Iterated Filtering and Smoothing with Application to Infectious Disease Models

by

Dao X. Nguyen

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Statistics)
in The University of Michigan
2016

Doctoral Committee:

Professor Edward L. Ionides, Chair
Associate Professor Yves F. Atchade
Associate Professor Aaron A. King
Associate Professor Stilian A. Stoev

For my dearest people

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my supervisor, Professor Edward Ionides, for his patience, guidance, assistance and encouragement throughout my graduate study. His scientific insight and perspectives have made me understand simulation-based inference, time series analysis, partially observed stochastic dynamic systems, especially computational methods in statistics. Without him, I would not have been able to learn the scientific skills required to be a scientist. Learning with him has been one of the most enriching and fruitful experiences of my life. My sincere thanks goes to the committee members. Professors Yves Atchade, Stilian Stoev and Aaron King have helped me a lot in thinking critically about my work, and most importantly to write good quality of scientific output. I would also like to thank Professors Vijay Nair and Long Nguyen for giving me the opportunity to join the Statistics Department. Thanks are also due to various professors, administrative staff and graduate students, whom I have shared with them, the most productive working environments. Finally, my deepest thanks, go to my parents, my wife, my son and my daughter, for their unconditional support and encouragement throughout many years.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

ABSTRACT

Iterated Filtering and Smoothing with Application to Infectious Disease Models

by

Dao X. Nguyen

Chair: Edward L. Ionides

Partially observed Markov process (POMP) models are ubiquitous tools for modeling time series data in many fields including statistics, econometrics, ecology, and engineering. Because of incomplete measurements, and possibly weakly identifiable parameters, making inferences on POMP models can be challenging. Standard methods for inference (e.g., maximum likelihood) with restrictive assumptions of linear Gaussian models have often led to unsatisfactory results when the assumptions are violated. To relax these assumptions, this dissertation develops a class of simulation-based algorithms called *iterated filtering and smoothing* for POMP models. First, a novel filter, called Bayes map iterated filtering, is introduced. This filter recursively combines parameter perturbations with latent variable reconstruction, stochastically optimizing the approximated likelihood of latent variable models and providing an assymptotic guarantee of the performance of this inference methodology. Second, a fast, light-weight algorithm, called second-order iterated smoothing is proposed to improve on the convergence rate of the approach. The goal of this part is to demonstrate that by exploiting Fisher Information as a by-product of the inference methodology, one can theoretically achieve both statistical and computational efficiencies without

sacrificing applicability to a general class of models. Third, a new technique for the proof of Bayes map iterated filtering algorithm, based on super-martingale inequality, is proposed. This approach with verifiable conditions is simpler than the previous approach and is generalizable to more sophisticated algorithms. Fourth, we validated the properties of the proposed methodologies through applying them to a challenging inference problem of fitting a malaria transmission model with control to time series data, finding substantial gains for our methods over current alternatives. Finally, a range of modern statistical methodologies for POMP modeling have been implemented in an open source R package, named **pomp**, to provide a flexible computational framework for the community.

CHAPTER I

Introduction

1.1 Motivation

Partially observed Markov process (POMP) models, which are synonymous with hidden Markov models or state space models, are defined as a doubly stochastic process where the underlying stochastic process can only be observed through another stochastic process (*Rabiner and Juang*, 1986). The past decade has seen the rapid development of POMP modeling in many fields including engineering, ecology and statistics. The reason could be that POMP modeling is especially appealing mechanisms for inference because most data are partially observable in nature. However, POMP modeling is hindered by the possibility of weak identifiability, which is often plagued by incomplete or noisy measurements. Thus, except when applied to certain relative small, or approximately linear and Gaussian, state-of-the-art statistical methods are needed to make efficient use of available data and to facilitate model criticism.

Inference methodology for stochastic dynamic systems is often termed "plug-and-play" if only simulated estimation is needed to plug into the inference procedure (*Bretó et al.*, 2009; *He et al.*, 2010). The plug-and-play methodology has been used to free researchers from demanding closed-form expression requirements for transition probabilities, imposed by previously available statistical methodology, allowing

researchers to broaden classes of modeling and considering novel hypotheses. Unlike the mainstream statistical techniques (Expectation-Maximization algorithms and Bayesian Markov Chain Monte Carlo), plug-and-play approaches are a relatively recent and exciting development because of their less restrictive requirements. Examples of plug-and-play methodologies follow the frequentist paradigm (*Ionides et al.*, 2011; *Lindström et al.*, 2012), the Bayesian paradigm (*Andrieu et al.*, 2010; *Toni et al.*, 2009), or work by matching selected summary statistics (*Wood*, 2010). Amongst these approaches, we are interested in iterated filtering, a frequentist plug and play method that applies a sequence of standard sequential Monte Carlo (SMC) filtering algorithms to recursively locate the maximum likelihood estimator of unknown system parameters. The reason is that likelihood maximization provides a general platform for hypothesis testing, interval estimation, and diagnosis of model misspecification. By making likelihood maximization computationally feasible for increasingly large systems, iterated filtering has provided computational savings. The methodological difficulties carrying out inference for partially observed stochastic dynamic system models are primarily computational. Although, the fundamental theories of likelihood-based or Bayesian inference are generally applicable to POMP, at least in principle, they aren't frequently used because of their computational costs. A modern and popular fully-Bayesian plug-and-play sequential Monte Carlo based method, (*Andrieu et al.*, 2010) considers only a single value of the model parameter vector at each of the numerically intensive integration steps, which imposes considerable computational cost. By contrast, iterated filtering obtains massive computational savings over alternative Monte Carlo methodologies by searching the model parameter space simultaneously with integrating out over latent dynamic state variables.

It is well-known that iterated filtering theory is based on stochastic approximation, for which the convergence rate is suboptimal. It is desirable to increase the convergence rate of this class of algorithms to enlarge the applicability in real-world

problems. Therefore, it is essential to improve the convergence rate of this class of inference algorithms while enjoying its advantages.

1.2 Contribution

There are several key contributions. The first contribution is a novel theoretical framework for a Bayes map iterated filtering and a new algorithm construction that dramatically outperforms previous approaches on a challenging inference problem in disease ecology. In order to increase empirical convergence rate for Bayes map iterated filtering, we generalize the idea of data cloning (*Lele et al.*, 2007) and the classical iterated filtering (*Ionides et al.*, 2011). While the core description of the iterated filtering is based on conditional moments of the perturbed parameters to approximate derivatives of the log-likelihood function, the proof of this method is particularly dealing with convergence of an iterated Bayes map. Specifically, it has been shown that if we apply Bayes map iteratively, we finally get a good approximation of the maximum likelihood. From a practical point of view, we find this new algorithm can lead to substantial numerical improvements in the process of inferring parameters of a POMP model. Methods that are not based on local polynomial approximations to the likelihood surface can be advantageous when the likelihood surface has nonlinear ridges, a situation that is highlighted by both the toy example and the scientific example in the manuscript.

The second contribution is the use of more natural random walk noise instead of independent white noise in the framework of iterated smoothing (*Doucet et al.*, 2013). The focus in this part is on presenting how to improve empirical convergence rate without increasing computational work load. It is an open problem whether the approach proposed by Doucet et al. can be applied in practice, while we find a theoretical simplification, which leads directly to a computationally simpler algorithm. Furthermore, it can be shown that this Taylor-series based algorithm remains com-

petitive with the iterated Bayes map approach. This approach therefore provides an alternative platform for potential future theoretical and practical progress on tackling this class of inference problems. Based theoretically on proofs of *Doucet et al.* (2013), we introduce a theoretical development that our empirical results suggest is critical for turning the insights into a computationally efficient algorithm. We present evidences for the computational capabilities of our algorithm on challenging scientific problems. By contrast, (*Doucet et al.*, 2013), to the best of our knowledge, did not demonstrate practical applications for their algorithm. In addition, we try to implement *Doucet et al.*'s algorithm and its variation closest to their theory and compare performances amongst different approaches. The comparison confirms that the random walk noise approach explores the likelihood surface more efficiently than the independent white noise approaches, a situation that is illustrated by both the toy and the scientific examples.

The third contribution is the use of super martingale theory, to prove the convergence of Bayes map iterated filtering. Using this general proof technique, which only relies on easily verifiable conditions, is of general interest. Such approaches, are simple, elegant and generalizable to more sophisticated algorithms. We also apply it to a more challenging data analysis, difficult to analyze before. Hence, we demonstrate it as a viable method for some challenging models, including causal inference system under reactive intervention, which severely violate the conditional independence of POMP model. In order to avoid conditional independence violation, we propose to use proper weight for a SMC proposal density. We verify the performance improvement of the method on a toy problem of stochastic volatility with return. The results confirm our previous finding that Bayes map iterated filtering is an effective plug and play approach and it is applicable in the new modeling framework.

Finally, in order to evaluate these proposed algorithms, extensive experiments have been conducted successfully on an infectious diseases datasets obtained from Na-

tional Institute of Malaria Research in India in collaboration with Mercedes Pascual. R package **pomp** (*King et al.*, 2015c) is used as modeling framework and inference environment for general data analysis implementation. Various cluster and paralellization libraries are used for the proposed algorithms. Experimental results show that the proposed algorithms improve substantially the convergence rate for a given computational budget. Most importantly, it provides a scalable framework for POMP models to deal with most real world datasets. Some recent Bayesian approaches also try to exploit the plug and play properties, using computational power of parallelization, however, our method is believed to give a very potential application by using approximated maximum likelihood with much simpler computational expenses. At the end, a conclusion with some open issues and future work are presented.

Our contribution to the statistical communities includes:

1. Iterated Bayes map filtering with better convergence rate and less computational time.

2. A light-weight second-order iterated smoothing with competitive computational performance.

3. A new elegant generalizable proof of Bayes map iterated filtering.

4. Contributing to open source R packages pomp.

5. Develop open source R packages, is2.

6. Scientific data analysis of models with feedback controls with application to malaria with controls datasets.

1.3 Overview of the dissertation

The dissertation is organized as follows.

Chapter 1 gives motivation, objectives and contributions of the dissertation.

Chapter 2 introduces a background of partially observed Markov model. The stochastic approximation, sequential Monte Carlo approximation of the likelihood and original iterated filtering with some brief mathematical formalizations of the conventional framework. This constitutes the theoretical foundations of the dissertation.

Chapter 3, presents a novel Bayes map iterated filtering, which is capable of improving convergence rate of the simulation-based inference approach by iteratively combine parameter perturbations with latent variable reconstruction.

Chapter 4 then focuses on developing a novel theoretical justification for the second order iterated smoothing algorithm. The main theoretical results show that we can approximate the first and the second-order derivative of the log-likelihood function using conditional moments. The approach of *Doucet et al.* (2013) can be modified to construct an algorithm which carries out smoothing using random walk perturbations, with the happy result that some computationally demanding covariance terms cancel out and do not have to be computed.

Chapter 5 first reprove Bayes map iterated filtering using supermartingale theory. It then proposes a framework for inference on mechanistic model, accounting for control effects. The chapter includes the carried out experiments. The results are carefully analyzed and discussed in details. It concludes at the end with open issues and future works.

Chapter 6 provide a software environment that can effectively handle broad classes of POMP models and take advantage of the wide range of statistical methodologies that have been proposed for such models. The pomp software package (*King et al.*, 2015c) differs from previous approaches by providing a general and abstract representation of a POMP model. The chapter also illustrates the specification of more complex POMP models, using a nonlinear epidemiological model with a discrete

population, seasonality, and extra-demographic stochasticity. It discusses the specification of user-defined models and the development of additional methods within the programming environment provided by pomp.

CHAPTER II

Background on Iterated Filtering Algorithms

2.1 Partially Observed Markov Model

We use capital letters to denote random variables and lower case letters to denote their values. Let $\{X(t),\ t \in \mathbb{T}\}$ be a Markov process with $X(t)$ taking values in a measurable space $\mathcal{X}$. The time index set, $\mathbb{T} \subset \mathbb{R}$, may be an interval or a discrete set and contains a finite subset $t_1 < t_2 < \cdots < t_N$ at which $X(t)$ is observed, together with some initial time $t_0 < t_1$. We write $X_{0:N} = (X_0,\ \ldots,\ X_N) = (X(t_0),\ \ldots,\ X(t_N))$. Hereafter for any generic sequence $\{X_n\}$, we shall use $X_{i:j}$ to denote $(X_i,\ X_{i+1},\ \ldots,\ X_j)$. The distribution of $X_{0:N}$ is characterized by the initial density $X_0 \sim \mu(x_0;\theta)$ and the condition density of X_n given X_{n-1}, written as $f_n(x_n|x_{n-1};\theta)$ for $1 \le n \le N$. Here, θ is an unknown parameter in $\mathbb{R}^d$. The process $\{X_n\}$ is only observed through another process $\{Y_n, n = 1, \ldots, N\}$ taking values in a measurable space $\mathcal{Y}$. The observations are assumed to be conditionally independent given $\{X_n\}$, and their probability density is of the form

$$p_{Y_n|Y_{1:n-1},\ X_{0:n}}(y_n|y_{1:n-1},\ x_{0:n};\theta) = g_n(y_n|x_n;\theta),$$

for $1 \le n \le N$. We assume that $X_{0:N}$ and $\mathrm{Y}_{1:N}$ have a joint density $p_{X_{0:N},\ Y_{1:N}}(x_{0:N},\ y_{1:N};\theta)$ on $\mathcal{X}^{N+1} \times \mathcal{Y}^N$. The data are a sequence of observations by $y^*_{1:N} = (y^*_1,\ \ldots,\ y^*_N) \in \mathcal{Y}^N$,

considered as fixed. We write the log likelihood function of the data for the POMP model as $\ell(\theta)$, given by

$$\begin{aligned}\ell(\theta) &= \log p_{Y_{1:N}}(y^*_{1:N};\theta)\\ &= \log \int \mu(x_0;\theta)\prod_{n=1}^{N} f_n(x_n|x_{n-1};\theta)\ g_n(y^*_n|x_n;\theta)\ dx_{0:N}.\end{aligned}$$

2.2 Stochastic Approximation

Stochastic approximation, first introduced in 1951 by *Robbins and Monro* (1951), has been subject to enormous literature, including statistical computation. The stochastic approximation of *Robbins and Monro* procedure is to find zeros of function $g(x)$, which can only be computed through noisy observations. The basic paradigm is a stochastic difference equation of the form $\theta_{n+1} = \theta_n + \epsilon_n Y_n$, where θ_n takes its values in some Euclidean space, Y_n is a random variable. The "step size" $\epsilon_n > 0$ is small and might go to zero as $n \to \infty$. The parameter θ is to be estimated to meet a goal asymptotically given that the random vector Y_n is a noise-corrupted observations sequence taken on the system when the parameter is set to θ_n. A major insight of *Robbins and Monro* was that, if the step sizes ϵ_n goes to zero slow enough as $n \to \infty$, then noise will be canceled out in the long run through implicit averaging of the observation. The Robbins Monro algorithm is essentially a recursive procedure for finding the root of a real value function $g(\cdot)$. It turns out to be a classical one in numerical analysis and Newton's procedure if $g(\cdot)$ is known and continuously differentiable. The sequence θ_n is computed recursively as

$$\theta_{n+1} = \theta_n - [g'(\theta_n)]^{-1} g(\theta_n), \qquad n = 1, 2, ... \tag{2.1}$$

where $g'(\cdot)$ denotes the derivative of $g(\cdot)$ with respect to θ. Suppose that $g(\theta) < 0$ for $\theta > \theta^*$, and $g(\theta) > 0$ for $\theta < \theta^*$, and that $g'(\theta)$ is strictly negative and is bounded

in a neighborhood of θ^*. Then θ_n converges to θ^* if the initial value θ_1 is in a small enough neighborhood of θ^*. In general, $g(\cdot)$ is neither differentiable nor known, the estimation is often replaced by a good approximation approach such as Monte-Carlo estimation. If the goal is to stochastically estimate the maximum of a function $g(\theta)$ where θ is d dimensional parameter. It is equivalent to finding the zeros points of the gradient $\nabla g(\theta)$ or *Kiefer and Wolfowitz* (1952)(KW) method. Let $c_n \to 0$ denote the finite difference width used for gradient approximation, ϵ_n denote the step size and $Y_{n,i}$ denote the observation taken at time n at parameter value θ_n along the ith unit vector direction e_i. We define θ_n recursively by

$$\theta_{n+1} = \theta_n + \epsilon_n Y_n, \tag{2.2}$$

where $Y_n = (Y_{n,1}, ..., Y_{n,d})$ and

$$Y_{n,i} = \frac{g(\theta_n + c_n e_i) - g(\theta_n - c_n e_i)}{2c_n}. \tag{2.3}$$

Additional difficulties arise due to bias in estimating $\nabla g(\theta)$. Let $g_i'(\theta)$, $g_i''(\theta)$,$g_i'''(\theta)$ denote the first, second and third derivative of g with respect to ith component of θ. Suppose $g_i'''(\theta)$ are continuous. By a Taylor expansion,

$$g\left(\hat{\theta}_n + c_n e_i\right) = g\left(\hat{\theta}_n\right) + c_n g_i'\left(\hat{\theta}_n\right) + \frac{1}{2}c_n^2 g_i''\left(\hat{\theta}_n\right) + \frac{1}{6}c_n^3 g_i'''\left(\bar{\theta}_n^{(i+)}\right), \tag{2.4}$$

$$g\left(\hat{\theta}_n - c_n e_i\right) = g\left(\hat{\theta}_n\right) - c_n g_i'\left(\hat{\theta}_n\right) + \frac{1}{2}c_n^2 g_i''\left(\hat{\theta}_n\right) - \frac{1}{6}c_n^3 g_i'''\left(\bar{\theta}_n^{(i-)}\right), \tag{2.5}$$

where $\bar{\theta}_n^{(i\pm)} = \hat{\theta}_n + \lambda^{\pm} c_n e_i$ for some $\lambda^{\pm} \in [0,1]$. Assume that the difference of noise term in any direction has a conditional mean zero, then by some calculation

$$E\left[Y_{n,i} | \hat{\theta}_0, ..., \hat{\theta}_n\right] = g_i'\left(\hat{\theta}_n\right) + b_{n,i}, \tag{2.6}$$

where $b_{n,i}$ is the ith term of the bias. Assume that $g_i''' \left(\bar{\theta}_n^{(i\pm)}\right)$ are bounded, this implies the bias $b_{n,i} = O\left(c_n^2\right)$. The algorithm converges if $b_n \to 0$. For example, if g is a quadratic loss function, it always holds.

2.3 Adaptive Stochastic Approximation

Stochastic approximation approach is in a sense a stochastic generalization of the gradient descent method. As a result, it is often possible to improve the speed of convergence by stochastic generalization of Newton method using Hessian of the objective functions. It is stated that this can lead to an asymptotically optimal convergence rate (*Gill et al.*, 1981). However, the computation of Hessian by supplying second derivative and matrix inversion is burdensome, Quasi-Newton methods can replace this by simpler updating formula (*Fletcher*, 1980). Suppose that the objective function is to minimize $r(x) = \|g(x)\|^2 = \sum_{i=1}^{d} g_i^2(x)$ where $g(x) = (g_1(x), ..., g_d(x))$, $x \in R^d, n \geq d$. Let $D(x)$ be the $d \times d$ derivative of $g(x)$. Then the derivative of $\|g(x)\|^2$ is $2D^T(x)\, g(x)$. The Hessian matrix of $\|g(x)\|^2$ is

$$H(x) = 2\left[D^T(x)\, D(x) + \sum_{i=1}^{d} H^{(i)}(x)\, g_i(x)\right], \tag{2.7}$$

where g_i is ith coordinate of g and $H^{(i)}$ is the Hessian of g_i. Thus one can by pass the expensive computation of Hessian either by general quasi Newton method (*Fletcher*, 1980) or by Gauss-Newton method which exploit the special structure of $r(x)$.

$$\nabla^2 r(x) = 2\sum_{i=1}^{n}\left[(\nabla g_i(x))(\nabla g_i(x))^T + \left(\nabla^2 g_i(x)\right) g_i(x)\right], \tag{2.8}$$

and since $f_i(x) \approx 0$ is near the minimum of $r(x)$,

$$\nabla^2 r(x) \approx 2\sum_{i=1}^{n}(\nabla g_i(x))(\nabla g_i(x))^T. \tag{2.9}$$

Note that $\nabla r(x) = 2D^T(x)\, g(x)$ where $D^T(x) = (\nabla f_1(x), ..., \nabla f_n(x))$. The Gauss-Newton recursion is

$$x_{n+1} = x_n - [D^T(x_n)\, D(x_n)]^{-1} D^T(x)\, g(x). \tag{2.10}$$

$D(x_n)$ is assumed to have rank k, when $n = k$, $D(x)$is invertible and

$$[D^T(x_n)\, D(x_n)]^{-1} D^T(x) = D^{-1}(x). \tag{2.11}$$

Hence, adaptive stochastic approximation can be a simple form of

$$x_{n+1} = x_n - D^{-1}(x)\, g(x). \tag{2.12}$$

In the context of parameter estimation, exploiting the structure of iterated filtering, the Hessian can be by passed for free, while the optimal convergence rate of adaptive stochastic approximation can be achieved.

2.4 Data Cloning

The data cloning method (*Lele et al.*, 2007) can briefly described as follows. Consider the latent variable model $\underline{X} \sim f(\underline{x}|\theta)$ with $\theta \in \Theta$, where $\Theta = R^d$ denoted the parameter space. Let $\underline{y} \sim g\left(\underline{y}|\theta\right)$ be the observation model. We have the likelihood

$$L(\theta, \underline{y}) = \int g\left(\underline{y}|\underline{X}, \theta\right) f(\underline{X}|\theta)\, d\underline{X}, \tag{2.13}$$

and we want to compute the MLE. Let $\ell(\theta) = \log L(\theta)$,

$$\theta_* = \arg\max_{\theta \in \Theta} \ell(\theta). \tag{2.14}$$

Let $\pi(\theta)$ be the prior distribution on the parameter space Θ, using the posterior distribution

$$\pi_n\left(\theta|\underline{y}\right) = \frac{\left[g\left(\underline{y}|\theta\right)\right]^n \pi(\theta)}{\int \left[g\left(\underline{y}|\theta\right)\right]^n \pi(\theta)\, d\theta}. \tag{2.15}$$

The authors show that under some regularities, if $\theta_n \sim \pi_n$, then

$$\theta_n \xrightarrow{P} \theta_*,$$

$$\sqrt{n}\Sigma\left(\theta_n - \theta_*\right) \Rightarrow N\left(0, I_d\right),$$

where $\Sigma \overset{\triangle}{=} \left\{-\nabla^2 \ell\left(\vartheta\right)|_{\vartheta=\theta_*}\right\}^{-1/2}$. The precise statement is as follows.

Assumption II.1. *$g(\cdot)$ as a function of θ has a local maximum at θ_*, $g(\theta_*) > 0$, $\pi(\theta_*) > 0$.*

Assumption II.2. *π is continuous at θ_*, $g(\cdot)$ is of class $\mathcal{C}^2$ in a neighborhood of $\theta_*$and $H(\theta) = \nabla^2 g(\theta_*)$ is strictly negative definite.*

Assumption II.3. *For any $\delta > 0$, $\gamma(\delta) := \sup\left\{g(\theta) : \|\theta - \theta_*\| > \delta\right\} < g(\theta_*)$.*

Theorem II.4. *Lele et al. (2010) Assume Assumptions A.1 - II.3. Set $\psi_n \overset{\triangle}{=} \sqrt{n}\Sigma\left(\theta_n - \theta_*\right)$. As $n \to \infty$, θ_n converges in probability to θ_* and ψ_n converges weakly to $N\left(0, I_d\right)$.*

Definition II.5. *(Neighborhood). Let $\delta > 0$, define $N(\delta) := \left\{\theta : \|\Sigma^{-1}\left(\theta - \theta_*\right)\| < \delta\right\}$.*

Definition II.6. *Let $\theta_n \in R^d$ with density function $\pi_n(\cdot)$, define the standardized variable $\psi_n = \sqrt{n}\Sigma^{-1}\left(\theta_n - \theta_*\right)$ that has the density $g_n(\theta) = \frac{|\Sigma|}{n^{d/2}}\pi_n\left(\theta_* + \frac{1}{\sqrt{n}}\Sigma\theta\right)$.*

Lemma II.7. *Assume Assumptions A.1 and A.2, for all $\theta \in \Theta$,*

$$\lim_{n\to\infty}\left(g\left(\theta_* + \frac{1}{\sqrt{n}}\Sigma\theta\right)\right)^n = \exp\left(-\|\theta\|^2/2\right) \tag{2.16}$$

uniformly on a compact sets.

Proof. Fix δ_0 so small that $\nabla^2 g(\theta)$is continuous in the neighborhood $N(\delta_0)$. We assume without lost of generality that $g(\theta_*) = 1$, using Taylor expansion, $\exists \theta_+ \in (\theta, \theta_*)$ such that

$$g(\theta) = g(\theta_*) + \nabla g(\theta_*)(\theta - \theta_*) + \frac{1}{2}(\theta - \theta_*)^T \left(-\nabla^2 g(\theta_+)\right)(\theta - \theta_*)$$

$$= 1 - \frac{1}{2}(\theta - \theta_*)^T \left(-\nabla^2 g(\theta_+)\right)(\theta - \theta_*). \tag{2.17}$$

It is justified as $\nabla g(\theta_*) = 0$ and for any n large enough, $\theta_* + \frac{1}{\sqrt{n}}\Sigma\theta$ is in $N(\delta_0)$, so

$$g\left(\theta_* + \frac{1}{\sqrt{n}}\Sigma\theta\right) = 1 - \frac{\theta^T \Sigma^T \left\{-\nabla^2 g(\theta_n)\right\} \Sigma\theta}{2n}, \tag{2.18}$$

for some θ_n in the line segment $\left(\theta_*, \theta_* + \frac{1}{\sqrt{n}}\Sigma\theta\right)$. For $\varepsilon > 0$, choose $\delta(\varepsilon) < \delta_0$ small enough such that for $\theta \in N(\delta(\varepsilon))$, we have $\nabla^2 g(\theta)$ is negative definite and $\left\|\Sigma^T \nabla^2 g(\theta)\Sigma - I\right\| < \varepsilon$. Then for $0 \le x, y \le n$, we have

$$\left|\left(1 - \frac{x}{n}\right)^n - \left(1 - \frac{y}{n}\right)^n\right| \le |x - y|,$$

and

$$\left|\left(1 - \frac{y}{n}\right)^n - \exp(-y)\right| \le \frac{y^2}{n}. \tag{2.19}$$

Fix $M > 1$ and $0 < \varepsilon < 1$ and let $n > \max\left((M/\delta(\varepsilon))^2, M^2\right)$. Then for $\|\theta\| < M$ we have $\theta_n \in N(\delta(\varepsilon))$, so using (2.19) with $x = \frac{1}{2}\theta^T \Sigma^T \left\{-\nabla^2 g(\theta_n)\right\} \Sigma\theta$ and $y = \|\theta\|^2/2$ we get

$$\left|\left(g\left(\theta_* + \frac{1}{\sqrt{n}}\Sigma\theta\right)\right)^n - \exp\left(-\|\theta\|^2/2\right)\right| < \frac{\varepsilon M^2}{2} + \frac{M^4}{4n}, \tag{2.20}$$

as ε is arbitrary small, Lemma II.7 is proved. □

Remark II.8. As π is continuous at θ_*, from the above Lemma II.7,

$$\pi\left(\theta_* + \frac{1}{\sqrt{n}}\Sigma\theta\right)\left(g\left(\theta_* + \frac{1}{\sqrt{n}}\Sigma\theta\right)\right)^n$$

converges to $\pi(\theta_*)\exp\left(-\|\theta\|^2/2\right)$ uniformly on compact sets.

Remark II.9. Lemma II.7 and Fatou's Lemma give $\pi(\theta_*)|\Sigma|(2\pi)^{d/2} \leq \liminf_n c(n)\, n^{d/2}$, specifically there is a constant $C > 0$ such that $\frac{1}{c(n)} \leq Cn^{d/2}$

Lemma II.10. *Assume Assumptions A.1 and A.2, the following are equivalent.*

(a) $\psi_n \Rightarrow N(0, I_d)$

(b) g_n *converges point-wise to a multivariate standard normal density function, i.e* $c(n)\, n^{d/2} \to \pi(\theta_*)|\Sigma|(2\pi)^{d/2}$

(c) $\theta_n \Rightarrow \delta_{\theta_*}$ *where* δ_{θ_*} *indicate a Dirac-delta mass distribution*

Proof. (a) implies (b). Since g_n can be written as

$$g_n(\theta) = \frac{|\Sigma|}{c(n)\, n^{d/2}}\pi\left(\theta_* + \frac{1}{\sqrt{n}}\Sigma\theta\right)\left(g\left(\theta_* + \frac{1}{\sqrt{n}}\Sigma\theta\right)\right)^n. \tag{2.21}$$

Let B be the compact Borel set with positive measure. From (a) we have

$$\frac{1}{(2\pi)^{d/2}}\int_B \exp\left(-\|\theta\|^2/2\right)d\theta \leq \lim_n \frac{|\Sigma|}{c(n)\, n^{d/2}}\int_B \pi\left(\theta_* + \frac{1}{\sqrt{n}}\Sigma\theta\right)\left(g\left(\theta_* + \frac{1}{\sqrt{n}}\Sigma\theta\right)\right)^n d\theta.$$

Uniform convergence from Lemma II.7 gives

$$\lim_n \int_B \pi\left(\theta_* + \frac{1}{\sqrt{n}}\Sigma\theta\right)\left(g\left(\theta_* + \frac{1}{\sqrt{n}}\Sigma\theta\right)\right)^n d\theta = \pi(\theta_*)\int_B \exp\left(-\|\theta\|^2/2\right)d\theta. \tag{2.22}$$

Hence, $c(n)\, n^{d/2} \to \pi(\theta_*)\, |\Sigma|\, (2\pi)^{d/2}$ as $n \to \infty$. From Lemma II.7:

$$g_n(\theta) \to \frac{1}{(2\pi)^{d/2}} \int_B \exp\left(-\|\theta\|^2/2\right) d\theta, \tag{2.23}$$

(b) implies (a) by Scheffe's theorem.

(a) implies (c) is easy.

(c) implies (b). Since $\nabla^2 g$ and π are continuous at θ_* and Σ is strictly positive definite, from Lemma II.7 we see that for any $\varepsilon > 0$, we can find $\delta > 0$ so that $\theta \in N(\delta)$ and

$$g(\theta) < 1 - \frac{1}{2}(1-\varepsilon)(\theta-\theta_*)^T \Sigma^{-2}(\theta-\theta_*), \tag{2.24}$$

$$\pi(\theta) \leq \pi(\theta_*) + \varepsilon.$$

From (c), we assume that n is large enough so $1 - \varepsilon \leq \int_{N(\delta)} \pi_n(\theta)\, d\theta$. Multiplying this inequality by $c(n)\, n^{d/2}\, (1-\varepsilon)^{-1}$and using (2.24) gives

$$c(n)\, n^{d/2} \leq (1-\varepsilon)^{-1} n^{d/2} \int_{N(\delta)} \pi(\theta)\, g^n(\theta)\, d\theta$$

$$\leq (1-\varepsilon)^{-1} n^{d/2} (\pi(\theta_*) + \varepsilon) \times \int_{N(\delta)} \left[1 - \frac{1}{2}(1-\varepsilon)(\theta-\theta_*)^T \Sigma^{-2}(\theta-\theta_*)\right]^n d\theta$$

$$\leq (1-\varepsilon)^{-1} n^{d/2} (\pi(\theta_*) + \varepsilon) \times \int_{N(\delta)} \exp\left[1 - \frac{n}{2}(1-\varepsilon)(\theta-\theta_*)^T \Sigma^{-2}(\theta-\theta_*)\right] d\theta$$

$$= (1-\varepsilon)^{-1} n^{d/2} (\pi(\theta_*) + \varepsilon)\, |\Sigma|\, (2\pi)^{d/2}.$$

Letting $n \to \infty$ and $\varepsilon \to 0$,

$$\limsup_n c(n)\, n^{d/2} \leq \pi(\theta_*)\, |\Sigma|\, (2\pi)^{d/2}. \tag{2.25}$$

Also from Remarks II.8 and II.9, we have (b). □

Corollary II.11. *Corollary to Lemma II.10. Assume Assumption A.1 - II.3,* $\theta_n \Rightarrow \delta_{\theta_*}$

Proof. From Assumption II.3 and Corollary II.11 to Lemma II.10, for any $\delta > 0$

$$\frac{1}{c(n)} \int\limits_{\|\theta-\theta_*\|>\delta} \pi(x) f^n(x)\, dx \leq C n^{p/2} \gamma(\delta)^n \to 0.$$

This implies $\theta_n \Rightarrow \delta_{\theta_*}$ and

$$\psi_n \Rightarrow N(0, I_d).$$

The result of Theorem II.4 follows. □

2.5 Sequential Monte Carlo

Sampling is one of the key concepts of sequential Monte Carlo approaches. Sampling techniques is a powerful resort whenever estimation statistics cannot be computed analytically. Most distribution are generated by using either analytical or approximative transformations of samples from a standard distribution. However, for distribution which is too complex to sample directly or by transformation, we must resort to other techniques, such as *sampling importance resampling* (SIR). Let q be the *importance distribution* which is considerably easier to sample from and let $x^{(1)},\ x^{(2)},\ \ldots,\ x^{(N)}$ be a sample of size N from q. Sampling from q instead of p introduce *importance weights* $\tilde{\omega}^{(i)}$

$$\tilde{\omega}^{(i)} = p(x^{(i)})/q(x^{(i)}),\ i = 1, 2,\ \ldots,\ N. \tag{2.26}$$

Note that the more likely with respect to p the larger sample weight is. In that sense, re-sampling can be performed from these samples using sampling with replace-

ment, in which each sample $x^{(i)}$ has the probability of its normalized weight:

$$\omega^{(i)} = \frac{\tilde{\omega}^{(i)}}{\sum_{i=1}^{N} \tilde{\omega}^{(i)}},\ i = 1, 2,\ \ldots,\ N. \tag{2.27}$$

Thus a new sample $\tilde{x}^{(1)}, \tilde{x}^{(2)},\ \ldots,\ \tilde{x}^{(N)}$ from p can be drawn. By the Markov property of the distribution of interest, the sampling techniques can be carried out sequentially. Given the POMP structure of latent variables x_t with initial distribution $p_0(x_0)$, we can compute the weights recursively for each time step t. To sample the posterior distribution, importance distribution $q_{0:t}$ is chosen to satisfy:

$$q_{0:t}(x_{0:t}|y_{0:t}) = q_{0:t-1}(x_{0:t-1}|y_{0:t-1})q_t(x_t|X_{t-1}, y_t). \tag{2.28}$$

The proportionality of importance weights are:

$$\tilde{\omega}_t^{(i)} \propto \tilde{\omega}_{t-1}^{(i)} \frac{p(\tilde{x}_t^{(i)}|\tilde{x}_{t-1}^{(i)})p(y_t|\tilde{x}_t^{(i)})}{q_t(\tilde{x}_t^{(i)}|\tilde{x}_{t-1}^{(i)}, y_t)}. \tag{2.29}$$

A new sample of N particles from the previous sample is drawn according to the normalized weights and set the new weights to $1/N$. Thus the outlined algorithm for each time step is as follows:

1. Sample $\tilde{x}_t^{(i)}$ from $q_t(\tilde{x}_t^{(i)}|x_{t-1}^{(i),y_t})$.
2. Compute weights: $\tilde{\omega}_t^{(i)} = \omega_{t-1}^{(i)} \dfrac{p(\tilde{x}_t^{(i)}|\tilde{x}_{t-1}^{(i)})p(y_t|\tilde{x}_t^{(i)})}{q_t(\tilde{x}_t^{(x)}1\tilde{x}_{t-1}^{(x)}, y_t)}$.
3. Normalize weights: $\omega_t^{(i)} = \tilde{\omega}_t^{(i)} / \sum_{j=1}^{N} \tilde{\omega}_t^{(j)}$.
4. Resample $x_t^{(i)}$ according to the weights $\omega_t^{(i)}$.

2.6 Iterated Filtering

Iterated filtering for maximum likelihood inference was introduced in *Ionides et al.* (2006), and later generalized to sequential Monte Carlo filters in *Ionides et al.* (2011). The idea is to explore the parameter space by stochastic perturbation to smooth out the likelihood function. In the long run, the added noise will be canceled out through filtering while the parameter can be asymptotically achieved. The hidden $\{X_t\}_{t\in\mathbb{N}}$ process is augmented by a time varying parameter process $\{\breve{\Theta}_n\}$. Let K be a density with compact support, zero mean and covariance matrix Σ^θ, and let ζ_t be an independent draw from K. The time varying parameter process is then defined as

$$\breve{\Theta}_0 = \theta + \tau\zeta_0, \tag{2.30}$$

$$\breve{\Theta}_n = \breve{\Theta}_{n-1} + \sigma\zeta_n. \tag{2.31}$$

The stochastically perturbed model is defined conditionally on the time varying parameter process

$$\begin{aligned} &g_{\breve{X}_{0:N},\breve{Y}_{1:N},\breve{\Theta}_{0:N}}(x_{0:N}, y_{1:N}, \breve{\theta}_{0:N}, \theta,\ \tau) \\ &\qquad = f_{X_{0:N},Y_{1:N}}(x_{0:N}, y_{1:N}; \breve{\theta}_{0:N}) g_{\breve{\Theta}_{0N}}(\breve{\theta}_{0:N}; \theta,\ \tau, \Sigma). \end{aligned} \tag{2.32}$$

Define the conditional mean and covariance

$$\breve{\theta}_n^F = \mathrm{E}_{\theta,\sigma,\tau}[\breve{\Theta}_n | Y_{1:n} = y_{1:n}], \tag{2.33}$$

$$\breve{V}_n^P = \mathrm{C}ov_{\theta,\sigma,\tau}[\breve{\Theta}_n | Y_{1:n-1} = y_{1:n-1}]. \tag{2.34}$$

Ionides et al. (2011) shows that the score function can be approximated with these moments.

Theorem II.12. *(Theorem 3 in Ionides et al. (2011)). Let K_1 be a compact subset of $\mathbb{R}^p$, C_1 is a constant, τ is small enough and $\lim_{\tau\to 0}\sigma(\tau)/\tau = 0$. It then holds that*

$$\sup_{\theta\in K_1} |\sum_{n=1}^{N}(\breve{V}_n^P)^{-1}(\breve{\theta}_n^F - \breve{\theta}_{n-1}^F) - \nabla\ell_N(\theta)| \leq C_1(\tau + \frac{\sigma^2}{\tau^2}). \tag{2.35}$$

The framework is computationally attractive since moments are often easier to estimate than gradients. In addition, an approximation of the log-likelihood is feasible through augmented filtered states. If the sequences $\{\tau_m\}$, $\{\sigma_m\}$ and $\{J_m\}$ satisfied theorem assumptions, $\tilde{\theta}_n^F$ and $\tilde{V}_n^P$ are conditional sample mean and covariance computed from a sequential Monte Carlo filter using J_m particles. It then follows from *Ionides et al.* (2011) that

$$\tau_m J_m |\mathrm{E}_{MC}[(\tilde{V}_{n,m}^P)^{-1}(\tilde{\theta}_{n,m}^F - \tilde{\theta}_{n-1,m}^F) - (\breve{V}_{n,m}^P)^{-1}(\breve{\theta}_{n,m}^F - \breve{\theta}_{n-1,m}^F)]| \leq C_2, \tag{2.36}$$

and

$$\tau_m^2 J_m \mathrm{Var}_{MC}[(\tilde{V}_{n,m}^P)^{-1}(\tilde{\theta}_{n,m}^F - \tilde{\theta}_{n-1,m}^F)] \leq C_3. \tag{2.37}$$

Theorem II.13. *(Theorem 4 in Ionides et al. (2011)). Let K_2 be a compact subset of $\mathbb{R}^d$,$\{\tau_m\}$,$\{\sigma_m\}$ and $\{J_m\}$ be sequences such that $\tau_m \to 0$, $\sigma_m\tau_m^{-1} \to 0$ and $\tau_m J_m \to \infty$, and define*

$$\widetilde{\nabla}\ell_N(\theta) = \sum_{n=1}^{N}(\tilde{V}_{n,m}^P)^{-1}(\tilde{\theta}_{n,m}^F - \tilde{\theta}_{n-1,m}^F). \tag{2.38}$$

It then holds that

$$\lim_{m\to\infty}\sup_{\theta\in K_2} |\mathrm{E}_{MC}[\widetilde{\nabla}\ell_N(\theta)] - \nabla\ell_N(\theta)| = 0, \tag{2.39}$$

$$\lim_{m\to\infty}\sup_{\theta\in K_2} |\tau_m^2 J_m \mathrm{Var}_{MC}[\widetilde{\nabla}\ell_N(\theta)]| < \infty. \tag{2.40}$$

Theorem II.13 shows that the score function can be approximated arbitrarily well by decreasing the variances σ_m^2, τ_n^2 while simultaneously increasing the number of particles J_m. It can be used to define a difference equations for the parameter

estimates θ_m, by iteratively updating the estimates in the direction of the gradient (steepest ascent method).

Theorem II.14. *(Theorem 5 in Ionides et al. (2011)). Let* $\{\gamma_m\},\ \{\tau_m\},\ \{\sigma_m\}$ *and* $\{J_m\}$ *be positive sequences such that* $\tau_m \to 0,\ \sigma_m\tau_m^{-1} \to 0,\ \tau_m J_m \to \infty,\ \sum_{m=1}^{\infty} \gamma_m = \infty$ *and* $\sum_{m=1}^{\infty} \gamma_m^2 J_m^{-1}\tau_m^{-2} < \infty$ *and define* $\hat{\theta}_m$ *according to:*

$$\hat{\theta}_{m+1} = \hat{\theta}_m + \gamma_m \sum_{n=1}^{N} (\tilde{V}_{n,m}^P)^{-1}(\tilde{\theta}_{n,m}^F - \tilde{\theta}_{n-1,m}^F). \tag{2.41}$$

The estimate will then converge to the MLE with probability one, $\theta_n \overset{a.s.}{\to} \theta^\star$.

Choosing $\gamma_m = m^{-1},\ \tau_m^2 = m^{-1},\ \sigma_m^2 = m^{-(1+\delta)}$ and $J_m = m^{(1/2+\delta)}$ where $\delta > 0$ is an example for satisfying the conditions in Theorem II.14 for convergence.

CHAPTER III

Bayes Map Iterated Filtering

3.1 Introduction

Variations on the original iterated filtering algorithm have been proposed to extend it to general latent-variable models (*Ionides et al.*, 2011) and to improve numerical performance (*Doucet et al.*, 2013; *Lindström et al.*, 2012). In this chapter, we study a new iterated filtering algorithm which generalizes the data cloning method (*Lele et al.*, 2007, 2010) and is therefore also related to other Monte Carlo methods for likelihood-based inference (*Doucet et al.*, 2002; *Gaetan and Yao*, 2003; *Jacquier et al.*, 2007). Data cloning methodology is based on the observation that iterating a Bayes map converges to a point mass at the maximum likelihood estimate. Combining such iterations with perturbations of model parameters improves the numerical stability of data cloning and provides a foundation for stable algorithms in which the Bayes map is numerically approximated by sequential Monte Carlo computations.

We investigate convergence of a sequential Monte Carlo implementation of an iterated filtering algorithm which combines data cloning, in the sense of Lele et al (*Lele et al.*, 2007), with the stochastic parameter perturbations used by the iterated filtering algorithm of *Ionides et al.* (2006). Lindström et al (*Lindström et al.*, 2012) proposed a similar algorithm, termed fast iterated filtering, but the theoretical support for that algorithm involved unproved conjectures. We present convergence results for

our algorithm, which we call IF2. Empirically, it can dramatically out-perform the previous iterated filtering algorithm of *Ionides et al.* (2006), which we refer to as IF1. Though IF1 and IF2 both involve recursively filtering through the data, the theoretical justification and practical implementations of these algorithms are fundamentally different.

IF1 approximates the Fisher score function, whereas IF2 implements an iterated Bayes map. IF1 has been used in applications for which no other computationally feasible algorithm for statistically efficient, likelihood-based inference was known (*King et al.*, 2008; *Laneri et al.*, 2010; *He et al.*, 2010; *Blackwood et al.*, 2013a; *Shrestha et al.*, 2013; *Blake et al.*, 2014). The extra capabilities offered by IF2 open up further possibilities for drawing inferences about nonlinear partially observed stochastic dynamic models from time series data. Iterated filtering algorithms implemented using basic sequential Monte Carlo techniques have the property that they do not need to evaluate the transition density of the latent Markov process. Algorithms with this property have been called plug-and-play (*Bretó et al.*, 2009; *He et al.*, 2010). Various other plug-and-play methods for POMP models have been recently proposed (*Andrieu et al.*, 2010; *Toni et al.*, 2009; *Wood*, 2010; *Shaman and Karspeck*, 2012), due largely to the convenience of this property in scientific applications.

3.2 An algorithm and related questions

For a general POMP model defined as above, we look for a maximum likelihood estimate (MLE), i.e., a value $\hat{\theta}$ maximizing $\ell(\theta)$. The IF2 algorithm defined below provides

Algorithm IF2. Iterated filtering

input:

Simulator for $f_{X_0}(x_0\,;\theta)$

Simulator for $f_{X_n|X_{n-1}}(x_n\,|\,x_{n-1}\,;\theta)$, n in $1:N$

Evaluator for $f_{Y_n|X_n}(y_n\,|\,x_n\,;\theta)$, n in $1:N$

Data, $y^*_{1:N}$

Number of iterations, M

Number of particles, J

Initial parameter swarm, $\{\Theta^0_j,\ j \text{ in } 1:J\}$

Perturbation density, $h_n(\theta\,|\,\varphi\,;\sigma)$, n in $1:N$

Perturbation sequence, $\sigma_{1:M}$

output: Final parameter swarm, $\{\Theta^M_j,\ j \text{ in } 1:J\}$

For m in $1:M$

$\Theta^{F,m}_{0,j} \sim h_0(\theta\,|\,\Theta^{m-1}_j\,;\sigma_m)$ for j in $1:J$ ule[-2mm]0mm5mm

$X^{F,m}_{0,j} \sim f_{X_0}(x_0;\Theta^{F,m}_{0,j})$ for j in $1:J$ ule[-2mm]0mm5mm

For n in $1:N$

$\Theta^{P,m}_{n,j} \sim h_n(\theta\,|\,\Theta^{F,m}_{n-1,j},\sigma_m)$ for j in $1:J$ ule[-2mm]0mm5mm

$X^{P,m}_{n,j} \sim f_{X_n|X_{n-1}}(x_n\,|\,X^{F,m}_{n-1,j};\Theta^{P,m}_j)$ for j in $1:J$ ule[-2mm]0mm5mm

$w^m_{n,j} = f_{Y_n|X_n}(y^*_n\,|\,X^{P,m}_{n,j};\Theta^{P,m}_{n,j})$ for j in $1:J$ ule[-2mm]0mm5mm

Draw $k_{1:J}$ with $\mathbb{P}(k_j = i) = w^m_{n,i}\Big/\sum_{u=1}^{J} w^m_{n,u}$

$\Theta^{F,m}_{n,j} = \Theta^{P,m}_{n,k_j}$ and $X^{F,m}_{n,j} = X^{P,m}_{n,k_j}$ for j in $1:J$ ule[-2mm]0mm5mm

End For

Set $\Theta^m_j = \Theta^{F,m}_{N,j}$ for j in $1:J$

End For

a plug-and-play Monte Carlo approach to obtaining $\hat{\theta}$.

A simplification of IF2 arises when $N = 1$, in which case iterated filtering is called iterated importance sampling (*Ionides et al.*, 2011) (see A.2). Algorithms

similar to IF2 with a single iteration ($M = 1$) have been proposed in the context of Bayesian inference (*Kitagawa*, 1998; *Liu and West*, 2001) (see A.6). When $M = 1$ and $h_n(\theta \,|\, \varphi\,; \sigma)$ degenerates to a point mass at φ, the IF2 algorithm becomes a standard particle filter (*Arulampalam et al.*, 2002; *Doucet et al.*, 2001). In the IF2 algorithm description, $\Theta_{n,j}^{F,m}$ and $X_{n,j}^{F,m}$ are the jth particles at time n in the Monte Carlo representation of the mth iteration of a filtering recursion. The filtering recursion is coupled with a prediction recursion, represented by $\Theta_{n,j}^{P,m}$ and $X_{n,j}^{P,m}$. The resampling indices $k_{1:J}$ in IF2 are taken to be a multinomial draw for our theoretical analysis, but systematic resampling is preferable in practice (*Arulampalam et al.*, 2002). A natural choice of $h_n(\theta \,|\, \varphi\,; \sigma)$ is a multivariate normal density with mean φ and variance $\sigma^2\Sigma$ for some covariance matrix Σ, but in general h_n could be any conditional density parameterized by σ. Combining the perturbations over all the time points, we define

$$h(\theta_{0:N} \,|\, \varphi\,; \sigma) = h_0(\theta_0 \,|\, \varphi\,; \sigma) \prod_{n=1}^{N} h_n(\theta_n \,|\, \theta_{n-1}\,; \sigma).$$

We define an extended likelihood function on Θ^{N+1} by

$$\begin{aligned} \breve{\ell}(\theta_{0:N}) \;\; &= \; \int \!\! \dots \!\! \int dx_0 \dots dx_N \Big\{ f_{X_0}(x_o\,; \theta_0) \times \\ &\prod_{n=1}^{N} f_{X_n|X_{n-1}}(x_n \,|\, x_{n-1}\,; \theta_n) f_{Y_n|X_n}(y_n^* \,|\, x_n\,; \theta_n) \Big\}. \end{aligned}$$

Each iteration of IF2 is a Monte Carlo approximation to a map

$$T_\sigma f(\theta_N) = \frac{\int \breve{\ell}(\theta_{0:N}) h(\theta_{0:N} | \varphi\,; \sigma) f(\varphi)\, d\varphi\, d\theta_{0:N-1}}{\int \breve{\ell}(\theta_{0:N}) h(\theta_{0:N} | \varphi\,; \sigma) f(\varphi)\, d\varphi\, d\theta_{0:N}}, \tag{3.1}$$

with f and $T_\sigma f$ approximating the initial and final density of the parameter swarm. For our theoretical analysis, we consider the case when the standard deviation of the parameter perturbations is held fixed at $\sigma_m = \sigma > 0$ for $m = 1, \dots, M$.

In this case, IF2 is a Monte Carlo approximation to $T_\sigma^M f(\theta)$. We call the fixed σ version of IF2 *homogeneous* iterated filtering since each iteration implements the same map. For any fixed σ, one cannot expect a procedure such as IF2 to converge to a point mass at the MLE. However, for fixed but small σ, we show that IF2 does approximately maximize the likelihood, with an error that shrinks to zero in a limit as $\sigma \to 0$ and $M \to \infty$. An immediate motivation for studying the homogeneous case is simplicity; it turns out that even with this simplifying assumption the theoretical analysis is not entirely straightforward. Moreover, the homogeneous analysis gives at least as much insight as an asymptotic analysis into the practical properties of IF2, when σ_m decreases down to some positive level $\sigma > 0$ but never completes the asymptotic limit $\sigma_m \to 0$. Iterated filtering algorithms have been primarily developed in the context of making progress on complex models for which successfully achieving and validating global likelihood optimization is challenging. In such situations, it is advisable to run multiple searches and continue each search up to the limits of available computation (*Ingber*, 1993).

If no single search can reliably locate the global maximum, a theory assuring convergence to a neighborhood of the maximum is as relevant as a theory assuring convergence to the maximum itself in a practically unattainable limit.

The map T_σ can be expressed as a composition of a parameter perturbation with a Bayes map that multiplies by the likelihood and renormalizes. Iteration of the Bayes map alone has a central limit theorem (CLT) (*Lele et al.*, 2007) which forms the theoretical basis for the data cloning methodology of *Lele et al.* (2007, 2010). Repetitions of the parameter perturbation may also be expected to follow a CLT.

One might therefore imagine that the composition of these two operations also has a Gaussian limit.

This is not generally true, since the rescaling involved in the perturbation CLT prevents the Bayes map CLT from applying (see A.4). Our agenda is to seek condi-

tions guaranteeing the following:

(A1) For every fixed $\sigma > 0$, $\lim_{m\to\infty} T_\sigma^m f = f_\sigma$ exists.

(A2) When J and M become large, IF2 numerically approximates f_σ.

(A3) As the noise intensity becomes small, $\lim_{\sigma\to 0} f_\sigma$ approaches a point mass at the MLE, if it exists.

Stability of filtering problems and uniform convergence of sequential Monte Carlo numerical approximations are closely related, and so A1 and A2 are studied together in Theorem III.1. Each iteration of IF2 involves standard sequential Monte Carlo filtering techniques applied to an extended model where latent variable space is augmented to include a time-varying parameter.

Indeed, all M iterations together can be represented as a filtering problem for this extended POMP model on M replications of the data. The proof of Theorem III.1 therefore leans on existing results. The novel issue of A3 is then addressed in Theorem III.2.

3.3 Convergence of IF2

First, we follow the notations and proofs of *Ionides et al.* (2015). Let $\{\breve{\Theta}_{0:N}^m, m = 1, 2, \dots\}$ be a Markov chain taking values in Θ^{N+1} such that $\breve{\Theta}_{0:N}^1$ has density $\int h(\theta_{0:N} \,|\, \varphi\,; \sigma) f(\varphi)\, d\varphi$, and $\breve{\Theta}_{0:N}^m$ has conditional density $h(\theta_{0:N} \,|\, \varphi_N\,; \sigma)$ given $\breve{\Theta}_{0:N}^{m-1} = \varphi_{0:N}$ for $m \geq 2$. Suppose that $\{\breve{\Theta}_{0:N}^m, m \geq 1\}$ is constructed on the canonical probability space $\Omega = \{(\theta_{0:N}^1, \theta_{0:N}^2, \dots)\}$ with $\theta_{0:N}^m = \breve{\Theta}_{0:N}^m(\vartheta)$ for $\vartheta = (\theta_{0:N}^1, \theta_{0:N}^2, \dots) \in \Omega$.

Let $\{\mathcal{F}_m\}$ be the corresponding Borel filtration. To consider a time-rescaled limit of $\{\breve{\Theta}_{0:N}^m, m = 1, 2, \dots\}$ as $\sigma \to 0$, let $\{W_\sigma(t), t \geq 0\}$ be a continuous-time, right-continuous, piecewise constant process defined at its points of discontinuity by $W_\sigma(k\sigma^2) = \breve{\Theta}_N^{k+1}$ when k is a nonnegative integer. Let $\{\breve{Z}_{0:N}^m, m = 1, 2, \dots\}$ be the

filtered process defined such that, for any event $E \in \mathcal{F}_M$,

$$\mathbb{P}_{\breve{Z}}(E) = \frac{\mathbb{E}_{\breve{\Theta}}[\breve{\ell}_{1:M} I_E]}{\mathbb{E}_{\breve{\Theta}}[\breve{\ell}_{1:M}]}, \tag{3.2}$$

where I_E is the indicator function for event E and

$$\breve{\ell}_{1:M}(\vartheta) = \prod_{m=1}^{M} \breve{\ell}(\theta^m_{0:N}).$$

In (3.2), $\mathbb{P}_{\breve{Z}}(E)$ denotes probability under the law of $\{\breve{Z}^m_n\}$, and $\mathbb{E}_{\breve{\Theta}}$ denotes expectation under the law of $\{\breve{\Theta}^m_n\}$. The process $\{\breve{Z}^m_n\}$ is constructed so that $\breve{Z}^m_N$ has density $T^m f$. We make the following assumptions.

(B1) $\{W_\sigma(t), 0 \le t \le 1\}$ converges weakly as $\sigma \to 0$ to a diffusion $\{W(t), 0 \le t \le 1\}$, in the space of right-continuous functions with left limits equipped with the uniform convergence topology. For any open set $A \subset \Theta$ with positive Lebesgue measure and $\epsilon > 0$, there is a $\delta(A, \epsilon) > 0$ such that $\mathbb{P}\big[W(t) \in A \text{ for all } \epsilon \le t \le 1 \,|\, W(0)\big] > \delta$.

(B2) For some $t_0(\sigma)$ and $\sigma_0 > 0$, $W_\sigma(t)$ has a positive density on Θ, uniformly over the distribution of $W(0)$ for all $t > t_0$ and $\sigma < \sigma_0$.

(B3) $\ell(\theta)$ is continuous in a neighborhood $\{\theta : \ell(\theta) > \lambda_1\}$ for some $\lambda_1 < \sup_\varphi \ell(\varphi)$.

(B4) There is an $\epsilon > 0$ with $\epsilon^{-1} > f_{Y_n|X_n}(y^*_n \,|\, x_n, \theta) > \epsilon$ for all $1 \le n \le N$, $x_n \in \mathbb{X}$ and $\theta \in \Theta$.

(B5) There is a C_1 such that $h_n(\theta \,|\, \varphi\,; \sigma) = 0$ when $|\theta - \varphi| > C_1\sigma$, for all σ.

(B6) There is a C_2 such that $\sup_{1\le n\le N} |\theta_n - \theta_{n-1}| < C_1\,\sigma$ implies $|\breve{\ell}(\theta_{0:N}) - \ell(\theta_N)| < C_2\,\sigma$, for all σ and all n.

Conditions B1 and B2 hold when $h_n(\theta \,|\, \varphi\,; \sigma)$ corresponds to a reflected Gaussian random walk and $\{W(t)\}$ is a reflected Brownian motion (see A.8). More generally, $h_n(\theta \,|\, \varphi\,; \sigma)$ is a location-scale family with mean φ away from a boundary, then $\{W(t)\}$

will behave like Brownian motion in the interior of Θ. B4 follows if $\mathbb{X}$ is compact and $f_{Y_n|X_n}(y_n^* \,|\, x_n\,;\theta)$ is positive and continuous as a function of θ and x_n. B5 can be guaranteed by construction. B3 and B6 are undemanding regularity conditions on the likelihood and extended likelihood.

A formalization of A1 and A2 can now be stated as follows.

Theorem III.1. *Let T_σ be the map of* (3.1) *and suppose B2 and B4. There is a unique probability density f_σ such that for any probability density f on Θ,*

$$\lim_{m\to\infty} \|T_\sigma^m f - f_\sigma\|_1 = 0, \tag{3.3}$$

where $\|f\|_1$ is the L^1 norm of f.

Let $\{\Theta_j^M,\ j = 1, \ldots, J\}$ be the output of IF2, with $\sigma_m = \sigma > 0$. There is a finite constant $C > 0$ such that, for any function $\phi : \Theta \to \mathbb{R}$ and all M,

$$\mathbb{E}\left\{\left|\frac{1}{J}\sum_{j=1}^{J}\phi(\Theta_j^M) - \int \phi(\theta) f_\sigma(\theta)\, d\theta\right|\right\} \le \frac{C\ \sup_\theta |\phi(\theta)|}{\sqrt{J}}. \tag{3.4}$$

Proof. B2 and B4 imply that T_σ^k is mixing, in the sense of *Le Gland and Oudjane* (2004), for all sufficiently large k.

The results of *Le Gland and Oudjane* (2004) are based on the contractive properties of mixing maps in the Hilbert projective metric. Although *Le Gland and Oudjane* (2004) stated their results in the case where T itself is mixing, the required geometric contraction in the Hilbert metric holds as long as T^k is mixing for all $K \le k \le 2K-1$ for some $K \ge 1$ (*Eveson*, 1995, Theorem 2.5.1). Corollary 4.2 of *Le Gland and Oudjane* (2004) implies (3.3), noting the equivalence of the Hilbert projective metric and the total variation norm shown in their Lemma 3.4. Then, Corollary 5.12 of *Le Gland and Oudjane* (2004) implies (3.4), completing the proof of Theorem III.1. A longer version of this proof is given in the supplement (see A.9). □

Results similar to Theorem III.1 can be obtained using Dobrushin contraction techniques *Del Moral and Doucet* (2004). Results appropriate for non-compact spaces can be obtained using drift conditions on a potential function *Whiteley et al.* (2012). Now we move on to our formalization of A3:

Theorem III.2. *Assume B1–B6. For* $\lambda_2 < \sup_\varphi \ell(\varphi)$,
$\lim_{\sigma\to 0} \int f_\sigma(\theta) 1_{\{\ell(\theta)<\lambda_2\}}\, d\theta = 0$.

Proof. Let $\lambda_0 = \sup_\varphi \ell(\varphi)$ and $\lambda_3 = \inf_\varphi \ell(\varphi)$. From B4, $\infty > \lambda_0 > \lambda_3 > 0$. For positive constants ϵ_1, ϵ_2, η_1, η_2 and $\lambda_1 < \lambda_0$, define

$$\begin{aligned} e_1 &= (1-\epsilon_1)\log(\lambda_0+\epsilon_2) + \epsilon_1 \log(\lambda_2+\epsilon_2), \\ e_2 &= (1-\eta_1)\log(\lambda_1-\eta_2) + \eta_1 \log(\lambda_3-\eta_2). \end{aligned}$$

We can pick ϵ_1, ϵ_2, η_1, η_2 and λ_1 so that $e_1 < e_2$. Suppose that $\{\breve{\Theta}_n^m\}$ is initialized with the stationary distribution $f = f_\sigma$ identified in Theorem III.1. Now, set M to be the greatest integer less than $1/\sigma^2$, and let F_1 be the event that $\{\Theta_N^m, m = 1, \ldots, M\}$ spends at least a fraction of time ϵ_1 in $\{\theta : \ell(\theta) < \lambda_2\}$.

Formally,

$$F_1 = \Big\{\vartheta \in \Omega : \frac{1}{M}\sum_{m=1}^{M} 1_{\{\ell(\theta_N^m)<\lambda_2\}} > \epsilon_1\Big\}.$$

We wish to show that $\mathbb{P}_{\breve{Z}}[F_1]$ is small for σ small. Let F_2 be the set of sample paths that spend at least a fraction of time $(1-\eta_1)$ up to time M in $\{\theta : \ell(\theta) > \lambda_1\}$, i.e.,

$$F_2 = \Big\{\vartheta \in \Omega : \frac{1}{M}\sum_{m=1}^{M} 1_{\{\ell(\theta_N^m)>\lambda_1\}} > (1-\eta_1)\Big\}.$$

Then, we calculate

$$\begin{aligned}
\mathbb{P}_{\breve{Z}}[F_1] &= \frac{\mathbb{E}_{\breve{\Theta}}\big[\breve{\ell}_{1:M} 1_{F_1}\big]}{\mathbb{E}_{\breve{\Theta}}\big[\breve{\ell}_{1:M}\big]} \\
&\leq \frac{\mathbb{E}_{\breve{\Theta}}\big[\breve{\ell}_{1:M} 1_{F_1}\big]}{\mathbb{E}_{\breve{\Theta}}\big[\breve{\ell}_{1:M} 1_{F_2}\big]} \\
&\leq \frac{\mathbb{E}_{\breve{\Theta}}\left[\prod_{m=1}^{M}\big\{\ell(\theta_N^m) + C_2\sigma\big\}\, 1_{F_1}\right]}{\mathbb{E}_{\breve{\Theta}}\left[\prod_{m=1}^{M}\big\{\ell(\theta_N^m) - C_2\sigma\big\}\, 1_{F_2}\right]} \qquad (3.5) \\
&\leq \frac{\mathbb{E}_{\breve{\Theta}}\big[\exp\{Me_1\} 1_{F_1}\big]}{\mathbb{E}_{\breve{\Theta}}\big[\exp\{Me_2\} 1_{F_2}\big]} \qquad (3.6) \\
&= \exp\big\{(e_1 - e_2)M\big\}\frac{\mathbb{P}_{\breve{\Theta}}[F_1]}{\mathbb{P}_{\breve{\Theta}}[F_2]}. \qquad (3.7)
\end{aligned}$$

We used B5 and B6 to arrive at (3.5), then to get to (3.6) we have taken σ small enough that $C_2\sigma < \epsilon_2$ and $C_2\sigma < \eta_2$. From B3, $\{\theta : \ell(\theta) > \lambda_1\}$ is an open set, and B1 therefore ensures each of the probabilities $\mathbb{P}_{\Theta_{1:M}}[F_1]$ and $\mathbb{P}_{\Theta_{1:M}}[F_2]$ in (3.7) tends to a positive limit as $\sigma \to 0$ given by the probability under the limiting distribution $\{W(t)\}$ (see A.1). The term $\exp\{(e_1 - e_2)M\}$ tends to zero as $\sigma \to 0$ since, by construction, $M \to \infty$ and $e_1 < e_2$. Setting $L = \{\theta : \ell(\theta) \leq \lambda_2\}$, and noting that $\{\breve{Z}_N^m, m = 1, 2, \dots\}$ is constructed to have stationary marginal density f_σ, we have

$$\begin{aligned}
\int_L f_\sigma(\theta)\, d\theta &= \frac{1}{M}\sum_{m=1}^{M}\Big\{\mathbb{P}_{\breve{Z}}\big[\breve{Z}_N^m \in L \,|\, F_1\big]\, \mathbb{P}_{\breve{Z}}[F_1] + \\
&\qquad\qquad \mathbb{P}_{\breve{Z}}\big[\breve{Z}_N^m \in L \,|\, F_1^c\big]\, \mathbb{P}_{\breve{Z}}[F_1^c]\Big\}, \\
&\leq \epsilon_1 + \mathbb{P}_{\breve{Z}}[F_1],
\end{aligned}$$

which can be made arbitrarily small by picking ϵ_1 small and σ small, completing the proof.

□

3.4 Demonstration of IF2 with nonconvex superlevel sets

Theorems 1 and 2 do not involve any Taylor series expansions, which are basic in the justification of IF1 (*Ionides et al.*, 2011). This might suggest that IF2 can be effective on likelihood functions without good low-order polynomial approximations. In practice, this can be seen by comparing IF2 with IF1 on a simple two-dimensional toy example ($\dim(\Theta) = \dim(\mathbb{X}) = \dim(\mathbb{Y}) = 2$) in which the superlevel sets $\{\theta : \ell(\theta) > \lambda\}$ are connected but not convex. We also compare with particle Markov chain Monte Carlo (PMCMC) implemented as the PMMH algorithm of *Andrieu et al.* (2010).

The justification of PMCMC also does not depend on Taylor series expansions, but PMCMC is computationally expensive compared to iterated filtering (*Bhadra*, 2010). Our toy example has a constant and non-random latent process, $X_n = \big(\exp\{\theta_1\}, \theta_2 \exp\{\theta_1\}\big)$ for $n = 1, \ldots, N$.

The known measurement model is

$$f_{Y_n|X_n}(y \,|\, x\,; \theta) \sim \text{Normal}\left[x, \begin{pmatrix} 100 & 0 \\ 0 & 1 \end{pmatrix}\right],$$

This example was designed so that a nonlinear combination of the parameters is well identified whereas each parameter is marginally weakly identified. For the truth, we took $\theta = (1, 1)$. We supposed that θ_1 is suspected to fall in the interval $[-2, 2]$ and θ_2 is expected in $[0, 10]$. We used a uniform distribution on this rectangle to specify the prior for PMCMC and to generate random starting points for all the algorithms. We set $N = 100$ observations, and we used a Monte Carlo sample size of $J = 100$ particles. For IF1 and IF2, we employed $M = 100$ filtering iterations, with initial random walk standard deviation 0.1 decreasing geometrically down to 0.01.

For PMCMC, we used 10^4 filtering iterations with random walk standard devi-

ation 0.1, awarding PMCMC 100 times the computational resources offered to IF1 and IF2. Independent, normally distributed parameter perturbations were used for IF1, IF2 and PMCMC. The random walk standard deviation for PMCMC is not immediately comparable to that for IF1 and IF2, since the latter add the noise at each observation time whereas the former adds it only between filtering iterations. All three methods could have their parameters fine-tuned, or be modified in other ways to take advantage of the structure of this particular problem. However, this example demonstrates a feature that makes tuning algorithms tricky: the nonlinear ridge along contours of constant $\theta_2 \exp(\theta_1)$ becomes increasingly steep as θ_1 increases, so no single global estimate of the second derivative of the likelihood is appropriate. Reparameterization can linearize the ridge in this toy example, but in practical problems with much larger parameter spaces one does not always know how to find appropriate reparameterizations, and a single reparameterization may not be appropriate throughout the parameter space.

Fig. 3.1 compares the the performance of the three methods, based on 30 Monte Carlo replications. These replications investigate the likelihood and posterior distribution for a single draw from our toy model, since our interest is in the Monte Carlo behavior for a given dataset. For this simulated dataset, the MLE is $\theta = (1.20, 0.81)$, shown as a green triangle in Fig. 3.1, panels A, B and C. In this toy example, the posterior distribution can also be computed directly by numerical integration. In Fig. 3.1A, we see that IF1 performs poorly on this challenge. None of the 30 replications approach the MLE. The linear combination of perturbed parameters involved in the IF1 update formula can all too easily knock the search off a nonlinear ridge. Fig. 3.1B shows that IF2 performs well on this test, with almost all the Monte Carlo replications clustering in the region of highest likelihood. Fig. 3.1C shows the end points of the PMCMC replications, which are nicely spread around the region of high posterior probability. However, Fig. 3.1D shows that mixing of the PMCMC Markov

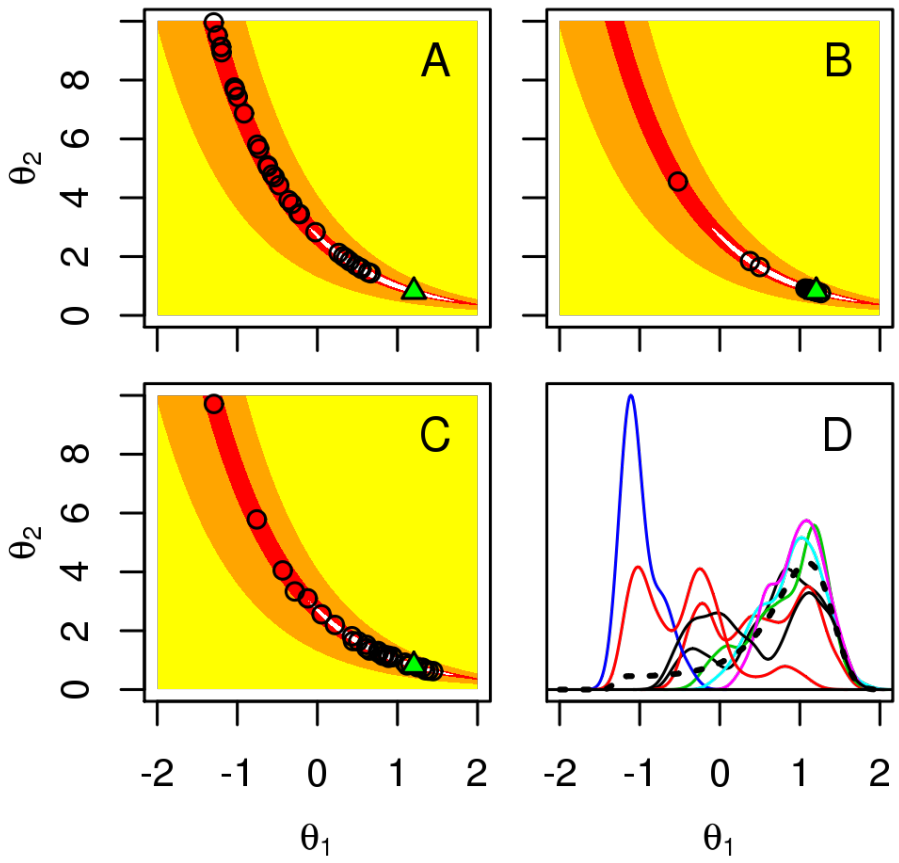

Figure 3.1: Results for the simulation study of the toy example. A. IF1 point estimates from 30 replications (circles) and the MLE (green triangle). The region of parameter space with likelihood within 3 log units of the maximum (white), with 10 log units (red), within 100 log units (orange) and lower (yellow). B. IF2 point estimates from 30 replications (circles) with the same algorithmic settings as IF1. C. Final parameter value of 30 PMCMC chains (circles). D. kernel density estimates of the posterior for θ_1 for the first 8 of these 30 PMCMC chains (solid lines), with the true posterior distribution (dotted black line).

chains was problematic.

3.5 Application to a cholera model

Highly nonlinear, partially observed, stochastic dynamic systems are ubiquitous in the study of biological processes. The physical scale of the systems vary widely from molecular biology (*Wilkinson*, 2012) to population ecology and epidemiology (*Keeling and Rohani*, 2009), but POMP models arise naturally at all scales. In the face of biological complexity, it is necessary to determine which scientific aspects of a system are critical for the investigation. Giving consideration to a range of potential mechanisms, and their interactions, may require working with highly parameterized models. Limitations in the available data may result in some combinations of parameters being weakly identifiable. Despite this, other combinations of parameters may be adequately identifiable and give rise to some interesting statistical inferences. To demonstrate the capabilities of IF2 for such analyses, we fit a model for cholera epidemics in historic Bengal developed by *King et al.* (2008). The model, the data, and the implementations of IF1 and IF2 used below are all contained in the open source R package `pomp` (*King et al.*, 2015b). The code generating the results in this article is provided as supplementary data.

Cholera is a diarrheal disease caused by the bacterial pathogen *Vibrio cholerae*. Without appropriate medical treatment, severe infections can rapidly result in death by dehydration. Many questions regarding cholera transmission remain unresolved: what is the epidemiological role of free-living environmental vibrio? how important are mild and asymptomatic infections for the transmission dynamics? how long does protective immunity last following infection? The model we consider splits up the study population of $P(t)$ individuals into those who are susceptible, $S(t)$, infected, $I(t)$, and recovered, $R(t)$.

$P(t)$ is assumed known from census data. To allow flexibility in representing

immunity, $R(t)$ is subdivided into $R_1(t), \ldots, R_k(t)$, where we take $k = 3$. Cumulative cholera mortality in each month is tracked with a variable $M(t)$ that resets to zero at the beginning of each observation period. The state process, $\{X(t) = (S(t), I(t), R_1(t), \ldots, R_k(t), M(t)), t \geq t_0\}$ follows a stochastic differential equation,

$$\begin{aligned}
dS &= \big\{k\epsilon R_k + \delta(S - H) - \lambda(t)S\big\}dtdP - (\sigma SI/P)\, dB, \\
dI &= \big\{\lambda(t)S - (m + \delta + \gamma)I\big\}dt + (\sigma SI/P)\, dB, \\
dR_1 &= \big\{\gamma I - (k\epsilon + \delta)R_1\big\}dt, \\
&\vdots \\
dR_k &= \big\{k\epsilon R_{k-1} - (k\epsilon + \delta)R_k\big\}dt,
\end{aligned}$$

driven by a Brownian motion $\{B(t)\}$.

Nonlinearity arises through the force of infection, $\lambda(t)$, specified as

$$\begin{aligned}
\lambda(t) = \bar{\beta} \exp\Big\{\beta_{\text{trend}}(t - t_0) + \textstyle\sum_{j=1}^{N_s} \beta_j s_j(t)\Big\}(I/P) + \\
\bar{\omega} \exp\Big\{\textstyle\sum_{j=1}^{N_s} \omega_j s_j(t)\Big\},
\end{aligned}$$

where $\{s_j(t), j = 1, \ldots, N_s\}$ is a periodic cubic B-spline basis; $\{\beta_j, j = 1, \ldots, N_s\}$ model seasonality of transmission; $\{\omega_j, j = 1, \ldots, N_s\}$ model seasonality of the environmental reservoir; $\bar{\omega}$ and $\bar{\beta}$ are scaling constants set to $\bar{\omega} = \bar{\beta} = 1\text{yr}^{-1}$, and we set $N_s = 6$. The data, consisting of monthly counts of cholera mortality, are modeled via $Y_n \sim \text{Normal}(M_n, \tau^2 M_n^2)$ for $M_n = \int_{t_{n-1}}^{t_n} m\, I(s)\, ds$. The inference goal used to assess IF1 and IF2 is to find high-likelihood parameter values starting from randomly drawn starting values in a large hyper-rectangle (see A.10).

A single search cannot necessarily be expected to reliably obtain the maximum of the likelihood, due to multi-modality, weak identifiability, and considerable Monte Carlo error in evaluating the likelihood. Multiple starts and restarts may be needed both for effective optimization and for assessing the evidence to validate effective optimiza-

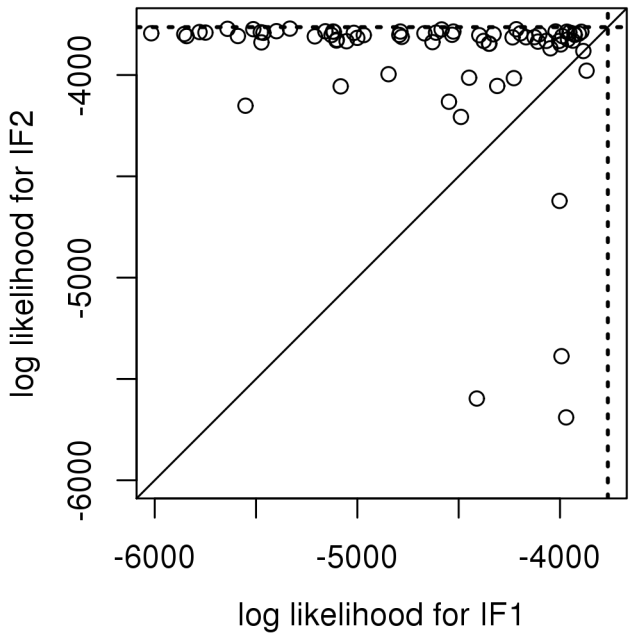

Figure 3.2: Comparison of IF1 and IF2 on the cholera model. Points are the log likelihood of the parameter vector output by IF1 and IF2, both started at a uniform draw from a large hyper-rectangle (see A.10). Likelihoods were evaluated as the median of 10 particle filter replications (i.e., IF applied with $M = 1$ and $\sigma_1 = 0$) each with $J = 2 \times 10^4$ particles. 17 poorly performing searches are off the scale of this plot (15 due to the IF1 estimate, 2 due to the IF2 estimate). Dotted lines show the maximum log likelihood reported by *King et al.* (2008).

tion. However, optimization progress made on an initial search provides a concrete criterion to compare methodologies. Since IF1 and IF2 have essentially the same computational cost, for a given Monte Carlo sample size and number of iterations, shared fixed values of these algorithmic parameters provide an appropriate comparison.

Fig. 3.2 compares results for 100 searches with $J = 10^4$ particles and $M = 100$ iterations of the search. An initial Gaussian random walk standard deviation of 0.1 geometrically decreasing down to a final value of 0.01 was used for all parameters except S_0, I_0, $R_{1,0}$, $R_{2,0}$ and $R_{3,0}$. For those initial value parameters, the random walk standard deviation decreased geometrically from 0.2 down to 0.02, but these perturbations were applied only at time t_0. Since some starting points may lead both IF1 and IF2 to fail to approach the global maximum, Fig. 3.2 plots the likelihoods of

parameter vectors output by IF1 and IF2 for each starting point. Fig. 3.2 shows that, on this problem, IF2 is considerably more effective than IF1. This maximization was considered challenging for IF1, and *King et al.* (2008) required multiple restarts and refinements of the optimization procedure.

Our implementation of PMCMC failed to converge on this inference problem (see A.5), and we are not aware of any previous successful PMCMC solution for a comparable situation. For IF2, however, this situation appears routine. Some Monte Carlo replication is needed because searches occasionally fail to approach the global optimum, but replication is always appropriate for Monte Carlo optimization procedures. A fair numerical comparison of methods is difficult. For example, it could hypothetically be the case that the algorithmic settings used here favor IF2. However, the settings used are those that were developed for IF1 by *King et al.* (2008) and reflect considerable amounts of trial and error with that method. Likelihood-based inference for general partially observed nonlinear stochastic dynamic models was considered computationally unfeasible prior to the introduction of IF1, even in situations considerably simpler than the one investigated in this section (*Wood*, 2010). We have shown that IF2 offers a substantial improvement on IF1, by demonstrating that it functions effectively on a problem at the limit of the capabilities of IF1.

3.6 Discussion

Theorems III.1 and III.2 assert convergence without giving insights into the rate of convergence. In the particular case of a quadratic log likelihood function and additive Gaussian parameter perturbations, $\lim_{M\to\infty} T_\sigma^M f$ is Gaussian, and explicit calculations are available (see A.3). If $\log \ell(heta)$ is close to quadratic and the parameter perturbation is close to additive Gaussian noise, then $\lim_{M\to\infty} T_\sigma^M f$ exists and is close to the limit for the approximating Gaussian system (see A.3). These Gaussian and near-Gaussian situations also demonstrate that the compactness conditions for

Theorem III.2 are not always necessary. In the case $N = 1$, IF2 applies to the more general class of latent variable models. The latent variable model, extended to include a parameter vector that varies over iterations, nevertheless has the formal structure of a POMP in the context of the IF2 algorithm. Some simplifications arise when $N = 1$ (see A.3 and A.4) but the proofs of Theorems 1 and 2 do not greatly change. A variation on iterated filtering, making white noise perturbations to the parameter rather than random walk perturbations, has favorable asymptotic properties (*Doucet et al.*, 2013).

However, practical algorithms based on this theoretical insight have not yet been published. Our experience suggests that white noise perturbations can be effective in a neighborhood of the MLE, but fail to match the performance of IF2 for global optimization problems in complex models. The main theoretical innovation of this chapter is Theorem III.2, which does not depend on the specific sequential Monte Carlo filter used in IF2. One could, for example, modify IF2 to use an ensemble Kalman filter (*Shaman and Karspeck*, 2012; *Yang et al.*, 2014) or an unscented Kalman filter (*Julier and Uhlmann*, 2004). Or, one could take advantage of variations of sequential Monte Carlo that may improve the numerical performance (*Cappé et al.*, 2007). However, basic sequential Monte Carlo is a general and widely used nonlinear filtering technique that provides a simple yet theoretically supported foundation for the IF2 algorithm. The numerical stability of sequential Monte Carlo for the extended POMP model constructed by IF2 is comparable, in our cholera example, to the model with fixed parameters (see A.7).

CHAPTER IV

Second-order Iterated Smoothing

During the past three decades, partially observed Markov process (POMP) models (also known as state space models) have become ubiquitous tools for modeling and time series data analysis of time series data in many disciplines, including econometrics, ecology and engineering. However, it can be difficult to make inferences about non-linear or non-Gaussian POMP models owing to the fact that there is no closed form expression for the likelihood function. Linear Gaussian models enable exact likelihood computation, via the Kalman filter, but can lead to unsatisfactory results when the assumptions are violated. In many situations, the transition probability density is intractable or too expensive to evaluate, but easy to sample from (*Bretó et al.*, 2009). Therefore, there has been a surge of interest in simulation-based inference for POMP models (*Ionides et al.*, 2006; *Toni et al.*, 2009; *Andrieu et al.*, 2010; *Wood*, 2010; *Chopin et al.*, 2013; *Ionides et al.*, 2015). Simulation-based methods have also been called plug-and-play (*Bretó et al.*, 2009; *He et al.*, 2010), likelihood-free (*Sisson et al.*, 2007; *Yıldırım et al.*, 2015) or equation-free (*Kevrekidis et al.*, 2004). These methodologies can be categorized into either Bayesian or frequentist approaches, and further categorized into full information or partial information approaches. Full information approaches are those which are based on the full likelihood of the data; partial information approaches are those based on summary statistics or

quasi-likelihoods, such as approximate Bayesian computing (*Toni et al.*, 2009) or synthetic likelihood (*Wood*, 2010). Here, we are concerned with full information, frequentist, simulation-based inference. The first algorithm developed to carry out such inference was the iterated filtering algorithm of *Ionides et al.* (2006), which we will call IF1. The theoretical properties of IF1 were studied by *Ionides et al.* (2011). *Doucet et al.* (2013) proposed some improvements to this algorithm by further exploiting both the score vector and the observed information matrix to improve the convergence rate. The algorithm of *Doucet et al.* (2013) involves using sequential Monte Carlo methods to carry out iterated smoothing, and we call their algorithm IS1. *Doucet et al.* (2013) showed that IS1 has second order convergence properties. However, in practical problems, IS1 has failed to show practical performance living up to its favorable asymptotic theory. This chapter develops a modification of the theory of *Doucet et al.* (2013) giving rise to a new algorithm, that we call IS2, which empirically shows clearly enhanced performance over IF1 and IS1 on our benchmarks in Section 4.4. Recently, a new iterated filtering algorithm, which we call IF2, has been developed with a different theoretical justification based on iterated perturbed Bayes maps (*Ionides et al.*, 2015). IS2 shows comparable performance to IF2 on our benchmarks. The substantial differences—both in the theoretical foundations and the resulting algorithms—between IF2 and IS2 indicate that IS2 provides a promising alternative approach to IF2 for future theoretical and methodological developments.

The key contributions of this chapter are three-fold. First, we demonstrate theoretically that random walk parameter perturbations can be used in place of the white noise perturbations of IS1 while preserving much of the theoretical support provided by *Doucet et al.* (2013). In particular, IS2 inherits second-order convergence properties from IS1. Second, we discover that the approximation of the observed information matrix using random walk noise is simpler than that using independent white noise. Consequently, IS2 enjoys a computationally cheap estimate of the observed informa-

tion matrix. Third, IS2 is not only attractive in theory, but we show it also has good numerical performance in practice.

The chapter is organized as follows. In Section 4.1, we introduce some notation and discuss some background to the computational challenge we investigate. In Section 4.2, we develop the required theory in the context of latent variable models, which are later extended to the case of partially observed stochastic dynamic systems. In Section 4.3, we state our theorems and present the IS2 algorithm, postponing proofs of our results to the appendix. Section 4.4 presents a toy problem and a challenging inference problem of fitting a malaria transmission model to time series data, showing empirical results in which IS2 beats IF1 and IS1 while performing comparably to IF2. Section 6.6 is a concluding discussion.

4.1 Problem definition

For POMP model defined as in Section 2, we seek the maximum likelihood estimator, $\hat{\theta} = \arg\max \ell(\theta)$. Maximization of the likelihood function using first order stochastic approximation (*Kushner and Clark*, 1978) involves a Monte Carlo approximation to a difference equation,

$$\theta_m = \theta_{m-1} + \gamma_m \nabla \ell(\theta_{m-1}),$$

where $\theta_0 \in \Theta$ is an arbitrary initial estimate and $\{\gamma_m\}_{m\geq 1}$ is a sequence of step sizes with $\sum_{m\geq 1} \gamma_m = \infty$ and $\sum_{m\geq 1} \gamma_m^2 < \infty$. Under appropriate regularity conditions, the algorithm converges to a local maximum of $\ell(\theta)$. The term $\nabla\ell(\theta)$ is shorthand for the $\mathbb{R}^d$-valued vector of partial derivatives,

$$\nabla \ell(\theta) = \frac{\partial \ell(\theta)}{\partial \theta},$$

which is also called the score vector. Stochastic approximation methods can sometimes be improved by exploiting the observed information matrix as in a Newton-Raphson approaches (*Spall*, 2003). In these second-order methods, the convergence rate is improved by using $-\{\nabla^2\ell(\theta)\}^{-1}$ in place of the step size γ_m, where $\nabla^2\ell(\theta)$ is the $d \times d$ Hessian matrix of $\ell(\theta)$, written with some abuse of notation as

$$\nabla^2\ell(\theta) = \nabla\nabla'\ell(\theta) = \frac{\partial^2\ell(\theta)}{\partial\theta^2}$$

where ∇' is a row vector of partial derivatives. The matrix $\nabla^2\ell(\theta)$ is also known as the observed information at θ. Carrying out a Newton-Raphson approach via a simulation-based algorithm boils down to simulation-based estimation of the score vector and observed information matrix.

Sequential Monte Carlo (SMC) approaches have previously been developed to estimate the score and observed information (*Poyiadjis et al.*, 2011; *Nemeth et al.*, 2013; *Dahlin et al.*, 2015). However these methods require the ability to evaluate transition densities, and sometimes also their derivatives, and so do not have the plug-and-play property of *Bretó et al.* (2009). One alternative plug-and-play approach which also approximates the score and observed information is finite difference method. However, this involves carrying out multiple independent filtering operations for a single Monte Carlo gradient estimate, causing significant computational burden in many practical situations. Moreover, it can also result in high variance estimates, which unfortunately are unsolved for plug-and-play setup by current variance reduction techniques in the literature (*Doucet et al.*, 2013). As a plug-and-play alternative, *Doucet et al.* (2013) used an artificial dynamics approach to estimate the observed information matrix using sequential Monte Carlo smoothing. The approach of *Doucet et al.* can be computationally intensive, reducing its practical advantage over the first order method of *Ionides et al.* (2011). We propose a computationally less demanding

approximation to the score and observed information. Theoretical properties of these approximation are shown in theorems IV.8 and IV.9, and more numerically stable approximations of these quantities are investigated in Theorems IV.10 and IV.11. Following the approach of *Ionides et al.* (2011) and *Doucet et al.* (2013), we first develop our theory (in Section 4.2) in the context of a latent variable model. Then, in Section 4.3, we extend this to the POMP framework.

4.2 Perturbed parameters and a latent variable model

Consider a parametric model consisting of a density $p_Y(y;\theta)$ with the log-likelihood of the data $y^* \in \mathcal{Y}$ given by $\ell(\theta) = \log p_Y(y^*;\theta)$. We define a stochastically perturbed model corresponding to a pair of random variables $(\breve{\Theta}, \breve{Y})$ having a joint probability density on $\mathbb{R}^d \times \mathcal{Y}$ given by

$$p_{\breve{\Theta},\breve{Y}}(\breve{\vartheta},\ y;\theta,\ \tau) = \tau^{-d}\kappa\left\{\tau^{-1}(\breve{\vartheta}-\theta)\right\} p_Y(y;\breve{\vartheta}).$$

Using a Taylor expansion up to the second order, *Ionides et al.* (2011) approximated the score function $\nabla\ell(\theta)$ in terms of moments of the conditional distribution of $\breve{\Theta}$ given $Y = y^*$. *Doucet et al.* (2013) developed a Taylor expansion to the fourth order and approximated both the score function $\nabla\ell(\theta)$ and the observed information matrix $\nabla^2\ell(\theta)$. The following lemmas for stochastically perturbed models are restated from *Doucet et al.*'s Theorems 2 and 3, since they are foundations for our proofs. We denote $|\cdot|$ the L_1-norm. For any vector $u \in \mathbb{R}^d$, $|u| = \sum_{i=1}^d |u_i|$ and for any matrix $v \in \mathbb{R}^{d\times d}$, $|v| = \sum_{i=1}^d \sum_{j=1}^d |v_{ij}|$. We suppose the following regularity conditions, identical to the assumptions of *Doucet et al.* (2013):

Assumption IV.1. *There exists $C < \infty$ such that for any integer $k \geq 1, 1 \leq$*

$i_1, \ldots, i_k \leq d$ *and* $\beta_1, \ldots, \beta_k \geq 1$,

$$\int \left| u_{i_1}^{\beta_1} u_{i_2}^{\beta_2} \cdots u_{i_k}^{\beta_k} \right| \kappa(u)\, du \leq C,$$

where κ *is a symmetric probability density on* $\mathbb{R}^d$ *with respect to Lebesgue measure and* $\Sigma = (\sigma_{i,j})_{i,j=1}^d$ *is the non-singular covariance matrix associated to* κ.

Assumption IV.2. *There exist* $\gamma,\ \delta,\ M > 0$, *such that for all* $u \in \mathbb{R}^d$,

$$|u| > M \Rightarrow \kappa(u) < e^{-\gamma |u|^\delta}.$$

Assumption IV.3. ℓ *is four times continuously differentiable and* δ *defined as in Assumption IV.2. For all* $\theta \in \mathbb{R}^d$, *there exists* $0 < \eta < \delta,\ \epsilon,\ D > 0$, *such that for all* $u \in \mathbb{R}^d$,

$$\mathcal{L}(\theta + u) \leq D e^{\epsilon |u|^\eta},$$

where $\mathcal{L} : \mathbb{R}^d \to \mathbb{R}$ *is the associated likelihood function* $\mathcal{L} = \exp \ell$.

Assumption IV.4. κ *satisfies* $\int u_i^4 \kappa(u) du = 3\sigma_i^4$.

In practice, κ can be chosen by the users and it is often convenient to chose a multivariate normal distribution, so that all four assumptions are satisfied explicitly. Parameters with a bounded domain can be transformed so that they are defined on the entire real line, to enable the applicability of multivariate normal perturbations.

Lemma IV.5. *(Doucet et al. Theorem 2) Suppose assumption IV.1, IV.2, IV.3, there exists a constant* C *such that:*

$$\left| \breve{\mathbb{E}} \left(\breve{\Theta} - \theta \,\middle|\, \breve{Y} = y^* \right) - \tau^2 \Sigma \nabla \ell(\theta) \right| < C\tau^4. \tag{4.1}$$

In order to prove the approximation of observed information matrix, *Doucet et al.* (2013) further assumed regularity of the perturbation kernel. Specifically, a non-

singular symmetric kernel was assumed, which is consistent with the practical choice of Gaussian perturbations.

Lemma IV.6. *(Doucet et al. Theorem 3) Suppose assumption IV.1, IV.2, IV.3 and IV.4, there exists a constant C such that:*

$$\left| \breve{\mathbb{E}} \left[\left(\breve{\Theta} - \theta \right) \left(\breve{\Theta} - \theta \right)^{\top} \middle| \breve{Y} = y^* \right] - \tau^2 \Sigma - \tau^4 \Sigma \left(\nabla^2 \ell(\theta) \right) \Sigma \right| < C\tau^6. \tag{4.2}$$

These approximations are useful for latent variable models, where the log likelihood of the model consists of marginalizing over a latent variable, X,

$$\ell(\theta) = \log \int p_{X,Y}(x, y^*; \theta)\, dx.$$

In this case, the expectations in Lemmas IV.5 and IV.6 can be approximated by Monte Carlo importance sampling, as proposed by *Ionides et al.* (2011) and *Doucet et al.* (2013). Results such as Lemma 1 and Lemma 2 do have potential applicability to situations other than POMP models, as discussed by *Doucet et al.* (2013) and *Ionides et al.* (2011, 2015). Here, our focus is on POMP models, which is the model class for which these methods have primarily been used.The latent variable setup we consider is identical to that of *Doucet et al.* (2013), which is also similar to that of *Ionides et al.* (2011). The three approaches become more distinct in their consequences for the extension from latent variable models to POMP models.

4.3 An iterated smoothing algorithm

The POMP model is a specific latent variable model with $X = X_{0:N}$ and $Y = Y_{1:N}$. We define a perturbed POMP model having a similar construction to our perturbed latent variable model with $\breve{X} = \breve{X}_{0:N}$, $\breve{Y} = \breve{Y}_{1:N}$ and $\breve{\Theta} = \breve{\Theta}_{0:N}$. *Ionides et al.* (2011) perturbed the parameters by setting $\breve{\Theta}_{0:N}$ to be a random walk starting at θ, whereas

Doucet et al. (2013) took $\breve{\Theta}_{0:N}$ to be independent additive white noise perturbations of θ. Our goal is to take advantage of the asymptotic developments of *Doucet et al.* (2013) while maintaining some practical advantages of random walk perturbations for finite computations. Specifically, we construct $\breve{\Theta}_{0:N}$ as follows.

Let $Z_0, \ \ldots, \ Z_N$ be $N+1$ independent draws from a density ψ. We introduce $N+2$ perturbation parameters, τ and $\tau_0, \ldots, \tau_N$, and construct a process $\breve{\Theta}_{0:N}$ by setting

$$\breve{\Theta}_n = \theta + \tau \sum_{i=0}^{n} \tau_i Z_i$$

for $0 \leq n \leq N$. We will later consider a limit where $\tau_{0:N}$ as fixed and the scale factor τ decreases toward zero, and subsequently another a limit where τ_0 is fixed but $\tau_{1:N}$ decrease toward zero together with τ. Let $p_{\breve{\Theta}_{0:N}}(\breve{\vartheta}_{0:N}; \theta, \ \tau, \ \tau_{0:N})$ be the probability density of $\breve{\Theta}_{0:N}$. We define the artificial random variables $\breve{\Theta}_{0:N}$ via their density,

$$p_{\breve{\Theta}_{0:N}}(\breve{\vartheta}_{0:N}; \theta, \ \tau, \ \tau_{0:N}) = (\tau\tau_0)^{-d} \psi \left\{ (\tau\tau_0)^{-1} (\breve{\vartheta}_0 - \theta) \right\} \\ \times \prod_{n=1}^{N} (\tau\tau_n)^{-d} \psi \left\{ (\tau\tau_n)^{-1} (\breve{\vartheta}_t - \breve{\vartheta}_{t-1}) \right\}.$$

We define the stochastically perturbed model with a Markov process $\{(\breve{X}_n, \breve{\Theta}_n), \ 0 \leq n \leq N\}$, observation process $\breve{Y}_{1:N}$ and parameter $(\theta, \ \tau, \ \tau_{0:N})$ by the factorization of their joint probability density

$$p_{\breve{X}_{0:N}, \breve{Y}_{1:N}, \breve{\Theta}_{0:N}}(x_{0:N}, y_{1:N}, \breve{\vartheta}_{0:N}; \theta, \ \tau, \ \tau_{0:N}) \\ = p_{\breve{\Theta}_{0:N}}(\breve{\vartheta}_{0:N}; \theta, \ \tau, \ \tau_{0:N}) p_{\breve{X}_{0:N}, \breve{Y}_{1:N} | \breve{\Theta}_{0:N}}(x_{0:N}, \ y_{1:N} | \breve{\vartheta}_{0:N}),$$

where

$$p_{\breve{X}_{0:N},\breve{Y}_{1:N}|\breve{\Theta}_{0:N}}(x_{0:N}, y_{1:N}|\breve{\vartheta}_{0:N}; \theta, \tau,\ \tau_{0:N}) \\ = \mu(x_0; \breve{\vartheta}_0) \prod_{n=1}^{N} f_n(x_n|x_{n-1}; \breve{\vartheta}_n) \prod_{n=1}^{N} g_n(y_n|x_n; \breve{\vartheta}_n)$$

This extended model can be used to define a perturbed parameter log-likelihood function, defined as

$$\breve{\ell}(\breve{\vartheta}_{0:N}) = \log p_{\breve{Y}_{1:N}|\breve{\Theta}_{0:N}}(y^*_{1:N}|\breve{\vartheta}_{0:N}; \theta, \tau, \tau_{0:N}). \tag{4.3}$$

Here, we are treating the data as fixed and note that the right hand side does not depend on θ, τ or $\tau_{0:N}$. We have designed (4.3) so that, setting

$$\breve{\vartheta}^{[N+1]} = (\theta, \theta, \ldots, \theta) \in \mathbb{R}^{d(N+1)},$$

we can write the log-likelihood of the unperturbed model as

$$\ell(\theta) = \breve{\ell}(\breve{\vartheta}^{[N+1]}).$$

In our POMP framework, $p_{\breve{\Theta}_{0:N}}$ is analogous to κ in the general latent variable model. However, to formally match these two frameworks we must bear in mind that $p_{\breve{\Theta}_{0:N}}$ carries out perturbations in $\Theta^{(N+1)d}$, so Lemmas 1 and 2 must be applied in that extended parameter space.

For our perturbed likelihood, we need an extended version of assumption IV.3, identical to assumption 5 of *Doucet et al.* (2013).

Assumption IV.7. *$\breve{\ell}$ is four times continuously differentiable. For all $\theta \in \mathbb{R}^d$, there exist $\epsilon > 0$, $D > 0$ and δ defined as in Assumption IV.2, such that for all $0 < \eta < \delta$*

and $u_{0:N} \in \mathbb{R}^{d(N+1)}$,

$$\breve{\mathcal{L}}(\breve{\vartheta}^{[N+1]} + u_{0:N}) \leq De^{\epsilon \sum_{n=1}^{N} |u_n|^\eta},$$

where $\breve{\mathcal{L}}(\breve{\vartheta}_{0:N}) = \exp\{\breve{\ell}(\breve{\vartheta}_{0:N})\}$ *is the perturbed likelihood.*

Let $\breve{\mathbb{E}}_{\theta,\tau,\tau_{0:N}}$, $\breve{\mathrm{C}}\mathrm{ov}_{\theta,\tau,\tau_{0:N}}$, $\breve{\mathrm{V}}\mathrm{ar}_{\theta,\tau,\tau_{0:N}}$ denote as the expectation, covariance and variance with respect to the associated posterior,

$$p_{\breve{\Theta}_{0:N}|\breve{Y}_{1:N}}(\breve{\vartheta}_{0:N}|y^*_{1:N}; \theta,\ \tau, \tau_{0:N}).$$

To simplify the heavy notation, hereafter, we will use $\breve{\mathbb{E}}$, $\breve{\mathrm{C}}\mathrm{ov}$, $\breve{\mathrm{V}}\mathrm{ar}$ instead of $\breve{\mathbb{E}}_{\theta,\tau,\tau_{0:N}}$, $\breve{\mathrm{C}}\mathrm{ov}_{\theta,\tau,\tau_{0:N}}$, $\breve{\mathrm{V}}\mathrm{ar}_{\theta,\tau,\tau_{0:N}}$ respectively. The following theorems IV.8 and IV.9 are our main results, they are similar to theorem 4 and 6 of *Doucet et al.* (2013) but for random walk noise instead of independent white noise and are much simpler.

Theorem IV.8. *Suppose assumption IV.1, IV.2 and IV.7, there exists a constant* C *independent of* $\tau, \tau_1, ...\tau_N$ *such that,*

$$\left|\nabla \ell\left(\theta\right) - \tau^{-2}\Psi^{-1}\left\{\tau_0^{-2}\breve{\mathbb{E}}\left(\breve{\Theta}_0 - \theta | \breve{Y}_{1:N} = y^*_{1:N}\right)\right\}\right| < C\tau^2,$$

where Ψ *is the non-singular covariance matrix associated to* ψ.

Proof. See appendix A.11.1. □

Note that the constant C in Theorem 1 may depend on N, the length of the time series. Here we consider the dataset to be fixed, so N is constant and we do not need to concern ourselves with the issue of how C scales with N.

We propose to use a random walk in parameter space to explore the likelihood surface. These random walk perturbations can investigate the struture of the likelihood surface in a sequence of small steps. This allows some searching of parameter space within a single filtering iteration, which may be more computationally efficient

than approaches, such as *Doucet et al.* (2013), which make white noise perturbations around a parameter value that is fixed for the duration of a filtering operation. However, one might suspect that the random walk perturbations should break the asymptotic guarantees developed by *Doucet et al.* (2013) for white noise perturbations. We show that this is not the case; random walk perturbations enjoy much of the theoretical support developed by *Doucet et al.* (2013), while being more computationally efficient empirically.We first state our theorems, leaving proofs to the appendix.

Theorem IV.9. *Suppose assumptions IV.1, IV.2, IV.4 and IV.7, there exist a C independent of $\tau, \tau_1, ...\tau_N$ such that the following hold true for random walk noise,*

$$-\nabla^2 \ell(\theta) = I_\tau(\theta) + C\tau^2,$$

where

$$I_\tau(\theta) = - \tau^{-4}\Psi^{-1}\left\{\tau_0^{-4}\left(\breve{\mathrm{V}}\mathrm{ar}\left(\breve{\Theta}_0|\breve{Y}_{1:N} = y^*_{1:N}\right) - \tau_0^2\tau^2\Psi\right)\right\}\Psi^{-1}. \quad (4.4)$$

Proof. See appendix A.11.2. □

Theorem IV.8 and IV.9 formally allow approximation of $\nabla\ell(\theta)$, $-\nabla^2\ell(\theta)$. However, they rely heavily on the computation of the conditional distribution of $\breve{\Theta}_0$ given $Y_{1:N}$, which is a computationally challenging smoothing problem. We therefore present some alternative variations on these results which lead to more stable Monte Carlo estimation. Our Theorems IV.10 and IV.11 consider a limit where τ_n is of order τ^2 for each $1 \leq n \leq N$, as $\tau \to 0$. This limit is similar to a limit studied in the context of IF1 by *Ionides et al.* (2011). This is not an ideal theoretical framework, since it approaches another limit which involves numerically difficult smoothing cal-

culations. However, the theorems can still carry out the useful purpose of motivating new algorithms whose finite sample properties are assessed empirically. We state two additional theorems as follows.

Theorem IV.10. *Suppose assumption IV.1, IV.2 and IV.7 hold. In addition, assume that* $\tau_n = O(\tau^2)$ *for all* $n = 1 \dots N$*, the following hold true,*

$$\left| \nabla \ell(\theta) - \frac{1}{N+1} \tau^{-2} \tau_0^{-2} \Psi^{-1} \sum_{n=0}^{N} \left\{ \breve{\mathbb{E}} \left(\breve{\Theta}_n - \theta | \breve{Y}_{1:N} = y^*_{1:N} \right) \right\} \right| = O(\tau^2). \quad (4.5)$$

Proof. See appendix A.11.3. □

Theorem IV.11. *Suppose assumptions IV.1, IV.2, IV.4 and IV.7 hold. In addition, assume that* $\tau_n = O(\tau^2)$ *for all* $n = 1 \dots N$*, the following hold true for random walk noise,*

$$-\nabla^2 \ell(\theta) = I_\tau(\theta) + O(\tau^2),$$

where

$$I_\tau(\theta) = -\frac{1}{N+1} \tau^{-4} \tau_0^{-4} \Psi^{-1} \left\{ \sum_{n=0}^{N} \left(\breve{\mathrm{V}}\mathrm{ar} \left(\breve{\Theta}_n | \breve{Y}_{1:N} = y^*_{1:N} \right) - \sum_{k=0}^{n} \tau_k^2 \Psi \right) \right\} \Psi^{-1}.$$

Proof. See appendix A.11.4. □

To the best of our knowledge, the approach of *Doucet et al.* (2013) has not previously been used for data analysis. In part, this could be due to the computational expense of its estimation of the covariance matrix estimation for the perturbed parameters. The computational cost of the full covariance estimation in the method of *Doucet et al.* (2013), between all pairs of time points, is $\mathcal{O}(N^2)$ at each time point and

so $\mathcal{O}(N^3)$ for an entire smoothing computation. As proposed by *Doucet et al.* (2013), one can omit covariances larger than some lag L, and one can use this same lag L for a fixed-lag particle smoothing algorithm using J particles. *Doucet et al.* (2013) studied the properties of such an algorithm, under strong mixing assumptions, to derive an algorithm with computational cost $\mathcal{O}(NL^2J)$. Here, we write equivalent results for our algorithm, based on the results proved by *Doucet et al.* (2013). These result study the approximation properties of the score function and information matrix estimators for specific values of θ and τ. Full analysis of Algorithm 1 using stochastic approximation theory, as in *Ionides et al.* (2011), would require some uniformity of this approximation when τ is small and θ is in a neighborhood of the maximum of the likelihood function. Specifically, we make the following assumption essentially identical to assumption 6 of *Doucet et al.*:

Assumption IV.12. *For $\tau \in \mathbb{R}^+$, $\theta \in \mathbb{R}^d$ and $n \in \{2, .., N\}$ define*

$$\begin{aligned} S_1(\theta,\tau) &= \left[\phi \in \mathbb{R}^d : \kappa\left\{(\phi-\theta)/\tau\right\} > 0\right], \\ S_n(\theta,\tau) &= \left[\phi \in \mathbb{R}^d : for\ every\ \theta' \in S_{n-1}(\theta,\tau), \kappa\left\{(\phi-\theta')/\tau\right\} > 0\right]. \end{aligned}$$

There is some $b > 0$ such that, for $B = [0, b]$, the following conditions hold.

1. *$S_n(\theta, \tau)$ is compact for all $n \in \{1, .., N\}$.*

2. *for all $n \in \{1, .., N\}$,*

$$\begin{aligned} \underline{\alpha}_n(\theta) &= \inf_{\tau\in B,\phi\in S_n(\theta,\tau),x\in\mathcal{X},x'\in\mathcal{X}} f_n(x'|x;\phi) > 0, \\ \overline{\alpha}_n(\theta) &= \sup_{\tau\in B,\phi\in S_n(\theta,\tau),x\in\mathcal{X},x'\in\mathcal{X}} f_n(x'|x;\phi) < \infty, \\ \rho_n(\theta) &= 1 - \underline{\alpha}_n(\theta)/\overline{\alpha}_n(\theta) > 0. \end{aligned}$$

Let $\rho(\theta) = \max_{n\in\{1,..,N\}} \rho_n(\theta)$.

3. *For $n \in \{2, .., N\}$, assume that there exists a probability measure $\lambda(dx)$ on $\mathcal{X}$, define*

$$\begin{aligned} h_1(\theta, \phi, \tau) &= \int g_1(y|x;\phi)\tau^{-d}\kappa\left\{(\phi-\theta)/\tau\right\} d\phi\mu(x;\theta)\lambda(dx), \\ h_n(\theta, \phi, \tau) &= \int g_n(y|x;\phi)\tau^{-d}\kappa\left\{(\phi-\theta)/\tau\right\} d\phi\lambda(dx). \end{aligned}$$

then for all $y \in \mathcal{Y}$,

$$\begin{aligned} \overline{g}_n(y;\theta) &= \sup_{\tau\in B,\phi\in S_n(\theta,\tau),x\in\mathcal{X}} g_n(y|x;\phi) < \infty, \\ \underline{g}_n(y;\theta) &= \inf_{\tau\in B,\phi\in S_1(\theta',\tau),\theta'\in S_{n-1}(\theta,\tau)} h_n(\theta',\phi,\tau) > 0 \\ \overline{g}_1(y;\theta) &= \sup_{\tau\in B,\phi\in S_1(\theta,\tau),x\in\mathcal{X}} g_1(y|x;\phi) < \infty, \\ \underline{g}_1(y;\theta) &= \inf_{\tau\in B,\phi\in S_1(\theta,\tau)} h_1(\theta,\phi,\tau) > 0 \end{aligned}$$

Theorem IV.13. *Suppose assumption IV.12, the following hold true for random walk noise:*

$$\tau^2\Psi S_{\tau,N}(\theta) = \tau^2\Psi S_{\tau,L,N}(\theta) + O(\rho(\theta)^L),$$

$$\tau^4\Psi\left\{I_{\tau,N}(\theta)\right\}\Psi = \tau^4\Psi I_{\tau,L,N}(\theta)\Psi + O(\rho(\theta)^L),$$

where

$$S_{\tau,N}(\theta) = \frac{1}{N+1}\tau^{-2}\tau_0^{-2}\Psi^{-1}\left\{\sum_{n=0}^{N}\left(\breve{\mathbb{E}}(\breve{\Theta}_n|\breve{Y}_{1:N} = y^*_{1:N}) - \theta\right)\right\},$$

$$S_{\tau,L,N}(\theta) = \frac{1}{N+1}\tau^{-2}\tau_0^{-2}\Psi^{-1} \left\{\sum_{n=0}^{N}\left(\breve{\mathbb{E}}(\breve{\Theta}_n|\breve{Y}_{1:(n+L)\wedge N} = y^*_{1:(n+L)\wedge N}) - \theta\right)\right\},$$

$$I_{\tau,N}(\theta) = -\frac{1}{N+1}\tau^{-4}\tau_0^{-4}\Psi^{-1} \left\{\sum_{n=0}^{N}\left(\breve{\mathrm{V}}\mathrm{ar}\left(\breve{\Theta}_n|\breve{Y}_{1:N} = y^*_{1:N}\right) - \sum_{k=0}^{n}\tau_k^2\Psi\right)\right\}\Psi^{-1},$$

$$I_{\tau,L,N}(\theta) = -\frac{1}{N+1}\tau^{-4}\tau_0^{-4}\Psi^{-1} \left\{\sum_{n=0}^{N}\left(\breve{\mathrm{V}}\mathrm{ar}\left(\breve{\Theta}_n|\breve{Y}_{1:(n+L)\wedge N} = y^*_{1:(n+L)\wedge N}\right) - \sum_{k=0}^{n}\tau_k^2\Psi\right)\right\}\Psi^{-1}.$$

Proof. It follows directly from *Olsson et al.* (2008), as in the proof of proposition 7 of *Doucet et al.* (2013). □

For completeness, we also state a Monte Carlo approximation result which is essentially identical to proposition 8 of *Doucet et al.* (2013).

Theorem IV.14. *(Doucet et al., Proposition 8). Suppose assumption IV.12, then for all integers $N \geq 1, 0 \leq L \leq N-1$, $J \geq 1$ is number of particles and for any $p \geq 2$, there exist constants C and C_p, not depending on J, such that:*

$$\tau^2\left|\mathbb{E}\left[\Psi\left\{S^J_{\tau,L,N}(\theta) - S_{\tau,N}(\theta)\right\}\right]\right| \leq \frac{C}{J},$$

$$\tau^4\left|\mathbb{E}\left[\Psi\left\{I^J_{\tau,L,N}(\theta) - I_{\tau,L,N}(\theta)\right\}\Psi\right]\right| \leq \frac{C}{J},$$

and

$$\tau^2\mathbb{E}^{1/p}\left[\left|\Psi\left\{S^J_{\tau,L,N}(\theta) - S_{\tau,L,N}(\theta)\right\}\right|^p\right] \leq \frac{C_p}{\sqrt{J}},$$

$$\tau^4\mathbb{E}^{1/p}\left[\left|\Psi\left\{I^J_{\tau,L,N}(\theta) - I_{\tau,L,N}(\theta)\right\}\Psi\right|^p\right] \leq \frac{C_p}{\sqrt{J}}.$$

where the expectations are with respect to the law of the bootstrap filter.

Pseudo-code for second order iterated smoothing (IS2) is given in Algorithm 1. The initial values of the state variables at time t_0 can be treated as unknown param-

eters. We call them initial value parameters (IVPs), and denote them in Algorithm 1 by a subset of the parameter indices, $I \subset \{1, \ldots, d\}$. Since IVP affect the dynamics only at time t_0, there can be no benefit to perturbing the parameters at other times, and so IVPs benefit from some special attention. The perturbations in lines 2 and 5 are taken to follow the normal distribution, though alternative densities with matching mean and variance could be chosen. Pseudo-code for the IS1 algorithm of *Doucet et al.* (2013) is given in the supplement (Algorithm S-1). Since IS1 is too computational expensive to apply in real problems, we propose a reduced second order iterated smoothing (RIS1) approach. The RIS1 algorithm is the same as Algorithm 1 except that we use white noise to update filter at each time point, in steps 2 and 5.

The IS2 algorithm, together with IS1 and RIS1 algorithms based on (*Doucet et al.*, 2013), were implemented in an open-source R package *is2* (*Nguyen and Ionides*, 2015), which is built based on *pomp* package (*King et al.*, 2015c). For efficient partile filter implementation, here we use systematic resampling instead of using multinomial resampling as in original bootstrap particle filter of *Gordon et al.* (1993).

Main features of IF1, IF2, IS1, RIS1 and IS2 are summarized in Table 4.1.

4.4 Numerical examples

4.4.1 Toy example: A linear, Gaussian model

In this section, we evaluate our algorithm, comparing it to existing simulation-based approaches in term of statistical performance and computational efficiency. We consider a bivariate discrete time Gaussian autoregressive process, with Gaussian measurement error. This model is chosen so that the Monte Carlo calculations can

Algorithm 1: Iterating smoothing (IS2): `is2(P, start=`θ_0`, Nmif=`M`, Np=`J`, rw.sd=`$\sigma_{1:p}$`, ic.lag=`L`, var.factor=`C`, cooling.factor=`a`)`, using notation from 6.1 where P is a pomp object with defined rprocess, dmeasure, init.state, and obs components.

input: Starting parameter, θ_0; simulator for $f_{X_0}(x_0\,;\theta)$; simulator for $f_{X_n|X_{n-1}}(x_n\,|\,x_{n-1}\,;\theta)$; evaluator for $f_{Y_n|X_n}(y_n\,|\,x_n\,;\theta)$; data, $y^*_{1:N}$; labels, $I\subset\{1,\dots,p\}$, designating IVPs; fixed lag, L, for estimating initial value parameters (IVPs); number of particles, J, number of iterations, M; cooling rate, $0<a<1$; perturbation scales, $\sigma_{1:p}$; initial scale multiplier, $C>0$.

for m *in* $1{:}M$ **do**
  Initialize: $[\Theta^F_{0,j}]_i \sim \text{Normal}\left([\theta_0]_i, (Ca^{m-1}\sigma_i)^2\right)$ for i in $1{:}p$, j in $1:J$.
  Initialize states: simulate $X^F_{0,j}\sim f_{X_0}\big(\,\cdot\,;\Theta^F_{0,j}\big)$ for j in $1{:}J$.
  Initialize filter mean for parameters: $\bar\theta_0=\theta_0$.
  for n *in* $1:N$ **do**
    Perturb: $\big[\Theta^P_{n,j}\big]_i\sim\mathcal{N}\big(\big[\Theta^F_{n-1,j}\big]_i,(c^{m-1}\sigma_i)^2\big)$ for $i\notin I$, j in $1:J$.
    Simulate prediction particles: $X^P_{n,j}\sim f_n\big(x_n|X^F_{n-1,j};\Theta^P_{n,j}\big)$ for j in $1:J$.
    Evaluate weights: $w(n,j)=g_n(y^*_n|X^P_{n,j};\Theta^P_{n,j})$ for j in $1:J$.
    Normalize weights: $\breve w(n,j)=w(n,j)/\sum_{u=1}^J w(n,u)$.
    Apply 3 to select indices $k_{1:J}$ with $P\{k_u=j\}=\breve w\,(n,j)$.
    Resample particles: $X^F_{n,j}=X^P_{n,k_j}$ and $\Theta^F_{n,j}=\Theta^P_{n,k_j}$ for j in $1:J$.
    Define and store ancestor let $a_1(n,k_j)=j$,
    $a_{l+1}(n,j)=a_1(n-l,a_l(n,j))$ for j in $1:J$, l in $1:L-1$
    Smooth mean: $\bar\theta^L_{n-L}=\sum_{j=1}^J\breve w(n,j)\Theta^P_{n-L,a_L(n,j)}$ if $n>L$.
    Smooth variance: $V^m_{n-L,n-L}=\sum_j\breve w(n,j)\big(\Theta^P_{n-L,a_L(n,j)}-\bar\theta^L_{n-L}\big)\big(\Theta^P_{n-L,a_L(n,j)}-\bar\theta^L_{n-L}\big)^\top$ if $n>L$.
  end
  Smooth mean: $\bar\theta^L_{N+l-L}=\sum_{j=1}^J\breve w(N,j)\Theta^P_{N+l-L,a_{L-l}(N,j)}$ for l in $1:L$.
  $V^m_{N+l-L,N+l-L}=\sum_j\breve w(N,j)\big(\Theta^P_{N+l-L,a_{L-l}(N,j)}-\bar\theta^L_{N+l-L}\big)\big(\Theta^P_{N+l-L,a_{L-l}(N,j)}-\bar\theta^L_{N+l-L}\big)^\top$ for l in $1:L$.
  Update: $S_m=c^{-2(m-1)}\Psi^{-1}\sum_{n=1}^N\big[\,\big(\bar\theta^L_n-\theta_{m-1}\big)\,\big]$.
  $I_m=-c^{-4(m-1)}\Psi^{-1}\left[\sum_{n=1}^N\big(V^m_{n,n}/(N+1)-c^{2(m-1)}\Psi\big)\right]\Psi^{-1}$.
  Update non-IVP parameters: $\theta_m=\theta_{m-1}+I_m^{-1}S_m$.
  Update IVPs: $\big[\theta_m\big]_i=\frac{1}{J}\sum_{j=1}^J\big[\Theta^F_{L,j}\big]_i$ for $i\in I$.
end

output: Monte Carlo maximum likelihood estimate, θ_M.

complexity: $\mathcal{O}(JM)$

Table 4.1: Summary of algorithms in iterated filtering/smoothing class

Method	Perturbation	Filter type	Complexity
IF1	random walk	filtering	$O(NJ)$
IF2	random walk	filtering	$O(NJ)$
IS2	random walk	smoothing	$O(NLJ)$
IS1	white noise	smoothing	$O(NL^2J)$
RIS1	white noise	smoothing	$O(NLJ)$

be verified using a Kalman filter. The model is given by the state space forms:

$$\begin{aligned} X_n|X_{n-1} = x_{n-1} &\sim \mathcal{N}(\alpha x_{n-1}, \sigma^\top \sigma), \\ Y_n|X_n = x_n &\sim \mathcal{N}(x_n, I_2). \end{aligned}$$

where α, σ are 2×2 matrices and I_2 is 2×2 identity matrix. Note that the optimal distribution can be derived in closed form. We simulate the data set with the following parameters:

$$\alpha = \begin{bmatrix} \alpha_1 & \alpha_2 \\ \alpha_3 & \alpha_4 \end{bmatrix} = \begin{bmatrix} 0.8 & -0.5 \\ 0.3 & 0.9 \end{bmatrix}, \ \sigma = \begin{bmatrix} 3 & 0 \\ -0.5 & 2 \end{bmatrix}.$$

We set the number of time points $N = 100$ and initial starting point $X_0 = (-3, 4)$. For this model, we try to estimate parameters α_2 and α_3. We ran our experiment with 25 iterations ($M = 25$) and with 1000 particles ($J = 1000$) on a workstation computer with a 2.7GHz processor. The initial random walk standard deviation of the perturbation should be small enough to not overshoot and big enough to make convergence not too slow. These can be assessed with convergence diagnostic plots (*King et al.*, 2015b). In practice, for both the toy example and the scientific example, we used initial perturbations of 0.02 for regular parameters and 0.2 for initial value parameters, based on values used in previous work (*Ionides et al.*, 2015). For this simple problem, we can reduce the perturbation intensity fairly quickly and still

get successful convergence; we used a geometric scheme with the standard deviation reduced by a factor of 0.95 at each iteration. We use bigger standard deviations perturbation of 0.23 and 2 respectively for both RIS1 and IS1 as the perturbation is applied only at the start of each filtering iteration. After some experimentation, we used $L = 1$ for the fixed lag smoothing.

Our approach, second order iterated smoothing (IS2) is compared against the iterated filtering (IF1) of *Ionides et al.* (2011), the perturbed Bayes map iterated filtering (IF2) of *Ionides et al.* (2015), the second-order iterated smoothing (IS1) of *Doucet et al.* (2013) and the reduced second-order iterated smoothing approach (RIS1)(see supplement A.12). As can be seen from Fig. 4.1, while MLEs of all approaches touch the true MLE at vertical broken line, the distribution of the estimated MLEs using IS2 have higher mean and smaller variance, implying higher empirical convergence rate in this case. In addition, the proposed approach gives results that are reasonably robust to the starting guesses, since we start at random values uniformly in a large rectangle. We note that, in this example RIS1 approach climbs up the likelihood surface more efficiently than IF1 approach, similar to IS1 approach but less efficiently than IF2 and IS2 approaches (Fig. 4.1). Algorithmically, IS2 has similar computational costs with the first order approaches IF1 & IF2 and with the second-order RIS1 approach while the original IS1 of *Doucet et al.* (2013) takes longer time than any other approaches because of extensive computing covariance between time points. Additional results demonstrating the performance of IS2 compared to other approaches can be found in the supplement A.13.

Average computational time of ten independent runs of each approach is given in Table 4.2. Additional overheads for fixed lag smoothing and estimating score and observed information matrix for this simple model make the computation time of IS2 and RIS1 quite large compared to computational time of IF1 and IF2. However, with complex models and large enough number of particles, these overheads become

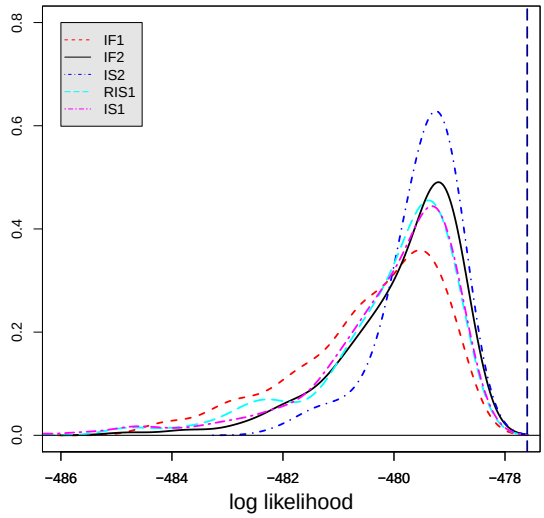

Figure 4.1: Comparison of estimators for the linear, Gaussian toy example, showing the densities of the MLEs estimated by the IF1, IF2, IS1, RIS1, and IS2 methods. The parameters α_2 and α_3 were estimated, started from 200 randomly uniform initial values over a large rectangular region $[-1, 1] \times [-1, 1]$. MLE is shown as a dashed vertical line.

Table 4.2: Computation times, in seconds, for the toy example.

	$J = 100$	$J = 1000$	$J = 10000$
IF1	4.201	14.820	140.889
IF2	4.141	14.055	125.624
IS2	6.858	27.014	281.736
IS1	10.156	47.901	466.182
RIS1	6.851	27.152	242.832

negligible and computational time of IF1, IF2, IS2 and RIS1 are similar. The relatively high $O(NL^2J)$ computational requirement of IS1 arises because this algorithm must compute computing covariances between smoothed particles at all pairs of time points up to some fixed lag. Even for $L = 1$, in this case, we require nearly double the computational time compared to IS2 and RIS1.

4.4.2 Application to a malaria transmission model

Many real world dynamic systems are highly nonlinear and partially observed. Further, some combinations of parameters may be weakly identifiable from the available data. To demonstrate the capabilities of iterated smoothing (IS2) for such situations, we consider a model for vivax malaria, a strain of malaria characterized by relapse following initial recovery from symptoms. Malaria transmission is challenging real world system to analyze, and therefore provides a rigorous performance benchmark. Mathematical modeling of malaria has been a foundation for developing malaria control strategies since the work of *Ross* (1910) and *Macdonald* (1957). We consider the $SEIH^3QS$ model of *Roy et al.* (2013) which splits up the study population of size $P(t)$ into seven classes: susceptible individuals, $S(t)$, exposure $E(t)$, infected individuals, $I(t)$, dormant classes $H_1(t)$, $H_2(t)$, $H_3(t)$ and recovered individuals, $Q(t)$. Note that since infection with malaria results in incomplete and waning immunity, the last S indicates the possibility that a recovered person can become susceptible to reinfection. Data are a sequence of monthly reported malaria morbidity, denoted by $y^*_{1:N}$. The latent force of infection $\lambda(t)$ passes through a delay stage, $\kappa(t)$, and the contributes to the current force of infection, $\mu_{SE}(t)$, with mean latency time τ_D. The state process is

$$\big(S(t), E(t), I(t), Q(t), H_1(t), H_2(t), H_3(t), \kappa(t), \mu_{SE}(t)\big),$$

where the birth rate for the S class ensures that $S(t)+E(t)+I(t)+Q(t)+\sum_i H_i(t) = P(t)$ while $P(t)$ is assumed known from the census data. The transition rates from stage H_1 to H_2, H_2 to H_3 and H_3 to Q are specified to be $3\mu_{HI}$. In this model, infected population enters dormancy via $I-\text{to}-H$ transition at rate μ_{IH}, and the treated humans join non-relapsing infected in moving to the Q class. We suppose that $\{X(t), t \geq t_0\}$ follows a stochastic differential equation in which the human stage of

the malaria pathogen lifecycle is modeled by

$$\begin{aligned}
dS/dt &= \delta P + \mathrm{d}P/dt + \mu_{IS} I + \mu_{QS} Q \\
&\quad + a\mu_{IH} I + b\mu_{EI} E - \mu_{SE}(t) S - \delta S, \\
dE/dt &= \mu_{SE}(t) S - \mu_{EI} E - \delta E, \\
dI/dt &= (1-b)\mu_{EI} E + 3\mu_{HI} H_n - (\mu_{IH} + \mu_{IS} + \mu_{IQ}) I - \delta I, \\
dH_1/dt &= (1-a)\mu_{IH} I - n\mu_{HI} H_1 - \delta H_1, \\
dH_i/dt &= 3\mu_{HI} H_{i-1} - 3\mu_{HI} H_i - \delta H_i \quad \text{for } i \in \{2, 3\}, \\
dQ/dt &= \mu_{IQ} I - \mu_{QS} Q - \delta Q,
\end{aligned}$$

where δ represent mortality rate as defined in the supplementary table (Section S-3). When it goes with a compartment class, it represents the average number of deceased people in that class per time unit. A simple representation of the malaria pathogen reproduction within the mosquito vector is given by

$$\begin{aligned}
\mathrm{d}\kappa/\mathrm{d}t &= [\lambda(t) - \kappa(t)]/\tau_D, \\
\mathrm{d}\mu_{SE}/\mathrm{d}t &= [\kappa(t) - \mu_{SE}(t)]/\tau_D.
\end{aligned}$$

The Gamma-distributed delay imposed on $\lambda(t)$ by $\kappa(t)$ and $\mu_{SE}(t)$ can also be written as

$$\mu_{SE}(t) = \int_{-\infty}^{t} \gamma(t-s)\lambda(s)\mathrm{d}s, \tag{4.6}$$

with $\gamma(s) = \frac{(2/\tau_D)^2 s^{2-1}}{(2-1)!} \exp(-2s/\tau_D)$, a gamma distribution with shape parameter 2. The latent force of infection contains a rainfall covariate $R(t)$, as described by *Roy et al.* (2013), and a Gamma white noise term,

$$\lambda(t) = \left(\frac{I + qQ}{P}\right) \times \exp\left\{\sum_{i=1}^{N_s} b_i s_i(t) + b_r R(t)\right\} \times \left[\frac{\mathrm{d}\Gamma(t)}{\mathrm{d}t}\right],$$

where q denotes a reduced infection risk from humans in the Q class and $\{s_i(t), i = 1, \ldots, N_s\}$ is a periodic cubic B-spline basis, with $N_s = 6$. We approximated the solution to this coupled system of stochastic differential equations using an Euler-Maruyama scheme (*Kloeden and Platen*, 1999) with a time step of 1/20 month.Let the number of new cases in the nth interval be $M_n = \rho \int_{t_{n-1}}^{t_n} [\mu_{EI} E(s) + 3\mu_{HI} H_3(s)] ds$ where the times of the N observations are $t_1 < t_2 < \cdots < t_N$ and the system is initialized at a time $t_0 = t_1 - 1/12$. The measurement model for Y_n given M_n is a negative binomial distribution with mean M_n and variance $M_n + M_n^2 \sigma_{\text{obs}}^2$. A table of parameter definitions and units is provided in the supplement (Section S-3).

We carried out inference for this model on data obtained from National Institutes of Malaria Research by *Roy et al.* (2013) using IF1, IF2, IS2 and RIS1. We ran our experiment on a Linux cluster, with $M = 50$ iterations and $J = 10^3$ particles. Unlike the toy example, the second order iterated smoothing with white noise (IS1) was left out as it is too computational demanding for this problem. Our approach is comparable to the recently developed algorithm IF2 (*Ionides et al.*, 2015) for this example. *Ionides et al.* (2015) compared IF2 against IF1 on a benchmark problem in epidemiological dynamics, and we use this approach to test IS2 and RIS1. In the presence of possible multi-modality, weak identifiability, and considerable Monte Carlo error of this model, we start 200 random searches in a large hyperrectangle (see supplement S-3). The random walk standard deviation is initially set to 0.1 for regular parameters while the cooling rate c is set to $0.1^{0.02} \approx 0.95$. These corresponding quantities for initial value parameter perturbations are 2 and $0.1^{0.02}$, respectively, but they are applied only at time zero. The standard deviation of independent perturbation for RIS1 is five times that of other methods. Figure 4.2 shows the distribution of the MLEs estimated by IF1, IF2, IS2 and RIS1. All distributions touch the global maximum as expected and the higher mean and smaller variance of IF2, IS2 estimation clearly demonstrate that they are considerably more effective than IF1 and IS1. Experimentation with

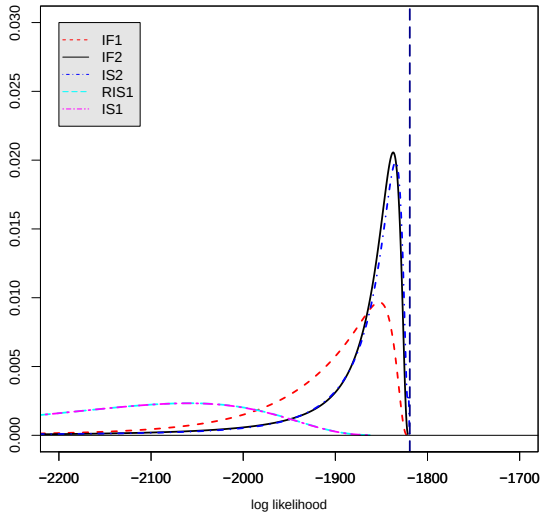

Figure 4.2: The density of the maximized log likelihood approximations estimated by IF1, IF2, IS2 and RIS1 for the malaria model when using $J = 1000$ and $M = 50$. The log likelihood at a previously computed MLE is shown as a dashed vertical line.

more extensive computation ($M = 100$ and $J = 10^4$) in Figure 4.3 suggests that the performance improvement of IS2 over IF2 occurs primarily in simpler models, such as the toy example, or during earlier stages of optimization on complex models. We have had similar experiences with other complex models (results not shown). Our interpretation is that the averaging in lines 18 and 19 involved in the parameter update rule for IS2 can be inefficient when the likelihood surface contains nonlinear ridges, whereas the IF2 algorithm does not carry out any averaging in parameter space. The computational times for IF1, IF2 and IS2 were 12.70, 12.34 and 14.56 hours respectively, confirming that the computational complexities are similar for all three methods. In this computational challenge, we see that both IS2 and IF2 offer substantial improvement over IF1.

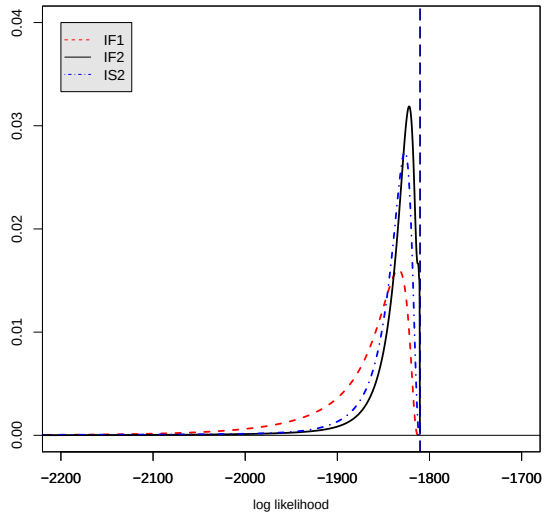

Figure 4.3: The density of the maximized log likelihood approximations estimated by IF1, IF2 and IS2 for the malaria model when using $J = 10000$ and $M = 100$

4.5 Conclusion

In this paper, we presented a novel approach for parameter estimation applicable to a general class of nonlinear, non-Gaussian POMP models. We used artificial dynamics to estimate simultaneously the parameters and the states of the latent process of the POMP model. We were also able to approximate the score vector and the observed information matrix to accelerate the convergence rate of the inference. Previous approaches for POMP models involving an estimated information matrix have either excluded the plug-and-play property or experienced heavy computational costs that made practical implementation for real world problems infeasible.

When the length of the time series goes to infinity, the parameter updating rule in our Algorithm 1 (IS2) approaches the time average of the smoothed perturbed parameters. It may be surprising that this simple updating rule has second order convergence properties, at least in some asymptotic sense.

We have shown that the iterated smoothing theory of *Doucet et al.* (2013) can be adapted to apply with random walk perturbations. In other words, we have

analyzed separately the two ways in which *Doucet et al.* (2013) modified *Ionides et al.* (2011): smoothing versus filtering, and white noise perturbations versus random walk perturbations. Our theoretical results are similar to *Doucet et al.* (2013). However, we have not been able to develop analogous results to the convergence analysis in their Section 2.4. Nevertheless, our empirical results are stronger. In principle, different simulation-based inference methods can readily be hybridized to build on the strongest features of multiple algorithms. Our results could also be applied to develop other plug-and-play methodologies which can take advantage of estimators of the derivatives of the likelihood. For example, it may be possible to use our approach to help design efficient proposal distributions for particle Markov chain Monte Carlo algorithms, taking into account the local geometry of the target distribution.

Iterated filtering methodology has been applied to study epidemiological dynamics in various situations (*King et al.*, 2008; *Laneri et al.*, 2010; *He et al.*, 2010; *Bhadra et al.*, 2011; *Camacho et al.*, 2011; *Shrestha et al.*, 2011; *Earn et al.*, 2012a; *Lavine and Rohani*, 2012; *Lavine et al.*, 2013b; *He et al.*, 2013; *Roy et al.*, 2013; *Blackwood et al.*, 2013a,b; *Shrestha et al.*, 2013; *Blake et al.*, 2014; *King et al.*, 2015a; *Laneri et al.*, 2015; *Martinez-Bakker et al.*, 2015; *Romero-Severson et al.*, 2015). However, simulation-based inference for POMP models has potential applicability for statistical inference on nonlinear POMP models arising throughout the biogical, physical and social sciences and in engineering. The theoretical and algorithmic innovations of this paper help to build a new direction for future developments on this frontier.

CHAPTER V

Bayes Map Iterated Filtering for POMP model under Reactive Control

5.1 Introduction

Iterated filtering was originally proposed by *Ionides et al.* (2006) and later theoretically developed by *Ionides et al.* (2011). Recently, *Lindström et al.* (2012) extended it to improve on numerical performance while *Doucet et al.* (2013) expanded it to include filtering/smoothing with quite attractive theoretical properties. *Ionides et al.* (2015) generalized *Lindström et al.* (2012)'s approach and combine the idea with data cloning (*Lele et al.*, 2007), developed a Bayes map iterated filtering with an entirely different theoretical approach. We revisit the approach of *Ionides et al.* (2015), using a different proof technique which relies on easily verifiable conditions, thus of more general interest. We also apply it to a more challenging data analysis, difficult to analyze before. The results confirm our previous finding that Bayes map iterated filtering is an effective plug and play approach and it is applicable in the new modeling framework.

The key contributions of this paper are three folds. Firstly, we develop and reprove Bayes map iterated filtering convergence using super martingale theory. It is simple, elegant and generalizable to more sophisticated algorithms. Secondly, we demonstrate

it as a viable method for some challenging models, including causal inference system under reactive intervention, which severely violate the conditional independence of POMP model. Finally, we show substantial improvements of the method on a toy problem and on a real world challenging problem of malaria with control compared to previous iterated filtering approaches.

The paper is organized as follows. In the next section we introduce some notations and we develop the framework of Bayes map iterated filtering. In Sections 5.3, we prove the convergence of this approximation filter to the maximum likelihood estimator (MLE) by super-martingale theorem. Section 5.4 shows a framework for inference POMP model with state depends on previous observation. We validate the proposed methodology by a toy example and a challenging inference problem of fitting a malaria transmission model with control to time series data in Section 5.5, showing substantial gains for our methods over current alternatives. We conclude in Section 5.6 with the suggesting of the future works to be extended.

5.2 Problem Definition

We consider a general latent variable model $\{X_n, Y_n,\ \theta\}$ where (X_n, Y_n) is a discrete time Markov chain with $\{X_n\}$ is also a Markov chain taking values in some measurable space $\mathscr{X}$ and Y_n is conditionally independent of the rest of the process given X_n. We suppose that X_n is unobserved while Y_n is observed and taking fixed values in observation space $\mathscr{Y}$. We also suppose that the joint density $f_{XY}(x,\ y; \theta)$ of a random variables (X, Y) depending on a parameter $\theta \in \Theta$. The marginal densities of X and Y are denoted $f_X(x; \theta)$ and $f_Y(y; \theta)$, respectively while the conditional density of the observed variable Y given the latent variable X, also known as *measurement model,* is denoted as $f_{Y|X}(y|x; \theta)$. Let μ be a probability distribution of X, f be a Markov kernel acting on $\mathscr{X}$ to itself, g be a Markov kernel acting from $\mathscr{X}$ to $\mathscr{Y}$. The likelihood function $L(\theta) = f_Y(y; \theta) = \int f_\theta(y|x) g_\theta(x) dx$. It is assumed that the like-

lihood is intractable. We are interested in computing the MLE. Set $\ell(\theta) = \log L(\theta)$, and

$$\theta_\star = \arg\max_{\theta\in\Theta} \ell(\theta).$$

The data cloning approach to computing $\theta_\star$ is to introduce the set of posterior distributions $\{\pi_n,\ n \geq 1\}$, where

$$\pi_n(\theta|y) = \frac{e^{n\ell(\theta)}\pi(\theta)}{\int e^{n\ell(\theta)}\pi(\theta)d\theta}$$

for some prior distribution π. The authors show that if $\Theta_n \sim \pi_n$, then Θ_n converges in probability to $\theta_\star$, and $\sqrt{n}(\Theta_n - \theta_\star)$ converges weakly to a Gaussian distribution. The precise statement of their result is as follows.

Assumption V.1. *$\theta \mapsto \ell(\theta)$ has a local maximum $\theta_\star$, and $\ell(\theta_\star) > 0,\ \pi(\theta_\star) > 0$.*

Assumption V.2. *π is continuous at $\theta_\star$, $\theta \mapsto \ell(\theta)$ is of class C^2 in a neighborhood of $\theta_\star$ and $-H_{\theta_\star}$ is positive definite, where $H_{\theta_\star} \stackrel{def}{=} \nabla^2\ell(\theta)|_{\theta=\theta_\star}$.*

Assumption V.3. *For any $\delta > 0$, $\gamma(\delta) < \ell(\theta_\star)$, where $\gamma(\delta) \stackrel{def}{=} \sup\{\ell(\theta),\ \|\theta - \theta_\star\| > \delta\}$.*

Theorem V.4. *Lele et al. (2007) Assume Assumptions V.1 - V.3. Set $\Sigma \stackrel{def}{=} \{-H_{\theta_\star}\}^{-1/2}$, and $\Psi_n \stackrel{def}{=} \sqrt{n}\Sigma(\Theta_n - \theta_\star)$. As $n \to \infty$, Θ_n converges in probability to $\theta_\star$, and Ψ_n converges weakly to $N(0,\ I_d)$.*

The main ingredient of the proof is the following Taylor expansion. For any $u \in \mathbb{R}^p$, there exists $t \in (0, 1)$ such that

$$\ell(\theta_\star + \Sigma u) - \ell(\theta_\star) = -\frac{1}{2}\|u\|^2 + \frac{1}{2}u'\Sigma(\nabla^2\ell(\theta_\star + t\Sigma u) - \nabla^2\ell(\theta_\star))\Sigma u.$$

For more details of the proof, see Chapter 2. As a consequence, we get the following lemma.

Lemma V.5. *Assume Assumption V.1 - V.2. Then for all* $\theta \in \Theta$*,*

$$\lim_{n\to\infty} n(\ell(\theta_\star + \frac{1}{\sqrt{n}}\Sigma\theta) - \ell(\theta_\star)) = -\|\theta\|^2/2,$$

and the convergence is uniform on compact sets.

Remark V.6. As in *Jacquier et al.* (2007), it is possible to sample (approximately) from π_n by using MCMC to sample from the augmented-data posterior distribution

$$\tilde{\pi}_n(\theta,\ x_1,\ \ldots,\ x_n|y) \propto \pi(\theta)\prod_{i=1}^{n} f_\theta(y|x_i)g_\theta(x_i).$$

One issue with the *Jacquier et al.* (2007) augmented-data posterior approach is that as n increases, and for large state space data, the posterior distribution becomes very high-dimensional and mixing of the MCMC sampler will become a problem. This is particularly true because as n increases the posterior distribution π_n becomes peeked around its various local modes and simulating from π_n (or the augmented-data posterior) becomes plagued by multi-modality issues. Typical techniques such as parallel/simulated tempering, equi-energy sampling will not be easy to implement here as these methods require to further duplicate the sampling space along several temperature.

Another limitation of the data cloning approach is that it is not clear how the technique can be used in situations where g_θ, the density of the state process is intractable. This sort of examples are common in scientific applications of state space models where the state process is a diffusion process or an ODE with stochastic coefficient.

We extend the data cloning approach by iterating sequentially the filtering of the time series data. In particular $\Theta = \mathbb{R}^d$, and we write $C(\Theta,\ \mathbb{R})$ as the space of continuous functions $\Theta \to \mathbb{R}$, equipped with the metric of convergence on compact

subsets. Let π_0 be a probability measure on Θ, $\{K_{n,i},\ 1 \le i \le n\}$ transition kernels on Θ. From $K_{n,i}$ we define the kernel (not necessarily probability kernel) $\overline{K}_{n,i}$ as

$$\overline{K}_{n,i}(u,\ dv) = K_{n,i}(u,\ dv)e^{\ell(v)}.$$

As usual we will make use of the multiplication between a measure and a kernel: $\pi K(v) = \int \pi(du)K(u,\ dv)$. We also multiply two kernels as in

$$K_1K_2(u,\ dv) = \int K_1(u,\ dy)K_2(y,\ dv).$$

We consider the sequence of probability measures $\{\tilde{\pi}_{n,i},\ 0 \le i \le n\}$, and $\{\pi_{n,i},\ 0 \le i \le$ n$\}$ defined recursively as follows. $\tilde{\pi}_{n,0} = \pi_{n,0} = \pi_0$ as given above, and for $-1 \le i \le 1$,

$$\tilde{\pi}_{n,i} = \tilde{\pi}_{n,i-1}\overline{K}_{n,i},$$

and

$$\pi_{n,i}(d\theta) = \frac{\tilde{\pi}_{n,i}(d\theta)}{c_{n,i}},$$
$$c_{n,i} = \tilde{\pi}_{n,i}(\Theta).$$

We assume that $\pi_0,\ \ell$, and $\{K_{n,i}\}$ are such that $c_n \in (0,\ \infty)$ for all $n \ge 1$.

Example V.7. *Take* $K_{n,i}(u,\ dv) = \delta_u(dv)$. *Then*

$$\begin{aligned}\overline{K}_{n,i-1}\overline{K}_{n,i}(u,\ dv) &= \int K_{n,i-1}(u,\ dy)e^{\ell(y)}K_{n,i}(y,\ dv)e^{\ell(v)} \\ &= K_{n,i}(u,\ dv)e^{\ell(u)+\ell(v)} \\ &= e^{2\ell(u)}\delta_u(dv).\end{aligned}$$

So that $\overline{K}_{n,1} \cdots \overline{K}_{n,n}(u,\ dv) = e^{n\ell(u)}\delta_u(dv)$. *It follows that*

$$\pi_n(A) = \frac{\int_A \pi_0(du)e^{n\ell(u)}}{\int \pi_0(du)e^{n\ell(u)}}.$$

Thus in this example π_n yields the data cloning sequence. Iterated filtering (*Ionides et al.*, 2011) is a simulation-based approach where the state, represented as particles, is iteratively filtered by observations to achieve maximum likelihood estimators (MLE). We revisit the Bayes map iterated filter (*Ionides et al.*, 2015), by unifying iterated filtering and data cloning.

5.3 New Theory of Bayes Map Iterated Filtering

Suppose we wish to solve the maximization problem

$$\theta_\star \stackrel{def}{=} \arg\max_{\theta \subset \mathbb{R}^{\mathrm{d}}} \ell(\theta) \tag{5.1}$$

for some smooth function ℓ, that we think of as a log-likelihood function. Our motivation comes from partially observed Markov processes, and latent variables models where ℓ can be written as $\ell(\theta) \stackrel{def}{=} \log \int q_\theta(x) f_\theta(y|x) d_{X_0}$. But for the time being ℓ is arbitrary.

Let us write P to denote the space of all probability measures on $\mathbb{R}^d$. Let K_n denote the probability kernel on $\mathbb{R}^d \times \mathcal{B}(\mathbb{R}^d)$ given by

$$K_n(\theta, d\theta) = \frac{1}{\sigma_n^d}\mathrm{K}(\sigma_n^{-1}(\vartheta - \theta))d\vartheta \tag{5.2}$$

for positive scale parameter σ_n, and where K is the density of the normal distribution on $\mathbb{R}^d$ with mean zero and covariance Λ. Without any loss of generality in the theory we assume that $\Lambda = I_d$, the $d \times d$ identity matrix.

We consider this Bayesian operator $\mathrm{B}_n : \mathcal{P} \to \mathcal{P}$ which transforms the probability

measure μ into $B_n(\mu)$ defined as

$$\mathrm{B}_n(\mu)(A) = \frac{\int \mu(d,\theta) \int K_n(\theta,\vartheta) e^{\ell(\vartheta)} 1_A(\vartheta)}{\int \mu(d\theta) \int K_n(\theta,\vartheta) e^{\ell(\vartheta)}}$$

$$= \frac{\int \mu(d\theta) \int \mathrm{K}(du) e^{\overline{\ell}(\theta+\sigma_n u)} 1_A(\theta+\sigma_n u) du}{\int \mu(d\theta) \int \mathrm{K}(u) e^{\overline{\ell}(\theta+\sigma_n u)} du} \tag{5.3}$$

by the change of variable $\vartheta = \theta + \sigma_n u$, where $\overline{\ell} = \ell - \ell(\theta_\star)$. Given some initial probability measure $\pi_0 \in P$ we consider the sequence of probability measures $\{\pi_n,\ n\ \geq 0\}$, defined recursively by

$$\pi_n = \mathrm{B}_n(\pi_{n-1}),\ n\ \geq 1.$$

The goal here is to show that as $n \to \infty,\ \pi_n$ concentrate around the solution $\theta_\star$ of the maximization problem (5.2). We assume that ℓ is strongly concave.

Assumption V.8. *The function $\ell : \mathbb{R}^d \to \mathbb{R}$ is concave, admits a unique maximum $\theta_\star \in \mathbb{R}^d$, and is twice differentiable. Furthermore, there exists $0 < \kappa \leq L < \infty$ such that for all $\theta \in \mathbb{R}^d$, the spectrum of $\nabla l^{(2)}(\theta)$ is contained in $[-L, -\kappa]$*

Assumption V.9. *Assume $\sum \sigma_n^2 < \infty$.*

By standard first-order optimality conditions, $\nabla \ell(\theta_\star) = 0$. Assumption V.8 implies that for all $\theta,\ u \in \mathbb{R}^d$

$$-\kappa \|u\|^2 \geq \nabla^{(2)} \ell(\theta) \cdot (u,\ u) \geq -L\|u\|^2 \tag{5.4}$$

We obtain the following lemma.

Lemma V.10. *Assume Assumption V.8, the following holds,*

$$\pi_n(h) = \frac{\int \pi_{n-1}(d\theta) e^{\ell(\theta)} h(\theta)}{\int \pi_{n-1}(d\theta) e^{\ell(\theta)}} + \sigma_n^2 \epsilon_{n,3}$$

for some $\epsilon_{n,3} < \infty$.

Proof. We have $\pi_n(h) \stackrel{def}{=} \int h(\theta)\pi_n(d\theta)$. It follows from (5.3) that,

$$\pi_{n+1}(h) = \frac{\int \pi_n(d\theta) \int \mathrm{K}(u) e^{\bar{\ell}(\theta+\sigma_{n+1}u)} h(\theta+\sigma_{n+1}u)dz}{U_{n+1}}$$

where

$$U_n \stackrel{def}{=} \int \pi_{n-1}(d\theta) \int \mathrm{K}(u) e^{\bar{\ell}(\theta+\sigma_n u)} du \tag{5.5}$$

For integer $\sigma > 0$, we introduce the operator T_σ that transforms a function $f : \mathbb{R}^d \to \mathbb{R}$ into

$$\mathrm{T}_\sigma f(\theta) = \int K_\sigma(\theta,\ d\vartheta) e^{\bar{\ell}(\vartheta)} f(\vartheta) = \int \mathrm{K}(u) e^{\bar{\ell}(\theta+\sigma u)} f(\theta+\sigma u) du.$$

When $\sigma = \sigma_n$, we simply write $\mathrm{T_n}$ instead T_{σ_n}. By iterating the operator B_n we see that

$$\pi_n(h) = \frac{\int \pi_0(d\theta) \mathrm{T}_1 \circ \ldots \circ \mathrm{T}_n h(\theta)}{\int \pi_0(d\theta) \mathrm{T}_1 \circ \ldots \circ \mathrm{T}_n 1(\theta)}.$$

Let $c_n \stackrel{\triangle}{=} \int \pi_{n-1}(d\theta) \int \mathrm{K}(u) e^{\ell(\theta+\sigma_n u)} du$ denotes the normalizing constant of π_n. We have

$$\pi_n(h) = \frac{1}{c_n} \int \pi_{n-1}(d\theta) \int \mathrm{K}(u) e^{\ell(\theta+\sigma_n u)} h(\theta+\sigma_n u) du \tag{5.6}$$

By a Taylor expansion of $e^{\ell(\theta)}$, we write

$$e^{\ell(\theta+\sigma_n u)} = e^{\ell(\theta)} + \sigma_n e^{\ell(\theta)} \langle \nabla\ell(\theta),\ u\rangle + \frac{\sigma_n^2}{2} e^{\ell(\theta)} e^{\ell(\overline{\theta})-\ell(\theta)} u' \{\nabla\ell(\overline{\theta})\nabla\ell(\overline{\theta})' + \nabla^{(2)}\ell(\overline{\theta})\} u,$$

for some $\overline{\theta}$ on the segment between θ and $\theta + \sigma_n u$. We set

$$H_n(\theta,\ u) \stackrel{\triangle}{=} e^{\ell(\overline{\theta})-\ell(\theta)} \{\nabla\ell(\overline{\theta})\nabla\ell(\overline{\theta})' + \nabla^{(2)}\ell(\overline{\theta})\}.$$

From Assumption V.8, we get that using this expansion and the fact that u is

bounded and $\int u\mathrm{K}(u)du = 0$, so

$$c_n = (1 + \sigma_n^2 \epsilon_{n,1}) \int \pi_{n-1}(d\theta) e^{\ell(\theta)} \tag{5.7}$$

for some bounded sequence $\{\epsilon_{n,1},\ n \geq 1\}$. Similar calculation yields

$$\int \mathrm{K}(u) e^{\ell(\theta+\sigma_n u)} h(\theta + \sigma_n u) du = e^{\ell(\theta)} \int h(\theta + \sigma_n u)\mathrm{K}(u)du$$

$$+\sigma_n \int \langle h(\theta + \sigma_n u)u,\ \nabla \ell(\theta)\rangle \mathrm{K}(u)du + \sigma_n^2 R_{n,1}(\theta)).$$

where $\{|R_{n,1}|_\infty,\ n \geq 1\}$ is a bounded sequence. By further expanding h, and using the fact that $\int u\mathrm{K}(u)du = 0$, and Assumption V.8

$$\left| \int \langle h(\theta + \sigma_n u)u,\ \nabla \ell(\theta)\rangle\, \mathrm{K}(u)du \right| = 0,$$

and expanding

$$e^{\ell(\theta)} \int h(\theta + \sigma_n u)\mathrm{K}(u)du$$

$$= e^{\ell(\theta)} \left(\int h(\theta)\mathrm{K}(u)du + \sigma_n \int \nabla h(\theta) u K(u)du + \sigma_n^2 \nabla^{(2)} h(\theta) R_{n,2}(\theta) \right)$$

where $\{|R_{n,2}|_\infty,\ n \geq 1\}$ is a bounded sequence and $\sup_{\theta\in\Theta} \|\nabla^{(2)} h(\theta)\| < \infty$, also $\int u\mathrm{K}(u)du = 0$. It follows that

$$\int \pi_{n-1}(d\theta) \int \mathrm{K}(u) e^{\ell(\theta+\sigma_n u)} h(\theta + \sigma_n u) du$$

$$= \int \pi_{n-1}(d\theta) e^{\ell(\theta)} h(\theta) + \sigma_n^2 \epsilon_{n,2} \int \pi_{n-1}(d\theta) e^{\ell(\theta)}$$

where $\{\epsilon_{n,3},\ n \geq 1\}$ is a bounded sequence. We combine (5.6) and (5.7) to write $\pi_n(h) = \left(\int \pi_{n-1}(d\theta) e^{\ell(\theta)} h(\theta) + \sigma_n^2 \epsilon_{n,2} \int \pi_{n-1}(d\theta) e^{\ell(\theta)} \right) / c_n$. We conclude as claimed that

$$\pi_n(h) = \left(\frac{\int \pi_{n-1}(d\theta) e^{\ell(\theta)} h(\theta)}{\int \pi_{n-1}(d\theta) e^{\ell(\theta)}} + \sigma_n^2 \epsilon_{n,2} \right) / \left(1 + \sigma_n^2 \epsilon_{n,1}\right)$$

Since $\pi_n(h)$ is bounded, $\dfrac{\int \pi_{n-1}(d\theta)e^{\ell(\theta)}h(\theta)}{\int \pi_{n-1}(d\theta)e^{\ell(\theta)}}$ is bound

$$\pi_n(h) = \left(\frac{\int \pi_{n-1}(d\theta)e^{\ell(\theta)}h(\theta)}{\int \pi_{n-1}(d\theta)e^{\ell(\theta)}} + \sigma_n^2\epsilon_{n,2}\right)\left(1 - \sigma_n^2\epsilon_{n,1}\right)$$

$$= \frac{\int \pi_{n-1}(d\theta)e^{\ell(\theta)}h(\theta)}{\int \pi_{n-1}(d\theta)e^{\ell(\theta)}} + \sigma_n^2\epsilon_{n,3}$$

for some bounded sequence $\{\epsilon_{n,3},\ n \geq 1\}$ □

Lemma V.11. *Denote*

$$\pi_N^1(h) = \frac{\int \pi_N(d\theta)e^{\ell(\theta)}h(\theta)}{\int \pi_N(d\theta)e^{\ell(\theta)}},$$

and for an integer $m > 1$,

$$\pi_N^m(h) = \frac{\int \pi_N^{m-1}(d\theta)e^{\ell(\theta)}h(\theta)}{\int \pi_N^{m-1}(d\theta)e^{\ell(\theta)}},$$

then

$$\pi_N^k(h) = \pi_{N+k}(h) - \sum_{i=N+1}^{N+k} \epsilon_{i,3}\sigma_i^2$$

for some bounded sequence $\{\epsilon_{i,3},\ i \geq 1\}$.

Proof. We prove by induction on k. For $k = 1$, it is true from Lemma V.10 that

$$\pi_N^1(h) = \pi_{N+1}(h) - \epsilon_{N+1}\sigma_{N+1}^2$$

for some bounded ϵ_{N+1}. Assume it is true for all k from 1 to m, we prove that it is also true for $k = m + 1$. By induction assumption, we have

$$\pi_N^m(h) = \pi_{N+m}(h) - \sum_{i=N+1}^{N+m} \epsilon_{i,3}\sigma_i^2.$$

From definition,

$$\pi_N^{m+1}(h) = \frac{\int \pi_N^m(d\theta)e^{\ell(\theta)}h(\theta)}{\int \pi_N^m(d\theta)e^{\ell(\theta)}} = \frac{\int \left(\pi_{N+m} - \sum_{i=N+1}^{N+m} \epsilon_{i,3}\sigma_i^2\right)(d\theta)e^{\ell(\theta)}h(\theta)}{\int \left(\pi_{N+m} - \sum_{i=N+1}^{N+k} \epsilon_{i,3}\sigma_i^2\right)(d\theta)e^{\ell(\theta)}}.$$

By similar arguments as in proof of Lemma V.10,

$$\pi_N^{m+1}(h) = \frac{\int \left(\pi_{N+m} - \sum_{i=N+1}^{N+k} \epsilon_{i,3}\sigma_i^2\right)(d\theta)e^{\ell(\theta)}h(\theta)}{\int \left(\pi_{N+m} - \sum_{i=N+1}^{N+k} \epsilon_{i,3}\sigma_i^2\right)(d\theta)e^{\ell(\theta)}}$$

$$= \frac{\int (\pi_{N+m})(d\theta)e^{\ell(\theta)}h(\theta)}{\int (\pi_{N+m})(d\theta)e^{\ell(\theta)}} - \sum_{i=N+1}^{N+m} \epsilon_{i,4}\sigma_i^2$$

for some bounded sequence $\{\epsilon_{i,4},\ i \geq 1\}$. Making use of Lemma V.10 again we have

$$\pi_N^{m+1}(h) = \pi_{N+m+1}(h) - \epsilon_{N+m+1}\sigma_{N+m+1}^2 - \sum_{i=N+1}^{N+m} \epsilon_{i,4}\sigma_i^2$$

$$= \pi_{N+m+1}(h) - \sum_{i=N+1}^{N+m+1} \epsilon_{i,4}\sigma_i^2$$

from which we can conclude that it is true for every k. □

Lemma V.12. *Theorem 1 of (Robbins and Siegmund, 1985). Let* $(\Omega, \mathrm{F}, \mathrm{P})$ *be a probability space and* $\mathrm{F}_1 \subset \mathrm{F}_2 \subset ...$ *a sequence of* $\mathrm{sub}-\sigma$*-algebras of* F*. For each* $n = 1, 2, ...,$ *let* $z_n, \beta_n, \xi_\mathrm{n}$ *and* ζ_n *be non-negative* $\mathrm{F_n}$*-measurable random variable such that*

$$\mathrm{E}(\mathrm{z_{n+1}}|\mathrm{F_n}) \leq \mathrm{z_n}(1 + \beta_\mathrm{n}) + \xi_\mathrm{n} - \zeta_\mathrm{n} \tag{5.8}$$

$\lim_{\mathrm{n}\to\infty} \mathrm{z_n}$ *exists and is finite and* $\sum_1^\infty \zeta_n < \infty$ *a.s. on*

$$\left\{\sum_1^\infty \beta_\mathrm{n} < \infty,\ \sum_1^\infty \xi_\mathrm{n} < \infty\right\}.$$

Theorem V.13. *Suppose that Assumptions V.8 and V.9 hold. Suppose also that π_0 is absolutely continuous w.r.t. the Lebesgue measure and $\breve{\Theta}_n \sim \pi_n$. Then $\breve{\Theta}_n$ converges a.s to θ_* as $n \to \infty$.*

Proof. By Lemma V.10,

$$\pi_n(h) = \frac{\int \pi_{n-1}(d\theta) e^{\ell(\theta)} h(\theta)}{\int \pi_{n-1}(d\theta) e^{\ell(\theta)}} + \sigma_n^2 \epsilon_n$$

$$= \pi_{n-1}(h) + \{\frac{\int \pi_{n-1}(d\theta) e^{\bar{\ell}(\theta)} h(\theta)}{\int \pi_{n-1}(d\theta) e^{\bar{\ell}(\theta)}} - \pi_{n-1}(h)\} + \sigma_n^2 \epsilon_n$$

where $\bar{\ell}(\theta) = \ell(\theta) - \ell(\theta_\star)$, and $\{\epsilon_n\}$ is a bounded sequence. For $t \in [0,1]$, we define $\pi_{n,t}(d\theta) = \frac{1}{c_{n,t}} \pi_n(d\theta) e^{t\bar{\ell}(\theta)}$, where $c_{n,t} = \int \pi_n(d\theta) e^{t\bar{\ell}(\theta)}$. We have

$$\frac{\int \pi_{n-1}(d\theta) e^{\bar{\ell}(\theta)} h(\theta)}{\int \pi_{n-1}(d\theta) e^{\bar{\ell}(\theta)}} - \pi_{n-1}(h) = \int \pi_{n-1,1}(d\theta) h(\theta) - \int \pi_{n-1,0}(d\theta) h(\theta)$$

$$= \int_0^1 \{\frac{d}{dt} \int \pi_{n-1,t}(d\theta) h(\theta)\} dt$$

where $h(\theta) = \ell(\theta_\star) - \ell(\theta)$

$$= \int_0^1 \left\{ \int \frac{d}{dt} \left[\frac{1}{c_{n-1,t}} \pi_{n-1}(d\theta) e^{t\bar{\ell}(\theta)} \right] h(\theta) \right\} dt$$

$$= \underbrace{\int_0^1 \left\{ \int \left[\frac{d}{dt} \frac{1}{c_{n-1,t}} \right] \pi_{n-1}(d\theta) e^{t\bar{\ell}(\theta)} h(\theta) \right\} dt}_{A} + \underbrace{\int_0^1 \left\{ \int \frac{1}{c_{n-1,t}} \pi_{n-1}(d\theta) \bar{\ell}(\theta) e^{t\bar{\ell}(\theta)} h(\theta) \right\} dt}_{B}$$

Since

$$\frac{d}{dt} \frac{1}{c_{n-1,t}} = \frac{-\frac{d}{dt} C_{n-1,t}}{c_{n-1,t}^2} = \frac{-\int \bar{\ell}\left(\tilde{\theta}\right) \pi_{n-1}(d\tilde{\theta}) e^{t\bar{\ell}(\tilde{\theta})}}{c_{n-1,t}^2}$$

we have

$$A = \int_0^1 - \left\{ \int \frac{\bar{\ell}\left(\tilde{\theta}\right) \pi_{n-1}(d\tilde{\theta}) e^{t\bar{\ell}(\tilde{\theta})}}{c_{n-1,t}} \int \frac{\pi_{n-1}(d\theta) e^{t\bar{\ell}(\theta)} h\left(\theta\right)}{c_{n-1,t}} \right\} dt$$

$$= \int_0^1 -E\left[\bar{\ell}\left(\breve{\Theta}\right)\right] E\left[h\left(\breve{\Theta}\right)\right] dt$$

and

$$B = \int_0^1 E\left[\bar{\ell}\left(\breve{\Theta}\right) h\left(\breve{\Theta}\right)\right] dt$$

So

$$\int_0^1 \{\frac{d}{dt} \int \pi_{n-1,t}(d\theta) h(\theta)\} dt$$

$$= \int_0^1 \mathrm{Cov}_{n-1,t}\ (\overline{\ell}(\breve{\Theta}),\ h(\breve{\Theta}))dt$$

where the covariance and variance are w.r.t to the distribution $\pi_{n-1,t}$. First, we choose $h(u) = \ell(\theta_\star) - \ell(u),\ u \in \Theta$, then we have

$$\pi_n(h) = \pi_{n-1}(h) - \int_0^1 \mathrm{Var}_{n-1,t}\ (\overline{\ell}(\breve{\Theta}))dt + \sigma_n^2 \epsilon_n$$

We then make use of the super-martingale theorem (Lemma V.12) to conclude that $\pi_n(h)$ converges to a finite limit $\pi_u(h)$ and $\sum_n \int_0^1 \mathrm{Var}_{n-1,t}(\overline{\ell}(\breve{\Theta}))dt < \infty$. Given ϵ, there exists an N such that $|\pi_N(h) - \pi_u(h)| < \epsilon$. By Lemma V.10,

$$\pi_{N+1}(h) = \frac{\int \pi_N(d\theta) e^{\ell(\theta)} h(\theta)}{\int \pi_N(d\theta) e^{\ell(\theta)}} + \sigma_{N+1}^2 \epsilon_{N+1}$$

so

$$\pi_N^m(h) = \frac{\int \pi_N(d\theta) e^{n\ell(\theta)} h(\theta)}{\int \pi_N(d\theta) e^{n\ell(\theta)}} = \pi_{N+m}(h) + \sum_{i=N+1}^{N+m} \epsilon_{i,4} \sigma_i^2$$

Since $\pi_N^m(h)$ is m iteration of data cloning of $\pi_N(h)$, by Theorem II.4, $\pi_N^m(h)$ converges to $\delta_{\theta_\star}$ as $m \to \infty$. As above $\sum_{i=N+1}^{N+m} \epsilon_{i,4}\sigma_i^2$ and ϵ can be made arbitrarily small, hence $\pi_u(h)$ converges to $h(\theta_\star) = 0$. Thus, we can conclude that $\mathrm{E}(\bar{\ell}(\breve{\Theta}_n)) \to 0$ or $\mathrm{E}(\ell(\theta_\star) - \ell(\breve{\Theta}_n)) \to 0$. Since $\ell(\theta_\star) - \ell(\breve{\Theta}_n) \geq 0$, we have $\ell(\theta_\star) - \ell(\breve{\Theta}_n) \to 0$. By the uniqueness and differentiable of log-likelihood, $\breve{\Theta}_n \to \theta_\star$ □

For completeness, the sequential Monte Carlo filter can be arbitrarily approximated to the exact filter by choosing sufficiently large number of particles (*Ionides et al.*, 2011). It can be seen that IF2 Monte Carlo approximation is exactly as a regular bootstrap particle filter approximation at any time point except the last step. In the last step, estimated parameter is kept fixed while state will be re-sampled at the initial starting time point which indeed can still be applied the regular Monte Carlo approximation.

5.4 Latent Model with State dependent on Observation

Suppose a POMP model with latent variable $X_n = (U_n, V_n)$ and observed variable Y_n having conditional density $p_{Y_n|V_n}(y_n|v_n)$ depending only on V_n. The proper weight (*Liu and Chen*, 1998) for an SMC proposal density $q_n(x_n|x_{n-1})$ is

$$w_n(x_n|x_{n-1}) = \frac{p_{Y_n|X_n}(y_n^*|x_n)p_{X_n|X_{n-1}}(x_n|x_{n-1})}{q_n(x_n|x_{n-1})}.$$

Consider the proposal

$$q_n(u_n, v_n|x_{n-1}) = p_{U_n|X_{n-1}}(u_n|x_{n-1})p_n(v_n).$$

This is partially plug-and-play, in the sense that the U_n part of the proposal is drawn from a simulator of the dynamic system. Computing the weights, we also see that the transition density for the U_n component cancels out and does not have to be

computed, i.e.,

$$w_n(x_n|x_{n-1}) = \frac{p_{Y_n|V_n}(y_n^*|v_n)p_{U_n|X_{n-1}}(u_n|x_{n-1})p_{V_n|U_n,X_{n-1}}(v_n|u_n,x_{n-1})}{p_{U_n|X_{n-1}}(u_n|x_{n-1})p_n(v_n)}$$
$$= \frac{p_{Y_n|V_n}(y_n^*|v_n)p_{V_n|U_n,X_{n-1}}(v_n|u_n,x_{n-1})}{p_n(v_n)}.$$

Now consider the case where the V_n component of the state space is perfectly observed, i.e., $Y_n = V_n$. In this case,

$$p_{Y_n|V_n}(y_n|v_n) = \delta(y_n - v_n),$$

interpreted as a point mass at v_n in the discrete case and a singular density at v_n in the continuous case. We can choose $p_n(v_n)$ to depend on the data, and a natural choice is

$$p_n(v_n) = \delta(y_n^* - v_n),$$

for which the proper weight is

$$w_n(x_n|x_{n-1}) = p_{Y_n|U_n,X_{n-1}}(y_n^*|u_n,x_{n-1}).$$

Now, assume that we can decompose $U_n = (G_n, H_n)$ and let $V_n = Y_n$. A complication is that transitions of the latent variables from (G_n, H_n) to (G_{n+1}, H_{n+1}) depends on the observed variable Y_n. Formally, we use the state variable $X_n = (G_n, H_n, Y_n)$ and model the measurement process as a perfect observation of the Y_n component of the state space. To define a recursive SMC filter, we write filter particle j at time $n-1$ as

$$X_{n-1,j}^F = (G_{n-1,j}^F, H_{n-1,j}^F, y_{n-1}^*).$$

Now we can construct prediction particles at time n,

$$(G^P_{n,j}, H^P_{n,j}) \sim p_{G_n,H_n|G_{n-1},H_{n-1},Y_{n-1}}(p_n|G^F_{n-1,j}, H^F_{n-1,j}, y^*_{n-1})$$

with corresponding weight

$$w_{n,j} = p_{Y_n|G_n,H_n}(y^*_n|G^P_{n,j}, H^P_{n,j}).$$

Re-sampling with probability proportional to these weights gives an SMC representation of the filtering distribution at time n. A derivation of this is given as proper weighting for a partially plug-and-play algorithm with a perfectly observed state space component. We will see this more clearly in the following examples.

5.5 Experiments

5.5.1 Toy experiment

Feedback from the observation process to the state process is not unusual situation, such as when a system is subject to a control mechanism constructed as a function of the observations. It is a fairly well established empirical observation that negative shocks to the index are associated with a subsequent increase in volatility. Here, we formally define leverage, R_n on day n as the correlation between index return on day $n-1$ and the increase in the log volatility from day $n-1$ to day n. We present a toy example of (*Bretó*, 2014), which models R_n as a random walk on a transformed scale,

$$R_n = \frac{\exp\{2G_n\} - 1}{\exp\{2G_n\} + 1}.$$

Following the notation and model representation in equation (4) of (*Bretó*, 2014), we have

$$Y_n = \exp\{H_n/2\}\epsilon_n, \tag{5.9}$$

$$H_n = \mu_h(1-\phi) + \phi H_{n-1} + \beta_{n-1} R_n \exp\{-H_{n-1}/2\} + \omega_n, \tag{5.10}$$

$$G_n = G_{n-1} + \nu_n, \tag{5.11}$$

where $\beta_n = Y_n \sigma_\eta \sqrt{1-\phi^2}$, $\{\epsilon_n\}$ is an i.i.d. $N(0,1)$ sequence, $\{\nu_n\}$ is an i.i.d. $N(0,\sigma_\nu^2)$ sequence, and $\{\omega_n\}$ is an i.i.d. $N(0,\sigma_\omega^2)$ sequence. We write $\{y_n^*, n = 1,\dots,N\}$ for the data. Note that the data are also placed in a covariate slot, which is a devise to allow the state process evolution to depend on the data, which it cannot do in a standard pomp. However, simulating from the model is convenient for investigating a fitted model. We check that if we can indeed filter and re-estimate parameters successfully from simulated data. Here, we use the number of particles Np=5000 (i.e., sequential Monte Carlo sample size), and the number of iterations Nmif = 100. We get stable results with an error in the likelihood of order 1 log unit for this example. In 6.5 seconds, we obtain an unbiased likelihood estimate of -3658.76 with a Monte Carlo standard error of 0.1. Notice that we could test the numerical performance of an iterated filtering likelihood maximization algorithm on simulated data. For our volatility model, the confidence interval computed from profile likelihood and probability coverage are stable, indicating that our approach are effective for this toy model with control. We will investigate a real-world problem in the next section.

5.5.2 Malaria with control

The two most fatal and deadliest malaria parasites are *P. falciparum* and *P. vivax*. Understanding their life-cycles plays a critical role in the containment and eradication of such diseases. A key aspect of this insight is to develop a quantitative representation

of malaria transmission model to capture the most essential factors at the population level. Conventionally, most of the existing models do not take into account the human control intervention, which can result in missing potential severe epidemics. A few studies (*Artzy-Randrup et al.*, 2010; *Baeza et al.*, 2014) explicitly consider a control intervention as a factor, increasing the mortality of the adult mosquito population. However, models in these studies are fully coupled mosquito-malaria, which traces the dynamics of the mosquito population (*Alonso et al.*, 2011). By including parameters from entomological literature, these models are rather complicated. However, they often do not fit the data well because of unavailable entomological data. Therefore, our goal here is to represent a simpler mathematical model, which accounts for the consequences of intervention mechanism based on the available data. Due to lacking mosquitoes data, only the most relevant covariates to human disease are included. This means that mosquito dynamics are represented implicitly through the force of infection of humans.

Adopting the well-established practice, we divide the studied population of size $P(t)$ into the following distinctive classes: susceptible to infection, $S(t)$, exposure $E(t)$, infected and gametocytemic individuals, $I(t)$ and recovered individuals, $Q(t)$. To allow flexibility in representing losing immunity, both infected and recovered individuals could be reinfected. Malaria mortality during each month is tracked with a variable $y(t)$. Complete parameter definitions and units are provided in supplement S-1. Assume $P(t)$ is known from the census data and birth rate for the S class satisfies $S(t) + E(t) + I(t) + Q(t) = P(t)$, the state process $X(t) =$

$\big(S(t), E(t), I(t), Q(t), \kappa_1(t), \mu_{SE}(t)\big)$, follows a stochastic differential equation,

$$human \rightarrow \begin{cases} dS/dt = (\delta P + \mathrm{d}P/dt) + \mu_{IS} I + \mu_{QS} Q - \mu_{SE}(t) S - \delta S, \\ dE/dt = \mu_{SE}(t) S - \mu_{EI} E - \delta E, \\ dI/dt = \mu_{EI} E - (\mu_{IS} + \mu_{IQ}) I - \delta I, \\ dQ/dt = \mu_{IQ} I - \mu_{QS} Q - \delta Q, \end{cases} \tag{5.12}$$

where $\mu_{SE}(t)$ are defined as current rate of transmission.

According to *Anderson and May* (1991); *Keeling and Rohani* (2009), the above equations are also known as large population limit of homogeneous individual-level interactions. In addition, to identify different levels and roles of immunity, we follow from the pioneer work of *Dietz et al.* (1974) on malaria model with several possible immunity representations.

Since λ determines the number of infected vectors, but does not affect transmission until the parasite finishes its development (sporogony) within the surviving mosquitoes, it is denoted as latent force of infection. A novel feature of this framework, compared to previously epidemiological models fitted to population-level time series data, is the implicit representation of the control intervention in the vector dynamics. Specifically, mosquitoes vector is modeled implicitly through these current and latent force of infection because of the unavailability of entologmological data and the parsimony of the model. The implicit model assists in avoiding unnecessarily complicated model components such as mosquito abundance, survival and behavior. This intervention is modeled by rising mortality of the adult mosquito population. Specifically, let mosquito adult mortality be δ_M and in the model of the reactive policy, we incorporate the effect of IRS intervention as an additional mortality rate δ_c. The total mortality now becomes $\delta_M = \delta_0 + \delta_C$ where δ_0 is the the natural mortality, which will shorten the average lifespan of an adult mosquito. Because of reactive control policy, we consider δ_c as a function of cases from the previous seasons given

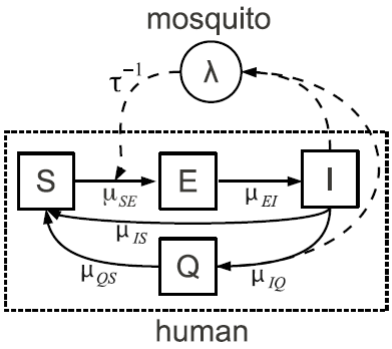

Figure 5.1: A compartment model of malaria transmission.

by $I_P = \int_{\tau_C} E_{(t)}\mu_{EI}dt$. This I_P corresponds to the total number of cases added over a given interval of time τ_C in the past emulates the API policy as described in (*Baeza et al.*, 2014). Through generating a distributed time delay between the current rate of transmission $\mu_{SE}(t)$ of a susceptible human at time t, the transmission rate from infected to susceptible humans can also be represented implicitly. The force of infection can then be derived from levels of infection in the human population at all previous times as

$$\lambda(t) = \beta(t)[\frac{I + qQ}{P}],$$

with $\beta(t) = \overline{\beta}\times$ seasonality $\times$ rainfall $\times$ control $\times$ noise (S3). Control term $C(t)$ reduces the latent force of infection λ as follows:

$$\lambda(t) = \overline{\beta}[\frac{I + qQ}{P}] \times \exp[\sum_{i=1}^{n_s} b_i s_i(t) + b_r Z(t) - b_c C(t)] \times [\frac{\mathrm{d}\Gamma(t)}{\mathrm{d}t}] \tag{5.13}$$

where q denotes a reduced infection risk from humans in the Q class. The modeled flows between these classes is shown in Figure 5.1 where the dynamic equations is defined above. The transmission term $\beta(t)$ includes four exogenous forcing namely control intervention, seasonality, rainfall, environmental noise. First, control intervention decreases the transmission rate by increasing adult mosquitoes morbidity rate as described above. Second, seasonality is modeled non-parametrically through the

coefficients $\{\beta_i\}$ to account for yearly periodic forces. These correspond of a periodic cubic B-spline basis $\{s_i(t),\ i = 1,\ \ldots,\ n_s\}$ constructed using n_s evenly spaced knots. By fitting the model, the shape and timing of this component is determined from the data. In this study, $n_s = 6$ because this gives the best fit whereas larger n_s with small improvement in fit did not give statistical support for the additional model complexity. Third, rainfall forcing is represented by $b_r Z(t)$ while time-varying covariates enter via the row vector Z_t with coefficients in a column vector b_r. Since the dimensional constant $\overline{\beta}$ gives $\mu_{S_1E}(t)$ units of t^{-1}, we set $\overline{\beta} = 1\ \mathrm{yr}^{-1}$. Fourth, environmental noise, variations in vector abundance and behavior is considered the main stochasticity for this system, which is modeled by gamma process $\Gamma(t)$ representing integrated noise with intensity σ^2. Integrate the force of infection over time, weighted by a probability of the delaying from parasite development, can give the inoculation rate

$$\mu_{SE}(t) = \int_{-\infty}^{t} \gamma(t-s)\lambda(s)\mathrm{d}s \tag{5.14}$$

where $\gamma(s) = \dfrac{(k/\tau)^k s^{k-1}}{(k-1)!}\exp(-ks/\tau)$. This integral equation follows classic Ross-Macdonald model (*Macdonald*, 1957; *Aron and May*, 1982) to combines *Plasmodium* development and mosquito survival. For the delay probability function $\gamma(s)$, Gamma distribution is chosen over exponential distribution as it allows a flexible shape for a characteristic time scale of parasite development. To facilitate the numerical solution allowing a differential representation (*Lloyd*, 2001), the gamma-distributed latency, with mean τ and variance τ^2/k, was chosen. We define $\lambda_1(t),\ \ldots,\ \lambda_k(t)$ to satisfy

$$d\kappa_1/dt = (\lambda d\Gamma/dt - \lambda_1)k\tau^{-1} \tag{5.15}$$

$$d\mu_{SE}/dt = (\lambda_{i-1} - \lambda_i)k\tau^{-1} \text{ for } i = 2,\ \ldots,\ k \tag{5.16}$$

where $\kappa_1,\ \mu_{SE}(\equiv \kappa_2)$ approximate Gamma-distributed in $\lambda(t)$.

Setting $\mu_{SE}(t) = \lambda_k(t)$, we get (5.15) - (5.16) is equivalent to (5.14). A stationary independent increments process is defined as $\Gamma(t) - \Gamma(s) \sim$ Gamma $([t-s]/\sigma^2,\ \sigma^2)$ where Gamma $(a,\ b)$ is the gamma distribution with mean ab and variance ab^2. Thus, one can describe the process $d\Gamma/dt$ as multiplicative gamma noise (*Bretó et al.*, 2009) even if this jump process is not differentiable generally. As noted, the rationale behind choosing Gamma noise over Gaussian noise is to enforce the positivity of $\mu_{S_1E}(t)$ and all the state variables in (5.12) - (5.14). *Artzner and Heath* (1997) showed that a Lévy jump process $\Gamma(t)$ with a nonnegative distribution is equivalent to $d\Gamma(t)/dt$ represents nonnegative white noise. The gamma process was selected in this case due to its relatively simple and well-studied nonnegative Lévy process. As such, all the stochastic differential equations (5.12) - (5.14) and (5.15) - (5.16) are driven by Lévy noise. They are numerically solved via the Euler method (*Protter and Talay*, 1997; *Jacod*, 2004) with a time-step of one day. Even though it is possible that Euler method can generate numerical approximations violating the positive constraint, it could be noted that it is empirically negligible. At the first glance, our model may appear to be an over-simplification, since it left out many of the biological aspects developed in previous models as well as spatial, socioeconomic, age-related and genetic inhomogeneities among the population. However, as stated in *Bhadra et al.* (2011), models based on homogeneous populations are often sufficient to capture the major features of disease transmission dynamics. Indeed, there are already publications in the malaria transmission literature employing this simplification, excluding modeling control intervention. Unlike them, we target the model with additional control parameter and show that it can help understanding the real mechanism. The measurement model identifies the relationship between data and the dynamic process. Let the number of new cases in the nth interval be $M_n = \rho \int_{t_{n-1}}^{t_n} [\mu_{EI} E(s)] ds$ where the time of the N observations is $\{t_n,\ n = 1,\ \ldots,\ N\}$ and assume it is initialized at some time $t_0 < t_1$. We denote Y_n, the reported number of confirmed cases, condition

on M_n we have $Y_n|M_n \sim \text{Negbin}(M_n,\ \sigma^2_{\text{obs}})$, where $\text{Negbin}(\alpha,\ \beta)$ is the negative binomial distribution with mean α and variance $\alpha + \alpha^2\beta$. To account for under-reporting or over-reporting issues, the negative binomial distribution is selected because it provides a model for count data that includes the possibility of over-dispersion relative to Poisson or binomial models. Since only a small fraction of malaria cases are treated in the public clinics that contribute to district statistics (*Kumar et al.*, 2007). This vague interpretation of ρ is due largely to inexactness in being classified as a case. Therefore in our model, only a small fraction of the people moving from class E to class I are detected by the surveillance system, that is $\rho \ll 1$. Let Z_t denote the thresholded rainfall integrated over a time interval $[t\ -u,\ t]$ and let $\breve{Z}_t$ denote the maximum interpolated continuous-time cubic spline $r(t)$ and 0, where the accumulated rainfall data are $\{r_n,\ n = 1,\ \ldots,\ N\}$ at times $t_1,\ \ldots,\ t_N$. The covariate was standardized by setting

$$Z_t = (\breve{Z}_t - \overline{Z})/\sigma_Z,$$

where

$$\overline{Z} = (t_N - t_0)^{-1} \int_{t_0}^{t_N} \breve{Z}_s ds,$$

$$\sigma^2_Z = (t_N - t_0)^{-1} \int_{t_0}^{t_N} (Z_s - \overline{Z})^2 ds.$$

The rate of super-infection is proportional to the rate of infection $\mu_{S_2I_2} = c\mu_{S_1E}$ with some constant of proportionality $0 \leq c \leq 1$. Thus, the threshold and lag effects of the biological systems are then represented parsimoniously and the tasks of quantifying the dynamic role of environmental covariates become manageable using the plug-and-play statistical methodology.

5.5.3 Data analysis

We choose an arid region of Northwest India to analyze because at the edge of the distribution of the disease, the role of rainfall variability is less controversial. We expect climate variability and climate change to be potentially most relevant to disease dynamics due to the limiting roles of rainfall and temperature concentrated within the monsoon season. With typical features of arid regions of India (*Swaroop*, 1949; *Bouma and van der Kaay*, 1994; *Kiszewski and Teklehaimanot*, 2004), Kheda is chosen as it experiences the seasonal epidemic malaria.

Malaria control programs have played critical roles in transmission dynamics in Kheda and we have both malaria case data and data on implemented control measures (from the National Institute of Malaria Research in India in collaboration by Mercedes Pascual). To ascertain the importance of control in explaining the data, the SEIQS model is tested against other non-control models with SEIRS respectively. For the district of Kheda, we have studied the role of vector control for *P. falciparum* as well as without explicitly including vector control in the analyzed data. Typical vector control protocols, for example insecticide treatments are carried out following years of unusually high malaria incidence, are mostly reactive. To investigate the effectiveness of this control strategies and allow quantitative assessment of potential modifications, inference models with control involvement is examined. Moreover, carrying out inference for models involving control will also enable disentangling the role of control from other epidemiological factors. However, adding control to the model can violate conditional independence of POMP model. Therefore, we will use proper weight as presented above to overcome this issue.

To assess if other covariates should be included in the model, we first plot the monthly cases of *P.falciparum* and monthly rainfall as well as monthly control. Due to typically seasonal epidemic malaria region (*Bhadra et al.*, 2011), the figure shows a lag relationship with rainfall leading malaria, in which malaria peaks a few months

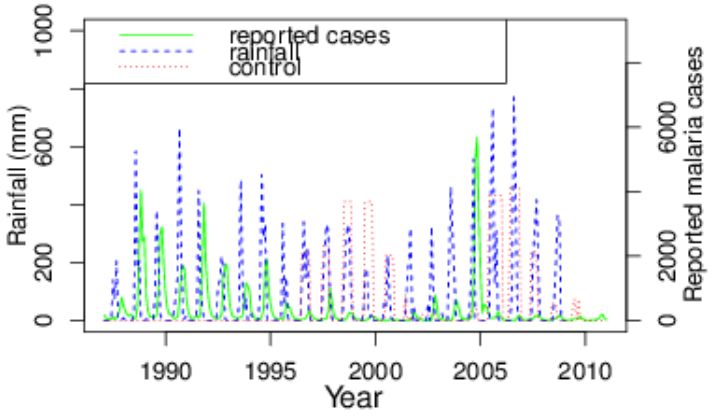

Figure 5.2: Monthly reported *P falciparum* malaria cases (solid line) and monthly rainfall from a local weather station (broken line) for Kheda.

after monsoon. This is expected as the high correlation (0.84) between monsoon and total fall cases, indicating a possible causal effect. However, interpreting the causal relationship is not straightforward as the intrinsic cycles (loss immunity in $2-4$ years) and extrinsic cycles (control, rainfall) may be confounding. A good model, therefore, should take into account all these confounding intrinsic and extrinsic covariates. In this case, we analyze both rainfall and control as well as loss of immunity in our model. The rainfall covariate is fixed by constructing $\tilde{Z}_t$ using $u = 5$ months and $v = 200$ mm. We use multiple regression while taking into consideration the nonlinear stochastic feedbacks and lagged relationships to control for potential confounding variables. It should be noted that the approach is flexible enough to address alternative hypotheses regarding malaria dynamics. We compare these models in terms of the Akaike Information Criterion (AIC), a likelihood-based selection criterion that penalizes higher model complexity. To estimate model parameters and compare different models given the data, we carried out likelihood-based inference via Bayes map iterated filtering (*Ionides et al.*, 2015). Since this is a plug-and-play methodology, to

analyze the data, the user only needs to provide sampling function for dynamic transition model and evaluating function for measurement model. The Euler solution to the dynamic model is generated by the code, which is a built-in feature of the open source R package **pomp**. We also use the R package **pomp** (*King et al.*, 2015c) to implement the Bayes map iterated filtering (mif2) algorithm. For SEIQS model, parameter definitions are given as in Table 5.2.

A comparison of mechanistic models to non-mechanistic model using goodness of fit might be of interest because it can help to identify which part of the data are not captured by the current scientific knowledges. In this regard, we compare our model to a standard Log-SARIMA model using AIC, justifying the need of including some additional intrinsic and extrinsic parameters. In many biological population studies, Log-SARIMA are suitably fitted for disease transmission because annually cycles of abundance frequently consist of alternatively exponential growth and exponential decay. As another benchmark used in (*Bhadra et al.*, 2011), in which rainfall covariate is included into the Log-SARIMA model via ARMAX frameworks *Shumway and Stoffer* (2006), is also used for comparison. The improvement is evident in term of units of log-likelihood.

It could be seen from Table 5.1 that all of the mechanistic models are adequate statistical explanations of the data since they defeat the benchmark non-mechanistic log-SARIMA model by a large margin of AIC. We now compare these models amongst each other. The likelihoods for both the SEIQS model and the simpler SEIR model improve significantly when the control covariate is used ($p < 0.001$ for the likelihood ratio test, using a chi-square approximation on one degree of freedom). Using Akaike Information Criterion (AIC) values to compare two models having different numbers of parameters shows that SEIQS model with control is the best. We concluded that incorporating characteristic aspects of control into models used for time series analysis help fitting the model better.

Model	Log likelihood (ℓ)	p	AIC
Log-SARIMA $(1,0,1)\times(1,0,1)_{12}$ with control	-1524	7	3062
IF2 without control	-1493.0	19	3024
IF2 with control	-1488.0	20	3016

Table 5.1: A likelihood-based comparison of the fitted models. AIC is defined as $-2\ell + 2p$.

Symbol	Brief description	Unit	Fixed value
μ_{XY}	Per-capita transition rate from X to $Y; X,\ Y \in \{S, E, I, Q\}$	yr^{-1}	_
$[X]_0$	Initial fraction in compartment $X; X \in \{S, E, I, Q\}$	_	_
$[\lambda_i]_0$	Initial values for the latent force of infection ($i = 1, \ldots, k$)	_	_
τ	Mean development delay for mosquitoes	yr	_
σ	Standard deviation of the process noise	$\text{yr}^{1/2}$	_
ρ	Reporting fraction	_	_
q	Relative infectivity of partially immune individuals	_	_
c	Coefficient of reinfection with clinical immunity	_	_
k	Shape parameter for the delay development kemel for mosquitoes	_	2
ψ	Dispersion parameter of the observation noise	_	_
n_s	Number of splines describing seasonality	_	6
β_i	Spline coefficients	_	_
$\overline{\beta}$	Dimensionality constant	yr^{-1}	1
β	Coefficient of climate (rainfall) covariate	_	_
u	Window for rainfall to affect transmission	mo	5
v	Threshold for integrated rainfall	mm	200
$1/\delta$	Average life expectancy	yr	50
$\triangle$	Time step for stochastic Euler integration day	day	1

Table 5.2: List of symbols used in the article with a description and units.

NOTE: Some parameters were fixed for the analysis presented in this article, and their value is given in the last column. Given their fixed values, as justified in *Laneri et al.* (2010), $k = 2$.

We use Occam's Razor principles by aiming at the simplest model that can capture the most important features of the biological system. In our model, we extend the work of *Bhadra et al.* (2011) where noise is modeled using nondecreasing Levy processes to enforce non negativity constraints. Some parameters are constrained to be positive while others are constrained to be in interval $(0, 1)$. A natural way to deal with these constraints is by using log transform scale for the positive parameters and using logit transform for the parameters in $(0, 1)$. In the unconstrained maximization problem, if the perturbations share a common scale, the algorithm will perform reasonably well. The tuning perturbation parameters are also called algorithmic parameters, which help to improve the numerical efficiency, but it has no impact in the scientific conclusion once the convergence reached. For scientific interpretation, the results in parameter estimation scale, are then transform back to the original scale. The general classes of non stationary partially observed systems is then modeled in such a way that complexity is close to the limit, in which the available data can support. Since it is rather new methodology, we try to investigate its ability in dealing with the potentially weak identifiability of some parameters, we computed confidence intervals for each MLE by using the profile likelihood method (*Bhadra et al.*, 2011; *Bretó et al.*, 2009).

Confidence intervals are preferred for drawing scientific conclusions to point estimate when the statistical evidence becomes weak, i.e. the confidence interval becomes wider. Without making any specific assumptions on the values of the 25 parameters, the profile likelihood plotted in Figure 5.3, shows that effective control is less than 45%. Indeed, due to the limited resource, the control is only reactive and depends on previous observations, which miss potential outbreak (e.g. outbreak in 2005).

Asymptotically, confidence intervals in Figure 5.3 can be derived from the chi-square cutoffs and the likelihood ratio tests (*Barndorff-Nielsen and Cox*, 1994). For the SEIQS model, the actual coverage probability of control is close to its nominal value in Figure 5.3 by simulating from the model at the MLE parameter values and reconstructing a profile on control for each simulation. Since the likelihood ratios are sufficiently large, substantial conclusions are statistically justified. Another interesting note is that the delay time for mosquito development is about more than a month from Figure 5.4, which seems plausible. In principle, exact nominal coverage at the MLE of a profile likelihood cutoff can be computed from simulation. However, in general, the difference may be large and the estimated parameters may be weakly identifiable. Note that in the presence of weak identifiability, interpretation should be carried out with cautions. However, as we can focus only on conclusions which are robust to identifiability, the model comparison via log-likelihoods in Table 5.1 is always valid.

Note that optimizing the likelihood function involves integrating over all about 25 unobserved state variables is a computational expensive task, especially in the presence of non-convexity and multi-modality. However, verifying that this is indeed approximately maximized is not that hard. There are several methods for doing it. One approach is to check the stability of the maximization by randomly start from different starting values. In order to further extensions of such analyses and work effectively with challenging data, advanced methodology is needed. Hence, such scientific challenge of learning about *P. falciparum* dynamics will be used to test our methodological advances. Figure 5.5 shows some results for a SEIQS model with control, applied to data from Kheda. It could be seen that IF2 does a great job of global optimization in a single search of 100 iterated filtering steps. A practical optimization scheme should include randomly multiple starting values, iteratively selecting the most successful candidates and further investigating their neighborhoods

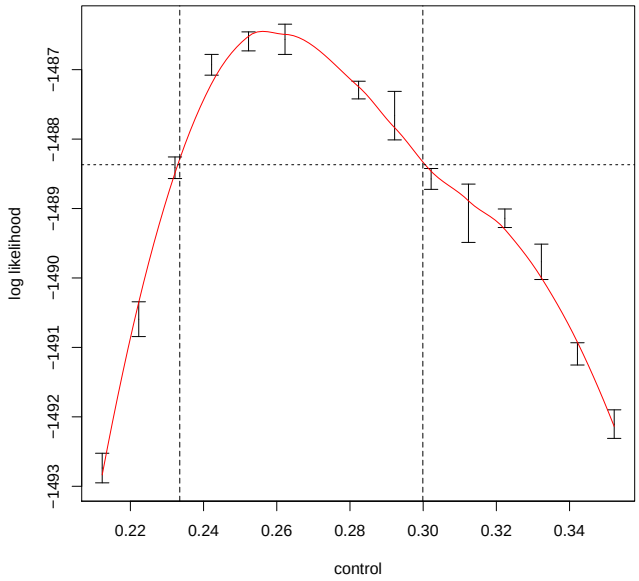

Figure 5.3: Profile likelihood plot for the control (bc) for the SEIQS model with rainfall. The profile is estimated via fitting a smooth curve through Monte Carlo evaluations shown as confidence interval segment SEIQS.

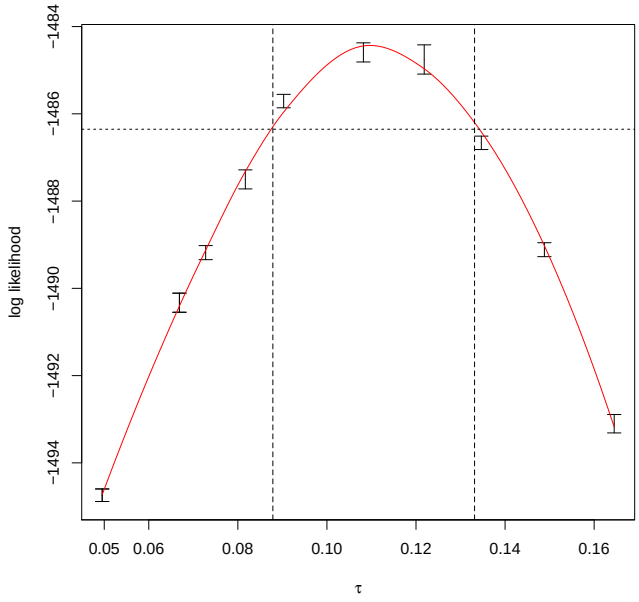

Figure 5.4: Profile likelihood plot for the mean development delay time of mosquitoes (τ) for the SEIQS model with rainfall.

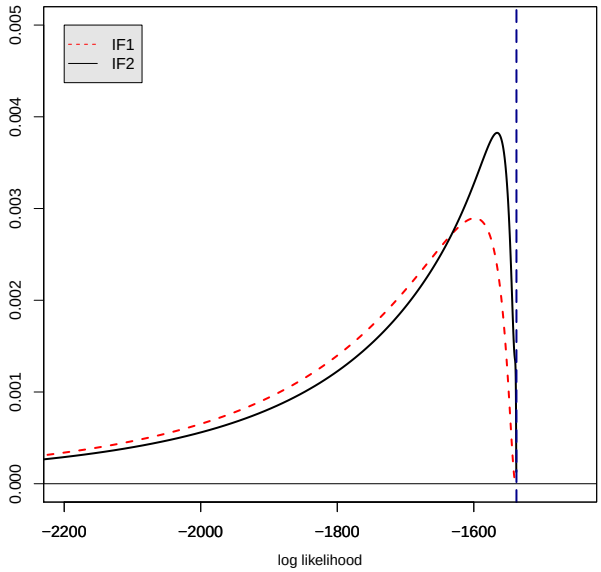

Figure 5.5: Comparison between IF1 and IF2.

with stochastic re-starts. Mean of the MLE computed using IF2 is higher clearly indicates that IF2 out-performs IF1. We reconfirm that Bayes map iterated filtering (*Ionides et al.*, 2015) converges faster and uses less resource while achieve the same statistical efficiency level as its counter part.

5.6 Conclusion

In this paper, we have revisited Bayes map iterated filtering theory using an elegant approach of super-martingale inequality. We have shown that combining two state of the art approaches results in an algorithm which has led to many advances including the statistical and computational efficiency. This is also very fruitful as it is extendable to a more generalized class of algorithm, based on martingale theory. Previous proof of Bayes map iterated filtering require some ad-hoc conditions, which is not easily verifiable. However, in this article, we use just general standard stochastic approximation conditions. We are going further down the road of more systematic

approach which could be easily generalized to state of the art algorithm in the optimization literatures. The convergence rate is not explicitly stated but automatically follows that of super-martingale theory. From theoretical point of view, it could be interesting perspective and insight.

In addition, from practical point of view, we have provided an efficient framework, applicable to a general class of nonlinear, non-Gaussian non-standard POMP models, especially suitable in the control feedback system. There are a lot of such systems, which are not well-treated by current available modeling framework. We simultaneously present the performance of our open source software package pomp to facilitate the needs of the community. The performance of this new approaches beat the other framework by a large margin of magnitude.

It may be surprising that this simple martingale inequality has the needed convergence properties, and can easily be generalized, at least in some asymptotic sense. It is not hard to show that the Bayes map iterated filtering theory can be adapted to apply with iterated smoothing and with either independent white noise or random walk perturbations while our empirical results still show strong evidences of the improvements. In principle, different simulation-based inference methods can readily be hybridized to build on the strongest features of multiple algorithms. Our results could also be applied to develop other plug-and-play methodologies which can take advantage of Bayes map. For example, it may be possible to use our approach to help design efficient proposal distributions for particle Markov chain Monte Carlo algorithms. The theoretical and algorithmic innovations of this paper help to build a new direction for future developments on this frontier. Applying this approach to methodologies like Approximate Bayesian Computation (ABC), Liu-West Particle Filter (LW-PF), Particle Markov chain Monte Carlo (PMCMC), with different samplers scheme, e.g. forward backward particle filter, forward smoothing or forward backward smoothing are foreseeable extensions.

CHAPTER VI

Statistical Inference for Partially Observed Markov Processes via the R Package pomp

6.1 Introduction

A partially observed Markov process (POMP) model consists of incomplete and noisy measurements of a latent, unobserved Markov process. The far-reaching applicability of this class of models has motivated much software development (*Commandeur et al.*, 2011). It has been a challenge to provide a software environment that can effectively handle broad classes of POMP models and take advantage of the wide range of statistical methodologies that have been proposed for such models. The pomp software package (*King et al.*, 2014) differs from previous approaches by providing a general and abstract representation of a POMP model. Therefore, algorithms implemented within pomp are necessarily applicable to arbitrary POMP models. Moreover, models formulated with pomp can be analyzed using multiple methodologies in search of the most effective method, or combination of methods, for the problem at hand. However, since pomp is designed for general POMP models, methods that exploit additional model structure have yet to be implemented. In particular, when linear, Gaussian approximations are adequate for one's purposes, or when the latent process takes values in a small, discrete set, methods that ex-

ploit these additional assumptions to advantage, such as the extended and ensemble Kalman filter methods or exact hidden-Markov-model methods, are available, but not yet as part of pomp. It is the class of nonlinear, non-Gaussian POMP models with large state spaces upon which pomp is focused.

A POMP model may be characterized by the transition density for the Markov process and the measurement density[1]. However, some methods require only simulation from the transition density whereas others require evaluation of this density. Still other methods may not work with the model itself but with an approximation, such as a linearization. Algorithms for which the dynamic model is specified only via a simulator are said to be *plug-and-play* (*Bretó et al.*, 2009; *He et al.*, 2010). Plug-and-play methods can be employed once one has "plugged" a model simulator into the inference machinery. Since many POMP models of scientific interest are relatively easy to simulate, the plug-and-play property facilitates data analysis. Even if one candidate model has tractable transition probabilities, a scientist will frequently wish to consider alternative models for which these probabilities are intractable. In a plug-and-play methodological environment, analysis of variations in the model can often be achieved by changing a few lines of the model simulator codes. The price one pays for the flexibility of plug-and-play methodology is primarily additional computational effort, which can be substantial. Nevertheless, plug-and-play methods implemented using pomp have proved capable for state of the art inference problems (e.g., *King et al.*, 2008; *Bhadra et al.*, 2011; *Shrestha et al.*, 2011, 2013; *Earn et al.*, 2012b; *Roy et al.*, 2013; *Blackwood et al.*, 2013a,b; *Bretó*, 2014; *Blake et al.*, 2014). The recent surge of interest in plug-and-play methodology for POMP models includes the development of nonlinear forecasting (*Ellner et al.*, 1998), iterated filtering (*Ionides et al.*, 2006, 2015), ensemble Kalman filtering (*Shaman and Karspeck*, 2012), ap-

[1]We use the term "density" in this article encompass both the continuous and discrete cases. Thus, in the latter case, i.e., when state variables and/or measured quantities are discrete, one could replace "probability density function" with "probability mass function".

proximate Bayesian computation (ABC) (*Sisson et al.*, 2007), particle Markov chain Monte Carlo (PMCMC) (*Andrieu et al.*, 2010), probe matching (*Kendall et al.*, 1999), and synthetic likelihood (*Wood*, 2010). Although the pomp package provides a general environment for methods with and without the plug-and-play property, development of the package to date has emphasized plug-and-play methods.

The pomp package is philosophically neutral as to the merits of Bayesian inference. It enables a POMP model to be supplemented with prior distributions on parameters, and several Bayesian methods are implemented within the package. Thus pomp is a convenient environment for those who wish to explore both Bayesian and non-Bayesian data analyses.

The remainder of this paper is organized as follows. 6.2 defines mathematical notation for POMP models and relates this to their representation as objects of *pomp* in the pomp package. 6.3 introduces several of the statistical methods currently implemented in pomp. 6.4 constructs and explores a simple POMP model, demonstrating the use of the available statistical methods. 6.5 illustrates the implementation of more complex POMPs, using a model of infectious disease transmission as an example. Finally, 6.6 discusses extensions and applications of pomp.

6.2 POMP models and their representation in pomp

Let θ be a p-dimensional real-valued parameter, $\theta \in \mathbb{R}^p$. For each value of θ, let $\{X(t\,;\theta), t \in T\}$ be a Markov process, with $X(t\,;\theta)$ taking values in $\mathbb{R}^q$. The time index set $T \subset \mathbb{R}$ may be an interval or a discrete set. Let $\{t_i \in T, i = 1, \ldots, N\}$, be the times at which $X(t\,;\theta)$ is observed, and $t_0 \in T$ be an initial time. Assume $t_0 \le t_1 < t_2 < \cdots < t_N$. We write $X_i = X(t_i\,;\theta)$ and $X_{i:j} = (X_i,\ X_{i+1}, \ldots, X_j)$. The process $X_{0:N}$ is only observed by way of another process $Y_{1:N} = (Y_1, \ldots, Y_N)$ with Y_n taking values in $\mathbb{R}^r$. The observable random variables $Y_{1:N}$ are assumed to be conditionally independent given $X_{0:N}$. The data, $y^*_{1:N} = (y^*_1,\ \ldots,\ y^*_N)$, are modeled

as a realization of this observation process and are considered fixed. We suppose that $X_{0:N}$ and $Y_{1:N}$ have a joint density $f_{X_{0:N},Y_{1:N}}(x_{0:n},\ y_{1:n}\,;\theta)$. The POMP structure implies that this joint density is determined by the initial density, $f_{X_0}(x_0;\theta)$, together with the conditional transition probability density, $f_{X_n|X_{n-1}}(x_n\,|\,x_{n-1}\,;\theta)$, and the measurement density, $f_{Y_n|X_n}(y_n\,|\,x_n\,;\theta)$, for $1 \le n \le N$. In particular, we have

$$f_{X_{0:N},Y_{1:N}}(x_{0:n},y_{1:n};\theta) = f_{X_0}(x_0;\theta)\prod_{n=1}^{N} f_{X_n|X_{n-1}}(x_n|x_{n-1};\theta)\, f_{Y_n|X_n}(y_n|x_n;\theta). \quad (6.1)$$

Note that this formalism allows the transition density, $f_{X_n|X_{n-1}}$, and measurement density, $f_{Y_n|X_n}$, to depend explicitly on n.

6.2.1 Implementation of POMP models

pomp is fully object-oriented: in the package, a POMP model is represented by an S4 object (*Chambers*, 1998; *Genolini*, 2008) of pomp. Slots in this object encode the components of the POMP model, and can be filled or changed using the constructor function pomp and various other convenience functions. Methods for the pomp class use these components to carry out computations on the model. 6.1 gives the mathematical notation corresponding to the elementary methods that can be executed on a pomp object.

The rprocess, dprocess, rmeasure, and dmeasure arguments specify the transition probabilities $f_{X_n|X_{n-1}}(x_n\,|\,x_{n-1}\,;\theta)$ and measurement densities $f_{Y_n|X_n}(y_n\,|\,x_n\,;\theta)$. Not all of these arguments must be supplied for any specific computation. In particular, plug-and-play methodology by definition never uses dprocess. An empty dprocess slot in a pomp object is therefore acceptable unless a non-plug-and-play algorithm is attempted. In the package, the data and corresponding measurement times are considered necessary parts of a pomp object whilst specific values of the parameters and latent states are not. Applying the simulate function to an object of pomp

Method	Argument to the pomp constructor	Mathematical terminology
rprocess	rprocess	Simulate from $f_{X_n\|X_{n-1}}(x_n \,\|\, x_{n-1}\,;\theta)$
dprocess	dprocess	Evaluate $f_{X_n\|X_{n-1}}(x_n \,\|\, x_{n-1}\,;\theta)$
rmeasure	rmeasure	Simulate from $f_{Y_n\|X_n}(y_n \,\|\, x_n\,;\theta)$
dmeasure	dmeasure	Evaluate $f_{Y_n\|X_n}(y_n \,\|\, x_n\,;\theta)$
rprior	rprior	Simulate from the prior distribution $\pi(\theta)$
dprior	dprior	Evaluate the prior density $\pi(\theta)$
init.state	initializer	Simulate from $f_{X_0}(x_0\,;\theta)$
timezero	t0	t_0
time	times	$t_{1:N}$
obs	data	$y^*_{1:N}$
states	—	$x_{0:N}$
coef	params	θ

Table 6.1: Constituent methods for pomp objects and their translation into mathematical notation for POMP models. For example, the rprocess method is set using the rprocess argument to the pomp constructor function.

returns another object of pomp, within which the data $y^*_{1:N}$ have been replaced by a stochastic realization of $Y_{1:N}$, the corresponding realization of $X_{0:N}$ is accessible via the states method, and the params slot has been filled with the value of θ used in the simulation.

To illustrate the specification of models in pomp and the use of the package's inference algorithms, we will use a simple example. The *Gompertz* (1825) model can be constructed via

```
R> library("pomp")
R> pompExample(gompertz)
```

which results in the creation of an object of pomp, named gompertz, in the workspace. The structure of this model and its implementation in pomp is described below, in 6.4. One can view the components of gompertz listed in 6.1 by executing

```
R> obs(gompertz)
R> states(gompertz)
R> as.data.frame(gompertz)
```

```
R> plot(gompertz)
R> timezero(gompertz)
R> time(gompertz)
R> coef(gompertz)
R> init.state(gompertz)
```

Executing pompExamples() lists other examples provided with the package.

6.2.2 Initial conditions

In some experimental situations, $f_{X_0}(x_0\,;\theta)$ corresponds to a known experimental initialization, but in general the initial state of the latent process will have to be inferred. If the transition density for the dynamic model, $f_{X_n|X_{n-1}}(x_n\,|\,x_{n-1}\,;\theta)$, does not depend on time and possesses a unique stationary distribution, it may be natural to set $f_{X_0}(x_0\,;\theta)$ to be this stationary distribution. Otherwise, and more commonly in the authors' experience, no clear scientifically motivated choice of $f_{X_0}(x_0\,;\theta)$ exists and one can proceed by treating the value of X_0 as a parameter to be estimated. In this case, $f_{X_0}(x_0\,;\theta)$ concentrates at a point, the location of which depends on θ.

6.2.3 Covariates

Scientifically, one may be interested in the role of a vector-valued covariate process $\{Z(t)\}$ in explaining the data. Modeling and inference conditional on $\{Z(t)\}$ can be carried out within the general framework for nonhomogeneous POMP models, since the arbitrary densities $f_{X_n|X_{n-1}}$, f_{X_0} and $f_{Y_n|X_n}$ can depend on the observed process $\{Z(t)\}$. For example, it may be the case that $f_{X_n|X_{n-1}}(x_n\,|\,x_{n-1}\,;\theta)$ depends on n only through $Z(t)$ for $t_{n-1} \le t \le t_n$. The covar argument in the pomp constructor allows for time-varying covariates measured at times specified in the tcovar argument. A example using covariates is given in 6.5.

6.3 Methodology for POMP models

Data analysis typically involves identifying regions of parameter space within which a postulated model is statistically consistent with the data. Additionally, one frequently desires to assess the relative merits of alternative models as explanations of the data. Once the user has encoded one or more POMP models as objects of pomp, the package provides a variety of algorithms to assist with these data analysis goals. 6.2 provides an overview of several inference methodologies for POMP models. Each method may be categorized as full-information or feature-based, Bayesian or frequentist, and plug-and-play or not plug-and-play.

Approaches that work with the full likelihood function, whether in a Bayesian or frequentist context, can be called full-information methods. Since low-dimensional sufficient statistics are not generally available for POMP models, methods which take advantage of favorable low-dimensional representations of the data typically lose some statistical efficiency. We use the term "feature-based" to describe all methods not based on the full likelihood, since such methods statistically emphasize some features of the data over others.

Many Monte Carlo methods of inference can be viewed as algorithms for the exploration of high-dimensional surfaces. This view obtains whether the surface in question is the likelihood surface or that of some other objective function. The premise behind many recent methodological developments in Monte Carlo methods for POMP models is that generic stochastic numerical analysis tools, such as standard Markov chain Monte Carlo and Robbins-Monro type methods, are effective only on the simplest models. For many models of scientific interest, therefore, methods that leverage the POMP structure are needed. Though pomp has sufficient flexibility to encode arbitrary POMP models and methods and therefore also provides a platform for the development of novel POMP inference methodology, pomp's development to date has focused on plug-and-play methods. However, the package developers welcome

(a) Plug-and-play

	Frequentist	Bayesian
Full information	Iterated filtering (mif, 6.3.2)	PMCMC (pmcmc, 6.3.3)
Feature-based	Nonlinear forecasting (nlf, 6.3.6), synthetic likelihood (probe.match, 6.3.4)	ABC (abc, 6.3.5)

(b) Not plug-and-play

	Frequentist	Bayesian
Full information	EM and Monte Carlo EM, Kalman filter	MCMC
Feature-based	Trajectory matching (traj.match), extended Kalman filter, Yule-Walker equations	Extended Kalman filter

Table 6.2: Inference methods for POMP models. For those currently implemented in pomp, function name and a reference for description are provided in parentheses. Standard Expectation-Maximization (EM) and Markov chain Monte Carlo (MCMC) algorithms are not plug-and-play since they require evaluation of $f_{X_n|X_{n-1}}(x_n \,|\, x_{n-1}\,;\theta)$. The Kalman filter and extended Kalman filter are not plug-and-play since they cannot be implemented based on a model simulator. The Kalman filter provides the likelihood for a linear, Gaussian model. The extended Kalman filter employs a local linear Gaussian approximation which can be used for frequentist inference (via maximization of the resulting quasi-likelihood) or approximate Bayesian inference (by adding the parameters to the state vector). The Yule-Walker equations for ARMA models provide an example of a closed-form method of moments estimator.

contributions and collaborations to further expand pomp's functionality in non-plug-and-play directions also. In the remainder of this Section, we describe and discuss several inference methods, all currently implemented in the package.

Algorithm 2: Sequential Monte Carlo (SMC, or particle filter): pfilter(P, Np = J), using notation from 6.1 where P is a pomp object with definitions for rprocess, dmeasure, init.state, coef, and obs.

input: Simulator for $f_{X_n|X_{n-1}}(x_n \,|\, x_{n-1}\,;\theta)$; evaluator for $f_{Y_n|X_n}(y_n \,|\, x_n\,;\theta)$; simulator for $f_{X_0}(x_0\,;\theta)$; parameter, θ; data, $y^*_{1:N}$; number of particles, J.

Initialize filter particles: simulate $X^F_{0,j} \sim f_{X_0}(\,\cdot\,;\theta)$ for j in $1\!:\!J$.
for n *in* $1\!:\!N$ **do**
  Simulate for prediction: $X^P_{n,j} \sim f_{X_n|X_{n-1}}\big(\,\cdot\,|X^F_{n-1,j};\theta\big)$ for j in $1\!:\!J$.
  Evaluate weights: $w(n,j) = f_{Y_n|X_n}(y^*_n|X^P_{n,j}\,;\theta)$ for j in $1\!:\!J$.
  Normalize weights: $\tilde{w}(n,j) = w(n,j)/\sum_{m=1}^{J} w(n,m)$.
  Apply 3 to select indices $k_{1:J}$ with $\mathbb{P}k_j = m = \tilde{w}(n,m)$.
  Resample: set $X^F_{n,j} = X^P_{n,k_j}$ for j in $1\!:\!J$.
  Compute conditional log likelihood: $\hat{\ell}_{n|1:n-1} = \log\big(J^{-1}\sum_{m=1}^{J} w(n,m)\big)$.
end

output: Log likelihood estimate, $\hat{\ell}(\theta) = \sum_{n=1}^{N} \hat{\ell}_{n|1:n-1}$; filter sample, $X^F_{n,1:J}$, for n in $1\!:\!N$.

complexity: $\mathcal{O}(J)$

6.3.1 The likelihood function and sequential Monte Carlo

The log likelihood for a POMP model is $\ell(\theta) = \log f_{Y_{1:N}}(y^*_{1:N}\,;\theta)$, which can be written as a sum of conditional log likelihoods,

$$\ell(\theta) = \sum_{n=1}^{N} \ell_{n|1:n-1}(\theta), \tag{6.2}$$

where

$$\ell_{n|1:n-1}(\theta) = \log f_{Y_n|Y_{1:n-1}}(y^*_n \,|\, y^*_{1:n-1}\,;\theta), \tag{6.3}$$

and we use the convention that $y^*_{1:0}$ is an empty vector. The structure of a POMP model implies the representation

$$\ell_{n|1:n-1}(\theta) = \log \int f_{Y_n|X_n}(y^*_n|x_n\,;\theta) f_{X_n|Y_{1:n-1}}(x_n \,|\, y^*_{1:n-1}\,;\theta)\, dx_n \tag{6.4}$$

(cf. 6.1). Although $\ell(\theta)$ typically has no closed form, it can frequently be computed by Monte Carlo methods. Sequential Monte Carlo (SMC) builds up a representation of $f_{X_n|Y_{1:n-1}}(x_n \,|\, y^*_{1:n-1}\,;\theta)$ that can be used to obtain an estimate, $\hat{\ell}_{n|1:n-1}(\theta)$, of $\ell_{n|1:n-1}(\theta)$ and hence an approximation, $\hat{\ell}(\theta)$, to $\ell(\theta)$. SMC (a basic version of which is presented as 2), is also known as the particle filter, since it is conventional to describe the Monte Carlo sample, $\{X^F_{n,j}, j \text{ in } 1\!:\!J\}$ as a swarm of particles representing $f_{X_n|Y_{1:n}}(x_n \,|\, y^*_{1:n}\,;\theta)$. The swarm is propagated forward according to the dynamic model and then assimilated to the next data point. Using an evolutionary analogy, the prediction step (3) mutates the particles in the swarm and the filtering step (7) corresponds to selection. SMC is implemented in pomp in the pfilter function. The basic particle filter in 2 possesses the plug-and-play property. Many variations and elaborations to SMC have been proposed; these may improve numerical performance in appropriate situations (*Cappé et al.*, 2007) but typically lose the plug-and-play property. *Arulampalam et al.* (2002), *Doucet and Johansen* (2009), and *Kantas et al.* (2015) have written excellent introductory tutorials on the particle filter and particle methods more generally.

Basic SMC methods fail when an observation is extremely unlikely given the model. This leads to the situation that at most a few particles are consistent with the observation, in which case the effective sample size (*Liu*, 2001) of the Monte Carlo sample is small and the particle filter is said to suffer from *particle depletion*. Many elaborations of the basic SMC algorithm have been proposed to ameliorate this problem. However, it is often preferable to remedy the situation by seeking a better model. The plug-and-play property assists in this process by facilitating investigation of alternative models.

In 6 of 2, systematic resampling (3) is used in preference to multinomial resampling. 3 reduces Monte Carlo variability while resampling with the proper marginal probability. In particular, if all the particle weights are equal then 3 has the appro-

Algorithm 3: Systematic resampling: 6 of 2.

input: Weights, $\tilde{w}_{1:J}$, normalized so that $\sum_{j=1}^{J} \tilde{w}_j = 1$.

Construct cumulative sum: $c_j = \sum_{m=1}^{j} \tilde{w}_m$, for j in $1 : J$.
Draw a uniform initial sampling point: $U_1 \sim \text{Uniform}(0, J^{-1})$.
Construct evenly spaced sampling points: $U_j = U_1 + (j-1)J^{-1}$, for j in $2 : J$.
Initialize: set $p = 1$.
for j *in* $1 : J$ **do**
    while $U_j > c_p$ **do**
        Step to the next resampling index: set $p = p + 1$.
    end
    Assign resampling index: set $k_j = p$.
end

output: Resampling indices, $k_{1:J}$.
complexity: $\mathcal{O}(J)$

priate behavior of leaving the particles unchanged. As pointed out by (*Douc et al.*, 2005), stratified resampling performs better than multinomial sampling and 3 is in practice comparable in performance to stratified resampling and somewhat faster.

6.3.2 Iterated filtering

Iterated filtering techniques maximize the likelihood obtained by SMC (*Ionides et al.*, 2006, 2011, 2015). The key idea of iterated filtering is to replace the model we are interested in fitting—which has time-invariant parameters—with a model that is just the same except that its parameters take a random walk in time. Over multiple repetitions of the filtering procedure, the intensity of this random walk approaches zero and the modified model approaches the original one. Adding additional variability in this way has four positive effects:

A1. It smooths the likelihood surface, which facilitates optimization.

A2. It combats particle depletion by adding diversity to the population of particles.

A3. The additional variability can be exploited to explore the likelihood surface and estimate of the gradient of the (smoothed) likelihood, based on the same filtering

procedure that is required to estimate of the value of the likelihood (*Ionides et al.*, 2006, 2011).

A4. It preserves the plug-and-play property, inherited from the particle filter.

Iterated filtering is implemented in the mif function. By default, mif carries out the procedure of *Ionides et al.* (2006). The improved iterated filtering algorithm (IF2) of *Ionides et al.* (2015) has shown promise. A limited version of IF2 is available via the method="mif2" option; a full version of this algorithm will be released soon. In all iterated filtering methods, by analogy with annealing, the random walk intensity can be called a temperature, which is decreased according to a prescribed cooling schedule. One strives to ensure that the algorithm will freeze at the maximum of the likelihood as the temperature approaches zero.

The perturbations on the parameters in 2,6 of 4 follow a normal distribution, with each component, $[\theta]_i$, of the parameter vector perturbed independently. Neither normality nor independence are necessary for iterated filtering, but, rather than varying the perturbation distribution, one can transform the parameters to make these simple algorithmic choices reasonable.

4 gives special treatment to a subset of the components of the parameter vector termed initial value parameters (IVPs), which arise when unknown initial conditions are modeled as parameters. These IVPs will typically be inconsistently estimable as the length of the time series increases, since for a stochastic process one expects only early time points to contain information about the initial state. Searching the parameter space using time-varying parameters is inefficient in this situation, and so 4 perturbs these parameters only at time zero.

6,7,8,9,10,11 of 4 are exactly an application of SMC (2) to a modified POMP model in which the parameters are added to the state space. This approach has been used in a variety of previously proposed POMP methodologies (*Kitagawa*, 1998; *Liu and West*, 2001; *Wan and van der Merwe*, 2000) but iterated filtering is distinguished by

having theoretical justification for convergence to the maximum likelihood estimate (*Ionides et al.*, 2011).

6.3.3 Particle Markov chain Monte Carlo

Full-information plug-and-play Bayesian inference for POMP models is enabled by particle Markov chain Monte Carlo (PMCMC) algorithms (*Andrieu et al.*, 2010). PMCMC methods combine likelihood evaluation via SMC with MCMC moves in the parameter space. The simplest and most widely used PMCMC algorithm, termed particle marginal Metropolis-Hastings (PMMH), is based on the observation that the unbiased likelihood estimate provided by SMC can be plugged in to the Metropolis-Hastings update procedure to give an algorithm targeting the desired posterior distribution for the parameters (*Andrieu and Roberts*, 2009). PMMH is implemented in pmcmc, as described in 5. In part because it gains only a single likelihood evaluation from each particle-filtering operation, PMCMC can be computationally relatively inefficient (*Bhadra*, 2010; *Ionides et al.*, 2015). Nevertheless, its invention introduced the possibility of full-information plug-and-play Bayesian inferences in some situations where they had been unavailable.

6.3.4 Synthetic likelihood of summary statistics

Some motivations to estimate parameter based on features rather than the full likelihood include

B1. Reducing the data to sensibly selected and informative low-dimensional summary statistics may have computational advantages (*Wood*, 2010).

B2. The scientific goal may be to match some chosen characteristics of the data rather than all aspects of it. Acknowledging the limitations of all models, this limited aspiration may be all that can reasonably be demanded (*Kendall et al.*, 1999; *Wood*, 2001).

B3. In conjunction with full-information methodology, consideration of individual features has diagnostic value to determine which aspects of the data are driving the full-information inferences (*Reuman et al.*, 2006).

B4. Feature-based methods for dynamic models typically do not require the POMP model structure. However, that benefit is outside the scope of the pomp package.

B5. Feature-based methods are typically *doubly plug-and-play*, meaning that they require simulation, but not evaluation, for both the latent process transition density and the measurement model.

When pursuing goal B1, one aims to find summary statistics which are as close as possible to sufficient statistics for the unknown parameters. Goals B2 and B3 deliberately look for features which discard information from the data; in this context the features have been called probes (*Kendall et al.*, 1999). The features are denoted by a collection of functions, $\mathbb{S} = (\mathbb{S}_1, \ldots, \mathbb{S}_d)$, where each $\mathbb{S}_i$ maps an observed time series to a real number. We write $S = (S_1, \ldots, S_d)$ for the vector-valued random variable with $S = \mathbb{S}(Y_{1:N})$, with $f_S(s\,;\theta)$ being the corresponding joint density. The observed feature vector is s^* where $s^*_i = \mathbb{S}_i(y^*_{1:N})$, and for any parameter set one can look for parameter values for which typical features for simulated data match the observed features. One can define a likelihood function, $\ell_{\mathbb{S}}(\theta) = f_S(s^*\,;\theta)$. Arguing that S should be approximately multivariate normal, for suitable choices of the features, *Wood* (2010) proposed using simulations to construct a multivariate normal approximation to $\ell_{\mathbb{S}}(\theta)$, and called this a *synthetic likelihood*.

Simulation-based evaluation of a feature matching criterion is implemented by probe (6). The feature matching criterion requires a scale, and a natural scale to use is the empirical covariance of the simulations. Working on this scale, as implemented by probe, there is no substantial difference between the probe approaches of *Kendall et al.* (1999) and *Wood* (2010). Numerical optimization of the synthetic likelihood is

implemented by probe.match, which offers a choice of the subplex method (*Rowan*, 1990; *King*, 2008) or any method provided by optim or the nloptr package (*Johnson*, 2014; *Ypma*, 2014).

6.3.5 Approximate Bayesian computation (ABC)

ABC algorithms are Bayesian feature-matching techniques, comparable to the frequentist generalized method of moments (*Marin et al.*, 2012). The vector of summary statistics $\mathbb{S}$, the corresponding random variable S, and the value $s^* = \mathbb{S}(y^*_{1:N})$, are defined as in 6.3.4. The goal of ABC is to approximate the posterior distribution of the unknown parameters given $S = s^*$. ABC has typically been motivated by computational considerations, as in point B1 of 6.3.4 (*Sisson et al.*, 2007; *Toni et al.*, 2009; *Beaumont*, 2010). Points B2 and B3 also apply (*Ratmann et al.*, 2009).

The key theoretical insight behind ABC algorithms is that an unbiased estimate of the likelihood can be substituted into a Markov chain Monte Carlo algorithm to target the required posterior, the same result that justifies PMCMC (*Andrieu and Roberts*, 2009). However, ABC takes a different approach to approximating the likelihood. The likelihood of the observed features, $\ell_S(\theta) = f_S(s^*\,;\theta)$, has an approximately unbiased estimate based on a single Monte Carlo realization $Y_{1:N} \sim f_{Y_{1:N}}(\,\cdot\,;\theta)$ given by

$$\hat{\ell}^{ABC}_{\mathbb{S}}(\theta) = \begin{cases} \epsilon^{-d} B_d^{-1} \prod_{i=1}^{d} \tau_i, & \text{if } \sum_{i=1}^{d} \left(\frac{s_i - s_i^*}{\tau_i} \right)^2 < \epsilon^2, \\ 0, & \text{otherwise,} \end{cases} \tag{6.6}$$

where B_d is the volume of the d-dimensional unit ball and τ_i is a scaling chosen for the ith feature. The likelihood approximation in 6.6 differs from the synthetic likelihood in 6 in that only a single simulation is required. As ϵ become small, the bias in 6.6 decreases but the Monte Carlo variability increases. The ABC implementation abc (presented in 7) is a random walk Metropolis implementation of ABC-MCMC

(Algorithm 3 of *Marin et al.*, 2012). In the same style as iterated filtering and PMCMC, we assume a Gaussian random walk in parameter space; the package supports alternative choices of proposal distribution.

6.3.6 Nonlinear forecasting

Nonlinear forecasting (NLF) uses simulations to build up an approximation to the one-step prediction distribution that is then evaluated on the data. We saw in 6.3.1 that SMC evaluates the prediction density for the observation, $f_{Y_n|Y_{1:n-1}}(y^*_n \,|\, y^*_{1:n-1}\,;\theta)$, by first building an approximation to the prediction density of the latent process, $f_{X_n|Y_{1:n-1}}(x_n \,|\, y^*_{1:n-1}\,;\theta)$. NLF, by contrast, uses simulations to fit a linear regression model which predicts Y_n based on a collection of L lagged variables, $\{Y_{n-c_1}, \ldots, Y_{n-c_L}\}$. The prediction errors when this model is applied to the data give rise to a quantity called the quasi-likelihood, which behaves for many purposes like a likelihood (*Smith*, 1993). The implementation in nlf maximises the quasi-likelihood computed in 8, using the subplex method (*Rowan*, 1990; *King*, 2008) or any other optimizer offerered by optim. The construction of the quasi-likelihood in nlf follows the specific recommendations of *Kendall et al.* (2005). In particular, the choice of radial basis functions, f_k, in 5 and the specification of m_k and s in 3,4 were proposed by *Kendall et al.* (2005) based on trial and error. The quasi-likelihood is mathematically most similar to a likelihood when $\min(c_{1:L}) = 1$, so that $\ell_Q(\theta)$ approximates the factorization of the likelihood in 6.2. With this in mind, it is natural to set $c_{1:L} = 1 : L$. However, *Kendall et al.* (2005) found that a two-step prediction criterion, with $\min(c_{1:L}) = 2$, led to improved numerical performance. It is natural to ask when one might choose to use quasi-likelihood estimation in place of full likelihood estimation implemented by SMC. Some considerations follow, closely related to the considerations for synthetic likelihood and ABC (6.3.5),(6.3.4).

C1. NLF benefits from stationarity since (unlike SMC) it uses all time points in the

simulation to build a prediction rule valid at all time points. Indeed, NLF has not been considered applicable for non-stationary models and, on account of this, nlf is not appropriate if the model includes time-varying covariates. An intermediate scenario between stationarity and full non-stationarity is seasonality, where the dynamic model is forced by cyclical covariates, and this is supported by nlf (cf. B1 in 6.3.4).

C2. Potentially, quasi-likelihood could be preferable to full likelihood in some situations. It has been argued that a two-step prediction criterion may sometimes be more robust than the likelihood to model misspecification (*Xia and Tong*, 2011) (cf. B2).

C3. Arguably, two-step prediction should be viewed as a diagnostic tool that can be used to complement full likelihood analysis rather than replace it (*Ionides*, 2011) (cf. B3).

C4. NLF does not require that the model be Markovian (cf. B4), although the pomp implementation, nlf, does.

C5. NLF is doubly plug-and-play (cf. B5).

C6. The regression surface reconstruction carried out by NLF does not scale well with the dimension of the observed data. NLF is recommended only for low-dimensional time series observations.

NLF can be viewed as an estimating equation method, and so standard errors can be computed by standard sandwich estimator or bootstrap techniques (*Kendall et al.*, 2005). The optimization in NLF is typically carried out with a fixed seed for the random number generator, so the simulated quasi-likelihood is a deterministic function. If rprocess depends smoothly on the random number sequence and on the parameters,

and the number of calls to the random number generator does not depend on the parameters, then fixing the seed results in a smooth objective function. However, some common components to model simulators, such as rnbinom, make different numbers of calls to the random number generator depending on the arguments, which introduces nonsmoothness into the objective function.

6.4 Model construction and data analysis: simple examples

6.4.1 A first example: the Gompertz model

The plug-and-play methods in pomp were designed to facilitate data analysis based on complicated models, but we will first demonstrate the basics of pomp using simple discrete-time models, the Gompertz and Ricker models for population growth (*Reddingius*, 1971; *Ricker*, 1954). The Ricker model will be introduced in 6.4.5 and used in 6.4.6; the remainder of 6.4 will use the Gompertz model. The Gompertz model postulates that the density, $X_{t+\Delta t}$, of a population of organisms at time $t+\Delta t$ depends on the density, X_t, at time t according to

$$X_{t+\Delta t} = K^{1-e^{-r\,\Delta t}}\, X_t^{e^{-r\,\Delta t}}\, \varepsilon_t. \tag{6.9}$$

In 6.9, K is the carrying capacity of the population, r is a positive parameter, and the ε_t are independent and identically-distributed lognormal random variables with $\log \varepsilon_t \sim \text{Normal}(0, \sigma^2)$. Additionally, we will assume that the population density is observed with errors in measurement that are lognormally distributed:

$$\log Y_t \;\sim\; \text{Normal}\left(\log X_t, \tau^2\right). \tag{6.10}$$

Taking a logarithmic transform of 6.9 gives

$$\log X_{t+\Delta t} \;\sim\; \text{Normal}\left(\left(1-e^{-r\,\Delta t}\right)\,\log K + e^{-r\,\Delta t}\,\log X_t, \sigma^2\right). \qquad (6.11)$$

On this transformed scale, the model is linear and Gaussian and so we can obtain exact values of the likelihood function by applying the Kalman filter. Plug-and-play methods are not strictly needed; this example therefore allows us to compare the results of generally applicable plug-and-play methods with exact results from the Kalman filter. Later we will look at the Ricker model and a continuous-time model for which no such special tricks are available.

The first step in implementing this model in pomp is to construct an R object of pomp that encodes the model and the data. This involves the specification of functions to do some or all of rprocess, rmeasure, and dmeasure, along with data and (optionally) other information. The documentation (?pomp) spells out the usage of the pomp constructor, including detailed specifications for all its arguments and links to several examples.

To begin, we will write a function that implements the process model simulator. This is a function that will simulate a single step ($t \to t+\Delta t$) of the unobserved process (6.9).

```
R> gompertz.proc.sim <- function(x, t, params, delta.t, ...) {
+    eps <- exp(rnorm(n = 1, mean = 0, sd = params["sigma"]))
+    S <- exp(-params["r"] * delta.t)
+    setNames(params["K"]^(1 - S) * x["X"]^S * eps, "X")
+  }
```

The translation from the mathematical description (6.9) to the simulator is straightforward. When this function is called, the argument x contains the state at time t. The parameters (including K, r, and σ) are passed in the argument params. Notice

that x and params are named numeric vectors and that the output must likewise be a named numeric vector, with names that match those of x. The argument delta.t species the time-step size. In this case, the time-step will be 1 unit; we will see below how this is specified.

Next, we will implement a simulator for the observation process, 6.10.

```
R> gompertz.meas.sim <- function(x, t, params, ...) {
+    setNames(rlnorm(n = 1, meanlog = log(x["X"]), sd = params["tau"]), "Y")
+  }
```

Again the translation from the measurement model 6.10 is straightforward. When the function gompertz.meas.sim is called, the named numeric vector x will contain the unobserved states at time t; params will contain the parameters as before. This return value will be a named numeric vector containing a single draw from the observation process (6.10).

Complementing the measurement model simulator is the corresponding measurement model density, which we implement as follows:

```
R> gompertz.meas.dens <- function(y, x, t, params, log, ...) {
+    dlnorm(x = y["Y"], meanlog = log(x["X"]), sdlog = params["tau"],
+      log = log)
+  }
```

We will need this later on for inference using pfilter, mif and pmcmc. When gompertz.meas.dens is called, y will contain the observation at time t, x and params will be as before, and the parameter log will indicate whether the likelihood (log=FALSE) or the log likelihood (log=TRUE) is required.

With the above in place, we build an object of pomp via a call to pomp:

```
R> gompertz <- pomp(data = data.frame(time = 1:100, Y = NA), times = "time",
+    rprocess = discrete.time.sim(step.fun = gompertz.proc.sim, delta.t = 1),
```

```
+    rmeasure = gompertz.meas.sim, t0 = 0)
```

The first argument (data) specifies a data frame that holds the data and the times at which the data were observed. Since this is a toy problem, we have as yet no data; in a moment, we will generate some simulated data. The second argument (times) specifies which of the columns of data is the time variable. The rprocess argument specifies that the process model simulator will be in discrete time, with each step of duration delta.t taken by the function given in the step.fun argument. The rmeasure argument specifies the measurement model simulator function. t0 fixes t_0 for this model; here we have chosen this to be one time unit prior to the first observation.

It is worth noting that implementing the rprocess, rmeasure, and dmeasure components as R functions, as we have done above, leads to needlessly slow computation. As we will see below, pomp provides facilities for specifying the model in C, which can accelerate computations manyfold.

Before we can simulate from the model, we need to specify some parameter values. The parameters must be a named numeric vector containing at least all the parameters referenced by the functions gompertz.proc.sim and gompertz.meas.sim. The parameter vector needs to determine the initial condition $X(t_0)$ as well. Let us take our parameter vector to be

```
R> theta <- c(r = 0.1, K = 1, sigma = 0.1, tau = 0.1, X.0 = 1)
```

The parameters r, K, σ, and τ appear in gompertz.proc.sim and gompertz.meas.sim. The initial condition X_0 is also given in theta. The fact that the initial condition parameter's name ends in .0 is significant: it tells pomp that this is the initial condition of the state variable X. This use of the .0 suffix is the default behavior of pomp: one can however parameterize the initial condition distribution arbitrarily using pomp's optional initializer argument.

We can now simulate the model at these parameters:

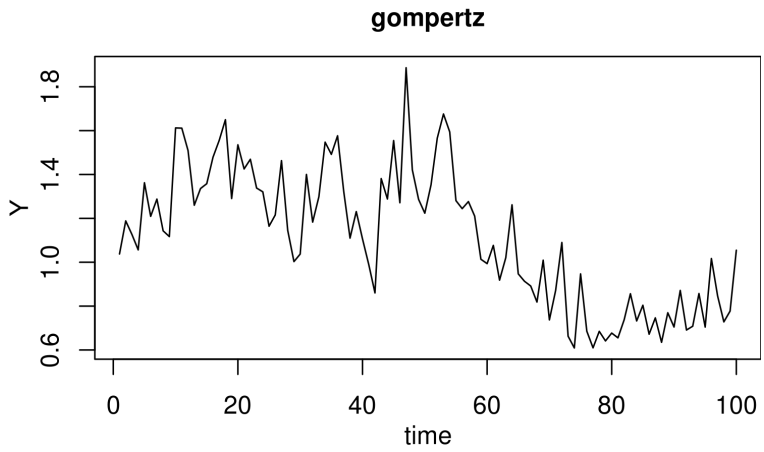

Figure 6.1: Simulated data from the Gompertz model (6.9, 6.10). This figure shows the result of executing plot(gompertz, variables = "Y").

```
R> gompertz <- simulate(gompertz, params = theta)
```

Now gompertz is identical to what it was before, except that the missing data have been replaced by simulated data. The parameter vector (theta) at which the simulations were performed has also been saved internally to gompertz. We can plot the simulated data via

```
R> plot(gompertz, variables = "Y")
```

6.1 shows the results of this operation.

6.4.2 Computing likelihood using SMC

As discussed in 6.3, some parameter estimation algorithms in the pomp package are doubly plug-and-play in that they require only rprocess and rmeasure. These include the nonlinear forecasting algorithm nlf, the probe-matching algorithm probe.match, and approximate Bayesian computation via abc. The plug-and-play full-information methods in pomp, however, require dmeasure, i.e., the ability to evaluate the likelihood of the data given the unobserved state. The gompertz.meas.dens above does

this, but we must fold it into the pomp object in order to use it. We can do this with another call to pomp:

```
R> gompertz <- pomp(gompertz, dmeasure = gompertz.meas.dens)
```

The result of the above is a new pomp object gompertz in every way identical to the one we had before, but with the measurement-model density function dmeasure now specified.

To estimate the likelihood of the data, we can use the function pfilter, an implementation of 2. We must decide how many concurrent realizations (*particles*) to use: the larger the number of particles, the smaller the Monte Carlo error but the greater the computational burden. Here, we run pfilter with 1000 particles to estimate the likelihood at the true parameters:

```
R> pf <- pfilter(gompertz, params = theta, Np = 1000)
R> loglik.truth <- logLik(pf)
R> loglik.truth
[1] 36.27102
```

Since the true parameters (i.e., the parameters that generated the data) are stored within the pomp object gompertz and can be extracted by the coef function, we could have done

```
R> pf <- pfilter(gompertz, params = coef(gompertz), Np = 1000)
```

or simply

```
R> pf <- pfilter(gompertz, Np = 1000)
```

Now let us compute the log likelihood at a different point in parameter space, one for which r, K, and σ are each 50% higher than their true values.

```
R> theta.guess <- theta.true <- coef(gompertz)
R> theta.guess[c("r", "K", "sigma")] <- 1.5 * theta.true[c("r", "K", "sigma")]
R> pf <- pfilter(gompertz, params = theta.guess, Np = 1000)
R> loglik.guess <- logLik(pf)
R> loglik.guess
[1] 25.19585
```

In this case, the Kalman filter computes the exact log likelihood at the true parameters to be 36.01, while the particle filter with 1000 particles gives 36.27. Since the particle filter gives an unbiased estimate of the likelihood, the difference is due to Monte Carlo error in the particle filter. One can reduce this error by using a larger number of particles and/or by re-running pfilter multiple times and averaging the resulting estimated likelihoods. The latter approach has the advantage of allowing one to estimate the Monte Carlo error itself; we will demonstrate this in 6.4.3.

6.4.3 Maximum likelihood estimation via iterated filtering

Let us use the iterated filtering approach described in 6.3.2 to obtain an approximate maximum likelihood estimate for the data in gompertz. Since the parameters of (6.9), (6.10) are constrained to be positive, when estimating, we transform them to a scale on which they are unconstrained. The following encodes such a transformation and its inverse:

```
R> gompertz.tf <- function(params, ...) exp(params)
R> gompertz.itf <- function(params, ...) log(params)
```

We add these to the existing pomp object via:

```
R> gompertz <- pomp(gompertz, parameter.transform = gompertz.tf,
+                   parameter.inv.transform = gompertz.itf)
```

The following codes initialize the iterated filtering algorithm at several starting points around theta.true and estimate the parameters r, τ, and σ.

```
R> estpars <- c("r", "sigma", "tau")
R> library("foreach")
R> mif1 <- foreach(i = 1:10, .combine = c) %dopar% {
+    theta.guess <- theta.true
+    rlnorm(n = length(estpars), meanlog = log(theta.guess[estpars]),
+           sdlog = 1) -> theta.guess[estpars]
+    mif(gompertz, Nmif = 100, start = theta.guess, transform = TRUE,
+        Np = 2000, var.factor = 2, cooling.fraction = 0.7,
+        rw.sd = c(r = 0.02, sigma = 0.02, tau = 0.02))
+  }
R> pf1 <- foreach(mf = mif1, .combine = c) %dopar% {
+    pf <- replicate(n = 10, logLik(pfilter(mf, Np = 10000)))
+    logmeanexp(pf)
+  }
```

Note that we have set transform = TRUE in the call to mif above: this causes the parameter transformations we have specified to be applied to enforce the positivity of parameters. Note also that we have used the foreach package (*Revolution Analytics and Weston*, 2014) to parallelize the computations.

Each of the 10 mif runs ends up at a different point estimate (6.2). We focus on that with the highest estimated likelihood, having evaluated the likelihood several times to reduce the Monte Carlo error in the likelihood evaluation. The particle filter produces an unbiased estimate of the likelihood; therefore, we will average the likelihoods, not the log likelihoods.

```
R> mf1 <- mif1[[which.max(pf1)]]
```

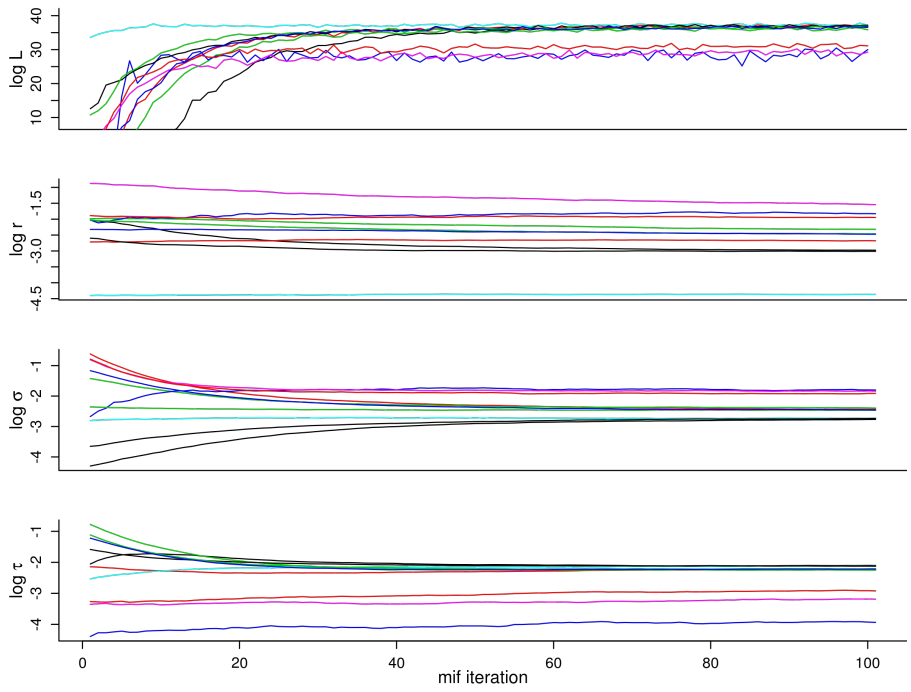

Figure 6.2: Convergence plots can be used to help diagnose convergence of the iterated filtering (IF) algorithm. These and additional diagnostic plots are produced when plot is applied to a mif or mifList object.

```
R> theta.mif <- coef(mf1)
R> loglik.mif <- replicate(n = 10, logLik(pfilter(mf1, Np = 10000)))
R> loglik.mif <- logmeanexp(loglik.mif, se = TRUE)
R> theta.true <- coef(gompertz)
R> loglik.true <- replicate(n = 10, logLik(pfilter(gompertz, Np = 20000)))
R> loglik.true <- logmeanexp(loglik.true, se = TRUE)
```

For the calculation above, we have replicated the iterated filtering search, made a careful estimation of the log likelihood, $\hat{\ell}$, and its standard error using pfilter at each of the resulting point estimates, and then chosen the parameter corresponding to the highest likelihood as our numerical approximation to the MLE. Taking advantage of the Gompertz model's tractability, we also use the Kalman filter to maximize the exact likelihood, ℓ, and evaluate it at the estimated MLE obtained by mif. The resulting estimates are shown in 6.3. Usually, the last row and column of 6.3 would

not be available even for a simulation study validating the inference methodology for a known POMP model. In this case, we see that the mif procedure is successfully maximizing the likelihood up to an error of about 0.1 log units.

6.4.4 Full-information Bayesian inference via PMCMC

To carry out Bayesian inference we need to specify a prior distribution on unknown parameters. The pomp constructor function provides the rprior and dprior arguments, which can be filled with functions that simulate from and evaluate the prior density, respectively. Methods based on random-walk Metropolis-Hastings require evaluation of the prior density (dprior), but not simulation (rprior), so we specify dprior for the Gompertz model as follows.

```
R> hyperparams <- list(min = coef(gompertz)/10, max = coef(gompertz) * 10)

R> gompertz.dprior <- function(params, ..., log) {
+    f <- sum(dunif(params, min = hyperparams$min, max = hyperparams$max,
+      log = TRUE))
+    if (log)
+      f else exp(f)
+  }
```

The PMCMC algorithm described in 6.3.3 can then be applied to draw a sample from the posterior. Recall that, for each parameter proposal, PMCMC pays the full price of a particle-filtering operation in order to obtain the Metropolis-Hastings acceptance probability. For the same price, iterated filtering obtains, in addition, an estimate of the derivative and a probable improvement of the parameters. For this reason, PMCMC is relatively inefficient at traversing parameter space. When Bayesian inference is the goal, it is therefore advisable to first locate a neighborhood of the MLE using, for example, iterated filtering. PMCMC can then be initialized in

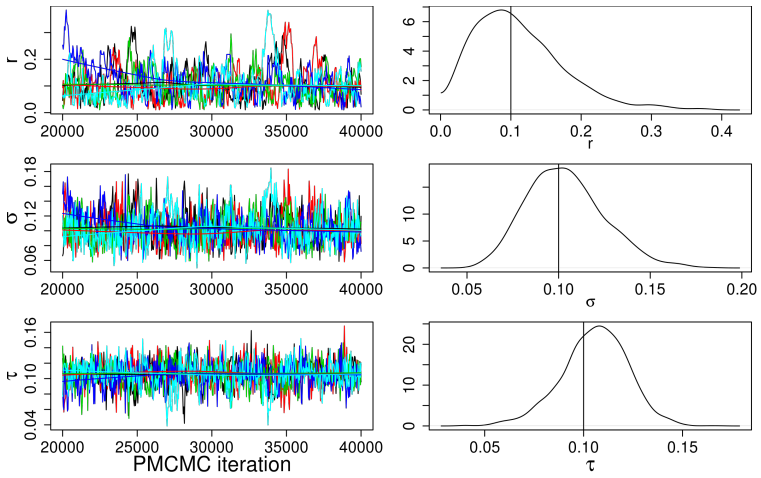

Figure 6.3: Diagnostic plots for the PMCMC algorithm. The trace plots in the left column show the evolution of 5 independent MCMC chains after a burn-in period of length 20000. Kernel density estimates of the marginal posterior distributions are shown at right. The effective sample size of the 5 MCMC chains combined is lowest for the r variable and is 180: the use of 40000 proposal steps in this case is a modest number. The density plots at right show the estimated marginal posterior distributions. The vertical line corresponds to the true value of each parameter.

this neighborhood to sample from the posterior distribution. The following adopts this approach, running 5 independent PMCMC chains using a multivariate normal random-walk proposal (with diagonal variance-covariance matrix, see ?mvn.diag.rw).

```
R> pmcmc1 <- foreach(i=1:5,.combine=c) %dopar% {
+    pmcmc(pomp(gompertz, dprior = gompertz.dprior), start = theta.mif,
+          Nmcmc = 40000, Np = 100, max.fail = Inf,
+          proposal=mvn.diag.rw(c(r = 0.01, sigma = 0.01, tau = 0.01)))
+  }
```

Comparison with the analysis of 6.4.3 reinforces the observation of *Bhadra* (2010) that PMCMC can require orders of magnitude more computation than iterated filtering. Iterated filtering may have to be repeated multiple times while computing profile likelihood plots, whereas one successful run of PMCMC is sufficient to obtain

all required posterior inferences. However, in practice, multiple runs from a range of starting points is always good practice since convergence cannot be reliably assessed on the basis of a single chain. To verify the convergence of the approach or to compare the performance with other approaches, we can use diagnostic plots produced by the plot command (see 6.3).

6.4.5 A second example: the Ricker model

In 6.4.6, we will illustrate probe matching (6.3.4) using a stochastic version of the Ricker map (*Ricker*, 1954). We switch models to allow direct comparison with *Wood* (2010), whose synthetic likelihood computations are reproduced below. In particular, the results of 6.4.6 demonstrate frequentist inference using synthetic likelihood and also show that the full likelihood is both numerically tractable and reasonably well behaved, contrary to the claim of *Wood* (2010). We will also take the opportunity to demonstrate features of pomp that allow acceleration of model codes through the use of R's facilities for compiling and dynamically linking C code.

The Ricker model is another discrete-time model for the size of a population. The population size, N_t, at time t is postulated to obey

$$N_{t+1} = r\,N_t\,\exp(-N_t + e_t), \qquad e_t \sim \text{Normal}\left(0, \sigma^2\right). \tag{6.12}$$

In addition, we assume that measurements, Y_t, of N_t are themselves noisy, according to

$$Y_t \sim \text{Poisson}(\phi\,N_t), \tag{6.13}$$

where ϕ is a scaling parameter. As before, we will need to implement the model's state-process simulator (rprocess). We have the option of writing these functions in R, as we did with the Gompertz model. However, we can realize manyfold speed-ups by writing these in C. In particular, pomp allows us to write snippets of C code that

it assembles, compiles, and dynamically links into a running R session. To begin the process, we will write snippets for the rprocess, rmeasure, and dmeasure components.

```
R> ricker.sim <- "
+     e = rnorm(0, sigma);
+     N = r * N * exp(-N + e);
+  "
R> ricker.rmeas <- "
+     y = rpois(phi * N);
+  "
R> ricker.dmeas <- "
+     lik = dpois(y, phi * N, give_log);
+  "
```

Note that, in this implementation, both N and e are state variables. The logical flag give-log requests the likelihood when FALSE, the log likelihood when TRUE. Notice that, in these snippets, we never declare the variables; pomp will construct the appropriate declarations automatically.

In a similar fashion, we can add transformations of the parameters to enforce constraints.

```
R> par.trans <- "
+     Tr = exp(r);
+     Tsigma = exp(sigma);
+     Tphi = exp(phi);
+     TN_0 = exp(N_0);
+  "
R> par.inv.trans <- "
+     Tr = log(r);
```

```
+     Tsigma = log(sigma);
+     Tphi = log(phi);
+     TN_0 = log(N_0);
+   "
```

Note that in the foregoing C snippets, the prefix T designates the transformed version of the parameter. A full set of rules for using Csnippets, including illustrative examples, is given in the package help system (?Csnippet).

Now we can construct a pomp object as before and fill it with simulated data:

```
R> pomp(data = data.frame(time = seq(0, 50, by = 1), y = NA),
+       rprocess = discrete.time.sim(step.fun = Csnippet(ricker.sim),
+         delta.t = 1), rmeasure = Csnippet(ricker.rmeas),
+       dmeasure = Csnippet(ricker.dmeas),
+       parameter.transform = Csnippet(par.trans),
+       parameter.inv.transform = Csnippet(par.inv.trans),
+       paramnames = c("r", "sigma", "phi", "N.0", "e.0"),
+       statenames = c("N", "e"), times = "time", t0 = 0,
+       params = c(r = exp(3.8), sigma = 0.3, phi = 10,
+         N.0 = 7, e.0 = 0)) -> ricker
R> ricker <- simulate(ricker,seed=73691676L)
```

6.4.6 Feature-based synthetic likelihood maximization

In pomp, probes are simply functions that can be applied to an array of real or simulated data to yield a scalar or vector quantity. Several functions that create useful probes are included with the package, including those recommended by *Wood* (2010). In this illustration, we will make use of these probes: probe.marginal, probe.acf, and probe.nlar. probe.marginal regresses the data against a sample from

a reference distribution; the probe's values are those of the regression coefficients. probe.acf computes the auto-correlation or auto-covariance of the data at specified lags. probe.nlar fits a simple nonlinear (polynomial) autoregressive model to the data; again, the coefficients of the fitted model are the probe's values. We construct a list of probes:

```
R> plist <- list(probe.marginal("y", ref = obs(ricker), transform = sqrt),
+                probe.acf("y", lags = c(0, 1, 2, 3, 4), transform = sqrt),
+                probe.nlar("y", lags = c(1, 1, 1, 2), powers = c(1, 2, 3, 1),
+                           transform = sqrt))
```

Each element of plist is a function of a single argument. Each of these functions can be applied to the data in ricker and to simulated data sets. Calling pomp's function probe results in the application of these functions to the data, and to each of some large number, nsim, of simulated data sets, and finally to a comparison of the two. [Note that probe functions may be vector-valued, so a single probe taking values in $\mathbb{R}^k$ formally corresponds to a collection of k probe functions in the terminology of 6.3.4.] Here, we will apply probe to the Ricker model at the true parameters and at a wild guess.

```
R> pb.truth <- probe(ricker, probes = plist, nsim = 1000, seed = 1066L)
R> guess <- c(r = 20, sigma = 1, phi = 20, N.0 = 7, e.0 = 0)
R> pb.guess <- probe(ricker, params = guess, probes = plist, nsim = 1000,
+    seed = 1066L)
```

Results summaries and diagnostic plots showing the model-data agreement and correlations among the probes can be obtained by

```
R> summary(pb.truth)
R> summary(pb.guess)
```

```
R> plot(pb.truth)
R> plot(pb.guess)
```

An example of a diagnostic plot (using a smaller set of probes) is shown in 6.4. Among the quantities returned by summary is the synthetic likelihood (6). One can attempt to identify parameters that maximize this quantity; this procedure is referred to in pomp as "probe matching". Let us now attempt to fit the Ricker model to the data using probe-matching.

```
R> pm <- probe.match(pb.guess, est = c("r", "sigma", "phi"), transform = TRUE,
+    method = "Nelder-Mead", maxit = 2000, seed = 1066L, reltol = 1e-08)
```

This code runs optim's Nelder-Mead optimizer from the starting parameters guess in an attempt to maximize the synthetic likelihood based on the probes in plist. Both the starting parameters and the list of probes are stored internally in pb.guess, which is why we need not specify them explicitly here. While probe.match provides substantial flexibility in choice of optimization algorithm, for situations requiring greater flexibility, pomp provides the function probe.match.objfun, which constructs an objective function suitable for use with arbitrary optimization routines.

By way of putting the synthetic likelihood in context, let us compare the results of estimating the Ricker model parameters using probe-matching and using iterated filtering (IF), which is based on likelihood. The following code runs 600 IF iterations starting at guess:

```
R> mf <- mif(ricker,start=guess,Nmif=100,Np=1000,transform=TRUE,
+            cooling.fraction=0.95^50,var.factor=2,ic.lag=3,
+            rw.sd=c(r=0.1,sigma=0.1,phi=0.1),max.fail=50)
R> mf <- continue(mf,Nmif=500,max.fail=20)
```

6.4 compares parameters, Monte Carlo likelihoods ($\hat{\ell}$), and synthetic likelihoods ($\hat{\ell}_{\mathbb{S}}$, based on the probes in plist) at each of (a) the guess, (b) the truth, (c) the

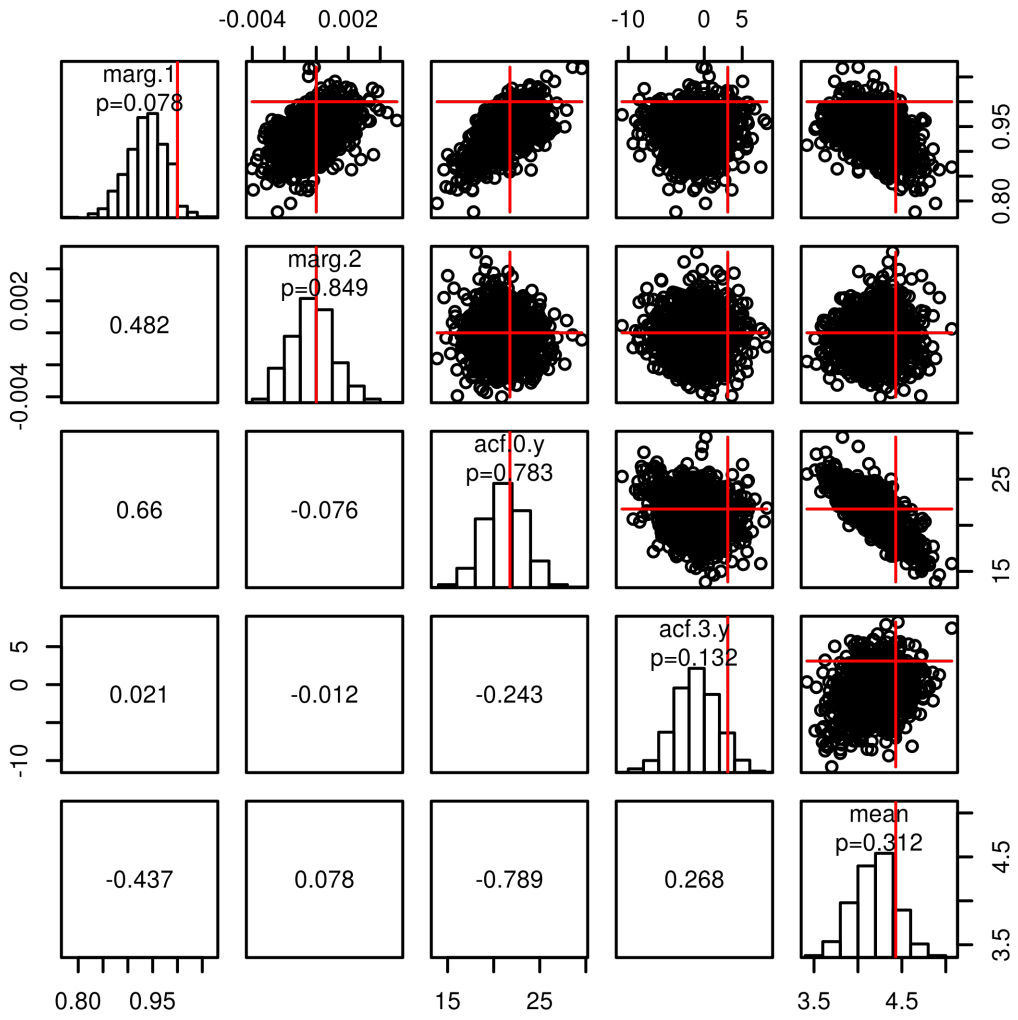

Figure 6.4: Results of plot on a probed.pomp-class object. Above the diagonal, the pairwise scatterplots show the values of the probes on each of 1000 data sets. The red lines show the values of each of the probes on the data. The panels along the diagonal show the distributions of the probes on the simulated data, together with their values on the data and a two-sided p value. The numbers below the diagonal are the Pearson correlations between the corresponding pairs of probes.

MLE from mif, and (d) the maximum synthetic likelihood estimate (MSLE) from probe.match. These results demonstrate that it is possible, and indeed not difficult, to maximize the likelihood for this model, contrary to the claim of *Wood* (2010). Since synthetic likelihood discards some of the information in the data, it is not surprising that 6.4 also shows the statistical inefficiency of maximum synthetic likelihood relative to that of likelihood.

6.4.7 Bayesian feature matching via ABC

Whereas synthetic likelihood carries out many simulations for each likelihood estimation, ABC (as described in 6.3.5) uses only one. Each iteration of ABC is therefore much quicker, essentially corresponding to the cost of SMC with a single particle or synthetic likelihood with a single simulation. A consequence of this is that ABC cannot determine a good relative scaling of the features within each likelihood evaluation and this must be supplied in advance. One can imagine an adaptive version of ABC which modifies the scaling during the course of the algorithm, but here we do a preliminary calculation to accomplish this. We return to the Gompertz model to faciliate comparison between ABC and PMCMC.

```
R> plist <- list(probe.mean(var = "Y", transform = sqrt),
+                probe.acf("Y", lags = c(0, 5, 10, 20)),
+                probe.marginal("Y", ref = obs(gompertz)))
+  psim <- probe(gompertz, probes = plist, nsim = 500)
+  scale.dat <- apply(psim$simvals, 2, sd)
R> abc1 <- foreach(i = 1:5, .combine = c) %dopar% {
+    abc(pomp(gompertz, dprior = gompertz.dprior), Nabc = 4e6,
+        probes = plist, epsilon = 2, scale = scale.dat,
+        proposal=mvn.diag.rw(c(r = 0.01, sigma = 0.01, tau = 0.01)))
+  }
```

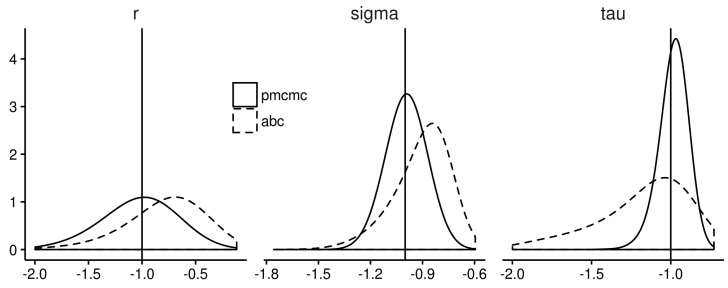

Figure 6.5: Marginal posterior distributions using full information via pmcmc (solid line) and partial information via abc (dashed line). Kernel density estimates are shown for the posterior marginal densities of $\log_{10}(r)$ (left panel), $\log_{10}(\sigma)$ (middle panel), and $\log_{10}(\tau)$ (right panel). The vertical lines indicate the true values of each parameter.

The effective sample size of the ABC chains is lowest for the r parameter (as was the case for PMCMC) and is 390, as compared to 180 for pmcmc in 6.4.4. The total computational effort allocated to abc here matches that for pmcmc since pmcmc used 100 particles for each likelihood evaluation but is awarded 100 times fewer Metropolis-Hastings steps. In this example, we conclude that abc mixes somewhat more rapidly (as measured by total computational effort) than pmcmc. 6.5 investigates the statistical efficiency of abc on this example. We see that abc gives rise to somewhat broader posterior distributions than the full-information posteriors from pmcmc As in all numerical studies of this kind, one cannot readily generalize from one particular example: even for this specific model and dataset, the conclusions might be sensitive to the algorithmic settings. However, one should be aware of the possibility of losing substantial amounts of information even when the features are based on reasoned scientific argument (*Shrestha et al.*, 2011; *Ionides*, 2011). Despite this loss of statistical efficiency, points B2–B5 of 6.3.4 identify situations in which ABC may be the only practical method available for Bayesian inference.

6.4.8 Parameter estimation by simulated quasi-likelihood

Within the pomp environment, it is fairly easy to try a quick comparison to see how nlf (6.3.6) compares with mif (6.3.2) on the Gompertz model. Carrying out a simulation study with a correctly specified POMP model is appropriate for assessing computational and statistical efficiency, but does not contribute to the debate on the role of two-step prediction criteria to fit misspecified models (*Xia and Tong*, 2011; *Ionides*, 2011). The nlf implementation we will use to compare to the mif call from 6.4.3 is

```
R> nlf1 <- nlf(gompertz, nasymp = 1000, nconverge = 1000, lags = c(2, 3),
+              start = c(r = 1, K = 2, sigma = 0.5, tau = 0.5, X.0 = 1),
+              est = c("r", "sigma", "tau"), transform = TRUE)
```

where the first argument is the pomp object, start is a vector containing model parameters at which nlf's search will begin, est contains the names of parameters nlf will estimate, and lags specifies which past values are to be used in the autoregressive model. The transform = TRUE setting causes the optimization to be performed on the transformed scale, as in 6.4.3. In the call above lags = c(2, 3) specifies that the autoregressive model predicts each observation, y_t using y_{t-2} and y_{t-3}, as recommended by *Kendall et al.* (2005). The quasi-likelihood is optimized numerically, so the reliability of the optimization should be assessed by doing multiple fits with different starting parameter values: the results of a small experiment (not shown) indicate that, on these simulated data, repeated optimization is not needed. nlf defaults to optimization by the subplex method (*Rowan*, 1990; *King*, 2008), though all optimization methods provided by optim are available as well. nasymp sets the length of the simulation on which the quasi-likelihood is based; larger values will give less variable parameter estimates, but will slow down the fitting process. The computational demand of nlf is dominated by the time required to generate the model

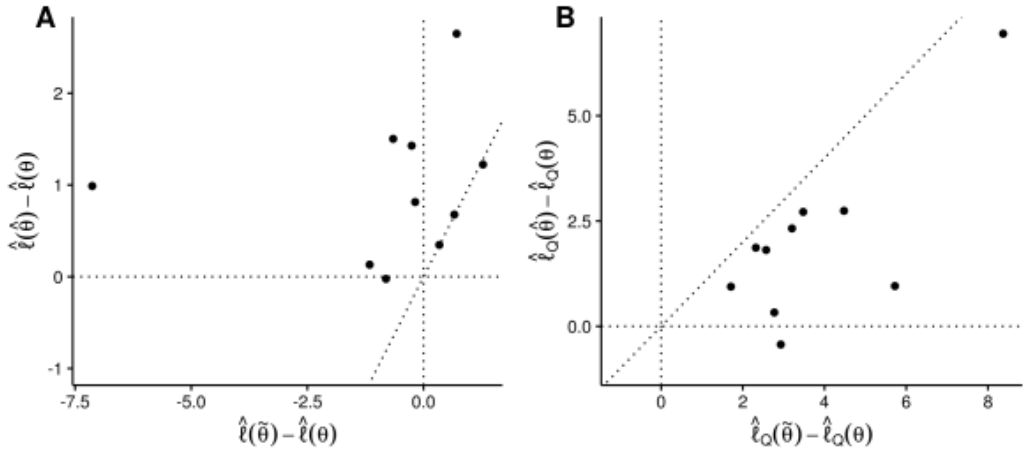

Figure 6.6: Comparison of mif and nlf for 10 simulated datasets using two criteria. In both plots, the maximum likelihood estimate (MLE), $\hat{\theta}$, obtained using iterated filtering is compared with the maximum simulated quasi-likelihood (MSQL) estimate, $\tilde{\theta}$, obtained using nonlinear forecasting. (A) Improvement in estimated log likelihood, $\hat{\ell}$, at point estimate over that at the true parameter value, θ. (B) Improvement in simulated log quasi-likelihood $\hat{\ell}_Q$, at point estimate over that at the true parameter value, θ. In both panels, the diagonal line is the 1-1 line.

simulations, so efficient coding of rprocess is worthwhile.

6.6 compares the true parameter, θ, with the maximum likelihood estimate (MLE), $\hat{\theta}$, from mif and the maximized simulated quasi-likelihood (MSQL), $\tilde{\theta}$, from nlf. 6.6A plots $\hat{\ell}(\hat{\theta}) - \hat{\ell}(\theta)$ against $\hat{\ell}(\tilde{\theta}) - \hat{\ell}(\theta)$, showing that the MSQL estimate can fall many units of log likelihood short of the MLE. 6.6B plots $\hat{\ell}_Q(\hat{\theta}) - \hat{\ell}_Q(\theta)$ against $\hat{\ell}_Q(\tilde{\theta}) - \hat{\ell}_Q(\theta)$, showing that likelihood-based inference is almost as good as nlf at optimizing the simulated quasi-likelihood criterion which nlf targets. 6.6 suggests that the MSQL may be inefficient, since it can give estimates with poor behavior according to the statistically efficient criterion of likelihood. Another possibility is that this particular implementation of nlf was unfortunate. Each mif optimization took 40.5 sec to run, compared to 8.2 sec for nlf, and it is possible that extra computer time or other algorithmic adjustments could substantially improve either or both estimators. It is hard

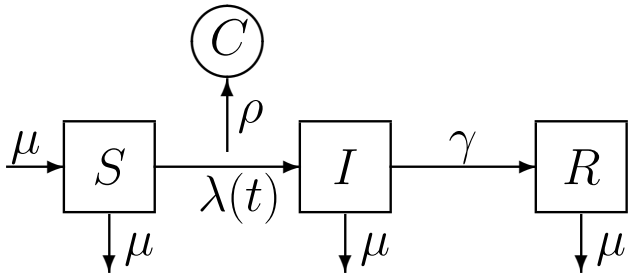

Figure 6.7: Diagram of the SIR epidemic model. The host population is divided into three classes according to infection status: S, susceptible hosts; I, infected (and infectious) hosts; R, recovered and immune hosts. Births result in new susceptibles and all individuals have a common death rate μ. Since the birth rate equals the death rate, the expected population size, $P = S + I + R$, remains constant. The S→I rate, λ, called the *force of infection*, depends on the number of infectious individuals according to $\lambda(t) = \beta\, I/N$. The I→R, or recovery, rate is γ. The case reports, C, result from a process by which new infections are recorded with probability ρ. Since diagnosed cases are treated with bed-rest and hence removed, infections are counted upon transition to R.

to ensure a fair comparison between methods, and in practice there is little substitute for some experimentation with different methods and algorithmic settings on a problem of interest. If the motivation for using NLF is preference for 2-step prediction in place of the likelihood, a comparison with SMC-based likelihood evaluation and maximization is useful to inform the user of the consequences of that preference.

6.5 A more complex example: epidemics in continuous time

6.5.1 A stochastic, seasonal SIR model.

A mainstay of theoretical epidemiology, the SIR model describes the progress of a contagious, immunizing infection through a population of hosts (*Kermack and McKendrick*, 1927; *Anderson and May*, 1991; *Keeling and Rohani*, 2009). The hosts are divided into three classes, according to their status vis-à-vis the infection (6.7). The susceptible class (S) contains those that have not yet been infected and are

thereby still susceptible to it; the infected class (I) comprises those who are currently infected and, by assumption, infectious; the removed class (R) includes those who are recovered or quarantined as a result of the infection. Individuals in R are assumed to be immune against reinfection. We let $S(t)$, $I(t)$, and $R(t)$ represent the numbers of individuals within the respective classes at time t.

It is natural to formulate this model as a continuous-time Markov process. In this process, the numbers of individuals within each class change through time in whole-number increments as discrete births, deaths, and passages between compartments occur. Let N_{AB} be the stochastic counting process whose value at time t is the number of individuals that have passed from compartment A to compartment B during the interval $[t_0, t)$, where t_0 is an arbitrary starting point not later than the first observation. We use the notation $N_{\bullet \mathrm{S}}$ to refer to births and $N_{\mathrm{A}\bullet}$ to refer to deaths from compartment A. Let us assume that the *per capita* birth and death rates, and the rate of transition, γ, from I to R are constants. The S to I transition rate, the so-called *force of infection*, $\lambda(t)$, however, should be an increasing function of $I(t)$. For many infections, it is reasonable to assume that the $\lambda(t)$ is jointly proportional to the fraction of the population infected and the rate at which an individual comes into contact with others. Here, we will make these assumptions, writing $\lambda(t) = \beta\, I(t)/P$, where β is the transmission rate and $P = S + I + R$ is the population size. We will go further and assume that birth and death rates are equal and independent of infection status; we will let μ denote the common rate. A consequence is that the expected population size remains constant.

The continuous-time Markov process is fully specified by the infinitesimal incre-

ment probabilities. Specifically, writing $\Delta N(t) = N(t+h) - N(t)$, we have

$$\begin{aligned}
\mathbb{P}\Delta N_{\bullet \mathrm{S}}(t)=1 \,|\, S(t), I(t), R(t) &= \mu\, P(t)\, h + o(h), \\
\mathbb{P}\Delta N_{\mathrm{SI}}(t)=1 \,|\, S(t), I(t), R(t) &= \lambda(t)\, S(t)\, h + o(h), \\
\mathbb{P}\Delta N_{\mathrm{IR}}(t)=1 \,|\, S(t), I(t), R(t) &= \gamma\, I(t)\, h + o(h), \\
\mathbb{P}\Delta N_{\mathrm{S}\bullet}(t)=1 \,|\, S(t), I(t), R(t) &= \mu\, S(t)\, h + o(h), \\
\mathbb{P}\Delta N_{\mathrm{I}\bullet}(t)=1 \,|\, S(t), I(t), R(t) &= \mu\, I(t)\, h + o(h), \\
\mathbb{P}\Delta N_{\mathrm{R}\bullet}(t)=1 \,|\, S(t), I(t), R(t) &= \mu\, R(t)\, h + o(h),
\end{aligned} \tag{6.14}$$

together with statement that all events of the form

$$\{\Delta N_{\mathrm{AB}}(t) > 1\} \qquad \text{and} \qquad \{\Delta N_{\mathrm{AB}}(t)=1, \Delta N_{\mathrm{CD}}(t)=1\}$$

for A, B, C, D with $(A, B) \neq (C, D)$ have probability $o(h)$. The counting processes are coupled to the state variables (*Bretó and Ionides*, 2011) via the following identities

$$\begin{aligned}
\Delta S(t) &= \Delta N_{\bullet \mathrm{S}}(t) - \Delta N_{\mathrm{SI}}(t) - \Delta N_{\mathrm{S}\bullet}(t), \\
\Delta I(t) &= \Delta N_{\mathrm{SI}}(t) - \Delta N_{\mathrm{IR}}(t) - \Delta N_{\mathrm{I}\bullet}(t), \\
\Delta R(t) &= \Delta N_{\mathrm{IR}}(t) - \Delta N_{\mathrm{R}\bullet}(t).
\end{aligned} \tag{6.15}$$

Taking expectations of (6.14),(6.15), dividing through by h, and taking a limit as $h \downarrow 0$, one obtains a system of nonlinear ordinary differential equations which is known as the deterministic skeleton of the model (*Coulson et al.*, 2004). Specifically, the SIR deterministic skeleton is

$$\begin{aligned}
\frac{dS}{dt} &= \mu\,(P - S) - \beta\,\frac{I}{P}\,S \\
\frac{dI}{dt} &= \beta\,\frac{I}{P}\,S - \gamma\, I - \mu\, I \\
\frac{dR}{dt} &= \gamma\, I - \mu\, R
\end{aligned} \tag{6.16}$$

It is typically impossible to monitor S, I, and R, directly. It sometimes happens, however, that public health authorities keep records of *cases*, i.e., individual infections. The number of cases, $C(t_1, t_2)$, recorded within a given reporting interval $[t_1, t_2)$ might perhaps be modeled by a negative binomial process

$$C(t_1, t_2) \sim \text{NegBin}(\rho\, \Delta N_{\text{SI}}(t_1, t_2), \theta) \tag{6.17}$$

where $\Delta N_{\text{SI}}(t_1, t_2)$ is the true incidence (the accumulated number of new infections that have occured over the $[t_1, t_2)$ interval), ρ is the *reporting rate*, (the probability that an infection is observed and recorded), θ is the negative binomial "size" parameter, and the notation is meant to indicate that $\mathbb{E}\left[C(t_1, t_2) \,|\, \Delta N_{\text{SI}}(t_1, t_2) = H\right] = \rho\, H$ and $\text{Var} C(t_1, t_2) \,|\, \Delta N_{\text{SI}}(t_1, t_2) = H = \rho\, H + \rho^2\, H^2/\theta$. The fact that the observed data are linked to an accumulation, as opposed to an instantaneous value, introduces a slight complication, which we discuss below.

6.5.2 Implementing the SIR model in pomp

As before, we will need to write functions to implement some or all of the SIR model's rprocess, rmeasure, and dmeasure components. As in 6.4.5, we will write these components using pomp's Csnippets. Recall that these are snippets of C code that pomp automatically assembles, compiles, and dynamically loads into the running R session.

To start with, we will write snippets that specify the measurement model (rmeasure and dmeasure):

```
R> rmeas <- "
+    cases = rnbinom_mu(theta, rho * H);
+  "
R> dmeas <- "
```

```
+    lik = dnbinom_mu(cases, theta, rho * H, give_log);
+  "
```

Here, we are using cases to refer to the data (number of reported cases) and H to refer to the true incidence over the reporting interval. The negative binomial simulator rnbinom_mu and density function dnbinom_mu are provided by R. The logical flag give_log requests the likelihood when FALSE, the log likelihood when TRUE. Notice that, in these snippets, we never declare the variables; pomp will ensure that the state variable (H), observable (cases), parameters (theta, rho), and likelihood (lik) are defined in the contexts within which these snippets are executed.

For the rprocess portion, we could simulate from the continuous-time Markov process exactly (*Gillespie*, 1977); the pomp function gillespie.sim implements this algorithm. However, for practical purposes, the exact algorithm is often prohibitively slow. If we are willing to live with an approximate simulation scheme, we can use the so-called "tau-leap" algorithm, one version of which is implemented in pomp via the euler.sim plug-in. This algorithm holds the transition rates λ, μ, γ constant over a small interval of time Δt and simulates the numbers of births, deaths, and transitions that occur over that interval. It then updates the state variables S, I, R accordingly, increments the time variable by Δt, recomputes the transition rates, and repeats. Naturally, as $\Delta t \to 0$, this approximation to the true continuous-time process becomes better and better. The critical feature from the inference point of view, however, is that no relationship need be assumed between the Euler simulation interval Δt and the reporting interval, which itself need not even be the same from one observation to the next.

Under the above assumptions, the number of individuals leaving any of the classes by all available routes over a particular time interval is a multinomial process. For example, if ΔN_{SI} and ΔN_{S} are the numbers of S individuals acquiring infection and

dying, respectively, over the Euler simulation interval $[t, t+\Delta t)$, then

$$(\Delta N_{\mathrm{SI}}, \Delta N_{\mathrm{S}}, S - \Delta N_{\mathrm{SI}} - \Delta N_{\mathrm{S}}) \sim \mathrm{Multinom}\left(S(t); p_{S\to I}, p_{S\to}, 1 - p_{S\to I} - p_{S\to}\right), \tag{6.18}$$

where

$$\begin{aligned} p_{S\to I} &= \frac{\lambda(t)}{\lambda(t)+\mu}\,\left(1 - e^{-(\lambda(t)+\mu)\,\Delta t}\right) \\ p_{S\to} &= \frac{\mu}{\lambda(t)+\mu}\,\left(1 - e^{-(\lambda(t)+\mu)\,\Delta t}\right). \end{aligned} \tag{6.19}$$

By way of shorthand, we say that the random variable $(\Delta N_{\mathrm{SI}}, \Delta N_{\mathrm{S}})$ in 6.18 has an *Euler-multinomial* distribution. Such distributions arise with sufficient frequency in compartmental models that pomp provides convenience functions for them. Specifically, the functions reulermultinom and deulermultinom respectively draw random deviates from, and evaluate the probability mass function of, such distributions. As the help pages relate, reulermultinom and deulermultinom parameterize the Euler-multinomial distributions by the size ($S(t)$ in 6.18), rates ($\lambda(t)$ and μ), and time interval Δt. Obviously, the Euler-multinomial distributions generalize to an arbitrary number of exit routes.

The help page (?euler.sim) informs us that to use euler.sim, we need to specify a function that advances the states from t to $t+\Delta t$. Again, we will write this in C to realize faster run-times:

```
R> sir.step <- "
+    double rate[6];
+    double dN[6];
+    double P;
+    P = S + I + R;
+    rate[0] = mu * P;       // birth
+    rate[1] = beta * I / P; // transmission
+    rate[2] = mu;           // death from S
```

```
+    rate[3] = gamma;        // recovery
+    rate[4] = mu;           // death from I
+    rate[5] = mu;           // death from R
+    dN[0] = rpois(rate[0] * dt);
+    reulermultinom(2, S, &rate[1], dt, &dN[1]);
+    reulermultinom(2, I, &rate[3], dt, &dN[3]);
+    reulermultinom(1, R, &rate[5], dt, &dN[5]);
+    S += dN[0] - dN[1] - dN[2];
+    I += dN[1] - dN[3] - dN[4];
+    R += dN[3] - dN[5];
+    H += dN[1];
+  "
```

As before, pomp will ensure that the undeclared state variables and parameters are defined in the context within which the snippet is executed. Note, however, that in the above we do declare certain local variables. In particular, the rate and dN arrays hold the rates and numbers of transition events, respectively. Note too, that we make use of pomp's C interface to reulermultinom, documented in the package help pages (?reulermultinom). The package help system (?Csnippet) includes instructions for, and examples of, the use of Csnippets.

Two significant wrinkles remains to be explained. First, notice that in sir.step, the variable H simply accumulates the numbers of new infections: H is a counting process that is nondecreasing with time. In fact, the incidence within an interval $[t_1, t_2)$ is $\Delta N_{\mathrm{SI}}(t_1, t_2) = \mathtt{H}(t_2) - \mathtt{H}(t_1)$. This leads to a technical difficulty with the measurement process, however, in that the data are assumed to be records of new infections occurring within the latest reporting interval, while the process model tracks the accumulated number of new infections since time t_0. We can get around this difficulty by re-setting H to zero immediately after each observation. We cause pomp

to do this via the pomp function's zeronames argument, as we will see in a moment. The section on "accumulator variables" in the pomp help page (?pomp) discusses this in more detail.

The second wrinkle has to do with the initial conditions, i.e., the states $S(t_0)$, $I(t_0)$, $R(t_0)$. By default, pomp will allow us to specify these initial states arbitrarily. For the model to be consistent, they should be positive integers that sum to the population size N. We can enforce this constraint by customizing the parameterization of our initial conditions. We do this in by furnishing a custom initializer in the call to pomp. Let us construct it now and fill it with simulated data.

```
R> pomp(data = data.frame(cases = NA, time = seq(0, 10, by=1/52)),
+       times = "time", t0 = -1/52, dmeasure = Csnippet(dmeas),
+       rmeasure = Csnippet(rmeas), rprocess = euler.sim(
+         step.fun = Csnippet(sir.step), delta.t = 1/52/20),
+       statenames = c("S", "I", "R", "H"),
+       paramnames = c("gamma", "mu", "theta", "beta", "popsize",
+         "rho", "S.0", "I.0", "R.0"), zeronames=c("H"),
+       initializer=function(params, t0, ...) {
+         fracs <- params[c("S.0", "I.0", "R.0")]
+         setNames(c(round(params["popsize"]*fracs/sum(fracs)),0),
+                  c("S","I","R","H"))
+       }, params = c(popsize = 500000, beta = 400, gamma = 26,
+            mu = 1/50, rho = 0.1, theta = 100, S.0 = 26/400,
+            I.0 = 0.002, R.0 = 1)) -> sir1
R> simulate(sir1, seed = 1914679908L) -> sir1
```

Notice that we are assuming here that the data are collected weekly and use an Euler step-size of 1/20 wk. Here, we have assumed an infectious period of 2 wk ($1/\gamma = 1/26$ yr) and a basic reproductive number, R_0 of $\beta/(\gamma+\mu) \approx 15$. We have

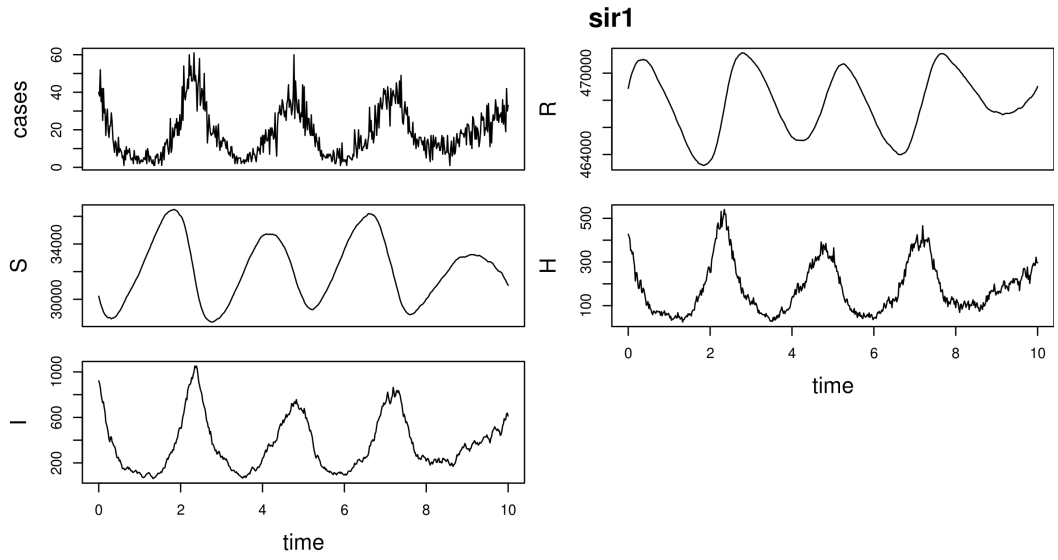

Figure 6.8: Result of plot(sir1). The pomp object sir1 contains the SIR model with simulated data.

assumed a host population size of 500,000 and 10% reporting efficiency. 6.8 shows one realization of this process.

Algorithm 4: Iterated filtering: `mif(P, start=`θ_0`, Nmif=`M`, Np=`J`, rw.sd=`$\sigma_{1:p}$`, ic.lag=`L`, var.factor=`C`, cooling.factor=`a`)`, using notation from 6.1 where P is a pomp object with defined rprocess, dmeasure, init.state, and obs components.

input: Starting parameter, θ_0; simulator for $f_{X_0}(x_0\,;\theta)$; simulator for $f_{X_n|X_{n-1}}(x_n\,|\,x_{n-1}\,;\theta)$; evaluator for $f_{Y_n|X_n}(y_n\,|\,x_n\,;\theta)$; data, $y^*_{1:N}$; labels, $I\subset\{1,\dots,p\}$, designating IVPs; fixed lag, L, for estimating IVPs; number of particles, J, number of iterations, M; cooling rate, $0<a<1$; perturbation scales, $\sigma_{1:p}$; initial scale multiplier, $C>0$.

for m *in* $1{:}M$ **do**
  Initialize parameters: $[\Theta^F_{0,j}]_i \sim \text{Normal}\left([\theta_0]_i, (Ca^{m-1}\sigma_i)^2\right)$ for i in $1{:}p$, j in $1:J$.
  Initialize states: simulate $X^F_{0,j}\sim f_{X_0}\big(\,\cdot\,;\Theta^F_{0,j}\big)$ for j in $1{:}J$.
  Initialize filter mean for parameters: $\bar\theta_0=\theta_0$.
  for n *in* $1:N$ **do**
    Perturb parameters: $\big[\Theta^P_{n,j}\big]_i \sim \text{Normal}\left(\big[\Theta^F_{n-1,j}\big]_i, (a^{m-1}\sigma_i)^2\right)$ for $i\notin I$, j in $1:J$.
    Simulate prediction particles: $X^P_{n,j}\sim f_{X_n|X_{n-1}}\big(\,\cdot\,|X^F_{n-1,j}\,;\Theta^P_{n,j}\big)$ for j in $1:J$.
    Evaluate weights: $w(n,j)=f_{Y_n|X_n}(y^*_n|X^P_{n,j}\,;\Theta^P_{n,j})$ for j in $1:J$.
    Normalize weights: $\tilde w(n,j)=w(n,j)/\sum_{u=1}^J w(n,u)$.
    Apply 3 to select indices $k_{1:J}$ with $\mathbb{P}k_u=j=\tilde w\,(n,j)$.
    Resample particles: $X^F_{n,j}=X^P_{n,k_j}$ and $\Theta^F_{n,j}=\Theta^P_{n,k_j}$ for j in $1:J$.
    Filter mean: $\big[\bar\theta_n\big]_i=\sum_{j=1}^J\tilde w(n,j)\big[\Theta^P_{n,j}\big]_i$ for $i\notin I$.
    Prediction variance: $[\bar V_{n+1}]_i=(a^{m-1}\sigma_i)^2+\sum_j\tilde w(n,j)\big([\Theta^F_{n,j}]_i-[\bar\theta_n]_i\big)^2$ for $i\notin I$.
  end
  Update non-IVP parameters: $[\theta_m]_i=[\theta_{m-1}]_i+V_1^i\sum_{n=1}^N\left(V_n^i\right)^{-1}\left(\bar\theta^i_n-\bar\theta^i_{n-1}\right)$ for $i\notin I$.
  Update IVPs: $[\theta_m]_i=\frac{1}{J}\sum_j\big[\Theta^F_{L,j}\big]_i$ for $i\in I$.
end

output: Monte Carlo maximum likelihood estimate, θ_M.

complexity: $\mathcal{O}(JM)$

Algorithm 5: Particle Markov Chain Monte Carlo: `pmcmc(P, start=`θ_0`, Nmcmc=`M`, Np=`J`, proposal=`q`)`, using notation from 6.1 where P is a pomp object with defined methods for rprocess, dmeasure, init.state, dprior, and obs. The supplied proposal samples from a symmetric, but otherwise arbitrary, MCMC proposal distribution, $q(\theta^P \,|\, \theta)$.

input: Starting parameter, θ_0; simulator for $f_{X_0}(x_0 \,|\, \theta)$; simulator for $f_{X_n|X_{n-1}}(x_n \,|\, x_{n-1}\,;\theta)$; evaluator for $f_{Y_n|X_n}(y_n \,|\, x_n\,;\theta)$; simulator for $q(\theta^P \,|\, \theta)$; data, $y^*_{1:N}$; number of particles, J; number of filtering operations, M; perturbation scales, $\sigma_{1:p}$; evaluator for prior, $f_\Theta(\theta)$.

1 Initialization: compute $\hat{\ell}(\theta_0)$ using 2 with J particles.

2 **for** m *in* $1{:}M$ **do**

3 Draw a parameter proposal, θ^P_m, from the proposal distribution: $\Theta^P_m \sim q\,(\,\cdot\,|\,\theta_{m-1})$.

4 Compute $\hat{\ell}(\theta^P_m)$ using 2 with J particles.

5 Generate $U \sim \text{Uniform}(0,1)$.

6 Set $\big(\theta_m, \hat{\ell}(\theta_m)\big) = \begin{cases} \big(\theta^P_m, \hat{\ell}(\theta^P_m)\big), & \text{if } U < \dfrac{f_\Theta(\theta^P_m)\exp(\hat{\ell}(\theta^P_m))}{f_\Theta(\theta_{m-1})\exp(\hat{\ell}(\theta_{m-1}))}, \\ \big(\theta_{m-1}, \hat{\ell}(\theta_{m-1})\big), & \text{otherwise.} \end{cases}$

7 **end**

output: Samples, $\theta_{1:M}$, representing the posterior distribution, $f_{\Theta|Y_{1:N}}(\theta \,|\, y^*_{1:N})$.

complexity: $\mathcal{O}(JM)$

Algorithm 6: Synthetic likelihood evaluation: `probe(P, nsim = `J`, probes = `$\mathbb{S}$`)`, using notation from 6.1 where P is a pomp object with defined methods for rprocess, rmeasure, init.state, and obs.

input: Simulator for $f_{X_n|X_{n-1}}(x_n \,|\, x_{n-1}\,;\theta)$; simulator for $f_{X_0}(x_0\,;\theta)$; simulator for $f_{Y_n|X_n}(y_n \,|\, x_n\,;\theta)$; parameter, θ; data, $y^*_{1:N}$; number of simulations, J; vector of summary statistics or *probes*, $\mathbb{S} = (\mathbb{S}_1, \ldots, \mathbb{S}_d)$.

1 Compute observed probes: $s^*_i = \mathbb{S}_i(y^*_{1:N})$ for i in $1{:}d$.

2 Simulate J datasets: $Y^j_{1:N} \sim f_{Y_{1:N}}(\,\cdot\,;\theta)$ for j in $1{:}J$.

3 Compute simulated probes: $s_{ij} = \mathbb{S}_i(Y^j_{1:N})$ for i in $1:d$ and j in $1{:}J$.

4 Compute sample mean: $\mu_i = J^{-1}\sum_{j=1}^{J} s_{ij}$ for i in $1{:}d$.

5 Compute sample covariance: $\Sigma_{ik} = (J-1)^{-1}\sum_{j=1}^{J}(s_{ij}-\mu_i)(s_{kj}-\mu_k)$ for i and k in $1{:}d$.

6 Compute the log synthetic likelihood:

$$\hat{\ell}_{\mathbb{S}}(\theta) = -\frac{1}{2}\,(s^*-\mu)^T\Sigma^{-1}(s^*-\mu) - \frac{1}{2}\,\log|\Sigma| - \frac{d}{2}\,\log(2\pi). \tag{6.5}$$

output: Synthetic likelihood, $\hat{\ell}_{\mathbb{S}}(\theta)$.
complexity: $\mathcal{O}(J)$

Algorithm 7: Approximate Bayesian Computation: `abc(P, start=`θ_0`, Nmcmc=`M`, probes=`$\mathbb{S}$`, scale=`$\tau_{1:d}$`, proposal=`q`, epsilon=`ϵ`)`, using notation from 6.1, where P is a pomp object with defined methods for rprocess, rmeasure, init.state, dprior, and obs.

input: Starting parameter, θ_0; simulator for $f_{X_0}(x_0\,;\theta)$; simulator for $f_{X_n|X_{n-1}}(x_n\,|\,x_{n-1}\,;\theta)$; simulator for $f_{Y_n|X_n}(y_n\,|\,x_n\,;\theta)$; simulator for $q(\theta^P\,|\,\theta)$; data, $y^*_{1:N}$; number of proposals, M; vector of probes, $\mathbb{S}=(\mathbb{S}_1,\dots,\mathbb{S}_d)$; perturbation scales, $\sigma_{1:p}$; evaluator for prior, $f_\Theta(\theta)$; feature scales, $\tau_{1:d}$; tolerance, ϵ.

Compute observed probes: $s^*_i=\mathbb{S}_i(y^*_{1:N})$ for i in $1:d$.

for m *in* $1:M$ **do**

  Draw a parameter proposal, θ^P_m, from the proposal distribution: $\Theta^P_m\sim q\,(\,\cdot\,|\,\theta_{m-1})$.

  Simulate dataset: $Y_{1:N}\sim f_{Y_{1:N}}(\,\cdot\,;\theta^P_m)$.

  Compute simulated probes: $s_i=\mathbb{S}_i(Y_{1:N})$ for i in $1:d$.

  Generate $U\sim\text{Uniform}(0,1)$.

  Set $\theta_m=\begin{cases}\theta^P_m, & \text{if } \displaystyle\sum_{i=1}^{d}\left(\frac{s_i-s^*_i}{\tau_i}\right)^2<\epsilon^2 \text{ and } U<\frac{f_\Theta(\theta^P_m)}{f_\Theta(\theta_{m-1})},\\ \theta_{m-1}, & \text{otherwise.}\end{cases}$

end

output: Samples, $\theta_{1:M}$, representing the posterior distribution, $f_{\Theta|S_{1:d}}(\theta\,|\,s^*_{1:d})$.

complexity: Nominally $\mathcal{O}(M)$, but performance will depend on the choice of ϵ, τ_i, and σ_i, as well as on the choice of probes $\mathbb{S}$.

Algorithm 8: Simulated quasi log likelihood for NLF. Pseudocode for the quasi-likelihood function optimized by `nlf(P, start=`θ_0`, nasymp=`J`, nconverge=`B`, nrbf=`K`, lags=`$c_{1:L}$`)`. Using notation from 6.1, P is a pomp object with defined methods for rprocess, rmeasure, init.state, and obs.

input: Simulator for $f_{X_n|X_{n-1}}(x_n \,|\, x_{n-1}\,;\theta)$; simulator for $f_{X_0}(x_0\,;\theta)$; simulator for $f_{Y_n|X_n}(y_n \,|\, x_n\,;\theta)$; parameter, θ; data, $y^*_{1:N}$; collection of lags, $c_{1:L}$; length of discarded transient, B; length of simulation, J; number of radial basis functions, K.

1 Simulate long stationary time series: $Y_{1:(B+J)} \sim f_{Y_{1:(B+J)}}(\,\cdot\,;\theta)$.

2 Set $Y_{\min} = \min\{Y_{(B+1):(B+J)}\}$, $Y_{\max} = \max\{Y_{(B+1):(B+J)}\}$ and $R = Y_{\max} - Y_{\min}$.

3 Locations for basis functions: $m_k = Y_{\min} + R \times [1.2 \times (k-1)(K-1)^{-1} - 0.1]$ for k in $1\!:\!K$.

4 Scale for basis functions: $s = 0.3 \times R$.

5 Define radial basis functions: $f_k(x) = \exp\{(x-m_k)^2/2s^2\}$ for k in $1\!:\!K$.

6 Define prediction function:
$H(y_{n-c_1}, y_{n-c_2}, \ldots, y_{n-c_L}) = \sum_{j=1}^{L}\sum_{k=1}^{K} a_{jk} f_k(y_{n-c_j})$.

7 Compute $\{a_{jk} : j \in 1\!:\!L, k \in 1\!:\!K\}$ to minimize

$$\hat{\sigma}^2 = \frac{1}{J}\sum_{n=B+1}^{B+J} \big[Y_n - H(Y_{n-c_1}, Y_{n-c_2}, \ldots, Y_{n-c_L})\big]^2. \tag{6.7}$$

8 Compute the simulated quasi log likelihood:

$$\hat{\ell}_Q(\theta) = -\frac{N-\overline{c}}{2}\log 2\pi\hat{\sigma}^2 - \sum_{n=1+\overline{c}}^{N} \frac{\big[y^*_n - H(y^*_{n-c_1}, y^*_{n-c_2}, \ldots, y^*_{n-c_L})\big]^2}{2\hat{\sigma}^2}, \tag{6.8}$$

where $\overline{c} = \max(c_{1:L})$.

output: Simulated quasi log likelihood, $\hat{\ell}_Q(\theta)$.

complexity: $\mathcal{O}(B) + \mathcal{O}(J)$

Table 6.3: Results of estimating parameters r, σ, and τ of the Gompertz model (6.9,6.10) by maximum likelihood using iterated filtering (4), compared with the exact MLE and with the true value of the parameter. The first three columns show the estimated values of the three parameters. The next two columns show the log likelihood, $\hat{\ell}$, estimated by SMC (2) and its standard error, respectively. The exact log likelihood, ℓ, is shown in the rightmost column. An ideal likelihood-ratio 95% confidence set, not usually computationally available, includes all parameters having likelihood within qchisq(0.95,df=3)/2 = 3.91 of the exact MLE. We see that both the mif MLE and the truth are in this set. In this example, the mif MLE is close to the exact MLE, so it is reasonable to expect that profile likelihood confidence intervals and likelihood ratio tests constructed using the mif MLE have statistical properties similar to those based on the exact MLE.

	r	σ	τ	$\hat{\ell}$	s.e.	ℓ
truth	0.1000	0.1000	0.1000	36.02	0.07	36.01
mif MLE	0.0127	0.0655	0.1200	37.61	0.08	37.62
exact MLE	0.0322	0.0694	0.1170	37.87	0.05	37.88

Table 6.4: Parameter estimation by means of maximum synthetic likelihood (6) vs. by means of maximum likelihood via iterated filtering (4). The row labeled "guess" contains the point at which both algorithms were initialized. That labeled "truth" contains the true parameter value, i.e., that at which the data were generated. The rows labeled "MLE" and "MSLE" show the estimates obtained using iterated filtering and maximum synthetic likelihood, respectively. Parameters r, σ, and τ were estimated; all others were held at their true values. The columns labeled $\hat{\ell}$ and $\hat{\ell}_{\mathbb{S}}$ are the Monte Carlo estimates of the log likelihood and the log synthetic likelihood, respectively; their Monte Carlo standard errors are also shown. While likelihood maximization results in an estimate for which both $\hat{\ell}$ and $\hat{\ell}_{\mathbb{S}}$ exceed their values at the truth, the value of $\hat{\ell}$ at the MSLE is smaller than at the truth, an indication of the relative statistical inefficiency of maximum synthetic likelihood.

	r	σ	ϕ	$\hat{\ell}$	s.e.$(\hat{\ell})$	$\hat{\ell}_{\mathbb{S}}$	s.e.$(\hat{\ell}_{\mathbb{S}})$
guess	20.0	1.000	20.0	-230.8	0.24	-5.6	0.50
truth	44.7	0.300	10.0	-139.0	0.10	17.7	0.09
MLE	45.0	0.186	10.2	-137.2	0.11	18.0	0.12
MSLE	42.1	0.337	11.3	-145.7	0.11	19.4	0.06

6.5.3 Complications: seasonality, imported infections, extra-demographic stochasticity.

To illustrate the flexibility afforded by pomp's plug-and-play methods, let us add a bit of real-world complexity to the simple SIR model. We will modify the model to take four facts into account:

1. For many infections, the contact rate is *seasonal*: $\beta = \beta(t)$ varies in more or less periodic fashion with time.

2. The host population may not be truly closed: *imported infections* arise when infected individuals visit the host population and transmit.

3. The host population need not be constant in size. If we have data, for example, on the numbers of births occurring in the population, we can incorporate this directly into the model.

4. Stochastic fluctuation in the rates λ, μ, and γ can give rise to *extrademographic stochasticity*, i.e., random process variability beyond the purely demographic stochasticity we have included so far.

To incorporate seasonality, we would like to assume a flexible functional form for $\beta(t)$. Here, we will use a three-coefficient Fourier series:

$$\log \beta(t) = b_0 + b_1 \cos 2\pi t + b_2 \sin 2\pi t. \tag{6.20}$$

There are a variety of ways to account for imported infections. Here, we will simply assume that there is some constant number, ι, of infected hosts visiting the population. Putting this together with the seasonal contact rate results in a force of infection $\lambda(t) = \beta(t)\ (I(t) + \iota)\ /N$.

To incorporate birth-rate information, let us suppose we have data on the number of births occurring each month in this population and that these data are in the

form of a data frame birthdat with columns time and births. We can incorporate the varying birth rate into our model by passing it as a covariate to the simulation code. When we pass birthdat as the covar argument to pomp, we cause a look-up table to be created and made available to the simulator. The package employs linear interpolation to provide a value of each variable in the covariate table at any requisite time: from the user's perspective, a variable births will simply be available for use by the model codes.

Finally, we can allow for extrademographic stochasticity by allowing the force of infection to be itself a random variable. We will accomplish this by assuming a random phase in β:

$$\lambda(t) = \left(\beta(\Phi(t))\,\frac{I(t)+\iota}{N}\right) \tag{6.21}$$

where the phase Φ satisfies the stochastic differential equation

$$d\Phi = dt + \sigma\, dW_t, \tag{6.22}$$

where $dW(t)$ is a white noise, specifically an increment of standard Brownian motion. This model assumption attempts to capture variability in the timing of seasonal changes in transmission rates. As σ varies, it can represent anything from a very mild modulation of the timing of the seasonal progression to much more intense variation.

Let us modify the process-model simulator to incorporate these complexities.

```
R> seas.sir.step <- "
+    double rate[6];
+    double dN[6];
+    double Beta;
+    double dW;
+    Beta = exp(b1 + b2 * cos(M_2PI * Phi) + b3 * sin(M_2PI * Phi));
```

```
+    rate[0] = births;                 // birth
+    rate[1] = Beta * (I + iota) / P; // infection
+    rate[2] = mu;                     // death from S
+    rate[3] = gamma;                  // recovery
+    rate[4] = mu;                     // death from I
+    rate[5] = mu;                     // death from R
+    dN[0] = rpois(rate[0] * dt);
+    reulermultinom(2, S, &rate[1], dt, &dN[1]);
+    reulermultinom(2, I, &rate[3], dt, &dN[3]);
+    reulermultinom(1, R, &rate[5], dt, &dN[5]);
+    dW = rnorm(dt, sigma * sqrt(dt));
+    S += dN[0] - dN[1] - dN[2];
+    I += dN[1] - dN[3] - dN[4];
+    R += dN[3] - dN[5];
+    P = S + I + R;
+    Phi += dW;
+    H += dN[1];
+    noise += (dW - dt) / sigma;
+  "
R> pomp(sir1, rprocess = euler.sim(
+    step.fun = Csnippet(seas.sir.step), delta.t = 1/52/20),
+    dmeasure = Csnippet(dmeas), rmeasure = Csnippet(rmeas),
+    covar = birthdat, tcovar = "time", zeronames = c("H", "noise"),
+    statenames = c("S", "I", "R", "H", "P", "Phi", "noise"),
+    paramnames = c("gamma", "mu", "popsize", "rho","theta","sigma",
+                   "S.0", "I.0", "R.0", "b1", "b2", "b3", "iota"),
+    initializer = function(params, t0, ...) {
```

```
+      fracs <- params[c("S.0", "I.0", "R.0")]
+      setNames(c(round(params["popsize"]*c(fracs/sum(fracs),1)),0,0,0),
+               c("S","I","R","P","H","Phi","noise"))
+    }, params = c(popsize = 500000, iota = 5, b1 = 6, b2 = 0.2,
+                  b3 = -0.1, gamma = 26, mu = 1/50, rho = 0.1, theta = 100,
+                  sigma = 0.3, S.0 = 0.055, I.0 = 0.002, R.0 = 0.94)) -> sir2
R> simulate(sir2, seed = 619552910L) -> sir2
```

6.9 shows the simulated data and latent states. The sir2 object we have constructed here contains all the key elements of models used within the pomp to investigate cholera (*King et al.*, 2008), measles (*He et al.*, 2010), malaria (*Bhadra et al.*, 2011), pertussis (*Blackwood et al.*, 2013a; *Lavine et al.*, 2013a), pneumococcal pneumonia (*Shrestha et al.*, 2013), rabies (*Blackwood et al.*, 2013b), and Ebola virus disease (*King et al.*, in press).

6.6 Conclusion

The pomp package is designed to be both a tool for data analysis based on POMP models and a sound platform for the development of inference algorithms. The model specification language provided by pomp is very general. Implementing a POMP model in pomp makes a wide range of inference algorithms available. Moreover, the separation of model from inference algorithm facilitates objective comparison of alternative models and methods. The examples demonstrated in this paper are relatively simple, but the package has been instrumental in a number of scientific studies (e.g., *King et al.*, 2008; *Bhadra et al.*, 2011; *Shrestha et al.*, 2011; *Earn et al.*, 2012b; *Roy et al.*, 2013; *Shrestha et al.*, 2013; *Blackwood et al.*, 2013a,b; *Lavine et al.*, 2013a; *Bretó*, 2014; *King et al.*, in press).

As a development platform, pomp is particularly convenient for implementing

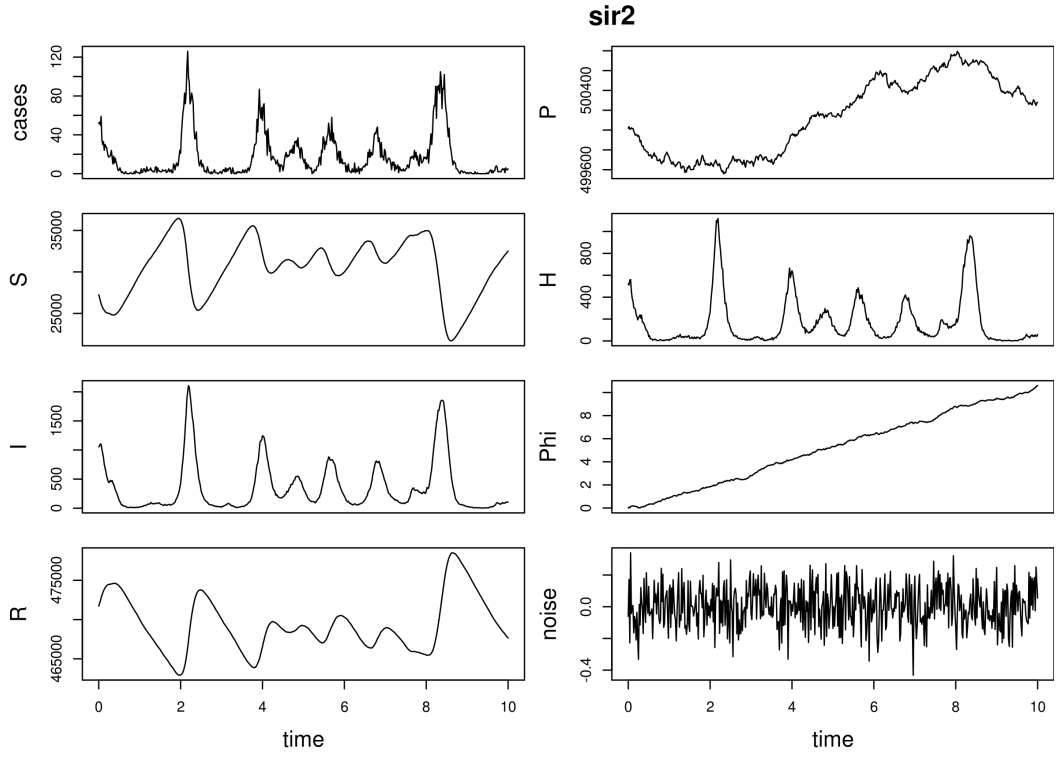

Figure 6.9: One realization of the SIR model with seasonal contact rate, imported infections, and extrademographic stochasticity in the force of infection.

algorithms with the plug-and-play property, since models will typically be defined by their rprocess simulator, together with rmeasure and often dmeasure, but can accommodate inference methods based on other model components such as dprocess and skeleton (the deterministic skeleton of the latent process). As an open-source project, the package readily supports expansion, and the authors invite community participation in the pomp project in the form of additional inference algorithms, improvements and extensions of existing algorithms, additional model/data examples, documentation contributions and improvements, bug reports, and feature requests.

Complex models and large datasets can challenge computational resources. With this in mind, key components of the pomp are written in C, and pomp provides facilities for users to write models either in R or, for the acceleration that typically proves necessary in applications, in C. Multi-processor computing also becomes necessary for ambitious projects. The two most common computationally intensive tasks are assessment of Monte Carlo variability and investigation of the role of starting values and other algorithmic settings on optimization routines. These analyses require only embarrassingly parallel computations and need no special discussion here.

The package contains more examples (via pompExamples), which can be used as templates for implementation of new models; the R and C code underlying these examples is provided with the package. In addition, pomp provides a number of interactive demos (via demo). Further documentation and an introductory tutorial are provided with the package and on the pomp website, `http://pomp.r-forge.r-project.org`.

APPENDICES

APPENDIX A

Supplements of chapters III and IV

A.1 Weak convergence for occupation measures

We study the convergence of the processes $\{W_\sigma(t), 0 \leq t \leq 1\}$ toward $\{W(t), 0 \leq t \leq 1\}$ as $\sigma \to 0$ for Theorem 2. We are interested in showing that the fraction of time $\{W_\sigma(t)\}$ spends in a set $\Theta_0 \subset \Theta$ over the discrete set of times $\{k\sigma^2, k = 1, \ldots, 1/\sigma^2\}$ converges in distribution to the fraction of time $\{W(t)\}$ spends in Θ_0. We choose $\{W_\sigma(t)\}$ to be a right-continuous step function approximation to a diffusion to simplify the relationship between the occupancy fraction over the discrete set of times and over the continuous interval. However, this simplification requires us to work with convergence to $\{W(t)\}$ in a space of processes with discontinuous sample paths, leading us to work with a Skorokhod topology.

Let $D_p[0,1]$ be the space of $\mathbb{R}^p$-valued functions on $[0,1]$ which are right-continuous with left limits. Let $X = \{X(t)\}_{t\in[0,1]}$ and $\{X_n(t)\}_{t\in[0,1]}$, $n \geq 1$, be stochastic processes with paths in $D_p[0,1]$. Let $\Rightarrow$ denote weak convergence, and suppose that $X_n \Rightarrow X$ as $n \to \infty$ in $D_p[0,1]$ equipped with the strong Skorokhod J_1 topology *Jacod and Shiryaev* (1987).

Proposition A.1 (Proposition VI.1.17 of *Jacod and Shiryaev* (1987))**.** *If X has continuous paths, then $X_n \Rightarrow X$ as $n \to \infty$ in the space $D_p[0,1]$ equipped with the uniform metric.*

Suppose that $f : \mathbb{R}^p \to \mathbb{R}$ is Borel measurable function and define the map $T_f : D_p[0,1] \to \mathbb{R}$

$$T_f(x) := \int_0^1 f(x(t))\, dt, \quad x \in D_p[0,1].$$

Now, let $\mathrm{Disc}(T_f)$ denote the set of discontinuity points of T_f, let $C_p[0,1]$ be the space of $\mathbb{R}^p$-valued continuous functions on $[0,1]$, and write Leb for Lebesgue measure.

Proposition A.2. *Suppose that f is bounded. We have that*

$$\mathrm{Disc}(T_f) \cap C_p[0,1] \subset \Big\{ x \in C[0,1] \,:\, \mathrm{Leb}(\{t \in [0,1] \,:\, x(t) \in \mathrm{Disc}(f)\}) > 0 \Big\} =: D_f. \tag{A.1}$$

Proof. Suppose that $x \in C_p[0,1]$ does not belong to the right–hand side of (A.1) and let $x_n \to x$ in J_1. Then, according to a standard property of the Skorokhod J_1 topology *Jacod and Shiryaev* (1987) we also have $\sup_{t\in[0,1]} |x_n(t) - x(t)| \to 0$, as $n \to \infty$. Now, since $x \notin D_f$, we have that for almost all $t \in [0,1]$, the point $x(t)$ is a continuity point of f. Therefore, $f(x_n(t)) \to f(x(t)),\ n \to \infty$, for almost all $t \in [0,1]$. Since f is bounded, the Lebesgue dominated convergence theorem then yields

$$T_f(x_n) \equiv \int_0^1 f(x_n(t))dt \longrightarrow \int_0^1 f(x(t))dt \equiv T_f(x), \quad \text{as } n \to \infty.$$

This completes the proof. □

In the context of stochastic processes, by the Continuous Mapping Theorem, we have convergence in distribution,

$$T_f(X_n) \xrightarrow{d} T_f(X), \quad \text{as } n \to \infty,$$

provided X has continuous paths and $\mathbb{P}(X \in \text{Disc}(f)) = 0$. In the case when $f(x) = 1_A(x)$, the latter translates to

$$\mathbb{P}\{\text{ The measure of the time } X \text{ spends on the boundary of } A \text{ is zero}\} = 1. \quad \text{(A.2)}$$

If the stochastic process has continuous marginal distribution and the set A has zero boundary, the Fubini's theorem readily implies (A.2). Indeed, the probability in (A.2) equals

$$\int_\Omega \int_0^1 1_{\partial A}(X(t,\omega))\, dt\, \mathbb{P}(d\omega) = \int_0^1 \mathbb{P}(X(t) \in \partial A)\, dt = 0,$$

provided that $\text{Leb}(\partial A) = 0$ and if $X(t)$ has a marginal density for each $t \in (0,1)$. The above arguments lead to the proof of the following result.

Lemma A.3. *Suppose that $X_n \Rightarrow X$ in $D_p[0,1]$, equipped with the uniform convergence topology. If the process X takes values in $C_p[0,1]$ and has continuous marginal distributions, then for all bounded Borel functions $f : \mathbb{R}^p \to \mathbb{R}$, that are continuous almost everywhere, i.e. such that $\text{Leb}(\text{Disc}(f)) = 0$, we have*

$$\int_0^1 f(X_n(t))\, dt \xrightarrow{d} \int_0^1 f(X(t))\, dt, \quad \text{as } n \to \infty.$$

A.2 Iterated importance sampling

When $N = 1$ in IF2, we obtain a general latent variable algorithm in which each iteration involves importance sampling but not filtering. This situation is called iterated importance sampling (*Ionides et al.*, 2011) and we call this special case of our algorithm IIS2. Iterated importance sampling has previously been used to provide a route into proving convergence of iterated filtering (*Ionides et al.*, 2011; *Doucet et al.*, 2013). However, in this article we found it more convenient to prove the full result for iterated filtering directly. Although IIS2 may have some independent value as

a practical algorithm, our only use of IIS2 in this article is to provide a convenient environment for explicit computations for Gaussian models in Section A.3 and non-Gaussian models in Section A.4.

Algorithm IIS2. Iterated importance sampling

---

input:

Simulator for $f_X(x\,;\theta)$	Evaluator for $f_{Y\mid X}(y\mid x\,;\theta)$
Data, y^*	Number of iterations, M
Initial parameter swarm, $\{\Theta_j^0,\ j \text{ in } 1\!:\!J\}$	Number of particles, J
Perturbation density, $h(\theta\mid\varphi\,;\sigma)$	Perturbation sequence, $\sigma_{1:M}$

output: Final parameter swarm, $\{\Theta_j^M,\ j \text{ in } 1\!:\!J\}$

For m in $1\!:\!M$

$\Phi_j^m \sim h(\theta\mid\Theta_j^{m-1}\,;\sigma_m)$ for j in $1\!:\!J$

$X_j^m \sim f_X(x\,;\Phi_j^m)$ for j in $1\!:\!J$

$w_j^m = f_{Y\mid X}(y^*\mid X_j^m\,;\Phi_j^m)$ for j in $1\!:\!J$

Draw $k_{1:J}$ with $\mathbb{P}(k_j=i) = w_{n,i}^m \Big/ \sum_{u=1}^J w_{n,u}^m$

$\Theta_j^m = \Phi_{k_j}^m$ for j in $1\!:\!J$

End For

A general latent variable model can be specified by a joint density $f_{XY}(x,y;\theta)$, with X taking values in $\mathbb{X}\subset\mathbb{R}^{\dim(\mathbb{X})}$, Y taking values in $\mathbb{Y}\subset\mathbb{R}^{\dim(\mathbb{Y})}$ and θ taking values in $\Theta\subset\mathbb{R}^{\dim(\Theta)}$. The data consist of a single observation, $y^*\in\mathbb{Y}$. The likelihood function is

$$\ell(\theta) = f_Y(y^*;\theta) = \int f_{XY}(x,y^*;\theta)\,dx,$$

and we look for a maximum likelihood estimate (MLE), i.e., a value $\hat\theta$ maximizing $\ell(\theta)$. The parameter perturbation step of Algorithm IIS2 is a Monte Carlo approximation

to a perturbation map H_σ where

$$H_\sigma g(\theta) = \int g(\varphi) h(\theta \,|\, \varphi \,; \sigma)\, d\varphi. \tag{A.3}$$

A natural choice for $h(\,\cdot\,|\,\varphi\,;\sigma)$ is the multivariate normal density with mean φ and variance $\sigma^2\Sigma$ for some covariance matrix Σ, but in general h could be any condition density parameterized by σ. The resampling step of Algorithm IIS2 is a Monte Carlo approximation to a Bayes map, B, given by

$$Bf(\theta) = f(\theta)\ell(\theta)\Big\{\int f(\varphi)\ell(\varphi)\, d\varphi\Big\}^{-1}. \tag{A.4}$$

When the standard deviation of the parameter perturbations is held fixed at $\sigma_m = \sigma > 0$, Algorithm IIS2 is a Monte Carlo approximation to $T_\sigma^M f(\theta)$ where

$$T_\sigma f(\theta) = BH_\sigma f(\theta) = \frac{\int f(\varphi)\,\ell(\theta) h(\theta \,|\, \varphi\,;\sigma)\, d\varphi}{\iint f(\varphi)\,\ell(\xi) h(\xi \,|\, \varphi\,;\sigma)\, d\varphi\, d\xi}. \tag{A.5}$$

A.3 Gaussian and near-Gaussian analysis of iterated importance sampling

The convergence results of Theorems 1 and 2 in the main text are not precise about the rate of convergence, either toward the MLE as $\sigma \to 0$ or toward the stationary distribution as $M \to \infty$. Explicit results are available in the Gaussian case and are also relevant to near-Gaussian situations. The near-Gaussian situation may arise in practice, since the parameter perturbations can be constructed to follow a Gaussian distribution and the log likelihood surface may be approximately quadratic due to asymptotic behavior of the likelihood for large sample sizes. The near-Gaussian situation for a POMP model does not require that the POMP itself is near Gaussian, only that the log likelihood surface is near quadratic. Here, we consider only the

univariate case, and only for iterated importance sampling. We offer this simplified case as an illustrative example, rather than an alternative justification for the use of our algorithm. In principle, these results can be generalized, but such results do not add much to the general convergence guarantees already obtained.

We investigate the eigenvalues and eigenfunctions for a Gaussian system, and then we appeal to continuity of the eigenvalues to study systems that are close to Gaussian. Here, we consider the case of a scalar parameter, $\dim(\Theta = 1)$, and an additive perturbation given by

$$h(\theta \,|\, \varphi \,; \sigma) = \kappa(\theta - \varphi). \tag{A.6}$$

We first study the unnormalized version of (A.5) defined as

$$Sf(\theta) = \big[f(\theta)\, \ell(\theta)\big] * \kappa(\theta) = \int \big[f(\theta - \varphi)\, \ell(\theta - \varphi)\big]\, \kappa(\varphi)\, d\varphi. \tag{A.7}$$

This is a linear map, and we obtain the eigenvalues and eigenfunctions when ℓ and h are Gaussian in Proposition A.4. Iterations of the corresponding normalized map, T_σ, converge to the normalized eigenfunction corresponding to the largest eigenvalue of S, which can be seen by postponing normalization until having carried out a large number of iterations of the unnormalized map. Suppose, without loss of generality, that the maximum of the likelihood is at $\theta = 0$. Let $\phi(\theta; \sigma)$ be the normal density with mean zero and variance σ^2.

Proposition A.4. *Let S_0 be the map constructed as in (A.7) with the choices $\ell(\theta) = \phi(\theta; \tau)$ and $\kappa(\theta) = \phi(\theta; \sigma)$. Let*

$$u^2 = \left(\sigma^2 + \sqrt{\sigma^4 + 4\sigma^2\tau^2}\right)/2 = \sigma\tau + o(\sigma). \tag{A.8}$$

The eigenvalues of S_0 are

$$\lambda_n = \sigma\tau\sqrt{2\pi}\left(\frac{u^2-\sigma^2}{u^2}\right)^{(n+1)/2},$$

for $n = 0, 1, 2, \ldots,$ and the corresponding eigenfunctions have the form

$$e_n = p_n(\theta)\phi(\theta; u), \tag{A.9}$$

where p_n is a polynomial of degree n.

Proof. Let P_n be the subspace of functions of the form $q(\theta)\phi(\theta; u)$ where q is a polynomial of degree less than or equal to n. We show that S_0 maps P_n into itself, and look at what happens to terms of degree n. Let H_n be the Hermite polynomial of degree n, defined by $(d/d\theta)^n\phi(\theta; 1) = (-1)^n H_n(\theta)\phi(\theta; 1)$. Let $\alpha = (1/u^2 + 1/\tau^2)^{-1/2}$, and set

$$f(\theta) = \alpha^{-2n} H_n(\theta/\alpha)\phi(\theta; u). \tag{A.10}$$

Then,

$$f(\theta)\ell(\theta) = \frac{\alpha}{\sigma\tau\sqrt{2\pi}}\,\alpha^{-2n} H_n(\theta/\alpha)\phi(\theta;\alpha) = \frac{\alpha}{\sigma\tau\sqrt{2\pi}}(-1)^n\frac{d^n}{d\theta^n}\phi(\theta;\alpha). \tag{A.11}$$

Since $\big[(d/d\theta)^n f\ell\big] * \kappa = (d/d\theta)^n\big[(f\ell) * \kappa\big]$, we get

$$(f\ell) * \kappa = \frac{\alpha}{\sigma\tau\sqrt{2\pi}}(-1)^n\frac{d^n}{d\theta^n}\phi(\theta; u) = \frac{\alpha}{\sigma\tau\sqrt{2\pi}}u^{-2n}H_n(\theta/u)\phi(\theta; u). \tag{A.12}$$

Writing $H_n(\theta) = h_0 + h_1\theta + \cdots + h_n\theta^n$, we see that the coefficient of the term in θ^n in (A.10) is $\alpha^{-n}h_n$, whereas in (A.12) it is $\frac{\alpha}{\sigma\tau\sqrt{2\pi}}u^{-n}$. We have shown that S_0 operating on P_n multiplies the coefficient of degree n by a factor of λ_n. Letting L_n be the matrix representing S_0 on P_n with the basis $b_0, \ldots, b_m$ given by $b_m(\theta) = \theta^m\phi(\theta; u)$, we see that L_n is lower triangular with diagonal entries $\lambda_0, \ldots, \lambda_n$. Therefore, the

eigenvalues are $\lambda_0, \ldots, \lambda_n$, and the eigenfunction corresponding to λ_m is in P_m. $\square$

The case where $\log \ell(\theta)$ is close to quadratic is relevant due to asymptotic log quadratic properties of the likelihood function. Choosing $\kappa(\theta)$ to be Gaussian, as in Proposition A.4, we have the following approximation result.

Proposition A.5. *Let S_ϵ be a map as in (A.7), with ℓ satisfying $\sup_\theta |\ell(\theta) - \phi(\theta;\tau)| < \epsilon$ and $\kappa(\theta) = \phi(\theta;\sigma)$. For ϵ small, the largest eigenvalue of S_ϵ is close to λ_0 and the corresponding eigenfunction is close to $\phi(\theta;u)$.*

Proof. Write $\ell(\theta) = \phi(\theta;\tau) + \eta(\theta)$, with $\sup_\theta |\eta(\theta)| < \epsilon$. Then,

$$\|S_\epsilon f - S_0 f\| = \|(f\eta) * \kappa\| \leq \|f\eta\| \leq \epsilon \|f\|. \tag{A.13}$$

Here, $\|\cdot\|$ is the L^2 norm of a function or the corresponding operator norm (largest absolute eigenvalue). Convolution with κ is a contraction in L^2, which is apparent by taking Fourier transforms and making use of Parseval's relationship, since all frequencies are shrunk by multiplying with the Fourier transform of κ. From (A.13), we have $\|S_0 - S_\epsilon\| < \epsilon$. This implies that S_ϵ has a largest eigenvalue μ_0 with $|\mu_0 - \lambda_0| < \epsilon$, based on the representation that

$$|\mu_0| = \|S\| = \sup_f \frac{\|S_\epsilon f\|}{\|f\|}. \tag{A.14}$$

Writing the corresponding unit eigenfunction as w_0, we have

$$w_0 = (1/\mu_0) S_\epsilon w_0 = (1/\mu_0)[S_0 w_0 + \eta], \tag{A.15}$$

where $\|\eta(\theta)\| < \epsilon$. Writing $w_0 = \sum_{i=1}^\infty \alpha_i e_i$, in terms of $\{e_i\}$ from (A.9), equa-

tion (A.15) gives

$$\sum_{i=1}^{\infty} \alpha_i e_i = \sum_{i=1}^{\infty} \alpha_i \frac{\lambda_i}{\mu_0} e_i + \eta = \sum_{i=1}^{\infty} \alpha_i \frac{\lambda_i}{\lambda_0} e_i + \tilde{\eta}, \tag{A.16}$$

where $\|\tilde{\eta}\| < \epsilon(1 + [\lambda_0(\lambda_0 - \epsilon)]^{-1})$. Comparing terms in e_i, we see that all terms $\alpha_1, \alpha_2, \ldots$ must be of order ϵ. □

A.4 A class of exact non-Gaussian limits for iterated importance sampling

We look for exact solutions to the equation $Tf = f$ where $T = BH$, as specified in (A.5) with $h(\theta \,|\, \varphi\,; \sigma) = \kappa(\theta - \varphi)$. This situation corresponds to iterated importance sampling with additive parameter perturbations that have no dependence on σ, as in equation (A.6). Now, for $g(x)$ being a probability density on Θ, define

$$\ell_g(x) = c \frac{g(x)}{\kappa * g(x)}, \tag{A.17}$$

where c is a non-negative constant. For likelihood functions of the form (A.17), supposing that ℓ_g is integrable, we obtain an eigenfunction $e(x) = \kappa * g(x)$ for the unnormalized map S defined in (A.7) via the following calculation:

$$\begin{aligned} Se(x) &= c \int \frac{g(x-u)}{(g*\kappa)(x-u)} (g*\kappa)(x-u)\kappa(u)\, du && \text{(A.18)} \\ &= c \int g(x-u)\kappa(u)\, du && \text{(A.19)} \\ &= c\big[g*\kappa(x)\big] \quad = \quad c\, e(x). && \text{(A.20)} \end{aligned}$$

Under conditions such as Theorem 1, it follows that $\kappa * g$ is the unique eigenfunction for T, up to a scale factor, and that $\lim_{M\to\infty} T^M f = e$. We do not anticipate practical applications for the conjugacy relationship we have established between the pair (ℓ_g, κ)

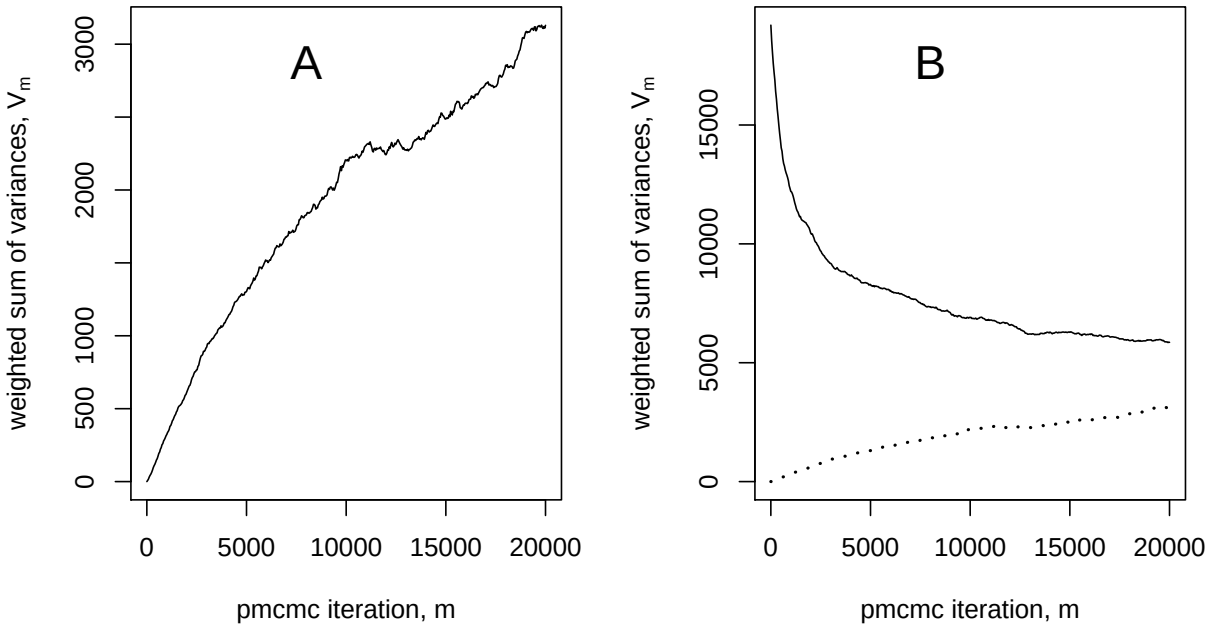

Figure A.1: PMCMC convergence assessment, using the diagnostic quantity in equation A.21. (A) Underdispersed chains, all started at the MLE. (B) Overdispersed chains, started with draws from the prior (solid line), and underdispersed chains (dotted line). The average acceptance probability was 0.04238, with Monte Carlo standard error 0.00072, calculated from iterations 5000 through 20000 for the 100 underdispersed PMCMC chains. For the overdispersed chains, the average acceptance probability was 0.04243 with standard error 0.00100.

since we see no reason why the likelihood should have the form of (A.17). However, this situation does serve to identify a range of possible limiting behaviors for T^M.

A.5 Applying PMCMC to the cholera model

We carried out PMCMC for the cholera model, with the prior being uniform on the hyper-rectangle specified by θ_{low} and θ_{high} in Table 1. Thus, the IF1 and IF2 searches were conducted starting with random draws from this prior. Since PMCMC is known to be computationally demanding, we investigated a simplified

challenge: investigating the posterior distribution starting at the MLE. This would be appropriate, for example, if one aimed to obtain Bayesian inferences using PMCMC but giving it a helping hand by first finding a good starting value obtained by a maximization procedure. We used the PMMH implementation of PMCMC in `pomp` (*King et al.*, 2015b) with parameter proposals following a Gaussian random walk with standard deviations given by $(\theta_{\text{high}} - \theta_{\text{low}})/100$. We started 100 independent chains at the estimated MLE in Table 1. Each PMCMC chain, with $J = 1500$ particles at each of $M = 2 \times 10^4$ likelihood evaluations, took around 30 hours to run on a single core of the University of Michigan Flux cluster. Writing $V_{m,d}$ for the sample variance of variable $d \in \{1, \ldots, \dim(\Theta)\}$ among the 100 chains at time $m \in \{1 \ldots, M\}$, and τ_d for the Gaussian random walk standard deviation for parameter d, we tracked the quantity

$$V_m = \sum_{d=1}^{\dim(\Theta)} \frac{V_{m,d}}{\tau_d^2}. \tag{A.21}$$

Supposing the posterior variance is finite, a necessary requirement for convergence to stationarity as m increased is for V_m to approach its asymptotic limit. Since all the chains start at the same place, one expects V_m to increase toward this limit. The number of iterations required for V_m to stabilize therefore provides a lower bound on the time taken for convergence of the chain. This test assesses the capability of the chain to explore the region of parameter space with high posterior probability density, rather than the capability to search for this region from a remote starting point. We also tested PMCMC on a harder challenge, investigating convergence of the MCMC chain to its stationary distribution from over-dispersed starting values. We repeated the computation described above, with 100 chains initialized at draws from the prior distribution. The results are shown in Figure A.1. From Figure A.1A, we see that the stationary distribution has not yet been approached for the chains starting at the MLE, since the variance of independent chains continues to increase up to $M = 2 \times 10^4$. As a harder test, the variance for the initially overdispersed

independent chains should approach that for the initially underdispersed chains, but we see in Figure A.1B that much more computation would be required to achieve this with the algorithmic settings used.

The PMCMC chains used here involved $JMN = (1.5\times10^3)\times(2\times10^4)\times(6\times10^2) = 1.8\times10^{10}$ calls to the dynamic process simulator (the dominating computational expense), and yet failed to converge. By contrast, IF2 with $JMN = (10^4)\times10^2\times(6\times10^2) = 6\times10^8$ calls to the dynamic process simulator was shown to be an effective tool for global investigation of the likelihood surface. As with all numerical comparisons, it is hard to assess whether poor performance is a consequence of poor algorithmic choices. Conceptually, a major difference between iterated filtering and PMCMC is that the filtering particles in IF2 investigate the parameter space and latent dynamic variable space simultaneously, whereas in PMCMC each filtering iteration is used only to provide a single noisy likelihood evaluation. It may not be surprising that algorithms such as PMCMC struggle in situations where filtering is a substantial computational expense and the likelihood surface is sufficiently complex that many thousands of Monte Carlo steps are required to explore it. Indeed, IF1 and IF2 remain the only algorithms that have currently been demonstrated computationally capable of efficient likelihood-based inference for situations of comparable difficulty to our example.

A.6 Applying Liu & West's method to the toy example

Bayesian parameter estimation for POMP models using sequential Monte Carlo with perturbed parameters was proposed by *Kitagawa* (1998). Similar approaches using alternative nonlinear filters have also been widely used *Anderson and Moore* (1979); *Wan and van der Merwe* (2000). Liu & West (*Liu and West*, 2001) proposed a development on the approach of *Kitagawa* (1998) which combines parameter perturbations with a contraction that is designed to counterbalance the variation

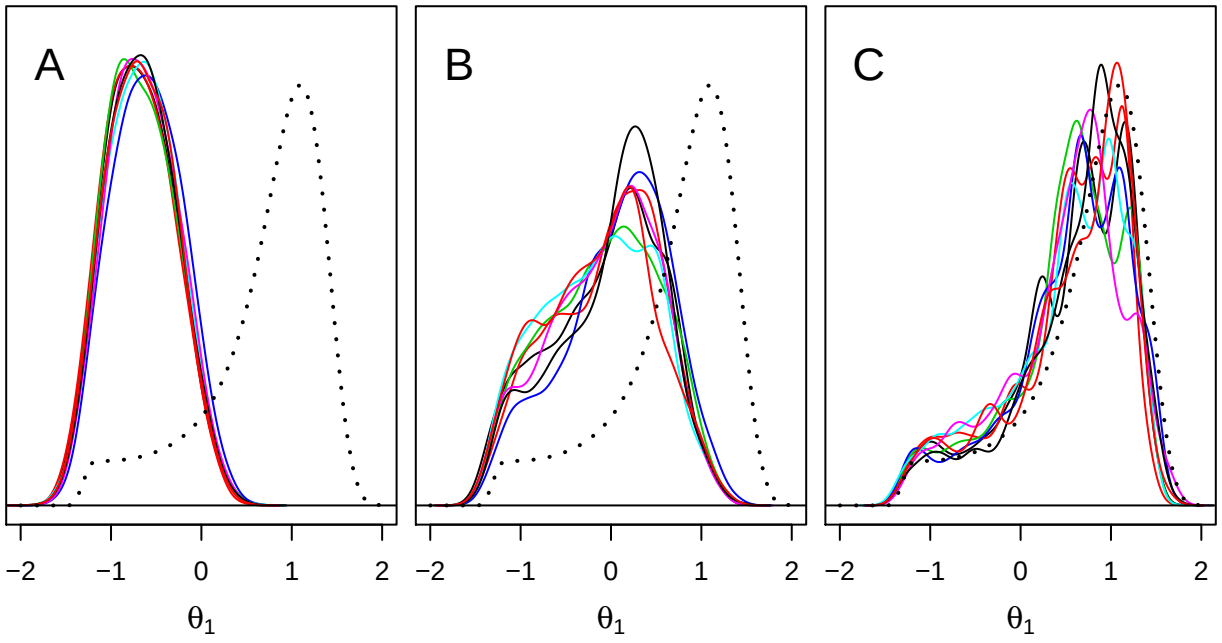

Figure A.2: The Liu & West algorithm *Liu and West* (2001) applied to the toy example with varying values of the discount factor: (A) $\delta = 0.99$; (B) $\delta = 0.999$; (C) $\delta = 0.9999$. Solid lines show 8 independent estimates of the marginal posterior density of θ_1. The black dotted line shows the true posterior density.

added by the perturbations, thereby approximating the posterior distribution of the parameters for the fixed parameter model of interest. Liu & West (*Liu and West*, 2001) also included an auxiliary particle filter procedure in their algorithm (*Pitt and Shepard*, 1999). The auxiliary particle filter is a version of sequential Monte Carlo which looks ahead to a future observation when deciding which particles to propagate. Generally, auxiliary particle filter algorithms do not have the plug-and-play property (*Bretó et al.*, 2009; *He et al.*, 2010) since they involve constructing weights that require evaluation of the transition density for the latent process. In addition, the auxiliary particle filter does not necessarily have superior performance over a basic sequential Monte Carlo filter (*Johansen and Doucet*, 2008). To compare with IF2 and PMCMC on our toy example, we therefore employ a version of the Liu & West algorithm, which we call LW, that omits the auxiliary particle filter procedure. LW carries out the key innovation of parameter perturbation and contraction (Steps 3 and 4 in Sec. 10.4 of *Liu and West* (2001)) while omitting the auxiliary particle filter (Steps 1 and 2, and the denominator in Step 5, in Sec. 10.4 of *Liu and West* (2001)). LW was implemented via the `bsmc2` function of the `pomp` package (*King et al.*, 2015b). If an effective auxiliary particle filter were available for a specific computation, it could also be used to enhance other sequential Monte Carlo based inference procedures such as IF1, IF2 and PMCMC.

For the numerical results reported in Fig. A.2 we used $J = 10^4$ particles for LW. This awards the same computational resources to LW that we gave IF1 and IF2 for the results in Fig. 1. The magnitude of the perturbations in LW is controlled by a discount factor (δ in the notation of *Liu and West* (2001)), and we considered three values, $\delta \in \{0.99, 0.999, 0.9999\}$. Liu & West *Liu and West* (2001) suggested that δ should take values in the range $\delta \in [0.95, 0.99]$, with smaller values of δ reducing Monte Carlo variability while increasing bias in the approximation to the target posterior distribution. For our toy example, we see from Fig. A.2A that the choice

$\delta = 0.99$ results in a stable Monte Carlo computation (since all eight realizations are close). However, Fig. A.2A also reveals a large amount of bias. Increasing δ to 0.999, Fig. A.2B shows some increase in the Monte Carlo variability and some decrease in the bias. Further increasing δ to 0.9999, Fig. A.2C shows the bias becomes small while the Monte Carlo variability continues to increase. Values of δ very close to one are numerically tractable for this toy model, but not in most applications. As δ approaches one, the ensuing numerical instability exemplifies the principal reason why Bayesian and likelihood-based inference for POMP models is challenging despite the development of modern nonlinear filtering techniques.

The justification provided by *Liu and West* (2001) for their algorithm is based on a Gaussian approximation to the posterior distribution. Specifically, *Liu and West* (2001) argued that the posterior distribution should be approximately unchanged by carrying out a linear contraction toward its mean followed by adding an appropriate perturbation. Therefore, it may be unsurprising that LW performs poorly in the presence of nonlinear ridges in the likelihood surface. Other authors have reported poor numerical performance for the algorithm of *Liu and West* (2001), e.g., Fig. 2 of *Storvik* (2002) and Fig. 2 of *Chopin et al.* (2013). Our results are consistent with these findings, and we conclude that the approach of *Liu and West* (2001) should be used with considerable caution when the posterior distribution is not close to Gaussian.

A.7 Consequences of perturbing parameters for the numerical stability of SMC

The IF2 algorithm applies sequential Monte Carlo (SMC) to an extended POMP model in which the time-varying parameters are treated as dynamic state variables. This procedure increases the dimension of the state space by the number of time-varying parameters. Empirically, SMC has been found effective in many low di-

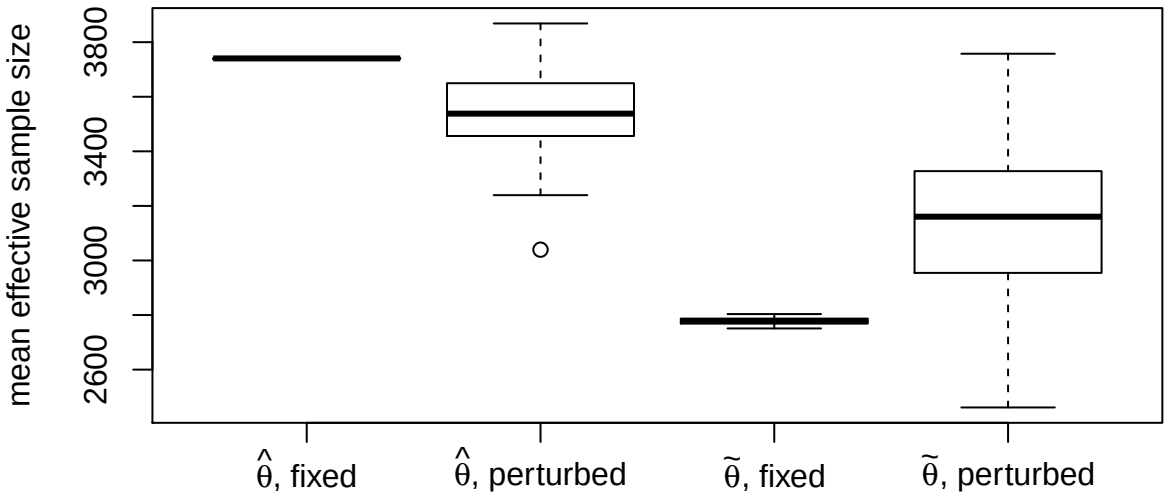

Figure A.3: Effective sample size (ESS) for SMC with fixed parameters and with perturbed parameters. We ran SMC for the cholera model with the parameter vector set at the MLE, $\hat{\theta}$, and at an alternative parameter vector $\tilde{\theta}$ for which the first 18 parameters in Table 1 were multiplied by a factor of 0.8. We defined the ESS at each time point by the reciprocal of the sum of squares of the normalized weights of the particles. The mean ESS was calculated as the average of these ESS values over the 600 time points. Repeating this computation 100 times, using $J = 10^4$ particles, gave 100 mean ESS values shown in the "fixed" columns of the box-and-whisker plot. Repeating the computation with additional parameter perturbations having random walk standard deviation of 0.01 gave the 100 mean ESS values shown in the "perturbed" column. For both parameter vectors, the perturbations greatly increase the spread of the mean ESS. At $\hat{\theta}$, the perturbations decreased the mean ESS value by 5% on average, whereas at $\tilde{\theta}$ the perturbations increased the mean ESS value by 13% on average. The MLE may be expected to be a favorable parameter value for stable filtering, and our interpretation is that the parameter perturbations have some chance of moving the SMC particles away from this favorable region. When started away from the MLE, the numerical stability of the IF2 algorithm benefits from the converse effect that the parameter perturbations will move the SMC particles preferentially toward this favorable region. For parameter values even further from the MLE than $\tilde{\theta}$, SMC may fail numerically for a fixed parameter value yet be feasible with perturbed parameters.

mensional systems but its numerical performance can degrade in larger systems. A natural concern, therefore, is the extent to which the extension of the state variable in IF2 increases the numerical challenge of carrying out SMC effectively. Two rival heuristics suggest different answers. One intuitive (but not universally correct) argument is that adding variability to the system stabilizes numerically unstable filtering problems, since it gives each particle at least a slim chance of following a trajectory compatible with the data. An opposing intuition, that SMC breaks down rapidly as the dimension increases, has theoretical support (*Bengtsson et al.*, 2008). However, the theoretical arguments of *Bengtsson et al.* (2008) may be driven more by increasing the observation dimension than increasing the state dimension, so their relevance in the present situation is not entirely clear.

We investigated numerical stability of SMC, in the context of our cholera example, by measuring the effective sample size (ESS) (*Liu*, 2001). We investigated the ESS for two parameter vectors, the MLE and an alternative value for which SMC is more numerically challenging. We carried out particle filtering with and without random walk perturbations to the parameters, obtaining the results presented in Fig. A.3. We found that the random walk perturbations led to a 5% decrease in the average ESS at the MLE, but a 13% increase in the average ESS at the alternative parameter vector. This example demonstrates that the random walk perturbations can have both a cost and a benefit for numerical stability, with the benefit outweighing the cost as the filtering problem becomes more challenging.

A.8 Checking conditions B1 and B2

We check B1 and B2 when Θ is a rectangular region in $\mathbb{R}^{\dim(\Theta)}$, with $h_n(\theta \,|\, \phi\,; \sigma)$ describing a Gaussian random walk having as a limit a reflected Brownian motion on Θ. A more general study of the limit of reflected random walks to reflected Brownian motions (in particular, including limits where the random walk step distribution

satisfies B5) was presented by *Bossy et al.* (2004). The specific examples of the IF2 algorithm given in our paper all employ Gaussian random walk perturbations for the parameters. The examples did not employ boundary conditions to constrain the parameter to a bounded set. While such conditions could be used to ensure practical stability of the algorithm, we view the conditions primarily as a theoretical device to assist the mathematical analysis of the algorithm.

Suppose that $\Theta = [a_1, b_1] \times [a_2, b_2] \times ... \times [a_{\dim(\Theta)}, b_{\dim(\Theta)}]$. For each coordinate direction $d = 1, \ldots, \dim(\Theta)$, let $R_d : \mathbb{R} \to [a_d, b_d]$ be the reflection map defined recursively by

$$R_d(x) = \begin{cases} x & \text{if} \quad x \in [a_d, b_d] \\ R_d(2b_d - x) & \text{if} \quad x > b_d \\ R_d(2a_d - x) & \text{if} \quad x < a_d \end{cases} .$$

Let $h_{n,d}(\theta_d \,|\, \phi_d \,; \sigma)$ be the density of $R_d(\phi_d + \sigma Z)$ where Z is a standard Normal random variable. Let $h_n(\theta \,|\, \phi \,; \sigma)$ be the joint density corresponding to the product of $h_{n,1}, \ldots, h_{n,\dim(\Theta)}$. This choice of h_n corresponds to a perturbation process for the parameter vector in the IF2 algorithm following a Gaussian random walk on Θ with reflective boundary conditions, independently in each coordinate direction. By construction, the finite dimensional distributions of $W_\sigma(t)$ at the set of times

$$\{k\sigma^2 : k = 0, 1, 2, \ldots \text{ and } k\sigma^2 \leq 1\}$$

exactly match the corresponding finite dimensional distributions of a reflected Brownian motion $\{W(t)\}$ taking values in Θ. This $\{W(t)\}$ gives a construction of the limiting process whose existence is assumed in B1. For $A \subset \Theta$, we see from this construction of $\{W(t)\}$ that the probability $\{W(t)\}$ is in A for all $\epsilon \leq t \leq 1]$ is greater than the corresponding probability for an unreflected Brownian motion, $\{W_{(u)}(t)\}$ with the same intensity parameter. It is routine to check that $\{W_{(u)}(t)\}$ has a pos-

itive probability of remaining in any open set A for all $\epsilon \leq t \leq 1$ uniformly over all values of $W_{(u)}(0) \in \Theta$. Thus, we have completed the check of condition B1.

To check B2, the positivity of the marginal density of $W(t)$ on Θ, uniformly over the value of $W(0)$, again follows since this density is larger than the known density for $W_{(u)}(t)$.

A.9 Additional details for the proof of Theorem 1

In the main text, a condensed proof of Theorem 1 is provided to describe the key steps in the argument. Here, we restate Theorem 1 and provide a more detailed proof. The reader is referred back to the main text for the notation and statement of conditions B2 and B4. Let $L^1(\Theta)$ denote the space of integrable real-valued functions on Θ with norm $\|f\|_1 = \int |f(\theta)|\, d\theta$. For non-negative measures μ and ν on Θ, let $\|\mu - \nu\|_{\mathsf{tv}}$ denote the total variation distance and let $H(\mu, \nu)$ denote the Hilbert metric distance *Eveson* (1995); *Le Gland and Oudjane* (2004). The measures μ and ν are said to be comparable if they are both nonzero and there exist constants $0 < a \leq b$ such that $a\,\nu(A) \leq \mu(A) \leq b\,\nu(A)$ for all measurable subsets $A \subset \Theta$. For comparable measures, $H(\mu, \nu)$ is defined by

$$H(\mu, \nu) = \log \frac{\sup_A \mu(A)/\nu(A)}{\inf_A \mu(A)/\nu(A)}, \tag{A.22}$$

with the supremum and infimum taken over measurable subsets $A \subset \Theta$ having $\nu(A) > 0$. For noncomparable measures, the Hilbert metric is defined by $H(0, 0) = 0$ and otherwise $H(\mu, \nu) = \infty$. The Hilbert metric is invariant to multiplication by a positive scalar, $H(a\mu, \nu) = H(\mu, \nu)$. This projective property makes the Hilbert metric convenient to investigate the Bayes map: in the context of the following proof, the projective property lets us analyze the linear map S_σ to study the nonlinear map T_σ.

Theorem 1. *Let T_σ be the map defined by [1] in the main text, and suppose B2 and B4. There exists a unique probability density f_σ such that for any probability density f on Θ,*

$$\lim_{m\to\infty} \|T_\sigma^m f - f_\sigma\|_1 = 0, \tag{A.23}$$

where $\|f\|_1$ is the L^1 norm of f. Let $\{\Theta_j^M,\ j = 1,\ldots,J\}$ be the output of IF2, with $\sigma_m = \sigma > 0$. There exists a finite constant C such that

$$\limsup_{M\to\infty} \mathbb{E}\left[\left|\frac{1}{J}\sum_{j=1}^{J}\phi(\Theta_j^M) - \int \phi(\theta) f_\sigma(\theta)\, d\theta\right|\right] \le \frac{C\sup_\theta |\phi(\theta)|}{\sqrt{J}}. \tag{A.24}$$

Proof. For $\theta_{0:N} \in \Theta^{N+1}$, we single out the last component of $\theta_{0:N}$ by writing $\breve{\ell}(\theta_{0:N}) = \breve{\ell}(\theta_{0:N-1}, \theta_N)$ and $h(\theta_{0:N} \,|\, \phi) = h(\theta_{0:N-1}, \theta_N \,|\, \phi)$. Then, for ϕ and θ in Θ, we define

$$s_\sigma(\phi,\theta) = \int h(\theta_{0:N-1}, \theta|\phi,\sigma)\breve{\ell}(\theta_{0:N-1},\theta)\, d\theta_{0:N-1}. \tag{A.25}$$

The function s_σ in (A.25) defines a linear operator $S_\sigma f(\theta) = \int s_\sigma(\phi,\theta) f(\phi)\mathrm{d}\phi$ that maps $L^1(\Theta)$ into itself. Notice that $T_\sigma f(\theta) = S_\sigma f(\theta)/\|S_\sigma f\|_1$. More generally, if μ is a probability measure on Θ, $S_\sigma\mu$ denotes the function $S_\sigma\mu(\theta) = \int s_\sigma(\phi,\theta)\mu(d\phi)$. Notice also that $S_\sigma^m f$, the m-th iterate of S_σ, can be written as $S_\sigma^m f(\theta) = \int s_\sigma^{(m)}(\phi,\theta) f(\phi) d\phi$, where $s_\sigma^{(1)}(\phi,\theta) = s_\sigma(\phi,\theta)$, and for $m \ge 2$, $s_\sigma^{(m)}(\phi,\theta) = \int s_\sigma(\phi,u) s_\sigma^{(m-1)}(u,\theta) du$. Using the definition of $\breve{\ell}$ and B4,

$$\begin{aligned} s_\sigma(\phi,\theta) &= \int h(\theta_{0:N-1},\theta|\phi,\sigma) \int f_X(x_{0:N}|\theta_{0:N-1},\theta) f_{Y|X}(y_{1:N}^*|x_{0:N})\, dx_{0:N}\, d\theta_{0:N-1} \\ &\ge \epsilon^N \int h(\theta_{0:N-1},\theta|\phi,\sigma)\, d\theta_{0:N-1}, \end{aligned} \tag{A.26}$$

and, similarly,

$$s_\sigma(\phi,\theta) \le \epsilon^{-N} \int h(\theta_{0:N-1},\theta|\phi,\sigma)\, d\theta_{0:N-1}. \tag{A.27}$$

By iterating the inequalities (A.26) and (A.27), assumption B2 implies that there exists $m_0 \geq 1$ such that for any $m \geq m_0$, there exist $0 < \delta_m < \infty$, a probability measure λ_m on Θ such that for all measurable subsets $A \subset \Theta$ and all $\theta \in \Theta$,

$$\delta_m \lambda_m(A) \leq \int_A s^{(m)}(\theta, \phi)\, d\phi \leq \delta_m^{-1} \lambda_m(A). \tag{A.28}$$

In other words, $S_\sigma^{m_0}$ is mixing in the sense of *Le Gland and Oudjane* (2004). In the terminology of *Eveson* (1995), this means that for each $m \geq m_0$, S^m has finite projective diameter (see Lemma 2.6.2 of *Eveson* (1995)). Therefore, by Theorem 2.5.1 of *Eveson* (1995), we conclude that S_σ has a unique non-negative eigenfunction f_σ with $\|f_\sigma\|_1 = 1$, and for any density f on Θ, as $q \to \infty$,

$$\left\| \frac{[S_\sigma^{m_0}]^q f}{\|[S_\sigma^{m_0}]^q f\|_1} - f_\sigma \right\|_1 = \|T_\sigma^{m_0 q} f - f_\sigma\|_1 \to 0.$$

This implies the statement (A.23), by writing for any $m \geq 1$, $m = qm_0 + r$, for $0 \leq r \leq m_0 - 1$, and $T_\sigma^m f = [T_\sigma^{qm_0}] T_\sigma^r f$.

Let the initial particle swarm $\{\Theta_j^0,\ 1 \leq j \leq J\}$ consist of independent draws from the density f. To prove (A.24), we decompose $M = qm_0 + r$, for some $r \in \{0, \ldots, m_0 - 1\}$, and we introduce the empirical measures $\mu^{(0)} = J^{-1} \sum_{j=1}^J \delta_{\Theta_j^{(r)}}$, and for $k = 1, \ldots, q$, $\mu^{(k)} = J^{-1} \sum_{j=1}^J \delta_{\Theta_j^{(r+m_0 k)}}$, so that $\mu^{(q)} = J^{-1} \sum_{j=1}^J \delta_{\Theta_j^{(M)}}$. We then write, for any bounded measurable function ϕ,

$$\begin{aligned}
\mu^{(q)}(\phi) - [T_\sigma^M f](\phi) &= \mu^{(q)}(\phi) - \left[T_\sigma^{m_0 q} \mu^{(0)}\right](\phi) + \left[T_\sigma^{m_0 q} \mu^{(0)}\right](\phi) - \left[T_\sigma^{m_0 q} T_\sigma^r f\right](\phi) \\
&= \sum_{i=1}^q \left\{ \left[T_\sigma^{m_0 (i-1)} \mu^{(q-i+1)}\right](\phi) - \left[T_\sigma^{m_0 i} \mu^{(q-i)}\right](\phi) \right\} \\
&\qquad + \left[T_\sigma^{m_0 q} \mu^{(0)}\right](\phi) - \left[T_\sigma^{m_0 q} T_\sigma^r f\right](\phi).
\end{aligned}$$

Using Theorem 2 of *Crisan and Doucet* (2002), we can find a finite constant C_3 such

that for all $k \geq 1$, and writing $\|\phi\|_\infty = \sup_\theta |\phi(\theta)|$,

$$\rho = \sup_{\phi:\, \|\phi\|_\infty = 1} \mathbb{E}\left[\left| \mu^{(k)}(\phi) - \left[T_\sigma^{m_0} \mu^{(k-1)}\right](\phi) \right| \right] \leq \frac{C_3}{\sqrt{J}}, \tag{A.29}$$

with B4 implying that the constant C_3 constructed by *Crisan and Doucet* (2002) does not depend on $\mu^{(k-1)}$. Since $S_\sigma^{m_0}$ is mixing and (A.28) holds, using Lemma 3.4, Lemma 3.5, Lemma 3.8 and Equation (7) of *Le Gland and Oudjane* (2004), we have

$$\begin{aligned}
\mathbb{E}\left[\left| \left[T_\sigma^{m_0 q} \mu^{(0)}\right](\phi) - \left[T_\sigma^{m_0 q} T_\sigma^r f\right](\phi) \right| \right] &\leq \|\phi\|_\infty \mathbb{E}\left[\left\| T_\sigma^{m_0 q} \mu^{(0)} - T_\sigma^{m_0 q} T_\sigma^r f \right\|_{\mathsf{tv}} \right] \\
&\leq \frac{2\|\phi\|_\infty}{\log 3} \mathbb{E}\left[H\left(S_\sigma^{m_0 q} \mu^{(0)}, S_\sigma^{m_0 q} T_\sigma^r f \right) \right] \\
&\leq \frac{2\|\phi\|_\infty}{\log 3} \left(\frac{1-\delta_{m_0}^2}{1+\delta_{m_0}^2} \right)^{q-2} \frac{1}{\delta_{m_0}^2} \mathbb{E}\left[\left\| T_\sigma^{m_0} \mu^{(0)} - T_\sigma^{m_0} T_\sigma^r f \right\|_{\mathsf{tv}} \right] \\
&\leq \frac{4\|\phi\|_\infty}{\log 3} \left(\frac{1-\delta_{m_0}^2}{1+\delta_{m_0}^2} \right)^{q-2} \frac{1}{\delta_{m_0}^2} \frac{\rho}{\delta_{m_0}^2}.
\end{aligned}$$

For $i = 3, \ldots, q$, a similar calculation gives

$$\begin{aligned}
\mathbb{E}\left[\left| T_\sigma^{m_0(i-1)} \mu^{(q-i+1)}(\phi) - T_\sigma^{m_0 i} \mu^{(q-i)}(\phi) \right| \right] &= \mathbb{E}\left[\left| T_\sigma^{m_0(i-1)} \mu^{(q-i+1)}(\phi) - T_\sigma^{m_0(i-1)} T_\sigma^{m_0} \mu^{(q-i)}(\phi) \right| \right] \\
&\leq \frac{4\|\phi\|_\infty}{\log 3} \left(\frac{1-\delta_{m_0}^2}{1+\delta_{m_0}^2} \right)^{i-3} \frac{1}{\delta_{m_0}^2} \frac{\rho}{\delta_{m_0}^2}.
\end{aligned}$$

The case $i = 1$ boils down to (A.29), where the case $i = 2$ gives by similar calculations:

$$\mathbb{E}\left[\left| T_\sigma^{m_0} \mu^{(q-1)}(\phi) - T_\sigma^{2m_0} \mu^{(q-2)}(\phi) \right| \right] \leq 2\|\phi\|_\infty \frac{\rho}{\delta_{m_0}^2}.$$

Hence, using (A.29),

$$\mathbb{E}\left[\left| \mu^{(q)}(\phi) - [T_\sigma^M f](\phi) \right| \right] \leq \frac{C_3 \|\phi\|_\infty}{\sqrt{J}} \left(1 + \frac{2}{\delta_{m_0}^2} + \frac{4}{\log 3} \left(\frac{1}{\delta_{m_0}^2} \right)^2 \sum_{j=0}^{q-2} \left(\frac{1-\delta_{m_0}^2}{1+\delta_{m_0}^2} \right)^j \right).$$

We conclude that there exists a finite constant C_4 such that

$$\mathbb{E}\left[\left|\frac{1}{J}\sum_{j=1}^{J}\phi(\Theta_j^M) - \int \phi(\theta)\left[T_\sigma^M f\right](\theta)\,d\theta\right|\right] \leq \frac{C_4\|\phi\|_\infty}{\sqrt{J}}. \tag{A.30}$$

Equation (A.24) follows by combining (A.30) with (A.23). □

A.10 Parameters and parameter ranges for the cholera model

Table S-1. Parameters for the cholera model.

	$\hat{\theta}$	θ_{low}	θ_{high}
γ	20.80	10.00	40.00
ϵ	19.10	0.20	30.00
m	0.06	0.03	0.60
$\beta_{\text{trend}} \times 10^2$	-0.50	-1.00	0.00
β_1	0.75	-4.00	4.00
β_2	6.38	0.00	8.00
β_3	-3.44	-4.00	4.00
β_4	4.23	0.00	8.00
β_5	3.33	0.00	8.00
β_6	4.55	0.00	8.00
ω_1	-1.69	-10.00	0.00
ω_2	-2.54	-10.00	0.00
ω_3	-2.84	-10.00	0.00
ω_4	-4.69	-10.00	0.00
ω_5	-8.48	-10.00	0.00
ω_6	-4.39	-10.00	0.00
σ	3.13	1.00	5.00
τ	0.23	0.10	0.50
S_0	0.62	0.00	1.00
I_0	0.38	0.00	1.00
$R_{1,0}$	0.00	0.00	1.00
$R_{2,0}$	0.00	0.00	1.00
$R_{3,0}$	0.00	0.00	1.00

$\hat{\theta}$ is the MLE reported by *King et al.* (2008). Three parameters were fixed ($\delta = 0.02$, $N_s = 6$ and $k = 3$) following *King et al.* (2008). Units are year^{-1} for γ, ϵ, m, β_{trend} and δ; all other parameters are dimensionless. θ_{low} and θ_{high} are the lower and upper bounds for a hyper-rectangle used to generate starting points for the search. Non-negative parameters (γ, ϵ, m, σ, τ) were logarithmically transformed for optimization. Unit scale parameters (S_0, I_0, $R_{1,0}$, $R_{2,0}$, $R_{3,0}$) were optimized on a logistic scale. These parameters were rescaled using the known population size to give the initial state variables, e.g., $S(t_0) = S_0\{S_0 + I_0 + R_{1,0} + R_{2,0} + R_{3,0}\}^{-1} P(t_0)$.

A.11 Proofs of chapter IV

A.11.1 Proof of Theorem IV.8

Let

$$R = \begin{bmatrix} \tau_0 I_{d\times d} & 0_{d\times d} & \cdots & 0_{d\times d} \\ \tau_0 I_{d\times d} & \tau_1 I_{d\times d} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \tau_0 I_{d\times d} & \tau_1 I_{d\times d} & \cdots & \tau_N I_{d\times d} \end{bmatrix}, \tag{A.31}$$

where $I_{d\times d}$ is identity matrix of dimension d and $0_{d\times d}$ is zero matrix of dimension d, then a random walk noise will be $R\tau Z_{0:N}$. From Assumption 5, $\breve{\ell}$ is four times continuously differentiable for $\theta^{[N+1]}$ and fixed N. Following similar arguments as in the proof of Proposition 9 of *Doucet et al.* (2013), given fixed $\rho > 0$, we can choose η as a function of $\min(\rho/(|R||u|), \rho/M, \delta)$, so that Lemma 1 holds. That is with $\Sigma = \text{Cov}(RZ_{0:N}) = \breve{\Psi}_N$, there exist η, and C independent of $\tau, \tau_1, ...\tau_N$ such that for

every $\tau < \eta$ we have:

$$\left|\breve{\mathbb{E}}\left(\breve{\Theta}_{0:N} - \theta^{[N+1]} \left| \breve{Y}_{1:N} = y^*_{1:N}\right.\right) - \tau^2 \breve{\Psi}_N \nabla \breve{\ell}\left(\theta^{[N+1]}\right)\right| < C\tau^4,$$

where

$$\breve{\Psi}_N = \begin{bmatrix} \tau_0^2\Psi & \tau_0^2\Psi & \cdots & \tau_0^2\Psi \\ \tau_0^2\Psi & \tau_0^2+\tau_1^2\Psi & \ddots & \tau_0^2+\tau_1^2\Psi \\ \vdots & \vdots & \ddots & \vdots \\ \tau_0^2\Psi & \tau_0^2+\tau_1^2\Psi & \cdots & \sum_{i=1}^N \tau_i^2\Psi \end{bmatrix}.$$

Since the step size τ_i is decreasing, C can be shown to be independent of $\tau_1, ...\tau_N$ by replacing $|R|$ with $d(N+1)^2\tau_0 < \infty$, which is independent of $\tau_i, i \in 1, ..N$. Note also that assumptions 1 and 2 are automatically satisfied for the multivariate normal distribution $\breve{\psi}_N$ of random variable $RZ_{0:N}$. As a result, for a random walk noise we have

$$\left|\nabla \breve{\ell}\left(\theta^{[N+1]}\right) - \tau^{-2}\breve{\Psi}_N^{-1}\breve{\mathbb{E}}\left(\breve{\Theta}_{0:N} - \theta^{[N+1]} \left| \breve{Y}_{1:N} = y^*_{1:N}\right.\right)\right| < C\tau^2.$$

An application of the Gaussian-Jordan inverse method gives,

$$\breve{\Psi}_N^{-1} = \begin{bmatrix} (\tau_0^{-2}+\tau_1^{-2})\Psi^{-1} & -\tau_1^{-2}\Psi^{-1} & \cdots & 0 \\ -\tau_1^{-2}\Psi^{-1} & (\tau_1^{-2}+\tau_2^{-2})\Psi^{-1} & \cdots & \vdots \\ 0 & -\tau_2^{-2}\Psi^{-1} & \cdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & (\tau_{N-1}^{-2}+\tau_N^{-2})\Psi^{-1} & -\tau_N^{-2}\Psi^{-1} \\ 0 & 0 & -\tau_N^{-2}\Psi^{-1} & \tau_N^{-2}\Psi^{-1} \end{bmatrix}.$$

We write $\nabla_n \breve{\ell}(\theta^{[N+1]})$ for the d-dimensional vector of partial derivatives of $\breve{\ell}(\theta^{[N+1]})$ with respect to each of the d components of θ_n. An application of the chain rule gives

the identity

$$\nabla \ell\left(\theta\right) = \sum_{n=0}^{N} \nabla_n \breve{\ell}\left(\theta^{[N+1]}\right),$$

giving rise to an inequality,

$$\left| \nabla \ell\left(\theta\right) - \tau^{-2} \sum_{n=0}^{N} \left\{ \breve{\Psi}_N^{-1} \breve{\mathbb{E}}\left(\breve{\Theta}_{0:N} - \theta^{[N+1]} | \breve{Y}_{1:N} = y_{1:N}^* \right) \right\}_n \right| < C\tau^2,$$

where $\{s\}_n$ is the entries $\{dn+1, ..., d(n+1)\}$ of a vector $s \in R^{d(N+1)}$. Decomposing the matrix multiplication by $\breve{\Psi}_N^{-1}$ into $d \times d$ blocks, we have

$$\tau^{-2} \sum_{n=0}^{N} \left\{ \breve{\Psi}_N^{-1} \breve{\mathbb{E}}\left(\breve{\Theta}_{0:N} - \theta^{[N+1]} | \breve{Y}_{1:N} = y_{1:N}^* \right) \right\}_n$$

$$= \tau^{-2} \sum_{n=0}^{N} \mathrm{SumCol}_n(\breve{\Psi}_N^{-1}) \breve{\mathbb{E}}\left(\breve{\Theta}_n - \theta | \breve{Y}_{1:N} = y_{1:N}^* \right), \tag{A.32}$$

where SumCol_n is the sum of the nth column in the $d \times d$ block construction of $\breve{\Psi}_N^{-1}$. Every column of $\breve{\Psi}_N^{-1}$ except the first sums to 0, and this special structure of $\tilde{\Psi}_N^{-1}$ gives a simple form,

$$\left| \sum_{n=0}^{N} \nabla_n \breve{\ell}\left(\theta^{[N+1]}\right) - \tau^{-2} \Psi^{-1} \tau_0^{-2} \breve{\mathbb{E}}\left(\breve{\Theta}_0 - \theta | \breve{Y}_{1:N} = y_{1:N}^* \right) \right| < C\tau^2.$$

This can be written as

$$\left| \nabla \ell\left(\theta\right) - \tau^{-2} \Psi^{-1} \tau_0^{-2} \breve{\mathbb{E}}\left(\breve{\Theta}_0 - \theta | \breve{Y}_{1:N} = y_{1:N}^* \right) \right| < C\tau^2.$$

A.11.2 Proof of Theorem IV.9

Using similar set up as above, let the random walk noise be $R\tau Z_{0:N}$ with R defined as in equation (A.31). By selecting $\breve{\psi}_N$ as multivariate normal distribution, Assumptions 4 is also satisfied. From Lemma 2, there exist η and C independent of

$\tau, \tau_1, ...\tau_N$ such that for $0 < \tau < \eta$,

$$\left|\nabla^2 \breve{\ell}\left(\theta^{[N+1]}\right) - \tau^{-4} \left[\breve{\Psi}_N^{-1}\left(\breve{\mathrm{C}}\mathrm{ov}_{\theta^{[N+1]},\tau}\left(\breve{\Theta}_{0:N}|\breve{Y}_{1:N} = y^*_{1:N}\right) - \tau^2\breve{\Psi}_N\right)\breve{\Psi}_N^{-1}\right]\right| < C\tau^2. \quad \text{(A.33)}$$

Define $\nabla^2_{s,n}\breve{\ell}\left(\theta^{[N+1]}\right)$ as

$$\nabla^2_{s,n}\breve{\ell}\left(\theta^{[N+1]}\right) = \frac{\partial^2 \breve{\ell}\left(\theta^{[N+1]}\right)}{\partial\theta_s \partial\theta_n}.$$

Applying the chain rule, we have

$$\nabla^2 \ell\left(\theta\right) = \sum_{s=0}^{N}\sum_{n=0}^{N} \nabla^2_{s,n}\breve{\ell}(\theta^{[N+1]}).$$

Adding up term in equation (A.33) we get

$$\left|\sum_{s=0}^{N}\sum_{n=0}^{N} \nabla^2_{s,n}\breve{\ell}(\theta^{[N+1]}) - \tau^{-4} \sum_{s=0}^{N}\sum_{n=0}^{N}\left[\breve{\Psi}_N^{-1}\left(\breve{\mathrm{C}}\mathrm{ov}_{\theta^{[N+1]},\tau}\left(\breve{\Theta}_{0:N}|\breve{Y}_{1:N} = y^*_{1:N}\right) - \tau^2\breve{\Psi}_N\right)\breve{\Psi}_N^{-1}\right]_{s,n}\right| < C\tau^2,$$

where $\{A\}_{s,n}$ is the entries of rows $\{ds+1, ..., d(s+1)\}$ and of columns $\{dn+1, ..., d(n+1)\}$ of a matrix $A \in R^{d(N+1)\times d(N+1)}$. Therefore,

$$\left|\nabla^2 \ell\left(\theta\right) - \tau^{-4} \sum_{s=0}^{N}\sum_{n=0}^{N}\left[\breve{\Psi}_N^{-1}\left(\breve{\mathrm{C}}\mathrm{ov}_{\theta^{[N+1]},\tau}\left(\breve{\Theta}_{0:N}|\breve{Y}_{1:N} = y^*_{1:N}\right) - \tau^2\breve{\Psi}_N\right)\breve{\Psi}_N^{-1}\right]_{s,n}\right| < C\tau^2.$$

Defining SumCol_n as in equation (A.32), we have

$$\sum_{s=0}^{N}\sum_{n=0}^{N}\left[\breve{\Psi}_N^{-1}\left(\breve{\mathrm{Cov}}_{\theta^{[N+1]},\tau}\left(\breve{\Theta}_{0:N}|\breve{Y}_{1:N}=y^*_{1:N}\right)-\tau^2\breve{\Psi}_N\right)\breve{\Psi}_N^{-1}\right]_{s,n}$$
$$=\sum_{s=0}^{N}\sum_{n=0}^{N}\mathrm{SumCol}_s(\breve{\Psi}_N^{-1})\mathrm{SumCol}_n(\breve{\Psi}_N^{-1})$$
$$\times\left(\breve{\mathrm{Cov}}_{\theta^{[N+1]},\tau}\left(\breve{\Theta}_s,\breve{\Theta}_n|\breve{Y}_{1:N}=y^*_{1:N}\right)-\sum_{k=0}^{s\wedge n}\tau_k^2\tau^2\Psi\right)\Psi^{-1}$$
$$=\left(\breve{\mathrm{Var}}_{\theta^{[N+1]},\tau}\left(\breve{\Theta}_0|\breve{Y}_{1:N}=y^*_{1:N}\right)-\tau_0^2\tau^2\Psi\right).$$

The last equality follows since $\breve{\Psi}_N^{-1}$ is symmetric matrix with block of $d\times d$ for which each colum except the first sums to 0. Thus, we obtain

$$\left|\nabla^2\ell\left(\theta\right)-\tau^{-4}\Psi^{-1}\left(\breve{\mathrm{Var}}_{\theta^{[N+1]},\tau}\left(\breve{\Theta}_0|\breve{Y}_{1:N}=y^*_{1:N}\right)-\tau_0^2\tau^2\Psi\right)\Psi^{-1}\right|<C\tau^2.$$

A.11.3 Proof of Theorem IV.10

Similar to proof of Theorem IV.8, from Lemma IV.5, there exist η and C independent of $\tau,\tau_1,...\tau_N$ such that for every $\tau<\eta$ we have,

$$\left|\nabla\breve{\ell}\left(\theta^{[N+1]}\right)-\tau^{-2}\breve{\Psi}_N^{-1}\breve{\mathbb{E}}\left(\breve{\Theta}_{0:N}-\theta^{[N+1]}\left|\breve{Y}_{1:N}=y^*_{1:N}\right.\right)\right|<C\tau^2. \tag{A.34}$$

For compactness of notation, we write $E_n = \Psi^{-1}\breve{\mathbb{E}}\left(\breve{\Theta}_n - \theta \,\middle|\, \breve{Y}_{1:N} = y^*_{1:N}\right)$ and $D_n = \nabla_n \breve{\ell}\left(\theta^{[N+1]}\right)$. Writing out terms of the vector equation in (A.34) gives

$$\begin{aligned}
(\tau_0^{-2} + \tau_1^{-2})E_0 - \tau_1^{-2}E_1 &= \tau^2 D_0 + O(\tau^4), && \text{(A.35)}\\
-\tau_1^{-2}E_0 + (\tau_1^{-2} + \tau_2^{-2})E_1 - \tau_2^{-2}E_2 &= \tau^2 D_1 + O(\tau^4), && \text{(A.36)}\\
&\vdots && \text{(A.37)}\\
-\tau_{N-1}^{-2}E_{N-2} + (\tau_{N-1}^{-2} + \tau_N^{-2})E_{N-1} - \tau_N^{-2}E_N &= \tau^2 D_{N-1} + O(\tau^4), && \text{(A.38)}\\
-\tau_N^{-2}E_{N-1} + \tau_N^{-2}E_N &= \tau^2 D_N + O(\tau^4). && \text{(A.39)}
\end{aligned}$$

Summing up (A.35) through (A.39) gives $\tau_0^{-2}E_0 = \tau^2\nabla\ell + O(\tau^4)$, as in Theorem IV.8. Substituting back into each row of (A.35) through (A.39), we get a set of equations,

$$\begin{aligned}
\tau_0^{-2}E_0 &= \tau^2 \textstyle\sum_{n=0}^N D_n + O(\tau^4),\\
\tau_1^{-2}(E_1 - E_0)) &= \tau^2 \textstyle\sum_{n=1}^N D_n + O(\tau^4),\\
&\vdots\\
\tau_{N-1}^{-2}(E_{N-1} - E_{N-2}) &= \tau^2 \textstyle\sum_{n=N-1}^N D_n + O(\tau^4),\\
\tau_N^{-2}(E_N - E_{N-1}) &= \tau^2 D_N + O(\tau^4).
\end{aligned}$$

Solving for E_n we get

$$\begin{aligned}
E_0 &= \tau^2\tau_0^2 \textstyle\sum_{n=0}^N D_n + O(\tau^4),\\
E_1 &= \tau^2\left(\tau_0^2 \textstyle\sum_{n=0}^N D_n + \tau_1^2 \sum_{n=1}^N D_n\right) + O(\tau^4),\\
&\vdots\\
E_{N-1} &= \tau^2\left(\tau_0^2 \textstyle\sum_{n=0}^N D_n + \tau_1^2 \sum_{n=1}^N D_n + \ldots + \tau_{N-1}^2 \sum_{n=N-1}^N D_n\right) + O(\tau^4),\\
E_N &= \tau^2\left(\tau_0^2 \textstyle\sum_{n=0}^N D_n + \tau_1^2 \sum_{n=1}^N D_n + \ldots + \tau_N^2 D_N\right) + O(\tau^4).
\end{aligned}$$

Using our assumption that for all $n = 1 \dots N$, $\tau_n = O(\tau^2)$, we get that $E_n = E_0 + O(\tau^4)$, from which we can conclude that

$$\frac{1}{N+1} \sum_{n=0}^{N} E_n = E_0 + O(\tau^4).$$

Application of Theorem IV.8 then completes the proof.

A.11.4 Proof of Theorem IV.11

From Lemma IV.6, we have

$$\left| \breve{\nabla}^2 \breve{\ell} \left(\theta^{[N+1]} \right) \right.$$
$$\left. -\tau^{-4} \left[\breve{\Psi}_N^{-1} \left(\breve{\mathrm{C}}\mathrm{ov}_{\theta^{[N+1]},\tau} \left(\breve{\Theta}_{0:N} | \breve{Y}_{1:N} = y^*_{1:N} \right) - \tau^2 \breve{\Psi}_N \right) \breve{\Psi}_N^{-1} \right] \right| < C\tau^2.$$

For compact notation, we write

$$\breve{C}ov_{s,n} = \breve{\mathrm{C}}\mathrm{ov} \left(\breve{\Theta}_s, \breve{\Theta}_n | \breve{Y}_{1:N} = y^*_{1:N} \right) - \tau_s \tau_n \tau^2 \Psi$$

and

$$\nabla^2_{s,n} = \nabla^2_{s,n} \breve{\ell} \left(\theta^{[N+1]} \right).$$

From the diagonal terms of the above matrix norm inequality, we derive $N+1$ equations,

$$\breve{\mathrm{C}}\mathrm{ov}_{0,0} = \tau^4 \left[\breve{\Psi}_N \nabla^2 \breve{\ell} \left(\theta^{[N+1]} \right) \breve{\Psi}_N \right]_{0,0} + O(\tau^6), \tag{A.40}$$

$$\breve{\mathrm{C}}\mathrm{ov}_{1,1} = \tau^4 \left[\breve{\Psi}_N \nabla^2 \breve{\ell} \left(\theta^{[N+1]} \right) \breve{\Psi}_N \right]_{1,1} + O(\tau^6), \tag{A.41}$$

$$\vdots \tag{A.42}$$

$$\breve{\mathrm{C}}\mathrm{ov}_{N-1,N-1} = \tau^4 \left[\breve{\Psi}_N \nabla^2 \breve{\ell} \left(\theta^{[N+1]} \right) \breve{\Psi}_N \right]_{N-1,N-1} + O(\tau^6), \tag{A.43}$$

$$\breve{\mathrm{C}}\mathrm{ov}_{N,N} = \tau^4 \left[\breve{\Psi}_N \nabla^2 \breve{\ell} \left(\theta^{[N+1]} \right) \breve{\Psi}_N \right]_{N,N} + O(\tau^6). \tag{A.44}$$

Using (A.40) through (A.44), and expanding out a matrix multiplication, we get

$$\left[\breve{\Psi}_N \nabla^2 \breve{\ell}\left(\theta^{[N+1]}\right) \breve{\Psi}_N\right]_{n,n} =$$
$$\Psi^2 \sum_{j=0}^{n} \left(\sum_{k=0}^{i} \tau_k^2\right) \left[\sum_{i=0}^{n} \left(\sum_{k=0}^{i} \tau_k^2\right) \nabla_{i,j}^2 \breve{\ell} + \sum_{i=n+1}^{N} \left(\sum_{k=0}^{n} \tau_k^2\right) \nabla_{i,j}^2 \breve{\ell}\right] +$$
$$\Psi^2 \sum_{j=n+1}^{N} \left(\sum_{k=0}^{n} \tau_k^2\right) \left[\sum_{i=0}^{n} \left(\sum_{k=0}^{i} \tau_k^2\right) \nabla_{i,j}^2 \breve{\ell} + \sum_{i=n+1}^{N} \left(\sum_{k=0}^{n} \tau_k^2\right) \nabla_{i,j}^2 \breve{\ell}\right].$$

Using our assumption that for all $n = 1 \ldots N$, $\tau_n = O(\tau^2)$, we get that

$$\breve{\mathrm{C}}\mathrm{ov}_{n,n} = \breve{\mathrm{C}}\mathrm{ov}_{0,0} + O(\tau^6),$$

from which we can conclude that

$$\frac{1}{N+1} \sum_{n=0}^{N} \breve{\mathrm{C}}\mathrm{ov}_{n,n} = \breve{\mathrm{C}}\mathrm{ov}_{0,0} + O(\tau^6).$$

An application of Theorem IV.9 then completes the proof.

A.12 Comparison of methods on the toy example

We continue with the toy example in the main text, a bivariate, linear, Gaussian discrete-time process. Fig. A.4 shows the results of 40 Monte Carlo replications so that we can see the clustering of the MLE estimates around the true MLE, corresponding to Fig. 1 in the main text. The computations in Fig. A.4 match the setup in the main text. For IS2, most of the replications clustered near the true MLE while none of them stays in a lower likelihood region. Fig. 1, in the main text, can be viewed as a statistical summary of Fig. A.4, with 200 Monte Carlo replications. These results indicate that IS2 is clearly the best of the investigated methods for this test.

We also checked how the methods compared when given additional computational

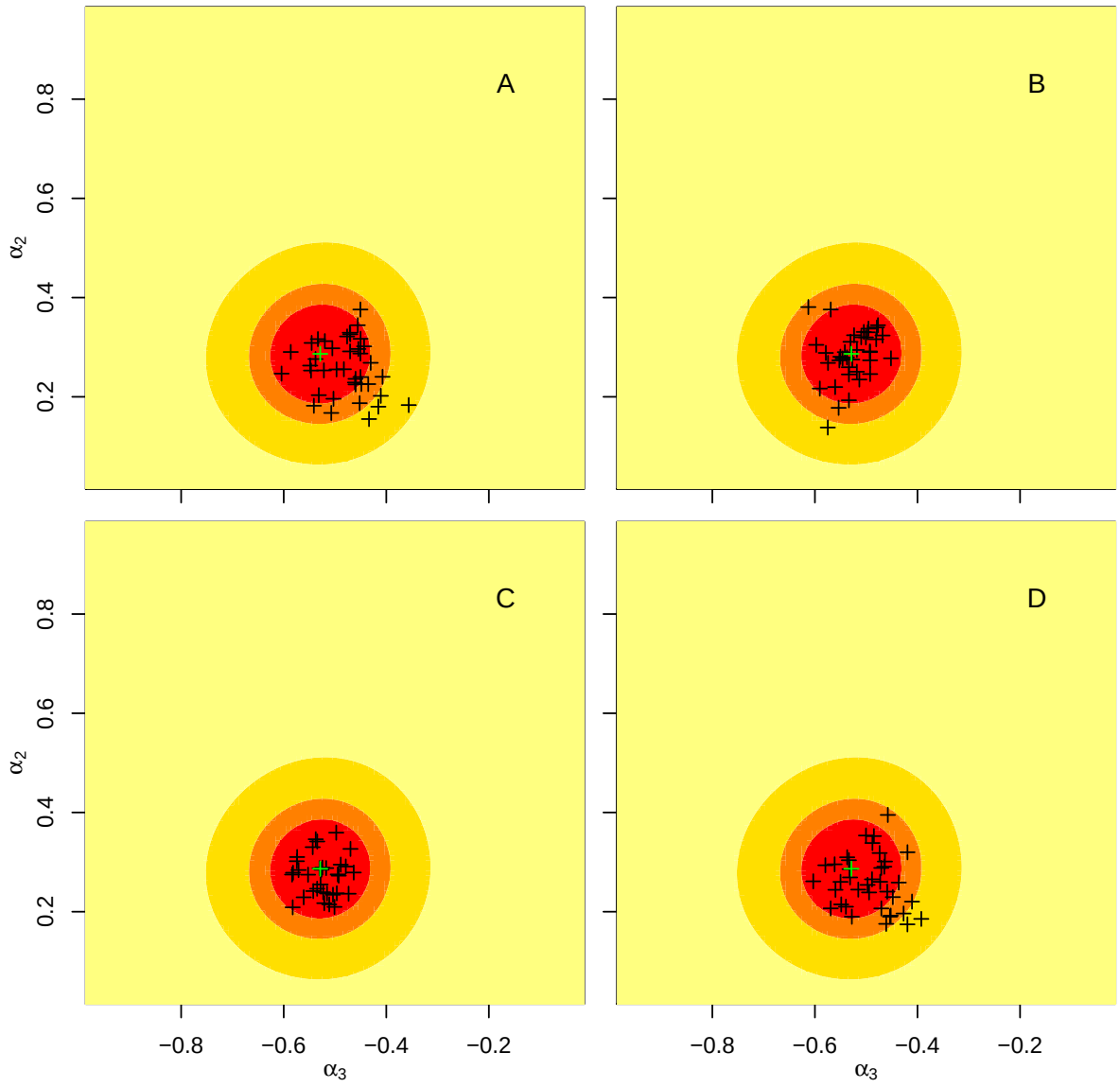

Figure A.4: Comparison of different estimators. The likelihood surface for the linear, Gaussian model, with likelihood within 2 log units of the maximum shown in red, within 4 log units in orange, within 10 log units in yellow, and lower in light yellow. The location of the MLE is marked with a green cross. The black crosses show final points from 40 Monte Carlo replications of the estimators: (A) IF1 method; (B) IF2 method; (C) IS2 method; (D) RIS1 method. Each method, except RIS1, was started uniformly over the rectangle shown, with $M = 25$ iterations, $N = 1000$ particles, and a random walk standard deviation decreasing from 0.02 geometrically to 0.011 for both α_2 and α_3. We use bigger random walk standard deviations for RIS1. Specifically random walk standard deviations decrease from 0.23 geometrically to 0.125 for both α_2 and α_3.

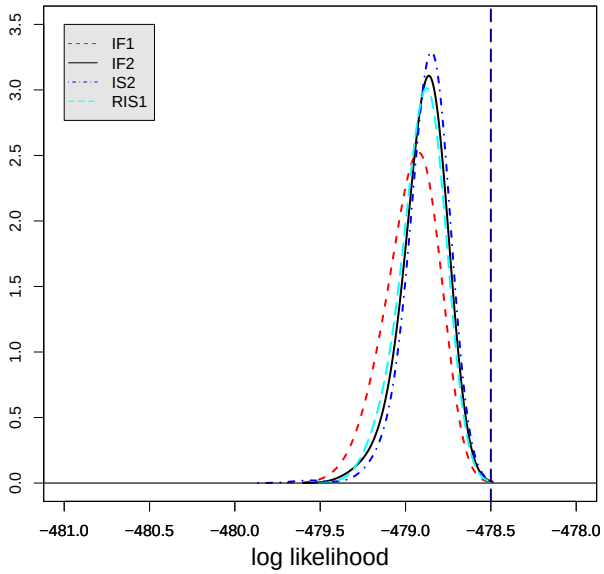

Figure A.5: The distributions of likelihoods corresponding to Monte Carlo MLE approximations estimated by IF1, IF2, RIS1 and IS2 methods for toy model. The MLE is shown as a dashed vertical line (dark blue in electronic version). The optimizations were started from 200 randomly uniform initial values over a rectangle.

resources, setting $M = 100$ iterations and $J = 10000$ particles, with the random walk standard deviation decreasing geometrically from 0.23 down to 0.0207 for RIS1 and from 0.02 down to 0.0018 for other methods. In this situation, IS2 is better than both IF2 and RIS1, and IF1 performed substantially worse than the other methods (Fig. A.5). All four of these methods have comparable computational demands for given M and J. IS1 requires substantially more computational resources, and we did not compute it for this comparison.

A.13 Algorithms IS1 and RIS1

The pseudo-code in Algorithm 9 corresponds to the iterated smoothing algorithm of *Doucet et al.* (2013). The computational complexity of approach in *Doucet et al.* (2013) is $\mathcal{O}(LN)$, the algorithm is expected to be slow, especially when computing covariance of every pair of time points with distance smaller than L. We also

propose a variant of IS1 using a computationally convenient approximation to this covariance; we call this method reduced IS1 (RIS1). reduced iterated smoothing algorithm of *Doucet et al.* (2013), called RIS1. RIS1 avoids the computational expense of computing covariances at different lags by simply ignoring these terms. This makes pseudo-code for RIS1 look more like IS2, but with white noise parameter perturbations in place of random walk perturbations. Specifically, RIS1 is a modification of IS2 for which, at line 5 in Algorithm 1, we do not update $\Theta^F_{t-1,n}$. For IS2, these covariance terms cancel in the theoretical analysis. However, there is no theorem to support RIS1 and it is only justified heuristically based on the observation that covariance between different time points may be small in practice. RIS1 is not presented for its theoretical interest, but for empirical interest in providing a computationally efficient benchmark for comparing between white noise and random walk noise.

Algorithm 9: Iterating smoothing (IS1)

input: Starting parameter, θ_0; simulator for $f_{X_0}(x_0\,;\theta)$; simulator for $f_{X_n|X_{n-1}}(x_n\,|\,x_{n-1}\,;\theta)$; evaluator for $f_{Y_n|X_n}(y_n\,|\,x_n\,;\theta)$; data, $y^*_{1:N}$; labels, $I \subset \{1,\ldots,p\}$, designating IVPs; fixed lag, L, for estimating initial value parameters (IVPs); number of particles, J, number of iterations, M; cooling rate, $0 < a < 1$; perturbation scales, $\sigma_{1:p}$; initial scale multiplier, $C > 0$.

for m *in* $1\!:\!M$ **do**
  Initialize: $[\Theta^F_{0,j}]_i \sim \text{Normal}\,([\theta_0]_i, (Ca^{m-1}\sigma_i)^2)$ for i in $1\!:\!p$, j in $1:J$.
  Initialize states: simulate $X^F_{0,j} \sim f_{X_0}\big(\,\cdot\,;\Theta^F_{0,j}\big)$ for j in $1\!:\!J$.
  Initialize filter mean for parameters: $\bar{\theta}_0 = \theta_0$.
  for n *in* $1:N$ **do**
    Perturb: $\big[\Theta^P_{n,j}\big]_i \sim \mathcal{N}\big(\big[\Theta^F_{n-1,j}\big]_i, (c^{m-1}\sigma_i)^2\big)$ for $i \notin I$, j in $1:J$.
    Simulate prediction particles: $X^P_{n,j} \sim f_n\big(x_n|X^F_{n-1,j};\Theta^P_{n,j}\big)$ for j in $1:J$.
    Evaluate weights: $w(n,j) = g_n(y^*_n|X^P_{n,j};\Theta^P_{n,j})$ for j in $1:J$.
    Normalize weights: $\breve{w}(n,j) = w(n,j)/\sum_{u=1}^J w(n,u)$.
    Apply 3 to select indices $k_{1:J}$ with $P\{k_u = j\} = \breve{w}\,(n,j)$.
    Resample particles: $X^F_{n,j} = X^P_{n,k_j}$ and $\Theta^F_{n,j} = \Theta^P_{n,k_j}$ for j in $1:J$.
    Define and store ancestor let $a_1(n,k_j) = j$,
    $a_{l+1}(n,j) = a_1(n-l, a_l(n,j))$ for j in $1:J$, l in $1:L-1$
  end
  for n *in* $1:N$ **do**
    Smooth mean: $\bar{\theta}^L_{n-L} = \sum_{j=1}^J \breve{w}(n,j)\Theta^P_{n-L,a_L(n,j)}$ if $n > L$.
    for l *in* $n : \min(n+L, N)$ **do**
      Smooth Covariance: $C^m_{n-L,l-L} = \sum_j \breve{w}(n,j)\big(\Theta^P_{n-L,a_L(n,j)} - \bar{\theta}^L_{n-L}\big)\big(\Theta^P_{l-L,a_L(n,j)} - \bar{\theta}^L_{l-L}\big)^\top$ if $n > L$.
    end
  end
  for j *in* $0:L$ **do**
    Smooth mean: $\bar{\theta}^L_{n-L} = \sum_{j=1}^J \breve{w}(n,j)\Theta^P_{n-L,a_L(n,j)}$ if $n > L$.
    for l *in* $n : \min(n+L, N)$ **do**
      Smooth Covariance: $C^m_{n-L,l-L} = \sum_j \breve{w}(n,j)\big(\Theta^P_{n-L,a_L(n,j)} - \bar{\theta}^L_{n-L}\big)\big(\Theta^P_{l-L,a_L(n,j)} - \bar{\theta}^L_{l-L}\big)^\top$ if $n > L$.
    end
  end
  Update: $S_m = c^{-2(m-1)}\Psi^{-1}\sum_{n=1}^N \big[\,\big(\bar{\theta}^L_n - \theta_{m-1}\big)\,\big]$.
  $I_m = -c^{-4(m-1)}\Psi^{-1}\left[\sum_{n=1}^N \left(C^m_{n,n} - c^{2(m-1)}\Psi + 2\sum_{s=n+1}^{(s+L)\wedge N} C^m_{s,n}\right)\right]\Psi^{-1}$.
  Update non-IVP parameters: $\theta_m = \theta_{m-1} + I_m^{-1}S_m$.
  Update IVPs: $\big[\theta_m\big]_i = \frac{1}{J}\sum_{j=1}^J \big[\Theta^F_{L,j}\big]_i$ for $i \in I$.
end

output: Monte Carlo maximum likelihood estimate, θ_M.
complexity: $\mathcal{O}(JL^2M)$

BIBLIOGRAPHY

BIBLIOGRAPHY

Alonso, P. L., et al. (2011), A research agenda to underpin malaria eradication, *PLoS Med*, *8*(1), e1000,406.

Anderson, B. D., and J. B. Moore (1979), *Optimal Filtering*, Prentice-Hall, New Jersey.

Anderson, R. M., and R. M. May (1991), *Infectious Diseases of Humans*, vol. 1, Oxford University Press, Oxford.

Andrieu, C., and G. O. Roberts (2009), The pseudo-marginal approach for efficient computation, *Annals of Statistics*, *37*(2), 697–725.

Andrieu, C., A. Doucet, and R. Holenstein (2010), Particle Markov chain Monte Carlo methods, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *72*(3), 269–342.

Aron, J. L., and R. M. May (1982), The population dynamics of malaria, in *The Population Dynamics of Infectious Diseases: Theory and Applications*, pp. 139–179, Springer, New York.

Artzner, D. F. E. J.-M., P., and D. Heath (1997), Lévy processes and stochastic calculus, *Thinking coherently, Risk*, *10*, 68–71.

Artzy-Randrup, Y., D. Alonso, and M. Pascual (2010), Transmission intensity and drug resistance in malaria population dynamics: Implications for climate change, *PloS one*, *5*(10), e13,588.

Arulampalam, M. S., S. Maskell, N. Gordon, and T. Clapp (2002), A tutorial on particle filters for online nonlinear, non-Gaussian Bayesian tracking, *IEEE Transactions on Signal Processing*, *50*, 174 – 188.

Baeza, A., M. J. Bouma, R. Dhiman, and M. Pascual (2014), Malaria control under unstable dynamics: Reactive vs. climate-based strategies, *Acta tropica*, *129*, 42–51.

Barndorff-Nielsen, O. E., and D. R. Cox (1994), Inference and asymptotics.

Beaumont, M. A. (2010), Approximate Bayesian computation in evolution and ecology, *Annual Review of Ecology, Evolution, and Systematics*, *41*, 379–406.

Bengtsson, T., P. Bickel, and B. Li (2008), 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*, edited by T. Speed and D. Nolan, pp. 316–334, Institute of Mathematical Statistics, Beachwood, OH.

Bhadra, A. (2010), Discussion of 'particle Markov chain Monte Carlo methods' by C. Andrieu, A. Doucet and R. Holenstein, *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, *72*, 314–315.

Bhadra, A., E. L. Ionides, K. Laneri, M. Pascual, M. Bouma, and R. C. Dhiman (2011), Malaria in Northwest India: Data analysis via partially observed stochastic differential equation models driven by Lévy noise, *Journal of the American Statistical Association*, *106*, 440–451.

Blackwood, J. C., D. A. T. Cummings, H. Broutin, S. Iamsirithaworn, and P. Rohani (2013a), Deciphering the impacts of vaccination and immunity on pertussis epidemiology in Thailand, *Proceedings of the National Academy of Sciences of the USA*, *110*, 9595–9600.

Blackwood, J. C., D. G. Streicker, S. Altizer, and P. Rohani (2013b), Resolving the roles of immunity, pathogenesis, and immigration for rabies persistence in vampire bats, *Proceedings of the National Academy of Sciences of the USA*, *110*, 20,83720,842.

Blake, I. M., R. Martin, A. Goel, N. Khetsuriani, J. Everts, C. Wolff, S. Wassilak, R. B. Aylward, and N. C. Grassly (2014), The role of older children and adults in wild poliovirus transmission, *Proceedings of the National Academy of Sciences of the USA*, *111*(29), 10,604–10,609.

Bossy, M., E. Gobet, and D. Talay (2004), A symmetrized Euler scheme for an efficient approximation of reflected diffusions, *Journal of Applied Probability*, pp. 877–889.

Bouma, M. J., and H. J. van der Kaay (1994), Epidemic malaria in India and the El Niño, pp. 1638–1639, southern Oscillation. The Lancet, 344(8937):.

Bretó, C. (2014), On idiosyncratic stochasticity of financial leverage effects, *Statistics & Probability Letters*, *91*, 20–26.

Bretó, C., and E. L. Ionides (2011), Compound Markov counting processes and their applications to modeling infinitesimally over-dispersed systems, *Stoch. Process. Their Appl.*, *121*(11), 2571–2591.

Bretó, C., D. He, E. L. Ionides, and A. A. King (2009), Time series analysis via mechanistic models, *Annals of Applied Statistics*, *3*, 319–348.

Camacho, A., S. Ballesteros, A. L. Graham, F. Carrat, O. Ratmann, and B. Cazelles (2011), Explaining rapid reinfections in multiple-wave influenza outbreaks: Tristan da Cunha 1971 epidemic as a case study, *Proceedings of the Royal Society of London, Series B*, *278*(1725), 3635–3643.

Cappé, O., S. Godsill, and E. Moulines (2007), An overview of existing methods and recent advances in sequential Monte Carlo, *Proc. IEEE*, *95*, 899–924.

Chambers, J. M. (1998), *Programming with Data*, New York, iSBN 0-387-98503-4.

Chopin, N., P. E. Jacob, and O. Papaspiliopoulos (2013), SMC^2: an efficient algorithm for sequential analysis of state space models, *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, *75*(3), 397–426.

Commandeur, J. J. F., S. J. Koopman, and M. Ooms (2011), Statistical software for state space methods, *Journal of Statistical Software*, *41*(1), 1–18.

Coulson, T., P. Rohani, and M. Pascual (2004), Skeletons, noise and population growth: The end of an old debate?, *Trends Ecol. Evol.*, *19*, 359–364.

Crisan, D., and A. Doucet (2002), A survey of convergence results on particle filtering methods for practitioners, *IEEE Transactions on Signal Processing*, *50*, 736–746.

Dahlin, J., F. Lindsten, and T. B. Schön (2015), Particle Metropolis-Hastings using gradient and Hessian information, *Statistics and Computing*, *25*(1), 81–92.

Del Moral, P., and A. Doucet (2004), Particle motions in absorbing medium with hard and soft obstacles, *Stochastic Analysis and Applications*, *22*, 1175–1207.

Dietz, K., L. Molineaux, and A. Thomas (1974), A malaria model tested in the African savannah, *Bulletin of the World Health Organization*, *50*(3-4), 347.

Douc, R., O. Cappé, and E. Moulines (2005), Comparison of resampling schemes for particle filtering, in *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005*, pp. 64–69, IEEE.

Doucet, A., and A. Johansen (2009), A tutorial on particle filtering and smoothing: Fifteen years later, in *Oxford Handbook of Nonlinear Filtering*, edited by D. Crisan and B. Rozovsky, Oxford University Press.

Doucet, A., N. De Freitas, and N. Gordon (2001), *Sequential Monte Carlo methods in practice*, Springer, New York.

Doucet, A., S. J. Godsill, and C. P. Robert (2002), Marginal maximum a posteriori estimation using Markov chain Monte Carlo, *Statistics and Computing*, *12*, 77–84.

Doucet, A., P. E. Jacob, and S. Rubenthaler (2013), Derivative-free estimation of the score vector and observed information matrix with application to state-space models, *ArXiv:1304.5768*.

Earn, D. J., D. He, M. B. Loeb, K. Fonseca, B. E. Lee, and J. Dushoff (2012a), Effects of school closure on incidence of pandemic influenza in Alberta, Canada, *Annals of Internal Medicine*, *156*(3), 173–181.

Earn, D. J. D., D. He, M. B. Loeb, K. Fonseca, B. E. Lee, and J. Dushoff (2012b), Effects of school closure on incidence of pandemic influenza in Alberta, Canada, *The Annals of Internal Medicine*, *156*(3), 173–181.

Ellner, S. P., B. A. Bailey, G. V. Bobashev, A. R. Gallant, B. T. Grenfell, and D. W. Nychka (1998), Noise and nonlinearity in measles epidemics: Combining mechanistic and statistical approaches to population modeling, *American Naturalist*, *151*(5), 425–440.

Eveson, S. P. (1995), Hilbert's projective metric and the spectral properties of positive linear operators, *3*, 411–440.

Fletcher, R. (1980), *Practical methods of optimization*, vol. 1, Wiley.

Gaetan, C., and J.-F. Yao (2003), A multiple-imputation Metropolis version of the EM algorithm, *Biometrika*, *90*(3), 643–654.

Genolini, C. (2008), A (not so) short introduction to S4, *Tech. rep.*, The R Project for Statistical Computing.

Gill, P. E., W. Murray, and M. H. Wright (1981), *Practical Optimization*, Academic, London.

Gillespie, D. T. (1977), Exact stochastic simulation of coupled chemical reactions, *Journal of Physical Chemistry*, *81*(25), 2340–2361.

Gompertz, B. (1825), On the nature of the function expressive of the law of human mortality, and on a new mode of determining the value of life contingencies, *Philosophical Transactions of the Royal Society of London*, *115*, 513–583.

Gordon, N. J., D. J. Salmond, and A. F. Smith (1993), Novel approach to nonlinear/non-gaussian bayesian state estimation, in *IEE Proceedings F (Radar and Signal Processing)*, vol. 140, pp. 107–113, IET.

He, D., E. L. Ionides, and A. A. King (2010), Plug-and-play inference for disease dynamics: Measles in large and small populations as a case study, *Journal of the Royal Society Interface*, *7*(43), 271–283.

He, D., J. Dushoff, T. Day, J. Ma, and D. J. D. Earn (2013), Inferring the causes of the three waves of the 1918 influenza pandemic in England and Wales., *Proceedings of the Royal Society of London, Series B*, *280*(1766), 20131,345.

Ingber, L. (1993), Simulated annealing: Practice versus theory, *Mathematical and Computer Modelling*, *18*, 29–57.

Ionides, E. L. (2011), Discussion on "Feature matching in time series modeling" by Y. Xia and H. Tong, *Statistical Science*, *26*, 49–52.

Ionides, E. L., C. Bretó, and A. A. King (2006), Inference for nonlinear dynamical systems, *Proceedings of the National Academy of Sciences of the USA*, *103*, 18,438–18,443.

Ionides, E. L., A. Bhadra, Y. Atchadé, and A. King (2011), Iterated filtering, *Annals of Statistics*, *39*, 1776–1802.

Ionides, E. L., D. Nguyen, Y. Atchadé, S. Stoev, and A. A. King (2015), Inference for dynamic and latent variable models via iterated, perturbed Bayes maps, *Proceedings of the National Academy of Sciences of the USA*, *112*(3), 719–724.

Jacod, J. (2004), The Euler scheme for Lévy driven stochastic differential equations: limit theorems, *The Annals of Probability*, *32*(3), 1830–1872.

Jacod, J., and A. N. Shiryaev (1987), *Limit theorems for stochastic processes*, Springer-Verlag, Berlin.

Jacquier, E., M. Johannes, and N. Polson (2007), MCMC maximum likelihood for latent state models, *Journal of Econometrics*, *137*(2), 615–640.

Johansen, A. M., and A. Doucet (2008), A note on the auxiliary particle filter, *78*, 1498–1504.

Johnson, S. G. (2014), *The NLopt Nonlinear-Optimization Package*, version 2.4.2.

Julier, S., and J. Uhlmann (2004), Unscented filtering and nonlinear estimation, *Proc. IEEE*, *92*, 401–422.

Kantas, N., A. Doucet, S. S. Singh, and J. M. Maciejowski (2015), On particle methods for parameter estimation in state-space models, arXiv:1412.8695.

Keeling, M., and P. Rohani (2009), *Modeling Infectious Diseases in Humans and Animals*, Princeton University Press, Princeton, NJ.

Kendall, B. E., C. J. Briggs, W. W. Murdoch, P. Turchin, S. P. Ellner, E. McCauley, R. M. Nisbet, and S. N. Wood (1999), Why do populations cycle? A synthesis of statistical and mechanistic modeling approaches, *Ecology*, *80*(6), 1789–1805.

Kendall, B. E., S. P. Ellner, E. McCauley, S. N. Wood, C. J. Briggs, W. W. Murdoch, and P. Turchin (2005), Population cycles in the pine looper moth: Dynamical tests of mechanistic hypotheses, *Ecological Monographs*, *75*(2), 259–276.

Kermack, W. O., and A. G. McKendrick (1927), A contribution to the mathematical theory of epidemics, *Proceedings of the Royal Society of London, Series A*, *115*, 700–721.

Kevrekidis, I. G., C. W. Gear, and G. Hummer (2004), Equation-free: The computer-assisted analysis of complex, multiscale systems, *American Institute of Chemical Engineers Journal*, *50*, 1346–1354.

Kiefer, J., and J. Wolfowitz (1952), Stochastic estimation of the maximum of a regression function, pp. 462–466, annals of Mathematical Statistics, 23:.

King, A. A. (2008), *subplex: Subplex Optimization Algorithm*, R package, version 1.1-4.

King, A. A., E. L. Ionides, M. Pascual, and M. J. Bouma (2008), Inapparent infections and cholera dynamics, *Nature*, *454*, 877–880.

King, A. A., M. Domenech de Celle, F. M. G. Magpantay, and P. Rohani (2015a), Avoidable errors in the modelling of outbreaks of emerging pathogens, with special reference to Ebola, *Proceedings of the Royal Society of London, Series B*, *282*, 20150,347.

King, A. A., D. Nguyen, and E. L. Ionides (2015b), Statistical inference for partially observed Markov processes via the R package pomp, *Journal of Statistical Software*, p. to appear.

King, A. A., D. Nguyen, and E. L. Ionides (2015c), Statistical inference for partially observed Markov processes via the R package pomp, *Journal of Statistical Software*, *to appear*.

King, A. A., M. Domenech de Cellès, F. M. G. Magpantay, and P. Rohani (in press), Avoidable errors in the modeling of outbreaks of emerging pathogens, with special reference to Ebola, *Proceedings of the Royal Society of London. Series B*.

King, A. A., et al. (2014), *pomp: Statistical Inference for Partially Observed Markov Processes*, R package, version 0.53-1.

Kiszewski, A. E., and A. Teklehaimanot (2004), A review of the clinical and epidemiologic burdens of epidemic malaria, pp. 128–135, american Journal of Tropical Medicine and Hygiene, 71 (2):.

Kitagawa, G. (1998), A self-organising state-space model, *Journal of the American Statistical Association*, *93*, 1203–1215.

Kloeden, P. E., and E. Platen (1999), *Numerical Soluion of Stochastic Differential Equations*, 3rd ed., Springer, New York.

Kumar, A., N. Valecha, T. Jain, and A. P. Dash (2007), Burden of malaria in India: Retrospective and prospective view, *The American Journal of Tropical Medicine and Hygiene*, *77*(6 Suppl), 69–78.

Kushner, H. J., and D. S. Clark (1978), *Stochastic Approximation Methods for Constrained and Unconstrained Systems*, Springer-Verlag, New York.

Laneri, K., A. Bhadra, E. L. Ionides, M. Bouma, R. C. Dhiman, R. S. Yadav, and M. Pascual (2010), Forcing versus feedback: Epidemic malaria and monsoon rains in Northwest India, *PLoS Computational Biology*, *6*(9), e1000,898.

Laneri, K., R. E. Paul, A. Tall, J. Faye, F. Diene-Sarr, C. Sokhna, J.-F. Trape, and X. Rodó (2015), Dynamical malaria models reveal how immunity buffers effect of climate variability, *Proceedings of the National Academy of Sciences of the USA*, *112*(28), 8786–8791.

Lavine, J. S., and P. Rohani (2012), Resolving pertussis immunity and vaccine effectiveness using incidence time series, *Expert Review of Vaccines*, *11*, 1319–1329.

Lavine, J. S., A. A. King, V. Andreasen, and O. N. Bjørnstad (2013a), Immune boosting explains regime-shifts in prevaccine-era pertussis dynamics, *PLoS ONE*, *8*(8), e72,086.

Lavine, J. S., A. A. King, V. Andreasen, and O. N. Bjrnstad (2013b), Immune boosting explains regime-shifts in prevaccine-era pertussis dynamics, *PLoS ONE*, *8*(8), e72,086.

Le Gland, F., and N. Oudjane (2004), Stability and uniform approximation of nonlinear filters using the Hilbert metric and application to particle filters, *14*, 144–187.

Lele, S. R., B. Dennis, and F. Lutscher (2007), Data cloning: Easy maximum likelihood estimation for complex ecological models using Bayesian Markov chain Monte Carlo methods, *Ecology Letters*, *10*(7), 551–563.

Lele, S. R., K. Nadeem, and B. Schmuland (2010), Estimability and likelihood inference for generalized linear mixed models using data cloning, *Journal of the American Statistical Association*, *105*, 1617–1625.

Lindström, E., E. L. Ionides, J. Frydendall, and H. Madsen (2012), Efficient iterated filtering, in *16th IFAC Symposium on System Identification*.

Liu, J., and M. West (2001), Combining parameter and state estimation in simulation-based filtering, in *Sequential Monte Carlo Methods in Practice*, edited by A. Doucet, N. de Freitas, and N. J. Gordon, pp. 197–224, Springer, New York.

Liu, J. S. (2001), *Monte Carlo Strategies in Scientific Computing*, Springer, New York.

Liu, J. S., and R. Chen (1998), Sequential Monte Carlo methods for dynamic systems, *Journal of the American Statistical Association*, *93*(443), 1032–1044.

Lloyd, A. L. (2001), Realistic distributions of infectious periods in epidemic models: Changing patterns of persistence and dynamics, *Theoretical Population Biology*, *60*, 59–71.

Macdonald, G. (1957), *The Epidemiology and Control of Malaria*, Oxford University Press, Oxford.

Marin, J.-M., P. Pudlo, C. P. Robert, and R. J. Ryder (2012), Approximate Bayesian computational methods, *Statistics and Computing*, *22*(6), 1167–1180.

Martinez-Bakker, M., A. A. King, and P. Rohani (2015), Unraveling the transmission ecology of polio, *PLoS Biology*, *13*(6), e1002,172.

Nemeth, C., P. Fearnhead, and L. Mihaylova (2013), Particle approximations of the score and observed information matrix for parameter estimation in state space models with linear computational cost, *ArXiv:1306.0735*.

Nguyen, D., and E. L. Ionides (2015), Iterated smoothing for partially observed Markov processes via the R package is2, *R Packages*.

Olsson, J., O. Cappé, R. Douc, and E. Moulines (2008), Sequential Monte Carlo smoothing with application to parameter estimation in nonlinear state space models, *Bernoulli*, *14*(1), 155–179.

Pitt, M. K., and N. Shepard (1999), Filtering via simulation: Auxillary particle filters, *Journal of the American Statistical Association*, *94*, 590–599.

Poyiadjis, G., A. Doucet, and S. S. Singh (2011), Particle approximations of the score and observed information matrix in state space models with application to parameter estimation, *Biometrika*, *98*(1), 65–80.

Protter, P., and D. Talay (1997), The Euler scheme for Lévy driven stochastic differential equations, *The Annals of Probability*, *25*(1), 393–423.

Rabiner, L. R., and B.-H. Juang (1986), An introduction to hidden Markov models, *ASSP Magazine, IEEE*, *3*(1), 4–16.

Ratmann, O., C. Andrieu, C. Wiuf, and S. Richardson (2009), Model criticism based on likelihood-free inference, with an application to protein network evolution, *Proceedings of the National Academy of Sciences of the USA*, *106*(26), 10,576–10,581.

Reddingius, J. (1971), *Gambling for Existence. A Discussion of some Theoretical Problems in Animal Population Ecology*, vol. 20 (Supplement), vi, 208 p.– pp., B. J. Brill, Leiden.

Reuman, D. C., R. A. Desharnais, R. F. Costantino, O. S. Ahmad, and J. E. Cohen (2006), Power spectra reveal the influence of stochasticity on nonlinear population dynamics, *Proceedings of the National Academy of Sciences of the USA*, *103*(49), 18,860–18,865.

Revolution Analytics, and S. Weston (2014), *foreach: Foreach Looping Construct for R*, R package version 1.4.2.

Ricker, W. E. (1954), Stock and recruitment, *Journal of the Fisheries Research Board of Canada*, *11*, 559–623.

Robbins, H., and S. Monro (1951), A stochastic approximation method, *Ann. Math. Statist*, *22*.

Robbins, H., and D. Siegmund (1985), A convergence theorem for non negative almost supermartingales and some applications, in *Herbert Robbins Selected Papers*, pp. 111–135, Springer.

Romero-Severson, E., E. Volz, J. Koopman, T. Leitner, and E. Ionides (2015), Dynamic variation in sexual contact rates in a cohort of HIV-negative gay men, *American Journal of Epidemiology*, p. kwv044.

Ross, R. (1910), *The Prevention of Malaria*, Dutton, Boston.

Rowan, T. (1990), Functional stability analysis of numerical algorithms, Ph.D. thesis, Department of Computer Sciences, University of Texas at Austin.

Roy, M., M. J. Bouma, E. L. Ionides, R. C. Dhiman, and M. Pascual (2013), The potential elimination of plasmodium vivax malaria by relapse treatment: Insights from a transmission model and surveillance data from NW India, *PLoS Neglected Tropical Diseases*, *7*(1), e1979.

Shaman, J., and A. Karspeck (2012), Forecasting seasonal outbreaks of influenza, *Proceedings of the National Academy of Sciences of the USA*, *109*, 20,425–20,430.

Shrestha, S., A. A. King, and P. Rohani (2011), Statistical inference for multi-pathogen systems, *PLoS Computational Biology*, *7*(8), e1002,135.

Shrestha, S., B. Foxman, D. M. Weinberger, C. Steiner, C. Viboud, and P. Rohani (2013), Identifying the interaction between influenza and pneumococcal pneumonia using incidence data, *Science Translational Medicine*, *5*(191), 191ra84.

Shumway, R. H., and D. S. Stoffer (2006), Time series regression and exploratory data analysis, *Time Series Analysis and Its Applications: With R Examples*, pp. 48–83.

Sisson, S. A., Y. Fan, and M. M. Tanaka (2007), Sequential Monte Carlo without likelihoods, *Proceedings of the National Academy of Sciences of the USA*, *104*(6), 1760–1765.

Smith, A. A. (1993), Estimating nonlinear time-series models using simulated vector autoregression, *Journal of Applied Econometrics*, *8*(S1), S63–S84.

Spall, J. C. (2003), *Introduction to Stochastic Search and Optimization*, Wiley, Hoboken.

Storvik, G. (2002), Particle filters for state-space models with the presence of unknown static parameters, *IEEE Transactions on Signal Processing*, *50*, 281–289.

Swaroop, S. (1949), Forecasting of epidemic malaria in the Punjab, India, *American Journal of Tropical Medicine and Hygiene*, *S1-29*(1), 1–17.

Toni, T., D. Welch, N. Strelkowa, A. Ipsen, and M. P. Stumpf (2009), Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems, *Journal of the Royal Society Interface*, *6*, 187–202.

Wan, E., and R. van der Merwe (2000), The unscented Kalman filter for nonlinear estimation, in *Adaptive Systems for Signal Processing, Communications, and Control, Symposium 2000*, pp. 153–158, IEEE.

Whiteley, N., N. Kantas, and A. Jasra (2012), Linear variance bounds for particle approximations of time-homogeneous Feynman–Kac formulae, *Stochastic Processes and their Applications*, *122*, 1840–1865.

Wilkinson, D. J. (2012), *Stochastic Modelling for Systems Biology*, Chapman & Hall, Boca Raton, FL.

Wood, S. N. (2001), Partially specified ecological models, *Ecological Monographs*, *71*, 1–25.

Wood, S. N. (2010), Statistical inference for noisy nonlinear ecological dynamic systems, *Nature*, *466*(7310), 1102–1104.

Xia, Y., and H. Tong (2011), Feature matching in time series modelling, *Statistical Science*, *26*(1), 21–46.

Yang, W., A. Karspeck, and J. Shaman (2014), Comparison of filtering methods for the modeling and retrospective forecasting of influenza epidemics, *PLoS computational biology*, *10*(4), e1003,583.

Yıldırım, S., S. S. Singh, T. Dean, and A. Jasra (2015), Parameter estimation in hidden markov models with intractable likelihoods using sequential Monte Carlo, *Journal of Computational and Graphical Statistics*, *24*, 846–865.

Ypma, J. (2014), *nloptr: R Interface to NLopt*, R package, version 1.0.0.