targets the ABC approximation of the posterior of interest, which we define below.

Under the above setup, we wish to perform simulation-based inference with respect to the posterior

$$\pi(\theta|y) \propto g(y|\theta)\pi(\theta),$$

where $\pi(\theta)$ is a prior on the parameter vector. The ABC approach introduces an auxiliary RV U, referred to as the pseudo-data, which is used to define an approximation of the likelihood that can subsequently be used to approximate the posterior distribution. We can then perform simulation-based inference with respect to this ABC approximation of the posterior, even when the likelihood is intractable.

The ABC approximation of the likelihood is given by

$$g^{\epsilon}(y|\theta) = \int g(u|\theta)G^{\epsilon}(u,y)du, \tag{2.25}$$

where $G^{\epsilon}(u,y)$ is the ABC kernel, which measures the proximity of the pseudo-data u to the observed data y, through the use of some suitably defined distance metric $\rho(s(u), s(y))$, where $s(\cdot)$ is a suitable summary statistic. The ABC kernel returns a value based upon the magnitude of this distance, relative to some pre-assigned ABC tolerance parameter ϵ. The accuracy of the ABC approximation of the likelihood is controlled by the specification of ρ, s, and ϵ, and this is discussed in further detail in Section 2.5.2.

We immediately note that, even if the exact likelihood is intractable, the ABC approximation of the likelihood (2.25) may be estimated by simulating $u \sim g(\cdot|\theta)$ and evaluating the ABC kernel $G^{\epsilon}(u, y)$. Indeed, it is noted here that an alternative estimator for the ABC approximation of the likelihood, with potentially improved accuracy, can be constructed by generating J independent pseudo-data $u_1, \ldots, u_J \sim g(\cdot|\theta)$:

$$\hat{\pi}^{\epsilon,J}(y|\theta) = \frac{1}{J}\sum_{j=1}^{J} G^{\epsilon}(u_j, y). \tag{2.26}$$

Throughout this thesis, we will refer to this as the *multiple pseudo-data* ABC (MPDABC) approach, and we note that it has been used by several authors, including Marjoram et al. (2003) who address the problem of targeting the ABC approximation of the posterior using an MH procedure.

The ABC approximation of the likelihood can be used to define the following ABC approximation of the posterior of interest:

$$\pi^{\epsilon}\left(\theta\left|y\right.\right) = \int \pi^{\epsilon}(\theta, u\left|y\right.)du, \tag{2.27}$$

where the integrand is an auxiliary joint density, defined as

$$\pi^{\epsilon}\left(\theta, u\left|y\right.\right) = \frac{\pi(\theta)g\left(u\left|\theta\right.\right)G^{\epsilon}\left(u, y\right)}{\int \pi(\theta)g\left(u\left|\theta\right.\right)G^{\epsilon}\left(u, y\right)dud\theta}. \tag{2.28}$$

A common choice of ABC kernel, and the one that shall be adopted for the remainder of this thesis, is the indicator function

$$G^{\epsilon}\left(u, y\right) = \mathbb{I}_{A_{\epsilon,y}}\left(u\right), \tag{2.29}$$

where the set $A_{\epsilon,y}$ is defined as

$$A_{\epsilon,y} = \left\{u\ :\ \rho\left(s\left(u\right), s\left(y\right)\right) < \epsilon\right\}.$$

2.5.2 Controlling the accuracy of the ABC approximation

As highlighted by Blum et al. (2012), the ABC approximations of the likelihood (2.25), and of the posterior (2.27) involve two component approximations: the approximation of the data and pseudo-data through the use of a summary statistic; and the subsequent approximation of the likelihood through the ABC kernel, given this choice of summary statistic. The choice of summary statistic s clearly determines the accuracy of the first approximation, whilst ρ and ϵ determine the accuracy of the second.

The problem of finding a distance metric that offers the greatest accuracy in the ABC approximation is often model-specific, and so we do not remark further on the effect of different choices of ρ. We simply note that common choices are the Euclidean $\mathbb{L}_p$-distances, i.e.

$$\rho(s(u), s(y)) = \left(\sum_{i=1}^{d_s} |s_i(u) - s_i(y)|^p \right)^{1/p},$$

for $p \geq 1$, and where $s_i(\cdot)$ is the i^{th} dimension of the d_s-dimensional summary statistic.

In contrast, the options available for the specification of s and ϵ provide the analyst with some control over the accuracy of the ABC approximation in general. However, improvements in accuracy often come at the cost of an increase in the computational expense of procedures that target the ABC approximation.

We first describe the choices of ϵ and s that improve the accuracy of the approximation. The chapter then concludes with some remarks on the implications of these preferred choices on the computational efficiency of performing inference with respect to the ABC approximation of a model.

Favourable Specifications

With regards to the approximation of the likelihood through the ABC kernel, the effect of the ABC tolerance is intuitive: smaller values of ϵ will result in a 'tighter' kernel G^{ϵ}, and a more accurate approximation of the likelihood function.

Central to the suitability of a chosen summary statistic is its sufficiency; a sufficient statistic will provide maximum information about the observed data and the pseudo-data to the kernel approximation of the likelihood, leading to a more accurate measurement of the fit of the pseudo-data to the observed data, and a more accurate ABC approximation of the likelihood.

Computational Constraints

As mentioned above, the accuracy of the ABC approximation will improve as ϵ shrinks. As ϵ decreases, however, so does the chance of the generated pseudo-data lying within the required $\mathbb{L}_p$-distance ϵ of the observed data. This leads to an increase in the computational budget required for a given level of accuracy, and a trade-off between accuracy and computational expense is introduced for the choice of ϵ.

It was noted above that the idealised specification for the ABC approximation of the likelihood would involve a sufficient summary statistic. In practice however, as noted by Marin et al. (2011), non-trivial sufficient statistics are rare, and an increase in the amount of information provided by summary statistics typically translates to an increase in the dimension d_s. With increased dimension, however, comes an increase in the computational expense incurred by the comparison of these statistics. Furthermore, in some situations, one might expect the $\mathbb{L}_p$-distance between the pseudo-data and observed data to increase with the dimension. This would, as above, decrease the chance of the pseudo-data being located within the required $\mathbb{L}_p$-distance ϵ of the observed data. Thus, assuming one is performing simulation based inference on the unknown parameter θ, an increase in d_s could result in an increase in the number of candidate parameters required, and thus a further increase in the computational budget required, in order to perform inference to a given level of accuracy.

For the choice of summary statistic, then, there is also a tradeoff between the level of information provided and the computational expense incurred. There is an expanding literature on the subject of dimension reduction methods for ABC summary statistics, whose aim is to minimize the dimension of the statistics whilst maximizing the information that they relay. Further details can be found in the recent review by Blum et al. (2012) and the references within.

Chapter 3

Filtering via Approximate Bayesian Computation

3.1 Introduction

Particle filtering for the exact SSM (1.1)-(1.2) is impossible in the case where the observation density $g_t(y_t | x_t)$ is intractable. Particle filtering also suffers from difficulties in high dimensions, even when the likelihood may be computed exactly, as the number of particles required to avoid significant weight degeneracy can grow exponentially in the dimension of the hidden state (Bengtsson et al., 2008; Bickel et al., 2008; Snyder et al., 2008). This has the effect of significantly increasing the computational budget required to accurately perform particle filtering.

In this chapter, we consider the ABC approximation of the SSM and we show that this approximation may be targeted successfully using particle filtering methods; see also the joint work, Jasra et al. (2010). We will see that particle filtering for an ABC target (hereafter referred to as ABC particle filtering) addresses the problems of both likelihood intractability and high-dimensional states, at the cost of introducing a deterministic bias in the resulting filtering estimates.

In Section 3.2, we formally introduce the ABC approximation of the generalised SSM. In Section 3.3, we present a theoretical exploration of the deterministic bias that is induced in the filtered expectation of a bounded test function, where the expectation is taken with respect to the ABC approximation of the filtering density. This theoretical analysis motivates the implementation of ABC particle filtering and details of the ABC particle filtering procedure are provided in Section 3.4. The performance of ABC particle filtering is then assessed for a linear Gaussian SSM in Section 3.5 and for a common nonlinear SSM from the SMC literature in Section 3.6.

3.2 ABC for State Space Models

Recall that the SSM is defined through the specification of the transition density $q_t(x_t|x_{t-1})$ and the observation density $g_t(y_t|x_t)$, as described in Section 1.2. In applying ABC to the SSM, a similar approach to that used in Section 2.5 is followed. At each time step t, we assume that we have access to a d_y-dimensional pseudo-datum, $u_t \in \mathbb{R}^{d_y}$. For the purposes of this thesis, it will suffice to assume that $u_t \sim g_t(\cdot|x_t)$, however it should be noted that this assumption may be relaxed in certain scenarios; this is discussed further in Chapter 6. Furthermore, we assume throughout this chapter that any parameter vector θ that may be associated with the SSM is known, and thus suppressed from the notation.

The likelihood at time t is approximated by the following marginalisation with respect to the pseudo-datum u_t:

$$g_t^\epsilon(y_t|x_t) = \int g_t^\epsilon(y_t, u_t|x_t)du_t = \int G_t^\epsilon(u_t, y_t)g_t(u_t|x_t)du_t, \tag{3.1}$$

where, here and for the remainder of this thesis, $G_t^\epsilon(u_t, y_t)$ is the ABC kernel at time t, whose value is determined by the distance metric $\rho(s(u_t), s(y_t))$, with summary statistic s, and the ABC tolerance parameter ϵ.

Using this approximation of the likelihood, it is possible to construct ABC approximations of both the joint smoothing density and the filtering density. Recall the form of the joint smoothing density (1.7), obtained from the exact SSM; the ABC approximation of the joint smoothing density, also referred to as the ABC smoothing density or ABC smoother, is defined here as

$$\pi^{\epsilon}\left(x_{1:t} \mid y_{1:t}\right) = \int \pi^{\epsilon}\left(x_{1:t}, u_{1:t} \mid y_{1:t}\right) du_{1:t} \tag{3.2}$$

with the auxiliary joint distribution

$$\pi^{\epsilon}\left(x_{1:t}, u_{1:t} \mid y_{1:t}\right) = \frac{\prod_{n=1}^{t} G_n^{\epsilon}\left(u_n, y_n\right) g_n\left(u_n \left| x_n\right.\right) q_n\left(x_n \left| x_{n-1}\right.\right)}{\int_{\mathbb{R}^{t(d_x+d_y)}} \prod_{n=1}^{t} G_n^{\epsilon}\left(u_n, y_n\right) g_n\left(u_n \left| x_n\right.\right) q_n\left(x_n \left| x_{n-1}\right.\right) dx_{1:t} du_{1:t}}. \tag{3.3}$$

From this approximation of the joint smoothing density, the ABC approximation of the filtering density at time t, also referred to as the ABC filtering density or ABC filter, is immediately available as the marginalisation of (3.2) with respect to the state sequence up to time $t-1$:

$$\pi^{\epsilon}\left(x_t \mid y_{1:t}\right) = \int \pi^{\epsilon}\left(x_{1:t} \mid y_{1:t}\right) dx_{1:t-1}. \tag{3.4}$$

It is remarked that the above ABC approximations of the filtering and joint smoothing densities can be considered to specify an SSM of their own, in which the dynamics of the hidden state are unchanged, and the RV of the observed data at time t, Y_t, retains the same dependence structure with respect to $Y_{1:t-1}$ and $X_{1:t}$ (Dean et al., 2010). We refer to the SSM specified by the ABC filtering and joint smoothing densities as the ABC approximation of the SSM, and the original SSM of interest may be referred to as the exact SSM. The fact that the ABC approximation of the SSM maintains the probabilistic structure of the exact SSM suggests that inference with respect to the ABC filtering and ABC joint smoothing densities may be achieved through the computational procedures designed for inference with respect to the corresponding exact densities. Furthermore, the unchanged probabilistic structure also facilitates the application of theoretical results pertaining to these existing computational methods. In Section 3.3, we present an analysis of the theoretical behaviour of the ABC approximation; we first detail the ABC setup that will be used for the majority of this thesis.

We consider the ABC approximations above, with the ABC kernel $G_t^\epsilon(u_t, y_t)$ specified as the indicator function on the d_y-dimensional ball of radius ϵ, centred at y_t, i.e.

$$G_t^\epsilon\left(u_t, y_t\right) = \mathbb{I}_{A_{\epsilon, y_t}}\left(u_t\right). \tag{3.5}$$

with $A_{\epsilon,y_t} = \{u : |y_t - u| < \epsilon\}$. Note that this specification also implicitly specifies the use of the identity statistic $s(u_t) = u_t$ and the $\mathbb{L}_1$-distance for the metric $\rho(s(u_t), s(y_t)) = \sum_{j=1}^{d_y} |u_{t,j} - y_{t,j}|$. It is under this specification that the theoretical analysis of the behaviour of the ABC approximation is carried out in Sections 3.3 and 4.4.

3.3 Exploring the Theoretical Bias of ABC Filtering

In this section, we present a brief theoretical analysis of the bias induced by the ABC approximation of the joint smoothing density. We combine this analysis with existing results concerning the behaviour of the SMC estimate of the exact joint smoothing distribution in order to analyse the behaviour of the SMC estimate of the ABC filtered expectation

$$\mathbb{E}_{\pi_t^\epsilon}\left[\varphi(X)\right] = \int \varphi(x_t)\pi^\epsilon(x_t|y_{1:t})dx_t,$$

for some bounded measurable function $\varphi : \mathbb{R}^{d_x} \to \mathbb{R}^{d_x}$.

The main theoretical results provided in this section establish that the bias of the ABC filtered expectation of a bounded test function is bounded above, and that this bias diminishes to 0^+ as the ABC tolerance parameter $\epsilon \searrow 0$. The results also address the behaviour of the upper bound of the bias with respect to time, and are presented in Propositions 3.3.1 and 3.3.2. These results, which consider the behaviour of the bias under moderate and restrictive assumptions, respectively, were proven by Dr. Jasra and Dr. S.S. Singh; the proofs can be found in our aforementioned paper (Jasra et al., 2010).

Define the bias at time t as

$$\mathbb{B}_f(t,\epsilon) := \left| \int \varphi\left(x_t\right)\left[\pi(x_t\,|y_{1:t}) - \pi^{\epsilon}(x_t\,|y_{1:t})\right] dx_t \right|. \tag{3.6}$$

Before stating the result, we detail the adopted assumptions.

(**A1**) *Boundedness of the Likelihood.*

For any $t \geq 1$, there exists a $\bar{g}_t$ such that for any $x \in \mathbb{R}^{d_x}$ we have

$$\left\|g_t\left(\cdot\mid x\right)\right\|_\infty := \sup_{u\in\mathbb{R}^{d_y}} \left|g_t\left(u|\, x\right)\right| \leq \bar{g}_t < +\infty$$

(**A2**) *Lipschitz Continuity of the Likelihood.*

For any $t \geq 1$, there exists a constant, $L_t \in (0,\infty)$ such that for any $x \in \mathbb{R}^{d_x}$, $y, \widetilde{y} \in \mathbb{R}^{d_y}$,

$$|g_t(y|\, x) - g_t(\widetilde{y}|\, x)| \leq L_t|y - \widetilde{y}|$$

where $|\cdot|$ is understood to be an $\mathbb{L}_1-$norm.

(**A3**) *Statistic and Metric.*

The set $A_{\epsilon,y}$ is:

$$A_{\epsilon,y} = \{u : |y-u| < \epsilon\}.$$

(**A4**) *Strong Mixing Conditions*

There exist probability densities κ_1 on $\mathbb{R}^{d_x}$ and κ_2 on $\mathbb{R}^{d_y}$, as well as constants

$0 < \lambda_1, \lambda_2 < \infty$ such that for all $(x, x') \in \mathbb{R}^{2d_x}$, $y \in \mathbb{R}^{d_y}$,

$$\begin{aligned} \frac{1}{\lambda_1}\ \kappa_1(x') \ &\leq\ q_t(x'|x) \ \leq\ \lambda_1\ \kappa_1(x'), \\ \frac{1}{\lambda_2}\ \kappa_2(y) \ &\leq\ g_t(y|x) \ \leq\ \lambda_2\ \kappa_2(y). \end{aligned}$$

for all $t \geq 1$, with

$$0 < \underline{\kappa}_2 < \kappa_2(y) < \overline{\kappa}_2 \quad \forall y \in \mathbb{R}^{d_y}.$$

In general, (A1-A3) are satisfied when the latent and observed processes evolve on compact subsets; it is these assumptions that are adopted for Proposition 3.3.1, which shows the validility of using the ABC approximation of the filter. (A4) is a particularly restrictive assumption, and will not be satisfied by many HMMs; it is under this further assumption, along with (A1-A3), that Proposition 3.3.2 is presented. This latter result indicates the potential for the ABC filter to have a controllable bias.

Proposition 3.3.1. *Assume (A1-A3). Then for any* $t \geq 2$, $y_{1:t}$, *there exists a* $C_t\left(y_{1:t}\right)$ *such that for any* $\varphi \in \mathcal{B}_b\left(\mathbb{R}^{d_x}\right)$, $\epsilon > 0$,

$$\mathbb{B}_f(t, \epsilon) \leq \epsilon C_t\left(y_{1:t}\right) \|\varphi\|_\infty \tag{3.7}$$

where

$$\begin{aligned} C_t\left(y_{1:t}\right) =& \frac{1}{\pi\left(y_t\left|y_{1:t-1}\right.\right)}\left(2L_t + \bar{g}_t C_{t-1}\left(y_{1:t-1}\right)\right), \quad t \geq 2, \\ C_1\left(y_1\right) =& \frac{2L_1}{\pi\left(y_1\right)}. \end{aligned}$$

The dependence on ϵ of the upper bound on the ABC filtering bias shows that as ϵ tends to zero, an expectation with respect to the ABC approximation of the filtering density tends to the expectation with respect to the true filtering density, i.e. the approximation disappears as $\epsilon \searrow 0$. However, since C_t will typically grow exponentially with t, the ABC bias should be controlled through the use of a decreasing sequence of tolerances $\{\epsilon_t\}_{t\geq 1}$. Furthermore, the appearance of the marginal likelihood of the data on the denominator of $C_t\left(y_{1:t}\right)$ means that the error could explode at any particular point for which the observed data are particularly unlikely.

We now adopt (A4) and state Proposition 3.3.2.

Proposition 3.3.2. *Assume (A1-A4). Assume also that $L_t \leq L < \infty$ for $t \geq 1$ and L_t as in (A2) Then for any $t \geq 1$, $y_{1:t}$, there exists a $C < \infty$ such that for any $\varphi \in \mathcal{B}_b\left(\mathbb{R}^{d_x}\right)$, $\epsilon > 0$,*

$$\mathbb{B}_f(t, \epsilon) \leq \epsilon C \left\|\varphi\right\|_{\infty}. \tag{3.8}$$

This result shows that, under the restrictive assumption (A4), the ABC bias does not accumulate over time.

For SMC methods, it is common to examine the accuracy of the resulting estimators in terms of their $\mathbb{L}_p$-error. For the SMC estimate of the filtered expectation, this is defined as

$$\mathbb{E}\left[\left|\int \varphi(x_t)\left[\pi(x_t|y_{1:t}) - \hat{\pi}(x_t|y_{1:t})\right] dx_t\right|^p\right]^{1/p},$$

for $p \in \mathbb{N}$, where the expectation is taken with respect to the joint distribution of all variables generated by the SMC procedure. Del Moral (2004) considers the behaviour of this $\mathbb{L}_p$-error under certain regularity conditions, and shows that, for bounded test functions φ, this error is bounded above by the term

$$\frac{B_p}{\sqrt{N}} \left\|\varphi\right\|_{\infty},$$

where B_p is dependent upon p only.

When considering the SMC approximation of the ABC filter, a similar result will hold; in this case, we acknowledge the possible dependence of B_p on ϵ, i.e.

$$\mathbb{E}\left[\left|\int \varphi(x_t)\left[\pi^{\epsilon}(x_t|y_{1:t}) - \hat{\pi}^{\epsilon}(x_t|y_{1:t})\right] dx_t\right|^p\right]^{1/p} \leq \frac{B_p(\epsilon)}{\sqrt{N}} \|\varphi\|_{\infty},$$

We can combine this result with the deterministic ABC bias in (3.7) to give a bound for the $\mathbb{L}_p$-error between the SMC estimate of the ABC filtered expectation $\mathbb{E}_{\pi_t^{\epsilon}}[\varphi(X)]$ and the exact filtered expectation:

$$\mathbb{E}\left[\left|\int \varphi(x_t)\left[\pi(x_t|y_{1:t}) - \hat{\pi}^{\epsilon}(x_t|y_{1:t})\right] dx_t\right|^p\right]^{1/p} \leq \left(\frac{B_p(\epsilon)}{\sqrt{N}} + \epsilon C_t\left(y_{1:t}\right)\right) \|\varphi\|_{\infty}, \quad (3.9)$$

which holds under the assumptions (A1-A3), and under the regularity assumptions in Del Moral (2004). These results indicate that use of a particle filtering procedure that targets the ABC approximation of the filtered expectation of the hidden state may perform adequately, with a recognisable bias. In the next section, we provide details of such a procedure, providing in particular a sensible method for determining the sequence of ABC tolerances $\{\epsilon_t\}_{t\geq 0}$.

3.4 Incorporating ABC into SMC Filtering Procedures

A method is proposed here for performing ABC particle filtering, where the approximation is specified by the indicator function ABC kernel, as given in (3.5).

The particle filter for targeting this ABC approximation of the SSM is a straightforward alteration of the particle filter detailed in Algorithm 3: at iteration t, for each particle $i = 1, \ldots, N$, one samples a single pseudo-datum $u_t^{(i)}$ prior to calculating the importance weight, and then allocates to that particle the incremental weight given by the evaluated indicator function $\mathbb{I}_{A_{\epsilon,y_t}}\left(u_t^{(i)}\right)$. The basic procedure for performing ABC particle filtering is detailed in Algorithm 7.

A generalisation of the above ABC kernel will also be considered. Recall the MPDABC kernel (2.26), defined previously in the context of static models. We

Data: $y_{1:T}$.
Parameters: x_0, N, ϵ.
Result: The SMC estimates of $\mathbb{E}_{\pi_t^\epsilon}\left[f(X_t)\right]$, for $t = 1, \ldots, T$.

1. Set $t = 1$. For $i = 1, \ldots, N$, sample $X_t^{(i)} \big| x_0$ and $U_t^{(i)} \big| X_t^{(i)}$, according to the proposal and observation densities, respectively, and calculate the filtering weights
$$W_t^{(i)} \propto \frac{G_t^\epsilon\left(U_t^{(i)}, y_t\right) q_t\left(X_t^{(i)} | x_0\right)}{k_t\left(X_t^{(i)} | x_0\right)}, \quad \sum_{i=1}^{N} W_t^{(i)} = 1.$$

2. If $\widehat{ESS}\left(\left\{W_t^{(i)}\right\}_{i=1}^N\right) < \frac{N}{2}$, then, for $i = 1, \ldots, N$, resample $X_t^{(i)}$ independently from the discrete distribution
$$\hat{\pi}\left(dx_t\right) = \sum_{i=1}^{N} W_t^{(i)} \delta_{X_{1:t}^{(i)}}\left(dx_t\right)$$
and set $W_t^{(i)} = \frac{1}{N}$.

3. Set $t = t + 1$. For $i = 1, \ldots, N$, sample $X_t^{(i)} \big| X_{t-1}^{(i)}$ and $U_t^{(i)} \big| X_t^{(i)}$ according to the proposal and observation densities, respectively, and calculate the filtering weights
$$W_t^{(i)} \propto W_{t-1}^{(i)} \cdot \frac{G_t^\epsilon\left(U_t^{(i)}, y_t\right) q_t\left(X_t^{(i)} \middle| X_{t-1}^{(i)}\right)}{k_t\left(X_t^{(i)} \middle| X_{t-1}^{(i)}\right)}, \quad \sum_{i=1}^{N} W_t^{(i)} = 1.$$

4. Calculate the particle filtering estimate $\hat{f}_{\pi_t}^N(X_t)$ according to (2.14).
If $t = T$, stop; otherwise, return to Step 2.

Algorithm 7: An SMC Algorithm for ABC Filtering

update the notation and re-define the MPDABC kernel as

$$G_t^\epsilon\left(u_t, y_t\right) = \frac{1}{J} \sum_{j=1}^{J} \mathbb{I}_{A_{\epsilon, y_t}}\left(u_{t,j}\right), \tag{3.10}$$

for $J \geq 1$ where $u_t = \{u_{t,1}, \ldots, u_{t,J}\}$. This choice of ABC kernel is motivated by the fact that, as $J \to \infty$, one recovers the integrated likelihood over the d_y-dimensional ball A_{ϵ,y_t}:

$$\frac{1}{J}\sum_{j=1}^{J}\mathbb{I}_{A_{\epsilon,y_t}}(u_{t,j}) \to \int_{A_{\epsilon,y_t}} g_t(u_t|x_t)du_t.$$

We refer to ABC particle filtering procedures that employ the ABC kernel (3.10) as MPDABC particle filters and henceforth use 'ABC particle filters' to describe those procedures that use the single indicator function (3.5).

MPDABC particle filtering is implemented in a similar fashion to ABC particle filtering. At each time step, one simulates J pseudo-data $\{u_{t,1}, \ldots, u_{t,J}\}$ prior to calculation of the importance weights, then replaces the likelihood in the incremental weight with the evaluated kernel $G_t^{\epsilon}(u_t, y_t)$ (3.10). Such a filtering procedure mimics a 'marginal' particle filtering procedure that targets the ABC approximation of the filtering distribution with the pseudo-data integrated out (3.4).

It is expected that the use of the MPDABC kernel will result in filtering weights with reduced variance, due to a Rao-Blackwellisation type argument (e.g. Casella and Robert, 1996), reducing the SMC error of the resulting estimates. Further, it is expected that the particles will be less likely to die out when performing MPDABC filtering, as there will be a smaller probability of each particle being assigned a zero-valued incremental weight. The estimates resulting from the MPDABC filtering procedure will, however, still be subject to the ABC bias described in Section 3.3. Furthermore, the computational expense of the MPDABC particle filter will be greater than that of the ABC particle filter; it will be of interest to observe, in practice, whether any significant gains in accuracy can be observed whilst maintaining a reasonable computational budget.

3.4.1 Weight Degeneracy for ABC Filtering

As a result of using the ABC kernels in (3.5) and (3.10), the incremental weights at any time step of the ABC particle filter in Algorithm 7 may be evaluated at 0. Where the particles are allocated a zero-valued weight, the particle may be said to have ‘died out’; similarly, one can talk about particles with positive weight being ‘alive’ in this context. SMC algorithms in which the particles can die out may be subjected to a specialized theoretical analysis, however this is not pursued in this thesis; for further details, see Del Moral (2004).

As described in Section 2.3.1, the exact particle filter suffers, to an extent, from the weight degeneracy phenomenon that occurs, in part, due to the recursive nature of the importance weights and the compounding of small and large weights for particles in areas of low and high probability density, respectively.

Consider the ABC particle filter that uses the indicator ABC kernel (3.5). At any particular time step t, even without resampling at $t-1$, all of the particles that are alive will have equal weighting, as all of their incremental weights since the last resampling move will have been 1. Thus, the weights will not experience the same compounding effect that is evident in the exact particle filter, and will therefore not be as prone to this type of weight degeneracy issue.

The ABC filter will be prone to an alternative type of weight degeneracy. Since particles can die out based on their incremental weights alone, it is possible for all particles to die out at a particular time step, even if they have been resampled at the previous time step. This is referred to here as a collapse of the algorithm and its occurrence is dependent on the dynamics of the state and observation processes, at the particular time at which the collapse occurs, as opposed to an inherent problem with the propagation of the weights.

In practice, there are a number of ways of protecting against algorithm collapse. By simply increasing the particle system size, one can reduce the chance of all particles dying out at a particular time step. Similarly, one can reduce the chance of obtaining zero-valued incremental weights by increasing the ABC tolerance parameter ϵ. It is also noted that the use of the MPDABC kernel (3.10) will lower the chance of obtaining zero-valued incremental weights, therefore protecting against algorithm collapse to an extent. Furthermore, we note that the use of more general kernels of the form

$$G_t^{\epsilon}(u,y) = \frac{1}{\epsilon}\phi\left(\frac{\rho(s(u_t),s(y_t))}{\epsilon}\right),$$

where $\phi(\cdot)$ is some smooth density kernel, will offer a more robust protection to this type of degeneracy in the ABC particle filter.

We maintain our ABC setup as previously described, i.e. using the ABC and MPDABC kernels specified in (3.5) and (3.10), respectively, and we look to reduce the chance of algorithm collapse through controlling ϵ.

3.4.2 Adaptively Tuned ABC for SSMs

The number of particles that are kept alive can be increased by increasing ϵ, however this has the unwanted effect of increasing the bias of the ABC approximation; we therefore wish to keep ϵ as low as possible, whilst keeping the chance of collapse minimised.

In the implementation of our particle filter, we use a simple adaptive method for generating a new ABC tolerance at each iteration, choosing ϵ_t at time t that encourages the pseudo-data to be assigned an evaluated ABC kernel of 1 for a reasonable proportion of the particles. At time $t = 1$ we generate, from the prior and the likelihood respectively, dummy samples $\left\{x_1^{(i)}, u_1^{(i)}\right\}_{i=1}^{l}$ for arbitrary, yet sufficiently large l and we set ϵ_1 to be the largest distance between y_1 and the samples $\left\{u_1^{(i)}\right\}_{i=1}^{l}$. Then, for a chosen $\alpha \in [1,\ldots,N]$, at time $t \geq 2$ we set ϵ_t as the α-largest distance between y_{t-1} and the samples $\left\{u_{t-1}^{(i)}\right\}_{i=1}^{l}$. The number of alive particles at each time step is now controlled by the parameter α.

3.4.3 Expected Benefits of ABC Particle Filtering

Exact particle filtering cannot be performed when the likelihood is intractable. Even when the likelihood may be computed exactly, particle filtering can prove difficult for SSMs defined on high-dimensional spaces, as the number of particles required to avoid significant weight degeneracy can grow exponentially in the dimension of the hidden state (Bengtsson et al., 2008; Bickel et al., 2008; Snyder et al., 2008). This has the effect of significantly increasing the computational budget required to accurately perform particle filtering.

ABC particle filtering does not require point-wise evaluation of the likelihood, and so immediately avoids the problem of likelihood intractability. A recent theoretical result by Beskos et al. (2012) (Proposition 5.2. of that paper) has shown that, when performing ABC particle filtering under the ABC specification considered here, the error due to the SMC approximation of the ABC filter can be controlled at a computational cost of $O(Nd)$, where the number of particles required will increase sub-exponentially with the dimension. It should be noted that Beskos et al. (2012) consider particle filtering without resampling, i.e. sequential importance sampling as defined in Section 2.3.1. The accuracy of the particle filtering estimate, calculated when using dynamic resampling, can be no worse than the results given in that paper, as resampling only serves to improve the variance of the importance weights at the subsequent time step, thus improving the resulting filtering estimate. This would suggest that, for a given number of particles, the SMC error for the ABC filter introduced here may be lower than for the exact filter in high dimensions. This can be attributed to the reduced susceptibility of the ABC particle filter to the SMC weight degeneracy issue, as discussed in Section 3.4.1. This improvement in SMC error, however, will be counteracted by the deterministic bias induced in the ABC approximation of the filtering density.

As mentioned in Section 1.3, alternative filtering methods have been proposed that also address the problem of likelihood intractability; one such approach is the convolution particle filter (Campillo and Rossi, 2009). The convolution particle filter applies the filtering recursions (1.9)-(1.10), using simulation-based kernel estimation techniques to estimate both the transition density $q_t(x_t|x_{t-1})$ in the predictive filter (1.9) and the likelihood $g_t(y_t|x_t)$ in the update equation (1.10) at each time t. Given samples from the filter at $t-1$, π_{t-1}, one can then obtain samples from the resulting mixture-of-kernels estimate of π_t. We remark first that this approach requires the user to simulate twice from the transition density, which could be unnecessary and wasteful. We also remark that, in independent simulation examples performed by Dr. Jasra, the ABC particle filter that is presented here was observed to significantly outperform the convolution particle filter. For these reasons, we do not consider the convolution particle filter further.

A number of alternative methods have recently been proposed to counter the difficulties faced in high-dimensional filtering. These include the use of an SMC sampler between particle filtering time steps (Beskos et al., 2012), and the use of a combination of the ensemble Kalman filter (EnKF) and the particle filter (Lei and Bickel, 2009).

The approach of Beskos et al. (2012) introduces a sequence of d_x+1 intermediary target distributions $\left\{\pi^{(n)}(x_{1:t}|y_{1:t})\right\}_{n=0}^{d_x}$, between the joint smoothing distributions $\pi_{1:t-1}$ and $\pi_{1:t}$, which are targeted at times $t-1$ and t of the particle filter. The first intermediary target density is

$$\pi^{(0)}(x_{1:t}|y_{1:t}) = q_t(x_t|x_{t-1})\pi(x_{1:t-1}|y_{1:t-1}),$$

i.e. one extends the joint smoothing distribution $\pi_{1:t-1}$ to the state space on which $\pi_{1:t}$ is defined, using only the transition density. Each successive intermediary target density is then defined as the product of the previous intermediary target and a tempered likelihood term:

$$\pi^{(n)}(x_{1:t}|y_{1:t}) = [g_t(y_t|x_t)]^{\phi\left(\frac{n}{d_x}\right)} \pi^{(0)}(x_{1:t}|y_{1:t}) \quad 1 \leq n \leq d_x,$$

for some annealing function $\phi(\cdot)$ that satisfies $\phi(1) = 1$. Note that this condition on $\phi(\cdot)$ results in the final intermediary target being equal to the required joint smoothing density at t. This tempered annealing approach reduces the variability of the incremental weights in the particle filter, and results in a more accurate SMC approximation of the joint smoothing distribution at each time step, and thus a more accurate filtered estimate.

The approach of Lei and Bickel (2009) takes advantage of the EnKF's ability to perform filtering on smaller subsets of the state whilst retaining information about the ensemble, and the particle filter's ability to adapt to nonlinear, non-Gaussian SSMs. It is noted that, whilst not investigated in this thesis, filtering via ABC has the potential to offer a complementary technique to the EnKF (see, e.g. Nott et al., 2012), as well as a competing one.

We now move to two examples, in order to evaluate the accuracy and algorithmic performance of the particle filter and its variants when targeting the ABC approximations of two common SSMs from the SMC literature.

3.5 Toy Example: A Linear Gaussian SSM

We consider the linear Gaussian SSM, with $d_x = d_y = d$:

$$\begin{aligned} X_t &= AX_{t-1} + V_t, \quad t \geq 1, \qquad (3.11) \\ Y_t &= BX_t + Z_t, \quad t \geq 1, \qquad (3.12) \end{aligned}$$

with

$$\begin{aligned} V_t, Z_t &\sim \mathcal{N}\left(0, \sigma^2 I_d\right) \quad (i.i.d.) \quad \text{and} \quad V_t \perp\!\!\!\perp Z_t \quad t \geq 1 \\ A = B &= I_d, \quad \text{the } d\text{-dimensional identity matrix,} \\ X_0 &= 0_d, \quad \text{a } d\text{-length vector with all elements equal to 0.} \end{aligned}$$

With this example, the aim will be to demonstrate the accuracy of particle filtering when applied to the both the linear Gaussian SSM (3.11)-(3.12) and its ABC approximation, and we will be interested in the estimation of the filtered state at the first $T = 600$ time steps. Denote briefly the estimate of the filtered state at time t by $\hat{x}_t$. Since this is a linear Gaussian SSM, the true filtered state is available via the Kalman filter (Kalman, 1960). The accuracy of the SMC procedures will therefore be measured in terms of the accuracy of the first moment estimates that they generate, in comparison with the first moment estimate provided by the Kalman filter. The deterioration of estimative accuracy as the dimension d increases, as well as the improvement in accuracy as N increases will be demonstrated for the exact particle filter. Of particular interest, however, will be how the accuracy of the ABC particle filter will be affected by these two control parameters.

Particle filtering was performed for sample sizes $N \in \{100, 400, 900, 1600, 2500\}$, for the SSM with dimension $d \in \{1, 2, 5, 10\}$. The initial state x_0 and variance σ^2 were treated as known throughout the numerical study, and results were obtained for $\sigma^2 \in \{0.1, 1.0, 5.0\}$.

To ensure repeatability, each procedure was implemented 10 times, with a different seed for the random number generator being set at the beginning of each execution. All of the results reported below were noted as being consistent over all repeated implementations. Finally, all simulations were executed using C++ on a CPU with a 2.00 GHz processor.

3.5.1 Analysis and Results

To summarise the performance of each of the particle filters, $\mathbb{L}_1$-errors were used. The $\mathbb{L}_1$-error, attributable to the estimate $\hat{x}_t$ of the true filtered state $\mathbb{E}_{\pi_t}[X_t] \in \mathbb{R}^d$, is defined as

$$\sum_{i=1}^{d} \left| \hat{x}_{t,i} - \mathbb{E}_{\pi_t}[X_{t,i}] \right| .$$

It is clear that an increase in dimension will result in an increase in $\mathbb{L}_1$-error that is not necessarily attributable to deterioration of the performance of the particle filtering procedure. In order to meaningfully compare the performance of the procedures for targeting state spaces of different dimensions, the errors were therefore adjusted to account for this. Define the dimension-adjusted $\mathbb{L}_1$-error at time t as

$$\frac{1}{d}\sum_{i=1}^{d}\left|\hat{x}_{t,i}-\mathbb{E}_{\pi_t}[X_{t,i}]\right|.$$

These will henceforth be referred to as the filtering errors, and they will often be described with reference to the particular procedure that produced the corresponding estimates, e.g. we may refer to both the ABC filtering errors and the exact filtering errors.

The exact filtering errors and the ABC filtering errors were calculated, using the Kalman filter to evaluate the true filtered state. Plots of these errors are shown in Figure 3.1 for the case where $\sigma^2 = 1$, with the rows corresponding to different dimensions of the SSM (with d increasing from top to bottom) and the columns corresponding to different values used for the particle system size N (with N increasing from left to right). In each subplot, the black line is the error sequence for the estimates obtained from the exact particle filter, whereas the red line is the error sequence for the ABC particle filter. The filtering errors obtained using $d \in \{1, 2\}$ are significantly smaller than those for $d \in \{5, 10\}$, and as a result, the scale of the error axes in the different rows of Figure 3.1 were chosen for the sake of clarity.

For all dimensions and particle system sizes considered, the filtering errors were observed to be significantly lower when targeting the exact SSM, than when using the ABC approximation. From the errors obtained using the exact particle filter, the deterioration in performance for increasing state space dimension can be observed clearly in Figure 3.1; the effect of particle system size on the accuracy of the exact particle filter is not obvious from this figure, although the expected improvement in accuracy is discernable when comparing the first and last columns (corresponding to $N = 100$ and $N = 2500$, respectively). This latter point is verified further below.

Staying with Figure 3.1, we can make some first remarks on the observed accu-

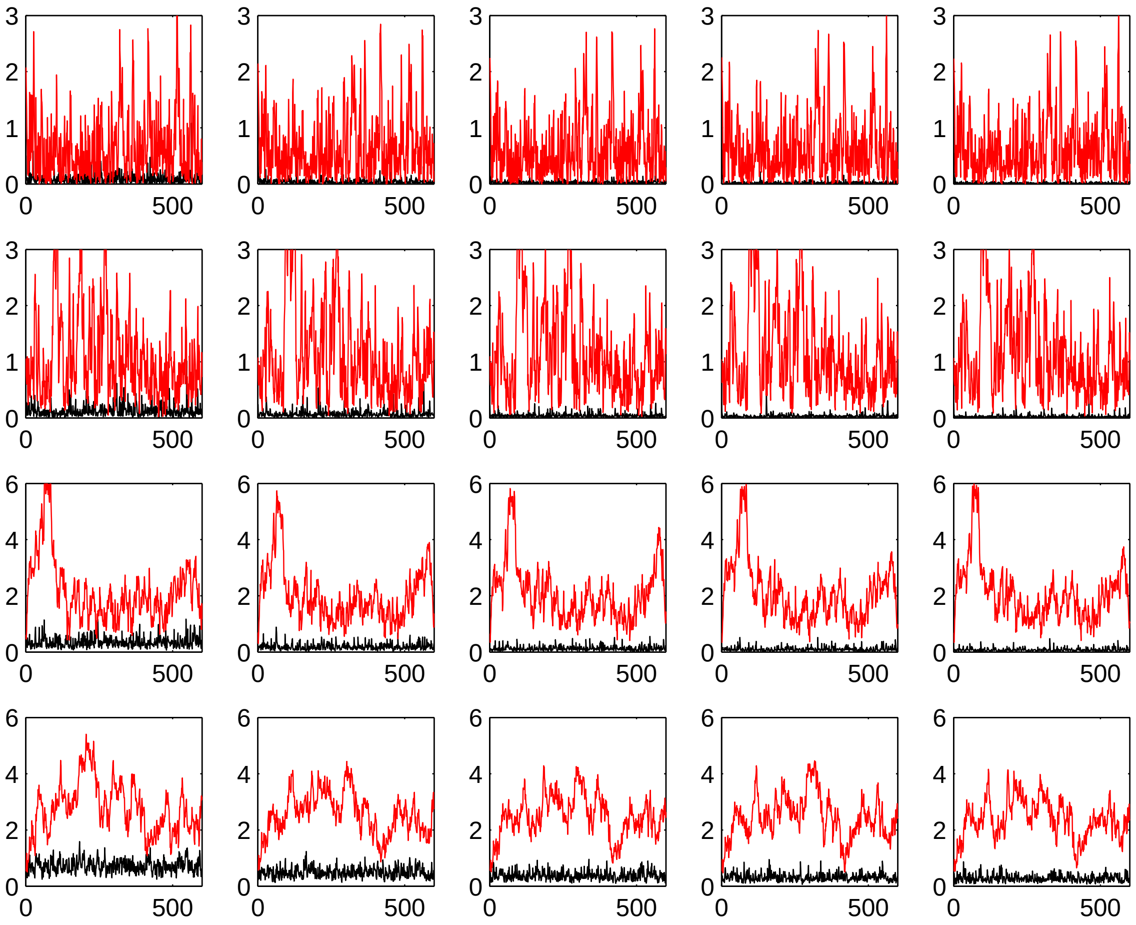


Figure 3.1: Filtering errors corresponding to the exact and ABC particle filters. The dimension-adjusted $\mathbb{L}_1$-errors are shown for the exact particle filtering (black) and ABC particle filtering (red) estimates of the filtered state in the linear Gaussian SSM (3.11)-(3.12) with $\sigma^2 = 1$.

racy of the ABC particle filter. Firstly, it is notable that, although the error observed under the ABC approximation is more volatile than the error for the exact particle filter, it appears stable with respect to the time parameter. This is an empirical verification of Proposition 3.3.2. It can also be seen in Figure 3.1 that the ABC filtering error increases as the dimension d of the SSM increases. From these results, it appears that the size of the particle system, N, has no effect on the accuracy of the ABC particle filter; this is explored in further detail below.

Figure 3.2 displays the histograms produced by all of the exact filtering errors observed under each parameter pairing $\{N, d\}$; each histogram displays the errors

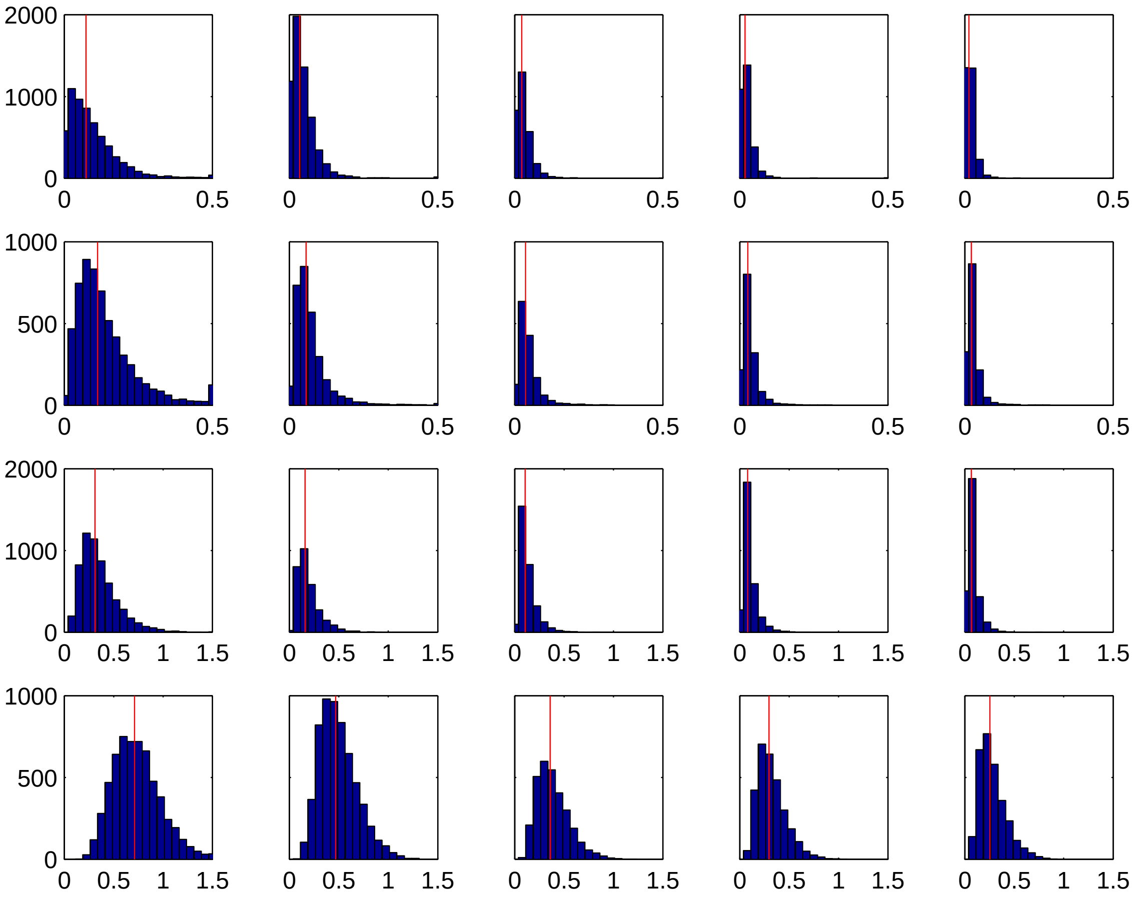


Figure 3.2: Histograms illustrating the skewness in the distribution of the exact filtering errors, obtained from applying exact particle filtering to the linear Gaussian SSM (3.11)-(3.12) with $\sigma^2 = 1$. The errors at each time point $t = 1, \ldots, T$, from all 10 independent implementations were included. The median filtering errors are marked by the red vertical lines.

observed for all $t \in \{1, \ldots, T\}$ and across all independent implementations of the exact particle filter for a given $\{N, d\}$. These histograms indicate that the distributions of the filtering errors are significantly skewed for low values of d. This was observed for both the exact and ABC filtering errors, and across all values of the noise parameter $\sigma^2 \in \{0.1, 1, 5\}$.

In order to gain more insight into the behaviour of these errors, we therefore consider the median of the filtering errors resulting from each procedure over the entire period $[0, T]$. The median filtering errors for the estimated first moment of the state in the SSM and its ABC approximation are presented in Tables 3.1 and

	N=100	N=400	N=900	N=1600	N=2500
d = 1	0.0754	0.0336	0.0248	0.0177	0.0145
d = 2	0.1077	0.0590	0.0368	0.0280	0.0218
d = 5	0.3125	0.1623	0.1078	0.0803	0.0646
d = 10	0.7038	0.4703	0.3528	0.2860	0.2590

Table 3.1: Median dimension-adjusted $\mathbb{L}_1$-errors observed over the period $[0, T]$ for the estimate of the filtered state in the linear, Gaussian SSM (3.11)-(3.12), averaged over 10 implementations of the exact particle filter.

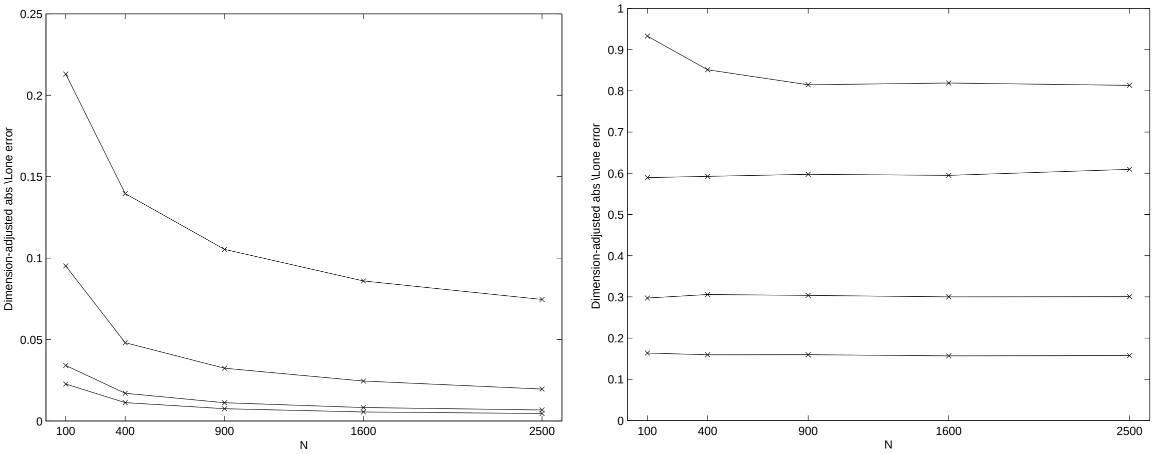
	N=100	N=400	N=900	N=1600	N=2500
d = 1	0.5007	0.4982	0.4722	0.4883	0.4770
d = 2	0.8864	0.9242	0.9266	0.9312	0.9264
d = 5	1.9369	1.7823	1.9341	1.9568	1.9762
d = 10	2.7313	2.5762	2.4918	2.4104	2.3565

Table 3.2: Median dimension-adjusted $\mathbb{L}_1$-errors observed over the period $[0, T]$ for the estimate of the filtered state in the linear, Gaussian SSM (3.11)-(3.12), averaged over 10 implementations of the ABC particle filter.

3.2, respectively, and displayed in Figures 3.3a and 3.3b, respectively.

Using the median filtering errors, it is simpler to observe the effect of the particle system size and dimension of the state on the performance of the exact particle filter. As expected, it can be seen in Table 3.1 that the errors decrease as N increases and, independently, as d decreases. This is further demonstrated in Figure 3.3a for noise parameter $\sigma^2 = 1$, and was found to hold for $\sigma^2 = 0.1$ and $\sigma^2 = 5$ also.

Table 3.2 and Figure 3.3b confirm that the accuracy of the ABC particle filter deteriorates as d increases and suggest that, for state spaces of moderate dimension ($d \leq 5$), the accuracy of the ABC particle filtering estimate is largely insensitive to the choice of $N \geq 100$. For $d = 10$, the ABC particle filter was observed to be less accurate when using fewer than 900 particles, however an insensitivity to $N \geq 900$ was observed in this dimension. Recalling the upper bound on the $\mathbb{L}_p$-error of the ABC particle filtering estimate, given in (3.9), we remark that this insensitivity to



(a) Exact Filter

(b) ABC Filter

Figure 3.3: Median exact (a) and ABC (b) filtering errors for the estimated filtered state in the linear Gaussian SSM (3.11)-(3.12) with $\sigma^2 = 1$. The errors for constant d are connected by the solid lines; from bottom to top, the dimensions used were $d \in \{1, 2, 5, 10\}$.

N is the first indication that the ABC bias dominates the $\mathbb{L}_p$-error. This is discussed further in the next section.

3.6 Example: A Nonlinear SSM

We consider the nonlinear Gaussian SSM, with $d_x = d_y = d$:

$$X_t = \frac{X_{t-1}}{2} + \frac{25X_{t-1}}{1+X_{t-1}^2} + 8\cos(1.2t) + V_t, \quad t \geq 1, \tag{3.13}$$

$$Y_t = \frac{X_t^2}{20} + Z_t, \quad t \geq 1, \tag{3.14}$$

with

$$V_t, Z_t \sim \mathcal{N}\left(0, \sigma^2 I_d\right) \quad (i.i.d.) \quad \text{and} \quad V_t \perp\!\!\!\perp Z_t \quad t \geq 1$$

$$X_0 = 0_d, \quad \text{a } d\text{-length vector with all elements equal to 0.}$$

A true filtered state, against which to measure the accuracy of the estimated filtered state, is unavailable for nonlinear SSMs; we shall use the exact particle filter, executed with $N = 10000$ particles, as a basis against which to measure the accuracy of the different procedures. The accuracy of particle filtering via the ABC approximation of the nonlinear SSM (3.13)-(3.14) is once again of interest and, as before, the filtering errors will be defined as the dimension-adjusted $\mathbb{L}_1$-errors.

Of added interest will also be the algorithmic performance of the ABC particle filter in comparison to the exact particle filter; this will be measured in terms of the algorithm's resampling rate, i.e. the proportion of time steps at which resampling was triggered by virtue of the estimated ESS (2.16) being less than the user-defined threshold of $Th = N/2$. Note that, since resampling is not available at the final time step, the resampling rate takes the form

$$\frac{1}{T-1}\sum_{t=1}^{T-1}\mathbb{I}\left(\widehat{ESS}\left(\left\{W_t^{(i)}\right\}_{i=1}^{N}\right) < \frac{N}{2}\right).$$

We will also consider MPDABC particle filtering, using the MPDABC kernel (3.10), with $J = 10$ pseudo-data per particle. This is expected to be more computationally expensive, yet less degenerate than the ABC particle filter, and we recall that it is of interest to measure any improvements in accuracy that it may be able to offer, whilst maintaining a reasonable computational expense.

For this example, we will also consider the accuracy of particle filtering via the Empirical and Theoretical Rejection SMC procedures detailed in Section 2.3.2. These will be referred to as the ERSMC particle filter and the TRSMC particle filter, respectively, and the accuracy of the estimates resulting from the application of these procedures to the nonlinear SSM and its ABC approximation will be studied. Note that, when using the indicator function as the ABC kernel in the ABC approximation of the SSM, the ERSMC ABC particle filter and the TRSMC ABC particle filter are equivalent. Thus, we will not consider the application of the ERSMC particle filter to the ABC approximation of the SSM.

The SSM (3.13)-(3.14) was targeted for varying dimensions and values of the noise parameter: $d \in \{1, 2, 5, 10\}$, $\sigma^2 \in \{1, 5, 10\}$ and each particle filter was run using N particles, with $N \in \{100, 400, 900, 1600, 2500\}$. For the ABC filtering procedures, the sequence of tolerance parameters $\{\epsilon_t\}_{t=1}^{T}$ were tuned adaptively, using the online procedure detailed in Section 3.4, with $\alpha = \lfloor 0.8N \rfloor$, i.e. the largest integer that is less than or equal to $0.8N$. To ensure consistency of the results, we repeated the implementation of each particle filter 10 times and all simulations were executed using C++ on a CPU with a 2.33 GHz processor.

3.6.1 Observed Algorithm Degeneracy

As described in Section 3.4, the ABC particle filter has the potential to suffer from algorithmic collapse, due to the potential for the incremental weights at each time step to be evaluated as 0. Consider the observation equation (3.14). Since the observations only provide information about the hidden state through a quadratic term, there is uncertainty about the sign of the state. As a result, the posterior here is bi-modal, resulting in a higher susceptibility to the complete collapse of the ABC particle filter

We will again use the method described in Section 3.4.2 for adaptively tuning our sequence of ABC tolerance parameters, in order to protect against algorithm collapse. We will see here that, although this approach is effective for the majority of parameter combinations $\{N, d, \sigma^2\}$, for some particularly unfavourable parameter triplets, the ABC particle filter collapsed regardless, and a suitable estimate was not provided.

The parameter settings that resulted in complete collapse of the ABC and MPDABC particle filters are presented in Table 3.3. As can be seen from the table, small particle system sizes, high-dimensional state spaces and low values for the noise parameter all appear to contribute to the collapse of these filters when targeting the ABC approximation of the nonlinear SSM. Furthermore, it is clear

			ABC		MPDABC		
N	d	σ^2	SMC	RSMC	SMC	ERSMC	TRSMC
100	2	1	✗	✗			
100	5	1	✗	✗			✗
100	10	1	✗	✗	✗	✗	✗
100	10	5	✗	✗			
100	10	10	✗	✗			
400	10	1	✗	✗			
900	10	1	✗	✗			

Table 3.3: Table showing parameter combinations for which the ABC particle filter collapsed at every independent implementation. For all other model and algorithm parameter combinations, multiple estimates of the filtered state process were obtained.

that the use of the MPDABC kernel results in a decrease in the susceptibility of the algorithm to a complete collapse, as was expected.

3.6.2 Analysis and Results

We now present an analysis of the accuracy and algorithmic performance of the ABC particle filters, both with and without the use of the RSMC methodology. Throughout, we present the results obtained when applying the particle filters to the nonlinear SSM (3.13)-(3.14) with the noise parameter fixed at $\sigma^2 = 5$. We remark here that all conclusions drawn below were observed to also hold for $\sigma^2 \in \{1, 10\}$ and across all independent implementations of each particle filter.

ABC Filtering vs. Filtering

We now analyse the accuracy of the ABC particle filter, by comparing the resulting filtering errors with the filtering errors obtained using the exact particle filter. For the moment, we focus on the exact and ABC particle filters that use dynamic multinomial resampling; we do not yet consider the use of RSMC.

Figure 3.4 shows examples of the exact filtering errors (black line) and ABC

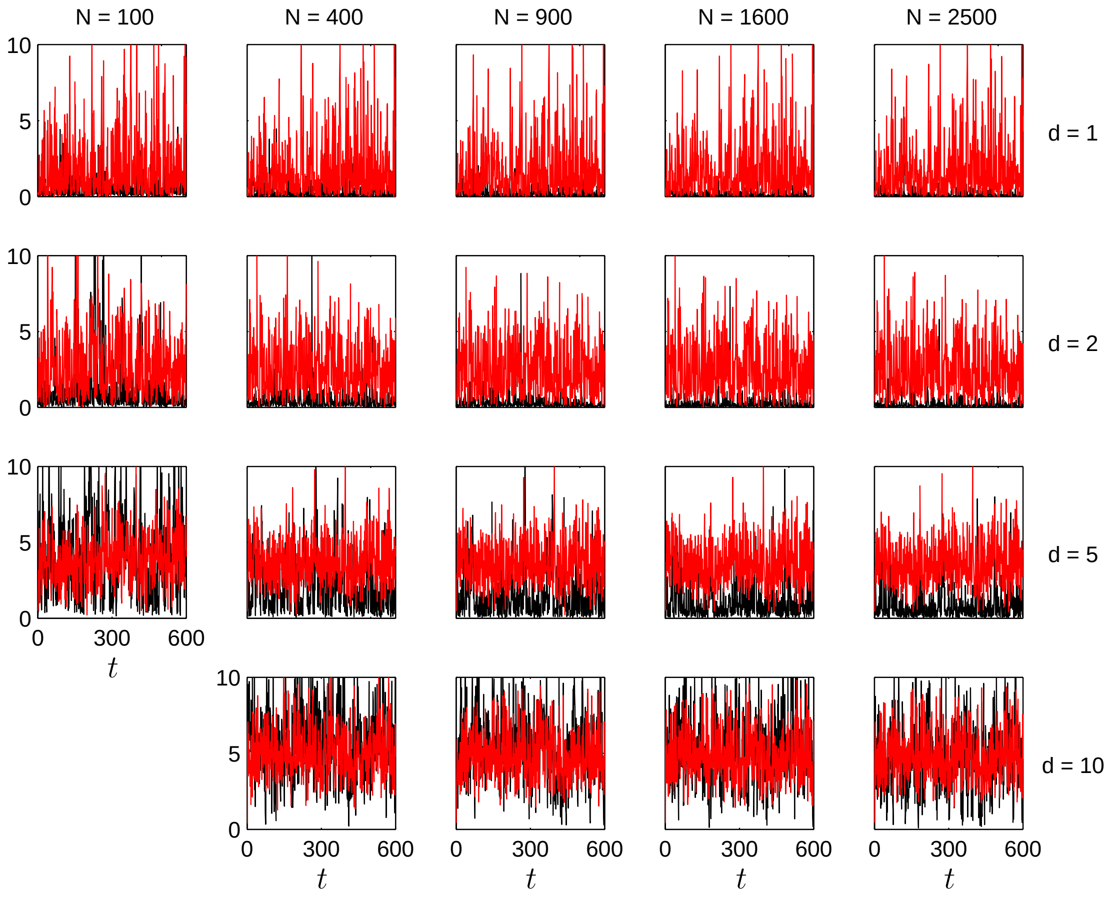


Figure 3.4: The dimension-adjusted $\mathbb{L}_1$-errors for the exact particle filtering (black) and ABC particle filtering (red) estimates of the filtered state in the nonlinear SSM (3.13)-(3.14) with $\sigma^2 = 5$.

filtering errors (red line). These plots correspond to a single implementation of each of the particle filters under each parameter pairing $\{N, d\}$, and the omitted plot corresponds to the parameter pairing $\{N, d\} = \{100, 10\}$, where the ABC particle filter collapsed completely for all 10 attempted implementations. As is clear, the errors for the ABC particle filter are much more comparable to those obtained using the exact particle filter than was the case for the previous example. For the higher dimensions ($d \in \{5, 10\}$), the ABC particle filter appears to be competing well with the exact particle filter. This requires further exploration.

Figure 3.5 shows, for each parameter pairing $\{N, d\}$, the differences between a set of exact filtering errors, and a set of ABC filtering errors, each of which were

	N = 100	N = 400	N = 900	N = 1600	N = 2500
d = 1	0.2458	0.1239	0.0871	0.0668	0.0550
d = 2	0.4503	0.2168	0.1463	0.1140	0.0975
d = 5	3.4395	1.6924	0.9165	0.6447	0.5266
d = 10	6.5746	5.7356	5.2929	5.0199	4.7088

Table 3.4: Median dimension-adjusted $\mathbb{L}_1$-errors observed over the period $[0, T]$ for the estimate of the filtered state in the nonlinear SSM (3.13)-(3.14), averaged over 10 implementations of the exact particle filter.

	N = 100	N = 400	N = 900	N = 1600	N = 2500
d = 1	1.1382	1.1226	1.1186	1.1074	1.1098
d = 2	2.3458	2.2872	2.2832	2.2835	2.2355
d = 5	3.8945	3.7086	3.6350	3.6397	3.6229
d = 10	-	4.9269	4.8108	4.7995	4.7547

Table 3.5: Median dimension-adjusted $\mathbb{L}_1$-errors observed over the period $[0, T]$ for the estimate of the filtered state in the nonlinear SSM (3.13)-(3.14), averaged over 10 implementations of the ABC particle filter.

obtained over a single implementation of each particle filter. These differences are represented by the grey line in each plot, where negative values indicate time points at which the ABC particle filter outperformed the exact particle filter; the black horizontal line on each plot is at zero. Again, similar results were observed over all independent implementations of the exact and ABC particle filters. The red horizontal line on each plot gives the corresponding difference between the median filtering errors and the median ABC filtering errors, which are given in Tables 3.4 and 3.5 respectively; these median errors have been averaged over all 10 independent implementations of each procedure.

There are three main conclusions to draw from these results. Firstly, note that, for all parameter pairings $\{N, d\}$, there are time instances at which the ABC particle filtering estimate is more accurate than the exact particle filtering estimate. As the dimension of the state space increases, so does the number of instances at which this

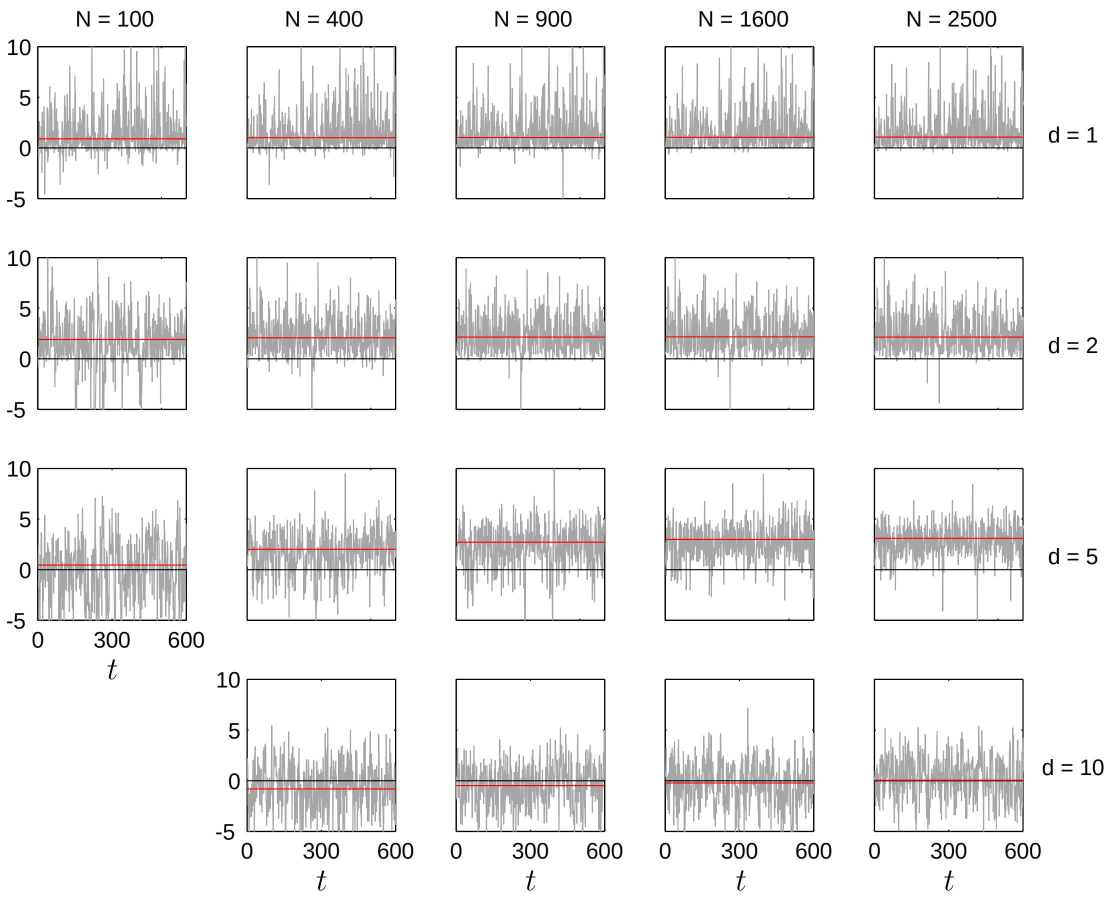
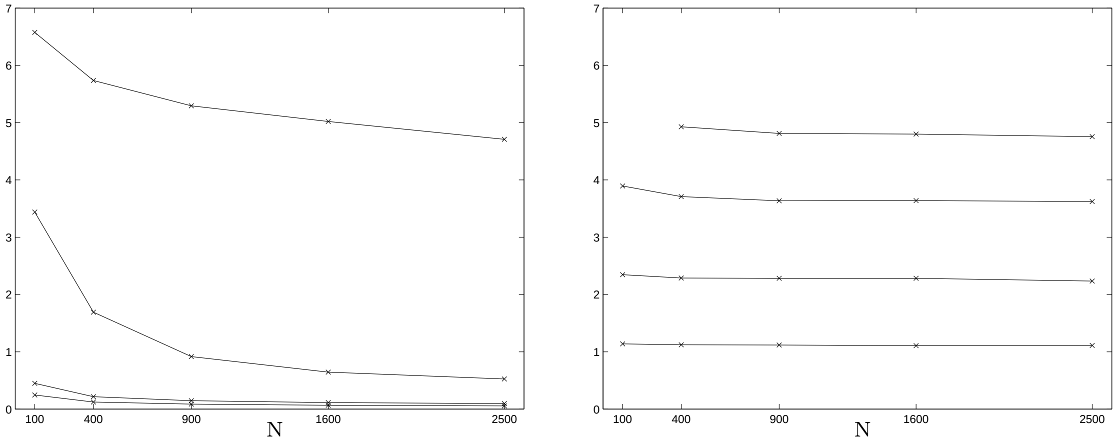


Figure 3.5: Differences between the exact filtering errors and ABC filtering errors obtained in estimating the filtered state in the nonlinear SSM (3.13)-(3.14) with $\sigma^2 = 5$. The grey line shows the ABC filtering errors subtracted from the exact filtering errors, at each time point $t = 1, \ldots, T$. The red line is the corresponding difference in the median exact filtering errors and median ABC filtering errors, which were averaged over 10 independent implementations of each filter. The black line is zero; negative values indicate instances at which the ABC particle filter was more accurate than the exact particle filter using the same number of particles.

happens. For low dimensions, however, the exact particle filter still outperforms the ABC particle filter when assessing performance over the entire time horizon $[0, T]$; this is seen in the difference in the median errors, indicated by the red horizontal lines.

Secondly, the ABC particle filter appears to marginally outperform the exact particle filter in high dimensions ($d = 10$), when using a relatively low number of particles ($N \leq 900$), and it remains competitive for high dimensions as the number



(a) Exact Particle Filter

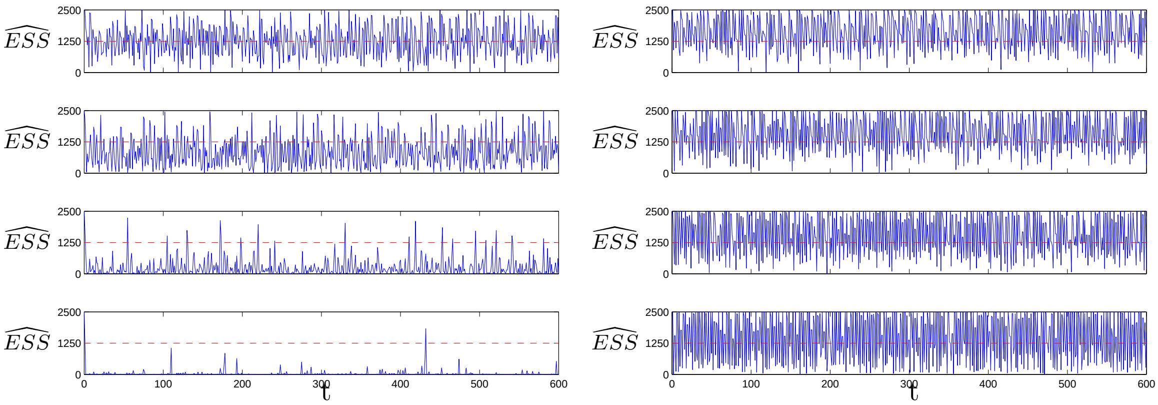
(b) ABC Particle Filter

Figure 3.6: Median exact (a) and ABC (b) filtering errors for the estimated filtered state in the nonlinear SSM (3.13)-(3.14) with $\sigma^2 = 5$. The errors for constant d are connected by the solid lines; from bottom to top, the dimensions used were $d \in \{1, 2, 5, 10\}$.

of particles increases. The ABC particle filter is also seen to be competitive when using only 100 particles for $d = 5$.

Thirdly, as the number of particles increases, the accuracy of the ABC particle filter, relative to that of the exact particle filter, decreases. It is well known that the accuracy of the particle filter increases as N increases, and so this is likely to be due to a slower improvement (if any) in the accuracy of the ABC particle filter than the improvement in the accuracy of the exact particle filter, as was also observed in Example 3.5. This is confirmed in Figures 3.6a and 3.6b, which show the effect of N and d on the errors obtained using the exact and ABC particle filters, respectively. Indeed, from Figure 3.6b, it is clear that, as was the case with the linear Gaussian SSM, the accuracy of the ABC filtered state in the nonlinear SSM is relatively insensitive to any increase in the number of particles. The ABC particle filter was applied to the nonlinear SSM using 10, 000 particles and, as with the previous example, no improvement in performance was noted over 10 independent implementations.

We recall the upper bound (3.9) to the $\mathbb{L}_p$-error for the ABC particle filtering



(a) Exact Particle Filter

(b) ABC Particle Filter

Figure 3.7: Plots of the estimated ESS, calculated at each time step of the exact (a) and ABC (b) particle filters. For all plots show, $N = 2500$ particles were used to estimate the filtered state in d-dimensional nonlinear SSM (3.13)-(3.14) with, from top to bottom, $d \in \{1, 2, 5, 10\}$, and $\sigma^2 = 5$. The resampling threshold $(N/2)$ is marked on both plots as the red dashed line.

estimate, and we note once again that the filtering errors are dominated by the bias. We further remark that the ABC filtering errors, which are dimension-adjusted, increase significantly with the dimension. Since the bias dominates these errors, this suggests that the deterministic ABC bias may increase with the dimension of the SSM. Consider the method used for adaptively-tuning ϵ_t. For fixed α, as one increases the dimension of the data, the generated value of ϵ_t will increase also. This increased value of ϵ_t contributes to the increase in the bias that is suggested here.

From Figure 3.6a, it is clear that the exact filtering error explodes as the dimension increases. This demonstrates the difficulty of performing exact particle filtering for high-dimensional hidden states, and can be linked to the weight degeneracy issue. To observe the extent to which weight degeneracy affects the performance of each of the particle filters, we consider the ESS.

Figures 3.7a and 3.7b present the estimated ESS at each time point $t = 1, \ldots, T$, as calculated during a 2500-particle implementation of the exact particle filter and the ABC particle filter, respectively, for $d \in \{1, 2, 5, 10\}$. For the exact particle filter, we can see that, as the dimension of the state increases, the ESS decreases such that, for high-dimensional states, it is consistently evaluated as being significantly lower than N. Recall that low ESS is an indication of there being very few particles of significant weight at each iteration, and so the accuracy of the SMC approximation degrades with increasing dimension for the exact particle filter. In an SIS procedure, i.e. a particle filter without resampling, this translates to the need to increase the number of particles with the dimension in order to maintain a given level of estimator accuracy; it can be shown (e.g Bengtsson et al., 2008) that the sample size N is required to grow exponentially in the dimension of the hidden state.

In contrast, the ESS plot for the ABC particle filter appears much more resilient to increases in dimension, indicating that the weight degeneracy issue affects the ABC particle filter to a much lesser extent than the exact particle filter, as was expected. For an SIS procedure targeting the ABC approximation of the SSM, i.e. ABC SIS, this translates to the number of particles required for a given level of estimator accuracy being sub-exponential. Indeed, these observations are an empirical verification of Proposition 5.2 of Beskos et al. (2012). That result implies that the numerical part of the $\mathbb{L}_p$-error of the estimate produced by an ABC SIS procedure, can be kept stable as the dimension of the state increases, using a particle system size N that grows sub-exponentially in the dimension of the hidden state.

The above observations, along with the theoretical result of Beskos et al. (2012), indicate that the SMC error associated with the ABC particle filter should be significantly lower than the corresponding error of the exact particle filter when applied to SSMs with high-dimensional hidden states. The fact that the ABC filtering errors are not observed to be significantly lower than the exact filtering errors further exemplifies the fact that the ABC bias dominates the $\mathbb{L}_p$-error (3.9) of the ABC particle filtering estimate of the filtered state.

	N = 100	N = 400	N = 900	N = 1600	N = 2500
d = 1	1.1615	1.1561	1.1853	1.1538	1.1429
d = 2	2.3662	2.1631	2.3248	2.3077	2.2660
d = 5	3.7982	3.6785	3.6331	3.6467	3.6539
d = 10	5.1446	4.8983	4.7948	4.7591	4.7450

Table 3.6: Median dimension-adjusted $\mathbb{L}_1$-errors observed over the period $[0, T]$ for the estimate of the filtered state in the nonlinear SSM (3.13)-(3.14), obtained using MPDABC particle filtering.

	N = 100	N = 400	N = 900	N = 1600	N = 2500
d = 1	102.16%	104.80%	106.14%	99.74%	97.90%
d = 2	98.74%	93.64%	99.75%	99.57%	101.09%
d = 5	99.65%	99.56%	100.58%	99.88%	99.48%
d = 10	-	98.80%	100.60%	99.93%	100.25%

Table 3.7: Relative magnitude of the median dimension-adjusted $\mathbb{L}_1$-errors of the estimated filtered state in the nonlinear SSM (3.13)-(3.14), obtained using MPDABC particle filtering, in comparison to those obtained using ABC particle filtering (Table 3.5).

MPDABC Filtering vs. ABC Filtering

As mentioned above, it is of interest to explore whether any improvement in the accuracy of an ABC particle filter can be empirically measured when using a composition of $J > 1$ independent indicator functions as the ABC kernel, as opposed to using only one. This multiple-pseudo data approach was tested for the nonlinear SSM (3.13)-(3.14), and the results are presented here with reference to the corresponding results for the single-pseudo data ABC particle filter. Once again, the accuracy of the particle filters is measured with respect to estimated filtered state obtained using an exact particle filter with $10,000$ particles. The median filtering errors for the MPDABC particle filter are presented in Table 3.6, with the relative magnitude of this error when compared to the corresponding ABC filtering error, given in Table 3.7.

	N = 100	N = 400	N = 900	N = 1600	N = 2500
d = 1	32.00%	31.29%	31.15%	31.02%	30.88%
d = 2	34.84%	34.52%	34.31%	34.24%	34.24%
d = 5	38.85%	38.95%	39.13%	39.10%	39.08%
d = 10	-	41.47%	41.74%	41.65%	41.74%

Table 3.8: Average resampling rates obtained over 10 implementations of the ABC particle filtering procedure, when applied to the nonlinear SSM (3.13)-(3.14).

	N = 100	N = 400	N = 900	N = 1600	N = 2500
d = 1	23.62%	23.39%	23.32%	23.31%	23.39%
d = 2	28.28%	28.25%	28.28%	28.20%	28.10%
d = 5	33.02%	32.79%	32.50%	32.40%	32.47%
d = 10	38.31%	38.48%	38.51%	38.53%	38.58%

Table 3.9: Average resampling rates obtained over 10 implementations of the MPDABC particle filtering procedure, when applied to the nonlinear SSM (3.13)-(3.14).

From Table 3.7, we can conclude that, empirically, there does not seem to be any consistent improvement offered by the use of a composition of $J = 10$ independent indicator functions, as opposed to a single indicator function as the ABC kernel in the approximation of the SSM. On the other hand, as indicated by Table 3.3, the MPDABC filter is significantly less prone to collapsing, and larger values of J may result in slight improvements in filter accuracy. Any advantage offered by the MPDABC approximation must be weighed up against the increase in computational cost.

The computational expense of the exact particle filter is determined to a large extent by the resampling rate, and so we consider first the effect of whether any improvement in the resampling rate is observed when using the MPDABC particle filter. The resampling rates for the ABC particle filter and the MPDABC particle filter are presented in Tables 3.8 and 3.9, respectively.

Comparing Tables 3.8 and 3.9, it is clear to see that the use of the MPDABC

	N = 100	N = 400	N = 900	N = 1600	N = 2500
d = 1	0.08	0.45	1.58	4.26	9.50
d = 2	0.11	0.61	2.01	5.16	11.26
d = 5	0.25	1.20	3.43	7.92	16.01
d = 10	-	2.89	7.29	14.89	27.18

Table 3.10: Mean processing times (in seconds) for the ABC particle filter applied to the nonlinear SSM (3.13)-(3.14).

	N = 100	N = 400	N = 900	N = 1600	N = 2500
d = 1	0.26	1.16	3.05	6.51	12.37
d = 2	0.43	1.87	4.72	9.69	17.74
d = 5	1.15	4.76	11.32	21.61	36.81
d = 10	3.31	13.40	30.87	56.67	92.16

Table 3.11: Mean processing times (in seconds) for the MPDABC filter applied to the nonlinear SSM (3.13)-(3.14).

kernel results in lower resampling rates. This is to be expected; as discussed in Section 3.4, the use of the MPDABC kernel should result in filtering weights with reduced variance, according to the principle of Rao-Blackwellisation. Thus, when using the MPDABC kernel in favour of the original ABC kernel, the effect of weight degeneracy is decreased and the particle filter is required to resample at fewer steps.

We consider also the processing times for the ABC and MPDABC particle filters; these are presented in Tables 3.10 and 3.11.

As can be seen from Tables 3.10 and 3.11, the use of multiple pseudo-data in the ABC approximation comes with a significant increase in computational cost, in spite of the observed improvement in the resampling rates. From this, we deduce that the increase in computational cost is associated with sampling J pseudo-data instead of a single pseudo-datum.

The results here indicate that the MPDABC particle filter offers no consistent improvement in accuracy over the ABC particle filter when estimating the filtered state in the nonlinear SSM (3.13)-(3.14). This is in spite of a marked improvement in the associated observed resampling rates and, therefore, a decreased susceptibility to the weight degeneracy issue. This decrease in the effects of weight degeneracy should improve the SMC approximation of the MPDABC filter; the fact that this is not evident in the filtering errors suggests that, once again, the bias in the MPDABC filter dominates the corresponding $\mathbb{L}_p$-error between the SMC estimate of the MPDABC filtered state and the exact filtered state.

Rejection SMC Filtering vs. SMC Filtering

Finally, we report the observed effects of including a rejection step in the SMC filtering procedures, as described in Section 2.3.2, for application to both the nonlinear model described above and its ABC and MPDABC approximations.

Recall that the objective here is to ascertain whether the theoretical advantage offered by the RSMC method, in terms of improved asymptotic variance in the SMC approximation of the joint smoothing distribution, can be observed in the corresponding filtering estimates for finite N. In order to simplify the analysis, we consider the mean, over all dimensions, of the filtered estimate at each time point $t \in [0, T]$. We then estimate the standard deviation of these mean filtering estimates at each t using the standard error, taken over all independent implementations of each procedure. The median of these standard errors, taken over all time points $t \in [0, T]$, is reported. Henceforth, this will be referred to as the median standard error of the filtering estimates, or simply the median standard error.

First, we consider the use of RSMC for performing exact particle filtering. Table 3.12 presents the median standard errors obtained using the SMC, ERSMC and TRSMC particle filters, for each parameter pairing $\{N, d\}$. All the results reported here correspond to the nonlinear SSM (3.13)-(3.14), with $\sigma^2 = 5$, although similar results were observed for $\sigma^2 \in \{1, 10\}$.

The median standard errors in Table 3.12 are also presented in Figure 3.8. For each of the exact particle filters considered, the median standard errors correspond-

	N = 100	N = 400	N = 900	N = 1600	N = 2500
d = 1	0.2037 0.1923 0.1971	0.1046 0.0940 0.0941	0.0716 0.0626 0.0644	0.0528 0.0480 0.0473	0.0425 0.0420 0.0415
d = 2	0.4337 0.4640 0.4329	0.1514 0.1400 0.1448	0.0989 0.0934 0.0957	0.0685 0.0724 0.0710	0.0588 0.0601 0.0574
d = 5	2.1624 2.2145 2.1600	1.3987 1.5048 1.4242	0.8047 0.8604 0.8188	0.5098 0.5347 0.5348	0.3366 0.3269 0.3284
d = 10	1.9551 1.9514 1.9215	1.8297 1.7901 1.8808	1.7830 1.7884 1.7933	1.7443 1.6902 1.6600	1.6156 1.6085 1.6060

Table 3.12: Median standard errors of the exact particle filtering estimates. The values in each cell correspond to the estimates produced by the SMC (top), ERSMC (middle) and TRSMC (bottom) particle filters, targeting the nonlinear SSM (3.13)-(3.14) specified by $\sigma^2 = 5$.

ing to those implementations that used $N = 100$ were found to be significantly larger than those corresponding to the implementations that used $N \geq 400$. Since we are interested in the behaviour of the estimator variance as N grows, the median standard errors corresponding to $N = 100$ are therefore omitted from Figure 3.8 for the sake of clarity in presenting the median standard errors for $N \geq 400$.

From Figure 3.8, we note that, when targeting the distribution of the one-dimensional hidden state, a marginal improvement in the estimated variance of the filtering estimates is observed when using the RSMC methodology. This improvement over the SMC procedure is observed for both the TRSMC and ERSMC procedures. As the dimension of the hidden state grows, the RSMC procedures display a lower median standard error than the SMC procedure for a number of parameter pairings $\{N, d\}$, however this advantage is not consistently observed, and any improvement that the RSMC procedures appear to offer is extremely marginal.

We remark that Figure 3.8 also illustrates the deterioration of the filtering procedure as the dimension d increases. In low dimensions, the variance of the filtering

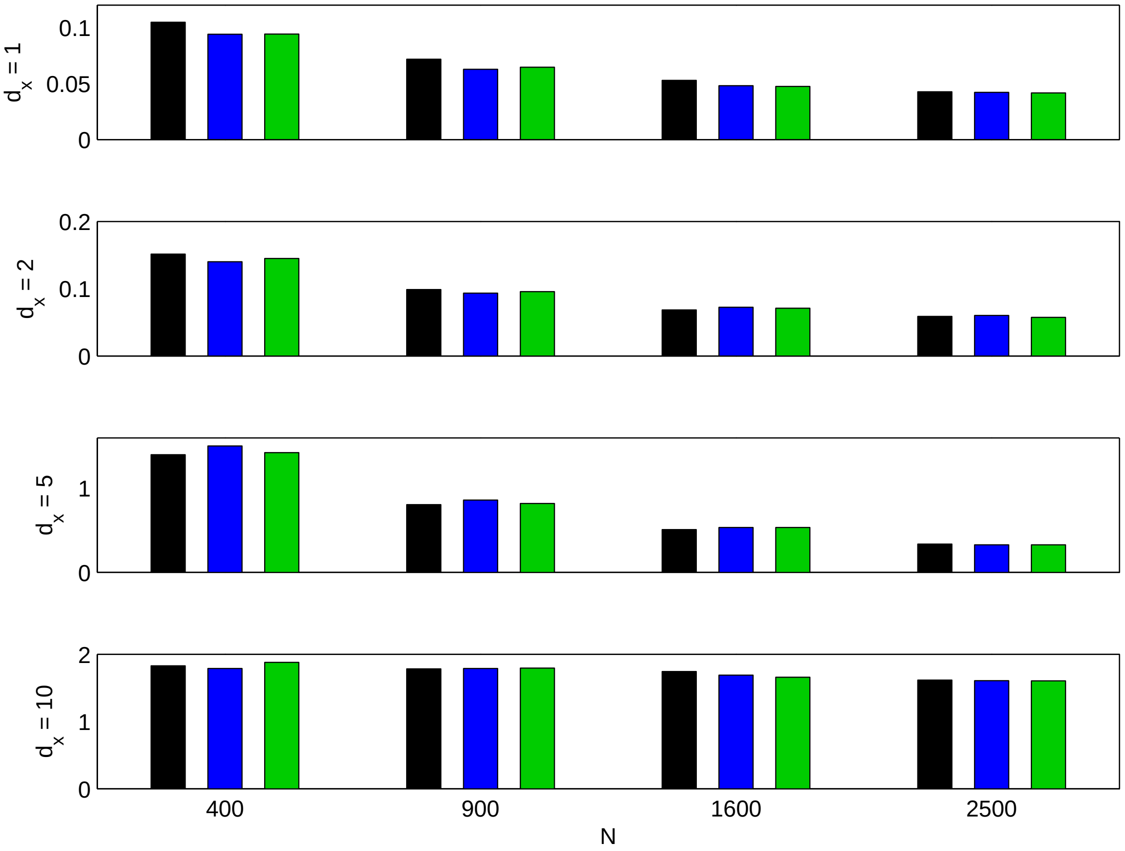


Figure 3.8: Median standard errors of the exact SMC, exact ERSMC and exact TRSMC particle filtering estimates of the filtered state in the nonlinear SSM (3.13)-(3.14), with $\sigma^2 = 5$. Each plot illustrates the standard errors corresponding to the estimates produced by the exact particle filters using SMC (black), ERSMC (blue) and TRSMC (green) kernels.

estimates can be seen to decrease as N increases, as would be expected. In high dimensions, however, the improvement in estimator variance is much less significant as N grows, due to the increase in SMC weight degeneracy and the resulting deterioration in the SMC approximation of both the joint smoothing distribution and the filtering distribution.

We now consider the use of RSMC within particle filtering procedures for estimating the ABC approximation of the filtered state. Recall that for the single pseudo-data ABC approach considered here, i.e. where the ABC kernel is the single indicator kernel (3.5), the ERSMC and TRSMC acceptance probabilities defined in (2.17) and (2.19) coincide, and thus the ERSMC and TRSMC procedures are equiv-

	N = 100	N = 400	N = 900	N = 1600	N = 2500
d = 1	0.7434 0.6883	0.4110 0.3393	0.2910 0.2564	0.2094 0.1804	0.1718 0.1543
d = 2	0.9782 0.9322	0.4781 0.4047	0.3075 0.2850	0.2381 0.2091	0.1913 0.1747
d = 5	0.8539 0.7847	0.4156 0.3766	0.2834 0.2495	0.2091 0.1895	0.1635 0.1524
d = 10	- -	0.5169 0.4799	0.3367 0.3232	0.2471 0.2291	0.1930 0.1879

Table 3.13: Median standard errors of the ABC filtering estimates. The values in each cell correspond to the estimates produced by the SMC (top) and RSMC (bottom) particle filters, targeting the ABC approximation of the nonlinear SSM (3.13)-(3.14) specified by $\sigma^2 = 5$.

alent. We also consider the use of the MPDABC kernel (3.10), where the ERSMC and TRSMC procedures are once again distinct.

As before, we present the median standard errors for the filtering estimates generated by the SMC and RSMC particle filters when targeting the ABC and MPDABC approximations of the filtered state. The median standard errors for the ABC filtering estimates are reported in Table 3.13 and illustrated in Figure 3.9. The standard errors obtained when using the MPDABC approximation are presented in Table 3.14 and Figure 3.10.

From Table 3.13 and Figure 3.9, it is immediately clear that the estimates provided by the RSMC ABC particle filter have markedly lower variance than those provided by the SMC ABC particle filter. This is an empirical verification of the theoretical improvement in the asymptotic estimator variance, and it is encouraging that this advantage can be realised for finite N. We also note that Figure 3.9 demonstrates the relative robustness of the ABC particle filter with respect to the SMC weight degeneracy issue and the resulting deterioration of the SMC approximation. This is shown by the fact that, even in high dimensions, the median standard errors for both the SMC and RSMC ABC particle filters decrease as N increases.

From Table 3.14 and Figure 3.10, we see that the RSMC procedure once again

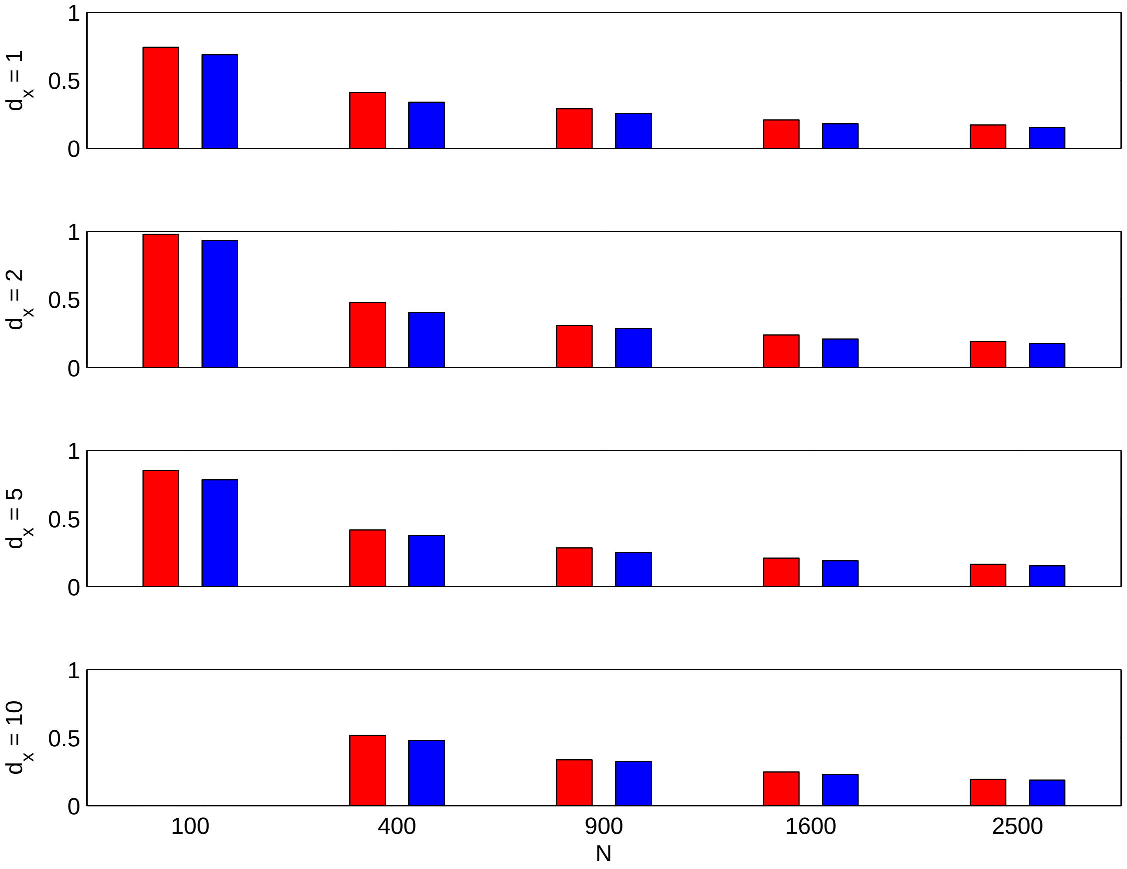


Figure 3.9: Median standard errors of the ABC SMC and RSMC particle filtering estimates of the filtered state in the nonlinear SSM (3.13)-(3.14), with $\sigma^2 = 5$. Each plot illustrates the standard errors corresponding to the estimates produced by the ABC particle filters using SMC (red) and RSMC (blue) kernels.

produces MPDABC filtered estimates of lower variance than those produced by the SMC particle filter and that this advantage is consistent for all values of $N \geq 400$. Figure 3.10 also indicates that the MPDABC particle filter also displays a robustness to the degeneracy issues that cause the poor performance of the particle filter in high dimensions.

The above results show that the reduction in asymptotic estimator variance that can be achieved when using the rejection kernel of Del Moral (2004) in the particle filter, can be realised for finite N in a number of scenarios. We now report briefly on any improvements in estimator accuracy that the RSMC methodology can offer.

The median filtering errors were calculated for the estimates provided by the

	N = 100	N = 400	N = 900	N = 1600	N = 2500
d = 1	0.5781 0.5604 0.6049	0.3347 0.2965 0.3004	0.2388 0.2095 0.1928	0.1735 0.1508 0.1523	0.1455 0.1231 0.1287
d = 2	0.7646 0.7519 0.7449	0.4036 0.3696 0.3813	0.2767 0.2528 0.2577	0.2097 0.1891 0.1907	0.1619 0.1522 0.1549
d = 5	0.7256 0.6823 0.6735	0.3632 0.3305 0.3481	0.2477 0.2233 0.2239	0.1848 0.1705 0.1726	0.1488 0.1357 0.1318
d = 10	1.4397 1.4576 1.2745	0.4069 0.3867 0.3871	0.2764 0.2464 0.2613	0.2050 0.1956 0.1942	0.1636 0.1496 0.1539

Table 3.14: Median standard errors of the MPDABC filtering estimates. The values in each cell correspond to the estimates produced by the SMC (top), ERSMC (middle) and TRSMC (bottom) particle filters, targeting the MPDABC approximation of the nonlinear SSM (3.13)-(3.14) specified by $\sigma^2 = 5$.

ERSMC and TRSMC particle filters. The magnitude of these errors was subsequently compared to the median filtering errors obtained using the SMC particle filter, and the relative magnitudes are reported in Tables 3.15 and 3.16. Similar results, corresponding to the implementation of the RSMC and SMC particle filters under the ABC and MPDABC approximations of the model, are presented in Tables 3.17-3.19.

From Tables 3.15-3.19, we note that the accuracy of the RSMC filtering estimates is extremely comparable to the accuracy of the SMC particle filtering estimates. It is also noted that, in some scenarios, the RSMC particle filters offer marginally improved accuracy over the SMC particle filter. We consider the exact particle filters first.

From Tables 3.15 and 3.16, we see that, when targeting the one-dimensional hidden state, both the ERSMC and TRSMC procedures offer a consistent improvement in accuracy. It is also observed that as the dimension increases, any improvement

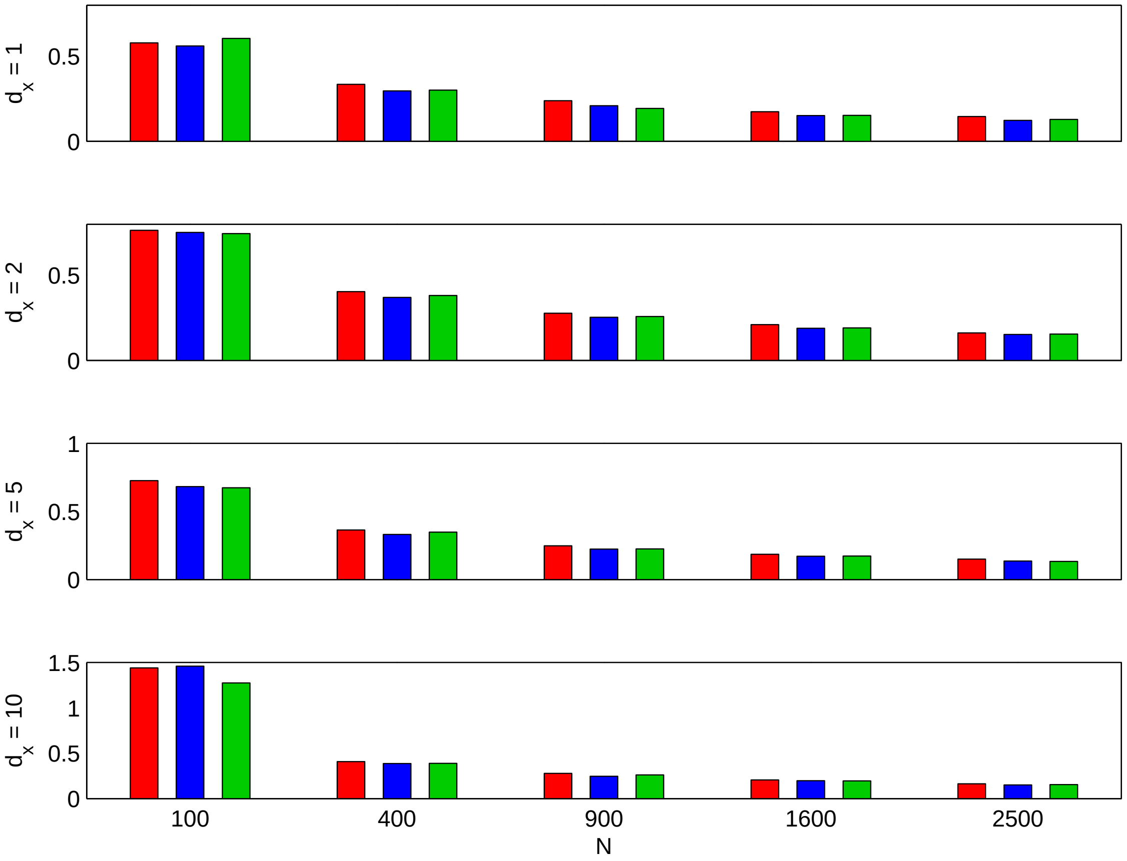


Figure 3.10: Median standard errors of the MPDABC SMC and RSMC particle filtering estimates of the filtered state in the nonlinear SSM (3.13)-(3.14), with $\sigma^2 = 5$. Each plot illustrates the standard errors corresponding to the estimates produced by the MPDABC particle filters using SMC (red), ERSMC (blue) and TRSMC (green) kernels.

in estimator accuracy offered by the use of RSMC, diminishes. Even in high dimensions, however, the accuracy of the RSMC particle filter remains extremely competitive when compared against the SMC particle filter.

A more consistent improvement in estimator accuracy is noted in Tables 3.17-3.19 and for the MPDABC filters, the use of RSMC improved the accuracy of the resulting estimates for all parameters tested. It is also remarked here that any improvement in estimator accuracy that is observed when incorporating the RSMC methodology, diminishes as the dimension of the hidden state increases.

	N = 100	N = 400	N = 900	N = 1600	N = 2500
d = 1	93.91%	96.30%	90.05%	90.31%	94.95%
d = 2	103.34%	98.18%	100.53%	98.28%	96.73%
d = 5	109.22%	94.47%	100.50%	102.61%	98.29%
d = 10	99.18%	99.27%	101.33%	98.35%	99.17%

Table 3.15: Relative magnitude of the median dimension-adjusted $\mathbb{L}_1$-errors obtained under ERSMC particle filtering, in comparison to those obtained under SMC particle filtering, in application to the exact nonlinear SSM (3.13)-(3.14).

	N = 100	N = 400	N = 900	N = 1600	N = 2500
d = 1	95.38%	93.56%	91.96%	92.75%	94.96%
d = 2	96.88%	98.90%	98.86%	99.22%	94.66%
d = 5	102.30%	100.17%	100.78%	97.58%	98.23%
d = 10	98.55%	100.10%	98.65%	97.13%	100.57%

Table 3.16: Relative magnitude of the median dimension-adjusted $\mathbb{L}_1$-errors obtained under TRSMC particle filtering, in comparison to those obtained under SMC particle filtering, in application to the exact nonlinear SSM (3.13)-(3.14).

3.7 Summary of Results

The numerical results presented in this chapter have demonstrated the properties of the ABC particle filter through application to a linear Gaussian SSM and a common nonlinear SSM from the SMC literature. We have also considered the use of the RSMC methodology, discussed in Section 2.3.2. We present here the principal conclusions of this chapter. In particular, the following points have been demonstrated:

- As the dimension of the hidden state increases, the (numerical) SMC error associated with the exact particle filtering estimate explodes. In contrast, the SMC error associated with the ABC particle filtering estimate is stable.

- The total error of the ABC particle filtering estimate was observed here to be dominated by the deterministic bias of the ABC approximation.

	N = 100	N = 400	N = 900	N = 1600	N = 2500
d = 1	98.24%	97.70%	98.52%	97.37%	97.30%
d = 2	98.06%	98.34%	98.88%	97.81%	100.14%
d = 5	97.66%	99.05%	99.58%	99.11%	99.53%
d = 10	-	99.37%	100.58%	99.60%	100.29%

Table 3.17: Relative magnitude of the median dimension-adjusted $\mathbb{L}_1$-errors obtained under RSMC particle filtering, in comparison to those obtained under SMC particle filtering, in application to the ABC approximation of the nonlinear SSM (3.13)-(3.14).

	N = 100	N = 400	N = 900	N = 1600	N = 2500
d = 1	97.01%	92.71%	95.40%	95.20%	98.05%
d = 2	98.59%	98.01%	97.63%	97.43%	98.48%
d = 5	99.22%	98.81%	99.35%	99.13%	99.29%
d = 10	-	98.73%	99.51%	99.86%	99.68%

Table 3.18: Relative magnitude of the median dimension-adjusted $\mathbb{L}_1$-errors obtained under ERSMC particle filtering, in comparison to those obtained under SMC particle filtering, in application to the MPDABC approximation of the nonlinear SSM (3.13)-(3.14).

	N = 100	N = 400	N = 900	N = 1600	N = 2500
d = 1	99.81%	94.99%	97.63%	95.99%	97.12%
d = 2	98.03%	98.12%	99.07%	97.72%	98.07%
d = 5	98.96%	99.30%	99.61%	99.49%	99.13%
d = 10	97.99%	99.07%	99.67%	99.97%	99.98%

Table 3.19: Relative magnitude of the median dimension-adjusted $\mathbb{L}_1$-errors obtained under TRSMC particle filtering, in comparison to those obtained under SMC particle filtering, in application to the MPDABC approximation of the nonlinear SSM (3.13)-(3.14).

- In low dimensions, the dominance of the ABC bias results in the ABC particle filter being significantly less accurate than exact particle filter. In high dimensions, however, the ABC particle filter was shown to be more accurate than the exact particle filter, due to the SMC error associated with the exact particle filtering estimates exceeding the combined bias and numerical error associated with the ABC particle filtering estimates.

- The MPDABC particle filtering estimator is also dominated by the ABC bias, and was observed to offer no significant improvements over the ABC particle filtering estimator.

- The use of an RSMC procedure, in place of a dynamic resampling SMC procedure, was shown to reduce the variance of the ABC and MPDABC particle filtering estimators, especially when using large N. Thus, when performing ABC particle filtering, the use of a RSMC procedure is advised.

- The exact RSMC particle filters were shown not to offer consistent improvements in estimator variance, when compared to exact particle filters that used dynamic resampling. We conclude that, when performing exact particle filtering with respect to the nonlinear SSM (3.13)-(3.14), the reduction in asymptotic estimator variance suggested by Del Moral (2004) cannot be observed for $N \leq 2500$.

Chapter 4

Smoothing via Approximate Bayesian Computation

4.1 Introduction

Consider the generalised state space model defined by the transition and observation densities (1.1)-(1.2). In the previous chapter, we considered the filtering problem: performing inference on the distribution of the unobserved Markov chain at time t, given observation of the process up to t. This chapter will extend the work of Chapter 3 by considering the use of SMC and PMCMC methods for performing inference with respect to the ABC approximation of the generalised SSM.

In Section 4.2, we detail the inferential problem of interest. In Section 4.3, a brief review of existing SMC smoothing procedures is provided, where we establish the state of the art for performing sequential inference. In Section 4.4, we provide the main theoretical result of the chapter, which establishes the behaviour of a class of expectations, taken with respect to the ABC approximation of the joint smoothing density. This theoretical result motivates the development of SMC and PMCMC procedures in an ABC framework, and this takes place in Sections 4.5 and 4.6. In Section 4.7, we test the performance of the proposed methods by applying them to the popular nonlinear SSM from Section 3.6.

4.2 The Smoothing Problem

Interest lies in performing inference on the joint density of the unobserved Markov chain $X_{1:t} \in \mathbb{R}^{td_x}$ given observation of the process $Y_{1:t} \in \mathbb{R}^{td_y}$, and we retain the assumption that x_0 is known.

When working with the smoothing problem, one is interested in performing either batch or sequential inference. In the former scenario, the analyst has access to the entire observation record, say $y_{1:T}$, prior to analysis, and can therefore make use of advanced MCMC methods, such as PMCMC, to directly target the distribution of interest, $\pi\left(dx_{1:T} \left| y_{1:T}\right.\right)$. In the latter scenario, the data arrives sequentially and one must target the sequence of smoothing distributions $\{\pi\left(dx_{1:t} \left| y_{1:t}\right.\right)\}_{t=1}^{T}$ sequentially. Our eventual aim will be to implement PMCMC methods for an ABC approximation of the target smoothing distribution, and so batch inference will be required, however the development of SMC based smoothing algorithms for an ABC setting will be the initial focus of this chapter; these can be used both for sequential inference and as the building blocks for batch inference methods.

The parameter vector θ will be treated as known for the development of SMC smoothing for an ABC target. However, as was seen in Section 2.4, parameter uncertainty can be tackled using PMCMC methodology, and so it is reintroduced into the notation in this chapter. Recall the joint smoothing density, given in (1.7) and reproduced here, with parameter vector $\theta \in \Theta$:

$$\pi_\theta\left(x_{1:t} \mid y_{1:t}\right) = \frac{\pi_\theta\left(x_{1:t}, y_{1:t}\right)}{\pi_\theta\left(y_{1:t}\right)} = \frac{\prod_{n=1}^{t} g_{n,\theta}\left(y_n \left| x_n\right.\right) q_{n,\theta}\left(x_n \left| x_{n-1}\right.\right)}{\int_{\mathbb{R}^{td_x}} \prod_{n=1}^{t} g_{n,\theta}\left(y_n \left| x_n\right.\right) q_{n,\theta}\left(x_n \left| x_{n-1}\right.\right) dx_{1:t}}. \tag{4.1}$$

As in Chapter 3, the problem of interest will be framed in terms of the evaluation of expectations with respect to the target distribution. We consider here the problem of computing expectations of additive functionals with respect to the joint smoothing densities $\{\pi_\theta(x_{1:t}|y_{1:t})\}_{t=1}^{T}$:

$$\mathcal{S}_t = \mathbb{E}_{\pi_{1:t,\theta}}\left[S_t(x_{1:t})|y_{1:t}\right] = \int_{\mathbb{R}^{td_x}} S_t(x_{1:t})\,\pi_\theta(dx_{1:t} \mid y_{1:t}) \tag{4.2}$$

where

$$S_t(x_{1:t}) = \sum_{n=1}^{t} s_n(x_{n-1}, x_n) \tag{4.3}$$

with $s_n : \mathbb{R}^{2d_x} \to \mathbb{R}^{d_x}$, $n \geq 1$.

Examples include, for $n = 1, \ldots, t$:

$$\begin{aligned} s_{n,j}(x_{n,j}) &= \frac{x_{n,j}}{t} \\ s_{n,j}(x_{n-1,j}, x_{n,j}) &= \frac{x_{n-1,j}x_{n,j}}{t} \end{aligned} \tag{4.4}$$

where $x_n = (x_{n,1}, \ldots, x_{n,d_x})$ and $s_n = (s_{n,1}, \ldots, s_{n,d_x})$, with $s_{n,j} : \mathbb{R} \to \mathbb{R}$ for $j = 1, \ldots, d_x$. These specifications of s_n can be used to construct estimates of $\mathcal{S}_t$ corresponding to the mean and first-order autocovariance, respectively.

As highlighted by Del Moral et al. (2009), the construction and estimation of additive functionals in this way can also be extremely useful in parameter estimation work. Consider the parameter vector θ to be unknown. Given the observed data up to time t, $y_{1:t}$, the score vector $\nabla \log \pi_\theta(y_{1:t})$ may be expressed as a composition of expectations of a similar form to (4.2) (Del Moral et al., 2009):

$$\nabla \log \pi_\theta(y_{1:t}) = \sum_{n=1}^{t} \mathbb{E}_{\pi_{1:t,\theta}}\left[\nabla \log q_{n,\theta}(X_n|X_{n-1}) \mid y_{1:t}\right] + \sum_{n=1}^{t} \mathbb{E}_{\pi_{1:t,\theta}}\left[\nabla \log g_{n,\theta}(y_n|X_n) \mid y_{1:t}\right].$$

As the first derivative of the log-likelihood, the score vector has direct applications in maximum likelihood (ML) estimation. A common iterative method for ML estimation is the Expectation-Maximization (EM) algorithm. This is a two-step procedure, involving a similar decomposition of the score vector. In the batch smoothing setup, given a current estimate θ of the parameter vector, the algorithm proceeds by first calculating the expected value of the log-likelihood function, where the expectation is with respect to the joint smoothing density $\pi_\theta\left(x_{1:t}\left|y_{1:t}\right.\right)$:

$$Q\left(\theta,\theta^*\right)=\sum_{n=1}^{t}\mathbb{E}_{\pi_{1:t,\theta}}\left[\,\log q_{n,\theta^*}\left(X_n\left|X_{n-1}\right.\right)\mid y_{1:t}\right]+\sum_{n=1}^{t}\mathbb{E}_{\pi_{1:t,\theta}}\left[\,\log g_{n,\theta^*}\left(y_n\left|X_n\right.\right)\mid y_{1:t}\right].$$

This function is subsequently maximized with respect to θ^*, and the parameter vector estimate is updated to the maximising argument. As with estimating the score vector, the usefulness of the additive functionals in computing $Q\left(\theta,\theta^*\right)$ in the Expectation step above is apparent.

4.3 Existing Approaches to Smoothing

We will consider here a number of SMC-based smoothing procedures that exist within the literature. The aim will be to offer alterations to the existing methods, first through the application of approximate Bayesian computation (ABC) and second through the use of Rejection SMC (RSMC).

The particle smoothing procedure, introduced in Section 2.3.1, is briefly described below in the context of the current estimation problem. As discussed previously, this suffers heavily from the path degeneracy problem, and the resulting estimates will be expected to display poor accuracy. Two further existing SMC smoothing algorithms are described below: Forward Filtering Backward Smoothing (FFBS; Doucet et al., 2000; Godsill et al., 2004) and Forward Smoothing (FS; Del Moral et al., 2009). Note that these are two implementations of the same procedure, where FS exploits the additive structure of the targeted expectations (4.2), and can be implemented for online sequential inference.

4.3.1 Particle Smoothing

As we have seen in Section 2.3.1, particle smoothing is achieved through recursively sampling the N particles $\left\{X_{1:t}^{(i)}\right\}_{i=1}^{N}$ at time t from the sequence of joint smoothing distributions $\pi_\theta\left(dx_{1:t}\left|y_{1:t}\right.\right)$. The particles are propagated forward using a combination of importance sampling and resampling steps, with the importance weights being calculated at each time step according to the recursive relationship (2.15). Recall that the smoothing distribution is approximated via the Monte Carlo estimate in (2.13), leading to the following particle smoothing estimate of the expectation of interest

$$\hat{\mathcal{S}}_t = \sum_{i=1}^{N} W_t^{(i)} S_t\left(X_{1:t}^{(i)}\right). \tag{4.5}$$

with $W_t^{(i)}$ the normalized nonnegative importance weights calculated at time t, the most recent iteration of the SMC algorithm. The full particle smoothing procedure is detailed in Algorithm 8; this can be implemented with computational cost $O(N)$.

As mentioned above, this approximation of the smoothing distribution suffers from the path degeneracy problem, which is avoided by both FFBS and FS, although at the cost of an increased computational complexity of $O(N^2)$.

We consider also the asymptotic variance of the particle smoothing estimate of the smoothed additive functional. It is shown by Poyiadjis et al. (2011), for additive functionals of the slightly restricted form

$$S_t\left(x_{1:t}\right) = \sum_{n=0}^{t} s_n\left(x_n\right), \tag{4.6}$$

and under favourable assumptions, that the asymptotic variance (as $N \to \infty$) of $\sqrt{N}\left(\hat{\mathcal{S}}_t - \mathcal{S}_t\right)$ is bounded below by a quadratic polynomial in the length of the observation period, t, i.e. the asymptotic variance of the particle smoothing estimate (4.5) is at least $O(t^2)$.

Data: $y_{1:t}$;
Parameters: x_0, N;
Result: An SMC estimate of the expected value of S_t (4.3), an additive functional, obtained using particle smoothing.

1. Set $t = 1$. For $i = 1, \ldots, N$, sample $X_t^{(i)} \sim K_t\left(x_0, dx_t\right)$, and compute

$$W_t^{(i)} \propto \frac{g_t\left(y_t \left| X_t^{(i)}\right.\right) q_t\left(X_t^{(i)} | x_0\right)}{k_t\left(X_t^{(i)} | x_0\right)}, \quad \sum_{i=1}^{N} W_t^{(i)} = 1.$$

2. If $\widehat{ESS}\left(\left\{W_t^{(i)}\right\}_{i=1}^{N}\right) < \frac{N}{2}$, then, for $i = 1, \ldots, N$, resample $X_t^{(i)}$ independently from the discrete distribution

$$\hat{\pi}\left(dx_t\right) = \sum_{i=1}^{N} W_t^{(i)} \delta_{X_{1:t}^{(i)}}\left(dx_t\right)$$

and set $W_t^{(i)} = \frac{1}{N}$.

3. Set $t = t + 1$. For $i = 1, \ldots, N$, sample $X_t^{(i)} \sim K_t(X_{t-1}^{(i)}, dx_t)$ and calculate the filtering weights

$$W_t^{(i)} \propto W_{t-1}^{(i)} \cdot \frac{g_t\left(y_t \left| X_t^{(i)}\right.\right) q_t\left(X_t^{(i)} \middle| X_{t-1}^{(i)}\right)}{k_t\left(X_t^{(i)} \middle| X_{t-1}^{(i)}\right)}, \quad \sum_{i=1}^{N} W_t^{(i)} = 1.$$

4. If $n < t$, return to Step 2, else calculate $\hat{\mathcal{S}}_t$ according to (4.5) and stop.

Algorithm 8: An SMC Algorithm for Particle Smoothing

4.3.2 Forward Filtering Backward Smoothing

Consider the expectation of interest in Equation (4.2). This can be rewritten as

$$\mathcal{S}_t = \sum_{n=1}^{t} \mathbb{E}_{\pi_{n-1:n,t,\theta}}\left[s_n\left(x_{n-1}, x_n\right) | y_{1:t}\right] = \sum_{n=1}^{t} \int_{\mathbb{R}^{2d_x}} s_n\left(x_{n-1}, x_n\right) \pi_\theta\left(dx_{n-1:n} \mid y_{1:t}\right).$$

This motivates the use of an SMC procedure to provide weighted samples from each marginal smoothing distribution $\pi_\theta\left(dx_n \mid y_{1:t}\right)$, for $n = 1, \ldots, t-1$, from which estimates of $\mathcal{S}_t$ can be calculated. Forward Filtering Backward Smoothing does exactly this, by appending a backward pass to a particle filtering algorithm at time t, which recursively calculates a series of weights $\left\{\widetilde{W}_{n|t}^{(i)}\right\}_{i=1}^{N}$, $n = t-1, \ldots, 1$, for approximating the marginal smoothing densities with the particle approximation

$$\hat{\pi}_\theta\left(dx_n \mid y_{1:t}\right) = \sum_{i=1}^{N} \widetilde{W}_{n|t}^{(i)} \delta_{X_n^{(i)}}\left(dx_n\right). \tag{4.7}$$

Note that when $n = t$, the above coincides with the particle approximation for the filtering distribution, and so we define $\widetilde{W}_{t|t}^{(i)} = W_t^{(i)}$. For clarity, we henceforth refer to the importance weights $\left\{W_{1:t}^{(i)}\right\}_{i=1}^{N}$ used in the forward pass of the algorithm as the filtering weights; the weights calculated in the backward pass of the FFBS procedure will be referred to as the smoothing weights.

The backward pass in the FFBS procedure makes use of the smoothing formula (see, e.g. Doucet et al., 2000; Godsill et al., 2004). Before stating the smoothing formula, we introduce the backward kernel representation of a density; this notation will prove useful in exploring the bias associated with the ABC approximation of the smoothed additive functional in Section 4.4. For a generic probability density p on $\mathbb{R}^{d_x}$, define the backward Markov transition density as

$$B_p(x_n|x_{n+1}) := \frac{p(x_n)q_{n+1,\theta}(x_{n+1}|x_n)}{\int_{\mathbb{R}^{d_x}} p(x_n)q_{n+1,\theta}(x_{n+1}|x_n)dx_n}, \tag{4.8}$$

under the assumption that $\int_{\mathbb{R}^{d_x}} p(x_n)q_{n+1,\theta}(x_{n+1}|x_n)dx_n < +\infty$. Now, the smoothing formula states that, for $n = 1, \ldots, t$,

$$\pi_\theta\left(x_n \mid y_{1:t}\right) = \int_{\mathbb{R}^{d_x}} \pi_\theta\left(x_{n+1} \mid y_{1:t}\right) B_{\pi_{n,\theta}}(x_n|x_{n+1})dx_{n+1} \tag{4.9}$$

Replacing $\pi_\theta\left(x_{n+1} \mid y_{1:t}\right)$ and $\pi_\theta\left(x_n \mid y_{1:n}\right)$ with their particle approximations gives the following approximation of the marginal smoothed distribution:

$$\hat{\pi}_\theta\left(dx_n \mid y_{1:t}\right) = \sum_{i=1}^{N} W_n^{(i)} \sum_{j=1}^{N} \widetilde{W}_{n+1|t}^{(j)} \, \frac{q_{n+1,\theta}\left(X_{n+1}^{(j)} \,\middle|\, X_n^{(i)}\right)}{\pi_\theta\left(X_{n+1}^{(j)} \mid y_{1:n}\right)} \, \delta_{X_n^{(i)}}\left(dx_n\right). \tag{4.10}$$

Noting that $\pi_\theta\left(x_{n+1} \mid y_{1:n}\right) = \int_{\mathbb{R}^{d_x}} \pi_\theta\left(x_n \mid y_{1:n}\right) q_{n+1,\theta}\left(x_{n+1} \left| x_n\right.\right) dx_n$, we can further approximate $\pi_\theta\left(X_{n+1}^{(j)} \middle| \, y_{1:n}\right)$ and substitute this approximation into the above to give

$$\hat{\pi}_\theta\left(dx_n \mid y_{1:t}\right) = \sum_{i=1}^{N} W_n^{(i)} \sum_{j=1}^{N} \widetilde{W}_{n+1|t}^{(j)} \, \frac{q_{n+1,\theta}\left(X_{n+1}^{(j)} \,\middle|\, X_n^{(i)}\right)}{\left[\sum_{l=1}^{N} W_n^{(l)} q_{n+1,\theta}\left(X_{n+1}^{(j)} \,\middle|\, X_n^{(l)}\right)\right]} \, \delta_{X_n^{(i)}}\left(dx_n\right).$$

Thus, the backward pass of the FFBS algorithm recursively calculates the smoothing weights $\left\{\widetilde{W}_{n|t}^{(i)}\right\}_{i=1}^{N}$ associated with the particles $\left\{X_n^{(i)}\right\}_{i=1}^{N}$ as

$$\widetilde{W}_{n|t}^{(i)} = W_n^{(i)} \sum_{j=1}^{N} \widetilde{W}_{n+1|t}^{(j)} \, \frac{q_{n+1,\theta}\left(X_{n+1}^{(j)} \,\middle|\, X_n^{(i)}\right)}{\left[\sum_{l=1}^{N} W_n^{(l)} q_{n+1,\theta}\left(X_{n+1}^{(j)} \,\middle|\, X_n^{(l)}\right)\right]} \tag{4.11}$$

for $n = t-1, \ldots, 1$ with $\widetilde{W}_{t|t}^{(i)} = W_t^{(i)}$.

Upon completion of this backwards pass, $\left\{X_n^{(i)}, \widetilde{W}_{n|t}^{(i)}\right\}_{i=1}^{N}$ will provide a weighted sample from the marginal smoothing distribution $\pi_\theta\left(x_n \left| y_{1:t}\right.\right)$, where $\left\{X_n^{(i)}\right\}_{i=1}^{N}$ is the sample generated at time n of the forward (filtering) pass, prior to resampling. The requirement that these are the non-resampled particles is crucial, however this requires the resulting FFBS algorithm to store, at each time $n = 1, \ldots, t$, the entire particle system (particles and filtering weights) $\left\{X_n^{(i)}, W_n^{(i)}\right\}_{i=1}^{N}$. This is a significant memory requirement and, in addition, the computational complexity of an FFBS algorithm can be prohibitive. Assuming that one is interested in estimating $\mathcal{S}_t$ at all time points $t = 1, \ldots, T$, one must execute the backward pass of the FFBS algorithm at each time t, with computational complexity $O(N^2 t)$. Clearly, the computational budget for each iteration of this procedure increases with the time parameter, making it an offline procedure. In this scenario, the FFBS procedure has computational budget $O\left(N^2T^2\right)$. If one is interested in estimating $\mathcal{S}_T$ only, as is often the case for batch smoothing, one need only execute the backward pass at the final iteration, and so the computational complexity of the FFBS procedure in this scenario is $O(N^2T)$, with T the batch size.

Godsill et al. (2004) provide a procedure for drawing independent samples from the required smoothed distribution, where, subsequent to performing SMC filtering via a forward pass, a backward pass is executed as follows:

```
Data: $y_{1:t}$, $\left\{X_{1:t}^{(i)}, W_t^{(i)}\right\}_{i=1}^{N}$ (an $N$-particle system obtained using an SMC
      filtering procedure);
Parameters: $t$, $N$;
Result: An independent $t$-length sample $\widetilde{X}_{1:t}$ from the joint smoothing
        distribution with density (4.1).

  1. Sample $\widetilde{X}_t$ from $\hat{\pi}_\theta\left(d\tilde{x}_t \left| y_{1:t}\right.\right) = \sum_{i=1}^{N} W_t^{(i)} \delta_{X_t^{(i)}}\left(d\tilde{x}_t\right)$
  2. Set $n = n - 1$.
     For $i = 1, \ldots, N$, calculate
        $\widetilde{W}_{n|n+1}^{(i)} \propto W_n^{(i)} q_{n+1,\theta}\left(\widetilde{X}_{n+1} \left| X_n^{(i)}\right.\right) \; ; \; \sum_{i=1}^{N} \widetilde{W}_{n|n+1}^{(i)} = 1$
  3. Sample $\widetilde{X}_n$ from $\sum_{i=1}^{N} \widetilde{W}_{n|n+1}^{(i)} \delta_{X_n^{(i)}}\left(d\tilde{x}_n\right)$. If $n = 1$ stop, otherwise return to 2.
```

Algorithm 9: Backward Pass of a FFBS backward-sampling procedure

Although the complexity of the pass detailed above is $O(Nt)$, its execution must be repeated for each independent sample required. Thus, in real terms, the backward sampler offers no computational savings over the FFBS procedure of Doucet et al. (2000).

4.3.3 Forward Smoothing

Del Moral et al. (2009) introduce an online procedure for calculating the FFBS estimate of expectations of the form (4.2). Referred to henceforth as Forward Smoothing (FS), this eliminates the need for a backward pass in the smoothing procedure by introducing an auxiliary function. Although the resulting algorithm retains a significant computational complexity of $O(N^2)$ per time step, this is constant with respect to the time parameter, allowing the possibility of online implementation for SMC-based smoothing.

Del Moral et al. (2009) introduce the following auxiliary function:

$$R_t\left(x_t\right) = \int_{\mathbb{R}^{(t-1)d_x}} S_t\left(x_{1:t}\right) \pi_\theta\left(x_{1:t-1} \mid y_{1:t-1}, x_t\right) dx_{1:t-1},$$

such that the expectation of interest at time t may be written in the form

$$\mathcal{S}_t = \int_{\mathbb{R}^{d_x}} R_t\left(x_t\right) \pi_\theta\left(x_t \mid y_{1:t}\right) dx_t.$$

Since this is the expectation of the auxiliary function with respect to the filter at time t, $\hat{\mathcal{S}}_t$ can be calculated using the filtering weights $\left\{W_t^{(i)}\right\}_{i=1}^N$, given $R_t(x_t)$, or a suitable estimate. Del Moral et al. (2009) also provide a recursion for calculating $\{R_t\}_{t\geq 1}$ online, thus facilitating a forward-only smoothing procedure. The recursion and its derivation are provided here. For $t \geq 2$,

$$R_t\left(x_t\right) = \int_{\mathbb{R}^{d_x}} \left[R_{t-1}\left(x_{t-1}\right) + s_t\left(x_{t-1}, x_t\right)\right] \pi_\theta\left(x_{t-1} \mid y_{1:t-1}, x_t\right) dx_{t-1}, \tag{4.12}$$

where $R_1\left(x_1\right) := 0$. This follows from:

$$\begin{aligned} R_t\left(x_t\right) &:= \int_{\mathbb{R}^{(t-1)d_x}} \left[S_{t-1}\left(x_{1:t-1}\right) + s_t\left(x_{t-1}, x_t\right)\right] \pi_\theta\left(x_{1:t-1} \mid y_{1:t-1}, x_t\right) dx_{1:t-1} \\ &= \int_{\mathbb{R}^{d_x}} \left[\int_{\mathbb{R}^{(t-2)d_x}} S_{t-1}\left(x_{1:t-1}\right) \pi_\theta\left(x_{1:t-2} \mid y_{1:t-1}, x_{t-1}\right) dx_{1:t-2}\right] \pi_\theta\left(x_{t-1} \mid y_{1:t-1}, x_t\right) dx_{t-1} \\ &\quad + \int_{\mathbb{R}^{d_x}} s_t\left(x_{t-1}, x_t\right) \pi_\theta\left(x_{t-1} \mid y_{1:t-1}, x_t\right) dx_{t-1}. \end{aligned}$$

From (4.12), one can construct a method for generating an estimate of $R_t(x_t)$ given the samples $\left\{X_{t-1}^{(i)}\right\}_{i=1}^{N}$ and $\left\{X_{t}^{(i)}\right\}_{i=1}^{N}$ and the filtering weights $\left\{W_{t-1}^{(i)}\right\}_{i=1}^{N}$. Noting first that

$$\pi_\theta\left(x_{t-1} \mid y_{1:t-1}, x_t\right) = \frac{q_{t,\theta}\left(x_t \left| x_{t-1}\right.\right) \pi_\theta\left(x_{t-1} \mid y_{1:t-1}\right)}{\int_{\mathbb{R}^{d_x}} q_{t,\theta}\left(x_t \left| x_{t-1}\right.\right) \pi_\theta\left(x_{t-1} \mid y_{1:t-1}\right) dx_{t-1}},$$

the progression to

$$\hat{R}_t\left(X_t^{(i)}\right) = \frac{\sum_{j=1}^{N} W_{t-1}^{(j)} q_{t,\theta}\left(X_t^{(j)} \left| X_{t-1}^{(i)}\right.\right)\left[\hat{R}_{t-1}\left(X_{t-1}^{(i)}\right) + s_t\left(X_{t-1}^{(j)}, X_t^{(i)}\right)\right]}{\sum_{j=1}^{N} W_{t-1}^{(j)} q_{t,\theta}\left(X_t^{(j)} \left| X_{t-1}^{(i)}\right.\right)} \tag{4.13}$$

for $i = 1, \ldots, N$, is straightforward. The Forward Smoothing estimate of $\mathcal{S}_t$ may then be calculated as

$$\hat{\mathcal{S}}_t = \sum_{i=1}^{N} W_t^{(i)} \hat{R}_t\left(X_t^{(i)}\right), \tag{4.14}$$

for $i = 1, \ldots, N$. The forward smoothing procedure is detailed in full in Algorithm 10.

An additional contribution of Del Moral et al. (2009) is the provision of an upper bound on the non-asymptotic mean square error of the estimate $\hat{\mathcal{S}}_t$ resulting from the application of the Forward Smoothing algorithm described above. The authors prove, once again for the restricted class of smoothed additive functionals specified by (4.6), that for any $t \geq 0$, and under the assumption that $\sup\limits_{n\geq 0} \|s_n\|_\infty < \infty$ for any $0 \leq n \leq t$, the following upper bound holds:

$$\mathbb{E}\left(\left|\hat{\mathcal{S}}_t - \mathcal{S}_t\right|^2\right) \leq a\frac{(t+1)}{N}\left(a + \sqrt{\frac{t+1}{N}}\right)^2,$$

where a is a finite constant, independent of time. The reader is directed to the aforementioned paper for details of the proof for this case and it should be noted that the authors also assert that the bound holds for the more general specification of additive functionals of the form in (4.3).

Data: $y_{1:T}$;
Parameters: x_0, T, N;
Result: An SMC estimate, at each time point $t = 1, \ldots, T$, of the expected value of S_t, an additive functional.

1. Set $t = 1$. For $i = 1, \ldots, N$, sample $X_t^{(i)} \sim K_t\left(x_0, dx_t\right)$, and calculate

$$\hat{R}_1\left(X_t^{(i)}\right) = \frac{\sum_{j=1}^{N} q_{t,\theta}\left(X_t^{(j)} | x_0\right) s_t\left(x_0, X_t^{(i)}\right)}{\sum_{j=1}^{N} q_{t,\theta}\left(X_t^{(j)} | x_0\right)}.$$

   For $i = 1, \ldots, N$, compute

$$W_t^{(i)} \propto \frac{g_{t,\theta}\left(y_t \middle| X_t^{(i)}\right) q_{t,\theta}\left(X_t^{(i)} | x_0\right)}{k_t\left(X_t^{(i)} | x_0\right)}, \qquad \sum_{i=1}^{N} W_t^{(i)} = 1.$$

2. If $\widehat{ESS}\left(\left\{W_t^{(i)}\right\}_{i=1}^{N}\right) < \frac{N}{2}$, then, for $i = 1, \ldots, N$, resample $X_t^{(i)}$ independently from the discrete distribution

$$\hat{\pi}\left(dx_t\right) = \sum_{i=1}^{N} W_t^{(i)} \delta_{X_{1:t}^{(i)}}\left(dx_t\right)$$

   and set $W_t^{(i)} = \frac{1}{N}$.

3. Set $t = t + 1$. For $i = 1, \ldots, N$, sample $X_t^{(i)} \sim K_t\left(X_{t-1}^{(i)}, dx_t\right)$ and calculate the filtering weights

$$W_t^{(i)} \propto W_{t-1}^{(i)} \cdot \frac{g_{t,\theta}\left(y_t \middle| X_t^{(i)}\right) q_{t,\theta}\left(X_t^{(i)} \middle| X_{t-1}^{(i)}\right)}{k_t\left(X_t^{(i)} \middle| X_{t-1}^{(i)}\right)}, \qquad \sum_{i=1}^{N} W_t^{(i)} = 1.$$

4. For $i = 1, \ldots, N$, calculate $\hat{R}_t\left(X_t^{(i)}\right)$ according to (4.13).

5. Calculate $\hat{\mathcal{S}}_t$ according to (4.14). If $t < T$, return to Step 2.

Algorithm 10: An SMC Algorithm for Forward Smoothing

Using a result from Del Moral et al. (2010), which itself establishes that the upper bound on the bias of the estimate $\hat{\mathcal{S}}_t$ is $O\left(N^{-1}\right)$, Del Moral et al. (2009) deduce that the asymptotic variance (as $N \to \infty$) of $\sqrt{N}\left(\hat{\mathcal{S}}_t - \mathcal{S}_t\right)$ has an upper bound of $O(t)$.

This result shows that estimates produced by Forward Smoothing (and FFBS) are significantly more accurate than those produced by particle smoothing, due to the $O(t^2)$ lower bound on the asymptotic variance of the particle smoothing estimate of the smoothed additive functional.

4.3.4 Smoothing via Rejection SMC

Recall the rejection-based SMC procedures detailed in Section 2.3.2. It was demonstrated in Algorithm 4 how to perform particle filtering via a Rejection SMC procedure, and the accuracy and algorithmic performance of RSMC filtering was explored in Chapter 3. We consider here the possibility of performing smoothing via an RSMC procedure. The use of RSMC will be considered for particle smoothing and forward smoothing, and their effects on the accuracy and algorithmic performance will be examined in Section 4.7.

As a precursor to describing the alteration of the particle smoothing and forward smoothing procedures to incorporate the rejection methodology, we stress the fact that the use of Rejection SMC in place of standard SMC requires a change only in the resampling step of the SMC procedure; the propagation and weighting of particles, as well as the estimation of quantities of interest, remains unchanged.

For performing particle smoothing via RSMC, it is noted that the difference between particle filtering and smoothing lies in the stage at which the particles and their weights are used to estimate the quantity of interest. In particular, the resampling step for these two procedures is the same, and so an RSMC particle smoothing procedure can be constructed by replacing the estimation step in an RSMC filtering algorithm (e.g. Step 5 in Algorithm 4, page 63), with Step 4 of the particle smoothing algorithm (Algorithm 8, page 129).

Since the use of RSMC affects only the resampling step of an SMC procedure, the rejection methodology of Section 2.3.2 can be incorporated into a forward smoothing procedure without disturbing the recursive calculation of the auxiliary function

Data: $y_{1:T}$;
Parameters: x_0, T, N;
Result: An SMC estimate, at each time point $t = 1, \ldots, T$, of the expected value of S_t, an additive functional.

1. Set $t = 1$. For $i = 1, \ldots, N$, sample $X_t^{(i)} \sim K_t\left(x_0, dx_t\right)$, and calculate

$$\hat{R}_1\left(X_t^{(i)}\right) = \frac{\sum_{j=1}^{N} q_{t,\theta}\left(X_t^{(j)} | x_0\right) s_t\left(x_0, X_t^{(i)}\right)}{\sum_{j=1}^{N} q_{t,\theta}\left(X_t^{(j)} | x_0\right)}.$$

For $i = 1, \ldots, N$, compute

$$W_t^{(i)} \propto \frac{g_{t,\theta}\left(y_t \left| X_t^{(i)}\right.\right) q_{t,\theta}\left(X_t^{(i)} | x_0\right)}{k_t\left(X_t^{(i)} | x_0\right)}, \quad \sum_{i=1}^{N} W_t^{(i)} = 1.$$

2. For $i = 1, \ldots, N$, calculate $\beta_t^{(i)}$ using (2.17), and with probability $\beta_t^{(i)}$ do not resample, otherwise resample $X_t^{(i)}$ according to the discrete distribution

$$\hat{\pi}_t\left(dx_{1:t}\right) = \sum_{i=1}^{N} W_t^{(i)} \delta_{X_{1:t}^{i}}\left(dx_{1:t}\right).$$

3. For $i = 1, \ldots, N$, set $W_t^{(i)} = \frac{1}{N}$.

4. Set $t = t + 1$. For $i = 1, \ldots, N$, sample $X_t^{(i)} \sim K_t\left(X_{t-1}^{(i)}, dx_t\right)$ and calculate the filtering weights

$$W_t^{(i)} \propto W_{t-1}^{(i)} \cdot \frac{g_{t,\theta}\left(y_t \left| X_t^{(i)}\right.\right) q_{t,\theta}\left(X_t^{(i)} \left| X_{t-1}^{(i)}\right.\right)}{k_t\left(X_t^{(i)} \left| X_{t-1}^{(i)}\right.\right)}, \quad \sum_{i=1}^{N} W_t^{(i)} = 1.$$

5. For $i = 1, \ldots, N$, calculate $\hat{R}_t\left(X_t^{(i)}\right)$ according to (4.13).

6. Calculate $\hat{\mathcal{S}}_t$ according to (4.14). If $t < T$, return to Step 2.

Algorithm 11: A Rejection SMC Forward Smoothing procedure

$R_t\left(x_t\right)$. We therefore introduce the RSMC forward smoothing algorithm in Algorithm 11.

Note that these RSMC smoothing procedures can be executed using the acceptance probability $\beta_t^{(i)}$, calculated as in (2.17), or its empirical alternative $\hat{\beta}_t^{(i)}$, calculated as in (2.19). Thus, we consider the four RSMC smoothing procedures: Theoretical and Empirical RSMC particle smoothing, and Theoretical and Empirical RSMC forward smoothing. The accuracy and algorithmic performance of these procedures are considered in Section 4.7.

4.4 Exploring the Theoretical Bias of ABC Smoothing

In this section, we consider an ABC approximation of the joint smoothing density (4.1). We seek to determine the nature of the bias induced by this ABC approximation, in the context of estimating smoothed additional functionals. The results presented in this section will motivate the development of existing SMC and PMCMC methods for use in an ABC framework, which follows in Sections 4.5 and 4.6.

The ABC approximation of (4.1) is

$$\pi_\theta^\epsilon \left(x_{1:t} \mid y_{1:t} \right) = \int_{\mathbb{R}^{d_y}} \pi_\theta^\epsilon \left(x_{1:t}, u_{1:t} \mid y_{1:t} \right) du_{1:t} \tag{4.15}$$

with the auxiliary joint distribution

$$\pi_\theta^\epsilon \left(x_{1:t}, u_{1:t} \mid y_{1:t} \right) = \frac{\prod_{n=1}^t G_n^\epsilon \left(u_n, y_n \right) g_{n,\theta} \left(u_n \left| x_n \right. \right) q_{n,\theta} \left(x_n \left| x_{n-1} \right. \right)}{\int_{\mathbb{R}^{t(d_x+d_y)}} \prod_{n=1}^t G_n^\epsilon \left(u_n, y_n \right) g_{n,\theta} \left(u_n \left| x_n \right. \right) q_{n,\theta} \left(x_n \left| x_{n-1} \right. \right) dx_{1:t} du_{1:t}}. \tag{4.16}$$

where $u_{1:t} \in \mathbb{R}^{td_y}$ are auxiliary data and $G_n^\epsilon \left(u_n, y_n \right)$ is the ABC kernel, used to measure the proximity of the auxiliary data to the observed data. The problem of interest becomes the estimation of the ABC smoothed additive functional

$$\mathcal{S}_t^\epsilon = \mathbb{E}_{\pi_{1:t,\theta}^\epsilon} \left[S_t \left(x_{1:t} \right) \left| y_{1:t} \right. \right].$$

We define the bias at time t as

$$\mathbb{B}_s(t,\epsilon) := \left| \mathbb{E}_{\pi_{1:t,\theta}}\left[\sum_{n=1}^{t} s_n(x_{n-1},x_n) \middle| y_{1:t} \right] - \mathbb{E}_{\pi^{\epsilon}_{1:t,\theta}}\left[\sum_{n=1}^{t} s_n(x_{n-1},x_n) \middle| y_{1:t} \right] \right|, \tag{4.17}$$

with $\mathbb{E}_{\pi_{1:t,\theta}}$ the expectation wrt the true joint smoothing density (4.1) and $\mathbb{E}_{\pi^{\epsilon}_{1:t,\theta}}$ the expectation wrt the ABC approximation (4.15). It is found that this bias is bounded, and that the bound increases only linearly in the time parameter.

Before introducing this bound and its derivation, a number of assumptions are made, and other notation is defined. We assume Lipschitz continuity of the likelihood, as in (A2); furthermore, as in Proposition 3.3.2, we assume that, for $\{L_t\}_{t\geq 1}$ defined as in (A2), $L_t \leq L < \infty$ for all $t \geq 1$. We also specify the ABC statistic and metric as in (A3) and make the following assumption, which is a little stronger than (A4):

(**A5**) There exist probability densities κ_1 on $\mathbb{R}^{d_x}$ and κ_2 on $\mathbb{R}^{d_y}$, as well as constants $0 < \lambda_1, \lambda_2 < \infty$ such that for all $t \geq 1$, and for all $(x_t, x_{t+1}) \in \mathbb{R}^{2d_x}$, $y_t \in \mathbb{R}^{d_y}$ and $\theta \in \Theta$,

$$\begin{aligned} \frac{1}{\lambda_1}\,\kappa_1(x_{t+1}) \;&\leq\; q_{t+1,\theta}(x_{t+1}|x_t) \;\leq\; \lambda_1\,\kappa_1(x_{t+1}), \\ \frac{1}{\lambda_2}\,\kappa_2(y_t) \;&\leq\; g_{t,\theta}(y_t|x_t) \;\leq\; \lambda_2\,\kappa_2(y_t), \end{aligned}$$

with

$$0 < \underline{\kappa}_1 < \kappa_1(x) < \overline{\kappa}_1 < \infty \quad \forall x \in \mathbb{R}^{d_x}$$
$$0 < \underline{\kappa}_2 < \kappa_2(y) < \overline{\kappa}_2 < \infty \quad \forall y \in \mathbb{R}^{d_y}.$$

Below we write $\mathcal{B}_b(E)$ to denote the set of all bounded measurable functions

$\varphi : E \to \mathbb{R}$ on a state-space E. We let $\|\varphi\|_\infty = \sup_{x \in E} |\varphi(x)|$ for $\varphi \in \mathcal{B}_b$. Also,

$$\beta(p) := \sup_{x,x' \in E} \sup_{A \in \mathcal{E}} \left| \int_A p(z|x)dz - \int_A p(z|x')dz \right|$$

is the Dobrushin coefficient for a Markov transition density p defined upon the measurable space $(E, \mathcal{E})$. Initially, we assume also that ϵ is constant with respect to the time parameter.

Proposition 4.4.1. *Assume (A2,A3,A5), and that $s_n \in \mathcal{B}_b(\mathbb{R}^{d_x} \times \mathbb{R}^{d_x})$. Then there exists a constant $C < \infty$ such that for any $\epsilon > 0$, $y_{1:t} \in \mathbb{R}^{t\, d_y}$, $t \geq 1$*

$$\mathbb{B}_s(t, \epsilon) \leq C\epsilon t$$

where $\mathbb{B}_s(t, \epsilon)$ is defined in (4.17).

Proof. To simplify notation in the subscripts, reference to the parameter vector θ is suppressed throughout the proof. We use a backward kernel representation of the smoothing density, making use of the notation introduced in (4.8). For any SSM one has that:

$$\pi(x_{n:t}|y_{1:t}) = \pi(x_t|y_{1:t}) \prod_{k=n}^{t-1} B_{\pi_k}(x_k|x_{k+1})$$

where π_k is the filter at time k. Hence, we have

$$\begin{aligned} \mathbb{B}_s(t, \epsilon) &= \left| \sum_{n=1}^{t} \int s_n(x_{n-1:n})[\pi(x_{n-1:t}|y_{1:t}) - \pi^\epsilon(x_{n-1:t}|y_{1:t})]dx_{n-1:t} \right| \\ &= \left| \sum_{n=1}^{t} \int s_n(x_{n-1:n}) \left[\pi(x_t|y_{1:t}) \prod_{k=n-1}^{t-1} B_{\pi_k}(x_k|x_{k+1}) \right. \right. \\ &\qquad \left. \left. - \pi^\epsilon(x_t|y_{1:t}) \prod_{k=n-1}^{t-1} B_{\pi_k^\epsilon}(x_k|x_{k+1}) \right] dx_{n-1:t} \right| . \end{aligned}$$

We will deal with each summand directly. We have the telescopic decomposition

$$\begin{aligned}
&\int s_n(x_{n-1:n})\Big[\pi(x_t|y_{1:t})\prod_{k=n-1}^{t-1} B_{\pi_k}(x_k|x_{k+1}) - \pi^{\epsilon}(x_t|y_{1:t})\prod_{k=n-1}^{t-1} B_{\pi_k^{\epsilon}}(x_k|x_{k+1})\Big]dx_{n-1:t}\\
&\quad = \sum_{s=1}^{t-n+1}\int s_n(x_{n-1:n})\Big[\pi(x_t|y_{1:t})\Big\{\prod_{k=t-s+1}^{t-1} B_{\pi_k^{\epsilon}}(x_k|x_{k+1})\Big\}\Big\{\prod_{k=n-1}^{t-s} B_{\pi_k}(x_k|x_{k+1})\Big\}\\
&\qquad\qquad - \pi(x_t|y_{1:t})\Big\{\prod_{k=t-s}^{t-1} B_{\pi_k^{\epsilon}}(x_k|x_{k+1})\Big\}\Big\{\prod_{k=n-1}^{t-s-1} B_{\pi_k}(x_k|x_{k+1})\Big\}\Big]dx_{n-1:t}\\
&\qquad + \int s_n(x_{n-1:n})[\pi(x_t|y_{1:t}) - \pi^{\epsilon}(x_t|y_{1:t})]\prod_{k=n-1}^{t-1} B_{\pi_k^{\epsilon}}(x_k|x_{k+1})dx_{n-1:t}.
\end{aligned}$$

By Proposition 3.3.2, the final term is controlled by $C_1\epsilon$ for some $C_1 < \infty$ which does not depend upon t or ϵ. Hence, we treat the sum over s; one can easily rewrite the summand as

$$\begin{aligned}
&\int s_n(x_{n-1:n})\pi(x_t|y_{1:t})\Big\{\prod_{k=t-s+1}^{t-1} B_{\pi_k^{\epsilon}}(x_k|x_{k+1})\Big\}\Big[B_{\pi_{t-s}^{\epsilon}}(x_{t-s}|x_{t-s+1})\\
&\qquad\qquad - B_{\pi_{t-s}}(x_{t-s}|x_{t-s+1})\Big] \times \Big\{\prod_{k=n-1}^{t-s-1} B_{\pi_k}(x_k|x_{k+1})\Big\}dx_{n-1:t}.
\end{aligned}$$

We rewrite this in operator notation:

$$\pi_t B_{\pi_{t-1}^{\epsilon}:\pi_{t-s+1}^{\epsilon}}\big[B_{\pi_{t-s}} - B_{\pi_{t-s}^{\epsilon}}\big]B_{\pi_{t-s-1}:\pi_{n-1}}(s_n)$$

where

$$\pi_t B_{\pi_{t-1}^{\epsilon}:\pi_{t-s+1}^{\epsilon}}(\varphi) := \int_{\mathbb{R}^{sd_x}} \pi(x_t|y_{1:t})\Big\{\prod_{k=t-s+1}^{t-1} B_{\pi_k^{\epsilon}}(x_k|x_{k+1})\Big\}\varphi(x_{t-s+1})dx_{t-s+1:t}$$

and in an abuse of notation

$$B_{\pi_{t-s-1}:\pi_{n-1}}(s_n)(x_{t-s}) := \int_{\mathbb{R}^{(t-s-n+1)d_x}}\Big\{\prod_{k=n-1}^{t-s-1} B_{\pi_k}(x_k|x_{k+1})\Big\}s_n(x_{n-1:n})dx_{n-1:t-s-1}.$$

Now, by standard results, we have

$$\begin{aligned}\left|\pi_t B_{\pi^\epsilon_{t-1}:\pi^\epsilon_{t-s+1}}\left[B_{\pi_{t-s}} - B_{\pi^\epsilon_{t-s}}\right] B_{\pi_{t-s-1}:\pi_{n-1}}(s_n)\right| \\ \leq \left\|\pi_t B_{\pi^\epsilon_{t-1}:\pi^\epsilon_{t-s+1}}\left[B_{\pi_{t-s}} - B_{\pi^\epsilon_{t-s}}\right]\right\|_{TV} \times \prod_{l=n-1}^{t-s-1} \beta(B_{\pi_l})\|s_n\|_\infty.\end{aligned}$$

By (A5) for every $x, y \in \mathbb{R}^{d_x}$, $n \in \{1, \ldots, t-1\}$,

$$\frac{\underline{\kappa}_1^2\underline{\kappa}_2}{\lambda_1^3\lambda_2\overline{\kappa}_1} \leq B_{\pi_n}(y|x) \leq \frac{\lambda_1^3\lambda_2\overline{\kappa}_1^2\overline{\kappa}_2}{\underline{\kappa}_1}$$

hence for any $x, x', y \in \mathbb{R}^{d_x}$, $n \in \{1, \ldots, t-1\}$, we have

$$B_{\pi_n}(y|x) \geq \frac{\frac{\underline{\kappa}_1^2\underline{\kappa}_2}{\lambda_1^3\lambda_2\overline{\kappa}_1}}{\frac{\lambda_1^3\lambda_2\overline{\kappa}_1^2\overline{\kappa}_2}{\underline{\kappa}_1}} B_{\pi_n}(y|x')$$

and thus

$$\beta(B_{\pi_n}) \leq 1 - \frac{\frac{\underline{\kappa}_1^2\underline{\kappa}_2}{\lambda_1^3\lambda_2\overline{\kappa}_1}}{\frac{\lambda_1^3\lambda_2\overline{\kappa}_1^2\overline{\kappa}_2}{\underline{\kappa}_1}} := \rho \in (0, 1).$$

By using Lemma 4.4.1 one has that for some $C_2 < \infty$ that does not depend on n, ϵ or t

$$\left\|\pi_t B_{\pi^\epsilon_{t-1}:\pi^\epsilon_{t-s+1}}\left[B_{\pi_{t-s}} - B_{\pi^\epsilon_{t-s}}\right]\right\|_{TV} \leq C_2\epsilon.$$

Putting together the above arguments, we have that

$$\left|\pi_t B_{\pi^\epsilon_{t-1}:\pi^\epsilon_{t-s+1}}\left[B_{\pi_{t-s}} - B_{\pi^\epsilon_{t-s}}\right] B_{\pi_{t-s-1}:\pi_{n-1}}(s_n)\right| \leq C_3\epsilon\rho^{t-s-n+1}$$

where $C_3 = C_2 \cdot \|s_n\|_\infty$, so that $C_3 < \infty$ and is independent of ϵ, n and t.

Thus we have that

$$\mathbb{B}_s(t, \epsilon) \leq \sum_{n=1}^{t}\left[\sum_{s=1}^{t-n+1} C_3\epsilon\rho^{t-s-n+1} + C_1\epsilon\right].$$

It is straightforward to show that

$$\sum_{s=1}^{t-n+1} \rho^{t-s-n+1} = \frac{1-\rho^{t-n+1}}{1-\rho},$$

so that

$$\mathbb{B}_s(t,\epsilon) \leq \sum_{n=1}^{t} \left[C_3 \frac{1-\rho^{t-n+1}}{1-\rho} + C_1 \right] \epsilon.$$

The right-hand side can easily be rewritten as

$$C_1 \epsilon t + C_3 \epsilon \sum_{n=1}^{t} \frac{1-\rho^{t-n+1}}{1-\rho}.$$

Since $\rho \in (0,1)$, this is strictly bounded above by

$$C_1 \epsilon t + C_3 \epsilon \sum_{n=1}^{t} \frac{1}{1-\rho} = \left[C_1 + \frac{C_3}{1-\rho} \right] \epsilon t.$$

Hence

$$\mathbb{B}_s(t,\epsilon) < C \epsilon t$$

for some positive, finite $C \in \mathbb{R}$, independent of n, ϵ, t. □

Lemma 4.4.1. *Assume (A2,A3,A5). Then there exist a $C < \infty$ such that for any $k \in \{2, \ldots, t-1\}$ $\epsilon > 0$, $\{y_{1:k}\}$ and $\varphi \in \mathcal{B}_b(\mathbb{R}^{d_x})$ we have*

$$\sup_{x \in^{d_x}} \left| \int \left[B_{\pi_k^\epsilon}(x,z) - B_{\pi_k}(x,z) \right] \varphi(z) dz \right| \leq C\epsilon$$

where $B.$ is defined in (4.8) and π_k and π_k^ϵ are the true filter and its ABC approximation, respectively.

Proof. We have the decomposition:

$$\int \left[B_{\pi_k^\epsilon}(x,z) - B_{\pi_k}(x,z) \right] \varphi(z) dz = \int \varphi(z) q(x|z) \left[\frac{\pi_k^\epsilon(z) - \pi_k(z)}{\int \pi_k^\epsilon(u) q(x|u) du} \right.$$
$$\left. + \pi_k(z) \left\{ \frac{\int [\pi_k(u) - \pi_k^\epsilon(u)] q(x|u) du}{\int \pi_k^\epsilon(u) q(x|u) du \int \pi_k(u) q(x|u) du} \right\} \right] dz$$

where we have suppressed both the data and the time index for the transition density q_{k+1} from the notation.

Dealing with the first part, we have for some C that does not depend upon x, k, ϵ or $\{y_{1:k}\}$

$$\int \varphi(z)q(x|z)\left[\frac{\pi_k^\epsilon(z) - \pi_k(z)}{\int \pi_k^\epsilon(u)q(x|u)du}\right]dz \leq \frac{\lambda_1}{\underline{\kappa_1}}C\epsilon\lambda_1\overline{\kappa_1}\|\varphi\|_\infty$$

where we have used (A5) to control q in both the numerator and denominator of the integrand, and then Proposition 3.3.2. Now, for the second part

$$\int \varphi(z)q(x|z)\pi_k(z)\left\{\frac{\int[\pi_k(u) - \pi_k^\epsilon(u)]q(x|u)du}{\int \pi_k^\epsilon(u)q(x|u)du \int \pi_k(u)q(x|u)du}\right\}dz \leq \left(\frac{\lambda_1}{\underline{\kappa_1}}\right)^2 \lambda_1\overline{\kappa_1}\|\varphi\|_\infty C\epsilon$$

where, again (A5) has been applied along with Proposition 3.3.2 and C does not depend upon x, k, ϵ or $\{y_{1:k}\}$. Using the uniformity in x of the above bounds allows us to conclude. □

Proposition 4.4.1 establishes a linear bound for the error induced by the ABC approximation of the joint smoothing density. That it is only linear encourages the use of ABC in this context and motivates the simulation studies that appear in Section 4.7.

As has been mentioned, it is often useful in practice to set the ABC tolerance parameter adaptively, in order to encourage the survival of a given proportion of particles in the resampling step of the SMC algorithms. In this situation, we extend the above result and state that

$$\mathbb{B}_s(t, \epsilon) \leq Ct \max_{1\leq n\leq t}\{\epsilon_n\}.$$

4.5 Incorporating ABC into SMC Smoothing Procedures

The theoretical results in the previous section motivate the use of the SMC and PMCMC procedures detailed in Section 4.3 for the estimation of smoothed additive functionals, defined with respect to the ABC approximation of the joint smoothing distribution. This section details the development of particle smoothing and forward smoothing procedures for implementation in an ABC framework.

As with the design of the ABC filtering procedures, the ABC smoothers will be constructed by altering the weighting steps within the algorithms. For particle and forward smoothing, respectively, this is achieved by replacing Step 3 in Algorithms 8 and 10 with the ABC weighting step in Algorithm 12:

```
3. Set $t = t + 1$. For $i = 1, \ldots, N$, sample $X_t^{(i)}$ and $U_t^{(i)}$ according to the
   proposal and observation densities, respectively, and calculate the filtering
   weights
```

$$W_t^{(i)} \propto W_{t-1}^{(i)} \cdot \frac{G_t^{\epsilon_t}\left(U_t^{(i)}, y_t\right) q_{t,\theta}\left(X_t^{(i)} \middle| X_{t-1}^{(i)}\right)}{k_t\left(X_t^{(i)} \middle| X_{t-1}^{(i)}\right)}, \qquad \sum_{i=1}^{N} W_t^{(i)} = 1.$$

Algorithm 12: A weighting step for an ABC smoothing procedure

As with the ABC filtering algorithms presented in Chapter 3, we focus our attention on the use of the indicator kernel:

$$G_t^{\epsilon}\left(u_t, y_t\right) = \mathbb{I}_{A_{\epsilon,y_t}}\left(u_t\right),$$

with $A_{\epsilon,y_t} = \{u; \rho\left(s(u), s(y_t)\right) < \epsilon\}$, the $\mathbb{L}_1$-distance metric ρ and the identity function $s(\cdot)$. As in the filtering framework, we note also the availability of a multiple-pseudo-data (MPD) ABC procedure, where the following kernel, also based upon the indicator function, is used:

$$G_t^{\epsilon}\left(u_t, y_t\right) = \frac{1}{J} \sum_{j=1}^{J} \mathbb{I}_{A_{\epsilon,y_t}}\left(u_{t,j}\right),$$

for $J \geq 1$ where, in an abuse of notation, $u_t = \{u_{t,1}, \ldots, u_{t,J}\}$.

As previously, the SMC procedures were implemented with dynamic resampling, using the estimated ESS as the resampling criterion, with an associated threshold of $N/2$ particles. An adapted ABC precision parameter, recalculated at each iteration such that the survival of a given proportion of particles at that iteration is highly probable, may prove beneficial in preventing a collapse of the particle system, and so this approach is adopted for the implementation of the SMC-based smoothing experiments in Section 4.7.

In Chapter 3, it was observed that the effect of the SMC weight degeneracy issue, detailed in Section 2.3.1, is a dramatic increase in the resampling rate of the particle filter. It was noted that the exact particle filter suffers from this degeneracy to an increased extent as the dimension of the hidden state increases, and it was further observed in Section 3.6 that the ABC particle filter is less susceptible to SMC weight degeneracy than the exact particle filter. As a result of its lower susceptibility to this particular weight degeneracy issue, the performance of the ABC filter was observed to be more robust to increases in model dimension.

This decrease in susceptibility to the SMC weight degeneracy issue is expected to carry over to both the particle smoothing and forward smoothing procedures when targeting the ABC approximation of the joint smoothing density (4.15). Since the particle filter and smoother are procedurally equivalent, this is intuitive. For the forward smoother, we note that the calculation of the filtering weights is exactly the same as in both the particle filter and smoother. It is the recursive nature of this calculation, and the resulting compounding of large incremental weights that leads to a significant increase in the variance of the weights at successive iterations, and it is this issue that is avoided by using the proposed ABC approximations to the likelihood.

The Rejection SMC smoothing procedures, as detailed in Section 4.3.4, may also be used to target the ABC approximation of the joint smoothing density. The resulting procedures for RSMC particle smoothing and RSMC forward smoothing for an ABC target are obtained by replacing the resampling step in the ABC smoothing algorithms with the rejection-resampling step (e.g. Step 2 in Algorithm 11). These RSMC smoothing procedures are also examined for the ABC approximation of a nonlinear SSM in Section 4.7.

4.6 Incorporating ABC into PMMH Procedures

In this section, we will consider the use of PMCMC methods for performing inference with respect to the ABC approximation (4.15) of the joint smoothing density, motivated by the theoretical results of Section 4.4. In particular, we will use PMMH procedures to estimate the ABC approximation of the smoothed additive functional $\mathcal{S}_T^{\epsilon}$. We also use this procedure to learn the unknown parameter vector θ in a batch setting; we do not perform online parameter estimation.

Consider the particle marginal Metropolis Hastings (PMMH) procedure from Section 2.4.3. Recall that the acceptance probability at each MH iteration is given by

$$\alpha\left(\left\{x_{1:T},\theta\right\},\left\{x_{1:T}^{*},\theta^{*}\right\}\right)=1\wedge\frac{\pi_{\theta^{*}}\left(y_{1:T}\right)\pi\left(\theta^{*}\right)}{\pi_{\theta}\left(y_{1:T}\right)\pi\left(\theta\right)}\frac{k\left(\theta\left|\theta^{*}\right.\right)}{k\left(\theta^{*}\left|\theta\right.\right)},$$

where $k(\theta^{*}|\theta)$ is the proposal density used to generate candidate values θ^{*} of the parameter vector, and $\pi_{\cdot}(y_{1:T})$ is the marginal likelihood of the observed data. Recall also that, in practice, the marginal likelihoods in the acceptance probability are replaced with their particle approximations, provided by the SMC proposal scheme that generates the candidate state $X_{1:T}^{*}$.

We proceed by proposing two PMMH procedures for estimating the ABC smoothed additive functional $\mathcal{S}_T^{\epsilon}$.

4.6.1 ABC PMMH with Particle Selection Updates

As with the PMCMC procedures described in Section 2.4, one need only draw a sample from the empirical SMC representation of $\pi_{1:T,\theta}^{\epsilon}$, and so an ABC PMMH procedure is proposed here in which an SMC procedure that targets the ABC approximation of the joint smoothing distribution $\pi_\theta^\epsilon(dx_{1:T}|y_{1:T})$ is used at each MH iteration to generate the system of particles $\left\{X_{1:T}^{(i)}, W_{1:T}^{(i)}\right\}_{i=1}^{N}$. Selecting from the generated particle system the single candidate particle $X_{1:T}^* = X_{1:T}^{(i)}$ with probability $W_T^{(i)}$, we calculate a candidate value for the smoothed additive functional,

$$\mathcal{S}_T^* = \sum_{t=1}^{T} s_t(X_{t-1}^*, X_t^*),$$

where we recall that $X_0^* = x_0$ is assumed known. This candidate smoothed additive functional is then accepted or rejected according to $\alpha\left(\{x,\theta\}, \{x^*,\theta^*\}\right)$, defined as above, where the marginal likelihoods are estimated using the filtering weights $\left\{W_{1:T}^{(i)}\right\}_{i=1}^{N}$. Recall that the marginal likelihood estimates can be calculated recursively, and so the filtering weights for each time step $t = 1, \ldots, T$ do not need to be stored.

The standard proposal scheme used here, where one particle from the generated N-particle system is used as the candidate state, will be referred to in this thesis as the particle selection update scheme. Similarly, the PMMH procedure described here will be referred to as particle selection PMMH. We now consider an alternative update scheme that makes use of all the particles generated at each MH iteration.

4.6.2 ABC PMMH with Forward Smoothing

In the particle selection PMMH procedure, the majority of the particles generated at each MH iteration are discarded without being used; this is a significant waste of information about the joint smoothing distribution. We introduce here a PMMH procedure that uses a forward smoothing update scheme, which makes use of all the particles generated at each MH iteration. This will be referred to as forward smoothing PMMH.

The implementation of the forward smoothing PMMH procedure is much the same as the particle selection PMMH procedure. Indeed, it is stressed in particular that this method does not involve a novel proposal scheme, as such; both the particle selection and forward smoothing update schemes are SMC procedures, and may use the same methods for generating the particle system $\left\{X_{1:T}^{(i)}, W_{1:T}^{(i)}\right\}_{i=1}^{N}$. Rather, the forward smoothing update scheme proposed here can be viewed as simply introducing a post-processing step at each time step of the particle selection update scheme.

At each MH iteration, the candidate value for the smoothed additive functional is proposed through the use of a forward smoothing procedure, as described in Algorithm 10. The joint marginal likelihood of the observed data can also be estimated at each iteration using the filtering weights generated in the forward pass of the forward smoothing update.

It will be of interest to compare the performance of both the particle selection PMMH and forward smoothing PMMH procedures when they are used to estimate both the smoothed additive functional and its ABC approximation. This will be considered in the next section.

4.7 Example: A Nonlinear SSM

We return to the example from Section 3.6 and consider the use of both particle smoothing and forward smoothing procedures, as well as a selection of PMMH procedures for performing inference with respect to the ABC approximation of the SSM (3.13)-(3.14). The RSMC alternatives to the smoothing procedures will also be considered.

Throughout, we consider the accuracy of estimating the expected mean state over the observation period $[0, 100]$, i.e. we are interested in estimating the smoothed additive functional, $\mathcal{S}_{100}$, defined by (4.2) and specified by the choice of $s_n\left(x_{n-1}, x_n\right)$ as in (4.3). We will use the nonlinear SSM as defined in (3.13)-(3.14) with $d_x =$

$d_y = d$, which will be further specified by

$$\begin{aligned} V_t &\sim \mathcal{N}\left(0, \sigma_X^2 I_d\right) \quad (i.i.d.), \\ Z_t &\sim \mathcal{N}\left(0, \sigma_Y^2 I_d\right) \quad (i.i.d.), \end{aligned}$$

with $\sigma_X^2 = 10$ and $\sigma_Y^2 = 1$. Once again the unobserved state is initialised by $X_0 = 0_d$, the d-length zero vector.

We perform both particle smoothing and forward smoothing for this problem, and we focus on the effects that the use of the ABC and RSMC methodologies have, both on the algorithmic performance of each of the procedures and on the accuracy of the resulting estimates of $\mathcal{S}_T$. In particular, we examine the behaviour of each of the smoothers and their estimates as the number of particles and the dimension d of the state and observation spaces spaces change.

A direct comparison between the particle smoothing and forward smoothing procedures in the SMC framework is omitted, as the SMC estimate of a smoothed additive functional is known to be less accurate when using particle smoothing than when using forward smoothing. Since both the size of the ABC bias (4.17), and the degree to which the SMC error is reduced when targeting the ABC smoothing distribution, are independent of the particular smoothing procedure used, the superior accuracy of the forward smoothing procedure will hold for the estimation of the ABC approximation of $\mathcal{S}_T$. The numerical comparison of these alternate smoothing procedures when applied to the ABC approximation is therefore also unnecessary.

4.7.1 Implementation Details

The numerical results presented here will culminate in a comparison of the PMMH procedures with updates provided by the competing particle selection and forward smoothing procedures. It will be of particular interest to compare the performance of these methods when executed with comparable computational cost, which will

	N_1	N_2	N_3	N_4	N_5
Particle Smoothing	4427	17,139	39,020	68,258	107,007
Forward Smoothing	100	200	300	400	500
	N_6	N_7	N_8	N_9	N_{10}
Particle Smoothing	155,698	207,938	275,755	343,524	424,085
Forward Smoothing	600	700	800	900	1,000

Table 4.1: Particle system sizes used for the implementation of the forward smoothing and particle smoothing procedures, in order to obtain comparable algorithm runtimes.

be determined by the number of particles used for each SMC update scheme. Recall that the computational cost of the forward smoothing algorithm is $O(N^2)$ and note that the particle selection update scheme will be of the same computational complexity as the particle smoothing procedure, $O(N)$.

A naïve initial approach to obtaining comparable computational costs would be to match each N-particle implementation of the forward smoothing procedure with an N^2-particle implementation of the particle smoothing procedure, however this does not achieve the desired comparability. This is due to the fact that it is only the estimation procedure within the forward smoothing algorithm that is $O(N^2)$; the particle propagation, particle weighting and resampling mechanisms are all identical to the corresponding sections of the particle smoothing procedure, and are $O(N)$.

To obtain suitable values of N to use in the final implementations, preliminary runs of the two smoothing procedures were executed to compare run times; for forward smoothing, N was set to take values from 100 to 1000, at intervals of 100, and the values of N for which the particle smoothing procedure had a matching runtime (to within 1% when measured in seconds) are given in Table 4.1. As mentioned above, direct comparisons are not made between the particle smoothers and forward smoothers in an SMC framework. Although the comparability of runtimes was therefore unnecessary in this setting, the above values of N were used in the implementation of the SMC smoothers, for sake of consistency.

In assessing the accuracy of the various SMC and PMMH procedures considered here, we require a 'true' value against which to measure the estimates $\hat{\mathcal{S}}_T$. For the SMC smoothing procedures, we use the mean estimate obtained over 50 implementations of the forward smoothing procedure with 5000 particles, and for the PMMH procedures we use the estimate generated by a standard PMMH procedure, using particle selection updates with $200,000$ particles; both of these procedures targeted the exact SSM.

The accuracy of each of the smoothing procedures used to estimate $\mathcal{S}_T$ was measured using the dimension-adjusted $\mathbb{L}_1$-errors calculated as

$$\frac{1}{d}\sum_{i=1}^{d}\left|\hat{\mathcal{S}}_{T,i}-\mathcal{S}_{T,i}\right|, \tag{4.18}$$

where $\hat{\mathcal{S}}_T$ is the SMC or PMMH estimate of the expected mean state sequence over the interval of interest and $\mathcal{S}_T$ is the corresponding assumed true value over that interval. These are the smoothing errors, and will be referred to with reference to the particular smoothing procedure to which they refer; e.g. if $\hat{\mathcal{S}}_T$ is calculated using the particle smoothing procedure then we refer to the particle smoothing errors or the ABC particle smoothing errors, depending on whether the true SSM or its ABC approximation was targeted, respectively.

To ensure consistency of the results, each of the SMC smoothing procedures were repeated, with the aim being to generate 50 independent estimates of $\mathcal{S}_t$ for each parameter pairing $\{N, d\}$. For some combinations of the parameters, however, the ABC smoothers were observed to collapse due to the ABC weight degeneracy issue. When an algorithm collapsed, the seed for the pseudo-random number generator was reset, and that implementation was repeated; a maximum of 50 independent repetitions per implementation were allowed before continuing to the next implementation. As a result, a maximum number of 2500 repetitions using the same parameter pairing $\{N, d\}$ before moving onto the next set of parameter values.

For each of the ABC smoothing procedures, a sequence of ABC tolerances $\{\epsilon_t\}_{t=1}^T$ was set adaptively using the scheme described in Section 3.4.2: at time t, the tolerance ϵ_t was set to be the α-largest distance between the set of auxiliary data generated at the previous time point, $\left\{u_{t-1}^{(i)}\right\}_{i=1}^N$, and the corresponding observed data y_{t-1}. The control parameter α was set at $\lfloor 0.8N \rfloor$ for all implementations considered.

When performing PMMH for an ABC target, a fixed value of ϵ was used for implementing the SMC proposal schemes across all MCMC iterations. This value was set at the start of each PMMH procedure, by running a single iteration of the SMC proposal, with adaptive tuning of the tolerance parameter as described above, again using $\alpha = \lfloor 0.8N \rfloor$. The ABC tolerance for the remainder of the PMMH procedure was then fixed at the maximum value in the sequence $\{\epsilon_t\}_{t=1}^T$ generated by this preliminary run.

4.7.2 Results and Analysis - SMC Smoothing Procedures

ABC Particle Smoothing vs. Particle Smoothing

We note first that none of the tested parameter combinations $\{N, d\}$ resulted in a consistent collapse of the ABC particle smoother, where we recall that a consistent collapse is defined here as the collapse of the smoother at every attempted implementation, due to the ABC weight degeneracy issue. The lack of observed complete collapses is to be expected here, as the particle system sizes that were used (see Table 4.1) were large enough to avoid the ABC weight degeneracy issue.

We proceed by comparing the smoothing errors (4.18) obtained by the particle smoothing procedure when applied to the nonlinear SSM and its ABC approximations. The mean smoothing errors, obtained over 50 implementations of each procedure, are presented in Figure 4.1, with the error bars corresponding to the standard errors obtained from each sample of 50 smoothing errors; the exact smoothing errors

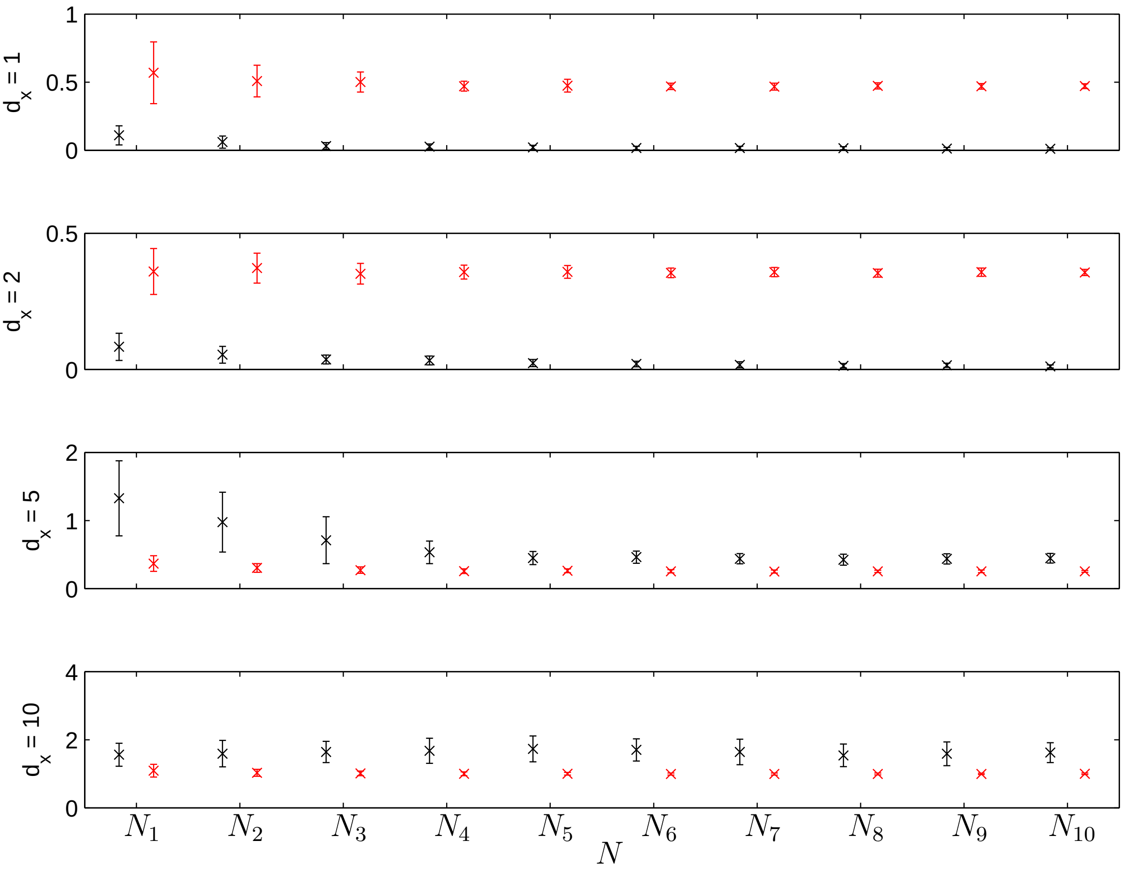


Figure 4.1: Smoothing errors for the exact (black) and ABC (red) particle smoothing estimates of the mean smoothed state in the nonlinear SSM (3.13)-(3.14), calculated over the interval of length $T = 100$. The crosses and errorbars indicate the mean and standard errors of the smoothing errors obtained over 50 independent implementations of each procedure. The nonlinear SSM is specified by $\sigma_X^2 = 10$, $\sigma_Y^2 = 1$.

are displayed in black, with the ABC smoothing errors in red.

From Figure 4.1, we immediately note the expected performance of the exact particle smoother. As the dimension d increases, the accuracy of the estimates deteriorates and for low to moderate dimensions ($d \in \{1, 2, 5\}$), there is an improvement in accuracy as N increases, illustrated by a decrease in both the mean and standard error of the exact particle smoothing errors. For $d = 10$, there is no such improvement in accuracy as N increases. As was observed for the particle filter in Section 3.6, this is indicative of the SMC approximation of the particle distribution breaking down due to significant SMC weight degeneracy. This is corroborated by

	$N = N_1$	$N = N_2$	$N = N_3$	$N = N_4$	$N = N_5$	$N = N_6$	$N = N_7$	$N = N_8$	$N = N_9$	$N = N_{10}$
d = 1	73.33%	73.13%	72.99%	73.11%	72.81%	72.87%	72.83%	72.85%	72.87%	72.85%
d = 2	96.67%	96.57%	96.44%	96.46%	96.42%	96.38%	96.40%	96.26%	96.36%	96.34%
d = 5	99.96%	99.94%	100.00%	100.00%	99.98%	100.00%	100.00%	100.00%	100.00%	100.00%
d = 10	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%

Table 4.2: Average resampling rates obtained over 50 implementations of the exact particle smoothing procedure, when applied to the nonlinear SSM (3.13)-(3.14) using $N \in N_{1:10}$.

the observed SMC resampling rates presented in Table 4.2, where a value of 100% implies that resampling due to weight degeneracy was required at each of the time steps at which it was considered.

From Table 4.2, it is also evident that there was significant weight degeneracy observed for $d = 5$, yet an improvement in accuracy was noted for increasing N. Indeed, from Figure 4.1, note that the accuracy of the exact particle smoother is very poor for low values of N, where the weight degeneracy issue has the greatest effect. For increasing N in this dimension, we note that there is a degree of masking, where, although the smoother was required to resample at almost every time step, the use of large particle systems reduced the effects of the particle path degeneracy phenomenon. It is further noted that additional implementations of the particle smoothing procedure were executed using values of N up to $N = 1,696,400$, and no such masking was observed in the exact particle smoothing estimates for $d = 10$.

Consider now the ABC particle smoothing errors presented in Figure 4.1. As seen in Sections 3.6 and 3.5 for the ABC particle filter, the mean ABC particle smoothing error is observed to be dominated by the bias of the ABC approximation, indicated by the apparent insensitivity of the mean ABC particle smoothing error to significant

	$N = N_1$	$N = N_2$	$N = N_3$	$N = N_4$	$N = N_5$
d = 1	31.92%	32.22%	32.24%	32.32%	32.30%
d = 2	34.08%	34.14%	34.10%	34.28%	34.28%
d = 5	37.21%	37.35%	37.37%	37.37%	37.37%
d = 10	44.08%	44.40%	44.44%	44.44%	44.44%
	$N = N_6$	$N = N_7$	$N = N_8$	$N = N_9$	$N = N_{10}$
d = 1	32.32%	32.32%	32.32%	32.32%	32.32%
d = 2	34.34%	34.32%	34.34%	34.34%	34.34%
d = 5	37.37%	37.37%	37.37%	37.37%	37.37%
d = 10	44.44%	44.44%	44.44%	44.44%	44.44%

Table 4.3: Average resampling rates obtained over multiple implementations of the particle smoothing procedure, when applied to the ABC approximation of the nonlinear SSM (3.13)-(3.14) using $N \in N_{1:10}$.

increases in N.

Note that, for all d tested, the variance of the errors is observed to decrease as N increases and, in moderate to high dimensions, the ABC particle smoothing errors are lower than the corresponding exact particle smoothing errors. Both of these observations indicate that the SMC approximation of the ABC approximation of the smoothing distribution has not collapsed, and that the ABC particle smoother is more resilient to the SMC weight degeneracy issue. This is confirmed by considering the resampling rates for the ABC particle smoother, presented in Table 4.3.

Although the resampling rates for the ABC particle smoother were observed to increase with the dimension d, indicating an increase in weight degeneracy, comparison with Table 4.2 confirms that the ABC particle smoother is less prone to weight degeneracy than the exact particle smoother.

In general, these results are encouraging for ABC particle smoothing, as the ABC bias, which is observed to dominate the particle smoothing error can be controlled through the tolerance parameter ϵ. Furthermore, these results suggest that ABC particle smoothing can be employed as a valid alternative to exact particle smoothing in high dimensions.

ABC Forward Smoothing vs. Forward Smoothing

We now turn to the implementations of the forward smoothing procedure, noting first that none of the tested parameter combinations $\{N, d\}$ resulted in a complete collapse of the ABC forward smoother, although a repeated collapse was observed when implementing the ABC forward smoother with $\{N, d\} = \{100, 10\}$; for this implementation, 24 of 50 independent implementations were successful. It is also noted that, for this parameter pairing, the ABC forward smoother was also observed to collapse multiple times when using the RSMC methodology of Section 2.3.2; for this procedure, 36 of 50 implementations were successful. For all other parameter combinations $\{N, d\}$ attempted, all 50 independent implementations of the ABC forward smoothers, using both SMC and RSMC resampling schemes, were successful.

The mean exact and ABC forward smoothing errors, calculated as in (4.18), along with the standard errors obtained from each sample of smoothing errors, are presented in Figure 4.2.

Broadly speaking, the observations that can be made here are similar to those made for the particle smoothing procedures. Consider first the errors presented for the exact forward smoother. In low dimensions ($d \in \{1, 2\}$), the accuracy of the exact forward smoothing procedure, measured in terms of the mean and standard error of the resulting smoothing errors, is observed to increase as N increases. In moderate to high dimensions ($d \in \{5, 10\}$), the exact forward smoothing procedure was also shown to be susceptible to significant SMC weight degeneracy; this is demonstrated in the smoothing errors with a lack of improvement in accuracy with increasing N. It should be noted that this is despite the resilience of the forward smoothing procedure to the particle path degeneracy issue. Here, degeneracy in the SMC weights at each time step directly affects the accuracy of the recursively calculated estimates of the auxiliary function $R_t(x_t)$, which in turn affect the accuracy of the forward smoothing estimate. This is in contrast to the effect of SMC weight

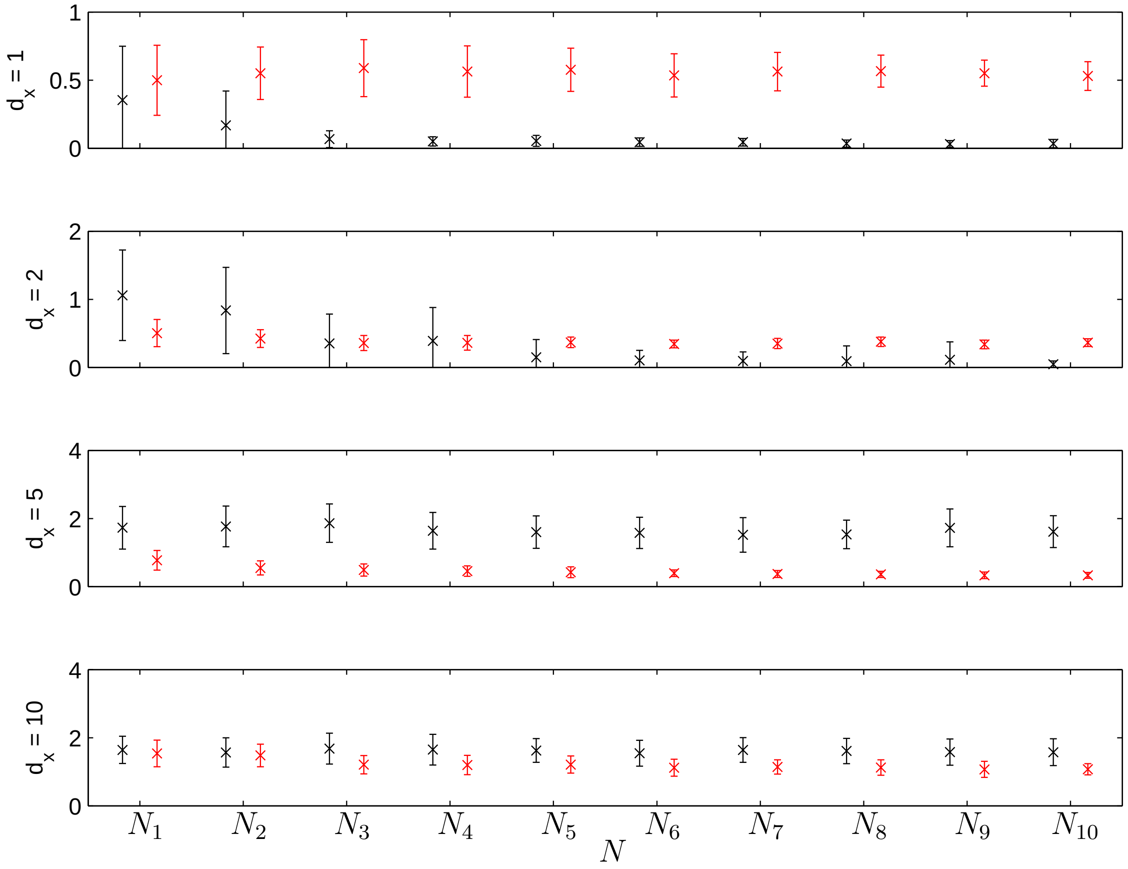


Figure 4.2: Smoothing errors for the exact (black) and ABC (red) forward smoothing estimates of the mean smoothed state in the nonlinear SSM (3.13)-(3.14), calculated over an interval of length $T = 100$. The crosses and errorbars indicate the mean and standard errors of the smoothing errors obtained over 50 independent implementations of each procedure. The nonlinear SSM is specified by $\sigma_X^2 = 10$, $\sigma_Y^2 = 1$.

degeneracy on the particle smoothing estimates, which is realised indirectly through the need to resample often and the resulting path degeneracy issue.

Recall that the effect of SMC weight degeneracy on the accuracy of the exact particle smoother was masked in moderate dimensions by the use of large values of N. Since the forward smoothing procedures were implemented here with far fewer particles, a relative lack of masking might be expected for the forward smoothing results. By increasing N sufficiently, one could achieve masking for the forward smoothing procedure, however this is an impractical solution due to the $O(N^2)$ computational cost of the algorithm.

Consider now the ABC forward smoothing errors, presented in Figure 4.2, in red. We note once again the apparent insensitivity of the magnitude of the ABC forward smoothing errors to significant increases in N and deduce that the ABC forward smoothing error is again being dominated by the bias of the ABC approximation of the smoothing distribution. As before, it is also noted from these results that the ABC forward smoother is more resilient to the SMC weight degeneracy issue than the exact forward smoother, and this is demonstrated in the decreasing variance of the ABC forward smoothing errors and the greater accuracy of these errors in moderate to high dimensions, relative to their exact counterparts. This relative resilience to the SMC weight degeneracy issue was also observed in the SMC resampling rates; the resampling rates for both the exact and ABC forward smoothers are omitted as they add no further information to that provided by the resampling rates for the exact and ABC particle smoothers.

As stated above, the $O(N^2)$ computational cost of the forward smoothing procedure means that one is unable to apply a 'brute force' approach in high dimensions by significantly increasing the number of particles in order to overcome the effect of SMC weight degeneracy. As a result, the superior accuracy of the ABC forward smoother over the exact forward smoother in moderate to high dimensions has added significance.

RSMC Smoothing

The particle smoothing and forward smoothing procedures were also implemented using the RSMC methodology detailed in Section 2.3.2. Recall that the principal objective here is to observe whether any improvement in the variance of the smoothing estimates can be observed when incorporating the RSMC methodology. With this in mind, we consider the standard errors of the smoothing estimates obtained over multiple independent implementations of each procedure. It will also be of interest, however, to observe any effect on accuracy that the use of RSMC methodology might have. We therefore also consider below the mean and standard errors of the smoothing errors obtained from the implementation of each procedure, both with and without the use of RSMC.

To begin, we consider the use of both ERSMC and TRSMC for targeting the exact model; the use of RSMC for performing inference with respect to the ABC approximation of the smoothing distribution will be considered below. The standard errors for the RSMC and SMC particle smoothing estimates are presented in Figure 4.3. The standard errors corresponding to those implementations that used $N \in \{N_1, N_2, N_3\}$ were observed to be significantly larger than those corresponding to $N \geq N_4$. These were therefore omitted for the sake of clarity in presenting the standard errors for $N \geq N_4$, and the vertical axis was scaled accordingly. The corresponding standard errors for the forward smoothing estimates and their RSMC alternatives are presented in Figure 4.4.

From Figure 4.3, we note first that, for $d \in \{1, 2, 5\}$, the standard errors decrease in general as N increases, confirming the results observed for the SMC particle smoother above. That this is also observed for the RSMC procedure is expected. In the one-dimensional case, a consistent marginal improvement over the SMC particle smoothing procedure can be observed for both the ERSMC (blue) and TRSMC (green) particle smoothing procedures. As the dimension increases, however, the advantage of the ERSMC procedure over the SMC benchmark is less evident than for the TRSMC procedure. There are occasions at which the ERSMC procedure provides estimates with lower standard error, e.g. for $\{N, d\} = \{N_9, d_2\}$, however this advantage is not observed in general as N grows. Despite this, the standard errors remain competitive for the both the ERSMC and TRSMC particle smoothers, in comparison to the SMC particle smoother.

From Figure 4.4, we note that no consistent improvement is observed for either of the RSMC forward smoothers over the SMC forward smoother. It is not surprising that the lower asymptotic variance offered by the use of RSMC is more evident in

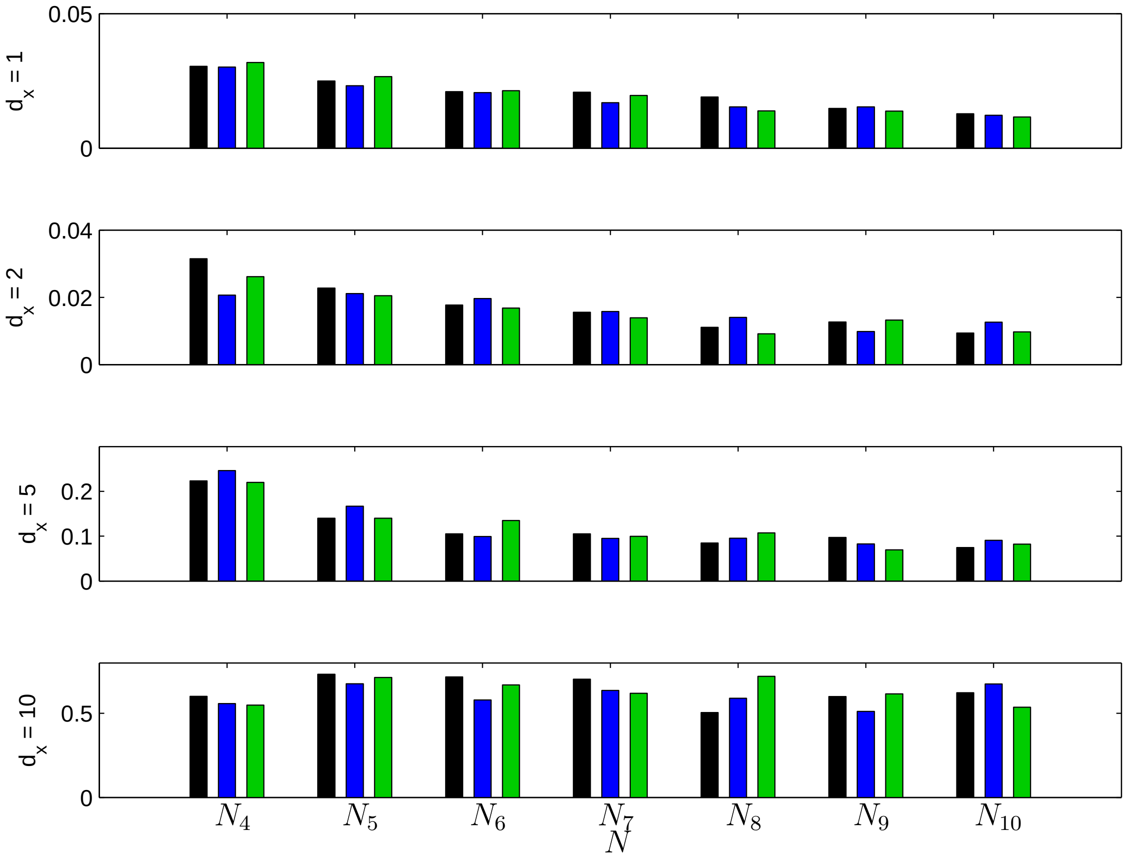


Figure 4.3: Standard errors of the exact SMC, ERSMC and TRSMC particle smoothing estimates of the mean smoothed state in the nonlinear SSM (3.13)-(3.14), calculated over an interval of length $T = 100$. Each plot illustrates the standard errors of 50 independent estimates produced by the exact particle smoothers using SMC (black), ERSMC (blue) and TRSMC (green) kernels. The nonlinear SSM is specified by $\sigma_X^2 = 10$, $\sigma_Y^2 = 1$.

the particle smoothing estimates than in the forward smoothing estimates, as the values of N used for the implementations of the particle smoothers were significantly greater than those used for the forward smoothing procedures; see Table 4.1.

From both Figure 4.3 and Figure 4.4, we further note that the use of RSMC does not seem to offer any additional protection against the SMC weight degeneracy issue that causes the SMC smoothing procedures to suffer as the dimension d increases.

We turn now to the use of RSMC in performing smoothing with respect to the ABC approximation of the nonlinear SSM (3.13)-(3.14), where we recall that the empirical and theoretical acceptance probabilities coincide in this scenario, and so

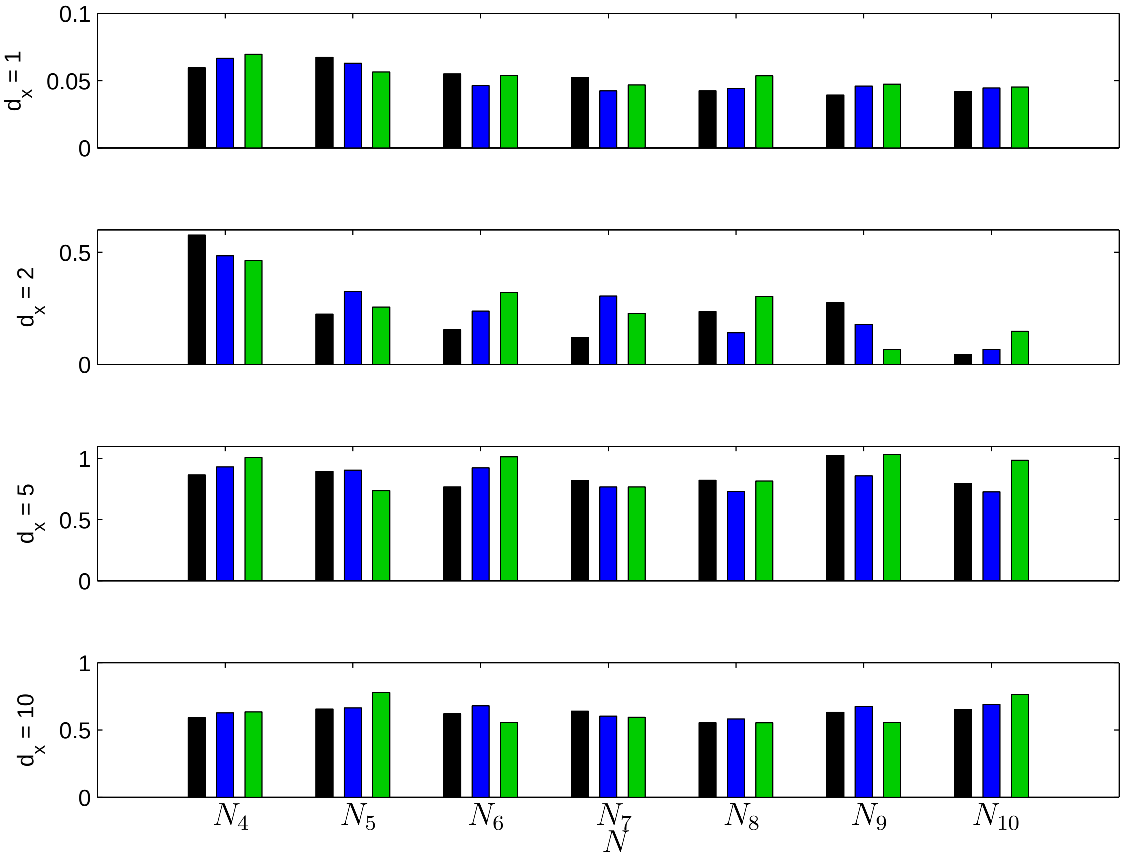


Figure 4.4: Standard errors of the exact SMC, ERSMC and TRSMC forward smoothing estimates of the mean smoothed state in the nonlinear SSM (3.13)-(3.14), calculated over a period of length $T = 100$. Each plot illustrates the standard errors of 50 independent estimates produced by the exact forward smoothers using SMC (black), ERSMC (blue) and TRSMC (green) kernels. The nonlinear SSM is specified by $\sigma_X^2 = 10$, $\sigma_Y^2 = 1$.

the ERSMC and TRSMC procedures are equivalent. The standard errors corresponding to the smoothing estimates produced by the ABC particle smoother and its RSMC alternative are presented in Figure 4.5, and the corresponding results for the ABC forward smoother and its RSMC alternative are presented in Figure 4.6. As with Figures 4.3 and 4.4 above, the results corresponding to the ABC smoothing procedures using $N \leq N_3$ particles are omitted from these figures for the sake of clarity.

For both the ABC particle smoother and the ABC forward smoother, these figures demonstrate that the advantage of lower asymptotic variance in the CLT

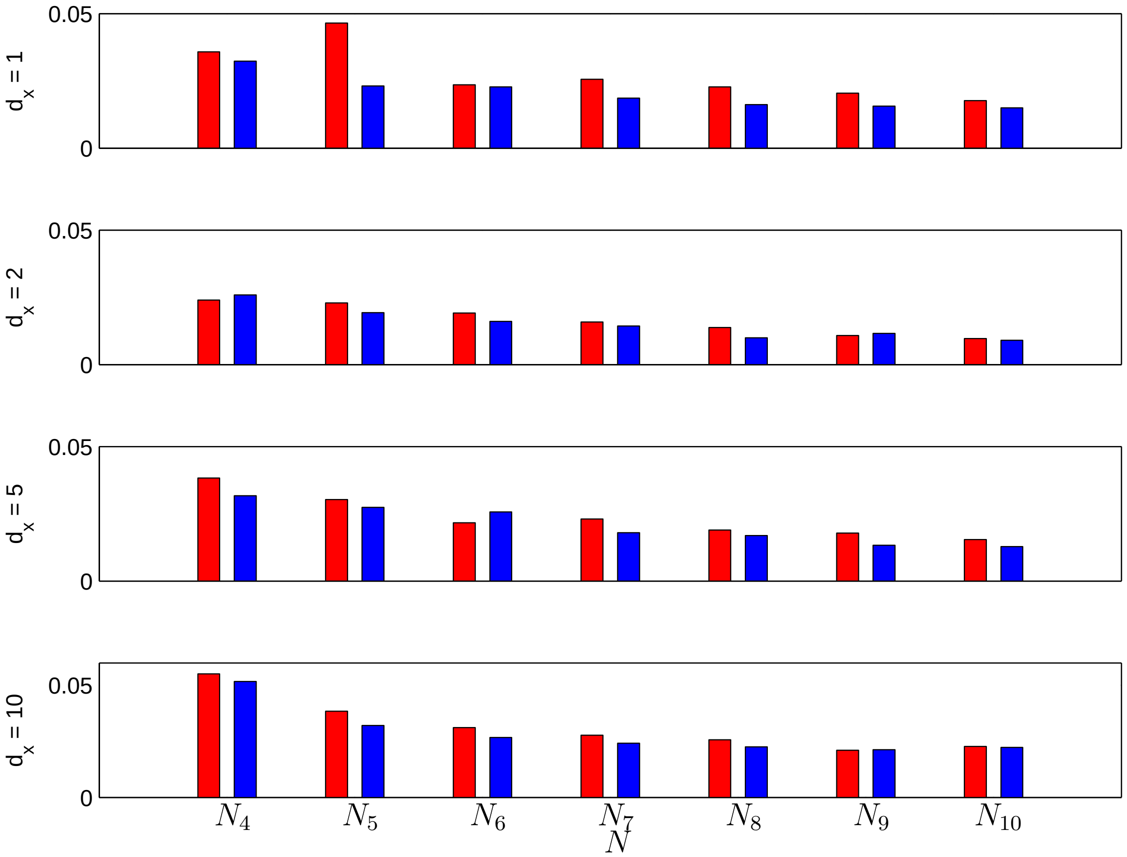


Figure 4.5: Standard errors of the ABC SMC and RSMC particle smoothing estimates of the mean smoothed state in the nonlinear SSM (3.13)-(3.14), calculated over a period of length $T = 100$. Each plot illustrates the standard errors of 50 independent estimates produced by the ABC particle smoothers using SMC (red) and RSMC (blue) kernels. The nonlinear SSM is specified by $\sigma_X^2 = 10$, $\sigma_Y^2 = 1$.

associated with the RSMC smoothing estimates is clearly realised for finite N. These figures also reaffirm the general improvement in estimator variance as N increases, in all dimensions tested, for both the ABC SMC and ABC RSMC particle smoothing procedures, as well as for the corresponding ABC forward smoothing procedures.

It is important to highlight that these observations are made where the SMC procedures use dynamic resampling. For the particle smoothing procedure, the use of dynamic resampling protects, to an extent, against the particle path degeneracy issue that is prevalent when resampling occurs too often, and so the variance of the resulting smoothing estimates is expected to be improved with its use. These empirical results therefore extend and complement the theoretical work of Del Moral

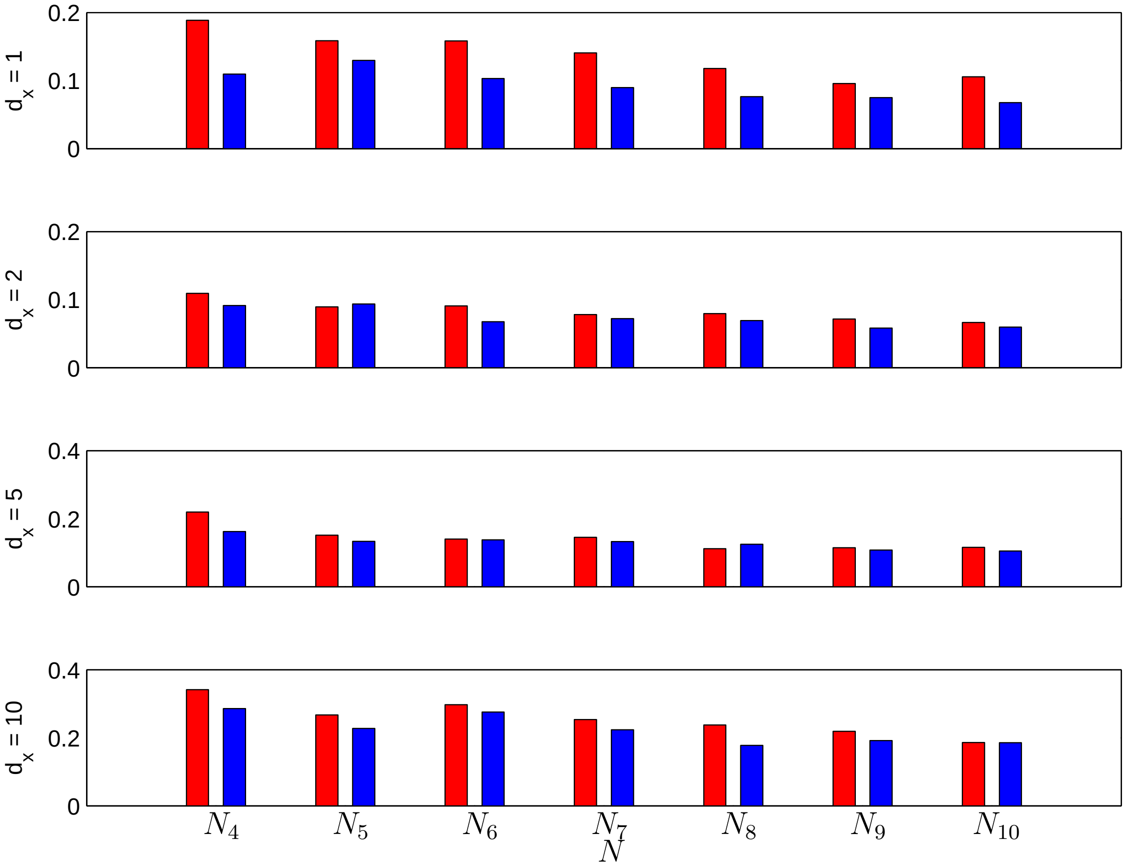


Figure 4.6: Standard errors of the ABC SMC and RSMC forward smoothing estimates of the mean smoothed state in the nonlinear SSM (3.13)-(3.14), calculated over a period of length $T = 100$. Each plot illustrates the standard errors of 50 independent estimates produced by the ABC forward smoothers using SMC (red) and RSMC (blue) kernels. The nonlinear SSM is specified by $\sigma_X^2 = 10$, $\sigma_Y^2 = 1$.

(2004), who considers the case where resampling is executed at every time step of the SMC procedure.

We now consider the accuracy of the RSMC smoothing estimates, measured in terms of the mean smoothing errors (4.18) obtained over all independent implementations of each procedure. The smoothing errors for the exact SMC and RSMC particle smoothers are presented in Figure 4.7 and the smoothing errors for the exact SMC and RSMC forward smoothers are presented in Figure 4.8. As before, the results corresponding to $N \leq N_3$ have been omitted to clarify the presentation of the results for $N \geq N_4$.

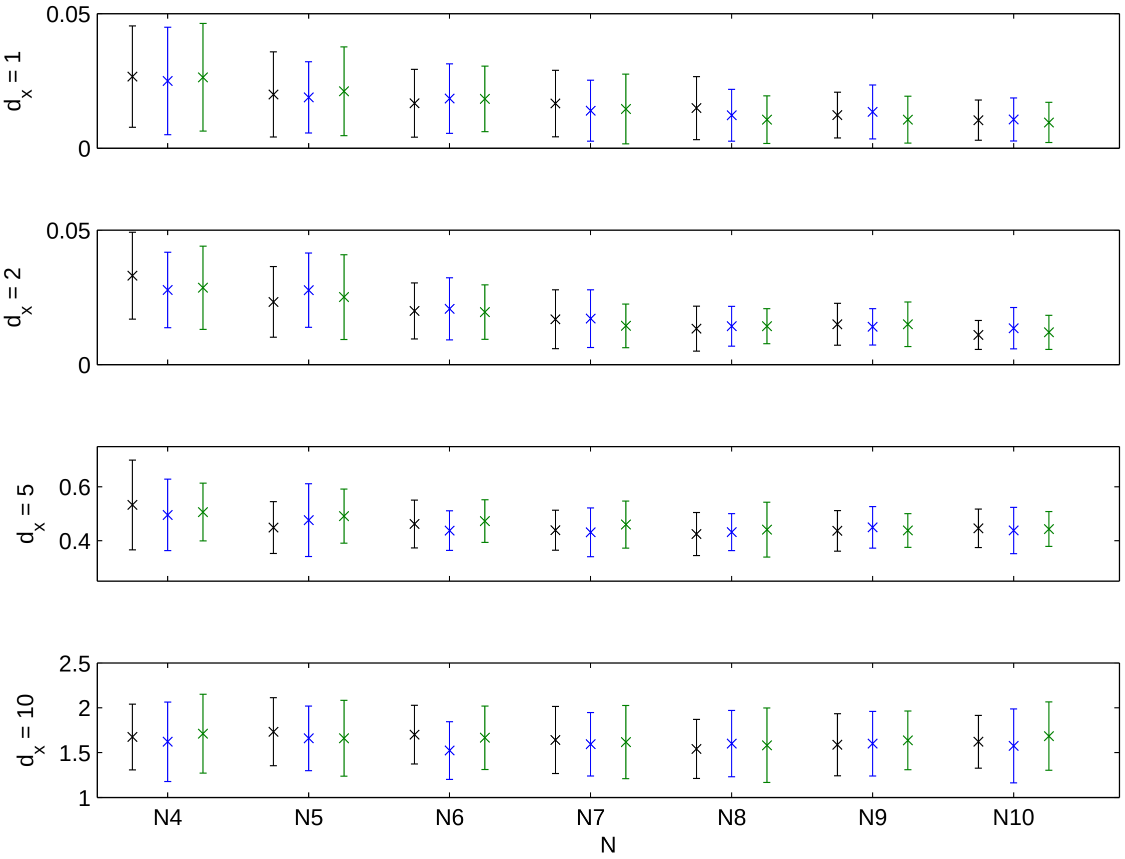


Figure 4.7: Smoothing errors for the exact SMC (black), ERSMC (blue) and TRSMC (green) particle smoothing estimates of the mean smoothed state in the nonlinear SSM (3.13)-(3.14), calculated over an interval of length $T = 100$. The crosses and errorbars indicate the mean and standard errors of the smoothing errors obtained over 50 independent implementations of each procedure. The nonlinear SSM is specified by $\sigma_X^2 = 10$, $\sigma_Y^2 = 1$.

From Figure 4.7, we note that, in terms of the mean smoothing errors obtained, the accuracy offered by the use of both the exact ERSMC and TRSMC particle smoothing procedures is extremely comparable to that of the exact SMC particle smoothing procedure, with neither a consistent improvement or deterioration in accuracy being evident in the comparison of the RSMC and SMC results, as N grows. The accuracy of the exact ERSMC and TRSMC forward smoothing procedures is also observed in Figure 4.8 to be largely comparable to that of the exact SMC forward smoothing procedures. It is further noted that, as would be expected, the significant deterioration in the accuracy of the procedure due to the weight and

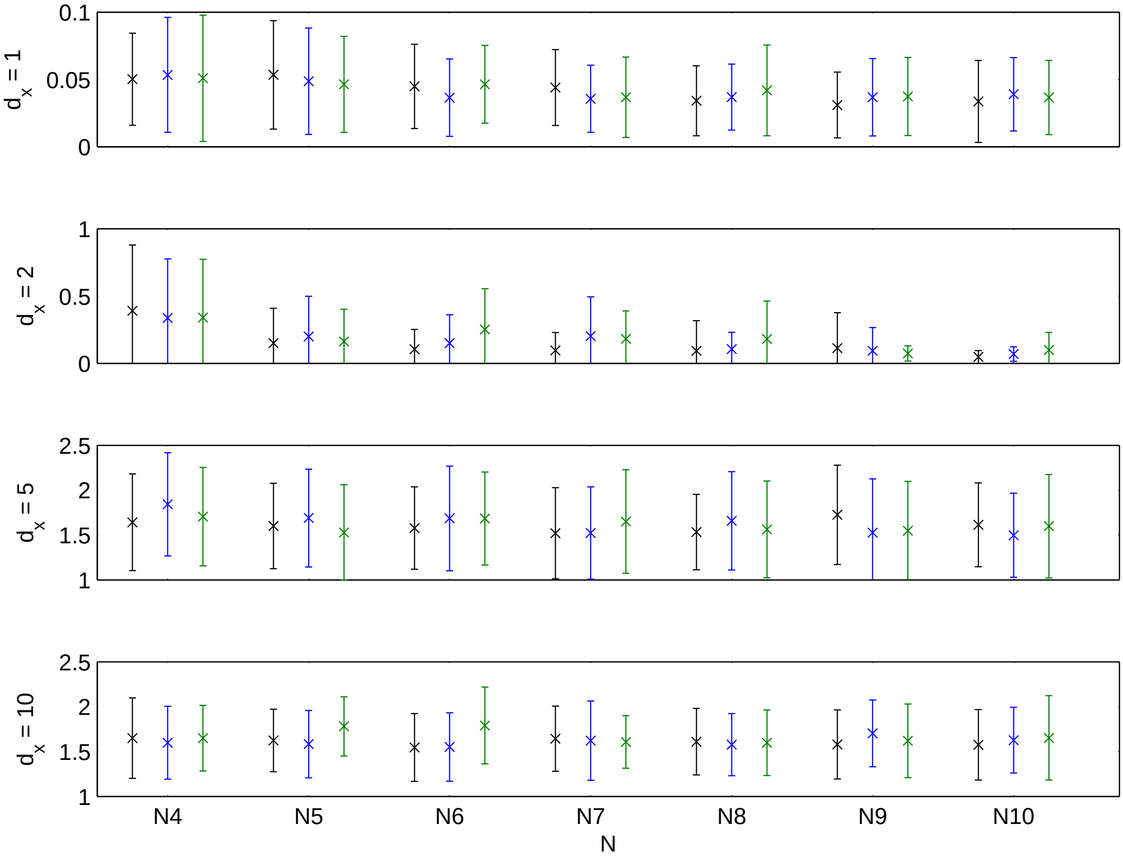


Figure 4.8: Smoothing errors for the exact SMC (black), ERSMC (blue) and TRSMC (green) forward smoothing estimates of the mean smoothed state in the nonlinear SSM (3.13)-(3.14), calculated over an interval of length $T = 100$. The crosses and errorbars indicate the mean and standard errors of the smoothing errors obtained over 50 independent implementations of each procedure. The nonlinear SSM is specified by $\sigma_X^2 = 10$, $\sigma_Y^2 = 1$.

particle degeneracy issues is also evident here for the implementation of the particle smoothers with $d = 10$ and for the implementations of the forward smoothers with $d \in \{5, 10\}$.

It has been seen that the use of RSMC has the potential to improve the variance of the exact smoothing estimates whilst maintaining a comparable accuracy in terms of the corresponding smoothing errors. In order to ascertain whether RSMC can be considered a viable alternative to SMC with dynamic resampling, we need to also consider the computational cost of each procedure. This is considered below, however we first consider the effect that the use of RSMC has on the accuracy of the

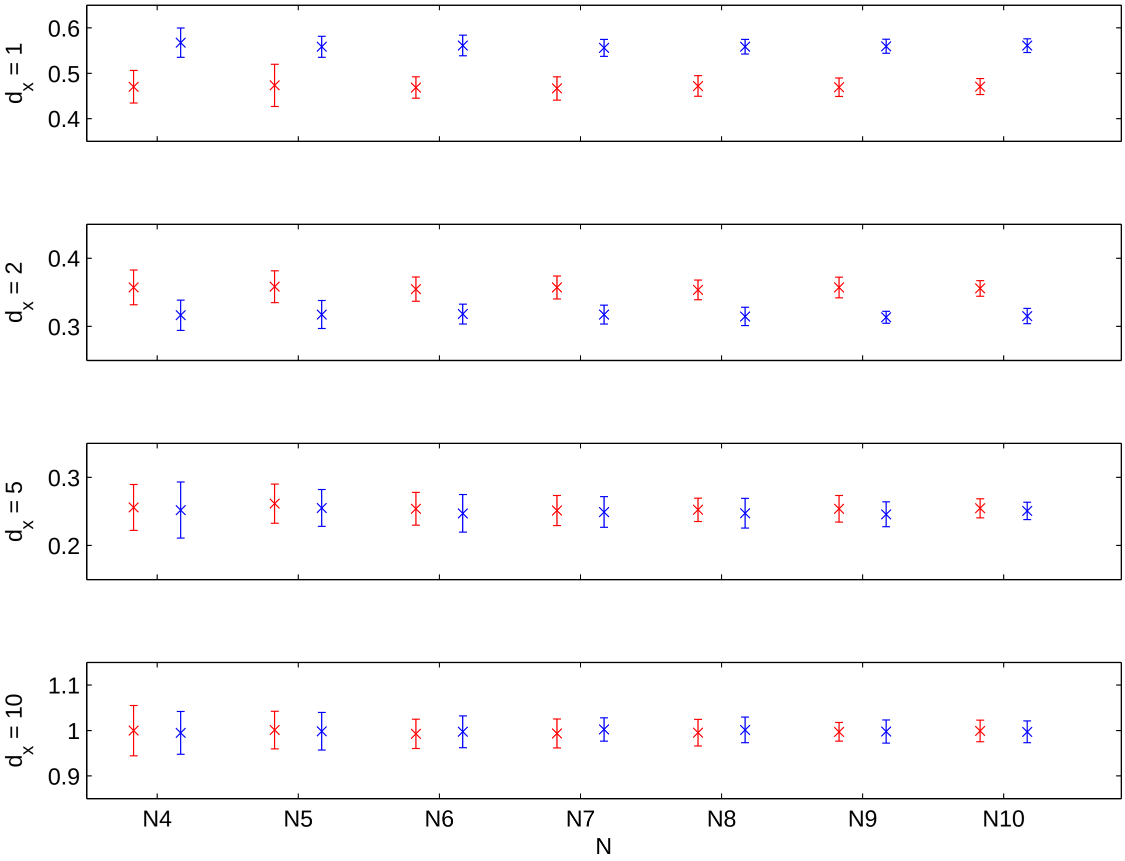


Figure 4.9: Smoothing errors for the ABC SMC (red) and ABC RSMC (blue) particle smoothing estimates of the mean smoothed state in the nonlinear SSM (3.13)-(3.14), calculated over an interval of length $T = 100$. The crosses and errorbars indicate the mean and standard errors of the smoothing errors obtained over 50 independent implementations of each procedure. The nonlinear SSM is specified by $\sigma_X^2 = 10$, $\sigma_Y^2 = 1$.

ABC smoothing procedures. Figure 4.9 presents the smoothing errors for the ABC particle smoothing procedures, and Figure 4.10 presents the errors for the ABC forward smoothing procedures; we consider the ABC forward smoothing procedures first.

From Figure 4.10, it is noted that the observed accuracy of the SMC and RSMC procedures for performing ABC forward smoothing are also very comparable, with the RSMC estimates even appearing to offer a marginal improvement over the SMC estimates in terms of mean smoothing error. Thus, combining with the clear reduction in estimator variance as N increases, observed in Figure 4.6, we need only

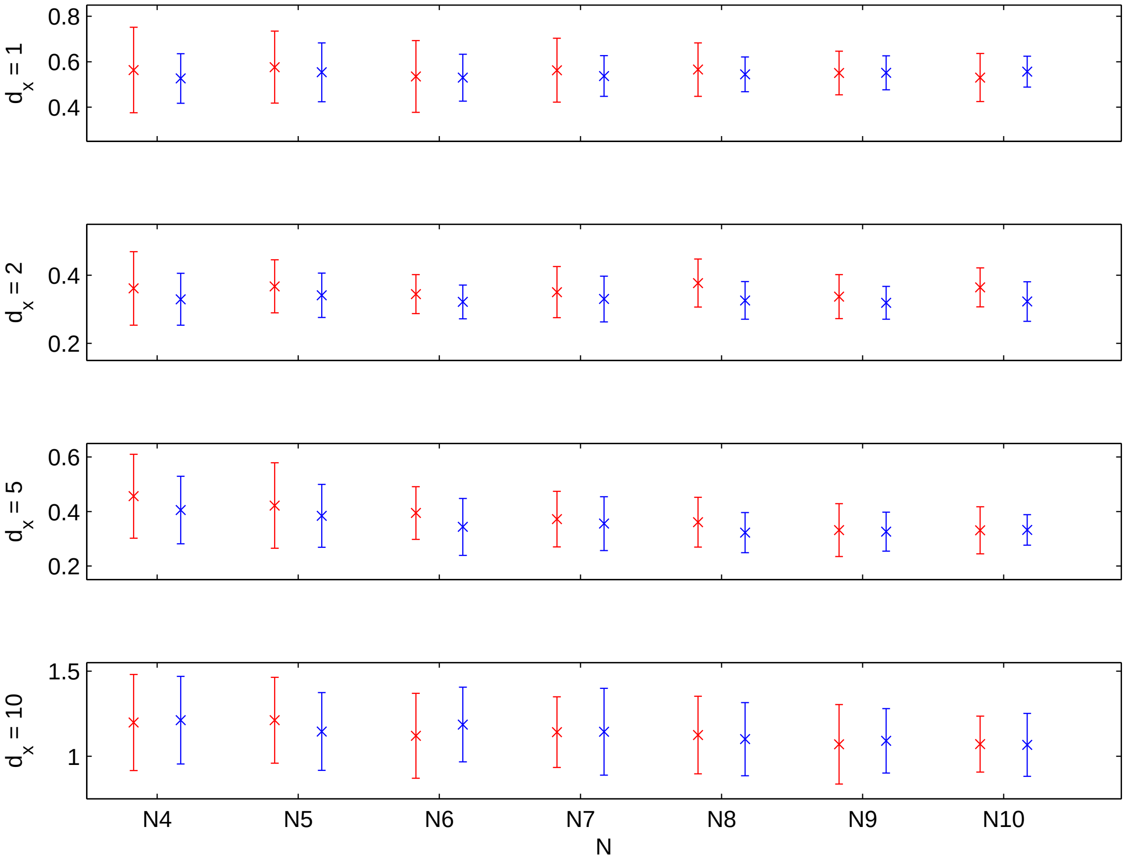


Figure 4.10: Smoothing errors for the ABC SMC (red) and ABC RSMC (blue) forward smoothing estimates of the mean smoothed state in the nonlinear SSM (3.13)-(3.14), calculated over an interval of length $T = 100$. The crosses and error-bars indicate the mean and standard errors of the smoothing errors obtained over 50 independent implementations of each procedure. The nonlinear SSM is specified by $\sigma_X^2 = 10$, $\sigma_Y^2 = 1$.

take into account any difference in computational performance of the alternative procedures before deciding whether the use of RSMC for ABC forward smoothing is preferable. Again, this is considered below.

Figure 4.9 offers different results to those observed in Figures 4.7, 4.8 and 4.10. When performing smoothing with respect to a one-dimensional state ($d = 1$) via an ABC approximation of the SSM, Figure 4.9 indicates that the use of an RSMC particle smoothing procedure significantly reduces the accuracy of the resulting estimates, in comparison to an SMC procedure with dynamic resampling. In direct contrast, when performing inference with respect to the ABC approximation of a smoothing distribution on two-dimensional space, the use of RSMC particle smoothing is observed to offer a significant and consistent improvement over SMC particle smoothing in the accuracy of the resulting estimates. Since this discrepancy in observed accuracies is observed for the RSMC and SMC particle smoothing procedures, yet there is a relative comparability in the corresponding forward smoothing procedures, we postulate that this can be attributed to excessive resampling in the RSMC procedure, leading to a path degeneracy issue.

To conclude the analysis of the smoothing procedures considered here, we consider the computational efficiency of the RSMC procedures in comparison with the corresponding SMC procedures. When performing exact smoothing via the ERSMC and TRSMC procedures, the processing times were observed to be extremely similar. This is to be expected, as the only difference in the computational cost of the two procedures arises from the calculation of the denominator of the acceptance probabilities. In practice, this value need only be calculated once per time step at little computational expense. As a result, the processing time for the exact TRSMC smoothing procedures are omitted, with the processing times for the exact ERSMC smoothing procedures serving to represent both the empirical and theoretical alternatives to RSMC.

The mean processing times for a single implementation of the SMC and RSMC particle smoothing procedures, applied to both the exact SSM (3.13)-(3.14) and its ABC approximation, are given in Tables 4.4-4.7. The processing times for the corresponding forward smoothing procedures are provided in Tables 4.8-4.11.

Recall that, for performing exact particle smoothing, the TRSMC procedure was observed to compete with the dynamic resampling SMC procedure in terms of estimator variance, whilst retaining a comparable accuracy as measured by the dimension-adjusted $\mathbb{L}_1$-errors of the smoothing estimates $\hat{\mathcal{S}}_T$. We also note from Tables 4.4 and 4.5 that the RSMC procedure for exact particle smoothing was observed to have a very comparable runtime, if marginally higher. Thus, we deduce

	$N = N_2$	$N = N_4$	$N = N_6$	$N = N_8$	$N = N_{10}$
d = 1	1.0108	4.4258	9.7826	17.1376	27.4512
d = 2	1.5606	6.1820	13.9160	24.3754	38.2332
d = 5	2.7808	11.2390	26.0350	45.5270	70.7568
d = 10	4.9566	20.2906	45.2510	81.8600	123.3148

Table 4.4: Mean processing times (in seconds), obtained over 50 independent applications of the SMC particle smoother to the exact nonlinear SSM.

	$N = N_2$	$N = N_4$	$N = N_6$	$N = N_8$	$N = N_{10}$
d = 1	1.2396	5.1656	11.6942	20.5058	31.3972
d = 2	1.6046	6.7168	15.0990	26.4816	40.7036
d = 5	2.9214	11.6446	26.6104	47.0046	71.9300
d = 10	5.0624	20.4168	46.7590	82.5646	126.6024

Table 4.5: Mean processing times (in seconds), obtained over 50 independent applications of the ERSMC particle smoother to the exact nonlinear SSM.

	$N = N_2$	$N = N_4$	$N = N_6$	$N = N_8$	$N = N_{10}$
d = 1	1.8548	7.3596	17.1774	29.5828	44.8888
d = 2	2.4576	9.7208	22.5078	39.7938	60.8722
d = 5	4.1114	16.9964	39.0258	69.4944	101.8816
d = 10	7.4704	29.9406	68.2582	120.8642	188.2086

Table 4.6: Mean processing times (in seconds), obtained over 50 independent applications of the SMC particle smoother to the ABC approximation of the SSM.

	$N = N_2$	$N = N_4$	$N = N_6$	$N = N_8$	$N = N_{10}$
d = 1	2.0338	8.4888	19.7714	35.7858	52.6326
d = 2	2.6566	11.0972	25.3702	46.1374	70.3412
d = 5	4.5342	18.0778	42.0240	72.5004	110.0044
d = 10	7.6290	30.4364	70.0202	124.0132	188.4936

Table 4.7: Mean processing times (in seconds), obtained over 50 independent applications of the RSMC particle smoother to the ABC approximation of the SSM.

that, for performing exact particle smoothing, the TRSMC procedure provides an alternative to the dynamic resampling based SMC procedure that has the potential to produce estimates with lower variance when using large particle systems. It is further remarked that, in the situation where the supremum in the denominator of the TRSMC acceptance probability is unavailable, the ERSMC procedure could provide an accessible alternative.

	$N = N_2$	$N = N_4$	$N = N_6$	$N = N_8$	$N = N_{10}$
d = 1	1.1232	4.3686	9.8096	17.2132	26.8020
d = 2	1.5998	6.2580	14.2900	25.0166	39.0418
d = 5	3.1618	12.4266	28.3480	49.9004	77.5224
d = 10	5.8152	24.8182	52.6138	91.6824	142.9032

Table 4.8: Mean processing times (in seconds), obtained over 50 independent applications of the SMC particle smoother to the exact SSM.

	$N = N_2$	$N = N_4$	$N = N_6$	$N = N_8$	$N = N_{10}$
d = 1	1.1754	4.5314	10.1178	17.9038	27.8734
d = 2	1.6818	6.5414	15.2508	25.9192	41.0166
d = 5	3.3050	12.6654	29.8238	50.5984	79.0068
d = 10	5.8114	23.0792	51.6424	91.7454	143.0206

Table 4.9: Mean processing times (in seconds), obtained over 50 independent applications of the ERSMC forward smoother to the exact SSM.

	$N = N_2$	$N = N_4$	$N = N_6$	$N = N_8$	$N = N_{10}$
d = 1	1.0736	4.1550	9.4308	16.8312	25.8450
d = 2	1.6112	6.2546	14.1610	24.8502	38.7978
d = 5	3.1330	12.3690	27.9514	50.5372	77.2082
d = 10	5.7852	22.8770	51.7906	91.8502	143.2292

Table 4.10: Mean processing times (in seconds), obtained over 50 independent applications of the SMC forward smoother to the ABC approximation of the SSM.

	$N = N_2$	$N = N_4$	$N = N_6$	$N = N_8$	$N = N_{10}$
d = 1	1.0370	4.0390	9.0972	16.0752	25.2446
d = 2	1.5590	5.9966	13.5144	23.8274	37.0882
d = 5	2.9958	11.8926	26.5716	47.0250	73.5766
d = 10	5.5552	21.4488	48.2644	85.5338	133.7458

Table 4.11: Mean processing times (in seconds), obtained over 50 independent applications of the RSMC forward smoother to the ABC approximation of the SSM.

As can be observed in Tables 4.8 and 4.9, the observed runtimes for the exact SMC forward smoother and its RSMC alternatives were extremely comparable. This is to be expected, as the computational complexity of the procedures is dominated by the estimation steps within, which remain unchanged when incorporating the RSMC methodology. As a result of the $O(N^2)$ computational cost, when performing exact forward smoothing via RSMC, a consistent observable improvement in estimator variance is likely to require a particle system size N that renders the computational expense of the algorithm prohibitive. Based upon the results presented here, the use of RSMC for performing exact forward smoothing cannot be expected to produce superior estimates to SMC with dynamic resampling within a reasonable computational budget, although comparable estimator accuracy may be obtained when SMC with dynamic resampling is undesirable.

We consider now the viability of the RSMC procedure for performing ABC particle smoothing. In the one-dimensional case, the improvement in estimator variance is clear, however this is made less relevant by the significant increase in the observed smoothing errors. In addition, the increase in runtimes observed in Tables 4.6 and 4.7, whilst small in absolute terms, represents a significant relative increase. Therefore, the use of RSMC for one-dimensional ABC particle smoothing would not be encouraged, given the above results. For $d \in \{2, 5\}$, the decrease in estimator variance is still evident. Further, the smoothing errors in Figure 4.9 indicates a significant and marginal increase in the accuracy of the estimates for $d = 2$ and $d = 5$, respectively. Although these advantages come with an equally significant relative increase in the observed runtimes, the improved accuracy could encourage the use of RSMC in performing ABC particle smoothing. In high dimensions, we note that the improvement in estimator variance from the incorporation of RSMC is still clear, as observed in Figure 4.5, and that the smoothing errors displayed in Figure 4.9 indicate comparable estimator accuracy. Therefore, RSMC can also be considered as a viable alternative to SMC with dynamic resampling when performing high-dimensional ABC particle smoothing.

Finally, we consider the use of RSMC for performing ABC forward smoothing. The improvement in estimator variance is evident in all dimensions $d \in \{1, 2, 5, 10\}$, and this is complemented by the extremely comparable smoothing errors presented in Figure 4.10. In addition, the observed runtimes for the ABC forward smoothing procedure were consistently lower when using the RSMC method. These results are extremely encouraging, and they suggest that the use of RSMC would not only be a viable alternative, but it would be preferable to using SMC with dynamic resampling when using forward smoothing to perform inference with respect to the ABC approximation of the smoothing distribution.

4.7.3 Results and Analysis - PMMH Procedures

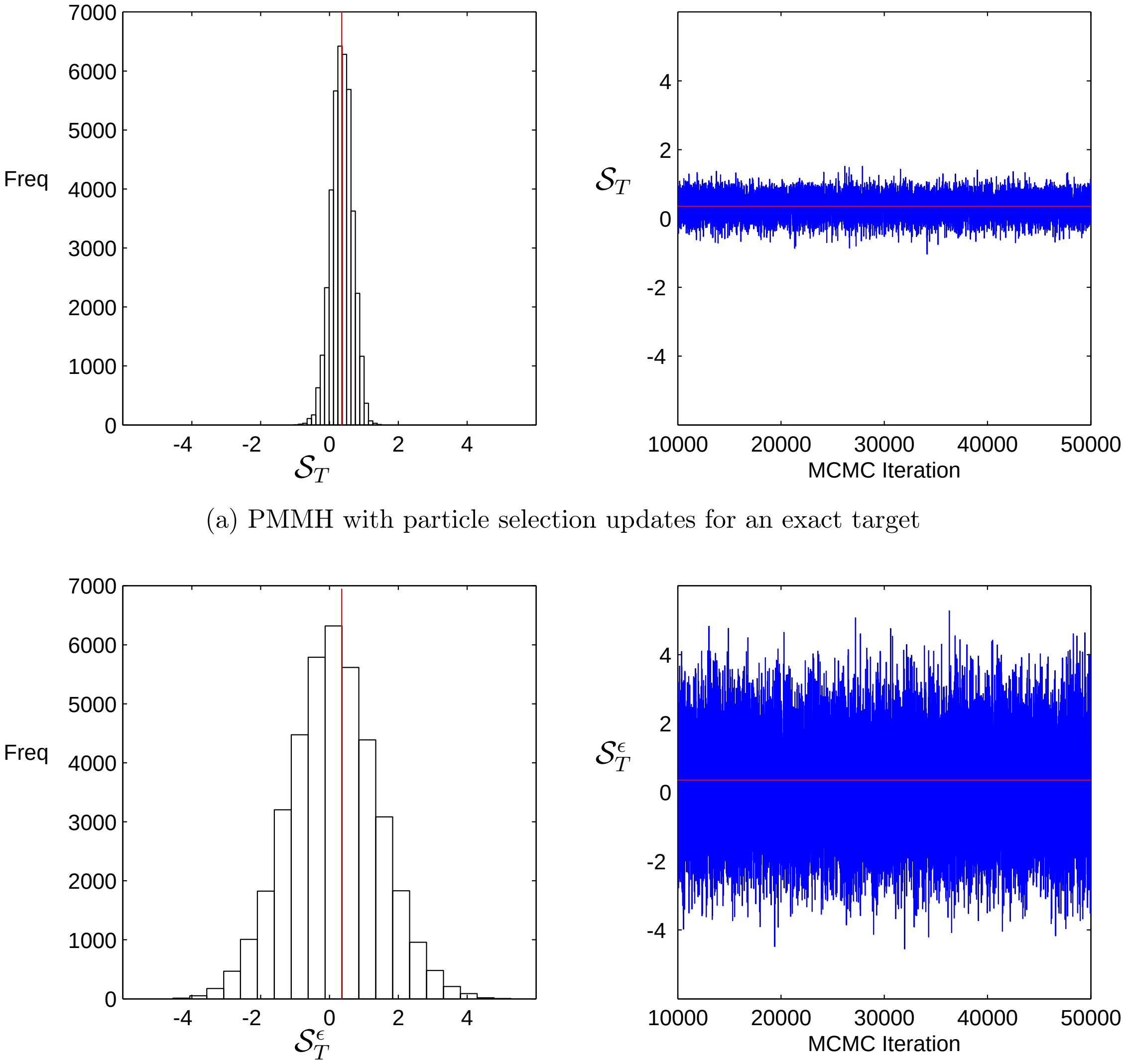
We consider here the implementation of the various PMMH procedures proposed in Section 4.6.

PMMH with Particle Selection

We consider here the use of PMMH with particle selection updates, for estimating $\mathcal{S}_T$ as specified by (4.2)-(4.4), and its ABC approximation $\mathcal{S}_T^\epsilon$. The PMMH setup used here is similar to that considered in Section 3.1 of Andrieu et al. (2010): we use $N = 2000$ particles for the SMC update schemes across $M = 50,000$ MH iterations, discarding a burn-in sample of $10,000$ iterations. If a particular SMC proposal scheme resulted in collapse, the seed for the pseudo-random number generator was reset and the proposal scheme was repeated. Each of the PMMH procedures was implemented 10 times to observe consistency of results.

Figure 4.11a presents the histogram and sample traces for the post-burn-in samples of $\mathcal{S}_T$, obtained from a single implementation of the exact particle-selection PMMH procedure, whilst 4.11b provides the corresponding plots for $\mathcal{S}_T^\epsilon$ obtained using a single implementation of the ABC particle-selection PMMH procedure.

We immediately note from Figure 4.11 that the variance of the PMMH sample is significantly higher for the estimate of the ABC smoothed additive functional than for the exact case; this can be linked to both an decrease in the SMC resampling rate and an increase in the MH acceptance rate when adopting an ABC approximation. The SMC resampling rates were obtained from the SMC proposal procedures corresponding to the 40,000 post-burn-in MH iterations; the average resampling



(a) PMMH with particle selection updates for an exact target

(b) PMMH with particle selection updates for an ABC target

Figure 4.11: Bar and trace plots showing the distribution and path of the post-burn-in samples obtained using an exact particle selection PMMH procedure (a) and an ABC particle selection PMMH procedure (b). These samples are used to estimate the mean smoothed state in the nonlinear SSM (3.13)-(3.14), over an interval of length $T = 100$. The nonlinear SSM is specified by $\sigma_X^2 = 10$, $\sigma_Y^2 = 1$.

rates for the exact and ABC procedures were obtained by averaging, first over the 40,000 post-burn-in iterations, and then again over 10 independent implementations of each PMMH procedure. The MH acceptance rate for each procedure was also averaged over 10 independent implementations. The average SMC resampling rate was observed to drop from 81.39% to 20.97% when adopting the ABC approximation, whereas the MH acceptance rate, also averaged over 10 independent PMMH implementations, increased from 39.46% to 62.59%.

A decreased resampling rate in the proposal schemes results in a more diverse genealogical tree representation of the smoothing distribution, from which the candidate value of $\mathcal{S}_T$ is selected for the MH procedure. Thus the sample of MH candidate values is observed to be more diverse. Given this increased diversity in the candidates, the increase in the MH acceptance rate clearly encourages greater diversity in the accepted sample. It is also remarked that, as well as the decrease in SMC resampling rate, the increase in MH acceptance rate can also be attributed to the decrease in SMC weight degeneracy resulting from the use of ABC. The increased diversity in the importance weights at each time step results in less variable estimates $\hat{\pi}_\theta(y_{1:T})$ of the marginal likelihood at each MCMC iteration, leading to more consistent ratios of consecutive marginal likelihood estimates, which in turn lead to higher MH acceptance rates.

The comparative accuracy of the particle-selection PMMH procedure and its ABC alternative is somewhat masked by the scale of the axes in Figure 4.11, chosen to emphasise the increase in sample variance. The PMMH estimates and corresponding MH sample standard errors are therefore presented in Figure 4.12.

It is noted that the ABC bias is clearly evident in the accuracy of the estimates provided by the ABC particle-selection PMMH procedure. Combining this with the significant increase in the sample standard errors that results from using an ABC approximation, we conclude that the ABC particle-selection PMMH procedure does not provide competitive estimates of one-dimensional smoothed additive functionals when compared against the corresponding exact procedure.

Particle Selection Updates vs Forward Smoothing Updates

We now consider the use of forward smoothing PMMH, as described in Section 4.6, for estimating both the exact smoothed additive functional $\mathcal{S}_T$ and its ABC approximation $\mathcal{S}_T^\epsilon$, and we seek to compare the performance of these procedures

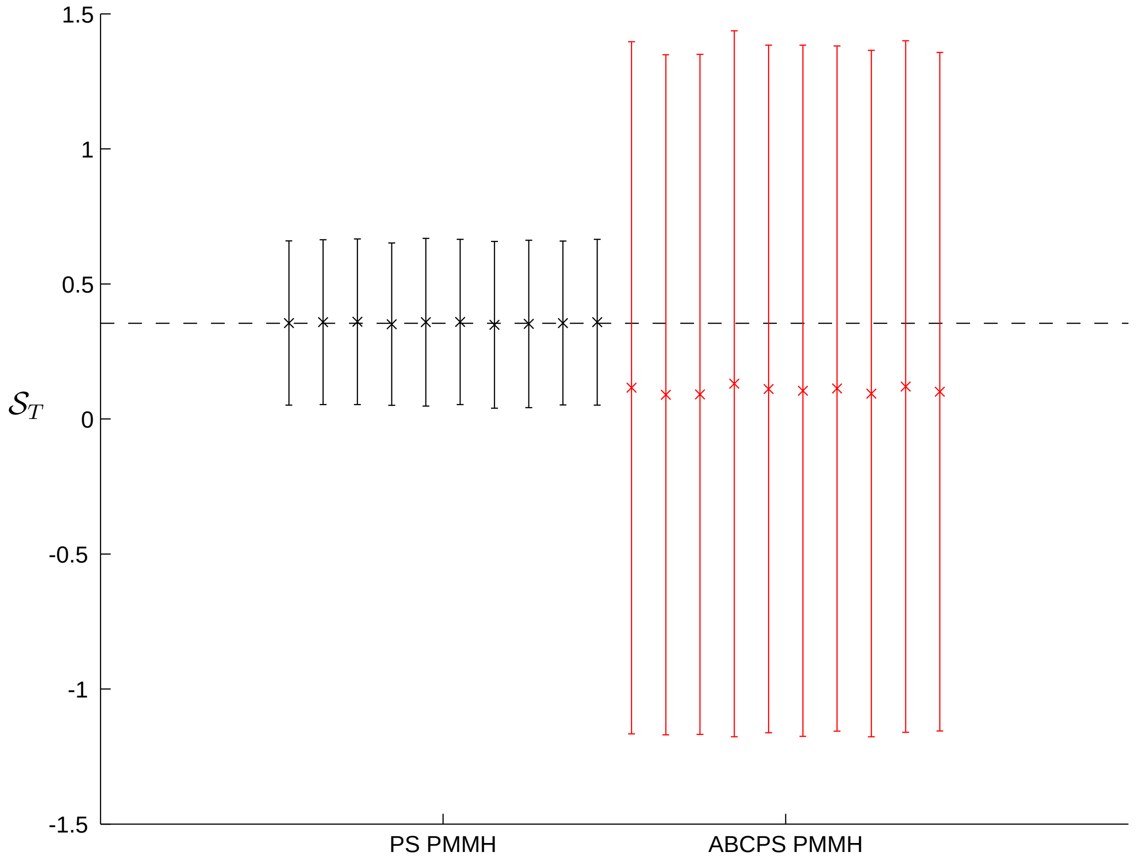


Figure 4.12: PMMH estimates of the mean smoothed state in the nonlinear SSM (3.13)-(3.14), calculated using the exact (black) and ABC (red) particle selection PMMH procedures to perform estimation over an interval of length $T = 100$. The crosses and errorbars correspond to the estimates and MH sample standard errors, respectively and the errors for 10 independent implementations of each procedure are shown. The dashed horizontal line gives the 'true' value of $\mathcal{S}_T$. The nonlinear SSM is specified by $\sigma_X^2 = 10$, $\sigma_Y^2 = 1$.

against the corresponding particle selection PMMH procedures. As detailed at the start of this section, the aim here will be to compare forward smoothing and particle-selection PMMH procedures with comparable algorithmic runtimes. Therefore, in order to observe any potential improvement with increasing particle system size, the procedures were implemented using $N \in \{N_1, \ldots, N_5\}$ particles, as detailed in Table 4.1. As before, all PMMH procedures were implemented using 50,000 MH iterations, with a burn-in period of 10,000 iterations.

The smoothing errors corresponding to the estimates provided by each of the

	$N = N_1$	$N = N_2$	$N = N_3$	$N = N_4$	$N = N_5$
PS PMMH	0.0053 (0.3054)	0.0037 (0.3073)	0.0002 (0.3088)	0.0003 (0.3055)	0.0003 (0.3068)
ABC PS PMMH	0.2460 (0.333)	0.2401 (0.3121)	0.2171 (0.3005)	0.2559 (0.3206)	0.2451 (0.344)
FS PMMH	0.0407 (0.0694)	0.0132 (0.0944)	0.0026 (0.0871)	0.0023 (0.0711)	0.0029 (0.0686)
ABC FS PMMH	0.0214 (0.3219)	0.0423 (0.2207)	0.0540 (0.1796)	0.0339 (0.1605)	0.0105 (0.1491)

Table 4.12: Smoothing errors for the PMMH estimates of the smoothed additive functional $\mathcal{S}_T$ and its ABC approximation $\mathcal{S}_T^\epsilon$, obtained using PMMH procedures with particle selection and forward smoothing updates. The corresponding PMMH sample standard errors are provided in brackets. Both the smoothing errors and the PMMH sample standard errors were calculated using the post-burn-in PMMH sample.

considered PMMH procedures are presented in Table 4.12, with the standard errors of the corresponding MH sample provided in brackets. The PMMH estimates and their corresponding MH sample standard errors were calculated post-burn-in, and are also depicted in Figure 4.13, where the ‘true’ value of $\mathcal{S}_T$ is given by the horizontal dashed line.

We immediately note the significant improvement in MH sample variance that the use of forward smoothing PMMH offers, when estimating both the smoothed additive functional $\mathcal{S}_T$ and its ABC estimate $\mathcal{S}_T^\epsilon$. This is to be expected, as the proposed values of the smoothed additive functional at each MH iteration will be more accurate, in general, than those generated by the particle selection procedure.

Of added interest here is the behaviour of these estimates as N increases. Figure 4.13 suggests that the MH sample variance corresponding to both the exact forward smoothing PMMH procedure and its ABC alternative, decreases as the particle system size increases. In contrast, Figure 4.13 also suggests that the sample variance for both the exact and ABC particle selection PMMH procedures is relatively insensitive to increasing N. These observations are corroborated by the values reported

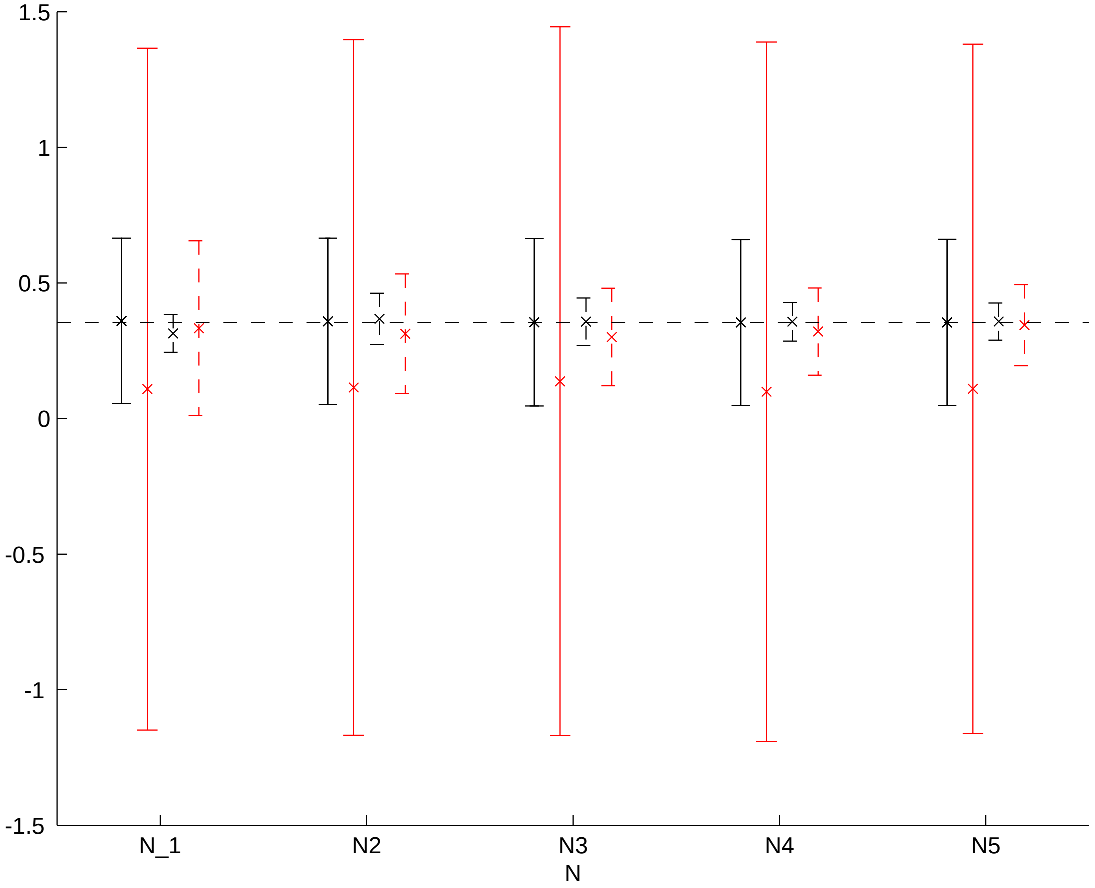


Figure 4.13: PMMH estimates of the mean smoothed state in the nonlinear SSM (3.13)-(3.14), over an interval of length $T = 100$. The estimates were obtained using the exact particle selection PMMH procedure (black, solid lines), the exact forward smoothing PMMH procedure (black, dashed lines) and their ABC alternatives (red, solid and dashed lines, respectively), for a variety of particle system sizes N. The crosses and errorbars correspond to the estimates and MH sample standard errors, respectively. The dashed horizontal line gives the 'true' value of $\mathcal{S}_T$. The nonlinear SSM is specified by $\sigma_X^2 = 10$, $\sigma_Y^2 = 1$.

in Table 4.12.

The significance of the contrasting behaviour of the PMMH procedures with particle selection and forward smoothing updates is amplified by the fact that the number of particles used for the particle selection PMMH procedures, and hence the increase in the number of particles, is much greater than that for the forward smoothing PMMH procedures. These observations indicate that the variance in the CLT for the particle smoothing PMMH estimator is dominated by the effect of some-

	$N = N_1$	$N = N_2$	$N = N_3$	$N = N_4$	$N = N_5$
PS PMMH	12,869	51,979	121,362	207,421	324,681
ABC PS PMMH	15,071	56,556	129,825	232,346	361,098
FS PMMH	13,708	53,683	122,379	214,616	335,822
ABC FS PMMH	22,061	66,417	136,193	225,832	347,568

Table 4.13: Processing times, in seconds, for a single implementation of each PMMH procedure. Each procedure was executed for $M = 50,000$ MH iterations.

thing other than N. Taking into consideration the fact that the asymptotic variance of the SMC particle smoothing estimator is at least $O(T^2)$, and thus dominated by the length of the observation period (as discussed in Section 4.3.1), we postulate that this is the case for the particle selection PMMH estimator also. We further recall that the asymptotic variance of the SMC forward smoothing estimator is linear in the period T and, as a result, dominated by its inverse quadratic dependence on N; we postulate that the asymptotic variance of the forward smoothing PMMH estimators also follow this dependence structure.

From Figure 4.13, it is once again clear to see that the deterministic bias associated with the ABC approximation is evident in the ABC particle selection PMMH estimates. It is extremely interesting to note, however, that this bias is not as evident in the ABC forward smoothing PMMH estimates. Consulting Table 4.12, and comparing the smoothing errors for the exact and ABC forward smoothing PMMH procedures, we see that, whilst the errors for the exact procedure clearly decrease as N increases, the errors for the ABC procedure do not. This would suggest that, whilst it is not as evident as in the particle selection PMMH estimates, the ABC bias is still present in the smoothing error associated with the forward smoothing PMMH estimator. Based upon these observations, we tentatively conclude that the error of the ABC forward smoothing PMMH estimator is dominated by the simulation error.

Table 4.13 displays the CPU times for each of the PMMH procedures considered.

Recall that the values of N for which the particle selection PMMH procedures were implemented (see Table 4.1), were chosen such that the particle selection updates and the forward smoothing updates were executed with comparable computational cost. From Table 4.13, we can see that this comparability holds to a certain extent, with the greatest difference in runtimes being noticed for the ABC forward smoothing PMMH procedures that used low values of N. This can be attributed to the ABC weight degeneracy phenomenon; for a number of iterations of the ABC forward smoothing PMMH procedure, the SMC approximation of the joint smoothing distribution has collapsed due to all of the particles being allocated a weight of 0. Whenever this occurs, the update scheme is repeated (using a different seed for the pseudo-random number generator). The proportion of iterations at which the ABC forward smoothing procedure collapsed was observed to decrease as N increased, as would be expected; using $N = 100$ particles, the ABC forward smoothing update procedure had to be executed 85,543 times to generate an MH sample of length 50,000, whereas this number was reduced to 54,120 when using $N = 500$ particles.

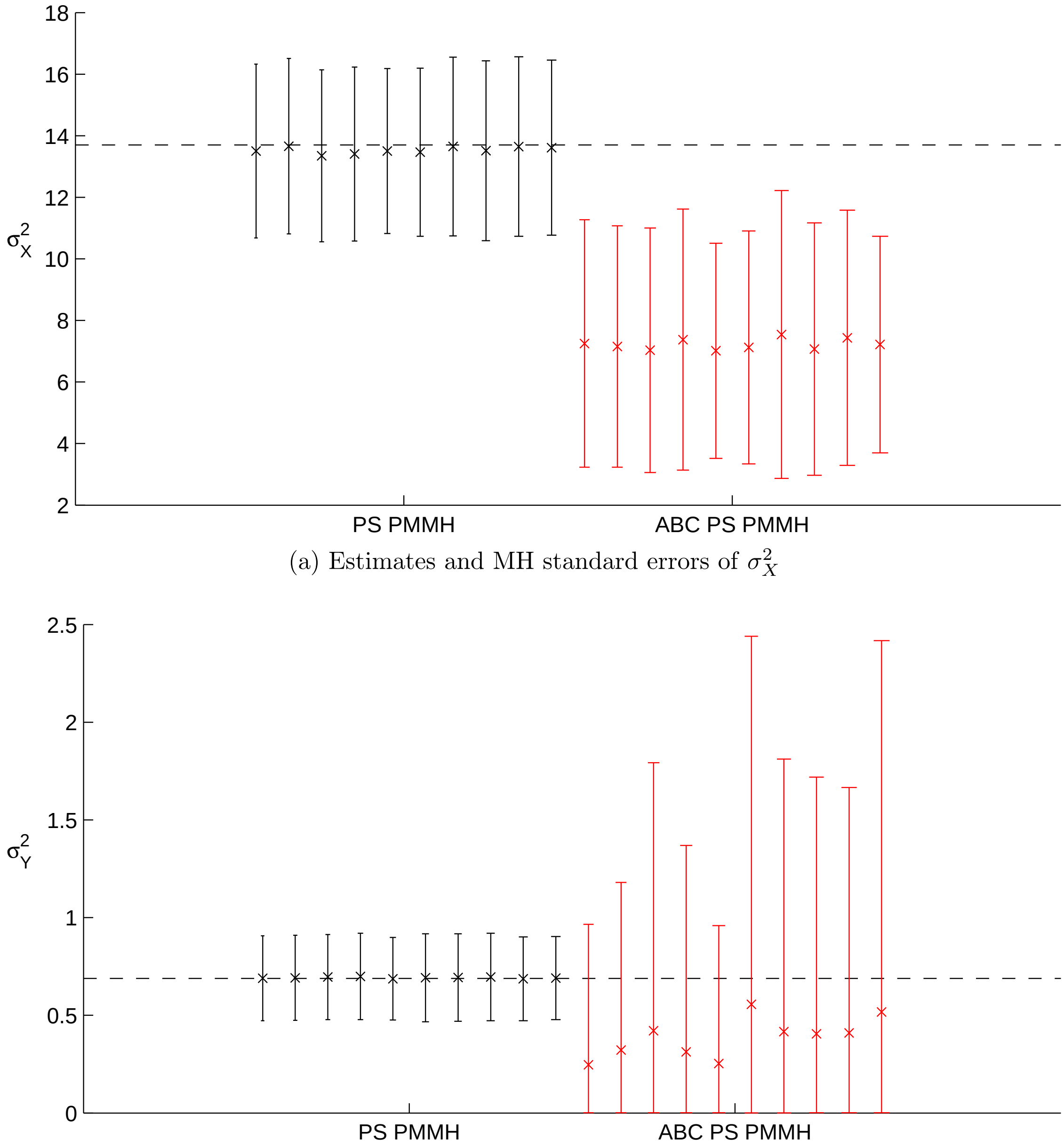
Collapse of the SMC approximation of the smoothing distribution was observed in some of the other implementations of the PMMH procedures, however the incidence rate was extremely low in comparison to that observed for the ABC forward smoothing updates, and the effect on the corresponding runtimes was negligible.

This increased computational budget due to the ABC weight degeneracy issue is obviously a drawback for the ABC forward smoothing procedure, and should be taken into account in practice when choosing the number of particles with which to implement the ABC forward smoothing PMMH procedure. Nevertheless, we stress that the invariance of the particle selection PMMH estimator with respect to N allows us to conclude that, given an N-particle implementation of particle selection PMMH, one could find a particle system size with which to implement the ABC forward smoothing PMMH procedure with the same number of MH iterations, such that the computational expense will be similar and the variance of the resulting MH sample will be significantly smaller.

We now consider the use of these PMMH methods for batch parameter estimation. We consider first the effect on parameter estimation that the use of the ABC approximation has, and we subsequently consider the effect of using forward smoothing updates. We return to the 10 independent 2000-particle implementations particle selection PMMH procedure and its ABC alternative; Figure 4.14 presents the parameter estimates obtained from the resulting post-burn-in MH parameter samples, i.e. the parameter values used in the successful SMC update schemes.

For batch parameter estimation, one typically requires a much larger observation period than has been used here; as a result, the estimates are not expected to be very accurate with respect to the true parameter values used to generate the hidden state and observed data. For comparative purposes, we use the parameter estimates generated by the post-burn-in MH parameter samples provided by the 200,000-particle implementation of the particle selection PMMH procedure, used above to generate the 'true' smoothed additive functional S_T. From Figure 4.14a, we note that the estimates of the hidden state noise parameter σ_X^2, generated using the ABC approximation of the SSM, are relatively stable across independent implementations, although they are ultimately inaccurate. The corresponding estimates for the noise of the observed data, given in Figure 4.14b are both inaccurate and inconsistent over independent implementations. It is clear from these results that the use of an ABC approximation is not suitable for performing batch inference with respect to static parameters associated with the nonlinear SSM (3.13)-(3.14).

We consider now the use of forward smoothing PMMH for performing batch parameter estimation, and in Figure 4.15, we present the estimates generated by implementations of the particle selection and forward smoothing PMMH procedures with particle system sizes $N \in \{N_1, \ldots, N_5\}$. It is immediately clear from Figure 4.15 that the use of forward-smoothing PMMH for batch parameter estimation provides results that are, in general, extremely comparable to those produced by the

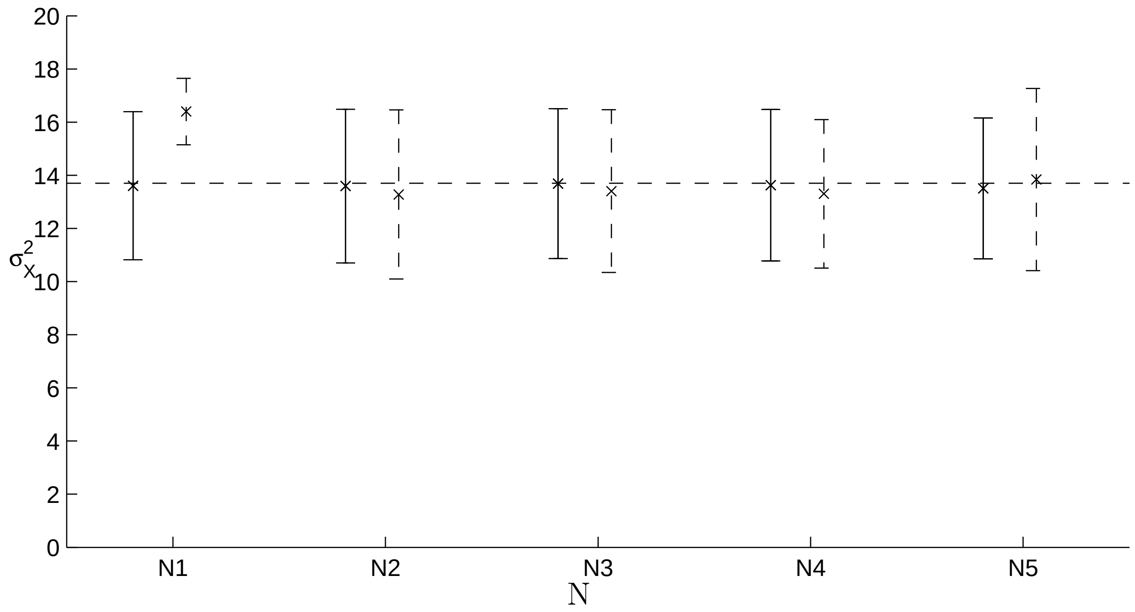


(a) Estimates and MH standard errors of σ_X^2

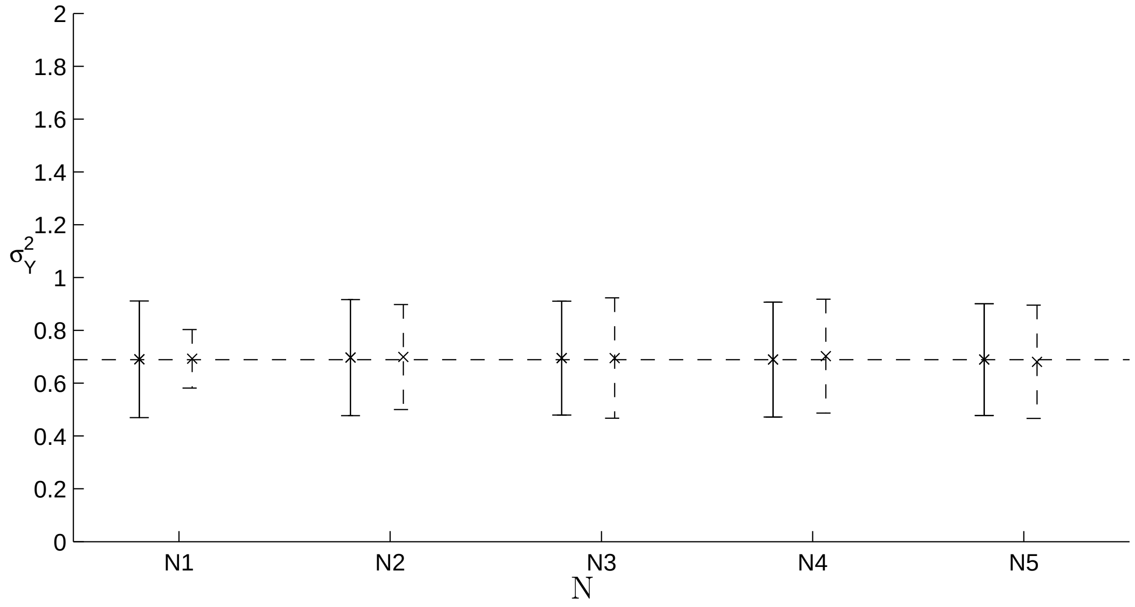
(b) Estimates and MH standard errors of σ_Y^2

Figure 4.14: Parameter estimates obtained from the post-burn-in parameter samples provided by 10 independent implementations of each of the exact particle selection PMMH procedure (black) and the ABC particle selection PMMH procedure (red). For all PMMH procedures considered here, $N = 2000$. The horizontal dashed lines correspond to the parameter estimates provided by a 200,000-particle implementation of the particle selection PMMH procedure, which has been used here as a benchmark.

particle selection PMMH procedure. Given the advantages offered by this approach with respect to the estimation of smoothed additive functionals, as mentioned above,



(a) Estimates and MH standard errors of σ_X^2



(b) Estimates and MH standard errors of σ_Y^2

Figure 4.15: Parameter estimates obtained from the post-burn-in parameter samples provided by the exact particle selection PMMH (solid error bars) and exact forward smoothing PMMH (dashed error bars) procedures. Each procedure was implemented with $N \in \{N_1, \dots, N_5\}$ particles. The horizontal dashed lines correspond to the parameter estimates provided by a 200,000-particle implementation of the particle selection PMMH procedure, which has been used here as a benchmark.

this adds weight to the conclusion that forward smoothing PMMH is viable, and can prove to be a preferable alternative to the accepted state-of-the-art, particle selection PMMH.

4.8 Summary of Results

In this chapter, we have considered the problem of performing inference with respect to the ABC approximation of the joint smoothing distribution. In particular, the following contributions have been made:

- We have provided a theoretical analysis of the deterministic bias associated with the estimation of ABC approximations of smoothed additive functionals. This bias was shown to grow only linearly with the length of the observation period T, and also linearly with the ABC tolerance parameter ϵ.

- We have introduced procedures for performing estimation of these ABC smoothed additive functionals via particle smoothing and forward smoothing, and we have analysed the behaviour of the SMC estimators that these procedures produce.

- We have considered the use of RSMC methodology for reducing the variance of the particle smoothing and forward smoothing estimators, both independently of, and in conjunction with the ABC approximation.

- We have also introduced a number of PMMH procedures for targeting the exact and ABC approximations of a smoothed additive functional.

All of the procedures that have been introduced in this chapter have been numerically analysed in Section 4.7. The principal conclusions of these numerical studies can be summarised as follows:

- For both ABC particle smoothing and ABC forward smoothing, the accuracy of the estimators is dominated by the deterministic ABC bias.

- For hidden states of high dimension, however, ABC particle smoothing can be more accurate than exact particle smoothing, and ABC forward smoothing can be more accurate than exact forward smoothing.

- The use of the RSMC methodology was shown to consistently reduce the variance in both the ABC particle smoothing estimators and the ABC forward smoothing estimators, as well as providing comparable accuracy in terms of the $\mathbb{L}_1$-errors of the corresponding estimates.

- For exact particle smoothing, the RSMC methodology was not observed to offer any consistent improvement in estimator variance, although the observed variances were comparable to those obtained when using dynamic resampling SMC. For exact forward smoothing, a consistent improvement in estimator variance cannot be expected within a reasonable comptutational budget.

- When using particle selection PMMH to estimate the mean smoothed state over the observation interval, the use of the ABC approximation significantly increases the variance of the resulting post-burn-in MH sample, and induces a clear bias in the final estimator.

- The use of exact forward smoothing update schemes within a PMMH procedure significantly reduces the variance in the resulting post-burn-in MH sample and provides final estimates of the mean smoothed state that are of comparable accuracy, when compared with those produced by an exact particle selection PMMH procedure of comparable computational expense.

- The use of ABC forward smoothing update schemes within a PMMH procedure provides a post-burn-in sample that has larger variance than that of the exact forward smoothing PMMH procedure, but a smaller variance than that of an exact particle selection PMMH procedure of comparable computational expense. In addition, the ABC forward smoothing PMMH procedure also induces an ABC bias in the resulting estimator, although one that is significantly smaller than that of the ABC particle selection PMMH estimator.

- For batch parameter estimation via PMMH, the use of the ABC approximation was observed here to render the parameter estimates inaccurate and unreliable.
- Comparable parameter estimates were obtained when using exact particle selection PMMH and exact forward smoothing PMMH procedures of similar computational expense.

Chapter 5

SMC Samplers for a Class of Point Process Models

5.1 Introduction

This chapter considers the family of SSMs in which both the observed and latent processes are point processes (PPs). SSMs in which the latent process is a PP are referred to in the literature as partially observed PPs, and provide a rich class of models for describing real data; they have previously been used for stochastic volatility modelling (Barndorff-Nielsen and Shephard, 2001) in finance, descriptions of queueing data (Fearnhead, 2004) in operational research, important seismological models (Daley and Vere-Jones, 2007) and for applications in nuclear physics (Snyder and Miller, 1998). This chapter will focus on the use of partially observed PPs in financial modelling, in particular for ultra-high frequency (UHF) data.

Time-inhomogeneous PP models in which the intensity is itself a stochastic process, although not necessarily a PP itself, are referred to in the literature as doubly stochastic PPs (DSPPs) and were first introduced by Cox (1955). In a dynamic setting, DSPP studies date back to at least Snyder (1972), who considers the time evolution of a set of statistics describing the posterior distribution of a latent Markov process, through the observation of a DSPP. Interest here will be on performing inference with respect to the latent process, and this will be reflected in the terminology used here; the model is treated as being a special class of partially observed PP models, rather than a special class of DSPP models.

Fitting these complex models in a Bayesian context requires the use of MCMC or SMC methods. The main developments using MCMC in this field include the work of Green (1995), Roberts et al. (2004) and Centanni and Minozzo (2006a,b). Relevant developments in SMC include the work of Del Moral et al. (2006, 2007) and applications of SMC methods to partially observed PPs can be found in Rydberg and Shephard (2000), Doucet et al. (2006) and Whiteley et al. (2011).

One of the first works applying computational methods to PP models, was Rydberg and Shephard (2000). They focus upon a Cox model where the unobserved PP parameterizes the intensity of the observations. Rydberg and Shephard (2000) used the auxiliary particle filter (Pitt and Shephard, 1999) to simulate from the posterior density of the intensity at a given time point. This was superseded by Centanni and Minozzo (2006a,b), which allows one to infer the intensity at any given time, up to the current observation.

Informally, the problem of interest is as follows. A process is observed discretely upon a given time-interval $[0, T]$. The objective is to draw inference at time-points $t_0 = 0 < t_1 < \cdots < t_{\widetilde{m}} < T = t_{\widetilde{m}+1}$, on the unobserved marked PP $(k_{t_n}, \phi_{1:k_{t_n}}, \zeta_{1:k_{t_n}})$, where $\phi_{1:k_{t_n}} = (\phi_1, \ldots, \phi_{k_{t_n}})$ are the ordered event times (constrained to $[0, t_n]$), with k_{t_n} the number of events up to t_n, and $\zeta_{1:k_{t_n}} = (\zeta_1, \ldots, \zeta_{k_{t_n}})$ are marks, given the observations $y_{1:r_{t_n}}$, with r_{t_n} the number of observations up to t_n. In other words to compute, for $n \geq 1$, at time t_n

$$\pi_n(k_{t_n}, \phi_{1:k_{t_n}}, \zeta_{1:k_{t_n}} | y_{1:r_{t_n}}) \quad \text{smoothing} \tag{5.1}$$

$$\pi_n(k_{t_n} - k_{t_{n-1}}, \phi_{k_{t_{n-1}}+1:k_{t_n}}, \zeta_{k_{t_{n-1}}+1:k_{t_n}} | y_{1:r_{t_n}}) \quad \text{filtering.} \tag{5.2}$$

In addition, there are static parameters specifying the probability model and these parameters can be estimated in a Bayesian manner.

Centanni and Minozzo (2006a,b) approach the problem of smoothing in this context using an MCMC-type algorithm, estimating static parameters using stochastic EM. This methodology cannot easily be adapted to the case where the static parameters are given a prior distribution. In addition, the theoretical validity of the approach has not been established; this is verified in Proposition 5.3.1. This chapter will further consider the application of SMC Samplers to the smoothing problem described above, with a running example from computational finance. In particular, a sampler is proposed, incorporating elements of the MCMC literature in a fashion similar to the work of Centanni and Minozzo (2006a,b). A key feature of this approach is that the user must select:

1. the sequence of distributions,

2. the mechanism by which particles are propagated.

If these points are not properly addressed, there can be a substantial discrepancy between the proposal and target; thus the variance of the importance weights, calculated at each step of the sampler, will be large and estimation inaccurate. This issue is particularly relevant when the targets are defined on a sequence of nested spaces, as is the case for the PP models – the space of the point process trajectories becomes larger with the time-parameter n. Thus, in choosing the sequence of target distributions, we are faced with the question of how much the space should be enlarged at each iteration of the SMC algorithm and how to choose a mechanism to propose particles in the new region of the space. This issue is referred to as *the difficulty of extending the space*.

Two solutions are proposed. The first is to *saturate* the state-space; a similar approach was used in a reversible-jump MCMC context by Carlin and Chib (1995). It is supposed that the observation interval, $[0, T]$, of the PP is known *a priori*. The sequence of target distributions is then defined on the whole interval and one sequentially introduces likelihood terms. This idea circumvents the problem of extending the space, at an extra computational cost. Inference for the original density of interest can be achieved by importance sampling (IS). This approach cannot be used if T is unknown. In the second approach, entitled *data-point tempering*, the sequence of target distributions is defined by sequentially introducing likelihood terms. This is achieved as follows: given that the PP has been sampled on $[0, t_n]$ the target is extended onto $[0, t_{n+1}]$ by sampling the missing part of the PP. Then one introduces likelihood terms into the target that correspond to the data (as in Chopin (2002)). Once all of the data have been introduced, the target density is (5.1). It should be noted that neither of the methods are online, but some simple fixes are detailed.

The rest of the chapter is organised as follows. Section 5.2 introduces the class of point process models that will be the focus of the rest of the chapter, and this will be introduced within the context of the modelling of UHF financial data. Two existing Monte Carlo approaches to the modelling of these point process models are presented in Section 5.3, including the SMC sampler approach that will be developed throughout the chapter. In Subsection 5.3.2, we consider the use of a generic SMC sampler for application to the models of interest; the poor performance of this algorithm is demonstrated and two alternative modifications to the sampler are proposed in Section 5.4. The two proposed samplers are implemented using both synthetic and genuine financial data in Section 5.5 and comparisons are drawn with the benchmark sampler. The chapter is concluded in Section 5.6.

5.2 The Model

The model we use to illustrate our ideas is from statistical finance. An important type of financial data is ultra high frequency data which consists of the irregularly spaced times of financial transactions and their corresponding monetary value. Standard models for the fitting of such data have relied upon stochastic differential equations driven by Wiener dynamics; a debatable setup due to the implied continuity of the sample paths of the latent process and the heavy-tail behaviour that is generally observable in the distribution of UHF log-returns. As noted in Centanni and Minozzo (2006b), it is more appropriate to model the data as a Cox process. Due to the high frequency of the data, it is important to be able to perform sequential/on-line inference; recall the convention, established in Section 1.5, that sequential inference refers to the ability to process data as it arrives, and on-line inference is achieved by sequential procedures that have fixed computational cost per iteration. Data are observed in $[0, T]$. In the context of finance, the assumption that T be fixed is entirely reasonable. For example, when the model is used in the context of equities, the model can be run for the trading day; indeed due to different (deterministic) patterns in financial trading, it is likely that the parameters below may be fixed at particular values for a given trading day.

A marked PP, of $r_T \geq 1$ points, is observed in time-period $[0, T]$. This is written $y_{1:r_T} = (\omega_{1:r_T}, \xi_{1:r_T}) \in \Omega_{r,T} \times \Xi^{r_T}$ with $\Omega_{r,T} = \{\omega_{1:r_T} \ ; \ 0 < \omega_1 < \cdots < \omega_{r_T} < T\}$, $\Xi \subseteq \mathbb{R}$. Here the $\omega_{1:r_T}$ are the transaction times and $\xi_{1:r_T}$ are the log-returns on the financial transactions. An appropriate model for such data, as in Centanni and Minozzo (2006b), is

$$\begin{aligned} p(\xi_{1:r_T}|\mu, \sigma) &= \prod_{i=1}^{r_T} p(\xi_i|\mu, \sigma) \\ p(\omega_{1:r_T}|\{\lambda_T\}) &\propto \prod_{i=1}^{r_T} \{\lambda_{\omega_i}\} \exp\left\{ - \int_0^T \lambda_u du \right\} \end{aligned}$$

where p is the generic notation for a probability density, $\xi_{1:r_T}$ are assumed to be t-distributed on 1 degree of freedom, with location μ and scale σ, and where λ_u is the intensity. The unobserved intensity process is assumed to follow the dynamics $d\lambda_t = -s\lambda_t dt + dJ_t$ with $\{J_t\}$ a compound Poisson process: $J_t = \sum_{j=1}^{k_t} \zeta_j$ with $\{k_t\}$ a Poisson process with rate parameter ν and i.i.d. jumps $\zeta_j \sim \mathcal{E}x(1/\gamma)$, $\mathcal{E}x(\cdot)$ is the exponential distribution. That is, for $t \in [0, T]$,

$$\lambda_t = \left\{ \lambda_0 e^{-st} + \sum_{j=1}^{k_t} \zeta_j e^{-s(t-\phi_j)} \right\} \tag{5.3}$$

with ϕ_j the jump times of the unobserved Poisson process and λ_0 fixed throughout. In practice, this initial intensity is obtained using a short preliminary time series.

We define the following notation:

$$\begin{aligned}
\bar{x}_n &= (k_{t_n}, \phi_{1:k_{t_n}}, \zeta_{1:k_{t_n}}), \\
\bar{x}_{n,1} &= (k_{t_n} - k_{t_{n-1}}, \phi_{k_{t_{n-1}}+1:k_{t_n}}, \zeta_{k_{t_{n-1}}+1:k_{t_n}}), \\
\bar{y}_n &= (\omega_{1:r_{t_n}}, \xi_{1:r_{t_n}}), \\
\bar{y}_{n,1} &= (\omega_{r_{t_{n-1}}+1:r_{t_n}}, \xi_{r_{t_{n-1}}+1:r_{t_n}}).
\end{aligned}$$

Here $\bar{x}_n$ (respectively $\bar{y}_n$) is the restriction of the hidden (observed) PP to events in $[0, t_n]$. Similarly $\bar{x}_{n,1}$ (respectively $\bar{y}_{n,1}$) is the restriction of the hidden (observed) PP to events in $[t_{n-1}, t_n]$.

The objective is to perform inference at times $0 < t_1 < \cdots < t_{\widetilde{m}} < T = t_{\widetilde{m}+1}$, that is, to update the posterior distribution conditional on the data arriving in $[t_{n-1}, t_n]$. To summarize, the posterior distribution at time t_n is

$$\begin{aligned}
\pi_n(\bar{x}_n, \mu, \sigma | \bar{y}_n) \propto & \prod_{i=1}^{r_{t_n}} \{p(\xi_i|\mu,\sigma)\lambda_{\omega_i}\} \exp\left\{ -\int_0^{t_n} \lambda_u du \right\} \\
& \times \prod_{i=1}^{k_{t_n}} \{\mathsf{p}(\zeta_i)\} \mathsf{p}(\phi_{1:k_{t_n}}) \mathsf{p}(k_{t_n}) \times p(\mu, \sigma) \\
= & \; g_{[0,t_n]}(\bar{y}_n | \bar{x}_n, \mu, \sigma) \times \mathsf{p}(\bar{x}_n) \times p(\mu, \sigma)
\end{aligned} \tag{5.4}$$

with $g_{[0,t_n]}$ the likelihood of the observed data over the interval $[0, t_n]$, $\mu \sim \mathcal{N}(\alpha_\mu, \beta_\mu)$, $\sigma \sim \mathcal{G}a(\alpha_\sigma, \beta_\sigma)$, $\phi_{1:k_t}|k_t \sim \mathcal{U}_{\Phi_{k,t_n}}$, $k_t \sim \mathcal{P}o(\nu t)$ and where $\mathcal{U}_A$ is the uniform distribution on the set A, $\mathcal{N}(\mu, \sigma^2)$ is the normal distribution of mean μ and variance

σ^2, $\mathcal{G}a(\alpha, \beta)$ the Gamma distribution of mean α/β and $\mathcal{P}o$ is the Poisson distribution. $\mathsf{p}(\bar{x}_n)$ is the prior on the latent point-process and $p(\mu, \sigma)$ is the prior on (μ, σ). Later a π_0 is introduced which will refer to an initial distribution. Note it is possible to perform inference on (μ, σ) independently of the unobserved PP; it will not significantly complicate the simulation methods to include them.

It is of interest to compute expectations w.r.t. the $\{\pi_n\}_{1\leq n\leq m^*}$, and this is possible, using the SMC methods below (Section 5.3.2). However, such algorithms are not of fixed computational cost; the sequence of spaces over which the $\{\pi_n\}_{1\leq n\leq m^*}$ lie is increasing. These methods can also be used to draw inference from the marginal posterior of the process, over $(t_{n-1}, t_n]$; such algorithms can be designed to be of fixed computational complexity, for example by constraining any simulation to a fixed-size state-space. This idea is considered further in Section 5.4.3.

5.3 Existing Approaches

5.3.1 MCMC for Latent Point Process Models

One of the approaches for performing smoothing for partially observed PP's is from (Centanni and Minozzo, 2006a). In this Section the parameters (μ, σ) are assumed known. Let

$$\bar{E}_n = \bigcup_{k\in\mathbb{N}_0} \Big(\{k\} \times \Phi_{k,t_n} \times (\mathbb{R}^+)^k\Big).$$

This is the support of the target densities for this method.

The following decomposition is adopted

$$\begin{aligned} \pi_n(\bar{x}_n|\bar{y}_n) &= \frac{g_{(t_{n-1},t_n]}(\bar{y}_{n,1}|\bar{x}_n)}{p_n(\bar{y}_{n,1}|\bar{y}_{n-1})}\mathsf{p}(\bar{x}_{n,1})\pi_{n-1}(\bar{x}_{n-1}|\bar{y}_{n-1}) \qquad (5.5)\\ p_n(\bar{y}_{n,1}|\bar{y}_{n-1}) &= \int g_{(t_{n-1},t_n]}(\bar{y}_{n,1}|\bar{x}_n)\mathsf{p}(\bar{x}_{n,1})\pi_{n-1}(\bar{x}_{n-1}|\bar{y}_{n-1})d\bar{x}_n. \end{aligned}$$

At iteration $n \geq 2$ of the algorithm, an RJMCMC kernel (although the analysis below is not restricted to such scenarios) is used for N steps to sample from the approximated smoothing density

$$\pi_n^N(\bar{x}_n|\bar{y}_n) \quad\propto\quad g_{(t_{n-1},t_n]}(\bar{y}_{n,1}|\bar{x}_n)\mathsf{p}(\bar{x}_{n,1})S_{x,n-1}^N(\bar{x}_{n-1})$$

where $S_{x,n-1}^N(\bar{x}_{n-1}) := \frac{1}{N}\sum_{i=1}^N \mathbb{I}_{\{\bar{X}_{n-1}^{(i)}\}}(\bar{x}_{n-1})$ with $\bar{X}_{n-1}^{(1)},\ldots,\bar{X}_{n-1}^{(N)}$ obtained from a reversible jump MCMC algorithm of invariant measure π_{n-1}^N. The algorithm for $n = 1$ targets π_1 exactly; there is no empirical density $S_{x,0}^N$. At time $n = 1$ the algorithm starts from an arbitrary point $\bar{x}_1^{(1)} \in \bar{E}_1$ and subsequent steps are initialized by a draw from the empirical $S_{x,n-1}^N$ and the prior p; $N-1$ additional samples are simulated. Let $K_1(\bar{x}_1,\cdot)$ denote the kernel with invariant distribution $\pi_1(\bar{x}_1|\bar{y}_1)$. For $n \geq 2$, denote the kernel with invariant distribution $\pi_n^N(d\bar{(x)}_n|\bar{y}_n)$ as $K_{S_{n-1}^N,n}(\bar{x}_n,\cdot)$.

Proposition 5.3.1, below, provides an upper bound for the expected $\mathbb{L}_p$-error of the resulting estimate of the expected value of a bounded test function, where the latter expectation is taken with respect to the smoothing distribution (5.1), and the former is taken with respect to the law of the process generated by the above procedure, given the observed data and an initial value $\bar{x}_1^{(1)} \in \bar{E}_1$. Under the restrictive assumption (A6), this result helps to establish the theoretical validity of this method, which was not established in that paper or, to my knowledge, anywhere else. In addition, it allows us to understand where and when the method may be of use; this is discussed in Section 5.3.3.

(**A6**) There exist an $\epsilon_1 \in (0,1)$ and probability measure κ_1 on $\bar{E}_1$ such that for any $\bar{x}_1 \in \bar{E}_1$

$$K_1(\bar{x}_1,\cdot) \geq \epsilon_1\kappa_1(\cdot).$$

For any $n \geq 2$, there exist an $\epsilon_n \in (0,1)$ and probability measure κ_n on $\bar{E}_n \setminus \bar{E}_{n-1}$ such that for any $\bar{x}_n \in \bar{E}_n$ and any collection of points $(\bar{X}_{n-1}^{(1)},\ldots,\bar{X}_{n-1}^{(N)}) \in \bar{E}_{n-1}^N$

$$K_{S_{n-1}^N,n}(\bar{x}_n,\cdot) \geq \epsilon_n S_{n-1}^N(\cdot)\kappa_n(\cdot).$$

For any $n \geq 2$

$$\sup_{\bar{x}_{n-1} \in \bar{E}_{n-1}} \int_{\bar{E}_n \setminus \bar{E}_{n-1}} \left| \frac{g_{(t_{n-1},t_n]}(\bar{y}_{n,1}|\bar{x}_n)}{p_n(\bar{y}_{n,1}|\bar{y}_{n-1})} \mathsf{p}\left(\bar{x}_{n,1}\right) \right| d\bar{x}_{n,1} < +\infty.$$

Proposition 5.3.1. *Assume (A6). Then for any $n \geq 1$, $\bar{y}_n$, $p \geq 1$ there exists $B_{p,n}(\bar{y}_n) < +\infty$ such that for any $f_n \in \mathcal{B}_b(\bar{E}_n)$*

$$\mathbb{E}_{\bar{x}_1^{(1)}} \left[\left| \frac{1}{N} \sum_{i=1}^{N} f_n(\bar{X}_n^{(i)}) - \int_{\bar{E}_n} f_n(\bar{x}_n) \pi_n(d\bar{x}_n) \right|^p \middle| \bar{y}_n \right]^{1/p} \leq \frac{B_{p,n}(\bar{y}_n) \|f_n\|}{\sqrt{N}}. \tag{5.6}$$

The proof of this error bound was provided by Dr. Jasra; see Martin et al. (2012a) for details.

5.3.2 SMC Samplers for Latent Point Process Models

Through the use of SMC samplers, we aim to approximate a sequence of related probability measures $\{\pi_n\}_{0 \leq n \leq m^*}$ defined upon the common space $(E, \mathcal{E})$. Note that $m^* > 1$ can depend upon the data and may not be known prior to simulation. For partially observed PPs the probability measures are defined upon nested state-spaces: this case can be similarly handled with minor modification. SMC samplers introduce a sequence of auxiliary probability measures $\{\widetilde{\pi}_n\}_{0 \leq n \leq m^*}$ on state-spaces of increasing dimension $(E_{[0,n]} := E_0 \times \cdots \times E_n, \mathcal{E}_{[0,n]} := \mathcal{E}_0 \otimes \cdots \otimes \mathcal{E}_n)$, such that they admit the $\{\pi_n\}_{0 \leq n \leq m^*}$ as marginals.

The following sequence of auxiliary densities is used:

$$\widetilde{\pi}_n(x_{0:n}) = \pi_n(x_n) \prod_{j=0}^{n-1} l_j(x_j | x_{j+1}) \tag{5.7}$$

where $\{l_n\}_{0 \leq n \leq m^*-1}$ are the densities admitted by the backward Markov kernels, with respect to some dominating measure. In our application π_0 is the prior, on E_1 (as defined below). It is clear that the densities (5.7) admit the $\{\pi_n\}$ as marginals, and hence these distributions can be targeted using a standard SMC sampler, as

Parameters: N, m^*;
Result: An SMC sample from the target distribution π_{m^*}

1. Set $n = 0$. For $i = 1 : N$, sample $X_0^{(i)} \sim \varrho_0$ and compute
$$W_0^{(i)} \propto \frac{\pi_0(X_0^{(i)})}{\varrho_0(X_0^{(i)})}, \quad \sum_{i=1}^{N} W_0^{(i)} = 1.$$
2. If $\widehat{ESS}\left(\left\{W_n^{(i)}\right\}_{i=1}^{N}\right) < \frac{N}{2}$, then, for $i = 1, \ldots, N$, resample $X_n^{(i)}$ independently from the discrete distribution
$$\hat{\pi}\left(dx_n\right) = \sum_{i=1}^{N} W_n^{(i)} \delta_{X_n^{(i)}}\left(dx_n\right)$$
and set $W_n^{(i)} = \frac{1}{N}$.
3. Set $n = n + 1$. For $i = 1, \ldots, N$ sample $X_n^{(i)} \sim K_n(X_{n-1}^{(i)}, dx_n)$, and compute:
$$W_n^{(i)} \propto W_{n-1}^{(i)} \frac{\pi_n(X_n^{(i)}) l_{n-1}(X_{n-1}^{(i)}, X_n^{(i)})}{\pi_{n-1}(X_{n-1}^{(i)}) k_n(X_{n-1}^{(i)}, X_n^{(i)})}, \quad \sum_{i=1}^{N} W_n^{(i)} = 1. \tag{5.8}$$
If $n = m^*$ stop, else return to Step 2.

Algorithm 13: A Generic SMC Sampler.

in Algorithm 13. As with the filtering and smoothing procedures, we use the ESS to control the degeneracy of the weights in the sampler, performing multinomial resampling whenever the estimated ESS drops below $N/2$.

One generic approach is to set K_n as an MCMC kernel of invariant distribution π_n and L_{n-1} as the reversal kernel $L_{n-1}(x_n, dx_{n-1}) = \pi_n(x_{n-1})K_n(x_{n-1}, dx_n)/\pi_n(x_n)$ which we term the *standard reversal kernel.* Extensions to this approach include the iterative application of an invariant MCMC kernel, and the application of a mixture of kernels. See Del Moral et al. (2006) for details on the algorithm in the latter scenario.

Nested Spaces

As described in Section 5.1, in complex problems it is often difficult to design efficient SMC algorithms. In the current example, the state-spaces of the consecutive densities are not common. The objective is to sample from a sequence of distributions defined, at time n, on the space

$$E_n = \left(\bigcup_{k \in \mathbb{N}_0} \{k\} \times \Phi_{k,t_n} \times (\mathbb{R}^+)^k \right) \times \mathbb{R} \times \mathbb{R}^+ \quad 1 \leq n \leq m^* - 1$$

with $E_0 = E_1$. That is, for any $1 \leq n \leq m^* - 1$, $E_n \subseteq E_{n+1}$. Two standard methods for extending the space, as in Del Moral et al. (2006) are to propagate particles by application of the 'birth' and 'extend' moves. For the model in Section 5.2, these are defined at time n as follows:

- **Birth**. A new jump is sampled uniformly in $[\phi_{k_{t_{n-1}}}, t_n]$ and a new mark from the prior. The incremental weight function is

  $$\tilde{w}_n(\bar{x}_{n-1:n}, \mu, \sigma) = \frac{\pi_n(\bar{x}_n, \mu, \sigma|\bar{y}_n)(t_n - \phi_{k_{t_{n-1}}})}{\pi_{n-1}(\bar{x}_{n-1}, \mu, \sigma|\bar{y}_n)\mathsf{p}(\zeta_{k_{t_n}})}.$$

- **Extend**. A new jump is generated according to a Markov kernel that corresponds to the random walk:

  $$\log\left\{\frac{\phi_{k_{t_n}} - \phi_{k_{t_n}-1}}{t_n - \phi_{k_{t_n}}}\right\} = \vartheta Z + \log\left\{\frac{\phi_{k_{t_{n-1}}} - \phi_{k_{t_{n-1}}-1}}{t_n - \phi_{k_{t_{n-1}}}}\right\}$$

  with $Z \sim \mathcal{N}(0,1)$, $\vartheta > 0$. The new mark is sampled from the prior. The backward kernel and incremental weight are discussed in Del Moral et al. (2007, Section 4.3).

Note, as remarked in Whiteley et al. (2011), we need to be able to sample any number of births. With an extremely small probability, a proposal from the prior is included to form a mixture kernel.

In addition to the above steps an RJMCMC sweep is included after the decision of whether or not to resample the particles is taken. An MCMC kernel of invariant measure π_n is applied; the kernel is much the same as in Green (1995). It is possible to iterate the application of this RJMCMC kernel, in which case we use the parameter M to define the number of iterates. In applying this MCMC kernel, the aim is to rejuvenate the particle histories, in a similar fashion to Gilks and Berzuini (2001); it is expected that this will reduce path degeneracy and improve the accuracy of inference based upon the smoothing density (5.1).

Simulation Experiment

We applied the benchmark sampler, as detailed in Algorithm 13 and using the extend move, above, to propagate the particles, to some synthetic data in order to monitor the performance of the algorithm. Standard practice in the reporting of financial data is to represent the time of a trade as a positive real number, with the integer part representing the number of days passed since January 1st 1900 and the non-integer part representing the fraction of 24 hours that has passed during that day; thus, one minute corresponds to an interval of length 1/1440. Therefore we use a synthetic data set with intensity of order of magnitude 10^3. The ticks ω_i were generated from a specified intensity process $\{\lambda_t\}$ that varied smoothly between three levels of constant intensity at $\lambda = 6000$, $\lambda = 2000$ and $\lambda = 4000$. The log returns ξ_i were sampled from the Cauchy-distribution, location $\mu = 0$ and scale $\sigma = 2.5 \times 10^{-4}$. The entire data set was of size $r_T = 3206$, $[0, T] = [0, 0.9]$ with $t_n = n * 0.003$. The intensity from which they were generated had constant levels at 6000 in the interval [0.05,0.18]; at 4000 in the interval [0.51,0.68]; and at 2000 in the intervals [0.28,0.42] and [0.78,0.90].

The sampler was implemented with all combinations $\{M, N\}$ for $N \in \{100, 1000\}$ and $M \in \{1, 5, 20\}$, resampling whenever the estimated effective sample size fell below $N/2$ (recall N is the number of particles and M the MCMC iterations).

When performing statistical inference, the intensity (5.3) used parameters $\gamma = 0.001$, $\nu = 150$ and $s = 20$.

It was found that for this SMC sampler, the system consistently collapses to a single particle representation of the distribution of interest within an extremely short time period. That is, resampling is needed at almost every time step, which leads to an extremely poor representation of the target density. Figure 5.1 shows the estimated ESS at each time step for a particular implementation. As can be seen, the algorithm behaves extremely poorly for this model.

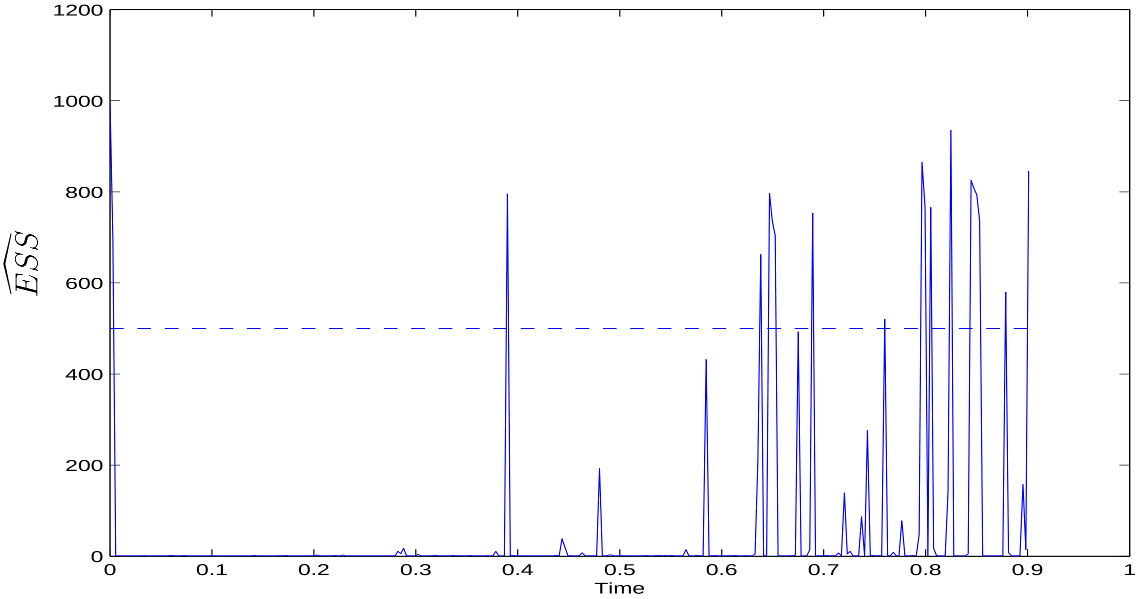


Figure 5.1: Estimated effective sample sizes, calculated at each time step of the benchmark SMC sampler described in Algorithm 13, implemented with $N = 1000$ particles and with $M = 5$ MCMC sweeps at each iteration. The dashed line indicates the resampling threshold at $N/2 = 500$ particles; resampling is needed at 94.4% of the time steps.

5.3.3 Discussion

We have reviewed two existing techniques for the Bayesian analysis of partially observed PP's. It should be noted that there are other methods, for example in Varini (2008). In that paper, the intensity has a finite number of functional forms and the uncertainty is related to the type of form at each inference time t_n.

The relative advantage of the approach of Centanni and Minozzo (2006a) is the fact that the state-space need not be extended. On page 1586 of Centanni and Minozzo (2006a) the authors describe the filtering/smoothing algorithm, for the process on the entire interval $[0, t_n]$ at time n; the theory discussed in Proposition 5.3.1 suggests that this method is not likely to work well as n grows. The constant $B_{p,n}(\bar{y}_n)$ in the upper bound given in that result is, for $n \geq 2$

$$B_{p,n}(\bar{y}_n) = \frac{2}{\epsilon_n(\bar{y}_n)}[B_p + 1] + \hat{k}_n B_{p,n-1}(\bar{y}_{n-1})$$

with $B_{p,1}(\bar{y}_1) = \frac{2}{\epsilon_1(\bar{y}_1)}[B_p+1]$, B_p a constant related to the Bürkholder/Davis inequalities (e.g. Shiryaev, 1996), $\epsilon_n(\bar{y}_n) \in (0, 1)$ and $\hat{k}_n > 0$ a constant that is model/data dependent which is possibly bigger than 1. The bound indicates that the error can increase over time, even under the exceptionally strong assumption (6). This is opposed to SMC methods which are provably stable, under similar assumptions (and under the assumption that the entire state is updated), as $n \to \infty$ (Del Moral, 2004). In other words, whilst the approach of Centanni and Minozzo is useful in difficult problems, it is less general with potentially slower convergence rate than SMC. Intuitively, it seems that the method of Centanni and Minozzo (2006a) is perhaps only useful when considering the process on $(t_{n-1}, t_n]$, as the process further back in time is not rejuvenated in any way. As a result, parameter estimation may not be very accurate. In addition, the method cannot be extended to a sequential algorithm such that fully Bayesian inference is possible. As noted above, SMC samplers can be used in such contexts, but requires a computational budget that grows with the time parameter n.

As mentioned above, SMC methods are provably stable under some conditions as the time parameter grows. However, some remarks related to the method in Algorithm 13 can help to shed some light on the poor behaviour in Section 5.3.2. Consider the scenario when one is interested in statistical inference on $[0, t_1]$. Suppose for simplicity, one can write the posterior on this region as

$$\pi(\bar{x}_1) \propto \exp\{\sum_{i=1}^{r_{t_1}} h_i(y_i|\bar{x}_1)\}\mathsf{p}(\bar{x}_1) \tag{5.9}$$

for fixed r_{t_1}, μ, σ. If one considers just pure importance sampling, then conditioning upon the data, one can easily show that for any π_1-(square) integrable f with $\int f(\bar{x}_1)\mathsf{p}(\bar{x}_1)d\bar{x}_1 = 0$, the asymptotic variance in the associated Gaussian central limit theorem is lower-bounded by:

$$\Big(\int f(\bar{x}_1)^2 \exp\{2\sum_{i=1}^{r_{t_1}} h_i(y_i|\bar{x}_1)\}\mathsf{p}(\bar{x}_1)d\bar{x}_1\Big)/\Big(\int \exp\{\sum_{i=1}^{r_{t_1}} h_i(y_i|\bar{x}_1)\}\mathsf{p}(\bar{x}_1)d\bar{x}_1\Big).$$

Then, for any mixing type sequence of data the asymptotic variance will for some f and in some scenarios, grow without bound as r_{t_1} grows - this is a very heuristic observation, that requires further investigation. Hence, given this discussion and our empirical experience, it seems that we require a new methodology, especially for complex problems.

An important remark related to the simulations in Section 5.3.2, is that it cannot be expected that simply increasing the number of particles will necessarily lead to a significantly better estimation procedure. The algorithm completely crashes to a single particle and it seems that naively increasing computation will not improve the simulations.

As discussed above, the inherent difficulty of sampling from the given sequence of distributions is that of extending the state-space. It is known that conditional on all parameters except the final jump, the optimal importance distribution is the full conditional density (Del Moral et al., 2006). In practice, for many problems it is either not possible to sample from this density, or to evaluate it exactly (which is required). In the case that it is possible to sample from the full conditional, but the normalizing constant is unknown, the normalizing constant problem can be dealt with via the random weight idea (Rousset and Doucet, 2006). In the context of this problem we found that the simulation from the full conditional density of $\phi_{k_{t_n}}$ was difficult, to the extent that sensible rejection algorithms and approximations for the random weight technique were extremely poor.

Another solution, in Del Moral et al. (2007), consists of stopping the algorithm when the estimated ESS drops below the required threshold, and using an additional SMC sampler to facilitate the extension of the state-space. However, in this example, the ESS is so low, that it cannot be expected to help. Due to above discussion, it is clear that a new technique is required to sample from the sequence of distributions; two ideas are presented below. It is noted that one further idea that could be adopted in the context of estimating static parameters is SMC^2 (Chopin et al., 2011), although this is not explored in any further detail in this thesis.

5.4 Proposed Methods

In the following Section, two approaches are presented to deal with the problems in Section 5.3.2. First, a state-space saturation approach, where sampling of PP trajectories is performed over a state space corresponding to a fixed observation interval. Second, a data-point tempering approach. In this approach, as the time parameter increases, the (artificial) target in the new region is simply the prior and the data are then sequentially added to the likelihood, softening the state-space extension problem. Both of these procedures use the basic structure of Algorithm 13, with some refinements, that are mentioned in the text. In particular, when MCMC kernels are used, one can resample before sampling; this is explained below.

5.4.1 State Space Saturation

A simple idea, which has been used in the context of reversible jump, is to saturate the state-space. The idea relies upon knowing the observation period of the PP ($[0, T]$) *a priori* to the simulation. This is realistic in a variety of applications. For example, in Section 5.2, often we may only be interested in performing inference for a day of trading and thus can set $[0, T]$.

In details, it is proposed to sample, in the case of the example in Section 5.2, from the sequence of target densities defined on the space

$$E = \left(\bigcup_{k \in \mathbb{N}_0} \{k\} \times \Phi_{k,T} \times (\mathbb{R}^+)^k \right) \times \mathbb{R} \times (\mathbb{R}^+)^2. \tag{5.10}$$

The (marginal, that is in the sense of (5.7)) target densities are now, denoted with an S as a super-script:

$$\begin{aligned} \pi_n^S(\bar{x}_n, \mu, \sigma | \bar{y}_n) \propto & \prod_{i=1}^{r_{t_n}} \{p(\xi_i|\mu,\sigma)\lambda_{\omega_i}\} \exp\left\{ - \int_0^{t_n} \lambda_u du \right\} \\ & \times \prod_{i=1}^{k_{t_n}} \{\mathsf{p}(\zeta_i)\} \mathsf{p}^S(\phi_{1:k_{t_n}}) \mathsf{p}^S(k_{t_n}) \times p(\mu,\sigma) \quad 1 \leq n \leq T \end{aligned}$$

where the prior on the point process is $\phi_{1:k_t}|k_t \sim \mathcal{U}_{\Phi_{k,T}}$, $k_t \sim \mathcal{P}o(\nu T)$. The initial distribution for each of the particles is the prior, and the initial weights are set proportional to 1. At time t_n, we then use for $K_n^S(\bar{x}_{n-1}, d\bar{x}_n)$ an MCMC kernel of invariant measure π_n^S to move the particles, and we use the standard reversal kernel $L_{n-1}^S(\bar{x}_n, dx_{n-1}) = \pi_n^S(\bar{x}_{n-1})K_n^S(\bar{x}_{n-1}, d\bar{x}_n)/\pi_n^S(\bar{x}_n)$ for the backward kernel, such that the incremental weight function used at each time-point is simply:

$$\tilde{w}_n(\bar{x}_{n-1}, \mu_{n-1}, \sigma_{n-1}) = \frac{\pi_n^S(\bar{x}_{n-1}, \mu_{n-1}, \sigma_{n-1}|\bar{y}_n)}{\pi_{n-1}^S(\bar{x}_{n-1}, \mu_{n-1}, \sigma_{n-1}|\bar{y}_{n-1})} \quad 1 \leq n \leq T.$$

Inference w.r.t. the original $\{\pi_n\}_{1 \leq n \leq m^*}$ can be performed via IS as the supports of the targets of interest are contained within the proposals (i.e. via the targets of the saturated algorithm).

As mentioned previously, since the incremental weight is independent of the proposed particle $\bar{x}_n$, the weighting and resampling steps of the sampler can be carried out prior to propagation via the MCMC kernel K_n^S. We note further that, although the latent PP is proposed from the entire observation interval $[0, T]$ at all time points, the generated process over the interval $(t_n, T]$ is not used; thus, we have a great deal of wasteful sampling at each time step. By combining these observations, we can save computational effort by

- generating a cloud of particles from the prior over $[0, T]$ prior to sequential implementation of the saturated sampler, and
- at each time t_n, combining the existing latent PP over $[0, t_{n-1}]$, $\bar{x}_{n-1}$, with its pre-proposed extension $\bar{x}_{n,1}$, and moving this extended particle according to $M \geq 1$ applications of an RJMCMC kernel of invariant density π_n^S.

This is the approach that is adopted in practice; the full saturated sampler is provided in Algorithm 14.

5.4.2 Data-Point Tempering

A second solution to the state-space extension problem, which allows data to be incorporated sequentially, is as follows. When the time parameter increases, the new part of the process is simulated according to the prior. Then each new data point is added to the likelihood in a sequential manner. In other words if there are r_T data points, then there are $m^* = r_T + \widetilde{m}$ time-steps of the algorithm.

To illustrate, consider only the scenario of the data in $[0, t_1]$, with $r_{t_1} > 0$. Then our sequence of (marginal) targets are: $\pi_0^{\mathrm{TE}}(\bar{x}_1, \mu, \sigma) = \mathsf{p}(\bar{x}_1) p(\mu, \sigma)$ and for $1 \leq n \leq r_{t_1}$

$$\pi_n^{\mathrm{TE}}(\bar{x}_1, \mu, \sigma | y_{1:n}) \propto \prod_{i=1}^{n} \left\{ p(\xi_i | \mu, \sigma) \lambda_{\omega_i} \right\} \exp\left\{ - \int_0^{t_1} \lambda_u du \right\} \mathsf{p}(\bar{x}_1) p(\mu, \sigma).$$

Then, when considering the extension of the point-process onto $[0, t_2]$, one has a

Parameters: N, m^*, λ_0;
Result: An SMC sample from the target distribution $\pi^S_{m^*}$

1. For $i = 1 : N$, sample $k^{(i)}_T$, $\phi^{(i)}_{1:k_T}|k^{(i)}_T$, $\zeta^{(i)}_{1:k_T}|k^{(i)}_T$ according to their respective priors. For $i = 1 : N$ and for $n = 1 : m^*$, set $\bar{X}^{(i)}_{n,1} = (k^{(i)}_{t_n} - k^{(i)}_{t_{n-1}}, \phi_{k^{(i)}_{t_{n-1}}+1:k^{(i)}_{t_n}}, \zeta_{k^{(i)}_{t_{n-1}}+1:k^{(i)}_{t_n}})$ and sample $\mu^{(i)}_n$ and $\sigma^{(i)}_n$ according to their priors.
2. Set $n = 1$. For $i = 1 : N$, set $\bar{X}^{(i)}_1 = (k^{(i)}_{t_1}, \phi^{(i)}_{1:k^{(i)}_{t_1}}, \zeta^{(i)}_{1:k^{(i)}_{t_1}})$, and set $W^{(i)}_1 = \frac{1}{N}$.
3. Set $n = n + 1$. Compute:
$$W^{(i)}_n \propto W^{(i)}_{n-1} \frac{\pi^S_n(\bar{X}^{(i)}_{n-1}, \mu^{(i)}_{n-1}, \sigma^{(i)}_{n-1}|\bar{y}_n)}{\pi^S_{n-1}(\bar{X}^{(i)}_{n-1}, \mu^{(i)}_{n-1}, \sigma^{(i)}_{n-1}|\bar{y}_{n-1})}, \quad \sum_{i=1}^{N} W^{(i)}_n = 1. \tag{5.11}$$
4. If $\widehat{ESS}\left(\left\{W^{(i)}_n\right\}_{i=1}^{N}\right) < \frac{N}{2}$, then, for $i = 1, \ldots, N$, resample $\left(\bar{X}^{(i)}_{n-1}, \mu^{(i)}_{n-1}, \sigma^{(i)}_{n-1}\right)$ independently from the discrete distribution
$$\hat{\pi}^S_n\left(d\bar{x}_{n-1}, d\mu_{n-1}, d\sigma_{n-1}\right) = \sum_{i=1}^{N} W^{(i)}_n \delta_{(\bar{X}^{(i)}_{n-1}, \mu^{(i)}_{n-1}, \sigma^{(i)}_{n-1})}\left(d\bar{x}_{n-1}, d\mu_{n-1}, d\sigma_{n-1}\right).$$
5. For $i = 1, \ldots, N$ sample $\bar{X}^{(i)}_n \sim K^{RJ}_n(\{\bar{X}^{(i)}_{n-1}, \bar{X}^{(i)}_{n,1}\}, d\bar{x}_n)$, where K^{RJ}_n is the composition of $M \geq 1$ RJMCMC kernels, each of which is as in Green (1995).
6. Estimate $\hat{\lambda}_{t_n} = \sum_{i=1}^{N} W^{(i)}_{t_n} \left\{\lambda_0 e^{-st_n} + \sum_{j=1}^{k^{(i)}_{t_n}} \zeta^{(i)}_j e^{-s(t_n - \phi^{(i)}_j)}\right\}$.
7. If resampling occurred in Step 4, set $W^{(i)}_n = \frac{1}{N}$. If $n = m^*$ stop, else return to Step 3.

Algorithm 14: An SMC Sampler with State Space Saturation

(marginal) target that is:

$$\pi^{\mathrm{TE}}_{r_{t_1}+1}(\bar{x}_2, \mu, \sigma|\bar{y}_1) \propto \prod_{i=1}^{r_{t_1}} \left\{p(\xi_i|\mu, \sigma)\lambda_{\omega_i}\right\} \exp\left\{ - \int_0^{t_1} \lambda_u du\right\} \mathsf{p}(\bar{x}_2) p(\mu, \sigma)$$

When one extends the state-space, we sample from the prior on the new segment, which leads to a unit incremental weight (up-to proportionality) - no backward kernel is required here. Then, when adding data, we simply use MCMC kernels to move the particles (the kernels as in Section 5.3.2) and the standard reversal kernel discussed in Section 5.3.2 for the backward kernel. This leads to an incremental weight that is the ratio of the consecutive densities at the previous state.

Full details of a tempered SMC sampler are given in Algorithm 15. Note that, as with the saturated sampler, the weighting and resampling steps can be executed before extending the latent process.

The potential advantage of this idea is that, when extending the state-space, there is no extra data, to potentially complicate the likelihood. Thus, it is expected that if the prior does not propose a significant number of new jumps, that the incremental weights should be of low variance, relative to those obtained when using the approach in Section 5.3.2. The subsequent steps, when considering the jumps in $[t_n, t_{n+1})$ are performed on a common state-space and hence should not be subject to as substantial a variability as when the state-space changes. This idea could also be adapted to the case that the likelihood on the new interval are tempered instead (e.g. Jasra et al., 2007).

A theoretical justification of this idea was provided by Dr. Jasra for the paper of Martin et al. (2012a); it was shown that, under various assumptions, the asymptotic variance of the estimate, provided by the tempered sampler, of the expected value of a test function, where the expectation is taken with respect to the smoothing density $\pi_{r_{t_1}}(\bar{x}_1)$, is bounded. However, the assumptions under which this result holds are extremely strong and so it is noted that it is potentially over-optimistic in more general conditions. The assumptions and result are included nonetheless, as it is illustrative of the potential of the tempered sampler for this class of models, and goes some way to justifying a numerical examination of the sampler.

```
Parameters: $N$, $m^* = r_T + \widetilde{m}$, $\lambda_0$;
Result: An SMC sample from the target distribution $\pi_{m^*}^{TE}$

1. For $i = 1 : N$, sample $k_T^{(i)}$, $\phi_{1:k_T}^{(i)}|k_T^{(i)}$, $\zeta_{1:k_T}^{(i)}|k_T^{(i)}$ according to their respective
   priors. For $i = 1 : N$ and for $n = 1 : m^*$, set
   $\bar{X}_{n,1}^{(i)} = (k_{t_n}^{(i)} - k_{t_{n-1}}^{(i)}, \phi_{k_{t_{n-1}}^{(i)}+1:k_{t_n}^{(i)}}, \zeta_{k_{t_{n-1}}^{(i)}+1:k_{t_n}^{(i)}})$ and sample $\mu_{n-1}^{(i)}$ and $\sigma_{n-1}^{(i)}$
   according to their priors.

2. Set $n = 0$. For $i = 1 : N$, set $\bar{X}_0^{(i)} = \{k_0^{(i)}, \phi_{1:k_0^{(i)}}^{(i)}, \zeta_{1:k_0^{(i)}}^{(i)}\} = \emptyset$ and set $W_1^{(i)} = \frac{1}{N}$.

3. Set $n = n + 1$. For $n' = r_{t_{n-1}} + 1 : r_{t_n}$,
   • Compute:
     $$W_{n'}^{(i)} \propto W_{n'-1}^{(i)} \frac{\pi_{n'}^{TE}(\bar{X}_{n-1}^{(i)}, \mu_{n-1}^{(i)}, \sigma_{n-1}^{(i)}|\bar{y}_{n'})}{\pi_{n'-1}^{TE}(\bar{X}_{n-1}^{(i)}, \mu_{n-1}^{(i)}, \sigma_{n-1}^{(i)}|\bar{y}_{n'-1})}, \quad \sum_{i=1}^{N} W_{n'}^{(i)} = 1. \qquad (5.12)$$
   • If $\widehat{ESS}\left(\left\{W_{n'}^{(i)}\right\}_{i=1}^{N}\right) < \frac{N}{2}$, then, for $i = 1 : N$, resample
     $\left(\bar{X}_{n-1}^{(i)}, \mu_{n-1}^{(i)}, \sigma_{n-1}^{(i)}\right)$ independently from the discrete distribution
     $$\hat{\pi}_{n'}^{TE}(d\bar{x}_{n-1}, d\mu_{n-1}, d\sigma_{n-1}) = \sum_{i=1}^{N} W_{n'}^{(i)} \delta_{(\bar{X}_{n-1}^{(i)}, \mu_{n-1}^{(i)}, \sigma_{n-1}^{(i)})}(d\bar{x}_{n-1}, d\mu_{n-1}, d\sigma_{n-1})$$
     and set $W_{n'}^{(i)} = \frac{1}{N}$.
   • For $i = 1, \ldots, N$ sample $\bar{X}_{n-1}^{(i)} \sim K_{n-1}^{RJ}(\bar{X}_{n-1}^{(i)}, d\bar{x}_{n-1})$, where $K_{n-1}^{RJ}$ is an
     RJMCMC kernel, as in Green (1995).

4. For $i = 1, \ldots, N$ sample $\bar{X}_n^{(i)} \sim K_n^{RJ}(\{\bar{X}_{n-1}^{(i)}, \bar{X}_{n,1}^{(i)}\}, d\bar{x}_n)$, where $K_n^{RJ}$ is the
   composition of $M \geq 1$ RJMCMC kernels, each of which is as in Green (1995).

5. Estimate $\hat{\lambda}_{t_n} = \sum_{i=1}^{N} W_{t_n}^{(i)} \left\{\lambda_0 e^{-st_n} + \sum_{j=1}^{k_{t_n}^{(i)}} \zeta_j^{(i)} e^{-s(t_n - \phi_j^{(i)})}\right\}$.

6. If $n = m^*$ stop, else return to Step 3.
```

Algorithm 15: An SMC Sampler with Data Point Tempering

Theoretical Examination of the Tempered Sampler

We return to the discussion of Section 5.3.3 and in particular, where the joint target density is (5.9). Consider the data-point tempering which starts with a draw from the prior and sequentially adds data points; the sampler runs for $r_{t_1}+1$ time-steps with

$$\pi_n(\bar{x}_1) \propto \exp\{\sum_{i=1}^{n} h_i(y_i|\bar{x}_1)\}\mathsf{p}(\bar{x}_1)$$

where, for each i, y_i, $\underline{h} \leq h_i(y_i|\bar{x}_1) \leq \overline{h}$ for all $\bar{x} \in \bar{E}_1$, with $-\infty < \underline{h} < \overline{h} < \infty$. The algorithm is assumed to resample at every time-step and to use MCMC kernels, which are further assumed to satisfy, for some $\tau \in (0,1)$, and for each $1 \leq n \leq r_{t_1}$, with r_{t_1}, $\bar{x}_1$, $\bar{x}_1'$

$$K_n(\bar{x}_1,\cdot) \geq \tau K_n(\bar{x}_1',\cdot).$$

A resampling sweep is also assumed after the final weighting of the particles, at the very final time-step. Write $\bar{X}_1^1,\dots,\bar{X}_1^N$ as the samples that approximate target (5.9). Suppose $f \in \mathcal{B}_b(\bar{E}_1)$, then there is a Gaussian central limit theorem for

$$\sqrt{N}\Big(\frac{1}{N}\sum_{i=1}^{N} f(\bar{X}_1^i) - \int_{\bar{E}_1} f(\bar{x}_1)\pi_{r_{t_1}}(\bar{x}_1)d\bar{x}_1\Big).$$

Writing the asymptotic variance as $\sigma^2_{\mathrm{TE},r_{t_1}}(f)$, we have the following result whose proof can be found in Martin et al. (2012a).

Proposition 5.4.1. *For SMC sampler described above, with final target* (5.9) *then we have for any $f \in \mathcal{B}_b(\bar{E}_1)$ that there exists a $B \in (0,+\infty)$ such that for any $r_{t_1} \geq 1$, $\bar{y}_1$*

$$\sigma^2_{TE,r_{t_1}}(f) \leq B.$$

The upper-bound does not grow with the number of data. That is, by increasing the computational complexity linearly in the number of data, one has an algorithm whose error does not grow as more data (and regions) are added. This is similar to the observation of Beskos et al. (2011), when increasing the dimension of the target density.

5.4.3 Online Implementation

A key characteristic that has not yet been addressed is the fact that each of the SMC samplers discussed here has a computational complexity that is increasing with time. In a procedure that would otherwise be well suited to providing online inference, this is an unattractive feature. A large contribution to this increasing computational budget derives from the MCMC sweeps at the end of each iteration. As the space over which the invariant MCMC kernel is being applied is increased, so does the expense of the algorithm. An improvement to the computational demand of the samplers can therefore be made by keeping the space over which the MCMC kernel is applied constant. The *reduced computational complexity (RCC)* alternative to each of the samplers is also designed by amending the algorithms such that, at time t_n, the MCMC sweep operates over, at most, 20 changepoints, i.e. over the interval $\left[\phi_{k_{t_n}-19}, t_n\right)$. Due to the well-known path degeneracy problem in SMC, the estimates will be poor approximations of the true values, when including static parameters and extending the space of the point process for a long time. We note, at least for our application, it is reasonable to consider T fixed and thus, this is less problematic.

5.5 Example: The Finance Problem Revisited

We now return to the example from Section 5.2 and the settings as in Section 5.3.2.

5.5.1 Simulated Data

The saturated and tempered samplers, as well as their RCC alternatives, were implemented using the simulated data set (in Section 5.3.2), in order to compare their respective performances against the benchmark sampler and to compare the accuracy of the resulting intensity estimates against an observed intensity process. All of the alternative samplers were implemented under the same conditions, using the

	M = 1		M = 5		M = 20	
	N=100	N=1000	N=100	N=1000	N=100	N=1000
Benchmark	31.3%	52.0%	42.3%	94.4%	74.0%	99.7%
Benchmark - RCC	37.6%	88.1%	69.0%	99.7%	99.4%	99.7%
Saturated	21.0%	21.3%	19.7%	20.1%	18.2%	17.6%
Saturated - RCC	20.7%	20.7%	18.5%	18.8%	15.4%	15.4%
Tempered	2.0%	2.0%	1.9%	1.9%	1.7%	1.7%
Tempered - RCC	2.0%	2.0%	1.7%	1.8%	1.4%	1.4%

Table 5.1: Resampling rates for each of the three SMC samplers and their reduced computational complexity alternatives, for the six algorithm parameterisations that were tested. The estimated ESS plots for the saturated and tempered samplers with $N = 1000$, $M = 5$ are given in Figure 5.2 for comparison with the corresponding estimated ESS plot for the benchmark sampler given in Figure 5.1

algorithm and model parameters as described for the implementation of the benchmark sampler. All results are averaged over 10 runs of the algorithm.

In assessing the performance of the sampler, quantities of interest are, once again, the resampling rate and the processing time, as well as the minimum estimated ESS recorded throughout the execution of the sampler. The resampling rates for all three samplers and their RCC alternatives are presented in Table 5.1, with the corresponding minimum estimated ESS's attained recorded in Table 5.2 and the corresponding processing times in Table 5.3. Figure 5.2 displays the evolution of the estimated ESS over a particular run of the algorithm. Figure 5.3 shows the estimated intensity at each time t_n, given data up to time t_n. From Table 5.1, it is clear to see that, for the saturated and tempered samplers, an increase in M results in a decrease in the resampling rates, i.e. a decrease in sampler degeneracy, as expected. It is also plain to see from Table 5.2 that, as N increases, so does the minimum estimated ESS, and thus the reliability of the estimates. From Tables 5.1 and 5.2, Figure 5.3 and comparing Figure 5.2 to Figure 5.1 it is clear that the saturated and tempered samplers significantly outperformed the benchmark sampler.

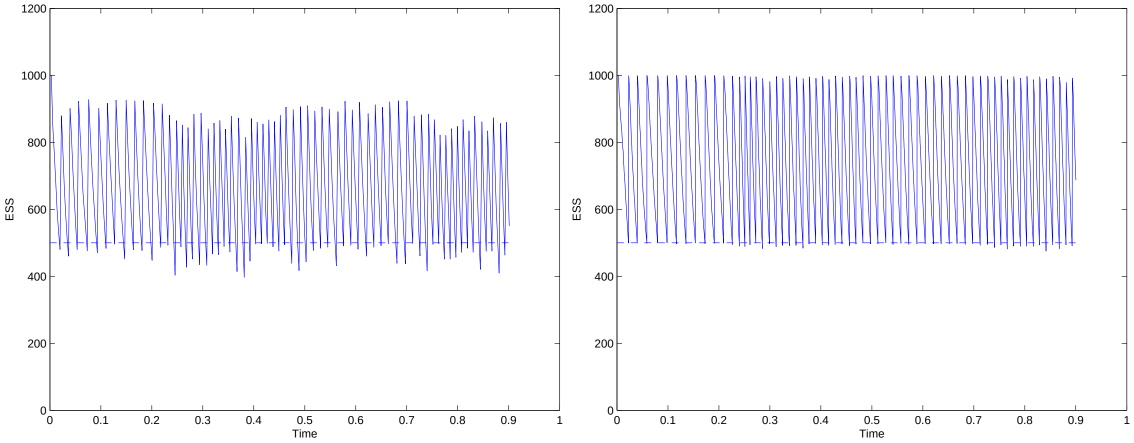
We use the posterior medians to report intensities. Since we have access to a 'true' intensity process, the accuracy of these estimated intensity process is measured

	M = 1		M = 5		M = 20	
	N=100	N=1000	N=100	N=1000	N=100	N=1000
Benchmark	1.0	1.0	1.0	1.0	1.0	1.0
Benchmark - RCC	1.0	1.0	1.0	1.0	1.0	1.0
Saturated	38.1	410.2	38.6	397.0	38.6	398.9
Saturated - RCC	38.5	401.2	40.6	394.4	43.0	425.9
Tempered	47.6	484.7	47.7	475.5	47.9	483.4
Tempered - RCC	47.8	475.7	48.4	481.7	48.3	486.6

Table 5.2: Minimum estimated ESSs calculated during implementation of each of the three SMC samplers and their reduced computational complexity alternatives, for the six algorithm parameterisations that were tested.

	M = 1		M = 5		M = 20	
	N=100	N=1000	N=100	N=1000	N=100	N=1000
Benchmark	612.9	9689.1	2849.7	45690.4	13352.1	144621.3
Benchmark - RCC	449.0	7910.9	1132.7	10657.6	3106.2	31208.5
Saturated	1125.3	10667.8	3234.3	39061.1	15381.9	141817.3
Saturated - RCC	637.5	6215.2	1200.7	11412.6	4391.9	47662.8
Tempered	1160.2	10633.4	3138.4	38679.6	14086.7	130899.1
Tempered - RCC	666.0	6424.4	1156.3	11209.1	3231.3	34795.3

Table 5.3: Processing times, in seconds, for each of the three samplers and their reduced computational complexity alternatives, for the six algorithm parameterisations that were tested.



(a) Saturated

(b) Tempered

Figure 5.2: Estimated effective sample sizes, calculated at each time step of the SMC samplers with state space saturation (left) and data point tempering (right), run with $N = 1000$ particles and with $M = 5$ MCMC sweeps at each iteration. The dashed line indicates the resampling threshold at $N/2 = 500$ particles; the corresponding resampling rates are 20.1% for the saturated sampler and 1.9% for the tempered sampler.

using the root mean square error (RMSE). Table 5.4 presents the RMSEs of the intensity estimates (given the data up to t_n, averaged over each t_n) and Table 5.5 presents the RMSEs of the smoothed (conditional upon the entire data set) intensity estimates resulting from each of the three samplers and their RCC alternatives. The most important result to note is the performance of the saturated and tempered samplers in comparison with the unaltered sampler. As can be seen in terms of accuracy for intensity estimates, the two proposed alterations to the sampler improve the performance consistently and significantly. Looking at the resampling rates and processing times, in Tables 5.1 and 5.3 respectively, we can see that, as expected, although the tempered sampler resampled the particles significantly less than the benchmark sampler, the individual incorporation of each data point resulted in a greater computational cost. These two aspects of the benchmark and tempered samplers appear to have countered each other, resulting in their processing times being largely similar.

We consider also the effect that changes in M and N have on the accuracy of estimates provided by the saturated and tempered samplers. For the saturated and tempered samplers, the results in Tables 5.4 and 5.5 corroborate the expected improvement in accuracy, in both for the sequential estimates at t_n given data up-to t_n and smoothed estimates (given the entire data), that results from an increase in the number of particles used. Whilst for the sequential estimates, there is no clear improvement in accuracy with increasing M, an improvement can be seen in the accuracy of the smoothed estimates.

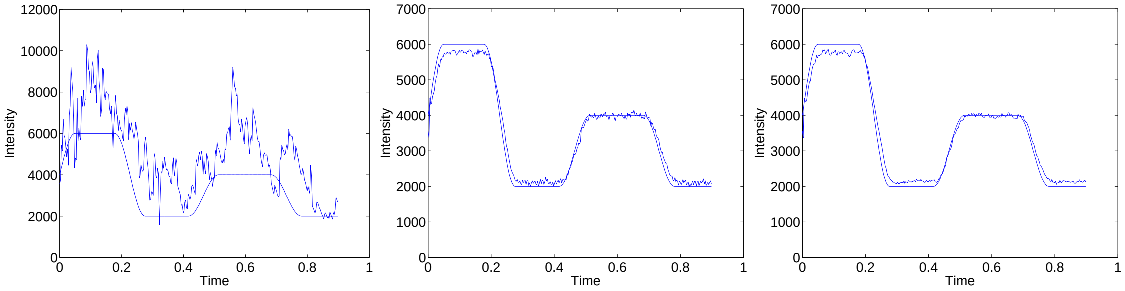
	M = 1		M = 5		M = 20	
	N=100	N=1000	N=100	N=1000	N=100	N=1000
Benchmark	688.561	1116.639	620.432	1942.992	1330.232	1501.263
Benchmark - RCC	676.932	2026.956	880.824	2247.313	1472.126	1264.533
Saturated	242.834	192.580	228.390	193.778	237.315	198.223
Saturated - RCC	229.449	189.279	224.692	193.379	225.592	194.623
Tempered	254.396	196.928	247.754	201.681	248.367	202.501
Tempered - RCC	256.012	191.407	227.241	197.043	230.805	200.227

Table 5.4: Table showing the root mean square error of the intensity. This is given the data up to t_n, averaged over each t_n and for each of the three samplers and their reduced computational complexity alternatives, for the six algorithm parameterisations that were tested.

	M = 1		M = 5		M = 20	
	N=100	N=1000	N=100	N=1000	N=100	N=1000
Benchmark	768.702	670.656	495.019	627.909	489.243	571.107
Benchmark - RCC	698.640	1034.890	572.794	572.841	535.004	599.031
Saturated	360.794	264.331	296.953	114.064	153.444	89.397
Saturated - RCC	478.871	265.477	405.767	266.980	468.853	205.243
Tempered	350.015	170.321	271.712	128.078	157.709	81.666
Tempered - RCC	485.825	249.529	475.348	193.898	514.107	180.914

Table 5.5: Table showing the root mean square error of the smoothed intensity process. This is given the entire data set and for each of the three samplers and their reduced computational complexity alternatives, for the six algorithm parameterisations that were tested.

Finally, using the simulated data, we consider the performance of the samplers when limiting the space over which the invariant MCMC kernels are applied, i.e. the RCC alternatives. As can be seen from Table 5.4, the RCC alteration does not



(a) Benchmark (b) Saturated (c) Tempered

Figure 5.3: Estimates (given the data up to t_n) of the intensity of a simulated data set, generated by the benchmark SMC sampler (left) and the samplers with state space saturation (centre) and data point tempering (right), run with $N = 1000$ particles and with $M = 5$ MCMC sweeps at each iteration. The model parameters were $\gamma = 0.001$, $\nu = 150$ and $s = 20$.

sacrifice any accuracy in the estimates of the intensity (given the data up to each time t_n), however it can be seen from Table 5.5 that the accuracy of the smoothed intensity estimates is rather poor. This is to be expected, due to path degeneracy; we note that one cannot estimate static parameters with the RCC approach unless the time window T is quite small.

5.5.2 Real Financial Data

All three samplers were also tested on real financial data, with the RCC alternatives also being used to generate intensity estimates, given the data up to t_n: the share price of ARM Holdings, plc., traded on the LSE was used. The entire data set was of size $r_T = 1819$, $[0, T] = [0, 0.3]$ (represents $3/10$ of a trading day, that is, $3/10$ of 24 hours; the first trade is just after 9am and the last around 16:15.) with $t_n = n{*}0.001$. Genuine financial data is likely to correspond to a more volatile latent intensity process than that which was used to generate the synthetic data set, and so the parameterisation of the target posterior should be chosen such that large jumps in the intensity process are possible, and such that the intensity may also revert quickly to a lower intensity level. Hence, we specify: $\{\gamma, \nu, s\} = \{0.001, 500, 250\}$.

	Data t_n RMSPEs	Smoothed RMSPEs	Processing Times (s)	Resampling Rates
Saturated	2.18876	2.13479	4064.5	39.5%
Saturated - RCC	2.19112	-	2193.1	39.9%
Tempered	2.34671	2.11468	4605.5	19.8%
Tempered - RCC	2.42776	-	2237.3	19.9%

Table 5.6: Table showing the root mean square prediction errors for the intensity estimates (given data up to time t_n and entire data (smoothed)) given by each of the three samplers for the parameter values $N = 1000$, $M = 5$. The RMSPEs for the smoothed intensity estimates given by the RCC alternatives to the samplers are also provided, along with the observed processing times and resampling rates for each sampler.

Each of the samplers were run using $N = 1000$ particles, applying $M = 5$ MCMC sweeps at each iteration, whilst the resampling rates and the minimum estimated ESS obtained for each procedure were monitored to ensure that the algorithms did not collapse.

Clearly, there is no 'known' intensity process against which to compare the point-wise estimates produced by the samplers. In addition, any inverse-duration based representation of the intensity against which useful comparisons could be drawn would involve making assumptions on the smoothness of the intensity process itself. Thus, we turn to measuring the one-step-ahead predictive accuracy of the estimators of the intensity. This is achieved as follows: denoting the intensity estimated over the interval $[t_{n-j}, t_n)$ as $\hat{\lambda}_{n,j}$, one predicts the expected number of ticks in the interval $[t_{n+i-j}, t_{n+i})$ as $\left(\hat{\lambda}_{n,j}\right)^{-1}$ for $i \geq 1$ and $j \geq 1$, where j is the number of periods over which the prediction is made and i is a lag index. The prediction errors are then calculated based on the predicted and observed number of ticks in the period $[t_{n+i-j}, t_{n+i})$; the root mean square prediction error (RMSPE) will be used. We will report on the one-step-ahead estimates ($i = 1$), estimating the intensity over each interval with $j = 1$.

Table 5.6 presents the RMSPEs for the intensity estimates resulting from the samplers and the RCC alternatives. It was observed that, in calculating the RMSPEs for lag indices $i = 1, \ldots, 100$ using each sampler, both the saturated and tempered samplers displayed the smallest error at $i = 1$, i.e. their respective one-step-ahead predictions were more accurate than those made for lags up to 2.64 hours (each observation interval corresponds to 0.0264 days = 1.584 minutes).

The RCC samplers provide significant computational savings and do not seem to degrade substantially, w.r.t. the error criteria. Again, we remark that in general one should not trust the estimates of the RCC, but as seen here, they can provide a guideline for the intensity values.

Finally, an existing intensity estimation procedure for inhomogeneous point processes was sought within Matlab and R, against which to compare the predictive errors of the above samplers; the only available procedure was found within the spatstat package in R. This procedure, however, simply estimates the intensity over an interval as the number of events within the observed interval, scaled to suit a suitable interval of interest (such as the trading day). When applied to the data, the resulting RMSPE for one-step-ahead prediction was found to be 11.2407. As would be expected, the model described here significantly outperforms this naïve estimation procedure.

5.6 Summary

In this chapter we have considered SMC simulation for partially observed point processes and implemented them for a particular doubly stochastic PP. Two solutions were given, one based upon saturating the state-space, which is suitable in a wide variety of applications and data-point tempering which can be used in sequential problems. We also discussed RCC versions of these algorithms, which reduce computation, but will be subject to the path degeneracy problem when including static parameters and considering the smoothing distribution. We saw that the methods can be successful, in terms of weight degeneracy versus the benchmark approach detailed in Del Moral et al. (2007). In addition, for real data it was observed that predictions using the RCC could be reasonable (relative to the normal versions of the algorithms), but caution on using these estimates should be used.

5

Chapter 6

Future Work

6.1 Summary of Contributions

In this thesis, we have presented a number of SMC and PMCMC procedures that facilitate both filtering and smoothing for SSMs in which traditional methods either cannot be applied, or typically perform poorly. In Chapter 3, we presented an ABC approximation of the general SSM, which facilitates the application of both SMC and PMCMC methods in scenarios where the observation density in the exact SSM is intractable. It was also shown in Chapters 3 and 4, that even when the observation density is analytically available, the use of the ABC approximation presented here can offer improvements in accuracy when performing filtering and smoothing with respect to a high dimensional hidden state.

We also considered the use of a rejection kernel within the SMC filtering and smoothing procedures, and it was shown that this can significantly reduce the variance of the resulting estimators, when targeting the ABC approximation of the SSM. This is, to our knowledge, the first empirical verification for finite particle system size N, of the improvement in asymptotic variance that is offered by the use of the rejection kernel (Del Moral, 2004).

In Chapter 5, we considered the class of SSMs in which both the state and observation processes are point processes (PPs), with the latent PP specifying the time-inhomogeneous intensity process of the observed PP. We demonstrated the recognised difficulties that are faced when using SMC samplers to perform inference with respect to these SSMs, and we presented two novel SMC samplers that address these problems.

6.2 Areas for Expansion

We now present a number of ideas for directions in which this work could be extended.

1. Recall that, in specifying the ABC approximation of the SSM, one requires access to the d_y-dimensional pseudo-datum, distributed according to the observation density: $u_t \sim g_t(\cdot|x_t)$. It is remarked that it is not always necessary to be able to sample directly from the distribution of the data, conditional on the current state. Recent work by Murray et al. (2012) indicates that, under certain conditions, one can make use of the following 'collapsed' ABC approximation, introduced by Ehrlich et al. (2012), where one is not required to sample from the likelihood directly. Suppose that the SSM (1.1)-(1.2) can be written as

$$X_t = \zeta_t(X_{t-1}, V_t) \quad t \geq 1,$$

$$Y_t = \xi_t(X_t, Z_t) \quad t \geq 1,$$

   where $\{V_t\}_{t\geq 1}$ and $\{Z_t\}_{t\geq 1}$ are i.i.d. random variables corresponding to the state and observation noise terms, respectively, and $\zeta_t : \mathbb{R}^{d_x} \times \mathbb{R}^{d_x} \longrightarrow \mathbb{R}^{d_x}$ and $\xi_t : \mathbb{R}^{d_x} \times \mathbb{R}^{d_y} \longrightarrow \mathbb{R}^{d_y}$ are functions that may be pointwise evaluated for each $X_t, V_t \in \mathbb{R}^{d_x}$, $Y_t, Z_t \in \mathbb{R}^{d_y}$ at each time point t. Then, assuming one can evaluate the marginal densities of V_t and Z_t, and sample from the corresponding distributions, the joint density of the state and observation noise processes, conditional on the observed data $y_{1:t}$ may be approximated using the collapsed ABC approximation:

$$\pi_t^\epsilon(v_{1:t}, z_{1:t}|y_{1:t}) \propto \prod_{n=1}^{t} G_n^\epsilon\bigg(\xi_n\Big(\zeta^{(n)}(x_0, v_{1:n}), z_n\Big), y_n\bigg)p(v_n)p(z_n).$$

Marginalising over the sampled values of $z_{1:t}$ provides a collapsed ABC approximation of the joint density of the state noise process $V_{1:t}$ conditional on the observed data $y_{1:t}$, which can be transformed into the original ABC filter.

The 'trick' here is that the joint smoothing distribution may be sampled from where the transition and/or observation *densities* have no closed form and cannot easily be sampled from, yet where one may generate samples from the respective noise terms. Such a situation may arise in high dimensions, for example, where the distribution of the data given the latent process may have an extremely complex dependence structure; it may be the case that this can be represented through the use of a function $\xi_t(X_t, Z_t)$, as above, where the high-dimensional Z_t has a much simpler dependence structure. For further examples of areas in which a collapsed SSM may be useful, see the work of Murray et al. (2012).

It would be of interest to explore the filtering and smoothing problems within the framework provided by a collapsed ABC approximation.

2. Recall that the RSMC procedures presented in Section 2.3.2 are motivated by the result of Del Moral (2004), who shows the improvement in asymptotic estimator variance that the RSMC procedures offer over SIR procedures that use multinomial resampling. In Chapters 3 and 4, this advantage of RSMC was also observed in finite-N implementations of the ABC filtering and ABC smoothing procedures. Chopin (2004) shows that the asymptotic estimator variance is reduced when using residual resampling at every time step instead of multinomial resampling. In addition, Douc et al. (2005) show that improvements in the variance of the particle filtering estimators can be obtained by

using either residual or stratified resampling. It would therefore be of interest to:

- directly compare the behaviour of the asymptotic variances when using RSMC and residual resampling; and to
- empirically compare the variance of the estimators produced by finite-N implementations of RSMC procedures and SMC procedures that use residual/stratified resampling schemes.

3. Propositions 3.3.2 and 4.4.1 are only valid under extremely restrictive, strong mixing assumptions, which will typically only hold when the hidden state and observation process evolve on compact or finite state spaces. In the context of establishing the asymptotic variance of the SMC approximation of the prediction filter (1.9), Whiteley (2011) successfully replaces the strong mixing assumptions of Del Moral and Guionnet (2001) with a multiplicative drift condition, weakening the restrictions but maintaining the result. It would be of great interest to see if the multiplicative drift condition could be similarly used to weaken the restrictions on these results concerning the bias of the ABC approximation.

4. In Chapter 4, we considered particle smoothing, which has computational complexity $O(N)$, and forward smoothing, $O(N^2)$, in the ABC framework. The accuracy of the generalised two-filter smoothing procedure ($O(N^2)$; Briers et al., 2010) could similarly be explored within an ABC framework. Recent work by Persing and Jasra (2012) shows that the two-filter smoothing procedure can be executed with computational complexity $O(N)$, and this provides additional motivation, as the accuracy of an ABC two-filter smoothing procedure would be expected to be greater than the ABC particle smoother, with similar computational complexity.

   It would also be of great interest to use an $O(N)$ implementation of the two-filter smoothing procedure as an update scheme for a PMCMC procedure within the ABC framework.

5. The derivative of the log-likelihood, e.g. $\partial(\log g_{t,\theta})/\partial\theta$, can be of great interest within the SSM framework; this was briefly discussed in Section 4.2, in the context of EM algorithms. Some analysis of the properties of this derivative have been established (Tadić and Doucet, 2005, e.g), although little work has focussed on the case where the likelihood is intractable. In this scenario, one could provide an ABC approximation of the derivative, most likely at the cost of an associated deterministic bias. A theoretical analysis of this bias would be of great interest.

References

C. Andrieu, A. Doucet, and C.P. Robert. Computational Advances for and from Bayesian Analysis. *Statistical Science*, 19(1):118–127, 2004.

C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3): 269–342, 2010.

Y.X. Bar-Shalom, R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*. John Wiley & Sons, Inc., 2001.

O.E. Barndorff-Nielsen and N. Shephard. Non-Gaussian Ornstein–Uhlenbeck-based models and some of their uses in financial economics (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63:167–241, 2001.

T. Bengtsson, P. Bickel, and B. Li. Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems. In *Probability and Statistics: Essays in Honor of David A. Freedman*, D. Nolan and T. Speed, editors, volume 2, pages 316–334. Institute of Mathematical Statistics, 2008.

A. Beskos, D. Crisan, and A. Jasra. On the Stability of Sequential Monte Carlo Methods in High Dimensions. Technical report, Imperial College London, 2011.

A. Beskos, D. Crisan, A. Jasra, and N. Whiteley. Error Bounds and Normalizing

Constants for Sequential Monte Carlo in High Dimensions. Technical report, University College London, 2012.

P. Bickel, B. Li, and T. Bengtsson. Sharp failure rates for the bootstrap filter in high dimensions. In *Pushing the Limits of Contemporary Statistics: Contributions in Honor of Jayanta K. Ghosh*, B. Clarke and S. Ghosal, editors, pages 318–329. Institute of Mathematical Statistics, 2008.

M.G.B. Blum, M.A. Nunes, D. Prangle, and S.A. Sisson. A comparitive review of dimension reduction methods in approximate Bayesian computation. Technical report, 2012.

M. Briers, A. Doucet, and S. Maskell. Smoothing Algorithms for State-Space Models. *Annals of the Institute of Statistical Mathematics*, 62:61–89, 2010.

F. Campillo and V. Rossi. Convolution Particle Filter for Parameter Estimation in General State-Space Models. *IEEE Transactions on Aerospace and Electronic Systems*, 45(45):1063–1071, 2009.

B.P. Carlin and S. Chib. Bayesian Model Choice via Markov Chain Monte Carlo Methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 57(3):473–484, 1995.

G. Casella and R.L. Berger. *Statistical Inference.* Duxbury, Thomson Learning, 2nd edition, 2002.

G. Casella and C.P. Robert. Rao-Blackwellisation of Sampling Schemes. *Biometrika*, 83(1):81–94, 1996.

S. Centanni and M. Minozzo. A Monte Carlo approach to filtering for a class of marked doubly stochastic Poisson processes. *Journal of the American Statistical Association*, 101:1582–1597, 2006a.

S. Centanni and M. Minozzo. Estimation and filtering by reversible jump MCMC for a doubly stochastic Poisson model for ultra-high-frequency financial data. *Statistical Modelling*, 6(2):97 – 118, 2006b. ISSN 1471082X.

N. Chopin. Central Limit Theorem for Sequential Monte Carlo Methods and its Application to Bayesian Inference. *The Annals of Statistics*, 32(6):2385–2411, 2004.

N. Chopin, P.E. Jacob, and O. Papaspiliopolous. SMC^2: A sequential Monte Carlo algorithm with particle Markov chain Monte Carlo updates. Technical report, ENSAE-CREST, 2011.

D.R. Cox. Some Statistical Methods Connected with Series of Events. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 17(2):129–164, 1955.

D.J. Daley and D. Vere-Jones. *An Introduction to the Theory of Point Processes.* Springer, 2007.

T.A. Dean, S.S. Singh, A. Jasra, and G.W. Peters. Parameter Estimation for Hidden Markov Models with Intractable Likelihoods. Technical report, University of Cambridge, 2010.

P. Del Moral. *Feynman Kac Formulae. Genealogical and Interacting Particle Systems with Applications.* Springer: New York, 2004.

P. Del Moral and A. Guionnet. On the stability of interacting processes with applications to filtering and genetic algorithms. *Annales de l'Institut Henri Poincaré (B) Probability and Statistics*, 37:155–194, 2001.

P. Del Moral, A. Doucet, and A. Jasra. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006.

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

P. Del Moral, A. Doucet, and S.S. Singh. Forward Smoothing using Sequential Monte Carlo. Technical report, 2009.

P. Del Moral, A. Doucet, and S.S. Singh. A backward particle interpretation of Feynman-Kac formulae. *ESAIM: Mathematical Modelling and Numerical Analysis*, 44(05):947–975, 2010.

R. Douc, O. Cappé, and E. Moulines. Comparison of resampling schemes for particle filtering. In *In 4th International Symposium on Image and Signal Processing and Analysis (ISPA)*, pages 64–69, 2005.

A. Doucet and A.M. Johansen. *The Oxford Handbook of Nonlinear Filtering*, chapter A Tutorial on Particle Filtering and Smoothing: Fifteen Years Later. Oxford University Press, 2011.

A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10:197–208, 2000.

A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice.* Springer, 2001.

A. Doucet, L. Montesano, and A. Jasra. Optimal Filtering For Partially Observed Point Processes Using Trans-Dimensional Sequential Monte Carlo. In *Proceedings on IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 5, pages 597–600, 2006.

B. Efron. The Jackknife, the Bootstrap, and Other Resampling Plans. In *CBMS-NSF Regional Conference Series in Applied Mathematics*, volume 38. SIAM, 1982.

E. Ehrlich, A. Jasra, N. Kantas, and S.S. Singh. Online Static Parameter Estimation for ABC Approximations of Hidden Markov Models. Technical report, Imperial College London, 2012.

G. Evensen. Inverse Methods and Data Assimilation in Nonlinear Ocean Models. *Physica D*, 177:108–129, 1994.

P. Fearnhead. Filtering recursions for calculating likelihoods for queues based on inter-departure time data. *Statistics and Computing*, 14:261–266, 2004. ISSN 0960-3174.

S. Geman and D. Geman. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

W.R. Gilks and C. Berzuini. Following a Moving Target - Monte Carlo Inference for Dynamic Bayesian Models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63:127–146, 2001.

S.J. Godsill, A. Doucet, and M. West. Monte Carlo Smoothing for Nonlinear Time Series. *Journal of the American Statistical Association*, 99(465):156–168, March 2004.

N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation. In *IEEE Proceedings: F Radar and Signal Processing*, volume 140, pages 107–113, 1993.

P.J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

W.K. Hastings. Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika*, 57(1):97–109, 1970.

A. Jasra, D.A. Stephens, and C.C. Holmes. On population-based simulation for static inference. *Statistics and Computing*, 17:263–279, 2007.

A. Jasra, S.S. Singh, J.S. Martin, and E. McCoy. Filtering via Approximate Bayesian Computation. *Statistics and Computing*, 2010.

S.J. Julier and J.K. Uhlmann. A New Extension of the Kalman filter to Nonlinear Systems. In *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls*, volume 3, page 26, 1997.

R.E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82:35–45, 1960.

R.E. Kalman and R.S. Bucy. New results in linear prediction and filtering theory. *Journal of Basic Engineering*, 83(1):95–108, March 1961.

G. Kitagawa. Monte Carlo Filter and Smoother for Non-Gaussian and Nonlinear State Space Models. *Journal of Computational and Graphical Statistics*, 5(1): 1–25, 1996.

A. Kong, J.S. Liu, and W.H. Wong. Sequential Imputations and Bayesian Missing Data Problems. *Journal of the American Statistical Association*, 89(425):278–288, 1994. ISSN 01621459.

J. Lei and P. Bickel. Ensemble Filtering for High-Dimensional Non-linear State Space Models. Technical report, UC Berkeley, 2009.

J.S. Liu. Metropolized independent sampling with comparisons to rejection sampling and importance samplng. *Statistics and Computing*, 6:113–119, 1996.

J.S. Liu. *Monte Carlo Strategies in Scientific Computing.* Springer, NY, 2001.

J.S. Liu and R. Chen. Sequential Monte Carlo Methods for Dynamic Systems. *Journal of the American Statistical Association*, 93(443):1032–1044, 1998. ISSN 01621459.

J.-M. Marin, P. Pudlo, C.P. Robert, and R.J. Ryder. Approximate Bayesian Computational Methods. Technical report, Université Montpellier, 2011.

P. Marjoram, J. Molitor, V. Plagnol, and S. Tavaré. Markov Chain Monte Carlo Without Likelihoods. *Proceedings of the National Academy of Sciences*, 100(26): 15324–15328, 2003.

J.S. Martin, A. Jasra, and E. McCoy. Inference for a Class of Partially Observed Point Process Models. *Annals of the Institute of Statistical Mathematics (to appear)*, 2012a.

J.S. Martin, A. Jasra, S.S. Singh, N. Whiteley, and E. McCoy. Approximate Bayesian Computation for Smoothing. Technical report, Imperial College London, 2012b.

N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, and A.H. Teller. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

S. Meyn and R. Tweedie. *Markov Chains and Stochastic Stability.* Cambridge University Press, 2nd edition, 2009.

L.M. Murray, E.M. Jones, and J. Parslow. On Collapsed State-space Models and the Particle Marginal Metropolis-Hastings Sampler. Technical report, CSIRO Mathematics, Informatics and Statistics, 2012.

D. J. Nott, L. Marshall, and T.M. Ngoc. The ensemble Kalman filter is an ABC algorithm. *Statistics and Computing*, (to appear), 2012.

A. Persing and A. Jasra. On the estimate of the normalizing constant for Feynman-Kac formulae via the generalized two-filter formula. Technical report, Imperial College London, 2012.

M.K. Pitt and N. Shephard. Filtering via Simulation: Auxiliary Particle Filters. *Journal of the American Statistical Association*, 94(446):590–599, 1999.

G. Poyiadjis, A. Doucet, and S. Singh. Particle approximations of the score and observed information matrix in state space models with application to parameter estimation. *Biometrika*, 98(1):65–80, 2011.

J.K. Pritchard, M.T. Seielstad, A. Perez-Lezaun, and M.W. Feldman. Population Growth of Human Y Chromosomes: A Study of Y Chromosome Microsatellites. *Molecular Biology and Evolution*, 16(12):1791–1798, 1999.

C.P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, NY, 2004.

G. O. Roberts, A. Gelman, and W. R. Gilks. Weak Convergence and Optimal Scaling of Random Walk Metropolis Algorithms. *The Annals of Applied Probability*, 7(1): 110–120, 1997.

G.O. Roberts and J.S. Rosenthal. General State Space Markov Chains and MCMC Algorithms. *Probability Surveys*, 1:20–71, 2004.

G.O. Roberts, O. Papaspiliopolous, and P. Dellaportas. Bayesian Inference for Non-Gaussian Ornstein-Uhlenbeck Stochastic Volatility Processes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 66(2):369–393, 2004.

M. Rousset and A. Doucet. Discussion of "Exact and Computationally Efficient Likelihood-Based Estimation for Discretely Observed Diffusion Processbes" by A. Beskos, O. Papaspiliopolous, G.O. Roberts and P. Fearnhead. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):333–382, 2006.

T.H. Rydberg and N. Shephard. *Nonlinear and Nonstationary Signal Processing*, chapter A modelling framework for the prices and times of trades made on the New York stock exchange, pages 217–246. Cambridge University Press, 2000.

A. Shiryaev. *Probability*. Springer-Verlag: New York, 1996.

V. Šmídl and A. Quinn. Variational Bayesian Filtering. *IEEE Transactions on Signal Processing*, 56(10 Part 2):5020–5030, 2008.

A.F.M. Smith and A.E. Gelfand. Bayesian Statistics Without Tears: A Sampling-Resampling Perspective. *The American Statistician*, 46(2):84–88, 1992.

C. Snyder, T. Bengtsson, P. Bickel, and J. Anderson. Obstacles to High-Dimensional Particle Filtering. *Monthly Weather Review*, 136:4629–4640, 2008. doi: 10.1175/2008MWR2529.1.

D.L. Snyder. Filtering and Detection for Doubly Stochastic Poisson Processes. *IEEE Transactions on Information Theory*, 18:91–102, 1972.

D.L. Snyder and M.I. Miller. *Random Point Processes in Space and Time*. Springer-Verlag: New York, 1998.

H.W. Sorenson and A.R. Stubberud. Non-linear Filtering by Approximation of the *a posteriori* Density. *International Journal of Control*, 8(1):33–51, 1968.

V.B. Tadić and A. Doucet. Exponential forgetting and geometric ergodicity for optimal filtering in general state-space models. *Stochastic Processes and their Applications*, 115:1408–1436, 2005.

S. Tavaré, D.J. Balding, R.C. Griffiths, and P Donnelly. Inferring coalescence times from DNA sequence data. *Genetics*, 145(2):505–518, 1997.

L. Tierney. Markov Chains for Exploring Posterior Distributions. *The Annals of Statistics*, 22(4):1701–1728, 1994.

L. Tierney. A Note on Metropolis-Hastings Kernels for General State Spaces. *The Annals of Applied Probability*, 8(1):1–9, 1998.

E. Varini. A Monte Carlo method for filtering a marked doubly stochastic Poisson process. *Statistical Methods and Applications*, 17:183–193, 2008.

J. Vermaak, C. Andrieu, A. Doucet, and S.J. Godsill. Particle Methods for Bayesian Modeling and Enhancement of Speech Signals. *IEEE Transactions on Speech and Audio Processing*, 10(3):173–185, 2002.

N. Whiteley. Stability properties of some particle filters. Technical report, University of Bristol, 2011.

N. Whiteley, A.M. Johansen, and S.J. Godsill. Monte Carlo Filtering of Piecewise Deterministic Processes. *Journal of Computational and Graphical Statistics*, 20: 119–139, 2011.