Computational Inference Algorithms for Spatiotemporal Processes and Other Complex Models

by

Joon Ha Park

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Statistics)
in the University of Michigan
2018

Doctoral Committee:

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

Joon Ha Park

joonhap@umich.edu

ORCID iD: 0000-0002-4493-7730

© Joon Ha Park 2018

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisor Professor Edward Ionides for his guidance and support throughout my PhD study. This work would not have been possible without the countless pieces of advice he offered. Whenever I inquired him about anything, he provided me with helpful feedback I very much needed. He has always guided my research with patience and careful attention, which enabled me to focus on the research topics I found most interesting. His general advice on research and teaching activities has been crucial to developing and broadening my perspectives on academia. As a statistician and professor, he has shown me a best example. I am also very grateful for the funding he provided when I needed to intensely focus on research.

I would also like to thank Professor Aaron King for his advice and support he provided numerous times. His intellectual vigor and passion for scientific inquiry have been inspiring. I learned many lessons on applying mathematical thinking to science and developing research tools for scientific research communities.

I thank Professor Yves Atchadé for his guidance on my research project. His advice allowed me to develop the research in better directions and improve the manuscript. I greatly appreciate that he has provided me with a further research opportunity on the topics I find most interesting. I am looking forward to learning a lot under his supervision.

I am also grateful for Professor Stilian Stoev for his teaching and support. He was kind to offer various kinds of support in response to my abrupt requests for help.

I would like to thank all committee members for careful review of my dissertation and valuable feedback. I appreciate the faculty, staff, and fellow graduate students at the Statistics Department for making the Department an excellent place for learning and professional growth and a friendly community I enjoyed to be a part of.

My deep gratitude goes to my parents, who made all this possible. I would also like to thank my sister Yoonjee for always being a caring sister. I would like to thank the University of Michigan hospitals and GradCare for providing wonderful medical services, which were crucial in bringing my two loved girls, Freya and Diane, to this world. It would not have been possible to finish this thesis without my wife Jimin's incredible amount of understanding and support. I would like to thank her for being the love of my life.

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF ALGORITHMS

Algorithm

ABSTRACT

Data analysis can be carried out based on a stochastic model that reflects the analyst's understanding of how the system in question behaves. The stochastic model describes where in the system randomness is present and how the randomness plays a role in generating data. The likelihood of the data defined by the model summarizes the evidence provided by observations of the system. Drawing inference from the likelihood of the data, however, can be far from being simple or straightforward, especially in modern statistical data analyses. Complex probability models and big data call for new computational methods to translate the likelihood of data into inference results. In this thesis, I present two innovations in computational inference for complex stochastic models.

The first innovation lies in the development of a method that enables inference on coupled dynamic systems that are partially observed. The high dimensionality of the model that defines the joint distribution of the coupled dynamic processes makes computational inference a challenge. I focus on the case where the probability model is not analytically tractable, which makes the computational inference even more challenging. A mechanistic model of a dynamic process that is defined via a simulation algorithm can lead to analytically intractable models. I show that algorithms that utilize the Markov structure and the mixing property of stochastic dynamic systems can enable fully likelihood based inference for these high dimensional analytically intractable models. I demonstrate theoretically that these algorithms can substantially reduce the computational cost for inference, and the reduction may be orders of magnitude in practice. Spatiotemporal dynamics of measles transmission are inferred from data collected at linked geographic locations, as an illustration that this algorithm can offer an advance in scientific inference.

The second innovation involves a generalization of the framework in which samples from a probability distribution with unnormalized density are drawn using Markov chain Monte Carlo algorithms. The new framework generalizes the widely used Metropolis-Hastings acceptance or rejection strategy. The resulting method is straightforward to implement in a broad range of MCMC algorithms, including the most frequently used ones such as random walk Metropolis, Metropolis adjusted Langevin, Hamiltonian Monte Carlo, or the bouncy particle sampler. Numerical studies show that this new framework enables flexible tuning of parameters and facilitates faster mixing of the Markov chain, especially when the target probability density has complex structure.

CHAPTER I

Introduction

Inference from the observations of the real world can be made using a stochastic model, which accounts for the randomness in data. The complexity of stochastic models varies with factors such as the complexity of the system in question, the amount of prior knowledge about the system to be incorporated, or the level of flexibility that we require in the model. Given a stochastic model, the information contained in the data is summarized in the likelihood function of the data.

Drawing inference from the likelihood function of the data may not be straightforward. The more complex the model becomes, the harder it becomes to extract information from this. Unless the likelihood function and the resulting estimators are analytically tractable, the inference will need to rely on computational techniques, through which the aspects of the random system that we are interested in are revealed and understood. Typically the computational tasks involved are the optimization of an objective function, which may be the likelihood function or an approximate version of it, or sampling from a target distribution, such as the posterior distribution of the parameters. The goal of these tasks is to numerically compute quantities that are useful in drawing inference.

The numerical computational methods may involve either deterministic or stochastic operations. The deterministic approach aims to compute certain feature of the target function or distribution using some analytical knowledge about the problem. The stochastic approach constructs artificial random processes that can be used to represent some features of interest of the target statistical object, such as likelihood functions or posterior distributions. The procedure of constructing artificial random processes in a computer is called Monte Carlo simulation. Despite the randomness in the representation, stochastic approaches can be more desirable in certain cases because they can be applied where no deterministic methods are available or can be more computationally efficient than deterministic alternatives.

1.1 Computational challenges

In this thesis, I focus on Monte Carlo approaches to numerical computation of the target statistical quantity. Monte Carlo numerical computation is a vast topic that is studied in diverse fields with different approaches. The methods for Monte Carlo numerical computation are constantly and rapidly evolving to address new challenges imposed by modern data analysis problems. This thesis concerns the following issues that lead to computational challenges.

Analytically intractable likelihood function The likelihood function may not have an analytically tractable form. The likelihood might be expressed as a high dimensional integral over a number of latent variables, which makes it impossible to evaluate it pointwise with analytical means. Models may also be implicitly defined by data-generating simulation algorithms. These models may not possess analytically tractable densities. Implicit models defined by simulation algorithms can arise frequently when we take a mechanistic approach to set up a stochastic model. Mechanistic models are based on principled understanding of the system in question, and the resulting model may only be readily represented via simulation algorithms.

High dimensionality of the model It can be very difficult to computationally represent high dimensional distributions. Monte Carlo approaches to computational inference generate samples from a target distribution, where the empirical distribution of the random sample is taken as a stochastic approximation to the target distribution. However, the number of possible states to be represented increases proportionally with the volume of the space, that is, exponentially with the dimensionality of the space. The computational cost needed for this representation can also increase exponentially. The steep increase in computational cost is closely tied to the amount of information to be represented.

Complex structure of the target function The target function may have complex structure. The level sets of the target function may have complicated geometric shapes or consist of disjoint connected sets. A target distribution may be constrained in the sense that the probability mass is concentrated in a narrow neighborhood around a lower dimensional hyperplane or manifold. Constrained distributions may arise if the variability in some directions is much smaller than that in other directions, or if random variables constituting the distribution are strongly correlated with each other. Understanding the numerical characteristics of these kinds of target function may require special measures. It can be difficult to find a sampling method that may resolve all difficulties in sampling for various types of complex distributions. In this thesis, I seek to develop a methodological framework with some degree of general applicability that can be useful for a range of sampling problems.

1.2 Overview

In this thesis, I propose two methodological innovations in computational inference for complex stochastic models. Relevant background information for each development is provided below, as well as summaries of my contributions.

1.2.1 Inference algorithms for coupled Markov processes with partial observations

Some probability models have certain structure that can be exploited to substantially enhance the efficiency in numerical computations. One instance is where the probability distribution has certain conditional independence structure. Conditional independence can be represented by graphical models. A frequently arising conditional independence structure observed in real world examples is the Markov property in temporal contexts. In a Markov process, the past and the future are independent given the present state. Due to this conditional independence structure that is linear in graphical representation, it is often a good strategy to numerically represent the distribution in a sequential manner.

In statistical data analysis framework, Markov process models are often used as a basis on which data are obtained as incomplete or partial observations. The data are modeled to be draws from measurement processes, conditional on the state of the underlying Markov process. The Markov process model and the measurement process model are jointly referred to as a partially observed Markov process (POMP) model. For POMP models, inference often requires understanding the posterior distribution of the Markov process given the data. Unless the model has an analytically tractable form on which the inference procedure can be based, the distribution of the latent Markov process is numerically represented using an ensemble of random draws. The numerical representation of the latent state can be sequentially carried out using the Markov structure. This approach can reduce the dimension of the space to be computationally represented, because it allows us to deal with the state of the process at a single time point rather than the sequence of states over all time points. Thus, the gain in computational efficiency can be huge compared to approaches that do not take into account the temporal structure.

Sequentially updating the representation of the latent distribution of the Markov process given the observations up to a certain time point is often referred to as a *filtering* procedure. A filtering procedure can be implemented as sequential importance sampling. At each importance sampling step, additional piece of information provided by the newest observation is incorporated into the computational representation.

Challenges arise when the space is high dimensional, since the number of samples needed to represent a distribution on the space can increase steeply. High dimensional observations can also lead to difficulties, because high dimensional measurement densities may designate a small volume in the high dimensional space as the only viable candidates for the hidden state. These difficulties may be manifested by unbalanced weights in importance sampling. Consequently, the numerical representation may become highly variable and lacking in internal diversity.

One approach that aims to solve this issue implements a sequence of bridging distributions. If the proposal distribution in importance sampling is very different from the target distribution in the sense that the ratio between the corresponding densities has high variance, multiple intermediate importance sampling steps that sequentially target the bridging distributions can reduce the gap between the proposal and the target distribution. Numerically efficient choices for bridging distributions can be obtained with relative ease if both the proposal and the target distribution have analytically tractable densities.

Contributions I propose an inference algorithm for coupled stochastic dynamic systems with partial observations. I focus on situations where the target distribution is high dimensional and does not have analytically tractable density. As discussed in the previous section, analytically intractable distributions can arise when a mechanistic model is defined using a simulation algorithm. I show that even for high dimensional analytically intractable distributions, stable importance sampling algorithms can be developed that enable inference on coupled Markov processes with partial observations. I demonstrate empirically and theoretically that the proposed algorithm scales more favorably than other methods proposed for high dimensional stochastic process models.

1.2.2 Flexible, numerically efficient sampling from complex distribution using Markov chain Monte Carlo

The task of sampling from a target distribution whose density can be evaluated up to a multiplicative constant arises frequently in Bayesian statistics when the normalizing constant of the posterior distribution is not computable. Markov chain Monte Carlo (MCMC) sampling is a very widely used class of methods for this task. MCMC constructs a Markov chain whose ergodic limit equals the target distribution. There are numerous MCMC methods, and different algorithms exhibit strengths in different circumstances. For example, there are methods known to scale better with increasing dimensions than other methods.

The wide use of MCMC methods is partly due to the fact that there exists a simple methodological strategy that allows for the construction of a Markov kernel that has the target distribution as its stationary distribution. For example, the Metropolis-

Hastings algorithm, which either accepts or rejects a proposal drawn from a kernel with certain probability, constructs a reversible Markov chain with respect to the target distribution.

Contributions I propose a generalization of the Metropolis-Hastings strategy that is conceptually simple and algorithmically easy to implement. This generalization allows for multiple proposals to be made for Metropolis-Hastings type acceptance. The multiple proposal framework can be applied not only to algorithms that use stochastic proposal kernels, but also to algorithms that employ deterministic kernels. The new framework increases flexibility in the implementation of various MCMC algorithms. I show that the enhanced flexibility can lead to increased computational efficiency, especially in tasks of sampling from complex distributions.

1.3 Organization of the thesis

In Chapter 2, I propose a new computational inference algorithm for coupled dynamic processes, which I call a guided intermediate resampling filter (GIRF). I describe the algorithm, provide theoretical results showing that the algorithm scales substantially better than standard methods, and explain how the algorithm can be implemented in practice. I illustrate the favorable scaling to high dimension with numerical results on a toy model. I also explain how parameter estimation can be carried out using this algorithm.

In Chapter 3, I apply the GIRF algorithm for a real scientific inference problem. The spatiotemporal transmission dynamics of measles at linked geographic locations in England and Wales in the twentieth century is analyzed using weekly case reports data. The strength of the spatial coupling of transmission dynamics in various locations is inferred using a fully likelihood based method. This result marks an advance in inference methodology, because inference on joint properties such as coupling using a fully likelihood based method has been considered difficult and avoided in practice.

In Chapter 4, I propose a generalization of the Metropolis-Hastings acceptance or rejection strategy in Markov chain Monte Carlo sampling. I introduce the new framework in a general setting, and explain how this framework can be combined with various MCMC algorithms that are frequently used in practice. Theoretical results showing the validity of the new method are provided, and its relationship with other approaches in the literature is explained. Discussions on how this novel framework can be practically used to improve computational efficiency, including the ways of flexibly tuning parameters in Hamiltonian Monte Carlo algorithms, follow. Numerical results show computational gains of using this framework when sampling from complex distributions.

CHAPTER II

A guided intermediate resampling particle filter for inference on high dimensional systems

Sequential Monte Carlo (SMC) methods, also known as particle filter methods, are a basic tool for inference on nonlinear partially observed Markov process (POMP) models. However, the performance of standard SMC algorithms quickly deteriorates as the model dimension increases. We present a novel particle filter which we call a guided intermediate resampling filter (GIRF). The GIRF is readily applicable to a broad range of models thanks to its plug-and-play property of requiring only a simulator of the process but not an evaluator of the transition density for inference. Theoretical and experimental results indicate that the GIRF scales much better than the standard particle filter, suggesting that the GIRF opens new possibilities for inference on highly nonlinear, non-Gaussian dynamic systems of moderate dimension.

2.1 Introduction

Partially observed Markov process (POMP) models offer a framework for likelihood based analysis of dynamic systems. A POMP model, otherwise known as the state space model, consists of a Markov state process representing the time evolution of the system and a measurement process that provides partial or noisy information about the states. Sequential Monte Carlo (SMC) methods are recursive algorithms that enable estimation of the likelihood and the posterior state distributions given data from a POMP model [Doucet et al., 2001, Cappé et al., 2007, Doucet and Johansen, 2011]. These approaches, also known as particle filter methods, approximate state distributions with a collection of simulated random variables, which are called particles.

Inference on some dynamic systems require fitting models with high dimensional state space to high dimensional data. Dynamic processes involving many spatial locations appear in the study of ecological, epidemiological and geophysical systems, for example. For these spatiotemporal models, both the state and measurement dimension tend to scale linearly with the number of spatial locations. Ensemble Kalman filter methods have been used to predict atmospheric dynamics for weather forecasts due to their good scalability to high dimensions [Houtekamer and Mitchell, 2001]. However, these methods can be ineffective for highly nonlinear and non-Gaussian systems, because they rely on locally linear and Gaussian approximations [Ades and Van Leeuwen, 2015, Lei et al., 2010, Miller et al., 1999]. In systems biology, models for networks of reactions often build upon deterministic differential equations or stochastic simulation [Kitano, 2002]. The model dimension typically increases with the number of system components, but even the state-of-the-art inference methods are not suitable for application beyond small systems [Owen et al., 2015].

Particle filter methods suffer from rapid deterioration in performance as the model dimension increases. This phenomenon occurs due to the weight degeneracy among particles. When highly unbalanced weights are given to the particles, resampling results in loss of particle diversity and poor approximation to the state distribution. Theoretical results demonstrating this phenomenon were established by Bengtsson et al. [2008] and Snyder et al. [2008]. These authors found out that the number of particles required for filtering increases exponentially in the variance of the log density of the observation given the state, which is closely tied to the space dimension. Heuristically, these results indicate that the curse of dimensionality (COD) is related to high dimensional measurement density, implying that particle depletion happens because each observation carries too much information. In this sense, the COD in particle filtering may be understood as a curse of too much information.

The view that too much information in the observations leads to particle depletion suggests that the difficulty in filtering might be combatted by controlling the rate at which the filtering algorithm introduces new information. We propose such an algorithm, which is shown both in theory and practice to perform well in moderately high dimensions. A high level summary of the algorithm, which we refer to as a guided intermediate resampling filter (GIRF), is as follows.

1. Divide each time interval between observations into sub-intervals, whose number is chosen in accordance with the space dimension of the POMP model.
2. For each sub-interval thus obtained,
    (a) Evolve the particles according to the transition kernel of the original state process.
    (b) Assess the fitness of each particle to future observations.
    (c) Resample the particles with weights reflecting the changes from the previous assessment.

A schematic diagram of this algorithm is provided in Figure 2.1. The assessments at step 2(b) can be made based on the approximations to the predictive likelihoods of a certain number of future observations. This way, the particles with low predictive likelihoods are pruned away, while the particles with higher predictive likelihoods

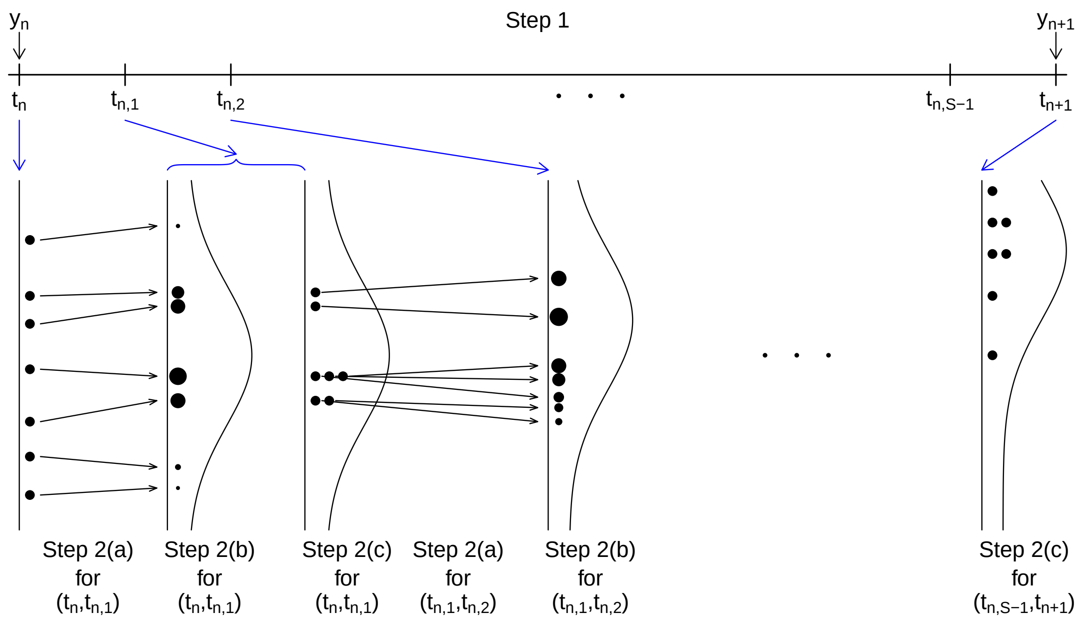
y_n
Step 1
y_{n+1}
t_n
$t_{n,1}$
$t_{n,2}$
. . .
$t_{n,S-1}$
t_{n+1}
. . .
Step 2(a) for $(t_n,t_{n,1})$
Step 2(b) for $(t_n,t_{n,1})$
Step 2(c) for $(t_n,t_{n,1})$
Step 2(a) for $(t_{n,1},t_{n,2})$
Step 2(b) for $(t_{n,1},t_{n,2})$
Step 2(c) for $(t_{n,S-1},t_{n+1})$

Figure 2.1: A schematic diagram of a GIRF algorithm

survive and propagate to the subsequent time point. The repeated assessment and resampling steps gradually guide the particles toward the correct posterior state distribution conditional on the data. We will usually set the number of sub-intervals between observations equal to the dimension of the state and measurement space for favorable scaling. This choice is justified in later sections.

In order to simulate the state process over shorter time intervals, we impose a constraint on the model that the transition distribution of the state process is infinitely divisible. Infinite divisibility of the transition distribution is a natural characteristic of all continuous time Markov processes, a widely used class of models across the physical and biological sciences.

We emphasize that a key difference that distinguishes our GIRF from other methods in the literature designed for high dimensional filtering is its practical utility. An inference method that can be implemented with only the simulator of the data generating process is said to have the *plug-and-play* property [Bretó et al., 2009, He et al., 2009]. In the context of SMC, the plug-and-play property means that only a simulator of the state process, but not an evaluator of its transition density, is required for inference. Our GIRF method, possessing the plug-and-play property, facilitates the use of a broad range of models, including mechanistic models defined by simulation algorithms or models defined by stochastic differential equations. Both kinds of models typically have analytically intractable transition densities, but their state processes can be simulated. The sample paths of diffusion processes can be approximately simulated with numerical methods such as the Euler-Maruyama method [Kloeden and Platen, 1999]. The plug-and-play property is essential for inference on POMP models whose state processes have intractable transition densities.

Our GIRF algorithm has connections to some well known methods in the particle filtering literature. First, it can be theoretically formulated either as a generalization or as a special case of the bootstrap filter by Gordon et al. [1993]. The latter interpretation places the algorithm within the general theory of SMC and provides immediate proofs for the unbiasedness of likelihood estimates and other results of convergence for the GIRF. Our method can also be seen as a generalization of the auxiliary particle filter (APF) proposed by Pitt and Shephard [1999]. The APF evolves and weights particles in a way that depends on the next observed data point. This approach often results in improved filtering, although it has been noted that this may not be always the case [Johansen and Doucet, 2008]. Our method is similar to the APF in that the particles are guided by oncoming observations. In the GIRF, adapted proposals for the next time step are obtained through a series of propagation and resampling steps at the intermediate time points.

The remainder of the chapter is organized as follows. Section 2.2 reviews several ideas in the literature that are related to high dimensional filtering. Section 2.3

introduces and explains our GIRF method. Section 2.4 reports some of its theoretical properties, including the main result (Theorem II.2) that establishes a finite sample error bound for the estimates obtained by the GIRF. This result, developed from first principles, offers a novel viewpoint on the filtering error and explains why our GIRF scales better to high dimensions than standard methods. Section 2.5 describes how one can estimate model parameters by combining the GIRF with the iterated filtering scheme of Ionides et al. [2015]. Implementation of our algorithm in Section 2.6 empirically show our algorithm's favorable scaling and its capability of facilitating spatiotemporal inference that has previously been considered inaccessible due to computational constraints. Section 2.7 concludes with a discussion.

2.2 Previous approaches to high dimensional filtering

Several theoretically motivated algorithms for high dimensional particle filtering have been proposed in the past few years. Rebeschini and Van Handel [2015] considered a filtering method that builds upon the assumption that the interaction between the spatial locations is local. The algorithm partitions the state variables into blocks and approximates the one step transitions of the state process as being independent between the blocks. A theoretical bound for the filtering error was derived, which only depends on the size of the largest block but not on the entire space dimension. Despite this very desirable scaling property, this approach has some practical limitations, because it is not applicable to highly interdependent spatial models and the filter estimates are not reliable near the boundaries of the blocks, which may constitute a substantial fraction of the total number of variables.

Beskos et al. [2014a,b] applied the annealed importance sampling proposed in Neal [2001] to high dimensional filtering and investigated its theoretical properties. The annealed importance sampling method introduces a series of bridging distributions between observations, whose densities are set proportional to a fractional power of the desired target density. Between two adjacent importance resampling, the particles are transformed according to a transition kernel whose stationary distribution equals the target bridging distribution. These transition kernels provide mixing that helps maintain the stability of the particle approximations. The authors gave stability results for the case where the original high dimensional state process is composed of many copies of independent and identically distributed (IID) one dimensional processes. In particular, Beskos et al. [2014a] showed that the importance weights are non-degenerate as the dimension goes to infinity even with fixed particle size. Beskos et al. [2014b] showed that both the $\mathcal{L}_2$ error of the filter estimates and the variance of the likelihood estimates are bounded uniformly in the space dimension. However, the main drawback of this approach, which reduces its practical value, is the absence of the plug-and-play property. Annealed importance sampling requires evaluable analytic expression of the density of the one-step transition in order to build artificial transition kernels between bridging distributions.

Beskos et al. [2017] studied the case where the spatial structure of the model can be hierarchically factorized and investigated the possibility of overcoming the COD. Specifically, they assumed that the one step transition density equals, or can be well approximated by, a product of terms which are functions of the state variables belonging to an increasing sequence of subsets of the dimensions of the space. The theoretical results they obtained by considering a few simple IID cases are promising, because they show that filtering can be stable when the number of particles increases linearly with the space dimension. These results provide useful insights into what might be achieved in more general cases.

Del Moral and Murray [2015] have proposed a particle filtering algorithm for highly informative observations that is almost identical to our method at its core, though our motivation and theoretical analysis differ. The authors were motivated by the study of perfectly observed diffusion processes, which share with high dimensional POMPs the difficulty that highly informative observations make computations challenging. In this thesis, I demonstrate the utility of a GIRF in high dimensions, both theoretically and empirically. We show that the GIRF may yield accurate estimates of the posterior state distributions given the data in high dimensions. In order to further avoid weight degeneracy, our method uses more than one future observations for particle assessment. This potential improvement was less relevant for the precisely measured processes considered by Del Moral and Murray [2015].

2.3 Method

2.3.1 A POMP model and Sequential Monte Carlo

We consider a Markov state process defined in continuous time and denoted by $\{X_t\,; t \geq 0\}$, with the random variable X_t taking values in a space $\mathbb{X}$. The measurement process yields observations $\{Y_n\,; n = 1, 2, \ldots, N\}$ that are incomplete, noisy measurements of X_t at discrete time points $t_n > 0$, $n = 1, \ldots, N$. The measurement Y_n is independent of other observations Y_m, $m \neq n$, and of the state process $\{X_t\}$, given the current state X_{t_n}. The observations $Y_n = y_n$ for $n = 1, \ldots, N$ are assumed to be fixed data. The state process evolves over time according to Markov transition kernels $K_{t,t'}$, where $0 \leq t \leq t'$. That is, the probability distribution of the random state $X_{t'}$ conditioned on $X_t = x_t$ is given by

$$X_{t'} \mid (X_t = x_t) \sim K_{t,t'}(dx\,; x_t).$$

We denote the initial state distribution at time $t_0 \geq 0$ by μ_{t_0}. We will occasionally express the distributions of random variables in terms of their densities. For example, the density of X_{t_n} given $X_{t_m} = x_{t_m}$ $(m < n)$ will be denoted by $p_{X_{t_n} \mid X_{t_m}}(x \mid x_{t_m})$ with respect to a reference measure on $\mathbb{X}$ written as dx. The measurement process for Y_n conditioned on $X_{t_n} = x_{t_n}$ is assumed to have density $g_n(\,\cdot \mid x_{t_n})$. We adopt the notation $n\!:\!m = \{n, n+1, \ldots, m\}$ for integers $n \leq m$. Some quantities of interest in an inference on a POMP model include the likelihood of data

$$\ell_{1:N}(y_{1:N}) = \mathbb{E}\left[\prod_{n=1}^{N} g_n(y_n \mid X_{t_n})\right],$$

where the expectation is taken with respect to the law of $\{X_t\,; t \geq 0\}$, and the filtering distribution of X_{t_n} conditioned on the observations $y_{1:n}$, whose density is denoted by $p_{X_{t_n} \mid Y_{1:n}}(x_{t_n} \mid y_{1:n})$.

Particle filter methods operate by recursively approximating the filtering distributions. The approximation at time t_n is realized by the sample draws $\{X_{t_n}^j\,; j \in 1\!:\!J\}$ and associated importance weights $\{\tilde{w}^j\,; j \in 1\!:\!J\}$. The weighted sum of point measures

$$\sum_{j=1}^{J} \tilde{w}^j \delta_{X_{t_n}^j}(dx) \tag{2.1}$$

is taken as an approximation to the filtering distribution. Heading to the next time point t_{n+1}, the particle filter first draws samples from the discrete weighted distribution (2.1). This step is called resampling. Next in the propagation step, the resampled particles are independently transformed according to some transition kernel. A set of importance weights are given to the transformed particles, such that the new weighted sum represents the filtering distribution of $X_{t_{n+1}}$ conditioned on $y_{1:n+1}$. The choice of the propagation kernel affects the performance of the particle filter as in the general case of importance sampling, where the proposal distribution determines the stability of the resulting estimates.

2.3.2 Guided intermediate resampling filter

In what follows, we assume that the transition kernel of the state process can be simulated but do not require its density to be evaluated. We also assume that the state transition kernels $K_{t,t'}$ for the state process $\{X_t\,; t \geq 0\}$ are infinitely divisible and can be expressed as

$$K_{t,t'} = K_{t,\tau_1} K_{\tau_1,\tau_2} \cdots K_{\tau_{n-1},\tau_n} K_{\tau_n,t'}$$

for any number of intermediate time points $t \leq \tau_1 \leq \cdots \leq \tau_n \leq t'$. For implementation of our GIRF algorithm, we pick $S{-}1$ intermediate time points $t_{n,s}$, $s \in 1\!:\!S{-}1$, within the observation time interval $[t_n, t_{n+1}]$ such that

$$t_{n,0} := t_n < t_{n,1} < \cdots < t_{n,S-1} < t_{n,S} := t_{n+1}$$

for $n \in 0\!:\!N{-}1$. As a rule of thumb, we will take $S = d$, the dimension of the measurement space.

The algorithm starts with an initial swarm of J particles $\{X^{F,j}_{t_0}\,; j \in 1\!:\!J\}$ of equal weights that represent the initial distribution of X_{t_0}. The superscript F stands for "filtered particles". The algorithm proceeds recursively. Suppose at some time $t_{n,s-1}$, we have a collection of particles, denoted by $\{X^{F,j}_{t_{n,s-1}}\,; j \in 1\!:\!J\}$. The particles are transformed according to the transition kernel $K_{t_{n,s-1},t_{n,s}}$ and called the proposed particles, denoted by $\{X^{P,j}_{t_{n,s}}\,; j \in 1\!:\!J\}$. These proposed particles at time $t_{n,s}$ are assessed based on how likely they are to generate the future observations $y_{n+1:n+B}$ for some $B \geq 1$. The assessments are made by what we call the guide function, $u_{t_{n,s}}\colon \mathbb{X} \to \mathbb{R}^+$. At the initial time point we require that $u_{t_0}(x) \equiv 1$ and at the last time point $u_{t_N}(x) = g_N\,(y_N \,|\, x)$ for all $x \in \mathbb{X}$. The weight for each particle at time

Algorithm 1: A guided intermediate resampling filter (GIRF)

```
Input  : Simulator for $\mu_{t_0}(dx)$
         Simulator for $K_{t_{n,s-1},t_{n,s}}(dx\,;x_{t_{n,s}})$ for $n \in 0\,{:}\,N{-}1$ and $s \in 1\,{:}\,S$
         Evaluator for $g_n(y_n \,|\, x_{t_n})$ for $n \in 1\,{:}\,N$
         Evaluator for $u_{t_{n,s}}(x_{t_{n,s}})$ for $n \in 0\,{:}\,N{-}1$ and $s \in 1\,{:}\,S$
         Data, $y_{1:N}$
         Number of particles, $J$
Output: Filtered particle swarm, $\left\{X^{F,j}_{t_N}\,;j \in 1\,{:}\,J\right\}$
         Likelihood estimate, $\hat{\ell}$
Initialize: $\hat{\ell} \leftarrow 1$, $X^{F,j}_{t_0} \sim \mu_{t_0}(dx)$ for $j \in 1\,{:}\,J$, and $u^j_{old} \leftarrow 1$ for $j \in 1\,{:}\,J$
for $n \leftarrow 0\,{:}\,N{-}1$ do
    If $n \geq 1$ then $u^j_{old} \leftarrow \dfrac{u^j_{old}}{g_n\left(y_n \,\middle|\, X^{F,j}_{t_n}\right)}$ for $j \in 1\,{:}\,J$
    for $s \leftarrow 1\,{:}\,S$ do
        $X^{P,j}_{t_{n,s}} \sim K_{t_{n,s-1},t_{n,s}}\left(dx\,;X^{F,j}_{t_{n,s-1}}\right)$ for $j \in 1\,{:}\,J$
        $u^j_{new} \leftarrow u_{t_{n,s}}\left(X^{P,j}_{t_{n,s}}\right)$ for $j \in 1\,{:}\,J$
        $w^j \leftarrow u^j_{new}/u^j_{old}$ for $j \in 1\,{:}\,J$
        $\hat{\ell} \leftarrow \hat{\ell} \times \left(\sum_{j=1}^J w^j\right)/J$
        Draw $a^j$ with $\mathbb{P}\left(a^j = i\right) = w^i/\sum_{i'=1}^J w^{i'}$ for $j \in 1\,{:}\,J$
        Set $X^{F,j}_{t_{n,s}} = X^{P,a^j}_{t_{n,s}}$ and $u^j_{old} = u^{a^j}_{new}$ for $j \in 1\,{:}\,J$
    end
    Set $X^{F,j}_{t_{n+1,0}} = X^{F,j}_{t_{n,S}}$ for $j \in 1\,{:}\,J$
end
```

$t_{n,s}$, $s \in 1 : S$, is determined by the ratio of the assessments at time $t_{n,s}$ and $t_{n,s-1}$. The algorithm sets the weight for the j-th particle to be

(2.2)

$$w^j = w_{t_{n,s}}\left(X^{P,j}_{t_{n,s}}, X^{F,j}_{t_{n,s-1}}\right) := \begin{cases} \dfrac{u_{t_{n,s}}\left(X^{P,j}_{t_{n,s}}\right)}{u_{t_{n,s-1}}\left(X^{F,j}_{t_{n,s-1}}\right)} & \text{if } t_{n,s-1} \notin t_{1:N} \\ \dfrac{u_{t_{n,s}}\left(X^{P,j}_{t_{n,s}}\right)}{u_{t_{n,s-1}}\left(X^{F,j}_{t_{n,s-1}}\right) \Big/ g_n\left(y_n \,\middle|\, X^{F,j}_{t_{n,s-1}}\right)} & \text{if } t_{n,s-1} \in t_{1:N}. \end{cases}$$

If $t_{n,s-1} \in t_{1:N}$, the denominator is divided by $g_n\left(y_n \,\middle|\, x^F_{t_n}\right)$, because at time $t_{n,s} = t_{n,1} > t_n$, the past observation y_n should no longer be considered in assessing the fitness. The weights at observation times t_n are taken as $w_{t_{n-1,S}}(X^{P,j}_{t_{n-1,S}}, X^{F,j}_{t_{n-1,S-1}})$. The particles are then resampled with normalized weights $w^j/\sum_{i=1}^J w^i$ and renamed as $\{X^{F,j}_{t_{n,s}}\,; j\in 1\,{:}\,J\}$. The pseudocode for our method is shown in Algorithm 1. Our implementation of the GIRF is available at `https://github.com/joonhap/GIRF.git`.

The likelihood estimate $\hat{\ell}$ of $\ell_{1:N}(y_{1:N})$ is obtained from Algorithm 1. This quantity can be much more stable than the likelihood estimate obtained from the standard bootstrap particle filter in high dimensions. This claim is supported by Theorem II.2 and by the argument given in the appendix section 2.B. Algorithm 1 is equivalent to the bootstrap particle filter if we take $S = 1$ and $u_{t_n}(x_{t_n}) = g_n(y_n\,|\,x_{t_n})$. It is equivalent to the auxiliary particle filter if we take $S = 1$ and $u_{t_n}(x_{t_n}) = g_n(y_n\,|\,x_{t_n})\cdot g_{n+1}\left(y_{n+1}\,|\,\mu_{t_{n+1}}(x_{t_n})\right)$ where $\mu_{t_{n+1}}(x_{t_n})$ denotes a deterministic or stochastic prediction for the state at time t_{n+1} based on $X_{t_n} = x_{t_n}$.

The particle swarm $\{X^{F,j}_{t_{n,s}}\,; j\in 1\,{:}\,J\}$ at time $t_{n,s}$ targets a density proportional to

$$u_{t_{n,s}}\left(x_{t_{n,s}}\right)\cdot p_{X_{t_{n,s}}\,|\,Y_{1\,:\,n}}\left(x_{t_{n,s}}\,\middle|\,y_{1\,:\,n}\right). \tag{2.3}$$

When $u_{t_{n,s}}(x_{t_{n,s}})$ approximates $p_{Y_{n+1:n+B}\,|\,X_{t_{n,s}}}(y_{n+1:n+B}\,|\,x_{t_{n,s}})$, the above expression (2.3) approximates the conditional density $p_{X_{t_{n,s}}\,|\,Y_{1:n+B}}(x_{t_{n,s}}\,|\,y_{1:n+B})$.

The following simple argument shows that (2.2) makes Algorithm 1 a properly weighted filter [Liu, 2008, Definition 2.5.1]. For each particle $X^{F,j}_{t_{n,s}}$, we define a parent particle at time $t_{n,s-1}$ as follows: if $X^{P,a^j}_{t_{n,s}}$ for some $a^j\in 1\,{:}\,J$ was called $X^{F,j}_{t_{n,s}}$ after resampling, then $X^{F,a^j}_{t_{n,s-1}}$, which propagated to $X^{P,a^j}_{t_{n,s}}$, is the parent particle of $X^{F,j}_{t_{n,s}}$. By successively tracing the parent particles, one can construct the ancestral lineage of a particle. Take a particle $X^{F,j}_{t_N}$ at time t_N and call its ancestor at time $t_{n,s}$ as $X^{F,a^j_{t_{n,s}}}_{t_{n,s}}$, where we write $a^j_{t_N} = j$. It turns out that the product of all importance weights throughout the resampling stages for this lineage gives the measurement density of $y_{1:N}$ given the states $X_{t_n}^{F,a_{t_n}^j}$:

(2.4)

$$\prod_{n=0}^{N-1}\prod_{s=1}^{S} w_{t_{n,s}}\left(X_{t_{n,s}}^{P,a_{t_{n,s-1}}^j}, X_{t_{n,s-1}}^{F,a_{t_{n,s-1}}^j}\right) = \left[\prod_{n=1}^{N-1} g_n\left(y_n \,\middle|\, X_{t_n}^{F,a_{t_n}^j}\right)\right]\cdot\prod_{n=0}^{N-1}\prod_{s=1}^{S}\frac{u_{t_{n,s}}\left(X_{t_{n,s}}^{F,a_{t_{n,s}}^j}\right)}{u_{t_{n,s-1}}\left(X_{t_{n,s-1}}^{F,a_{t_{n,s-1}}^j}\right)}$$

$$= \left[\prod_{n=1}^{N-1} g_n\left(y_n \,\middle|\, X_{t_n}^{F,a_{t_n}^j}\right)\right]\cdot\frac{u_{t_N}\left(X_{t_N}^{F,a_{t_N}^j}\right)}{u_{t_0}\left(X_{t_0}^{F,a_{t_0}^j}\right)} = \prod_{n=1}^{N} g_n\left(y_n \,\middle|\, X_{t_n}^{F,a_{t_n}^j}\right).$$

The computational cost of Algorithm 1 typically scales as $O(JSd)$. If we take $S = d$ and use a fixed number of particles, it scales as $O(d^2)$. However, the number of particles will generally need to increase with d in order to keep the errors at a constant order of magnitude. In Section 2.4, we show a novel theoretical result on the filter accuracy, namely that for any f with $\|f\|_\infty \leq 1$,

$$\left|\frac{1}{J}\sum_{j=1}^{J} f\left(X_{t_N}^{F,j}\right) - \mathbb{E}\big[f(X_{t_N})\,|\,Y_{1:N} = y_{1:N}\big]\right| \leq v(S)$$

with high probability, where the bound $v(S)$ depends on the number of sub-intervals per observation S, the space dimension d, the number of particles J, the choice of the guide functions $u_{t_{n,s}}$, and other attributes of the POMP model. The rate at which the number of particles is required to increase with d can be deduced from the bound $v(S)$.

2.3.3 Choice of the guide functions

Although Algorithm 1 is a properly weighted filter for any guide function $u_{t_{n,s}} : \mathbb{X} \to \mathbb{R}^+$, its numerical efficiency depends on the choice of the guide function. We take $u_{t_{n,s}}(x)$ to be an approximation to the predictive likelihood of $y_{n+1:n+B}$ given $X_{t_{n,s}} = x$,

$$u_{t_{n,s}}(x) \approx p_{Y_{n+1:n+B}\,|\,X_{t_{n,s}}}\left(y_{n+1:n+B}\,|\,x\right). \tag{2.5}$$

When $n+B > N$, we take $u_{t_{n,s}}(x) \approx p_{Y_{n+1:N} \,|\, X_{t_{n,s}}}(y_{n+1:N} \,|\, x)$ instead. At observation times t_n, $n \in 1\!:\!N$, the guide function u_{t_n} is defined as $u_{t_{n-1,S}}$. We illustrate how the algorithm works with this guide function using a simple example consisting of a twenty dimensional Brownian motion and a measurement process independent in each dimension. The POMP model is defined on the interval $[t_0, t_1]$ as

(2.6)
$$X_{t_0} \sim N(0, I), \qquad X_{t'} \,\big|\, X_t \sim N\left(X_t, \frac{t'-t}{t_1-t_0} I\right) \quad \text{for } t \le t', \qquad Y_{t_1} \,\big|\, X_{t_1} \sim N(X_{t_1}, I).$$

Here, I denotes the twenty dimensional identity matrix. The guide function was set to be the exact predictive likelihood with $B=1$, namely $u_{t_{0,s}}(x) := p_{Y_1 \,|\, X_{t_{0,s}}}(y_1 \,|\, x)$. The time interval $[t_0, t_1]$ was divided into $S = 20$ sub-intervals of equal length. Figure 2.2 shows the first two coordinates of the filtered particles $X^{F,j}_{t_{0,s}}$ at three intermediate time points. The mean of the initial distribution is marked by a green 'O', and the observation y_1 by a purple triangle. At time $t_{0,s}$, the conditional distribution given $Y_1 = y_1$ equals $X_{t_{0,s}} \,\big|\, (Y_1 = y_1) \sim N\big[\frac{1}{3}\left(1+\frac{s}{S}\right) y_1,\, \frac{1}{3}\left(1+\frac{s}{S}\right)\left(2-\frac{s}{S}\right) I\big]$. The mean of this conditional distribution for each s is marked by a red 'X', and the 95% coverage region by a blue dashed circle. As time progresses, the red 'X' shifts from the origin toward y_1, and the coverage region changes in size. The filtered particles almost exactly follow the conditional distributions at the intermediate steps, showing that they are gradually guided toward $p_{X_{t_1} \,|\, Y_1}$ as s increases from zero to twenty.

If the guide function is taken as in (2.5) and S is close to the space dimension d, the GIRF may be rescued from the weight degeneracy. We now give a heuristic argument for this claim. Theorem II.2 in Section 2.4 will provide a rigorous argument. First, we consider the resampling weights for $s \ge 2$. Suppose the ancestors of a particle $X^{F,j}_{t_{n+1}}$ are denoted by $\big\{X^{F,a^j_{t_{n,s}}}_{t_{n,s}}\,; s \in 1\!:\!S\big\}$, where $a^j_{t_{n,S}} = j$. The product of resampling

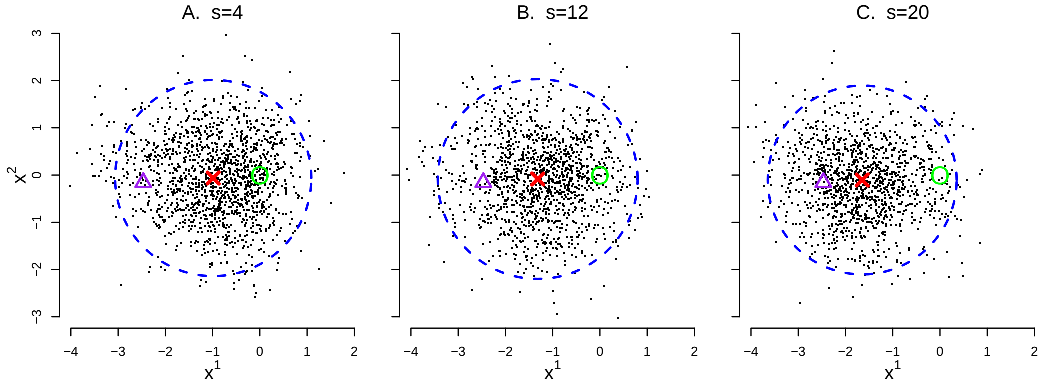

Figure 2.2: The first two coordinates of the filtered particles $X_{t_{0,s}}^{F,j}$ from a GIRF run at three intermediate time steps (A, s=4; B, s=12; C, s=20) for twenty dimensional linear Gaussian model given by (2.6).

weights $w_{t_{n,s}}\big(X_{t_{n,s}}^{P,a_{t_{n,s-1}}^j}, X_{t_{n,s-1}}^{F,a_{t_{n,s-1}}^j}\big)$ for $s \in 2\,{:}\,S$ approximates

$$\prod_{s=2}^{S} w_{t_{n,s}}\left(X_{t_{n,s}}^{P,a_{t_{n,s-1}}^j}, X_{t_{n,s-1}}^{F,a_{t_{n,s-1}}^j}\right) = \frac{u_{t_{n+1}}\left(X_{t_{n+1}}^{F,j}\right)}{u_{t_{n,1}}\left(X_{t_{n,1}}^{F,a_{t_{n,1}}^j}\right)} \approx \frac{p_{Y_{n+1:n+B}\,|\,X_{t_{n+1}}}\left(y_{n+1:n+B}\,\middle|\,X_{t_{n+1}}^{F,j}\right)}{p_{Y_{n+1:n+B}\,|\,X_{t_{n,1}}}\left(y_{n+1:n+B}\,\middle|\,X_{t_{n,1}}^{F,a_{t_{n,1}}^j}\right)}.$$

The logarithm of the right hand side of the above expression can be expected to be of order $O_p(Bd)$ if the POMP model consists of d weakly coupled processes, for which behavior should be similar to the IID case. Thus, if the terms $w_{t_{n,s}}\big(X_{t_{n,s}}^{P,a_{t_{n,s-1}}^j}, X_{t_{n,s-1}}^{F,a_{t_{n,s-1}}^j}\big)$ for $s \in 2\,{:}\,S$ are of roughly the same magnitude, the logarithm of each resampling weight may be on the order of $\frac{1}{S-1}O_p(Bd)$. If we take $S = d$ and B is not too large, the resampling weights may be $O_p(1)$ in d. This reasoning is closely related to Assumption 2 in Section 2.4.

The resampling weights at s=1 require additional consideration. The resampling weight for $X_{t_{n,1}}^{P,j}$ is given by

(2.7)
$$\frac{u_{t_{n,1}}\left(X_{t_{n,1}}^{P,j}\right)}{u_{t_n}\left(X_{t_n}^{F,j}\right)\Big/g_n\left(y_n\,\middle|\,X_{t_n}^{F,j}\right)} \approx \frac{p_{Y_{n+1:n+B}\,|\,X_{t_{n,1}}}\left(y_{n+1:n+B}\,\middle|\,X_{t_{n,1}}^{P,j}\right)}{p_{Y_{n:n+B-1}\,|\,X_{t_n}}\left(y_{n:n+B-1}\,\middle|\,X_{t_n}^{F,j}\right)\Big/g_n\left(y_n\,\middle|\,X_{t_n}^{F,j}\right)}$$
$$= \frac{p_{Y_{n+1:n+B-1}\,|\,X_{t_{n,1}}}\left(y_{n+1:n+B-1}\,\middle|\,X_{t_{n,1}}^{P,j}\right)}{p_{Y_{n+1:n+B-1}\,|\,X_{t_n}}\left(y_{n+1:n+B-1}\,\middle|\,X_{t_n}^{F,j}\right)} \cdot p_{Y_{n+B}\,|\,X_{t_{n,1}},Y_{n+1:n+B-1}}\left(y_{n+B}\,\middle|\,X_{t_{n,1}}^{P,j},\,y_{n+1:n+B-1}\right).$$

The term $\frac{p_{Y_{n+1:n+B-1}\,|\,X_{t_{n,1}}}\left(y_{n+1:n+B-1}\,|\,X^{P,j}_{t_{n,1}}\right)}{p_{Y_{n+1:n+B-1}\,|\,X_{t_n}}\left(y_{n+1:n+B-1}\,|\,X^{F,j}_{t_n}\right)}$ may be of $O_p(1)$ by the same reasoning as above. The term $p_{Y_{n+B}\,|\,X_{t_{n,1}},Y_{n+1:n+B-1}}(y_{n+B}\,|\,X^{P,j}_{t_{n,1}},\, y_{n+1:n+B-1})$ is related to the mixing of the POMP conditional on data. Conditional mixing of a POMP means that a distant future observation provides substantially less information about the current state than the nearest future observation does, provided that all the observations until that distant time point are already known. The additional information about $X_{t_{n,s}}$ provided by y_{n+B} when $y_{n+1:n+B-1}$ are known is represented by the likelihood $p_{Y_{n+B}\,|\,X_{t_{n,s}},Y_{n+1:n+B-1}}\left(y_{n+B}\,\middle|\,x_{t_{n,s}}, y_{n+1:n+B-1}\right)$. Under conditional mixing, this likelihood yields balanced values when evaluated at the support of the distribution $p_{X_{t_{n,s}}\,|\,Y_{1:n+B-1}}(x_{t_{n,s}}\,|\,y_{1:n+B-1})$. Since an approximation to this distribution is targeted by the particles $\{X^{P,j}_{t_{n,1}}\,; j \in 1\,{:}\,J\}$, the values of $p_{Y_{n+B}\,|\,X_{t_{n,1}},Y_{n+1:n+B-1}}(y_{n+B}\,|\,X^{P,j}_{t_{n,1}},\, y_{n+1:n+B-1})$ for $j \in 1\,{:}\,J$ may be balanced. Thus, the resampling weight at $s{=}1$ shown in (2.7) may not suffer from weight degeneracy, if conditional mixing is obtained for B not too large. Assumption 3 in Section 2.4 formalizes the conditional mixing argument in a manner that is relevant to our theoretical investigation.

The state distribution conditioned on several future observations is called the fixed lag smoothing distribution. Its use for stable filtering has been studied in the literature, for example, in Clapp and Godsill [1999], Chen et al. [2000], and Doucet et al. [2006]. Our contribution is to connect this approach with intermediate resampling algorithms and the COD. Fixed lag smoothing distributions tend to be less affected by outliers in the observed data than filtering distributions [Lin et al., 2013]. Intuitively, looking ahead to B observations in the future for particle assessment allows the information provided by the observation y_{n+B} to be processed over a longer time interval, $[t_n, t_{n+B}]$.

Practical design of the guide functions

In practical situations, finding a good approximation to the predictive likelihood of future observations can be a difficult task. It may be particularly demanding when the transition density of the state process is intractable, which is the situation when plug-and-play methods are particularly desired. Therefore, designing practically accessible guide functions is critical in the application of the GIRF. Here we suggest several ways of making such designs.

1. Semi-analytical approach with moment matching The predictive likelihood of multiple future observations is typically more difficult to estimate than the predictive likelihood of a single observation. Thus after approximating $p_{Y_{n+b}\,|\,X_{t_{n,s}}}\left(y_{n+b}\,\middle|\,x_{t_{n,s}}\right)$ and calling the approximation $u_{t_{n,s}\nearrow t_{n+b}}$ for $b \in 1\!:\!B$, we may set

$$u_{t_{n,s}}\left(x_{t_{n,s}}\right) = \prod_{b=1}^{B} u_{t_{n,s}\nearrow t_{n+b}}\left(x_{t_{n,s}}\right). \tag{2.8}$$

For $s = S$ and $b = 1$, we can evaluate the measurement density at $t_{n,S} = t_{n+1}$, so we set

$$u_{t_{n,S}\nearrow t_{n+1}}\left(x_{t_{n,S}}\right) = g_{n+1}\left(y_{n+1}\,\middle|\,x_{t_{n,S}}\right).$$

The approximate predictive likelihood $u_{t_{n,s}\nearrow t_{n+b}}$ may be taken sensibly depending on the model. We suggest one way as follows. First, we make a projection from the current state $X_{t_{n,s}} = x_{t_{n,s}}$ to time t_{n+b} with a deterministic process that approximates the conditional mean of the state process $\{X_t\,;t \geq t_{n,s}\}$ given $x_{t_{n,s}}$. The projected state will be denoted by $\tilde{x}_{t_{n+b}}$. We also assume that the variability of $X_{t_{n+b}}$ given $X_{t_{n,s}} = x_{t_{n,s}}$ according to the law of the state process can be approximately characterized by $\Sigma_1(x_{t_{n,s}})$. We assume that the measurement density of Y_{n+b} given $X_{t_{n+b}} = \tilde{x}_{t_{n+b}}$, denoted by $g_{n+b}(\,\cdot\,|\,\tilde{x}_{t_{n+b}})$, is characterized by a scale parameter

$\Sigma_2(\tilde{x}_{t_{n+b}})$. We make the dependence explicit by writing $g_{n+b}\big[\,\cdot\,|\,\tilde{x}_{t_{n+b}}, \Sigma_2(\tilde{x}_{t_{n+b}})\big]$. The combined variability, denoted by $\Sigma_1\left(x_{t_{n,s}}\right) + \Sigma_2\left(\tilde{x}_{t_{n+b}}\right)$, is then taken as the scale parameter for the approximate predictive likelihood of Y_{n+b} given the current state $X_{t_{n,s}} = x_{t_{n,s}}$. In other words, we define

$$\text{(2.9)} \qquad u_{t_{n,s}\nearrow t_{n+b}}\left(x_{t_{n,s}}\right) := g_{n+b}\left[y_{n+b}\,\middle|\,\tilde{x}_{t_{n+b}}, \Sigma_1\left(x_{t_{n,s}}\right) + \Sigma_2\left(\tilde{x}_{t_{n+b}}\right)\right].$$

If the state process distribution and the measurement distribution belong to different scale families, $u_{t_{n,s}\nearrow t_{n+b}}$ may be obtained by approximating an unnormalized convolution density (see Section 3.2 for an example).

2. SMC-type likelihood estimation for models with weakly interacting state variables

For certain POMP models, the correlation between the components of the state process $\{X_t\}$ may be weak. For example, the state process may be a collection of coupled dynamic processes corresponding to different geographic locations, where the dynamics at one location is affected by the dynamics at other locations only by a small degree. When the correlations between the components of $\{X_t\}$ are weak, we may use the following approximation:

$$\text{(2.10)} \qquad p_{Y_{n+1:n+B}|X^j_{t_{n,s}}}(y^{[1:d]}_{n+1:n+B}\,|\,X^j_{t_{n,s}}) \approx \prod_{i=1}^{d} p_{Y^{[i]}_{n+1:n+B}|X_{t_{n,s}}}(y^{[i]}_{n+1:n+B}\,|\,X^j_{t_{n,s}}).$$

Here, the superscripts between brackets indicate the component of the observation variable: $y^{[1:d]}_{n+1:n+B}$ denotes the original d-dimensional observation vectors, and $y^{[i]}_{n+1:n+B}$ denotes the i-th components of the observation vectors. The approximation (2.10) can be particularly valid and useful in the cases where each observation $y^{[i]}_n$ depends only on $X^{[i]}_{t_n}$, which is not uncommon, and where the measurement distribution has large variance relative to the variability of the state process. This approximation may be understood in connection with variational inference.

Each term in the right hand side of (2.10) can be estimated using a standard SMC algorithm, where only the observations $y^{[i]}_{n+1:n+B}$ are used in filtering. The procedure is described as follows. For each particle $X^j_{t_{n,s}}$ at time t in the GIRF algorithm, we use J' number of particles, all of which is initialized at $X^j_{t_{n,s}}$. The joint state process $\{X_t\}$ is used to simulate the particle forward in time. The standard bootstrap SMC algorithm is run for the time period from time t to t_{n+B}. The J' particles in each time step (say t_{n+b}) are weighted according to the measurement density of only $y^{[i]}_{n+b}$. Since the measurement density is one dimensional, the weights will be as balanced as in one dimensional filtering problems. Thus $p_{Y^{[i]}_{n+1:n+B}|X_{t_{n,s}}}(y^{[i]}_{n+1:n+B} \,|\, X^j_{t_{n,s}})$ may be precisely estimated using only a moderate number of particles J', and there will be no need to go through intermediate time steps as in the GIRF algorithm. Note that this procedure still takes into account the coupling between the components of $\{X_t\}$, because we use the joint state process $\{X_t\}$ for particle propagation; only the observations for other components are ignored for filtering.

The computation of the likelihood estimates $\hat{p}(y^{[i]}_{n+1:n+B} \,|\, X^j_{t_{n,s}})$ may seem costly. The computation for each particle $X^j_{t_{n,s}}$ scales as $O(J'd^2)$. The total computational cost of running the GIRF thus scales as $O(JNSJ'd^2)$. When we take $S = d$, the cost scales as the cube of the space dimension. However, this seemingly steep cost is much more favorable than the typical exponential increase for standard SMC methods. For weakly coupled POMP models, this method may enable analyses that are otherwise infeasible.

A potential issue in the approximation (2.10) is that the likelihood estimate $\hat{p}(y^{[i]}_{n+1:n+B} \,|\, X^j_{t_{n,s}})$ has Monte Carlo variability. The resampling weights in the main GIRF are given by the ratio of these likelihood estimates between two consecutive intermediate time points. The ratio can be unstable if the variability in

$\hat{p}(y^{[i]}_{n+1:n+B} \,|\, X^j_{t_{n,s}})$ is high. A moderately large number of J' will often be able to make the variance in the likelihood estimates sufficiently small. However, a seed fixing strategy may additionally help in stabilizing the ratio between the likelihood estimates. For the estimation of $\hat{p}(y^{[i]}_{n+1:n+B} \,|\, X^j_{t_{n,s}})$, the same seed may be used for every $j \in 1\!:\!J$ at every intermediate time point $t_{n,s}$. Then, since the computation of $\hat{p}(y^{[i]}_{n+1:n+B} \,|\, X^j_{t_{n,s}})$ and $\hat{p}(y^{[i]}_{n+1:n+B} \,|\, X^j_{t_{n,s+1}})$ use the same sequence of random numbers, the Monte Carlo variability in both computations will likely cancel each other, making the ratio less variable. Of course, the fixed random seed should be used only for the likelihood estimation, and the random numbers used for all other operations (i.e., those used in the main GIRF algorithm) should not be affected.

3. Artificially increased variances of measurement densities In estimating $p_{Y_{n+1:n+B}|X_{t_{n,s}}}$, artificially increasing the variances of the measurement densities can help make the likelihood estimates more stable. When $p_{Y_{n+1:n+B}|X_{t_{n,s}}}$ is estimated using the SMC type approach described above, artificially increased measurement densities can make the estimates of $p_{Y_{n+1:n+B}|X_{t_{n,s}}}$ have less Monte Carlo variability. Svensson et al. [2018] have recently shown empirical results illustrating the advantages of using artificially increased variance of measurement densities. When using analytical methods for approximating the likelihood, larger measurement variances can yield more balanced values of approximated likelihood estimates among particles. Even when the approximation of the likelihood is inaccurate, balanced estimates can reduce numerical problems in resampling of particles in the GIRF algorithm.

2.4 Theoretical results

We first introduce some notation. For a bounded measurable function $f \in \mathcal{B}_b(\mathbb{X})$, we denote its integral with respect to a measure μ by μf, and the integral with respect to a Markov kernel K conditional on the starting state x by $Kf(x)$. The propagation of measure μ by a kernel K is defined as $(\mu K)f := \mu(Kf)$. At the time step $t_{n,s}$ in Algorithm 1, we denote the empirical distributions corresponding to the proposed particles and the filtered particles by $F^P_{t_{n,s},J} = \frac{1}{J}\sum_{j=1}^{J} \delta_{X^{P,j}_{t_{n,s}}}$ and $F^F_{t_{n,s},J} = \frac{1}{J}\sum_{j=1}^{J} \delta_{X^{F,j}_{t_{n,s}}}$ respectively. The empirical distribution of the J matching pairs $\big(X^{P,j}_{t_{n,s}}, X^{F,j}_{t_{n,s-1}}\big)$ on the product space $\mathbb{X}^2$ will be denoted by $H_{t_{n,s},J}$. The σ-algebra generated by the set of random draws $D^P_{t_{n,s}} := \{X^{F,j}_{t_{n',s'}}\,; t_{n',s'} \leq t_{n,s-1}, j \in 1\!:\!J\} \cup \{X^{P,j}_{t_{n',s'}}\,; t_{n',s'} \leq t_{n,s}, j \in 1\!:\!J\}$ is denoted by $\mathcal{B}^P_{t_{n,s},J}$, and the σ-algebra generated by $D^P_{t_{n,s}} \cup \{X^{F,j}_{t_{n,s}}\,; j \in 1\!:\!J\}$ is denoted by $\mathcal{B}^F_{t_{n,s},J}$.

Our GIRF can be cast into the framework of the standard particle filters by extending the state space to $\mathbb{X}^2$ where the new state variable is the pair $(X_{t_{n,s-1}}, X_{t_{n,s}})$. This extension is necessary because the resampling weights (2.2) depends on both $X^{P,j}_{t_{n,s}}$ and $X^{F,j}_{t_{n,s-1}}$. The likelihood estimates obtained from the standard particle filters are unbiased [Del Moral and Jacod, 2001]. It follows that the likelihood estimates from the GIRF are also unbiased. The consistency and the asymptotic normality of the filter estimates from the GIRF also follow naturally from the standard particle filter theory [Chopin, 2004, Del Moral, 2004].

Theorem II.1. *The likelihood estimate $\hat{\ell}$ of Algorithm 1 is unbiased for $\ell_{1:N}(y_{1:N})$.*

Proof. See Section 2.A. □

Next we show that the particle approximation to the filtering distribution by Algorithm 1 can have significantly smaller error than the standard filters in high dimensions. The GIRF converts a filtering problem with highly informative observations into one that deals with a slower rate of incoming information at the expense of operating on a refined time scale. Thus mixing of processes happens over greater number of time steps in this stretched time scale. For this reason, results in the literature which imply that the number of particles needs to increase exponentially in the number of time steps needed for conditional mixing, such as Theorem 3.1 of Del Moral and Guionnet [2001], is not very useful in this case. When we take $S = d$, a bound increasing exponentially in S is no better than a bound increasing exponentially in d. A new type of error bound will be given below that increases linearly in the number of time steps.

We introduce some more notation. For any t, t' such that $t_0 \leq t \leq t' \leq t_N$, we define

$$Q_{t,t'}(f)(X_t) := \mathbb{E}\left[f(X_{t'}) \prod_{t \leq t_n < t'} g_n(y_n \,|\, X_{t_n}) \middle| X_t \right], \tag{2.11}$$

for any bounded measurable function f. Note that, if no observation was made in $[t, t')$, we have

$$Q_{t,t'}(f) = K_{t,t'} f, \tag{2.12}$$

and if a single observation t_n was made in this interval,

$$Q_{t,t'}(f) = K_{t,t_n} \left\{ (K_{t_n,t'} f) \cdot g_n(y_n \,|\, \cdot) \right\}. \tag{2.13}$$

The collection $\{Q_{t,t'}\,; t \leq t'\}$ forms a semigroup, in the sense that $Q_{t,\tau} Q_{\tau,t'}(f) = Q_{t,t'}(f)$ for $t \leq \tau \leq t'$ [Del Moral, 2004]. We denote the set of all intermediate time points in Algorithm 1 by $\mathbb{T} = \{t_{n,s}\,; n \in 0\!:\!N{-}1, s \in 1\!:\!S\}$. Given that one has filtered particles $\{X_t^{F,j}\,; j \in 1 : J\}$ at time $t \in \mathbb{T}$, we define for $t' \in \mathbb{T} \cap [t, t_N]$

$$b_{t,t'}(f) := \int \frac{Q_{t,t'}(u_{t'} \cdot f)}{u_t} \, dF_{t,J}^F \tag{2.14}$$

for all bounded measurable functions f on $\mathbb{X}$. Note that this definition implies $b_{t,t}(f) = \int f \, dF_{t,J}^F$. If $t = t_{n,s}$ for some $s \in 1\!:\!S$ and $n \in 0\!:\!N{-}1$, we will write

$t^- = t_{n,s-1}$. If $t = t_m$ for some $m \in 0\!:\!N$, we will write $n(t) = m$. Since resampling weights at time t are proportional to $w_t(X_t^{P,j}, X_{t^-}^{F,j})$, we have

$$\mathbb{E}\left[\int f(x_t)\, dF_{t,J}^F(x_t) \middle| \mathcal{B}_{t,J}^P\right] = \frac{\displaystyle\int f(x_t) \cdot w_t(x_t, x_{t^-}) dH_{t,J}(x_t, x_{t^-})}{\displaystyle\int w_t(x_t, x_{t^-}) dH_{t,J}(x_t, x_{t^-})}. \tag{2.15}$$

The conditional expectation of the numerator in the above expression with respect to $\mathcal{B}_{t^-,J}^F$ equals

$$\begin{aligned}
&\mathbb{E}\left[\int f(x_t) \cdot w_t(x_t, x_{t^-}) dH_{t,J}(x_t, x_{t^-}) \middle| \mathcal{B}_{t^-,J}^F\right] \\
&= \begin{cases} \displaystyle\int \frac{K_{t^-,t}(u_t \cdot f)}{u_{t^-}}\, dF_{t^-,J}^F & \text{if } t^- \notin t_{1:N} \\ \displaystyle\int \frac{K_{t^-,t}(u_t \cdot f)}{u_{t^-}} \cdot g_{n(t^-)}\, dF_{t^-,J}^F & \text{if } t^- \in t_{1:N} \end{cases} \\
&= \int \frac{Q_{t^-,t}(u_t \cdot f)}{u_{t^-}}\, dF_{t^-,J}^F = b_{t^-,t}(f),
\end{aligned} \tag{2.16}$$

by (2.2), (2.12), (2.13), and (2.14). Note that here we implicitly assumed that $g_{n(t^-)}$ is a function of X_{t^-}, such that $g_{n(t^-)}(X_{t^-}) := g_{n(t^-)}(y_{n(t^-)} \,|\, X_{t^-})$. At time t_N, we are interested in knowing how accurate the quantity $b_{t_N,t_N}(f) = \frac{1}{J}\sum_{j=1}^J f(X_{t_N}^{F,j})$ is as an approximation to $\mathbb{E}[f(X_{t_N})|Y_{1:N} = y_{1:N}]$. We establish a bound on the error in this approximation under a set of assumptions.

Our first assumption concerns how close the guide function u_t is to the predictive likelihood of B future observations. In what follows, if $t = t_{m,s}$ for some $s \in 1\!:\!S$, we will write $t^\rightarrow := t_{(m+B)\wedge N}$, where we write $a \wedge b = \min(a, b)$. The expression $Q_{t,t^\rightarrow}(g_{n(t^\rightarrow)})(x)$ denotes the predictive likelihood of $Y_{m+1:n(t^\rightarrow)} = y_{m+1:n(t^\rightarrow)}$ given $X_t = x$, see (2.11).

Assumption 1. *There exists a constant $C_1 \geq 1$ such that for all $t \in \mathbb{T}$,*

$$\frac{Q_{t,t^\rightarrow}\left(g_{n(t^\rightarrow)}\right)}{u_t}(x) \leq C_1 \frac{Q_{t,t^\rightarrow}\left(g_{n(t^\rightarrow)}\right)}{u_t}(x'), \tag{2.17}$$

for all $x, x' \in \mathbb{X}$. In particular, if $t \in (t_{N-B}, t_N] \cap \mathbb{T}$,

$$\frac{Q_{t,t_N}(g_N)}{u_t}(x) \leq C_1 \frac{Q_{t,t_N}(g_N)}{u_t}(x'),$$

for all $x, x' \in \mathbb{X}$.

Uniform bounds across $\mathbb{X}$ as in (2.17) typically follows as a result of the compactness of the space $\mathbb{X}$ and the continuity of the functions being bounded. However, we do not expect that the compactness condition is critical in real applications of the algorithm.

Our second assumption says that the predictive likelihood of future observations experiences a bounded change between two consecutive intermediate points t^- and t. Specifically, we assume that conditioned on X_{t^-}, the predictive likelihood given X_t has bounded variance relative to the square of its mean.

Assumption 2. *There exists $C_2 \geq 1$ such that for all $t \in \mathbb{T}$ and for all $x \in \mathbb{X}$,*

$$\frac{K_{t^-,t}\left\{Q_{t,t^\rightarrow}\left(g_{n(t^\rightarrow)}\right)\right\}^2}{\left[K_{t^-,t}Q_{t,t^\rightarrow}\left(g_{n(t^\rightarrow)}\right)\right]^2}(x) \leq C_2^2.$$

Assumption 2 is related to a key reason that a GIRF operates on a refined time scale. If the time interval was not divided, the constant C_2 would typically increase exponentially as the space dimension d increases. To see this, if we consider a POMP consisting of d IID one-dimensional processes, the predictive likelihood $Q_{t,t^\rightarrow}(g_{n(t^\rightarrow)})(X_t)$ will be expressed as a product of d independent random variables. Thus both its mean and variance will be exponential in d. In a GIRF, however, the constant C_2 can be of constant order in d, if we divide the time interval into d sub-intervals. Examples are given in the online supplementary text 2.D to illustrate this point.

We lastly assume that the POMP has a reasonable amount of conditional mixing. We note that, when $t_m < t \leq t_{m+1}$, the likelihood of $Y_{m+B+1:N} = y_{m+B+1:N}$

conditioned on $Y_{m+1:m+B} = y_{m+1:m+B}$ and the current state X_t is given by

$$\begin{aligned} &p_{Y_{m+B+1:N}\,|\,X_t, Y_{m+1:m+B}}\left(y_{m+B+1:N}\,\middle|\,x,\ y_{m+1:m+B}\right) \\ &\quad = \frac{p_{Y_{m+1:N}\,|\,X_t}\left(y_{m+1:N}\,\middle|\,x\right)}{p_{Y_{m+1:m+B}\,|\,X_t}\left(y_{m+1:m+B}\,\middle|\,x\right)} = \frac{Q_{t,t_N}\left(g_N\right)}{Q_{t,t^{\rightarrow}}\left(g_{n(t^{\rightarrow})}\right)}(x). \end{aligned}$$

Also, for any bounded measurable function f on $\mathbb{X}$ we have

$$\mathbb{E}\left[f(X_{t_N})\,|\,X_t = x, Y_{m+1:N} = y_{m+1:N}\right] = \frac{Q_{t,t_N}(g_N \cdot f)}{Q_{t,t_N}(g_N)}(x).$$

Assumption 3. *There exist constants $C_3 \geq 1$ such that for all $t \in \mathbb{T}$,*

$$\frac{Q_{t,t_N}(g_N)}{Q_{t,t^{\rightarrow}}\left(g_{n(t^{\rightarrow})}\right)}(x) \leq C_3 \frac{Q_{t,t_N}(g_N)}{Q_{t,t^{\rightarrow}}\left(g_{n(t^{\rightarrow})}\right)}(x') \tag{2.18}$$

for all $x, x' \in \mathbb{X}$. Also, there exists $n_* \in 1\!:\!N{-}1$ and $C_4 \in (0,1)$ such that for any measurable function f with $\|f\|_\infty \leq 1$,

$$\left|\frac{Q_{t_{n_*},t_N}(g_N \cdot f)}{Q_{t_{n_*},t_N}(g_N)}(x) - \frac{Q_{t_{n_*},t_N}(g_N \cdot f)}{Q_{t_{n_*},t_N}(g_N)}(x')\right| \leq C_4 \tag{2.19}$$

for all $x, x' \in \mathbb{X}$.

The first inequality (2.18) states that, conditioned on the observations $y_{m+1:(m+B)\wedge N}$, the probability of having the later observations $y_{(m+B+1)\wedge N:N}$ has bounded dependence on the current state X_t. One can make C_3 smaller by taking B larger, because more conditional mixing will happen in the longer interval $[t, t_{m+B+1}]$. The second inequality (2.19) says that the state at time t_{n_*} has bounded influence on the state at t_N, conditional on the observations made after time t_{n_*}. One can similarly make C_4 smaller by taking n_* more distant from N.

We also assume that multinomial resampling is used in Algorithm 1. The indices a^j are drawn independently of each other given $\{w^j\,; j \in 1\!:\!J\}$ under multinomial resampling.

Theorem II.2. *Suppose Assumption 1, 2, and 3 hold and multinomial resampling is used. Then for any f with $\|f\|_\infty \leq 1$ and for any $a > 1$,*

$$\left|\int f dF^F_{t_N,J} - \mathbb{E}[f(X_{t_N})|Y_{1:N} = y_{1:N}]\right| \leq C_4 + \frac{2aC_1}{\sqrt{J}}(C_2+1)\big[S\left\{C_3(N-B-n_*)+B\right\}+C_3-1\big] \tag{2.20}$$

with probability at least $1 - \frac{4S(N-n_*)}{a^2}$.

Proof. See Section 2.C. □

Theorem II.2 states that for a given POMP model, the size of the error in the estimated filtering distribution will be bounded by a number that increases linearly in S with high probability, provided that a is large. On the other hand, if we are to keep the probability $\frac{S(N-n_*)}{a^2}$ with which the bound is violated at a fixed level, the number a needs to increase proportionally to $\sqrt{S}$, and thus the error bound increases at a rate of $O(S^{\frac{3}{2}})$. Although this seems to suggest that the error bound increases if we take larger S, the bound will actually be reduced if we take $S = d$ instead of $S = 1$ due to the scaling property of C_2, which may be of order $O(1)$ in d when $S = d$ (two examples with bounded C_2 are given in the supplementary text, Section 2.D).

However, C_1 will generally scale exponentially in space dimension d. When we consider d IID one dimensional processes again, the obvious choice of $u_t(x)$ we take as the product of d identical copies one dimensional guide function $\tilde{u}_t$, that is $u_t(x_1, \ldots, x_d) := \prod_{k=1}^d \tilde{u}_t(x_k)$, will make the number C_1 increase in the form of c_1^d for some $c_1 \geq 1$. In this case, c_1 will represent the maximum discrepancy between the one-dimensional predictive likelihood and the guide function $\tilde{u}_t$, and the closer the approximation becomes, the closer c_1 will be to one. Thus good guide functions that are closer to the predictive likelihood can reduce c_1 and slow down the rate at which C_1 increases. If u_t exactly equals the predictive likelihood, C_1 equals unity.

Section 2.6.1 shows that the GIRF can empirically scale approximately polynomially in d in this case.

Two competing factors, namely temporal mixing and spatial dimensionality, determine the magnitude of C_3 and C_4. Reasonably small C_3 and C_4 are possible in high dimensions, especially when the model is composed of weakly coupled processes, each of which has good marginal mixing. In a model consisting of d copies of IID processes, the speed of conditional mixing is unrelated to d. For many applications, the speed of mixing may be only loosely dependent on d.

The bound (2.20) has a term C_4, which does not vanish as J increases to infinity. One could have a bound without C_4, but that bound grows linearly with N instead of $N - n_*$. When there are many observations, this alternative bound will be larger than the bound given in (2.20), which makes use of the conditional mixing over the time interval $[t_{n_*}, t_N]$.

The implications of Theorem II.2 may be summarized as follows. The error bound decreases as the predictive likelihood can be accurately approximated (small C_1), each observation time interval is divided in number that is at least comparable to the space dimension (small C_2), and the conditional mixing happens relatively fast (small C_3 and C_4).

2.5 Parameter estimation with iterated filtering

Our GIRF can be easily combined with existing plug-and-play parameter inference methods that build upon the particle filter. The iterated filtering algorithm of Ionides et al. [2015] finds the maximum likelihood estimate (MLE) of multi-dimensional parameters via an SMC approximation to an iterated, perturbed Bayes map. This algorithm, when implemented via a plug-and-play SMC filtering approach, provides

Algorithm 2: An iterated guided intermediate resampling filter for parameter estimation

```
Input  : Simulator for $\mu_{t_0}(dx\,;\theta)$
         Simulator for $K_{t_{n,s-1},t_{n,s}}\big(dx\,;x_{t_{n,s}},\theta\big)$ for $n \in 0\,{:}\,N{-}1$ and $s \in 1\,{:}\,S$
         Evaluator for $g_n(y_n \,|\, x_{t_n},\theta)$ for $n \in 1\,{:}\,N$
         Evaluator for $u_{t_{n,s}}\big(x_{t_{n,s}},\theta\big)$ for $n \in 0\,{:}\,N{-}1$ and $s \in 1\,{:}\,S$
         Data, $y_{1:N}$
         Number of particles, $J$
         Number of iterations, $M$
         Initial parameter swarm, $\big\{\Theta^{0,j}\,;j \in 1\,{:}\,J\big\}$
         Perturbation kernel for initial value parameter, $\kappa_0(d\theta\,;\phi,\sigma)$
         Perturbation kernel, $\kappa_{n,s}(d\theta\,;\phi,\sigma)$ for $n \in 0\,{:}\,N{-}1$ and $s \in 1\,{:}\,S$
         Sequence of perturbation sizes, $\sigma_{1\,:\,M}$
Output: Final parameter swarm $\big\{\Theta^{M,j}\,;j \in 1\,{:}\,J\big\}$
for $m \leftarrow 1\,{:}\,M$ do
    Run Algorithm 1 on the extended state space $\Big(X_{t_{n,s}},\Theta^m_{t_{n,s}}\Big)$ with initial draws from
     (2.21) and subsequent draws from (2.22)
    Set $\Theta^{m,j} = \Theta^{F,m,j}_{t_N}$ for $j \in 1\,{:}\,J$
end
```

plug-and-play inference on unknown model parameters. Iterated filtering runs a sequence of particle filter on the augmented space comprising the state variable and the parameter, where the parameters are subject to random perturbations at each time point. The size of perturbations decrease over iterations to induce convergence. In the limit where the perturbation size approaches zero, Ionides et al. [2015] showed that the distribution of filtered parameters approaches a point mass at the MLE under regularity conditions.

Iterated filtering starts with an initial set of parameters $\{\Theta^{0,j}\,; j \in 1\,{:}\,J\}$. The parameter vector may contain initial value parameters (IVPs) which encode the value of X_{t_0} but play no role in the dynamics of the system. At the start of the m-th iteration, the parameter component of each particle is perturbed from its current position $\Theta^{m-1,j}$ with kernel κ_0. The IVPs are only perturbed at this point. A preset decreasing sequence $(\sigma_m)_{m=1:M}$ determines the size of perturbation. The initial state variables $X^{F,j}_{t_0}$ are drawn from the parameterized initial state distribution. The initialization can be summarized as follows:

$$\Theta^{F,m,j}_{t_0} \sim \kappa_0\left(d\theta\,;\Theta^{m-1,j},\sigma_m\right), \qquad X^{F,j}_{t_0} \sim \mu_{t_0}\left(dx;\Theta^{F,m,j}_{t_0}\right). \tag{2.21}$$

The usual filtering procedure follows, where the non-IVPs are continuously perturbed. In case where a GIRF is run within iterated filtering, the non-IVPs are perturbed at each intermediate time $t_{n,s}$ with kernel $\kappa_{n,s}$. The states are then drawn from the parameterized transition kernel:

$$\Theta^{P,m,j}_{t_{n,s}} \sim \kappa_{n,s}\left(d\theta\,;\Theta^{F,m,j}_{t_{n,s-1}},\sigma_m\right), \qquad X^{P,j}_{t_{n,s}} \sim K_{t_{n,s-1},t_{n,s}}\left(dx\,;X^{F,j}_{t_{n,s-1}},\Theta^{P,m,j}_{t_{n,s}}\right). \tag{2.22}$$

The weighting and resampling steps are as usual. At the end of filtering, the parameter swarm $\Theta^{F,m,j}_{t_N}$ are set as $\Theta^{m,j}$. After M iterations, the final parameter swarm $\Theta^{M,j}$, which have converged almost to a single point, are considered as the MLE. The pseudocode for the iterated GIRF is given in Algorithm 2.

A GIRF may in theory be combined with the particle Markov chain Monte Carlo (PMCMC) method proposed in Andrieu et al. [2010]. However, when the likelihood estimates have high Monte Carlo error variance, the mixing of the Markov chain can be extremely slow. MCMC methods running with unbiased likelihood estimators, including the PMCMC, are known to achieve best efficiency when the errors in the likelihood estimates are about one log unit [Doucet et al., 2015]. Unfortunately, these errors can be well over one hundred log units in high dimensional models. Iterated filtering can be more useful in these scenarios, because inference can still be made from the noisy maximum likelihood estimates. For inference on a parameter of interest, one can estimate the profile likelihood curve by maximizing over all other parameters and use this profile to obtain approximate MLEs and confidence intervals [Ionides et al., 2017]. We demonstrate this approach in our analysis of spatiotemporal data of Section 3.1.

2.6 Implementation

We implemented our algorithm on high dimensional Gaussian processes. The guide function u_t was taken to be an approximation to the predictive likelihood of $B = 2$ future observations. We observed that if we took $B = 1$, the resampling weights at s=1 were more unbalanced, and consequently the errors in the filter estimates were larger.

2.6.1 Correlated Brownian motion

In order to see how the performance of the GIRF depends on the space dimension and the choice of the guide function, we first applied our algorithm to multi-dimensional correlated Brownian motions. Each dimension of the Brownian motion was identically distributed with increments per unit time having mean zero and process noise variance σ_p^2. The correlation coefficient matrix A for the increments was chosen such that its all off-diagonal entries equaled α. The initial state distribution at time $t_0 = 0$ was given by the point mass at the origin of $\mathbb{R}^d$. Measurements were made at positive integer time points $1\!:\!50$, with independent Gaussian noises of mean zero and measurement error variance σ_m^2. The POMP model can be expressed as follows, where I denotes the d dimensional identity matrix:

$$X_{t+\delta} = X_t + N\left(0, \sigma_p^2 \delta A\right), \qquad Y_t = X_t + N\left(0, \sigma_m^2 I\right).$$

The guide function $u_{t_{n,s}}$ was defined as in (2.8), where the approximate predictive likelihood $u_{t_{n,s} \nearrow t_{n+b}}$ was chosen as described in (2.9). Since the process had zero drift, the forward state projection by the deterministic mean process was given by $\tilde{x}_{t_{n+b}} = x_{t_{n,s}}$. The variance of $X_{t_{n+b}}$ conditioned on $X_{t_{n,s}} = x_{t_{n,s}}$ was equal to

$(t_{n+b} - t_{n,s}) \cdot \sigma_p^2 A$, so the guide function was defined as

$$u_{t_{n,s}}\left(x_{t_{n,s}}\right) = \prod_{b=1}^{B} \phi_d\left[y_{t_{n+b}}\,; x_{t_{n,s}},\, (t_{n+b} - t_{n,s}) \cdot \sigma_p^2 A + \sigma_m^2 I\right], \tag{2.23}$$

where $\phi_d(\,\cdot\,; \mu, \Sigma)$ denotes the density of the d-dimensional Gaussian distribution with mean μ and variance Σ. Evaluating (2.23) typically requires procedures such as the Cholesky decomposition and takes $O\left(d^3\right)$ computations. Since this could be demanding for large d, we also used an approximation of (2.23) obtained by ignoring the off-diagonal elements of A,

$$u_{t_{n,s}}\left(x_{t_{n,s}}\right) = \prod_{b=1}^{B} \phi_d\left[y_{t_{n+b}}\,; x_{t_{n,s}},\, \left\{(t_{n+b} - t_{n,s}) \cdot \sigma_p^2 + \sigma_m^2\right\} I\right]. \tag{2.24}$$

We implemented our GIRF on this model with varying dimensions and correlation coefficients. All data were generated with $\sigma_p = \sigma_m = 1$. The number of sub-intervals within a unit time interval S was taken to equal to the dimension d. The guide function u_t approximated the predictive likelihood of two future observations (i.e., $B = 2$). We parallelized the computation by applying the island particle method of Vergé et al. [2015] to Algorithm 1 in a straightforward way. Sixty particle islands with one thousand particles in each island were used in all experiments.

In our first set of experiments, we varied the space dimension from twenty to fifty, one hundred, and two hundred while fixing the correlation coefficient at zero. Each filtering on average took 15 seconds, 81 seconds, 5 minutes, and 20 minutes respectively. Figure 2.3 shows the mean squared error (MSE) of the estimates of the filtering mean at time 50. Exact values of the filtering means and the likelihoods of data for this linear Gaussian model were computed by Kalman filtering. The plotted values represent the average over all d components. The results were obtained from forty independent filtering repetitions for each case. The estimated squared biases, shown in triangles, were roughly $\frac{1}{40}$ times the MSE, meaning that the estimator was

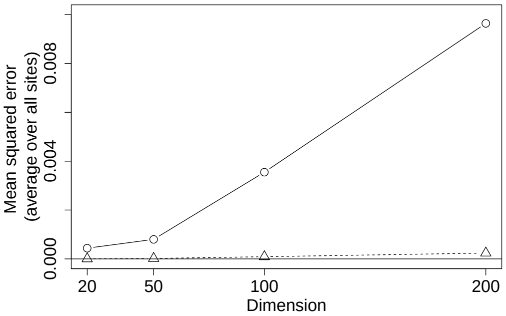

Figure 2.3: The MSE of the estimates of the filtering mean: ○, MSE; △, bias squared

	Space dimension			
Likelihood	20	50	100	200
True	-1916.30	-4703.83	-9499.10	-18908.62
Estimate	-1916.28 (0.06)	-4703.72 (0.17)	-9501.20 (0.36)	-18932.69 (0.87)

Table 2.1: Log likelihood estimates on a correlated linear Gaussian model with varying dimensions

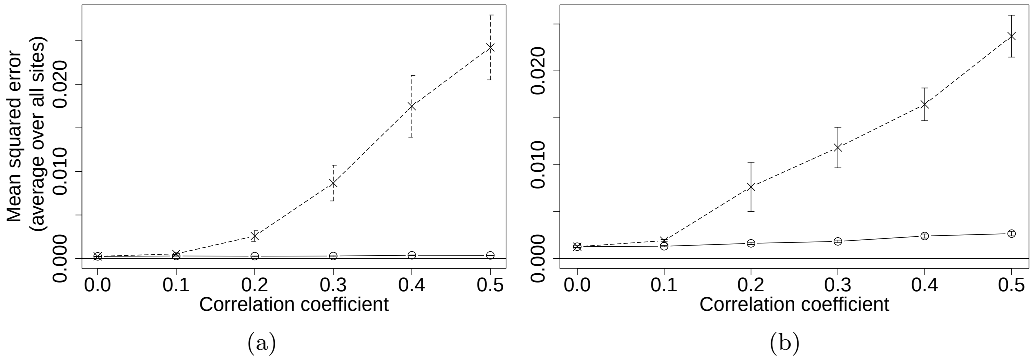

Figure 2.4: The MSE of the estimates of the filtering mean under varying degrees of correlation, (a) $d = 20$, (b) $d = 50$: ○, exact covariance used; ×, diagonal covariance used

	Correlation coefficient					
Likelihood	0.0	0.1	0.2	0.3	0.4	0.5
True	-1904.04	-1897.75	-1884.24	-1866.33	-1844.90	-1820.02
Estimate	-1903.92	-1897.71	-1884.25	-1866.31	-1844.90	-1820.05
[exact covariance]	(0.05)	(0.05)	(0.06)	(0.05)	(0.06)	(0.06)
Estimate	-1903.92	-1897.79	-1884.91	-1868.35	-1852.62	-1831.77
[diagonal covariance]	(0.05)	(0.09)	(0.20)	(0.59)	(0.44)	(0.71)
True	-4790.18	-4750.63	-4701.90	-4644.46	-4579.29	-4505.73
Estimate	-4790.49	-4750.35	-4702.02	-4644.88	-4579.82	-4505.96
[exact covariance]	(0.19)	(0.24)	(0.29)	(0.27)	(0.38)	(0.62)
Estimate	-4790.49	-4754.44	-4722.03	-4685.83	-4649.51	-4609.30
[diagonal covariance]	(0.19)	(0.43)	(0.57)	(0.68)	(0.89)	(0.83)

Table 2.2: Log likelihood estimates under varying degrees of correlation, top, $d = 20$; bottom, $d = 50$

effectively unbiased. The MSE at dimension two hundred of less than 0.01 was very small compared to the variance of the exact filtering distribution at time 50, which was 0.62. These results demonstrate that our GIRF scales much better than the standard particle filter. Snyder et al. [2008] reported that at least 10^{11} particles were required for the same problem in two hundred dimension.

The estimated log likelihoods, shown in Table 2.1, were also surprisingly accurate. The estimated standard errors of the log likelihoods are shown in parentheses. In dimensions twenty and fifty, the true likelihood was well within one standard error of the likelihood estimates. In dimensions one hundred and two hundred, the likelihood estimates are more than one standard error below the true likelihood. Since the likelihood estimator is guaranteed to be unbiased, this shows that the likelihood estimate is above the true value with small probability, while the likelihood estimate is below the true value with high probability. This phenomenon reflects that filtering becomes less accurate as the dimension increases.

In the second set of experiments, the dimension was fixed at either twenty or fifty, and the correlation coefficient varied from 0 to 0.5 with intervals of 0.1. Figure 2.4 shows the MSE of the estimates of the filtering mean at time 50. The error bars indicate the sizes of the standard errors of the MSE. When the guide function used the exact covariance as in (2.23), the MSEs were almost constant or increasing very slowly as the correlation α increased. When we used the diagonal approximation as in (2.24), the errors in the filter estimates increased much more rapidly as α increased. However, the errors were still reasonably small. The MSEs were about 0.02 both in twenty and fifty dimensions at $\alpha = 0.5$, where the variance of the filtering distribution was 0.52 and 0.51 respectively. The log likelihood estimates reported in Table 2.2 shows a similar pattern in the filtering accuracy. These results indicate that the performance of the GIRF depends on how well the guide functions approximate the predictive likelihoods of future observations. This agrees with our theoretical investigation.

2.7 Discussion

Analyses of high dimensional dynamic systems have relied on mathematical simplification of the model or information reduction technique. The ensemble Kalman filters are widely used inference methods that belong to the former category [Evensen, 1994, 2003]. The linear update scheme for the posterior state distribution in these filters helps avoid the computational challenge such as the degenerate resampling weights experienced in the particle filter and facilitates very high dimensional applications in geophysical sciences [Houtekamer and Mitchell, 2001]. However, these methods tend to produce inaccurate results when the system is highly non-linear and non-Gaussian [Lei et al., 2010, Miller et al., 1999]. For tracing epidemic peaks or fadeouts, for instance, non-linearity or the discreteness of infection cases can play a key role in the stochastic evolution of the system, making approximations by Gaussian distributions unsuitable.

Methods based on information reduction technique are generally constructed with the aid of domain experts' knowledge on the key features of the model. Approximate Bayesian computation, for example, approximates the posterior probability of a parameter θ given data using the distances between carefully chosen summary statistics of the observed data and those of simulated data under the parameter θ. This simplified approach in principle enables analyses of data for any model that can be simulated. However, information reduction methods can fail to capture full complexities in the model or result in inaccurate parameter estimates [Fasiolo et al., 2016].

Also, different conclusions might be drawn depending on the summary statistics and the distance measures being used.

The GIRF supports the likelihood-based inference on high dimensional nonlinear dynamic models. Likelihood-based inference can add to the reliability of scientific conclusions, because the likelihood of data is uniquely defined by a model and provides a common measure of fit. Sharp likelihood-based analyses using fully developed models can lead to scientific discoveries of fine resolution, which might not be obtained with other analysis methods.

The GIRF proceeds on a time scale finer than the observation time scale. At the intermediate time points, the particle ensemble approximates the state distribution conditional on the observations up to a certain future time point. Each of the increased number of resampling steps deals with minced amount of information and suffers less from the weight degeneracy. Looking to future observations for particle guidance also helps avoid depletion of particles thanks to the conditional mixing property.

Theoretical investigation revealed that our GIRF can produce accurate filtering estimates in high dimensions under certain assumptions. These conditions offer a perspective on what causes the COD and how the GIRF partially solves the problem. However, the assumptions may be difficult to check in real applications. We appeal to J. W. Tukey's advice, "So long as one does not ask for certainties one can be carefully imprecise about assumptions" [Jones, 1987]. The reliability of the filtering results can be partly checked by the effective sample size at each resampling step [Kong et al., 1994]. Appropriate simulation studies can also add to the credibility of the inference drawn from the results.

The main practical limitation in employing the GIRF is the construction of the guide function. The numerical efficiency of the filter depends on how well this guide function approximates the predictive likelihoods of future observations. A good guide function may be challenging to develop for a new application. A model with slow conditional mixing increases the difficulty, because the guide function will need to estimate the predictive likelihoods of a larger number of future observations in order to prevent particle depletion.

2.8 Appendix for chapter II

2.A Proof of Theorem II.1

We augment the state space in order to make the weight function $w_{t_{n,s}}$ defined in (2.2) depend only on the current state variable at time $t_{n,s}$. Let $\{Z_{t_{n,s}}\,; n \in 0\!:\!N-1,\ s \in 1\!:\!S\}$ be a process defined on discrete time points $\mathbb{T}$ such that $Z_{t_{n,s}} = (X_{t_{n,s}}, X_{t_{n,s-1}})$. Let $Z_{t_0} := (X_{t_0}, x^*)$ where x^* is an arbitrary point in $\mathbb{X}$. Define π_i, $i = 1, 2$, be the mapping from a pair of elements to its i-th entry (i.e., $\pi_1(x_1, x_2) = x_1,\ \forall x_1, x_2$). The transition kernel of the discrete process $\big\{Z_{t_{n,s}}\,; n = 0\!:\!N-1,\ s \in 1\!:\!S\big\}$, denoted by $\check{K}_{t_{n,s}},\colon \mathcal{B}(\mathbb{X}^2) \times \mathbb{X}^2 \to [0,1]$, satisfies that

$$\check{K}_{t_{n,s}}\left(A_1 \times A_2\,; z_{t_{n,s-1}}\right) = K_{t_{n,s-1},t_{n,s}}\left(A_1\,; \pi_1(z_{t_{n,s-1}})\right) \cdot \delta_{\pi_1\left(z_{t_{n,s-1}}\right)}\left(A_2\right), \qquad n \in 0\!:\!N-1,\ s \in 1\!:\!S \tag{2.25}$$

for $A_1, A_2 \in \mathcal{B}(\mathbb{X})$. Then the bootstrap particle filter with the initial particle draws given by $\{Z^{F,j}_{t_0}\,; j \in 1\!:\!J\} = \{(X^{F,j}_{t_0}, x^*)\,; j \in 1\!:\!J\}$, subsequent draws made according to $Z^{P,j}_{t_{n,s}} \sim \check{K}_{t_{n,s}}(\,\cdot\,; Z^{F,j}_{t_{n,s-1}})$, and resampling weights proportional to $w_{t_{n,s}}\big(\pi_1(z_{t_{n,s}}), \pi_2(z_{t_{n,s}})\big)$ from (2.2) is algorithmically equivalent to the GIRF illustrated in Algorithm 1 when we equate $Z^{P,j}_{t_{n,s}}$ with the pair $(X^{P,j}_{t_{n,s}}, X^{F,j}_{t_{n,s-1}})$ in Algorithm 1. Moreover, the likelihood estimate from this particle filter $\prod_{n=0}^{N-1}\prod_{s=1}^{S}\big[\sum_{j=1}^{J} w_{t_{n,s}}\big(\pi_1(Z^{P,j}_{t_{n,s}}), \pi_2(Z^{P,j}_{t_{n,s}})\big)\big]$ is exactly the same as $\hat{\ell}$ in Algorithm 1. Therefore, the likelihood estimate obtained from this particle filter is unbiased for

$$\mathbb{E}\left[\prod_{n=0}^{N-1}\prod_{s=1}^{S} w_{t_{n,s}}\left(\pi_1(Z_{t_{n,s}}), \pi_2(Z_{t_{n,s}})\right)\right] = \mathbb{E}\left[\prod_{n=1}^{N} g_n(y_n \,|\, X_{t_n})\right],$$

due to the unbiasedness property for the standard particle filters [Del Moral and Jacod, 2001]. The equality in the above equation comes from (2.4). We conclude that $\hat{\ell}$ is an unbiased estimate for $\ell_{1:N}(y_{1:N})$.

2.B A heuristic argument for the stability of the likelihood estimate obtained by a GIRF

We provide an argument for the claim that the likelihood estimate

$$\hat{\ell} = \prod_{n=0}^{N-1} \prod_{s=1}^{S} \frac{1}{J} \sum_{j=1}^{J} w_{t_{n,s}} \left(X^{P,j}_{t_{n,s}}, X^{F,j}_{t_{n,s-1}} \right) \tag{2.26}$$

obtained by the GIRF proposed in Algorithm 1 can be in general much more stable than the likelihood estimate obtained by the standard particle filter

$$\hat{\ell}^{std} = \prod_{n=1}^{N} \frac{1}{J} \sum_{j=1}^{J} g_n \left(y_n \,\middle|\, X^{P,j}_{t_n} \right). \tag{2.27}$$

Theorem II.2 shows that the average of the weight terms $\frac{1}{J}\sum_{j=1}^{J} w_{t_{n,s}}(X^{P,j}_{t_{n,s}}, X^{F,j}_{t_{n,s-1}})$ in (2.26) is close to

$$\mathbb{E}_{X_{t_{n,s-1}} \sim p^{u,y}_{t_{n,s-1}}} w_{t_{n,s}} \left(X_{t_{n,s}}, X_{t_{n,s-1}} \right)$$

with high probability, where the distribution of $X_{t_{n,s-1}}$ has density proportional to

$$p^{u,y}_{t_{n,s-1}}(x_{t_{n,s-1}}) \propto p_{X_{t_{n,s-1}}|Y_{1:n'}}(x_{t_{n,s-1}} \,|\, y_{1:n'}) \cdot u_{t_{n,s-1}}(x_{t_{n,s-1}}).$$

Here, $n' = n$ if $s \geq 2$ and $n' = n-1$ if $s = 1$. Since this Monte Carlo estimate $\frac{1}{J}\sum_{j=1}^{J} w_{t_{n,s}}(X^{P,j}_{t_{n,s}}, X^{F,j}_{t_{n,s-1}})$ at each intermediate time point asymptotically converges to the expected value it approximates, the product of these estimates also asymptotically converges to the target expected value, namely the likelihood of data. However, the core reason that the likelihood estimate (2.26) is relatively more stable with finite sample comes from the fact that (2.26) is the product of averages of the Monte Carlo weights whereas (2.27) is the average of the products of the Monte Carlo weights. This comparison is analogous to the relationship between the likelihood estimates from SMC methods and those from sequential importance sampling methods, which tend to be numerically unstable due to the lack of resampling steps. We will make this statement clear as follows.

We will consider the case where we take $u_{t_n}(x) = g_n(y_n \,|\, x_{t_n})$ for $n \in 1\!:\!N$ in Algorithm 1. We will compare the n-th terms in the products (2.26) and (2.27). We denote the ancestor particle of $X^{P,j}_{t_n}$ at time $t_{n-1,s}$ as $X^{F,a^j_{t_{n-1,s}}}_{t_{n-1,s}}$ for $s \in 1\!:\!S{-}1$, with $a^j_{t_{n,S-1}} = j$. Note that this definition results in $X^{F,a^j_{t_{n,s}}}_{t_{n,s}} = X^{P,a^j_{t_{n,s-1}}}_{t_{n,s}}$. Then, the measurement density of y_n at t_n can be expressed as a product of weights defined in (2.2),

$$\begin{aligned}
g_n\left(y_n \,\middle|\, X^{P,j}_{t_n}\right) &= \left\{ \prod_{s=1}^{S} \frac{u_{t_{n-1,s}}\left(X^{P,a^j_{t_{n-1,s-1}}}_{t_{n-1,s}}\right)}{u_{t_{n-1,s-1}}\left(X^{F,a^j_{t_{n-1,s-1}}}_{t_{n-1,s-1}}\right)} \right\} \cdot g_{n-1}\left(y_{n-1} \,\middle|\, X^{F,a^j_{t_{n-1}}}_{t_{n-1}}\right) \\
&= \prod_{s=1}^{S} w_{t_{n-1,s}}\left(X^{P,a^j_{t_{n-1,s-1}}}_{t_{n-1,s}}, X^{F,a^j_{t_{n-1,s-1}}}_{t_{n-1,s-1}}\right).
\end{aligned}$$

We write $W^j_s = w_{t_{n-1,s}}(X^{P,a^j_{t_{n-1,s-1}}}_{t_{n-1,s}}, X^{F,a^j_{t_{n-1,s-1}}}_{t_{n-1,s-1}})$. The weight terms $\{W^j_s\,;j \in 1\!:\!J\}$ at the s-th step are conditionally independent of each other given the particle draws $\{X^{F,a^j_{t_{n-1,s-1}}}_{t_{n-1,s-1}}\,;j \in 1\!:\!J\}$. For simplicity of argument, we assume that all weight terms W^j_s for $s \in 1\!:\!S$ and $j \in 1\!:\!J$ are IID with mean μ and variance σ^2. The logarithm of the n-th term in the expression for $\hat{\ell}$ given by (2.26) converges to a normal distribution

$$\log \prod_{s=1}^{S} \frac{1}{J} \sum_{j=1}^{J} W^j_s \Rightarrow N\left(S \log \mu,\, \frac{S\sigma^2}{J\mu^2}\right)$$

as J tends to infinity by the central limit theorem and the delta method. As for the n-th term of $\hat{\ell}^{std}$, we first observe that the variance of $\prod_{s=1}^{S} W^j_s$ is given by

$$\operatorname{Var}\left(\prod_{s=1}^{S} W^j_s\right) = \mathbb{E}\prod_{s=1}^{S}\left(W^j_s\right)^2 - \left(\mathbb{E}\prod_{s=1}^{S} W^j_s\right)^2 = \left(\mu^2+\sigma^2\right)^S - \mu^{2S}. \tag{2.28}$$

Thus the logarithm of the n-th term in the expression for $\hat{\ell}^{std}$ given by (2.27) converges in distribution

$$\log \frac{1}{J} \sum_{j=1}^{J} \prod_{s=1}^{S} W^j_s \Rightarrow N\left(\log \mu^S,\, \frac{\left(\mu^2+\sigma^2\right)^S - \mu^{2S}}{J\mu^{2S}}\right) \tag{2.29}$$

as J tends to infinity. We can see that the asymptotic variance $\frac{\left(\mu^2+\sigma^2\right)^S-\mu^{2S}}{J\mu^{2S}}$ in (2.29) is larger than the asymptotic variance $\frac{S\sigma^2}{J\mu^2}$ in (2.28) roughly by a factor of $\frac{2^S}{S}$ when $\mu^2 \approx \sigma^2$. Thus in high dimensions where we take $S \approx d$, the variance of the log likelihood estimate $\log \hat{\ell}$ by the GIRF can be much smaller than the variance of the log likelihood estimate $\log \hat{\ell}^{std}$ from the standard particle filter.

2.C Proof of Theorem II.2

Our main theoretical innovation in this proof is the novel way of bounding the error terms in the telescoping series (2.32) below.

Let f be a measurable function such that $\|f\|_\infty \leq 1$. One can observe that

$$\mathbb{E}\left[f(X_{t_N})|Y_{1:N}=y_{1:N}\right] = \frac{\mathbb{E}\left[f(X_{t_N})\cdot\prod_{n=1}^{N} g_n\left(y_n\,|\,X_{t_n}\right)\right]}{\mathbb{E}\left[\prod_{n=1}^{N} g_n\left(y_n\,|\,X_{t_n}\right)\right]} = \frac{\mu_{t_0}Q_{t_0,t_N}(g_N\cdot f)}{\mu_{t_0}Q_{t_0,t_N}(g_N)}.$$

We first show that at time t_{n_*},

$$\left|\frac{b_{t_{n_*},t_N}(f)}{b_{t_{n_*},t_N}(1)} - \frac{\mu_{t_0}Q_{t_0,t_N}(g_N\cdot f)}{\mu_{t_0}Q_{t_0,t_N}(g_N)}\right| \leq C_4 \tag{2.30}$$

for $b_{t,t'}$ defined in (2.14). Define the probability distribution $\eta_{t_{n_*}}$ such that

$$\eta_{t_{n_*}}f = \frac{\mu_{t_0}Q_{t_0,t_{n_*}}\left\{Q_{t_{n_*},t_N}(g_N)\cdot f\right\}}{\mu_{t_0}Q_{t_0,t_{n_*}}\left\{Q_{t_{n_*},t_N}(g_N)\right\}} = \frac{\mathbb{E}\left[\prod_{n=1}^{N} g_n\left(y_n\,|\,X_{t_n}\right)\cdot f\left(X_{t_{n_*}}\right)\right]}{\mathbb{E}\left[\prod_{n=1}^{N} g_n\left(y_n\,|\,X_{t_n}\right)\right]}$$

for bounded measurable functions f. Indeed, $\eta_{t_{n_*}}$ is the smoothing distribution of $X_{t_{n_*}}$ conditioned on the observations $Y_{1:N}=y_{1:N}$. We have

$$\int \frac{Q_{t_{n_*},t_N}\left(g_N\cdot f\right)}{Q_{t_{n_*},t_N}\left(g_N\right)}d\eta_{t_{n_*}} = \frac{\mu_{t_0}Q_{t_0,t_{n_*}}\left\{Q_{t_{n_*},t_N}(g_N)\cdot\frac{Q_{t_{n_*},t_N}(g_N\cdot f)}{Q_{t_{n_*},t_N}(g_N)}\right\}}{\mu_{t_0}Q_{t_0,t_{n_*}}Q_{t_{n_*},t_N}(g_N)} = \frac{\mu_{t_0}Q_{t_0,t_N}(g_N\cdot f)}{\mu_{t_0}Q_{t_0,t_N}(g_N)}.$$

Thus,

$$\inf_{x\in\mathbb{X}}\frac{Q_{t_{n_*},t_N}(g_N\cdot f)}{Q_{t_{n_*},t_N}(g_N)}(x) \leq \frac{\mu_{t_0}Q_{t_0,t_N}(g_N\cdot f)}{\mu_{t_0}Q_{t_0,t_N}(g_N)} \leq \sup_{x\in\mathbb{X}}\frac{Q_{t_{n_*},t_N}(g_N\cdot f)}{Q_{t_{n_*},t_N}(g_N)}(x).$$

Also, since $\left\{X^{F,j}_{t_{n*}}; j \in 1\colon J\right\}$ takes values in $\mathbb{X}$,

$$\inf_{x\in\mathbb{X}} \frac{Q_{t_{n*},t_N}(g_N\cdot f)}{Q_{t_{n*},t_N}(g_N)}(x) \le \frac{\dfrac{Q_{t_{n*},t_N}(g_N\cdot f)}{u_{t_{n*}}}\left(X^{F,j}_{t_{n*}}\right)}{\dfrac{Q_{t_{n*},t_N}(g_N)}{u_{t_{n*}}}\left(X^{F,j}_{t_{n*}}\right)} \le \sup_{x\in\mathbb{X}} \frac{Q_{t_{n*},t_N}(g_N\cdot f)}{Q_{t_{n*},t_N}(g_N)}(x).$$

Thus,

$$\inf_{x\in\mathbb{X}} \frac{Q_{t_{n*},t_N}(g_N\cdot f)}{Q_{t_{n*},t_N}(g_N)}(x) \le \frac{b_{t_{n*},t_N}(f)}{b_{t_{n*},t_N}(1)} = \frac{\dfrac{1}{J}\displaystyle\sum_j \dfrac{Q_{t_{n*},t_N}(g_N\cdot f)}{u_{t_{n*}}}\left(X^{F,j}_{t_{n*}}\right)}{\dfrac{1}{J}\displaystyle\sum_j \dfrac{Q_{t_{n*},t_N}(g_N)}{u_{t_{n*}}}\left(X^{F,j}_{t_{n*}}\right)} \le \sup_{x\in\mathbb{X}} \frac{Q_{t_{n*},t_N}(g_N\cdot f)}{Q_{t_{n*},t_N}(g_N)}(x). \tag{2.31}$$

It follows from Assumption 3 that

$$\left|\frac{b_{t_{n*},t_N}(f)}{b_{t_{n*},t_N}(1)} - \frac{\mu_{t_0}Q_{t_0,t_N}(g_N\cdot f)}{\mu_{t_0}Q_{t_0,t_N}(g_N)}\right| \le \sup_{x\in\mathbb{X}} \frac{Q_{t_{n*},t_N}(g_N\cdot f)}{Q_{t_{n*},t_N}(g_N)}(x) - \inf_{x\in\mathbb{X}} \frac{Q_{t_{n*},t_N}(g_N\cdot f)}{Q_{t_{n*},t_N}(g_N)}(x) \le C_4,$$

as claimed in (2.30). Next, we bound the difference

$$\left|b_{t_N,t_N}(f) - \frac{b_{t_{n*},t_N}(f)}{b_{t_{n*},t_N}(1)}\right|.$$

We write this difference as a telescoping series.

$$\left|b_{t_N,t_N}(f) - \frac{b_{t_{n*},t_N}(f)}{b_{t_{n*},t_N}(1)}\right| = \left|\sum_{\substack{t\in\mathbb{T},\\ t_{n*}<t\le t_N}} \frac{b_{t,t_N}(f)}{b_{t,t_N}(1)} - \frac{b_{t-,t_N}(f)}{b_{t-,t_N}(1)}\right| \le \sum_{\substack{t\in\mathbb{T},\\ t_{n*}<t\le t_N}} \left|\frac{b_{t,t_N}(f)}{b_{t,t_N}(1)} - \frac{b_{t-,t_N}(f)}{b_{t-,t_N}(1)}\right|. \tag{2.32}$$

Note that

$$\left|\frac{b_{t,t_N}(f)}{b_{t,t_N}(1)} - \frac{b_{t-,t_N}(f)}{b_{t-,t_N}(1)}\right| \le \left|\frac{b_{t,t_N}(f)}{b_{t,t_N}(1)} - \frac{\mathbb{E}\left[b_{t,t_N}(f)\middle|\mathcal{B}^P_{t,J}\right]}{\mathbb{E}\left[b_{t,t_N}(1)\middle|\mathcal{B}^P_{t,J}\right]}\right| + \left|\frac{\mathbb{E}\left[b_{t,t_N}(f)\middle|\mathcal{B}^P_{t,J}\right]}{\mathbb{E}\left[b_{t,t_N}(1)\middle|\mathcal{B}^P_{t,J}\right]} - \frac{b_{t-,t_N}(f)}{b_{t-,t_N}(1)}\right|. \tag{2.33}$$

We will first consider the difference between $b_{t,t_N}(f)$ and $\mathbb{E}\left[b_{t,t_N}(f)\middle|\mathcal{B}^P_{t,J}\right]$ in the first term of the right hand side of (2.33) for $t \in (t_{n*}, t_N] \cap \mathbb{T}$. This corresponds to the error introduced by resampling at time t. We observe that

$$\begin{aligned}\operatorname{Var}\left(b_{t,t_N}(f)\middle|\mathcal{B}^P_{t,J}\right) &= \operatorname{Var}\left(\frac{1}{J}\sum_{j=1}^{J}\frac{Q_{t,t_N}(g_N\cdot f)}{u_t}\left(X_t^{F,j}\right)\middle|\mathcal{B}^P_{t,J}\right)\\ &= \frac{1}{J}\operatorname{Var}\left(\frac{Q_{t,t_N}(g_N\cdot f)}{u_t}\left(X_t^{F,1}\right)\middle|\mathcal{B}^P_{t,J}\right)\\ &\le \frac{1}{J}\mathbb{E}\left[\left(\frac{Q_{t,t_N}(g_N\cdot f)}{u_t}\left(X_t^{F,1}\right)\right)^2\middle|\mathcal{B}^P_{t,J}\right]\\ &\le \frac{1}{J}\max_j\left\{\frac{Q_{t,t_N}(g_N)}{u_t}\left(X_t^{P,j}\right)\right\}^2,\end{aligned}$$

because $\|f\|_\infty \le 1$ and $X_t^{F,1}$ takes one of J values $X_t^{P,j}$, $j = 1,\dots,J$. This implies that, by Markov's inequality, for any $a > 1$,

$$\left(b_{t,t_N}(f) - \mathbb{E}\left[b_{t,t_N}(f)\middle|\mathcal{B}^P_{t,J}\right]\right)^2 \le \frac{a^2}{J}\max_j\left\{\frac{Q_{t,t_N}(g_N)}{u_t}\left(X_t^{P,j}\right)\right\}^2 \tag{2.34}$$

with probability at least $1-\frac{1}{a^2}$. This also implies that we have

$$\left(b_{t,t_N}(1) - \mathbb{E}\left[b_{t,t_N}(1)\middle|\mathcal{B}^P_{t,J}\right]\right)^2 \le \frac{a^2}{J}\max_j\left\{\frac{Q_{t,t_N}(g_N)}{u_t}\left(X_t^{P,j}\right)\right\}^2 \tag{2.35}$$

with probability at least $1-\frac{1}{a^2}$. Write the first term on the right hand side of (2.33) as

$$\left|\frac{b_{t,t_N}(f)}{b_{t,t_N}(1)} - \frac{\mathbb{E}\left[b_{t,t_N}(f)\middle|\mathcal{B}^P_{t,J}\right]}{\mathbb{E}\left[b_{t,t_N}(1)\middle|\mathcal{B}^P_{t,J}\right]}\right| = \left|\frac{A'}{A}-\frac{B'}{B}\right|,$$

where $A, B > 0$, which is bounded by

(2.36)
$$\left|\frac{A'}{A}-\frac{B'}{B}\right| = \left|\frac{A'B - B'B + B'B - AB'}{AB}\right| \le \frac{|A'-B'|}{A} + \frac{|A-B|}{A}\cdot\frac{|B'|}{B} \le \frac{|A-B| + |A'-B'|}{A},$$

because we have $|B'| \le B$ from $\|f\|_\infty \le 1$. Note that

$$A = b_{t,t_N}(1) = \int \frac{Q_{t,t_N}(g_N)}{u_t}dF^F_{t,J} \ge \min_j\left\{\frac{Q_{t,t_N}(g_N)}{u_t}(X_t^{P,j})\right\}.$$

But by Assumption 1

$$\min_j\left\{\frac{Q_{t,t_N}(g_N)}{u_t}(X_t^{P,j})\right\} \ge \frac{1}{C_1}\max_j\left\{\frac{Q_{t,t_N}(g_N)}{u_t}(X_t^{P,j})\right\}$$

if $t > t_{N-B}$ so that $t^{\rightarrow} = t_N$, and by Assumption 1 and 3,

(2.37)

$$\begin{aligned}\min_j \left\{ \frac{Q_{t,t_N}(g_N)}{u_t} \left(X_t^{P,j}\right) \right\} &\geq \min_j \left\{ \frac{Q_{t,t_N}\left(g_N\right)}{Q_{t,t^{\rightarrow}}\left(g_{n(t^{\rightarrow})}\right)} \left(X_t^{P,j}\right) \right\} \cdot \min_j \left\{ \frac{Q_{t,t^{\rightarrow}}\left(g_{n(t^{\rightarrow})}\right)}{u_t} \left(X_t^{P,j}\right) \right\} \\ &\geq \frac{1}{C_3} \max_j \left\{ \frac{Q_{t,t_N}(g_N)}{Q_{t,t^{\rightarrow}}\left(g_{n(t^{\rightarrow})}\right)} \left(X_t^{P,j}\right) \right\} \cdot \frac{1}{C_1} \max_j \left\{ \frac{Q_{t,t^{\rightarrow}}\left(g_{n(t^{\rightarrow})}\right)}{u_t} \left(X_t^{P,j}\right) \right\} \\ &\geq \frac{1}{C_1 C_3} \max_j \left\{ \frac{Q_{t,t_N}(g_N)}{u_t} \left(X_t^{P,j}\right) \right\}\end{aligned}$$

if $t \leq t_{N-B}$ so that $t^{\rightarrow} < t_N$. Thus from (2.34), (2.35), and (2.36), we have

$$\left| \frac{b_{t,t_N}(f)}{b_{t,t_N}(1)} - \frac{\mathbb{E}\left[b_{t,t_N}(f) \middle| \mathcal{B}_{t,J}^P\right]}{\mathbb{E}\left[b_{t,t_N}(1) \middle| \mathcal{B}_{t,J}^P\right]} \right| \leq \frac{2C_1 a}{\sqrt{J}} \cdot C_3^{\mathbb{1}[t \leq t_{N-B}]} \tag{2.38}$$

with probability at least $1 - \frac{2}{a^2}$. Here, $\mathbb{1}[\,\cdot\,]$ denotes an indicator function.

Next, we consider the second term in (2.33)

(2.39)

$$\left| \frac{\mathbb{E}\left[b_{t,t_N}(f) \middle| \mathcal{B}_{t,J}^P\right]}{\mathbb{E}\left[b_{t,t_N}(1) \middle| \mathcal{B}_{t,J}^P\right]} - \frac{b_{t^-,t_N}(f)}{b_{t^-,t_N}(1)} \right| = \left| \frac{\displaystyle\int \frac{Q_{t,t_N}(g_N \cdot f)}{u_t} \cdot w_t dH_{t,J}}{\displaystyle\int \frac{Q_{t,t_N}(g_N)}{u_t} \cdot w_t dH_{t,J}} - \frac{\displaystyle\int \frac{Q_{t^-,t_N}(g_N \cdot f)}{u_{t^-}} dF_{t^-,J}^F}{\displaystyle\int \frac{Q_{t^-,t_N}(g_N)}{u^{t^-}} dF_{t^-,J}^F} \right|.$$

We have

$$\int \frac{Q_{t,t_N}(g_N \cdot f)(x_t)}{u_t(x_t)} \cdot w_t(x_t, x_{t^-}) dH_{t,J}(x_t, x_{t^-}) = \int \frac{Q_{t,t_N}(g_N \cdot f)(x_t)}{u_{t^-}/g_{n(t^-)}^{\mathbb{1}[t^- \in t_{1:N}]}(x_{t^-})} dH_{t,J}(x_t, x_{t^-})$$

from (2.16). One can write from (2.12) and (2.13)

$$Q_{t^-,t_N}(g_N \cdot f) = g_{n(t^-)}^{\mathbb{1}[t^- \in t_{1:N}]} K_{t^-,t} Q_{t,t_N}(g_N \cdot f),$$

so we have

$$\mathbb{E}\left[\frac{Q_{t,t_N}\left(g_N \cdot f\right)\left(X_t^{P,j}\right)}{\left\{u_{t^-}/g_{n(t^-)}^{\mathbb{1}[t^- \in t_{1:N}]}\right\}\left(X_{t^-}^{F,j}\right)} \middle| \mathcal{B}_{t^-,J}^F \right] = \frac{K_{t^-,t} Q_{t,t_N}(g_N \cdot f)\left(X_{t^-}^{F,j}\right)}{\left\{u_{t^-}/g_{n(t^-)}^{\mathbb{1}[t^- \in t_{1:N}]}\right\}\left(X_{t^-}^{F,j}\right)} = \frac{Q_{t^-,t_N}(g_N \cdot f)}{u_{t^-}}\left(X_{t^-}^F\right).$$

Hence,

$$
\begin{aligned}
&\mathbb{E}\left[\left\{\int \frac{Q_{t,t_N}(g_N\cdot f)}{u_{t^-}/g_{n(t^-)}^{\mathbb{1}[t^-\in t_{1:N}]}}dH_{t,J} - \int \frac{Q_{t^-,t_N}(g_N\cdot f)}{u_{t^-}}dF^F_{t^-,J}\right\}^2 \middle| \mathcal{B}^F_{t^-,J}\right] \\
&= \operatorname{Var}\left(\frac{1}{J}\sum_j \frac{Q_{t,t_N}(g_N\cdot f)\left(X_t^{P,j}\right)}{\left\{u_{t^-}/g_{n(t^-)}^{\mathbb{1}[t^-\in t_{1:N}]}\right\}\left(X_{t^-}^{F,j}\right)} \middle| \mathcal{B}^F_{t^-,J}\right) \\
&= \frac{1}{J^2}\sum_j \frac{\operatorname{Var}\left[Q_{t,t_N}(g_N\cdot f)\left(X_t^{P,j}\right)\middle|\mathcal{B}^F_{t^-,J}\right]}{\left\{K_{t^-,t}Q_{t,t_N}(g_N)\left(X_{t^-}^{F,j}\right)\right\}^2}\cdot\frac{\left\{K_{t^-,t}Q_{t,t_N}(g_N)\left(X_{t^-}^{F,j}\right)\right\}^2}{\left\{u_{t^-}/g_{n(t^-)}^{\mathbb{1}[t^-\in t_{1:N}]}\left(X_{t^-}^{F,j}\right)\right\}^2} \\
&\leq \frac{1}{J^2}\sum_j \frac{K_{t^-,t}\left\{Q_{t,t_N}\left(g_N\cdot|f|\right)\right\}^2}{\left\{K_{t^-,t}Q_{t,t_N}(g_N)\right\}^2}\left(X_{t^-}^{F,j}\right)\cdot\left\{\frac{Q_{t^-,t_N}(g_N)}{u_{t^-}}\left(X_{t^-}^{F,j}\right)\right\}^2 \\
&\leq \frac{1}{J}C_2^2\max_j\left\{\frac{Q_{t^-,t_N}(g_N)}{u_t}(X_{t^-}^{F,j})\right\}^2
\end{aligned}
$$

where the last inequality is due to Assumption 2. By Markov's inequality, for any $a > 1$,

$$\left|\int \frac{Q_{t,t_N}(g_N\cdot f)}{u_t}\cdot w_t dH_{t,J} - b_{t^-,t_N}(f)\right| \leq \frac{aC_2}{\sqrt{J}}\max_j\left\{\frac{Q_{t^-,t_N}(g_N)}{u_{t^-}}\left(X_{t^-}^{F,j}\right)\right\}$$

with probability at least $1-\frac{1}{a^2}$. Using the inequality (2.36) again, we obtain

$$\left|\frac{\mathbb{E}\left[b_{t,t_N}(f)\middle|\mathcal{B}^P_{t,J}\right]}{\mathbb{E}\left[b_{t,t_N}(1)\middle|\mathcal{B}^P_{t,J}\right]} - \frac{b_{t^-,t_N}(f)}{b_{t^-,t_N}(1)}\right| \leq 2\frac{aC_2}{\sqrt{J}}\frac{\max_j\left\{\frac{Q_{t^-,t_N}(g_N)}{u_{t^-}}\left(X_{t^-}^{F,j}\right)\right\}}{b_{t^-,t_N}(1)} \tag{2.40}$$

with probability at least $1-\frac{2}{a^2}$. By the same reasoning as in (2.37), we have

$$\frac{\max_j\left\{\frac{Q_{t^-,t_N}(g_N)}{u_{t^-}}\left(X_{t^-}^{F,j}\right)\right\}}{b_{t^-,t_N}(1)} \leq C_1C_3^{\mathbb{1}[t\leq t_{N-B,1}]}.$$

Thus, summing (2.38) and (2.40), we obtain

$$\left|\frac{b_{t,t_N}(f)}{b_{t,t_N}(1)} - \frac{b_{t^-,t_N}(f)}{b_{t^-,t_N}(1)}\right| \leq \frac{2aC_1}{\sqrt{J}}\left(C_2C_3^{\mathbb{1}[t\leq t_{N-B,1}]} + C_3^{\mathbb{1}[t\leq t_{N-B}]}\right)$$

with probability at least $1-\frac{4}{a^2}$. If we add the above inequality for $t\in(t_{n_*},t_N]\cap\mathbb{T}$, we reach the conclusion that

$$\left|b_{t_N,t_N}(f) - \frac{b_{t_{n_*},t_N}(f)}{b_{t_{n_*},t_N}(1)}\right| \leq \frac{2aC_1}{\sqrt{J}}\Big[C_2\big\{\big(S(N-B-n_*)+1\big)C_3+(SB-1)\big\}+S(N-B-n_*)C_3+SB\Big] \tag{2.41}$$

with probability at least $1 - \frac{4S(N-n_*)}{a^2}$. Using the fact $C_3 \geq 1$, the RHS in (2.41) can be replaced by a slightly larger but simpler bound

$$\frac{2aC_1}{\sqrt{J}}(C_2+1)\big[\,\{S(N-B-n_*)+1\}\,C_3 + (SB-1)\big].$$

Combining (2.30) and (2.41), we have

$$\left|\int f dF^F_{t_N,J} - \frac{\mu_{t_0}Q_{t_0,t_N}(g_N \cdot f)}{\mu_{t_0}Q_{t_0,t_N}(g_N)}\right| \leq C_4 + \frac{2aC_1}{\sqrt{J}}(C_2+1)\big[\,\{S(N-B-n_*)+1\}\,C_3 + (SB-1)\big]$$

with probability at least $1 - \frac{4S(N-n_*)}{a^2}$.

2.D The constant C_2 in Assumption 2 can be $O(1)$ in d.

We show in the first example that for a process consisting of d independent Brownian motions with noisy measurements, the number C_2 in Assumption 2 can be of constant magnitude in space dimension d. The second example shows that independence between dimensions is not necessary for C_2 to be of constant order.

Example 1. Consider d independent and identically distributed one dimensional Brownian motions. For simplicity of argument, let the number of future observations used for particle assessment be $B = 1$. The general case involves more complicated equations but follows the same logic. Let σ_p^2 be the variance of the one dimensional Brownian motion over the unit interval, and let σ_m^2 be the variance of measurement error.

$$X_{t+\delta} = X_t + \sigma_p\sqrt{\delta}\cdot N(0, I_{d\times d}), \text{ for } \delta > 0,$$

$$Y_n = X_{t_n} + \sigma_m \cdot N(0, I_{d\times d}).$$

Let the density of the one dimensional Gaussian distribution with mean μ and variance σ^2 at point x be denoted by $\phi(x\,;\mu,\sigma^2)$. Since $B = 1$, we have

$$Q_{t_{n,s},t_{n+1}}(g_{n+1}\cdot f) = K_{t_{n,s},t_{n+1}}(g_{n+1}\cdot f), \quad s \in 1:S.$$

By the independence assumption of the d Brownian motions, the transition kernel is given by a product measure

$$\begin{aligned}K_{t_{n,s},t_{n+1}}&\left(dx^1_{t_{n+1}}\cdots dx^d_{t_{n+1}}\,;x^1_{t_{n,s}},\cdots,x^d_{t_{n,s}}\right)\\&=k_{t_{n,s},t_{n+1}}\left(dx^1_{t_{n+1}}\,;x^1_{t_{n,s}}\right)\otimes\cdots\otimes k_{t_{n,s},t_{n+1}}\left(dx^d_{t_{n+1}}\,;x^d_{t_{n,s}}\right)\end{aligned}$$

where k denotes the transition kernel for each one dimensional component of $\{X_t\}$. We assume that $t_{n+1}-t_n=1$ and that $t_{n,s}-t_{n,s-1}=\frac{1}{S}$, $s\in 1\colon S$. Then for any bounded measurable $f_0\colon \mathbb{R}\to\mathbb{R}$,

$$k_{t_{n,s},t_{n+1}}f_0\left(x_{t_{n,s}}\right)=\int_{-\infty}^{\infty}f_0\left(x_{t_{n+1}}\right)\phi\left[x_{t_{n+1}}\,;x_{t_{n,s}},\left(1-\frac{s}{S}\right)\sigma_p^2\right]dx_{t_{n+1}}$$

Also, independent measurements implies that

$$g_{n+1}\left(x^1_{t_{n+1}},\cdots,x^d_{t_{n+1}}\right)=\phi\left(y^1_{n+1}\,;x^1_{t_{n+1}},\sigma_m^2\right)\cdot\cdots\cdot\phi\left(y^d_{n+1}\,;x^d_{t_{n+1}},\sigma_m^2\right).$$

Since both the state process and the measurement process are Gaussian, we have for $i\in 1\colon d$,

$$\int\phi\left(y^i_{n+1}\,;x^i_{t_{n+1}},\sigma_m^2\right)k_{t_{n,s},t_{n+1}}\left(dx^i_{t_{n+1}}\,;x^i_{t_{n,s}}\right)=\phi\left[y^i_{n+1}\,;x^i_{t_{n,s}},\sigma_m^2+\left(1-\frac{s}{S}\right)\sigma_p^2\right].$$

It follows that

$$Q_{t_{n,s},t_{n+1}}(g_{n+1})\left(x_{t_{n,s}}\right)=\prod_{i=1}^{d}\phi\left[y^i_{n+1}\,;x^i_{t_{n,s}},\sigma_m^2+\left(1-\frac{s}{S}\right)\sigma_p^2\right].$$

We see

$$\frac{Q_{t_{n,s},t_{n+1}}(g_{n+1})\left(X_{t_{n,s}}\right)}{K_{t_{n,s-1},t_{n,s}}Q_{t_{n,s},t_{n+1}}(g_{n+1})\left(X_{t_{n,s-1}}\right)}=\prod_{i=1}^{d}\frac{\phi\left[y^i_{n+1}\,;X^i_{t_{n,s}},\sigma_m^2+\left(1-\frac{s}{S}\right)\sigma_p^2\right]}{\phi\left[y^i_{n+1}\,;X^i_{t_{n,s-1}},\sigma_m^2+\left(1-\frac{s-1}{S}\right)\sigma_p^2\right]}.$$

We observe that

$$\prod_{i=1}^{d} \frac{\phi\left[y_{n+1}^i\,;X_{t_{n,s}}^i,\sigma_m^2+\left(1-\dfrac{s}{S}\right)\sigma_p^2\right]}{\phi\left[y_{n+1}^i\,;X_{t_{n,s-1}}^i,\sigma_m^2+\left(1-\dfrac{s-1}{S}\right)\sigma_p^2\right]}$$

$$= \frac{\left\{\sigma_m^2+\left(1-\frac{s-1}{S}\right)\sigma_p^2\right\}^{d/2}}{\left\{\sigma_m^2+\left(1-\frac{s}{S}\right)\sigma_p^2\right\}^{d/2}} \frac{\exp\left\{-\dfrac{\|y_{n+1}-X_{t_{n,s}}\|^2}{2\left\{\sigma_m^2+\left(1-\frac{s}{S}\right)\sigma_p^2\right\}}\right\}}{\exp\left\{-\dfrac{\|y_{n+1}-X_{t_{n,s-1}}\|^2}{2\left\{\sigma_m^2+\left(1-\frac{s-1}{S}\right)\sigma_p^2\right\}}\right\}},$$

where $\|\cdot\|$ denotes the usual Euclidean distance. Let $\Delta := X_{t_{n,s}} - X_{t_{n,s-1}}$, and the above expression equals

(2.42)

$$\left\{1+\frac{\frac{1}{S}\sigma_p^2}{\sigma_m^2+\left(1-\frac{s}{S}\right)\sigma_p^2}\right\}^{\frac{d}{2}}$$

$$\cdot \exp\left\{-\frac{\left[2\Delta\cdot(X_{t_{n,s-1}}-y_{n+1})+\|\Delta\|^2\right]\cdot\left[\sigma_m^2+\left(1-\frac{s-1}{S}\right)\sigma_p^2\right]+\frac{\sigma_p^2}{S}\|y_{n+1}-X_{t_{n,s-1}}\|^2}{2\left[\sigma_m^2+\left(1-\frac{s}{S}\right)\sigma_p^2\right]\cdot\left[\sigma_m^2+\left(1-\frac{s-1}{S}\right)\sigma_p^2\right]}\right\}.$$

Note that conditional on $X_{t_{n,s-1}}$, due to independence between dimensions

$$\Delta\cdot\left(X_{t_{n,s-1}}-y_{n+1}\right) \overset{d}{=} N\left[0,\,\frac{\sigma_p^2}{S}\|X_{t_{n,s-1}}-y_{n+1}\|^2\right]$$

and

$$\|\Delta\|^2 \overset{d}{=} \frac{\sigma_p^2}{S}\chi_d^2 \approx \frac{d}{S}\sigma_p^2+\frac{\sqrt{d}}{S}\sigma_p^2 N(0,2),$$

where the last approximation is due to the central limit theorem. Suppose we set

$S = d$. Then as $d \to \infty$, we can calculate that

$$\begin{aligned}
&\frac{\mathbb{E}\left[\left\{Q_{t_{n,s},t_{n+1}}(g_{n+1})\left(X_{t_{n,s}}\right)\right\}^2\middle| X_{t_{n,s-1}}\right]}{\left\{K_{t_{n,s-1},t_{n,s}}Q_{t_{n,s},t_{n+1}}(g_{n+1})\left(X_{t_{n,s-1}}\right)\right\}^2} \\
&= \mathbb{E}\Bigg[O(1)\cdot\exp\Bigg\{-\frac{2\Delta\cdot\left(X_{t_{n,s-1}}-y_{n+1}\right)}{\sigma_m^2+\left(1-\frac{s}{S}\right)\sigma_p^2} \\
&\qquad\qquad + \frac{\frac{\sigma_p^2}{S}\|y_{n+1}-X_{t_{n,s-1}}\|^2}{\left\{\sigma_m^2+\left(1-\frac{s}{S}\right)\sigma_p^2\right\}\left\{\sigma_m^2+\left(1-\frac{s-1}{S}\right)\sigma_p^2\right\}}\Bigg\}\Bigg| X_{t_{n,s-1}}\Bigg] \\
&= O(1)\cdot\exp\left\{2\frac{\frac{\sigma_p^2}{S}\|y_{n+1}-X_{t_{n,s-1}}\|^2}{\left\{\sigma_m^2+\left(1-\frac{s}{S}\right)\sigma_p^2\right\}^2} + \frac{\frac{\sigma_p^2}{S}\|y_{n+1}-X_{t_{n,s-1}}\|^2}{\left\{\sigma_m^2+\left(1-\frac{s}{S}\right)\sigma_p^2\right\}\left\{\sigma_m^2+\left(1-\frac{s-1}{S}\right)\sigma_p^2\right\}}\right\} \\
&= O(1),
\end{aligned}$$

because $\frac{1}{S}\|y_{n+1}-X_{t_{n,s-1}}\|^2 = O(1)$. Thus we conclude that

$$\frac{K_{t_{n,s-1},t_{n,s}}\left\{Q_{t_{n,s},t_{n+1}}(g_{n+1})\right\}^2}{\left\{K_{t_{n,s-1},t_{n,s}}Q_{t_{n,s},t_{n+1}}(g_{n+1})\right\}^2}\left(X_{t_{n,s-1}}\right) = O(1).$$

Therefore the constant C_2 in Assumption 2 is $O(1)$ under the limit $S = d \to \infty$.

Example 2. Consider d Brownian motions that are perfectly correlated. That is,

$$X^1 = X^2 = \cdots = X^d,$$

where

$$X^1_{t+\delta} = X^1_t + \sigma_p\sqrt{\delta}\cdot N(0,1), \quad \text{for } \delta > 0.$$

The measurement model stays the same as in the previous example.

$$Y_n = X_{t_n} + \sigma_m \cdot N(0, I_{d\times d}).$$

This POMP model is equivalent to a one dimensional Brownian motion where d independent observations are made, and we consider the process $\{X_t\}$ this way. The density of the measurement model is given by

$$g_{n+1}(x) = \frac{1}{\sqrt{2\pi}^d\sigma_m^d}e^{-\sum_{i=1}^d (y^i_{n+1}-x)^2/2\sigma_m^2}.$$

Setting $B = 1$ again for simplicity of argument, we compute that

$$\begin{aligned} &Q_{t_{n,s},t_{n+1}} g_{n+1}\left(x_{t_{n,s}}\right) \\ &= \frac{1}{\sqrt{2\pi}^{d+1}\sigma_m^d}\frac{1}{\sqrt{\left(1-\frac{s}{S}\right)\sigma_p^2}}\int_{\mathbb{R}} \exp\left\{-\frac{\sum_i \left(y_{n+1}^i - x\right)^2}{2\sigma_m^2}\right\}\cdot\exp\left\{-\frac{\left(x-x_{t_{n,s}}\right)^2}{2\left(1-\frac{s}{S}\right)\sigma_p^2}\right\}dx \\ &= \frac{1}{\sqrt{2\pi}^d}\frac{1}{\sigma_m^d\sigma_p\sqrt{\left(1-\frac{s}{S}\right)}}\sqrt{\frac{(\sigma_m^2/d)\cdot\left(1-\frac{s}{S}\right)\sigma_p^2}{\sigma_m^2/d+\left(1-\frac{s}{S}\right)\sigma_p^2}}\exp\left\{-\frac{\left(x_{t_{n,s}}-\bar{y}_{n+1}\right)^2}{2\left\{\sigma_m^2/d+\left(1-\frac{s}{S}\right)\sigma_p^2\right\}}\right\} \\ &\qquad\qquad\cdot\exp\left\{\frac{d\bar{y}_{n+1}^2-\|y_{n+1}\|^2}{2\sigma_m^2}\right\}, \end{aligned}$$

where $\bar{y}_{n+1} = \frac{1}{d}\sum_{i=1}^{d} y_{n+1}^i$. It follows that

$$\begin{aligned} &\frac{Q_{t_{n,s},t_{n+1}}(g_{n+1})\left(X_{t_{n,s}}\right)}{K_{t_{n,s-1},t_{n,s}}Q_{t_{n,s},t_{n+1}}(g_{n+1})\left(X_{t_{n,s-1}}\right)} \\ &= \sqrt{\frac{\sigma_m^2/d+\left(1-\frac{s-1}{S}\right)\sigma_p^2}{\sigma_m^2/d+\left(1-\frac{s}{S}\right)\sigma_p^2}}\exp\left\{-\frac{\left(X_{t_{n,s}}-\bar{y}_{n+1}\right)^2}{2\left\{\sigma_m^2/d+\left(1-\frac{s}{S}\right)\sigma_p^2\right\}}+\frac{\left(X_{t_{n,s-1}}-\bar{y}_{n+1}\right)^2}{2\left\{\sigma_m^2/d+\left(1-\frac{s-1}{S}\right)\sigma_p^2\right\}}\right\}. \end{aligned}$$

Hence, if we write $\Delta := X_{t_{n,s}} - X_{t_{n,s-1}}$,

(2.43)

$$\begin{aligned} &\mathbb{E}\left[\left\{\frac{Q_{t_{n,s},t_{n+1}}(g_{n+1})\left(X_{t_{n,s}}\right)}{K_{t_{n,s-1},t_{n,s}}Q_{t_{n,s},t_{n+1}}(g_{n+1})\left(X_{t_{n,s-1}}\right)}\right\}^2\middle| X_{t_{n,s-1}}\right] \\ &\qquad= \left(1+\frac{\frac{1}{S}\sigma_p^2}{\sigma_m^2/d+\left(1-\frac{s}{S}\right)\sigma_p^2}\right)\mathbb{E}\left[\exp\left\{-\frac{\left(X_{t_{n,s-1}}-\bar{y}_{n+1}+\Delta\right)^2}{\sigma_m^2/d+\left(1-\frac{s}{S}\right)\sigma_p^2}\right\}\middle| X_{t_{n,s-1}}\right] \\ &\qquad\qquad\qquad\cdot\exp\left\{\frac{\left(X_{t_{n,s-1}}-\bar{y}_{n+1}\right)^2}{\sigma_m^2/d+\left(1-\frac{s-1}{S}\right)\sigma_p^2}\right\} \\ &\qquad= \left(1+\frac{\frac{1}{S}\sigma_p^2}{\sigma_m^2/d+\left(1-\frac{s}{S}\right)\sigma_p^2}\right)\sqrt{1+\frac{\frac{2}{S}\sigma_p^2}{\sigma_m^2/d+\left(1-\frac{s}{S}\right)\sigma_p^2}}^{-1} \\ &\qquad\qquad\cdot\exp\left\{-\frac{\left(X_{t_{n,s-1}}-\bar{y}_{n+1}\right)^2}{\sigma_m^2/d+\left(1-\frac{s}{S}+\frac{2}{S}\right)\sigma_p^2}\right\}\cdot\exp\left\{\frac{\left(X_{t_{n,s-1}}-\bar{y}_{n+1}\right)^2}{\sigma_m^2/d+\left(1-\frac{s-1}{S}\right)\sigma_p^2}\right\} \\ &\qquad= \frac{\sigma_m^2/d+\left(1-\frac{s-1}{S}\right)\sigma_p^2}{\sqrt{\left\{\sigma_m^2/d+\left(1-\frac{s}{S}\right)\sigma_p^2\right\}\left\{\sigma_m^2/d+\left(1-\frac{s-2}{S}\right)\sigma_p^2\right\}}} \\ &\qquad\qquad\qquad\cdot\exp\left\{\frac{\frac{1}{S}\sigma_p^2\left(X_{t_{n,s-1}}-\bar{y}_{n+1}\right)^2}{\left\{\sigma_m^2/d+\left(1-\frac{s-2}{S}\right)\sigma_p^2\right\}\left\{\sigma_m^2/d+\left(1-\frac{s-1}{S}\right)\sigma_p^2\right\}}\right\}, \end{aligned}$$

where we have used the formula for the moment generating function of the non-central chi-square variable $(X_{t_{n,s-1}} - \bar{y}_{n+1} + \Delta)^2$. When $S = d$,

$$\frac{\left(X_{t_{n,s}} - \bar{y}_{n+1}\right)^2}{\sigma_m^2/d + \left(1 - \frac{s-1}{S}\right)\sigma_p^2} \sim \chi_1^2,$$

and hence (2.43) is $O(1)$. It follows that $C_2 = O(1)$.

CHAPTER III

Analysis of spatiotemporal measles transmission dynamics using a guided intermediate resampling filter

In this chapter, as an illustration of the fact that the guided intermediate resampling filter method presented in the previous chapter enables likelihood based inference on coupled dynamic systems of scientific interest, I present the data analysis results of spatiotemporal epidemic data for measles collected at linked geographical locations. The goal is to make inference on the transmission dynamics of measles, and in particular, to estimate the mode and strength of spatial coupling between the transmission dynamics in the geographic locations.

3.1 Coupled spatiotemporal measles transmission model

Population dynamics of infectious diseases exhibit highly nonlinear stochastic behavior. Compared to other diseases, the epidemic dynamics of measles is well understood and is characterized by patterns that are closely replicable using a mechanistic model. We adopted the model developed by He et al. [2009], but added spatial interaction between multiple cities. Using the approach described in Section 2.5, we made inference on the spatial coupling parameter, which could only be correctly estimated when the filter recovered the full joint distribution.

The model compartmentalize the population of each city into susceptible (S), exposed (E), infectious (I), and recovered/removed (R) categories. Their sizes for the k-city are denoted by S_k, E_k, I_k, and R_k. The population dynamics can be described on average by the following set of differential equations:

$$\begin{aligned} \mathbb{E}dS_k(t) &= r_k(t)dt - \mathbb{E}dN_{SE,k}(t) - \mu S_k(t)dt \\ \mathbb{E}dE_k(t) &= \mathbb{E}dN_{SE,k}(t) - \mathbb{E}dN_{EI,k}(t) - \mu E_k(t)dt \qquad\qquad k = 1, \cdots, d. \\ \mathbb{E}dI_k(t) &= \mathbb{E}dN_{EI,k}(t) - \mathbb{E}dN_{IR,k}(t) - \mu I_k(t)dt \end{aligned}$$

Here, $N_{SE,k}(t)$, $N_{EI,k}(t)$, and $N_{IR,k}(t)$ denote the cumulative number of transitions between the corresponding compartments up to time t in city k, μ denotes per-capita mortality rate, and r_k the recruitment rate of susceptible population. The term $N_{SE,k}(t)$, representing the cumulative number of infections in the k-th city, has the expected increment of

$$\begin{aligned} &\mathbb{E}\left[N_{SE,k}(t+dt) - N_{SE,k}(t)\right] \\ &\quad = \beta(t) \cdot S_k(t) \cdot \left[\left(\frac{I_k}{P_k}\right)^{\alpha} + \sum_{l \neq k} \frac{v_{kl}}{P_k}\left\{\left(\frac{I_l}{P_l}\right)^{\alpha} - \left(\frac{I_k}{P_k}\right)^{\alpha}\right\}\right] dt + o(dt), \end{aligned} \tag{3.1}$$

where the population of city k was denoted by P_k and the number of travelers from city k to l by v_{kl}. The expected increment of transitions from the exposed to the infectious N_{EI} and from the infectious to the recovered compartments N_{IR} are modeled as

$$\mathbb{E}\left[N_{EI}(t+dt) - N_{EI}(t)\right] = \nu_{EI}E(t)dt + o(dt), \tag{3.2}$$

$$\mathbb{E}\left[N_{IR}(t+dt) - N_{IR}(t)\right] = \nu_{IR}I(t)dt + o(dt). \tag{3.3}$$

Here $\beta(t)$ denotes the seasonal transmission coefficient and α the mixing exponent, and ν_{EI} and ν_{IR} are the per capita progression rates between the respective compartments [He et al., 2009]. This model assumes that the spatial interaction of the measles transmission dynamics between the cities was mediated by infectious travelers moving from one city to another, where they can infect susceptible population in the destination city. We used a gravity model inspired by Xia et al. [2004] and described the number of travelers with the equation

$$v_{kl} = G \cdot \frac{\bar{d}}{\bar{P}^2} \cdot \frac{P_k \cdot P_l}{d_{kl}}, \tag{3.4}$$

where d_{kl} denotes the distance between city k and city l. The gravitation constant G in (3.4) is scaled with respect to the average population of all twenty cities $\bar{P}$ and their average distance $\bar{d}$. We assume the transmission coefficient $\beta(t)$ in (3.1) depends on whether it is school term or holiday, because most measles infections happen via transmissions between children:

$$\beta(t) = \begin{cases} \big(1 + 2(1-p)a\big)\bar{\beta} & \text{during school term} \\ \big(1 - 2p\,a\big)\bar{\beta} & \text{during school holiday.} \end{cases}$$

Here, $p = 0.739$ is the proportion of the year taken up by the school term, a the amplitude of variation, and $\bar{\beta}$ the annual average of the transmission rate. School holidays in the calendar day include: Christmas, 356–365 and 0–6; Easter, 100–115; summer, 199–252; autumn half-term, 300–308.

We add randomness to state progression by modeling the cumulative transitions from one compartment to the next on an infinitesimal time interval as multinomial random variables with random success probabilities that are distributed according to Gamma distributions, as described in Bretó et al. [2009]. This choice makes the processes continuous-time Markovian and allows for over-dispersion compared to Poisson processes [Bretó and Ionides, 2011]. For all cumulative transition processes $\{N_{\cdot\cdot}(t)\}$, we let the noise intensity to equal σ^2 [Bretó et al., 2009, Karlin and Taylor, 1981]. Over a short time interval $[t, t+\delta]$, the infinitesimal increment $N_{\cdot\cdot}(t+\delta) - N_{\cdot\cdot}(t)$

is Poisson distributed with the mean parameter given by the product of a gamma random variable $\mathrm{Gamma}(\delta/\sigma^2, \sigma^2)$ and the mean transition rate $\mathbb{E}\frac{dN_{\cdot\cdot}}{dt}$. In the limit as δ approaches zero, this amounts to

$$N_{\cdot\cdot}(t+\delta) - N_{\cdot\cdot}(t) \sim \mathrm{NegBin}\left(\frac{\delta}{\sigma^2}, \frac{\sigma^2 \cdot \mathbb{E}\frac{dN_{\cdot\cdot}(t)}{dt}}{\sigma^2 \cdot \mathbb{E}\frac{dN_{\cdot\cdot}(t)}{dt} + 1} \right), \tag{3.5}$$

where the negative binomial random variable $\mathrm{NegBin}(r, p)$ has the probability mass function

$$\mathbb{P}[\mathrm{NegBin}(r,p) = k] = \binom{k+r-1}{k} \cdot (1-p)^r p^k, \quad k = 0, 1, 2, \ldots$$

with mean $\frac{pr}{1-p}$ and variance $\frac{pr}{(1-p)^2}$. Bretó et al. [2009] explained a construction of such stochastic compartment models, which we adopt in our implementation.

The data consisted of the weekly reported case numbers in each city. The model assumed that a certain fraction ρ, called the reporting probability, of the transitions from the infectious compartment to the recovered compartment were, on average, counted as reported cases. The measurement model was chosen to allow for over-dispersion relative to the binomial distribution with success probability ρ. We used a discrete normal distribution with over-dispersion parameter ψ. Specifically, we define a cumulative distribution F depending on the number of weekly total transitions ΔN_{IR},

$$F(y\,; \rho, \psi, \Delta N_{IR}) := \Phi\left[y\,; \rho\Delta N_{IR},\, \rho(1-\rho)\Delta N_{IR} + \psi^2\rho^2\Delta N_{IR}^2 + 1 \right], \tag{3.6}$$

where $\Phi(\,\cdot\,; \mu, \sigma^2)$ is the cdf of the normal distribution with mean μ and variance σ^2. We then let the probability of having y_n reported cases in the n-th week, where ΔN_{IR} transitions from the infectious to the recovered compartment happened in that week, as

$$p(y_n) = F(y_n + 0.5\,; \rho, \psi, \Delta N_{IR}) - F(y_n - 0.5\,; \rho, \psi, \Delta N_{IR}).$$

Table 3.1: Table of model parameters and the values used to generate artificial data.

symbol	description	units	values
R_0	basic reproduction number	–	20
a	amplitude of seasonality	–	0.163
α	mixing exponent	–	0.97
μ	mortality rate	week^{-1}	3.2×10^{-4}
ν_{EI}^{-1}	latent period	week	1.0
ν_{IR}^{-1}	infectious period	week	1.0
σ^2	white-noise intensity	week	0.08
ρ	reporting probability	–	0.5
ψ	reporting overdispersion	–	0.25
G	gravitation constant	–	500
c	cohort entry fraction	–	0.4
τ	recruitment delay	year	4
$\frac{S_k(0)}{P_k(0)}$	initial susceptible proportion	–	0.4
$\frac{E_k(0)}{P_k(0)}$	initial exposed proportion	–	0.00027
$\frac{I_k(0)}{P_k(0)}$	initial infectious proportion	–	0.00032

The susceptible recruitment rate $r(t)$ was defined as follows. In the calendar year x, a certain fraction c of the annual births of the calendar year $x-4$ enters the susceptible compartment at the school admission date, which is the 251st day of a year. The remaining $1-c$ fraction enters the susceptible compartment continuously with a constant rate throughout the year.

3.2 The implementation of the GIRF

The state process $X(t)$ is composed of the components $S(t), E(t), I(t)$, and $N_{IR}^{week}(t)$ for each city, where $N_{IR}^{week}(t)$ is the weekly cumulative transitions from the infectious to the recovered compartment, with the relation $N_{IR}^{week}(t) = N_{IR}(t) - N_{IR}(t_n)$ where t_n is the start of the week that t is in. In the implementation of the GIRF on this model, we defined the guide function $u_{t_{n,s}}$ using the formula (2.8) in Section 2.3.3 with the approximate predictive likelihood $u_{t_{n,s}\nearrow t_{n+b}}$, which was defined as follows. We note that the procedure explained below provides a fairly general way of approximating the predictive likelihood.

First, we approximate the distribution of $X_{t_{n+b}}$ with a moment matching method. Suppose that the measurement Y_{n+b} is determined by a random variable $Z = z(X_{t_{n+b}})$, for some function z. In our example, this is the weekly cumulative infections, that is $z(X_{t_{n+b}}) := N_{IR}(t_{n+b}) - N_{IR}(t_{n+b-1})$. We make a projection from time $t_{n,s}$ to t_{n+b} with the deterministic mean process $\bar{X}(t)$, and take $z\big(\bar{X}(t_{n+b})\big)$ as an approximation to the conditional mean of $z\big(X(t_{n+b})\big)$ given $X_{t_{n,s}}$. The deterministic mean process may be obtained by setting the parameters governing the variability of the state process to zero. For implicit models, one can simulate the deterministic mean process with numerical procedures such as the Euler method.

We then estimate the variance of the projected Z. Let $\tau_0 = t_{n,s} < \tau_1 < \cdots < \tau_k = t_{n+b}$ be the time points at which the numerical simulation method computes the deterministic projection. We denote the σ-algebra containing all information about the state process up to time t as $\mathcal{F}_t$. We decompose the conditional variance of Z given X_{τ_0} in the following way.

$$\begin{aligned}
\mathrm{Var}(Z \,|\, \mathcal{F}_{\tau_0}) &= \mathbb{E}(Z^2 \,|\, \mathcal{F}_{\tau_0}) - \mathbb{E}(Z \,|\, \mathcal{F}_{\tau_0})^2 \\
&= \mathbb{E}\left[\mathbb{E}(Z \,|\, \mathcal{F}_{\tau_k})^2 \,|\, \mathcal{F}_{\tau_0}\right] - \mathbb{E}\left[\mathbb{E}(Z \,|\, \mathcal{F}_{\tau_0})^2 \,|\, \mathcal{F}_{\tau_0}\right] \\
&= \sum_{i=1}^{k} \mathbb{E}\left[\mathbb{E}(Z \,|\, \mathcal{F}_{\tau_i})^2 \,|\, \mathcal{F}_{\tau_0}\right] - \mathbb{E}\left[\mathbb{E}(Z \,|\, \mathcal{F}_{\tau_{i-1}})^2 \,|\, \mathcal{F}_{\tau_0}\right] \\
&= \sum_{i=1}^{k} \mathbb{E}\left[\mathrm{Var}\left\{\mathbb{E}(Z \,|\, \mathcal{F}_{\tau_i}) \,|\, \mathcal{F}_{\tau_{i-1}}\right\} \,|\, \mathcal{F}_{\tau_0}\right]
\end{aligned}$$

In other words, the variance of projection may be expressed as a sum of the expected values of the conditional variances over each sub-interval (τ_{i-1}, τ_i). The outermost conditional expectation with respect to $\mathcal{F}_{\tau_0}$ for each term is approximated by the value of $\mathrm{Var}\left\{\mathbb{E}(Z \,|\, \mathcal{F}_{\tau_i}) \,|\, \mathcal{F}_{\tau_{i-1}}\right\}$ where $X_{\tau_{i-1}}$ is at the mean projected value from time τ_0 to time τ_{i-1}. That is, if we call the mean projected states at time τ_{i-1} as

$\bar{X}_{\tau_{i-1}}$, we approximate the above expression with

$$\sum_{i=1}^{k} \mathrm{Var}\left\{\mathbb{E}(Z \mid \mathcal{F}_{\tau_i}) \,\middle|\, X_{\tau_{i-1}} = \bar{X}_{\tau_{i-1}}\right\}.$$

Thus we need an expression for $\mathbb{E}(Z \mid \mathcal{F}_{\tau_i})$ in terms of X_{τ_i} and a way of computing its conditional variance with respect to $\mathcal{F}_{\tau_{i-1}}$. The conditional mean is computed by locally linearizing the deterministic mean process. That is, if the dynamics of the mean process can be approximated as

$$d\bar{X}(t) = A\left\{\bar{X}(t)\right\}\bar{X}(t)dt,$$

for some matrix function A, we may approximate $\bar{X}(\tau_{i+1}) \approx e^{A\left\{\bar{X}(\tau_i)\right\}\cdot(\tau_{i+1}-\tau_i)}\bar{X}(\tau_i)$. Thus, we make the approximation

$$\mathbb{E}(Z \mid \mathcal{F}_{\tau_i}) \approx e^{\sum_{j=i}^{k-1} A\left\{\bar{X}(\tau_j)\right\}\cdot(\tau_{j+1}-\tau_j)}X(\tau_i).$$

Based on the above approximation, we compute the conditional variance of $\mathbb{E}(Z \mid \mathcal{F}_{\tau_i})$ with respect to $\mathcal{F}_{\tau_{i-1}}$ using the distributional properties of the transition kernel. In our example, $\mathbb{E}[Z \mid \mathcal{F}_{\tau_i}]$ may be approximated as a linear combination $c_0 + c_1 E(\tau_i) + c_2 I(\tau_i)$, where we have included the dependence on $S(\tau_i)$ in the constant term c_0 because the change in $S(t)$ can be assumed to be negligible over the time interval (τ_{i-1}, τ_i). Now, $I(\tau_i)$ is given by

$$I(\tau_i) = I(\tau_{i-1}) + N_{EI}(\tau_i) - N_{EI}(\tau_{i-1}) - N_{IR}(\tau_i) + N_{IR}(\tau_{i-1}),$$

and a similar expression can be obtained for $S(\tau_i)$. Thus the conditional variance of $c_1 E(\tau_i) + c_2 I(\tau_i)$ given $X(\tau_i)$ can be computed from the conditional variance of $N_{EI}(\tau_i)$ and that of $N_{IR}(\tau_{i+1})$ given $X(\tau_{i-1})$, with the simplifying assumption that $N_{EI}(\tau_i)$ and $N_{IR}(\tau_{i+1})$ are conditionally independent given $X(\tau_{i-1})$. The conditional variance of $N_{IR}(\tau_{i+1})$ can, for example, be approximated as

$$\nu_{IR} I(\tau_i) \cdot (\tau_{i+1} - \tau_i) \cdot \left\{\nu_{IR} I(\tau_i)\sigma^2 + 1\right\},$$

since the cumulative transitions N_{SE}, N_{EI}, and N_{IR} are modeled as locally negative binomial processes given by (3.5).

Once $\mathbb{E}(Z \,|\, \mathcal{F}_{t_{n,s}})$ and $\mathrm{Var}(Z \,|\, \mathcal{F}_{t_{n,s}})$ have been approximated, we approximate the distribution of Z given $X_{t_{n,s}}$ as the distribution that belongs to the same family of distributions as the local transitions, but with the mean and variance given as above. In our example, we take the negative binomial process with the computed mean and variance. If the measurement process also belongs to the same family of distributions, the variance of the projection and the variance of the measurement process may be added to give the approximated predictive likelihood model. In the cases where the projected state process distribution and the measurement process distribution have considerably different tail behaviors, one might approximate the predictive likelihood as a discretized convolution

$$\begin{aligned}\mathbb{P}\left(Y_{n+b} = y_{n+b} \,|\, X_{t_{n,s}}\right) &= \mathbb{E}\left[\mathbb{P}(Y_{n+b} = y_{n+b} \,|\, X_{t_{n+b}}) \,\middle|\, X_{t_{n,s}}\right] \\ &\approx \sum_{j=1}^{k} \mathbb{P}\left[Y_{n+b} = y_{n+b} \,\middle|\, X_{t_{n+b}} = x_j\right] \cdot \mathbb{P}\left[X_{t_{n+b}} = x_j \,\middle|\, X_{t_{n,s}}\right] \cdot (x_{j+1} - x_j)\end{aligned}$$

where x_j, $j = 1, \ldots, k$, may be taken to be the points at the sample space where the probability $\mathbb{P}\left[Y_{n+b} = y_{n+b} \,\middle|\, X_{t_{n+b}} = x_j\right] \cdot \mathbb{P}\left[X_{t_{n+b}} = x_j \,\middle|\, X_{t_{n,s}}\right]$ is non-negligible. The length of the last interval $x_{k+1} - x_k$ can be chosen appropriately to approximate the tail probability.

The above procedure is almost impossible to perform for high dimensional process $X(t)$. In that case, we divide the components of $X(t)$ into groups of highly correlated ones, and estimate the predictive likelihood for each group. A guide function may be computed as the product of these quantities. In our example, we treated each city as a separate group. This approximation may be considered as a kind of variational inference technique. However, even if we approximate the predictive likelihood for each city separately, we are still taking into account the spatial interaction between the cities, because the deterministic mean process is simulated under the influence of spatial coupling.

We note that the approximate predictive likelihood, as well as the guide function u_t, is a means of helping the GIRF guide the particles in the right direction. Although more accurate approximation to the predictive likelihoods leads to better performance of the filter, some degree of inaccuracy in the approximation may be handled by the filter, provided that sufficient number of particles are used.

Instead of applying the lengthy procedure described above, we also have used the simple approximation where the variance of projection was estimated as

$$\text{Var}\left[N_{IR}(t_{n+b}) - N_{IR}(t_{n+b-1})\right] \approx \int_{t_{n+b-1}}^{t_{n+b}} \left\{\bar{I}(t)\cdot\nu_{IR}\cdot\sigma^2 + 1\right\}\cdot\bar{I}(t)\cdot\nu_{IR}\,dt,$$

where $\bar{I}(t)$ denotes the size of the infectious compartment in the deterministic mean process $\bar{X}(t)$. The variance thus estimated was simply added to the variance of the discrete normal measurement model for the approximation of $\mathbb{P}(Y_{n+b} = y_{n+b} \,|\, X_{t_{n,s}})$. The performance of this simpler approximation was comparable to the more scrupulous approximation detailed above.

3.3 Monte Carlo adjusted profile confidence intervals

When the likelihood of data from a one-parameter model can be exactly evaluated, the 95%-confidence interval for the maximum likelihood estimate of the parameter can be obtained by a cut-off on the likelihood curve at $\frac{z_{0.975}^2}{2} = 1.92$, where $z_{0.975}$ is the 0.975 quantile of the standard normal distribution. In large, complex models where the likelihoods of data are estimated with Monte Carlo methods with non-negligible amount of error, the uncertainty in the likelihood estimates has to be taken into account in computing the cut-off. Ionides et al. [2017] developed a general procedure for constructing confidence intervals for a parameter of interest when the profile likelihoods with respect to that parameter can be estimated with some Monte Carlo errors. The procedure for constructing the Monte Carlo adjusted profile (MCAP) confidence intervals are as follows.

We assume that the Monte Carlo profile points $\breve{\ell}^P_{1:K}$ are evaluated at $\phi_{1:K}$. We fit a smooth curve $\breve{\ell}^S(\phi)$ through the profile points using a local smoother, such as the R function `loess` [Cleveland et al., 1992]. The MLE of the parameter ϕ can be taken as the point $\breve{\phi}$ at which the maximum of the smoothed curve $\breve{\ell}^S$ is attained. In order to quantify the Monte Carlo error in the estimated maximum likelihood $\breve{\ell}^S(\breve{\phi})$, we make a local quadratic fit near the maximum, using the weights $w_{1:K}$ that were used in evaluating the smoothed curve $\breve{\ell}^S$ at $\breve{\phi}$. Write the fitted quadratic equation as $-\breve{a}\phi^2 + \breve{b}\phi + \breve{c}$. The variance and covariance of the coefficients $\breve{\mathrm{Var}}[\breve{a}]$, $\breve{\mathrm{Var}}[\breve{b}]$, and $\breve{\mathrm{Cov}}[\breve{a}, \breve{b}]$ can be obtained as usual. Using the delta method, the standard error of the maximum $\frac{\breve{b}}{2\breve{a}}$ can be estimated as

$$\mathrm{SE}^2_{\mathrm{mc}} = \frac{1}{4\breve{a}^2}\left(\breve{\mathrm{Var}}[\breve{b}] - \frac{2\breve{b}}{\breve{a}}\breve{\mathrm{Cov}}[\breve{a}, \breve{b}] + \frac{\breve{b}^2}{\breve{a}^2}\breve{\mathrm{Var}}[\breve{a}]\right).$$

On the other hand, the statistical error originating from the randomness in data can be estimated with the usual formula

$$\mathrm{SE}_{\mathrm{stat}} = \frac{1}{\sqrt{2\breve{a}}}.$$

Assuming that the size of the Monte Carlo error is roughly the same across the possible realizations of the data, we can reasonably approximate the total standard error of the Monte Carlo maximum likelihood estimate as

$$\mathrm{SE}_{\mathrm{total}} = \sqrt{\mathrm{SE}^2_{\mathrm{stat}} + \mathrm{SE}^2_{\mathrm{mc}}}.$$

It follows that the cut-off for an approximate $(1-\alpha)$ confidence interval can be obtained as

$$\delta = \chi_\alpha \left(\breve{a} \times \mathrm{SE}^2_{\mathrm{total}}\right) = \chi_\alpha \left(\breve{a} \times \mathrm{SE}^2_{\mathrm{mc}} + \frac{1}{2}\right),$$

where χ_α is the $(1-\alpha)$ quantile of the χ-square distribution on one degree of freedom.

3.4 Analysis of artificially generated data from the model

We first implemented the GIRF algorithm to artificially generated data for 832 weeks from year 1949 to 1964 for twenty cities in England and Wales in the mid 20th century. We simulated the model using the real birth and population data and the parameters summarized in Table 3.1. In order to estimate the profile likelihood curve for G, we estimated other parameters using Algorithm 2, while fixing the gravitation constant at various levels. Twenty cities was enough to stretch our computational resources—one might like to study larger collections of cities, but one should bear in mind that inference for the full nonlinear coupled dynamics of epidemics in only twenty cities is a scientific advance.

We assumed the initial states were known. The choice of the guide function u_t was as described in Section 3.2. The number of sub-intervals for each observation time interval was taken to equal the number of cities, that is $S = 20$. We first estimated the reporting probability ρ, before making inference on G. After confirming that the estimated profile likelihood plot for ρ achieved the maximum at the true value of 0.5, we made inference on G while the reporting probability was fixed at the this value. This two-stage approach was motivated by the following two reasons. First, the curvature of the log likelihood in the direction of the reporting probability was much greater than that in the direction of the gravitation constant. Thus, the Monte Carlo error in finding the MLE for the reporting probability could overshadow the effect of the gravitation constant, unless filtering was iterated many times to reduce

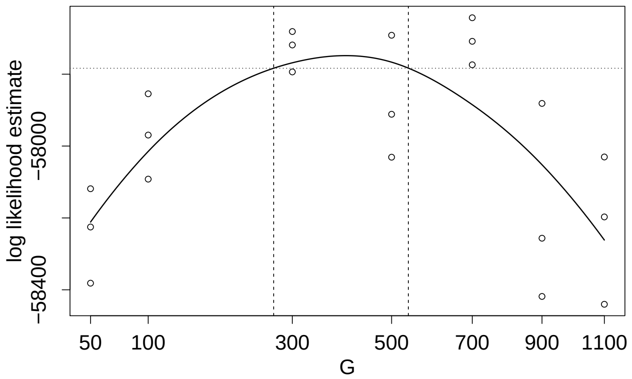

Figure 3.1: Estimated profile likelihood for G from artificially generated data and the approximate 95% confidence interval.

the maximization error. Second, little direct correlation between the reporting probability and the gravitation constant conditional on data was expected from model construction. A problem-specific inference procedure like this may not be necessary with large number of particles and sufficient number of iterations of filtering for parameter estimation. However, such a procedure can bring a substantial increase in computational efficiency in practice.

We repeated the parameter estimation procedure independently for five times for each value of G. Each repetition used four thousand particles and comprised eight filtering iterations while the parameter perturbation size decreased at a geometric factor of 0.92. The parameter estimates obtained at the end of the last iteration were taken as the estimated Monte Carlo MLEs. The likelihoods at the estimated MLEs were then evaluated with Algorithm 1. For each likelihood evaluation, five particle islands of four thousand particles each were used. The estimated profile likelihoods were both affected by the Monte Carlo error in finding the MLEs and the Monte Carlo error in evaluating the likelihoods. Each filtering took on average 69 hours.

We constructed an approximate 95% confidence interval for the gravitation constant G. Diggle and Gratton [1984] considered methods of parameter inference from noisy estimates of likelihoods from models that are implicitly defined by simulation algorithms. Ionides et al. [2017] further developed the methods and proposed a procedure to construct Monte Carlo adjusted profile (MCAP) confidence intervals. We used this procedure for our analysis. A short description of the procedure is provided in Section 3.3. Figure 3.1 shows the estimates of profile log likelihoods and the approximate 95% confidence interval for G. We kept the three points with highest estimated likelihood out of five repetitions for each value of G, to make the analysis robust to occasional unsuccessful Monte Carlo searches. A smooth fit through the estimated profile likelihoods was obtained using the non-parametric local regression procedure `loess` [Cleveland et al., 1992, implemented in R-3.4.1]. This procedure was carried out on a transformed scale of $\sqrt{G}$ for a better quadratic fit. The approximate confidence interval was found to be $(268, 539)$, indicated by the vertical lines, using a Monte Carlo adjusted profile cut-off of 35.1 log units. The confidence interval contained the truth at $G = 500$. Although the estimated profile likelihood points had considerable Monte Carlo error, they contained enough information to make inference on the spatial coupling parameter.

3.5 Data analysis: Measles cases in England and Wales in year 1949–1964

We also made inference on G from the real weekly case reports data collated by Bolker and Grenfell [1995]. The same dataset was also examined by He et al. [2009]. We used the city-specific reporting probabilities estimated by He et al. [2009], which we found to closely match the ratios of the cumulative number of reported cases to the total number of births in the corresponding period in all cities. This agreed with what was expected from the model construction.

Since the initial condition was not known, both the IVPs and the non-IVPs were estimated. The IVPs and the non-IVPs were alternatingly estimated for eight iterations of the following procedure. For the estimation of the IVPs given the non-IVPs obtained from the previous iteration, we ran the GIRF over the first three weeks starting with perturbed IVPs. We only filtered for the first three weeks because the information about the initial states was concentrated on the early data points. We used fifty islands each comprising eighty particles for this task, because using more islands helped prevent a quick collapse of the parameter swarm into a single point. The filtering over the first three weeks was iterated sixty times, during which the swarm of IVPs moved toward the region of higher likelihood conditioned on the non-IVPs obtained from the previous iteration. Once the IVPs were estimated, the non-IVPs were estimated conditioned on the IVP values. For this, we conducted one filtering run over the whole data, where only the non-IVPs were perturbed continuously at every intermediate time point. This process used one island comprising four thousand particles. Once the eight rounds of the alternating estimation of the IVPs and the non-IVPs were finished, the final values of the non-IVPs were taken as the MLE. Finally, the IVPs were estimated conditioned on the estimated MLE for the non-IVPs before likelihood evaluation. The likelihoods at the MLEs were computed with Algorithm 1, using five islands of four thousand particles. This entire procedure was repeated independently seven times for each fixed value of G. Each filtering took on average 76 hours.

Figure 3.2 shows the estimated profile likelihood for the gravitation constant G. We kept the four points with highest estimated likelihood out of seven for each value of G. The construction of the MCAP confidence interval took place on the $\sqrt{G}$ scale. The MLE for G was estimated to be 321, and the 95%-confidence interval $(254, 387)$ with a cut-off of 33.1. The data contained enough information to enable inference

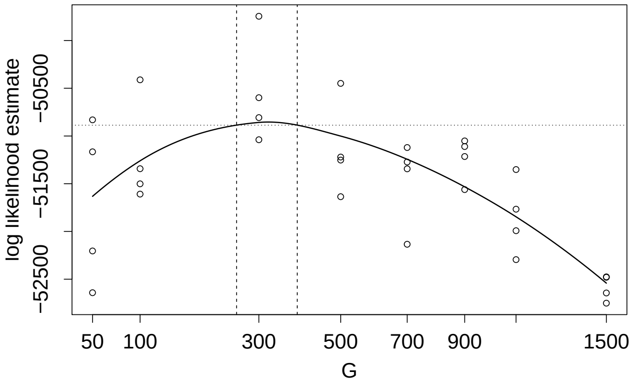

Figure 3.2: Estimated profile likelihood for G from real case data and the approximate 95% confidence interval.

on G.

3.6 Post-hoc analysis of inference results

In this section, we provide some post-hoc analysis plots for the spatiotemporal measles transmission considered in Section 3.1. Figure 3.3 shows the results of filtering on the artificially generated data, which was analyzed in Section 3.4, with the same parameter set used to generate the data. The true and estimated values for the susceptible, exposed, and infectious compartment sizes and the weekly total diagnoses or recoveries are shown for three cities of varying sizes, London, Cardiff, and Halesworth. These plots show that the state trajectories were correctly estimated. The estimated mean, the median, and the tenth and ninetieth percentiles are shown in black, blue, and green lines, respectively. The mean and median were almost the same, so they are not visually distinguishable in the plot. The estimated state means plus minus two standard errors are marked with grey shades. The true state trajectory, marked by red curves, lay mostly within the estimated tenth and ninetieth percentiles throughout the time period, for the exposed and infectious compartment sizes and the weekly diagnoses or recoveries (the second, third, and fourth rows). The susceptible compartment size reflects the cumulative number of infections, so it has a long memory, or equivalently, slow mixing. The estimated trajectory for the susceptible compartment closely follow the changes in the true trajectory, but there are biases in the estimates that are roughly constant for long periods of time. These biases mostly result from the inaccurate estimates of the number of infections at epidemic peaks. Due to the measurement error, the filtering distribution has some degree of spread in the number of infections at the peaks. However, the long term effect of the number of infections at the peaks remains in the susceptible compartment size. Therefore, accurately estimating the susceptible compartment size with a filtering method is a fundamentally difficult task. Still, our filtering results produced reasonable estimates of the susceptible compartment size to the extent that the inaccuracy in the estimated number of susceptibles do not seriously impact the ability to estimate other compartment sizes. These results show that severe particle depletion did not occur in our analysis, which used a complex twenty dimensional model.

For the real weekly case reports data analyzed in Section 3.5, we compared the real observation sequence with the sequences generated by our model at the estimated MLE. We simulated the model at the estimated MLE point which produced the highest likelihood estimate in Figure 3.2, except for the spatial coupling parameter G, which we varied. The purposes of this comparison were to gauge the degree of model misspecification and to see the differences between the observation sequences generated with different values of G. Figure 3.4 show the simulated data sequences for London, Cardiff, and Halesworth when G was set to 0, 100, 321 (the estimated MLE), 1500. When the data was simulated at $G = 0$, the measles epidemic died out within two years, which was certainly different from the observed data. On the other hand, when we set G to 100, 321, or 1500, the generated data sequences showed a similar pattern as the real data, and no clearly distinguishable visual differences could be found between the three simulated sequences. These results suggest that the model we used had the capacity to generate data similar to the real observations, but that it would probably be very hard or impossible to accurately estimate the value of G with ad-hoc methods. In Section 3.5, however, we estimated the MLE for G and its 95% confidence interval, which had fairly small width, using a likelihood-based inference.

3.7 Remarks

Sharp deterioration of standard particle filters with increasing dimensions has been an obstacle to making inference from spatiotemporal data using coupled nonlinear dynamic models. Our GIRF may not be enough for very high dimensional models, but it does offer an advance in analyzing coupled highly nonlinear dynamic systems of moderate dimensions. Potential applications may be found in areas such as ecology, behavioral sciences, or epidemiology, when the data are collected at linked spatial locations or structured into many categories. Many scientific and statistical challenges remain involving analysis of partially observed, highly nonlinear, coupled stochastic systems, and we have shown that the GIRF approach provides a framework for progress in this enterprise.

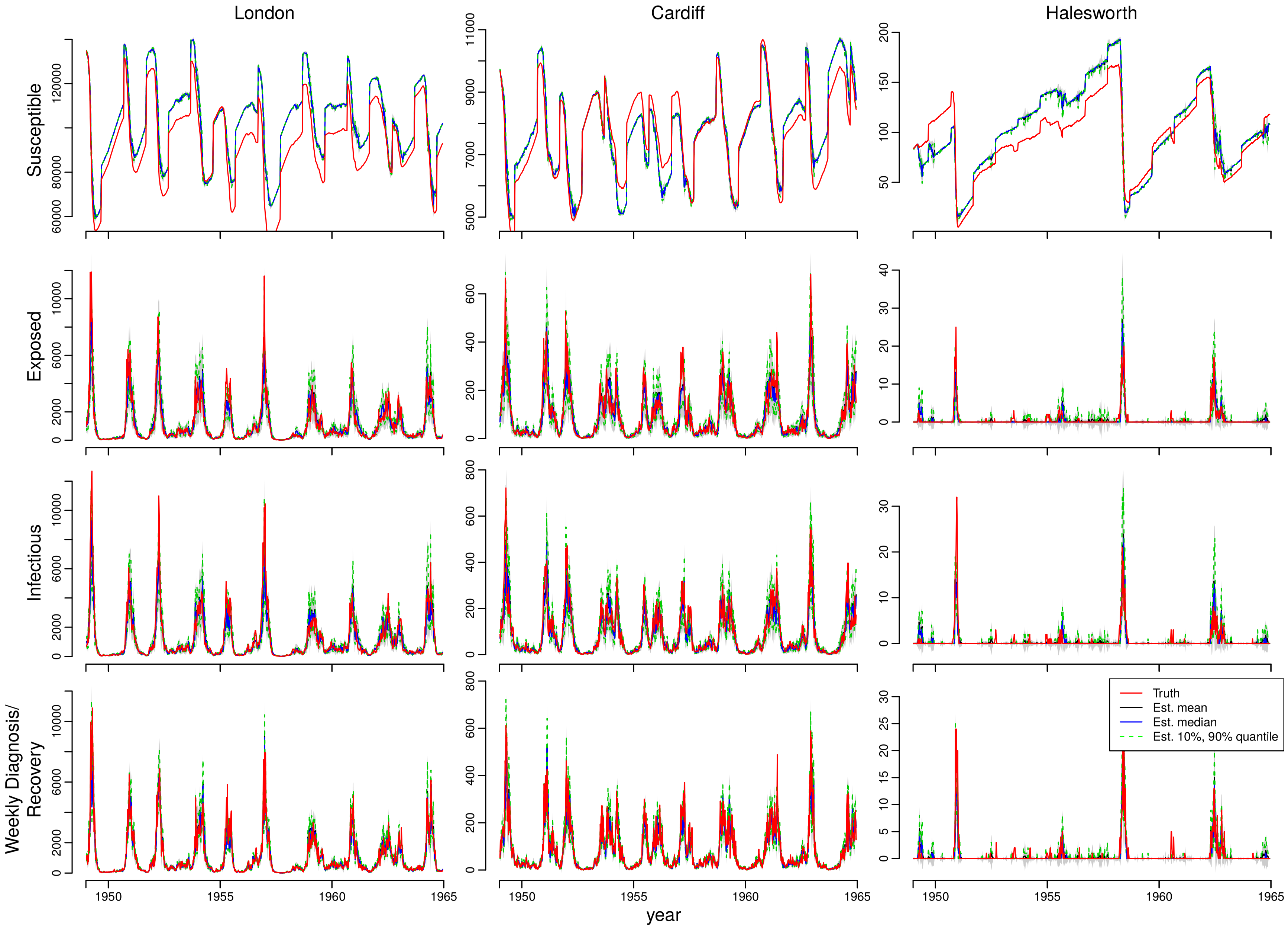

Figure 3.3: The filter estimates for the susceptible, exposed, and infectious compartment sizes and the weekly diagnoses or recoveries in the population of London, Cardiff, and Halesworth. The estimated mean, median, and tenth and ninetieth percentiles of the filtered particles are shown in black, blue, and green lines, respectively. The estimated mean plus minus two standard errors are shaded in grey. The true numbers are shown in red. In all the cities, the estimated state means for the exposed and infectious compartment sizes and the weekly diagnoses or recoveries were very close to the true values, and the truth was mostly between the estimated tenth and ninetieth percentiles. The susceptible compartment size, which has slow mixing due to being a cumulative process, was estimated with a trajectory that follows the truth but with some degree of long lasting biases.

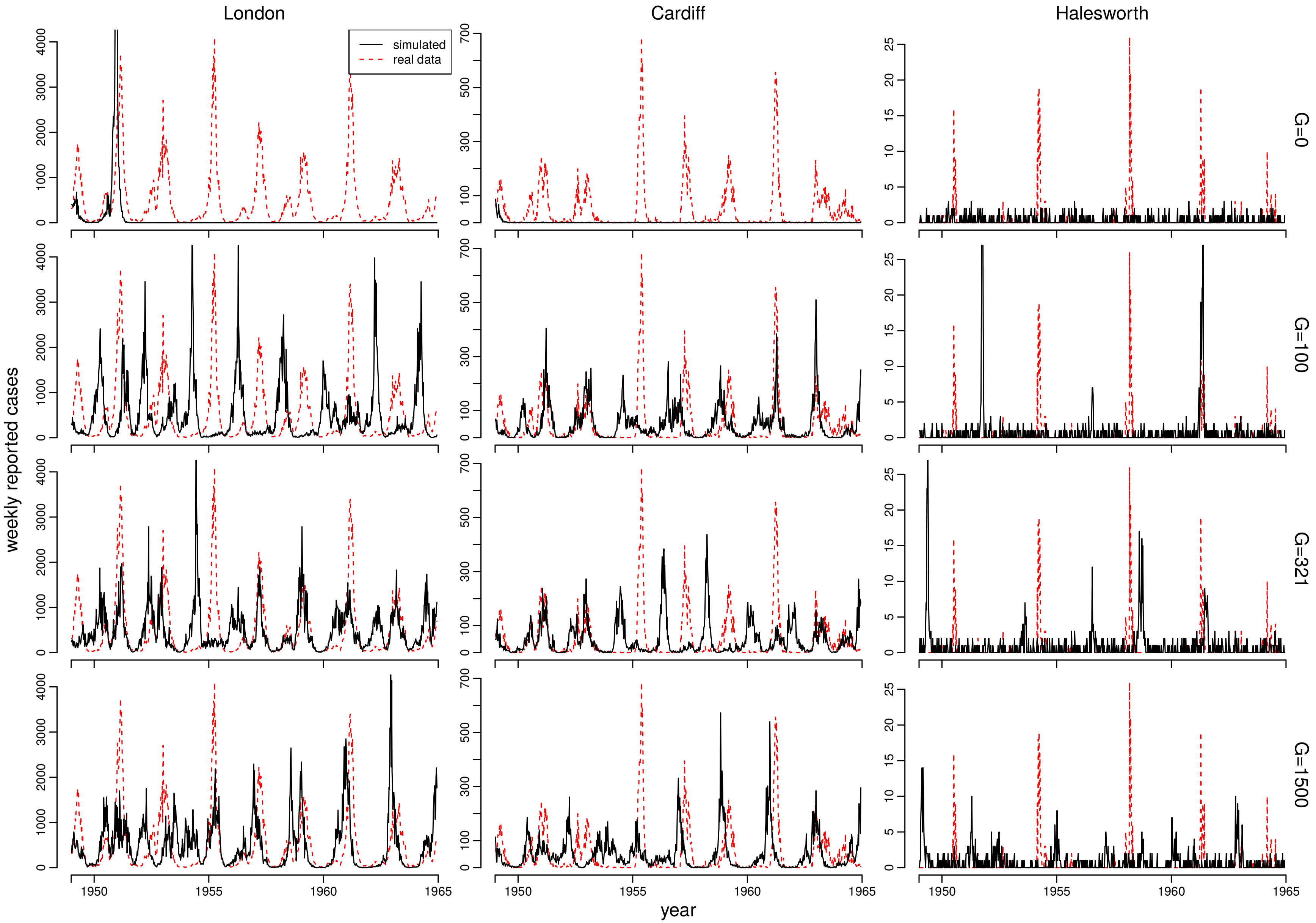

Figure 3.4: We simulated the measles model in Section 3.1 varying G, with other parameters set at the Monte Carlo MLE from Figure 3.2 that produced the highest likelihood estimate. The simulated weekly reported cases for London, Cardiff, and Halesworth are plotted in black solid lines. The real case data are plotted in red dashed lines for comparison. The simulated data at $G = 0$ show case elimination, which is certainly different from the real data. However, the simulated data at $G = 100$, 321, and 1500 are hard to distinguish by simply looking at the plots. On the contrary, likelihood based inference in Section 3.5 using the GIRF enabled the estimation of the MLE, found to be $G = 321$.

CHAPTER IV

Multiple proposal Markov chain Monte Carlo

In this chapter, I explore a new framework in Markov chain Monte Carlo (MCMC) sampling in which multiple proposals are made for the next state of the Markov chain. The framework, which offers a new insight into the Metropolis-Hastings (M-H) strategy, can be used to generalize various existing MCMC algorithms. I illustrate that the multiple proposal framework can be applied not only to algorithms that employ random proposal kernels, but also to piecewise deterministic algorithms such as Hamiltonian Monte Carlo (HMC) or the bouncy particle sampler (BPS). The generalization offers practical benefits by facilitating better mixing of the Markov chain. When the multiple proposal framework is combined with Hamiltonian Monte Carlo methods, it facilitates flexible tuning of the step size of the numerical approximation procedure called the leapfrog method and increases computational efficiency. When combined with the bouncy particle sampler, it allows the Markov chain to pass through regions of low target density and explore the sample space more efficiently.

4.1 Introduction

Markov chain Monte Carlo (MCMC) methods have been very widely used to sample from distributions with unnormalized densities. The goal of MCMC methods is to obtain samples from the target distribution with density

$$\bar{\pi}(x) := \frac{\pi(x)}{Z}$$

defined on the space $\mathbb{X}$, where $\pi(x)$ denotes an unnormalized density, and Z denotes the corresponding normalizing constant. MCMC methods construct a Markov chain, which can start at an arbitrary initial state $X^{(0)}$. Given the current state of the Markov chain $X^{(i)}$, the next state $X^{(i+1)}$ is drawn from a kernel $K(\cdot|X^{(i)})$, which has the target distribution $\bar{\pi}$ as its invariant distribution. Many MCMC methods adopt the Metropolis-Hastings (M-H) strategy, in which a sample from the kernel $K(\cdot|X^{(i)})$ is drawn in two stages [Hastings, 1970]. First, a proposal Y is drawn from a proposal kernel Q, and second, the proposal is either accepted as $X^{(i+1)}$ or rejected with certain probability. When the proposal is rejected, the next state of the chain equals the current state $X^{(i)}$. The invariance of the target distribution is achieved by appropriately choosing the acceptance probability.

There exist various MCMC methods that use different types of proposal kernels. Metropolis-Hastings algorithms refer to a general class of algorithms that use random proposal kernels with density $q(y|x)$ with respect to the same reference measure for the target density $\bar{\pi}(x)$. Other algorithms exist that make piecewise deterministic proposals. Hamiltonian Monte Carlo methods [Duane et al., 1987] and the bouncy particle sampler methods [Peters et al., 2012, Bouchard-Côté et al., 2017] belong to this category and share the similarity that they originated from simulation methods of physical systems.

In this chapter, I show that the multiple proposal framework can be applied to both types of MCMC algorithms that make random or piecewise deterministic proposals.

4.2 Multiple proposal Metropolis-Hastings algorithms

4.2.1 Algorithm description

M-H type kernels make proposals for the next state of the Markov chain using a proposal distribution, whose conditional density is denoted by $q(y \,|\, x)$. In the standard M-H sampling, given the current state $X^{(i)}$, the proposal Y drawn from $Y \sim q(\cdot \,|\, X^{(i)})$ is accepted with the probability

$$\alpha(Y, X^{(i)}) := \min\left(1, \frac{\pi(Y)\, q(X^{(i)} \,|\, Y)}{\pi(X^{(i)})\, q(Y \,|\, X^{(i)})}\right).$$

This is algorithmically implemented by drawing a uniform random variable $\Lambda \sim \mathrm{unif}(0,1)$ and accepting the proposal if and only if $\Lambda < \frac{\pi(Y)\, q(X^{(i)} \,|\, Y)}{\pi(X^{(i)})\, q(Y \,|\, X^{(i)})}$. When accepted, we take $X^{(i+1)} \leftarrow Y$ and when rejected, $X^{(i+1)} \leftarrow X^{(i)}$. This acceptance probability ensures that the detailed balance equations hold for the Markov chain $\{X^{(i)}\}$ and its stationary distribution is $\bar{\pi}$.

In multiple proposal Metropolis-Hastings algorithms, we make subsequent proposals from the rejected values. The number of sequential proposals we make, denoted by N, can be any fixed or random number, provided that it is independent of the proposal draws and the decision of whether the proposals are acceptable or not. Each of these N proposals are deemed either acceptable or not, and we take the L-th acceptable value as the next state of the Markov chain. If there are less than L acceptable values among the N proposals, the next state of the Markov chain remains the same as the current state. The number L can be fixed or random, and can be jointly drawn with N.

Throughout this chapter, for two integers n and m, we will denote by $n\!:\!m$ the sequence $(n, n+1, \dots, m)$ if $n \leq m$ and the sequence $(n, n-1, \dots, m)$ if $n > m$. Also, given a sequence $(a_n)_{n\in\mathbb{Z}^+} = (a_1, a_2, \dots)$, we will denote by $a_{n:m}$ the subsequence

```
Algorithm 3: Multiple proposal Metropolis Hasting algorithm
  Input   : The distribution of the maximum number of proposals and the maximum number
            of accepted proposals $\nu(N, L)$
            Proposal kernels $\{q_n(y_n \mid y_{n-1:0})\}$
            Number of iterations, $M$
  Output: Markov chain $\left(X^{(i)}\right)_{i\in 1:M}$
  Initialize: Set $X^{(0)}$ arbitrarily
  for $i \leftarrow 0 : M-1$ do
      Draw $(N, L) \sim \nu(\cdot, \cdot)$
      Draw $\Lambda \sim \mathrm{unif}(0, 1)$
      Set $X^{(i+1)} \leftarrow X^{(i)}$
      Set $n_a \leftarrow 0$
      for $n \leftarrow 1 : N$ do
          Draw $Y_n \sim q_n(\cdot \mid Y_{n-1:0})$, where we understand $Y_0 := X^{(i)}$
          if $\Lambda < \frac{\pi(Y_n)\left\{\prod_{j=2}^{n} q_{n-j+1}(Y_{j-1} \mid Y_{j:n})\right\} q_n(X^{(i)} \mid Y_{1:n})}{\pi(X) q_1(Y_1 \mid X^{(i)}) \left\{\prod_{j=2}^{n} q_j(Y_j \mid Y_{j-1:1}, X^{(i)})\right\}}$ then $n_a \leftarrow n_a + 1$
          if $n_a = L$ then
              Set $X^{(i+1)} \leftarrow Y_n$
              Break
          end
      end
  end
```

$(a_j)_{n \le j \le m}$.

The algorithm starts at an arbitrarily chosen initial state $X^{(0)}$. We denote by $X^{(i)}$ the state of the Markov chain after i updates. Let $N, L \in \mathbb{Z}^+ := \{1, 2, \dots\}$ with $N \geq L$ be drawn from a distribution whose probability mass function is denoted by $\nu(N, L)$. The algorithm draws $\Lambda \sim \mathrm{unif}(0, 1)$, independently of N and L. The algorithm draws the first proposal $Y_1 \sim q_1(\cdot \mid X^{(i)})$, independently of Λ and N. The proposal Y_1 is called acceptable if $\Lambda < \frac{\pi(Y_1) q_1(X^{(i)} \mid Y_1)}{\pi(X^{(i)}) q_1(Y_1 \mid X^{(i)})}$. The second proposal $Y_2 \sim q_2(\cdot \mid Y_1, X^{(i)})$ is drawn given the value of Y_1 and $X^{(i)}$. The proposal Y_2 is acceptable if $\Lambda < \frac{\pi(Y_2) q_1(Y_1 \mid Y_2) q_2(X^{(i)} \mid Y_1, Y_2)}{\pi(X^{(i)}) q_1(Y_1 \mid X^{(i)}) q_2(Y_2 \mid Y_1, X^{(i)})}$. The n-th proposal Y_n, $n \leq N$ is drawn from $q_n(\cdot \mid Y_{n-1:1}, X^{(i)})$ and called acceptable if

$$\Lambda < \frac{\pi(Y_n) \left\{\prod_{j=2}^{n} q_{n-j+1}(Y_{j-1} \mid Y_{j:n})\right\} q_n(X^{(i)} \mid Y_{1:n})}{\pi(X) q_1(Y_1 \mid X^{(i)}) \left\{\prod_{j=2}^{n} q_j(Y_j \mid Y_{j-1:1}, X^{(i)})\right\}}. \tag{4.1}$$

The procedure is repeated until L acceptable proposals are drawn or until N proposals are drawn, whichever comes sooner. The next state of the Markov chain $X^{(i+1)}$

is set to the L-th accepted value, or to $X^{(i)}$ if there are less than L acceptable values among $Y_1, \ldots, Y_N$. The pseudocode for this algorithm is shown in Algorithm 3. The standard Metropolis-Hastings algorithm corresponds to the case where N and L are both equal to 1.

We note that the proposal kernels q_n, $n \in 1\colon N$, can be simply taken as

$$q_n(y_n \,|\, y_{n-1:0}) \equiv q(y_n \,|\, y_{n-1})$$

for some proposal kernel density $q(\cdot \,|\, \cdot)$. In addition, if the kernel q is symmetric, that is, if $q(x \,|\, y) \equiv q(y \,|\, x)$, then the acceptability criterion (4.1) simplifies into

$$\Lambda < \frac{\pi(Y_n)}{\pi(X^{(i)})}.$$

We note that the multiple proposal Metropolis-Hastings algorithm with $L = 1$ constructs Markov chains with the same distribution as those constructed by delayed rejection (DR) methods [Tierney and Mira, 1999, Mira et al., 2001, Green and Mira, 2001]. The proof of this claim as well as a brief description of the delayed rejection method is provided in appendix (Section 4.A). However, the framework we present has several advantages over the delayed rejection:

1. First, our algorithmic framework is conceptually and algorithmically simpler. The expression for the acceptance rule is more concise than that given in the original papers on delayed rejection. Our framework provides a new perspective on why the rather convoluted acceptance probability formula in [Mira et al., 2001] is necessary.

2. Our framework is more broadly applicable than the delayed rejection method. For example, some MCMC algorithms, such as Hamiltonian Monte Carlo or the bouncy particle sampler methods, use piecewise deterministic kernels to draw

proposals. For these methods, the application of the delayed rejection is not straightforward. However, our framework can be applied to these methods.

3. Our framework is more general than the delayed rejection method. As mentioned earlier, when we use random proposal kernels with well defined densities, the delayed rejection approach is identical to the case where we take $L = 1$ in our framework, in terms of the law of the Markov chain.

4.2.2 Detailed balance of the multiple proposal scheme

We show that the Markov chain constructed by the multiple proposal Metropolis-Hastings algorithm (Algorithm 3) is reversible with stationary distribution $\bar{\pi}$. In other words, the stationary chain with initial distribution $\bar{\pi}$ satisfies detailed balance. If we denote the Markov kernel constructed by Algorithm 3 as $K(dx'; x)$, then the detailed balance with respect to $\bar{\pi}$ states that, for any $x, x' \in \mathbb{X}$,

$$\pi(x)K(dx'; x)dx = \pi(x')K(dx; x')dx'.$$

In the following proof, we denote the l-th rank of a given finite sequence $a_{n:m}$ by $r_l(a_{n:m})$; that is, if we reorder the sequence $a_{n:m}$ as $a_{(1)} \geq a_{(2)} \geq \cdots \geq a_{(m-n+1)}$, then $r_l(a_{n:m}) := a_{(l)}$. If l is greater than the length of the sequence $a_{n:m}$, we define $r_l(a_{n:m}) := 0$. We also define $r_0(a_{n:m}) := \infty$.

Proposition IV.1. *Algorithm 3 establishes detailed balance of the Markov chain $\left(X^{(i)}\right)$ with respect to the target density $\bar{\pi}$.*

Proof. It suffices to show the claim for fixed N and L. The general case immediately follows because the Markov kernel $K(dx'; x)$ is constructed as a mixture over N and L.

We will prove the n-step transition from the current state to the next state of the Markov chain satisfies detailed balance, for arbitrary n. That is, the probability density of taking y_n as the next state of the Markov chain starting from the current state y_0 through a sequence of proposals $y_{1:n-1}$ is the same as the probability density of taking y_0 starting from y_n after going through the proposals in reverse order $y_{n-1:1}$. The case for $n = 1$ is well known from the original Metropolis-Hastings algorithm.

We fix $n \geq 2$. For each $i \in 0\!:\!n$, we define a function $p_i : \mathbb{X}^{n+1} \to \mathbb{R}^+$ as

$$p_i(y_0, y_1, \ldots, y_n) = p_i(y_{0:n}) := \bar{\pi}(y_i) \cdot \prod_{j=1}^{i} q_j(y_{i-j} | y_{i-j+1:i}) \cdot \prod_{j=1}^{n-i} q_j(y_{i+j} | y_{i+j-1:i}).$$

We consider $y_0, \ldots, y_n$ fixed, and denote

$$h_i := p_i(y_{0:n}), \qquad \bar{h}_i := p_i(y_{n:0}),$$

and

$$c_i := \frac{h_i}{h_0}, \qquad \bar{c}_i := \frac{\bar{h}_i}{\bar{h}_0}.$$

By construction, $h_i = \bar{h}_{n-i}$ for all $i \in 0\!:\!n$. The probability density of drawing y_0 from $\bar{\pi}$ and subsequently drawing $y_{1:n}$ from proposal kernels $q_1, \ldots, q_n$ equals h_0. For a drawn uniform random variable $\Lambda \sim \text{unif}(0, 1)$, the first proposal y_1 is acceptable if and only if $\Lambda < c_1 = \frac{\pi(y_1)q_1(y_0|y_1)}{\pi(y_0)q_1(y_1|y_0)}$. Likewise, the i-th proposal is acceptable if and only if $\Lambda < c_i$.

The n-th proposal y_n is the L-th acceptable value if and only if there are exactly $L - 1$ acceptable proposals among $y_{1:n-1}$, and y_n is acceptable, that is,

$$\Lambda \geq r_L(c_{1:n-1}) \quad \text{and} \quad \Lambda < r_{L-1}(c_{1:n-1}) \quad \text{and} \quad \Lambda < c_n.$$

Since Λ is always less than one, this condition is equivalent to

$$\left\{r_L(c_{1:n-1}) \wedge c_n \wedge 1\right\} \leq \Lambda < \left\{r_{L-1}(c_{1:n-1}) \wedge c_n \wedge 1\right\},$$

where we denote $a \wedge b := \min(a, b)$. Therefore, the probability density of drawing

$y_{0:n}$ and taking y_n as the next state of the Markov chain equals

$$\begin{aligned} h_0 \cdot \Big[\big\{ r_{L-1}(c_{1:n-1}) \wedge c_n \wedge 1 \big\} - \big\{ r_L(c_{1:n-1}) \wedge c_n \wedge 1 \big\} \Big] \\ = \big\{ r_{L-1}(h_{1:n-1}) \wedge h_n \wedge h_0 \big\} - \big\{ r_L(h_{1:n-1}) \wedge h_n \wedge h_0 \big\} \\ = \big\{ r_{L-1}(\bar{h}_{1:n-1}) \wedge \bar{h}_0 \wedge \bar{h}_n \big\} - \big\{ r_L(\bar{h}_{1:n-1}) \wedge \bar{h}_0 \wedge \bar{h}_n \big\} \\ = \bar{h}_0 \Big[\big\{ r_{L-1}(\bar{c}_{1:n-1}) \wedge \bar{c}_n \wedge 1 \big\} - \big\{ r_L(\bar{c}_{1:n-1}) \wedge \bar{c}_n \wedge 1 \big\} \Big] . \end{aligned}$$

Noting that the expression in the last line equals the probability density of drawing $y_n, \dots, y_0$ in reverse order starting from y_n and taking y_0 as the next state of the Markov chain, we see that the n-step detailed balance holds. We reach the claimed detailed balance by simply combining the detailed balance for $n = 1 : N$. □

Corollary IV.2. *The target density $\bar{\pi}$ is a stationary distribution of the Markov chain constructed by Algorithm 3.*

4.2.3 Multiple proposal Metropolis adjusted Langevin algorithms

Metropolis adjusted Langevin algorithms (MALAs) have better scaling with dimension than random walk Metropolis algorithms [Roberts et al., 1997, Roberts and Rosenthal, 1998]. Specifically, the asymptotic efficiency of Metropolis adjusted Langevin algorithm scales as $O(d^{-1/3})$ whereas the efficiency of random walk Metropolis algorithms scale as $O(d^{-1})$.

The multiple proposal Metropolis-Hastings framework can be readily applied to MALAs. Given the current state of the Markov chain X, successive proposals $Y_1, Y_2, \dots$ are made according to

$$Y_n \sim N(Y_{n-1} + \nabla \log \pi(Y_{n-1})\epsilon, \sqrt{2\epsilon} I), \qquad n \geq 1$$

where $Y_0 := X$ is understood. In this section, we denote the density of the proposal Y_n at y_n given $Y_{n-1} = y_{n-1}$ by $q(y_n \,|\, y_{n-1})$. For a drawn uniform random variable

$\Lambda \sim \text{unif}(0,1)$, a proposal Y_n is called acceptable if

$$U < \frac{\pi(Y_n)\prod_{j=1}^{n} q(Y_{j-1}\,|\,Y_j)}{\pi(X)\prod_{j=1}^{n} q(Y_j\,|\,Y_{j-1})}. \tag{4.2}$$

The L-th acceptable value is taken as the next state of the Markov chain.

Roberts and Rosenthal [1998] showed that the optimal acceptance probability for the Metropolis-adjusted Langevin algorithm is 0.574 up to three decimal places. Taking small jump step size ϵ increases the acceptance probability, but makes the Markov chain stay near the same location for a longer period of time and hampers mixing. On the contrary, when implementing a multiple proposal MALA, the trade-off between the acceptance probability and the jump step size is relaxed. One can use a large jump size ϵ, and the algorithm may still find an acceptable place after multiple jumps. By taking large N, the Markov chain may land on an acceptable state far away from the starting point. In Appendix 4.B, we argue that the right hand side of (4.2) does not become too small even for large n and indeed increases on average as n increases, using a martingale argument.

4.3 Multiple proposal piecewise deterministic MCMC algorithms

4.3.1 Algorithm description

Some MCMC algorithms use piecewise deterministic proposal kernels to update the state of the Markov chain. Some of these algorithms extend the target distribution on space $\mathbb{X}$ to a joint distribution on a product space $\mathbb{X} \times \mathbb{V}$ whose marginal on $\mathbb{X}$ equals the original target distribution. Elements in space $\mathbb{V}$ determine how the piecewise deterministic kernel makes proposals. In this section, we will first describe the algorithms in an abstract setting. We will then present how specific algorithms, such as Hamiltonian Monte Carlo or the bouncy particle sampler, fit into this framework.

In the extended space $\mathbb{X} \times \mathbb{V}$, we assume that the target distribution has density

$\bar{\pi}(x)\psi(v)$ for $(x, v) \in \mathbb{X} \times \mathbb{V}$ with respect to a product reference measure denoted by $dx\,dv$. The $\mathbb{X}$-component of the reference measure dx is the same as the original reference measure on $\mathbb{X}$ for which the original target distribution has density $\bar{\pi}(x)$. The variable $v \in \mathbb{V}$ is often called the *momentum* variable in HMC and the *velocity* variable in the BPS. We define a collection of deterministic maps $S_\tau : \mathbb{X} \times \mathbb{V} \to \mathbb{X} \times \mathbb{V}$ for possibly various values of τ. The map S_τ may be interpreted as the evolution of a particle for time duration τ in a system, such that $S_\tau(x, v)$ denotes the final position-velocity pair of a particle that moves in the system with initial position x and initial velocity v.

In order to make sure that the target density $\bar{\pi}(x)\psi(v)$ is stationary in the algorithm, we impose some conditions on $\{S_\tau\}$ and $\psi(\cdot)$.

- **Measure preserving condition.** First, the map S_τ for each τ preserves the reference measure $dx\,dv$: that is, for every measurable set $A \in \mathbb{X} \times \mathbb{V}$,

$$\int \mathbb{1}_{[S_\tau(x,v)\in A]} dxdv = \int_A dxdv. \tag{4.3}$$

  Then for any integrable measurable function f, we have

$$\int f\{S(x, v)\} dxdv = \int f(x, v) dxdv.$$

- **Reversibility condition.** Second, we assume that there exists a velocity reflection operator $R(x) : \mathbb{V} \to \mathbb{V}$ defined for every point $x \in \mathbb{X}$, such that

$$R(x) \circ R(x) = \mathrm{id} \qquad \text{for all } x \in \mathbb{X}, \tag{4.4}$$

$$R(x) \text{ preserves the reference measure } dv, \tag{4.5}$$

$$\psi\{R(x)v\} = \psi(v) \qquad \text{for all } (x, v) \in \mathbb{X} \times \mathbb{V}, \tag{4.6}$$

  and if we define a map $T : \mathbb{X} \times \mathbb{V} \to \mathbb{X} \times \mathbb{V}$ as $T(x, v) := (x, R(x)v)$,

$$T \circ S_\tau \circ T \circ S_\tau = \mathrm{id} \qquad \text{for all } \tau. \tag{4.7}$$

Algorithm 4: Multiple proposal piecewise deterministic MCMC

```
Input  : The distribution of the maximum number of proposals and the maximum number
         of accepted proposals $\nu(N, L)$
         Time step length distribution $\mu(d\tau)$
         Velocity distribution density $\psi(v)$
         Time evolution operators $\{S_\tau\}$
         Velocity reflection operator $R(x)$
         Velocity refreshment probability $p^{\mathrm{ref}}$
         Number of iterations, $M$
Output: Markov chain $\left(X^{(i)}\right)_{i \in 1:M}$
Initialize: Set $X^{(0)}$ arbitrarily and draw $V^{(0)} \sim \psi(\cdot)$.
for $i \leftarrow 0 : M-1$ do
    Draw $N, L \sim \nu(\cdot, \cdot)$
    Draw $\tau \sim \mu(\cdot)$
    Draw $\Lambda \sim \mathrm{unif}(0, 1)$
    Set $(X^{(i+1)}, V^{(i+1)}) \leftarrow (X^{(i)}, R(X^{(i)}) V^{(i)})$
    Set $n_a \leftarrow 0$
    for $n \leftarrow 1 : N$ do
        Set $(Y_n, W_n) = S_\tau(Y_{n-1}, W_{n-1})$, where we understand $Y_0 := X^{(i)}$ and $W_0 := V^{(i)}$
        if $\Lambda < \dfrac{\pi(Y_n)\psi(W_n)}{\pi(X^{(i)})\psi(V^{(i)})}$ then $n_a \leftarrow n_a + 1$
        if $n_a = L$ then
            Set $(X^{(i+1)}, V^{(i+1)}) \leftarrow (Y_n, W_n)$
            Break
        end
    end
    With probability $p^{\mathrm{ref}}$, refresh $V^{(i+1)} \sim \psi(\,\cdot\,)$
end
```

In the above, id denotes the identity maps in the corresponding space $\mathbb{V}$ or $\mathbb{X} \times \mathbb{V}$. The reversibility condition can be understood as an abstraction of an aspect of the Hamiltonian dynamics that if we reverse the velocity of a particle and advance in time, the particle traces back its past trajectory. Its meaning will become clearer in the context of explicit cases of HMC or the BPS. The proof of the following lemma is provided in appendix.

Lemma IV.3. *Suppose* (4.4) *and* (4.7) *hold. Define recursively* $S_\tau^n := S_\tau^{n-1} \circ S_\tau$ *where* $S_\tau^1 = S_\tau$*. Then for any* $n \geq 1$*, we have* $T \circ S_\tau^n \circ T \circ S_\tau^n = id$*. Moreover,* S_τ *is a bijective map.*

Multiple proposal piecewise deterministic MCMC algorithms operate in a similar fashion as multiple proposal Metropolis-Hastings algorithms do. The pseudocode is shown in Algorithm 4. The main difference from Algorithm 3 is that proposals are obtained deterministically by the relation $(Y_n, W_n) = S_\tau(Y_{n-1}, W_{n-1})$ and that the acceptability criterion takes into account the density ψ as well. If there are less than L acceptable proposals in the sequence of proposals, the next state of the Markov chain is set to $(X^{(i+1)}, V^{(i+1)}) = (X^{(i)}, R(X^{(i)})V^{(i)})$. In order to facilitate better mixing, the velocity $V^{(i+1)}$ may be refreshed with a certain probability p^{ref} at the end of each iteration by drawing from $\psi(\cdot)$. The output $\left(X^{(i)}\right)_{i\in 1:M}$ is obtained by simply discarding the velocity variables $\left(V^{(i)}\right)_{i\in 1:M}$.

We finally note that the time length τ for the evolution map S_τ can be drawn either collectively or separately for each $n \in 1\!:\!N$ in Algorithm 4. The pseudocode in Algorithm 4 shows the case where τ is drawn collectively such that the same value of τ is used for all $n \in 1\!:\!N$. Instead, the line *Draw* $\tau \sim \mu(\cdot)$ can be moved right below the *for* $n \leftarrow 1\!:\!N$ *do* line such that for each $n \in 1\!:\!N$, a different value of τ_n is drawn independently and S_{τ_n} is used to obtain (Y_n, W_n).

The invariance of the target distribution $\pi(x)\psi(v)$ can be shown in a similar way as in Section 4.2.2.

Proposition IV.4. *Algorithm 4 constructs a reversible Markov chain with respect to the density* $\bar{\pi}(x)\psi(v)$.

Proof. See Appendix 4.D. □

4.4 Connection to Hamiltonian Monte Carlo methods

HMC methods are often explained in an analogy with the Hamiltonian dynamic system in which a particle moves according to the physical law described by the Hamiltonian equation of motion [Duane et al., 1987]. In this section, we will explain the algorithm in the framework of piecewise deterministic MCMC presented in the previous section.

Standard Hamiltonian Monte Carlo methods on continuous space start at an arbitrary point with the momentum drawn from $\psi(\cdot)$. For consistency in notation, we denote a momentum random variable by V and a nonrandom dummy variable by v. We define a Hamiltonian system for the target distribution with density $\bar{\pi}(x)\psi(v)$ as

$$H(x, v) := -\log \pi(x) - \log \psi(v). \tag{4.8}$$

The Hamiltonian is defined up to an additive constant by using $\pi(x)$ instead of possibly intractable $\bar{\pi}(x)$. An analogy with a physical Hamiltonian system can be drawn by interpreting the first term $-\log \pi(x)$ as the static potential energy and the second term $-\log \psi(v)$ as the kinetic energy. The Hamiltonian equation of motion (HEM) is defined as

$$\begin{aligned} \frac{dx}{dt} &= \frac{\partial H}{\partial v}, \\ \frac{dv}{dt} &= -\frac{\partial H}{\partial x}. \end{aligned} \tag{4.9}$$

A particle in a Hamiltonian system moves according to the HEM (4.9).

Hamiltonian Monte Carlo methods construct a Markov chain that simulates the time evolution of the Hamiltonian system (4.8). Given the current state of the Markov chain $(X^{(i)}, V^{(i)})$, a proposal for the next state of the Markov chain $(X^{(i+1)}, V^{(i+1)})$ is taken as a deterministic approximation to the final position and momentum of the particle with initial position $X^{(i)}$ and initial velocity $V^{(i)}$ after a certain time duration τ. We will denote the map from the initial position-momentum pair to the final pair by S_τ. Standard HMC methods can be described as a special case of Algorithm 4 applied to this map S_τ, where N and L are fixed at one and p^{ref} is set to unity.

An approximate numerical solution to the HEM (4.9) can be obtained by the leapfrog algorithm [Duane et al., 1987]. The leapfrog algorithm incrementally updates the momentum and the position variables in small time steps of length $\epsilon = \tau/m$ for some suitable choice of m. One iteration of the leapfrog algorithm alternately updates the momentum and position (x, v) to $(\tilde{x}, \tilde{v})$ as follows:

$$\begin{aligned} \tilde{v} &\leftarrow v + \frac{\epsilon}{2}\nabla\{-\log\pi(x)\} \\ \tilde{x} &\leftarrow x + \epsilon\tilde{v} \\ \tilde{v} &\leftarrow \tilde{v} + \frac{\epsilon}{2}\nabla\{-\log\pi(\tilde{x})\} \end{aligned} \tag{4.10}$$

In cases where evaluation of the gradient of $\log\pi$ is not possible, a different, analytically tractable density $\hat{\pi}$ may be used in (4.10). This change leads to definition of a new Hamiltonian $\hat{H}(x, v) = -\log\hat{\pi}(x) - \log\psi(v)$. Still, the acceptability criterion $\Lambda < \frac{\pi(Y_n)\psi(W_n)}{\pi(X^{(i)})\psi(V^{(i)})}$ in Algorithm 4 makes the HMC method target the original target density $\bar{\pi}(x)$.

It is worth noting that if an exact solution to the HEM (4.9) can be simulated, then the map S_τ^{exact} that maps the initial position-momentum pair to the final position-momentum pair along the solution path preserves the Hamiltonian: that is, for all (x, v),

$$H(x, v) = H\{S_\tau^{\text{exact}}(x, v)\}.$$

In this case, the acceptance probability is always equal to unity and the algorithm becomes rejection-free. When the solution is numerically approximated by the leapfrog algorithm, smaller leapfrog time step generally leads to more accurate approximation and higher acceptance probability. When an alternative Hamiltonian $\hat{H}$ is used to run the leapfrog algorithm, the acceptance probability generally becomes lower as the corresponding $\hat{\pi}$ becomes more different from the target density π.

We can check the measure preserving condition and the reversibility condition in 4.3.1 for both S_τ^{exact} and the one obtained by the leapfrog algorithm S_τ^{leap}. First, we check the reversibility condition. The velocity reversal operator can be simply taken as $R(x) = -\text{id}$ for all $x \in \mathbb{X}$. The conditions (4.4) and (4.5) are easily checked. If we take $\psi(v)$ to be a function of $|v|$, checking of (4.6) is also trivial. The condition (4.7) for S_τ^{exact} follows from the fact that the HEM (4.9) takes the same form under the transformation $\tilde{t} = -t$ and $\tilde{v} = -v$. Checking (4.7) for S_τ^{leap} only requires simple algebraic computations.

As for the measure preserving condition, it suffices to check that the determinant of the Jacobian of the map S_τ has absolute value equal to unity:

$$\left|\frac{\partial^2 S_\tau(x,v)}{\partial x \partial v}\right| = 1, \qquad \text{for all } (x,v) \in \mathbb{X} \times \mathbb{V}.$$

The above equation for S_τ^{exact} is provided by Liouville's theorem [Liouville, 1838]. Neal [2011] and Betancourt [2017] provide heuristic presentations of this fact. For S_τ^{leap}, we note that each of the transformations in (4.10) is a translation and thus has unit Jacobian.

4.4.1 Issues of tuning in the original HMC

Tuning of parameters is important to run HMC efficiently. The leapfrog step size ϵ and the number of leapfrog steps n^{leap} affects the accuracy of the approximated trajectory of the solution to the HEM, the acceptance probability of the proposal, and the autocorrelation of the resulting Markov chain. In this subsection, I briefly review the issues of tuning parameters in the original HMC setting based on Neal [2011] and references therein. On the basis of this background, the benefits of the multiple proposal scheme will be discussed in Section 4.4.2.

Asymptotically, as the leapfrog step size ϵ tends to zero, the one-step leapfrog approximation of the trajectory of the solution to the HEM scales as ϵ^3, and the leapfrog approximation error for a fixed length (that is, when $\epsilon \cdot n^{\text{leap}}$ is held fixed) scales as ϵ^2 [Leimkuhler and Reich, 2004]. However, if ϵ is not small enough, the leapfrog approximation may not be stable and diverge to infinity. Thus taking ϵ small enough is crucial to avoid unreasonably small acceptance probability. The upper limit on the size of ϵ to avoid unstable trajectories is roughly on the order of the size of the variability in the most restricted direction; in other words, in the case where the target distribution is multidimensional Gaussian with covariance matrix Σ, the leapfrog trajectory tends to be unstable if the step size ϵ exceeds roughly the square root of the smallest eigenvalue of Σ. However, the number of leapfrog jumps needed to traverse a fixed length is proportional to ϵ^{-1}, so taking ϵ too small can be computationally wasteful. The number of leapfrog jumps n^{leap} should be sufficiently large to ensure that the consecutive states of the Markov chain are not highly correlated.

The effective size of MCMC samples is conceptually defined as the number of independent samples that result in the same degree of variability in the resulting estimates as that obtained by the given MCMC samples. Thus a desirable choice for n^{leap} makes the leapfrog trajectory long enough such that the end point is almost independent of the starting point. The fact that the number of leapfrog jumps needed to obtain an almost independent sample point is proportional to ϵ^{-1} gives HMC a distinctive advantage over the random walk Metropolis algorithm [Neal, 2011]. In random walk Metropolis, if the proposal is Gaussian with standard deviation ζ, the standard deviation of combined n random walk proposals equals $\sqrt{n}\zeta$. Thus, the typical number of proposals needed to obtain an almost independent point needs to satisfy $\sqrt{n}\zeta \approx 1$, that is, $n \approx \zeta^{-2}$. However, the size of ζ needed to obtain reasonable acceptance probability is again on the order of the standard deviation in the most constrained direction (e.g., the square root of the smallest eigenvalue of the covariance matrix if the target distribution is Gaussian), which is of the same order as the required upper limit on the leapfrog step size ϵ.

The overall computational cost required to obtain a state almost independent of the current state is proportional to $1/(\epsilon \cdot a)$, where a is the average acceptance probability over all states in the target space $\mathbb{X}$. Neal [2011] gives a simple heuristic argument that the average difference in Hamiltonian between the start and end point of the trajectory scales as $\mathbb{E}\Delta H \approx \epsilon^4$. The same argument shows that the acceptance probability can be approximated by $2\Phi\left(-\sqrt{\mathbb{E}\Delta H/2}\right)$ where $\Phi(\cdot)$ denotes the cdf of the standard normal distribution. An approximate average cost to obtain an independent sample is thus given by

$$\frac{1}{(\mathbb{E}\Delta H)^{1/4} a(\mathbb{E}\Delta H)}.$$

This expression is minimized when $\mathbb{E}\Delta H = 0.41$ and $a(\mathbb{E}\Delta H) = 0.65$, which is consistent with the empirical findings in the literature [Neal, 1994, Creutz, 1988, Sexton and Weingarten, 1992].

The above argument also can be applied to see how HMC scales with space dimension d. If the target distribution is close to a product of the distribution for d independent variables, $\mathbb{E}\Delta H$ scales proportionally with d. Since we know $\mathbb{E}\Delta H$ scales as ϵ^4, in order to have reasonable acceptance probability, ϵ must scale such that $d \cdot \epsilon^4 \approx 1$, i.e., $\epsilon \approx d^{-1/4}$. This means that the number of leapfrog jumps needed to obtain an almost independent state is proportional to $d^{1/4}$. Since the computational cost for each leapfrog jump is proportional to d, the overall cost of computation scales as $d^{5/4}$.

4.4.2 Flexible tuning of HMC using multiple proposal scheme

In multiple proposal HMC, we have additional degrees of flexibility in tuning. In addition to the leapfrog jump size ϵ and the number of leapfrog jumps n^{leap}, we can tune the maximum number of proposed trials N, the maximum number of accepted proposals L, and the probability of velocity refreshment p^{ref}. In the original HMC (i.e., without multiple proposal), the leapfrog step size ϵ needs to be small enough to guarantee that the acceptance probability is not too small—if the proposal obtained after computing n^{leap} leapfrog jumps are rejected, these computations are wasted in the sense that they do not play a part in obtaining a new independent sample. However, if multiple proposals can be tried, that is if $N > 1$, smaller acceptance probability may still be feasible, because the computations that led to the current proposal is, instead of being wasted, continued to obtain subsequent proposals. The computations are wasted only if all N proposals are rejected (when we consider $L = 1$). Since the number of leapfrog jumps needed to obtain an almost independent sample is still proportional to ϵ^{-1}, the overall computational cost to get an independent sample is roughly proportional to

$$\frac{1}{\epsilon \cdot \{1 - (1-a)^N\}},$$

where a is the average acceptance probability for a single proposal. When a is not too small, the overall acceptance probability $1 - (1-a)^N$ becomes close to one with only a moderate number of N. For $N = 5$, the value of expected difference in Hamiltonian $\mathbb{E}\Delta H$ that minimizes the overall computational cost

$$\frac{1}{(\mathbb{E}\Delta H)^{1/4}\left[1 - \left\{1 - 2\Phi\left(-\sqrt{\mathbb{E}\Delta H/2}\right)\right\}^N\right]}$$

is $\mathbb{E}\Delta H = 1.77 (=: \mu_{\text{mp},5})$, at which $a(\mathbb{E}\Delta H) \approx 2\Phi(-\sqrt{\mathbb{E}\Delta H/2}) = 0.35$ and $\epsilon \approx (\mathbb{E}\Delta H)^{1/4} = 1.15$. In comparison with the original HMC where the optimal values were $\mathbb{E}\Delta H = 0.41 (=: \mu_{\text{orig}})$, at which $a(\mathbb{E}\Delta H) \approx 0.65$ and $\epsilon \approx 0.80$, the overall computational efficiency increases by a factor of

$$\frac{\mu_{\text{mp},5}^{1/4}\left[1 - \left\{1 - 2\Phi\left(-\sqrt{\mu_{\text{mp},5}/2}\right)^N\right\}\right]}{\mu_{\text{orig}}^{1/4} \cdot 2\Phi\left(-\sqrt{\mu_{\text{orig}}/2}\right)} = \frac{1.016}{0.521} = 1.95.$$

We argued previously that the advantage of HMC over random walk based methods comes from the fact that HMC can make long moves more easily. The number of steps needed is proportional to the distance traversed in HMC, but is proportional to its square in random walk based methods. If the number of leapfrog jumps is too small, the Markov chain from HMC essentially behaves like a random walk, because the velocity is refreshed to a random value before the trajectory makes a long move in one direction.

However, if n^{leap} is too large, the trajectory may double back on itself, because the solution to the HEM is confined to a level set of the Hamiltonian. We cannot simply stop the leapfrog jumps when the trajectory starts doubling back on itself, because then the stopping condition is correlated with the location of the proposal. Such a choice in general destroys the detailed balance and makes the algorithm target a wrong distribution. In order to solve this issue, Hoffman and Gelman [2014] proposed the No-U-Turn sampler (NUTS) where the simulated trajectory is successively extended to twice the current length in either forward or backward direction in the form of a binary tree, until a 'U-turn' is observed in any of the sub-binary tree. The next state of the Markov chain is selected randomly as one of the states in the trajectory. Due to the symmetric nature of the growth of the binary tree, the relationships of the stopping condition with the current state and with the next state of the Markov chain are the same, and the detailed balance holds.

In multiple proposal HMC, the length of the trajectory can be controlled by two additional tuning parameters, the maximum number of accepted proposals L and the velocity refreshing probability p^{ref}. If L is greater than one, the algorithm makes more than one set of leapfrog jumps in each iteration, making the trajectory longer. Also, if the velocity is not refreshed at the end of the current iteration, the next iteration extends the trajectory from the current iteration exactly as if the whole trajectory is one leapfrog simulation path. Here, due to the same issue of destroying detailed balance, L cannot be chosen based on the trajectory. However, the refreshment probability p^{ref} can be chosen depending on the current state of the Markov chain. This follows from the fact that the target probability distribution on the extended space $\mathbb{X} \times \mathbb{V}$ for both the state and velocity variable is given by the product of two independent distributions for the state and for the velocity. Suppose that after simulating the leapfrog trajectory and either accepting or rejecting the proposal, the state of the state and velocity pair is denoted by (x, v). We know from Proposition IV.4, the joint density of the state and velocity right after acceptance or rejection equals $\bar{\pi}(x)\psi(v)$. Suppose that with probability $p^{\text{ref}}(x)$ that depends on x, the velocity is re-drawn from the density $\psi(\cdot)$. Denote the final velocity by v', whether it was re-drawn or not. Then the probability that the pair (x, v') is in a Borel subset A of $\mathbb{X} \times \mathbb{V}$ equals

$$\begin{aligned}
(4.11)\quad \mathbb{E}\int \mathbf{1}[(x, v') \in A]\bar{\pi}(x)\psi(v)dxdv \\
= \int \mathbf{1}[(x, v') \in A]\bar{\pi}(x)\psi(v)p^{\text{ref}}(x)\psi(v')dxdvdv' \\
+ \int \mathbf{1}[(x, v) \in A]\bar{\pi}(x)\psi(v)\{1 - p^{\text{ref}}(x)\}dxdv.
\end{aligned}$$

Here, the first integral on the right hand side corresponds to the case where the velocity is refreshed and the second integral where the velocity is not refreshed. But the equation can be simplified to

$$\begin{aligned}
&\int \mathbf{1}[(x,v') \in A]\bar{\pi}(x)p^{\mathrm{ref}}(x)\psi(v')dxdv \\
&\qquad + \int \mathbf{1}[(x,v) \in A]\bar{\pi}(x)\{1-p^{\mathrm{ref}}(x)\}\psi(v)dxdv \\
&= \int \mathbf{1}[(x,v) \in A]\bar{\pi}(x)\psi(v)\{p^{\mathrm{ref}}(x)+1-p^{\mathrm{ref}}(x)\}dxdv \\
&= \int \mathbf{1}[(x,v) \in A]\bar{\pi}(x)\psi(v)dxdv.
\end{aligned}$$

This shows that the state-velocity pair after the probabilistic refreshment of velocity is distributed according to the target density $\bar{\pi}(x)\psi(v)$.

A simpler proof that shows not only the stationarity but also the reversibility of the velocity refreshment follows from the detailed balance equation

$$\bar{\pi}(x)\psi(v)p^{\mathrm{ref}}(x)\psi(v')dxdvdv' = \bar{\pi}(x)\psi(v')p^{\mathrm{ref}}(x)\psi(v)dxdvdv'$$

where the left hand side is interpreted as the probability that v is refreshed to v' and the right hand side that v' is refreshed to v.

Of course, the refreshment probability $p^{\mathrm{ref}}(x)$ can take a value either zero or one. At such points, whether the velocity is refreshed or not is decided deterministically. We also make a cautionary remark here. Although it might be tempting to choose p^{ref} dependent on the past history of the Markov chain as well as the current state, doing so can destroy the detailed balance and $\bar{\pi}(\cdot)\psi(\cdot)$ may no longer be a stationary density. To see this, let $(X, X^{(-1)}, X^{(-2)}, \ldots, X^{(-m)})$ denote the current and m past states of the Markov chain, and suppose that p^{ref} depends on these random variables. But $(X, X^{(-1)}, \ldots, X^{(-m)})$ and the current velocity V are not necessarily independent; knowing the previous states of X can reveal some information about the current velocity. Thus, if we denote the joint probability density of $(X, X^{(-1)}, \ldots, X^{(-m)}, V)$

as $p(x, x^{(-1)}, \ldots, x^{(-m)}, v)$, the detailed balance equation does not hold in general:

$$p(x, x^{(-1)}, \ldots, x^{(-m)}, v)\psi(v') \neq p(x, x^{(-1)}, \ldots, x^{(-m)}, v')\psi(v).$$

The independence between X and V holds only when they are marginalized over the past history.

The state-dependent velocity refreshment probability $p^{\mathrm{ref}}(x)$ can be used to flexibly tune the length of leapfrog trajectories. The average number of leapfrog jumps at which the trajectory starts doubling back scales roughly proportionally to the size (diameter) of the level set of the potential energy $U(x)$. Thus, the velocity refreshment probability $p^{\mathrm{ref}}(x)$ can be chosen inversely proportional to the size of the level set containing the current point x. The flexibility with which the average length of the trajectory can be taken depending on the current state may help improve the numerical efficiency of HMC. The size of the level set may be learned from the trace plots obtained from preliminary runs by counting the number of leapfrog steps that makes the trajectory double back at various starting points. Levy et al. [2017] has recently proposed a method that learns several state-dependent tuning parameters that are related to leapfrog jumps using neural networks. Although the methods requires a quite extensive pilot runs to tune these parameters, the numerical results showed that the performance of HMC can be substantially improved after tuning. If desired, the velocity refreshment probability p^{ref} may be learned in a similar fashion. An adaptive approach where the refreshment probability is tuned 'on-the-fly' may also be possible.

4.4.3 Numerical example

We used a one hundred dimensional Gaussian model considered in Neal [2011] to study the efficiency of the multiple proposal scheme. The Gaussian model consists of one hundred independent univariate random variables, so the covariance matrix is diagonal. The standard deviation in the first dimension equals 0.01 and the last equals 1.00, where the standard deviations in between increase with uniform increments of 0.01. This ill conditioned model is an example where HMC performs much better than random walk Metropolis algorithms. Both the leapfrog step size and the proposal standard deviation for random walk should be on the order of the smallest standard deviation (0.01 in this example) in order to have reasonable acceptance probabilities, but the number of jumps needed to find an almost independent sample in the direction of the largest standard deviation (1.00 in this example) is linear in the largest-to-smallest ratio of the standard deviations (100 in this example) for HMC, whereas the number is proportional to its square for random walk Metropolis.

We varied the the maximum number of proposals in an iteration N and the average leapfrog jump size. The value of N varied among one, five, and ten, and the average leapfrog jump sizes varied among 0.01, 012, 0.014, and 0.016. In order to avoid the situations where the leapfrog trajectory is close to a loop or a half loop, the leapfrog jump sizes at each iteration was randomly drawn uniformly from twenty percent around the average jump size (±20%). For Gaussian densities, the leapfrog trajectory diverges if the step size is greater than twice the standard deviation of the distribution. In this example, the leapfrog jump sizes were chosen such that the value after twenty percent inflation is still less than twice of the smallest standard deviation $\sigma_1 = 0.01$. Following Neal [2011], we fixed the number of leapfrog jumps to 150. The maximum number of accepted proposals L was fixed at one.

We generated $M = 100,000$ sample points for each experiment. We computed the acceptance probability and the effective sample size for the first component (with $\sigma_1 = 0.01$) and for the last component (with $\sigma_{100} = 1.00$). The effective sample

N	Leapfrog jump size	Acceptance probability	Time (secs)	ESS_1	$\frac{\text{ESS}_1}{\text{sec}}$	ESS_{100}	$\frac{\text{ESS}_{100}}{\text{sec}}$
1	0.010	0.935	646	6.87×10^4	106	7.64×10^4	118
1	0.012	0.900	650	5.57×10^4	86	1.17×10^5	180
1	0.014	0.845	638	3.96×10^4	62	1.39×10^5	217
1	0.016	0.756	647	3.13×10^4	48	1.35×10^5	209
5	0.010	0.992	665	1.10×10^5	165	8.92×10^4	134
5	0.012	0.985	689	1.13×10^5	164	1.51×10^5	219
5	0.014	0.974	708	1.19×10^5	168	2.13×10^5	301
5	0.016	0.947	752	1.10×10^5	147	2.33×10^5	309
10	0.010	0.997	667	1.15×10^5	172	9.03×10^4	135
10	0.012	0.994	706	1.21×10^5	171	1.55×10^5	219
10	0.014	0.990	728	1.38×10^5	190	2.26×10^5	310
10	0.016	0.979	789	1.40×10^5	178	2.59×10^5	328

Table 4.1: Acceptance probabilities and effective sample sizes for the the first component (with $\sigma_1 = 0.01$) and for the last component ($\sigma_{100} = 1.00$) at various N (the maximum number of proposals in an iteration) and the average leapfrog jump sizes. The average computation time in seconds and the effective sample sizes per second is also shown.

size was computed by estimating the spectral density of the time series at frequency zero using the R library `coda` [Plummer et al., 2006]. The results are shown in Table 4.1. The run time was separately measured by running forty independent runs with $M = 10,000$. The average run time of the forty runs were scaled for $M = 100,000$ by multiplying by ten. The effective sample sizes per second for the first and the last components are also shown.

We see that the acceptance probability increases as N increases. The acceptance probability for $N = 1$ (the standard HMC) decreases from 0.94 to 0.76 as the leapfrog jump size increases from 0.01 to 0.016. However, the acceptance probability for $N = 10$ ranges between 0.979 and 0.997. The computation time for $N = 10$ is between 3.5% and 22% greater than that for $N = 1$, depending on the leapfrog jump size. For $N > 1$, as we increase the leapfrog jump size, the acceptance probability decreases, and the algorithm does more computation to try subsequent proposals. However, for larger N, the probability of finding at least one acceptable proposal increases, and the overall acceptance probability increases. The effective sample size per second for $N = 10$ is also greater than that for $N = 1$. The highest effective sample size per second among the four leapfrog jump sizes for $N = 10$ is about 79% higher than that for $N = 1$ for the first component with $\sigma_1 = 0.01$. For the component with $\sigma_{100} = 1.00$, the efficiency increase is 51%.

4.4.4 Extension to discrete spaces

Extensions of HMC methods to discrete spaces have been considered in recent years [Zhang et al., 2016, Nishimura et al., 2017, Dinh et al., 2017]. I will explain the HMC methods on discrete spaces in the framework described in Section 4.3.1. The piecewise deterministic proposal map S_τ can be defined in various ways depending on the structure of the sample space. In this subsection, we illustrate a simple example of sampling from discrete sample space having d-dimensional lattice-like structure, that is, when the space is defined as a product $\mathbb{X} = \mathbb{X}_1 \times \mathbb{X}_2 \times \cdots \mathbb{X}_d$ where each component $\mathbb{X}_i$, $i \in 1\!:\!d$, is a set with finite or countable number of elements. We suppose each space component $\mathbb{X}_i$, $i \in 1\!:\!d$, is isomorphic to one of three types of sets, $\mathbb{Z}$, $\mathbb{Z}^+$, or $\{1, \ldots, k\}$ for some k. That is, we assume that for each $i \in 1\!:\!d$, there exist a set A_i of one of the three types just mentioned and a bijective map $\iota_i : A_i \to \mathbb{X}_i$. If $x_i = \iota_i(a) \in \mathbb{X}_i$ for some $a \in A_i$ and if $a + 1$ exists in A_i, we define the next element of x_i as $x_i^+ := \iota_i(a+1)$ and say x_i^+ exists in $\mathbb{X}_i$. Note that x_i^+ may not exist if $A_i = \{1, \ldots, k\}$ and $x_i = \iota_i(k)$. Likewise, we define the previous element of x_i as $x_i^- := \iota_i(a-1)$, if $a - 1$ exists in A_i.

We define the momentum variable to be an element of $\mathbb{V} = \{-1, 0, 1\}^d$. Each entry of v represents the direction in which the particle moves in each coordinate. We define the one step evolution map $S_1 : \mathbb{X} \times \mathbb{V} \to \mathbb{X} \times \mathbb{V}$ as $S_1(x, v) := (\tilde{x}, \tilde{v})$ where the i-th coordinates of $\tilde{x}$ and $\tilde{v}$ are defined as

$$(\tilde{x}_i, \tilde{v}_i) = \begin{cases} (x_i^+, v_i) & \text{if } v_i = 1 \text{ and } x_i^+ \text{exists.} \\ (x_i, -v_i) & \text{if } v_i = 1 \text{ and } x_i^+ \text{does not exist.} \\ (x_i^-, v_i) & \text{if } v_i = -1 \text{ and } x_i^- \text{exists.} \\ (x_i, -v_i) & \text{if } v_i = -1 \text{ and } x_i^- \text{does not exist.} \\ (x_i, v_i) & \text{if } v_i = 0 \end{cases}$$

Here, x_i and v_i denote the i-th components of x and v. The evolution map S_τ for $\tau \in \mathbb{Z}^+$ is defined as iterative composition of S_1, that is, $S_\tau := S_1^\tau$. We take the counting measure on $\mathbb{X}$ as the reference measure with respect to which the density $\bar{\pi}(x)$ is defined. We also take the counting measure on $\mathbb{V}$ as the reference measure on $\mathbb{V}$. The density of the velocity distribution $\psi(v)$ is taken such that $\psi(v) = \psi(-v)$ for all $vin\mathbb{V}$. One possible choice is to take ψ to be a function of $\|v\|_0 := \sum_{i=1}^d \mathbb{1}_{[v_i \neq 0]}$ only. In this case, $\psi(\cdot)$ defines a distribution on the number of nonzero components of v, and all elements of $\mathbb{V}$ with the same number of nonzero components are equally likely.

We check that the above construction satisfies the two conditions presented in Section 4.3.1. If we take $R(x) = -\text{id}$ for all $x \in \mathbb{X}$, (4.4) and (4.6) are easily satisfied. Since $R(x)$ is a bijection, it preserves the counting measure on $\mathbb{V}$. We can also check that $T \circ S_1$ is a self-inverse map, so the condition (4.7) follows. By Lemma IV.3, we see that S_τ is a bijection, so the counting measure on $\mathbb{X} \times \mathbb{V}$ is preserved by S_τ.

So far the algorithm just described does not seem to be related to HMC. We will now show how this algorithm may be seen as a HMC algorithm. The main idea is to view the uniform $(0, 1)$ random variable we draw at each iteration of Algorithm 4 as the kinetic energy of the particle. The connection is made by setting the initial kinetic energy K equal to $-\log \Lambda$. In this HMC formulation, we let each component of the momentum v to take any real value, and the momentum space becomes $\mathbb{V} = \mathbb{R}^d$. The Hamiltonian of a particle at location x and momentum v is defined as

$$H(x, v) := -\log \pi(x) + \sum_{i=1}^{d} |v_i|.$$

We assume that the kinetic energy $K := \sum_{i=1}^{d} |v_i| = \|v\|_1$ is equally shared by all momentum components with nonzero magnitude. That is, we let $v = \frac{K}{\|v\|_0}\big(\mathrm{sign}(v_1), \ldots, \mathrm{sign}(v_d)\big)$, where $\mathrm{sign}(a) = 1$ if $a \in \mathbb{R}$ is positive, $\mathrm{sign}(a) = -1$ if a is negative, and $\mathrm{sign}(0) = 0$. The Hamiltonian equation of motion (4.9) at discrete time $t \in \mathbb{Z}$ is interpreted in the following way. The first equation $\frac{dx}{dt} = \frac{\partial H}{\partial v}$ is interpreted as

$$x_i(t+1) - x_i(t) = \frac{\partial \|v\|_1}{\partial v_i} = \mathrm{sign}(v_i).$$

The second equation $\frac{dv}{dt} = -\frac{\partial H}{\partial x}$ is interpreted as

$$\|v(t+1)\|_1 - \|v(t)\|_1 = \log \pi\{x(t+1)\} - \log \pi\{x(t)\},$$

such that $v(t+1) = \frac{\|v(t)\|_1 - \log \pi\{x(t)\} + \log \pi\{x(t+1)\}}{\|v\|_0}\big(\mathrm{sign}(v_1), \ldots, \mathrm{sign}(v_d)\big)$. From this relation, we can easily see that the Hamiltonian of the system is preserved along the path, that is, $H\{x(t), v(t)\} = H\{x(t+1), v(t+1)\}$. A state (x, v) is physically admissible if the kinetic energy $K = \|v\|_1 \geq 0$. Under the relation $K(0) = -\log \Lambda$, the condition $K(t) \geq 0$ is equivalent to

$$\Lambda = \exp\{-K(0)\} = \exp\left[-K(t) - \log \pi\{x(0)\} + \log \pi\{x(t)\}\right] \leq \frac{\pi\{x(t)\}}{\pi\{x(0)\}}.$$

This agrees with the acceptability criterion in Algorithm 4. Since Λ is a uniform $(0, 1)$ random variable, the initial kinetic energy $K(0)$ is distributed according to the exponential distribution with unit rate parameter. This formulation agrees with the Laplacian Hamiltonian Monte Carlo formulation considered in Zhang et al. [2016] and Nishimura et al. [2017].

4.4.5 Extension to hybrid spaces

HMC methods can also be extended to hybrid spaces that contain both continuous and discrete variables. To show how this can be done, we will first assume that we have a valid time evolution map S^c that only changes the continuous state variables and the corresponding velocity variables and a valid time evolution map S^d that only changes the discrete state variables and the corresponding velocity variables. The map S^c for continuous variables can depend on the current state of discrete variables, and vice versa. We also assume that there are velocity reflection operators R^c and R^d that are tied to the maps S^c and S^d respectively and satisfy the reversibility conditions (4.4) to (4.7).

The algorithm for hybrid spaces can be described in the framework given by Section 4.3.1. We note that this algorithm is not specifically tied to Hamiltonian Monte Carlo methods. We define a map T that reflects both continuous and discrete velocity variables:

$$T(x^c, v^c, x^d, v^d) = \left(x^c, R^c(x)v^c, x^d, R^d(x)v^d\right).$$

The map T is self-inverse. The algorithm updates the joint state of continuous and discrete variables via the map $S^d \circ S^c \circ S^d$. Since both S^d and S^c are measure preserving, so is their composition $S^d \circ S^c \circ S^d$. Thus, the only condition that we need to check to show the validity of the algorithm is that the map

$$T \circ S^d \circ S^c \circ S^d$$

is self-inverse. From the condition (4.7), we know that $T \circ S^d \circ T = (S^d)^{-1}$ and the same relation holds for S^c. Thus, we see that

$$\begin{aligned}T \circ S^d \circ S^c \circ S^d &= T \circ S^d \circ T \circ T \circ S^c \circ T \circ T \circ S^d \circ T \circ T \\ &= (S^d)^{-1} \circ (S^c)^{-1} \circ (S^d)^{-1} \circ T \\ &= (T \circ S^d \circ S^c \circ S^d)^{-1}.\end{aligned}$$

Therefore, the algorithm that updates the joint state via $S^d \circ S^c \circ S^d$ is reversible with respect to the stationary density $\bar{\pi}(x^d, x^c)\psi^c(x^c)\psi^d(x^d)$. The above argument also readily shows that the update via $S^c \circ S^d \circ S^c$ is also valid.

More generally, if the state variable x can be partitioned as $(x^{(1)}, x^{(2)}, \ldots, x^{(n)})$, algorithms that sequentially update only one component at a time can be obtained. We define time evolution maps $S^{(1)}, S^{(2)}, \ldots, S^{(n)}$ that change only the corresponding components of x. Tied to these maps are the reflection operators $R^{(1)}, R^{(2)}, \ldots, R^{(n)}$ that only reflect the corresponding velocity components. Then, for example, updating the state via the map $S^{(1)} \circ S^{(2)} \circ \cdots \circ S^{(n)} \circ \cdots \circ S^{(2)} \circ S^{(1)}$ gives a valid algorithm. In general, any "palindrome" that remains the same when the order of composition is reversed will provide a valid algorithm.

4.5 Connection to the bouncy particle sampler

Recently, non-reversible, piecewise deterministic MCMC sampling methods called the bouncy particle sampler (BPS) algorithms have been explored [Peters et al., 2012, Bouchard-Côté et al., 2017]. The original BPS algorithms investigated in Peters et al. [2012] and Bouchard-Côté et al. [2017] constructs a rejection free continuous time Markov chain. In this thesis, we focus on presenting a new, alternative version of the BPS, in which the evolution of the Markov process is determined at discrete time points with probabilistic rejection operation. This alternative version of the algorithm is easier to implement for any distribution with evaluable unnormalized target density. The connection between the original, continuous time BPS and the discrete time version we present will be explained later in the section with some discussion of their merits and drawbacks. We note that a different version of discrete time BPS has been proposed in Vanetti et al. [2017].

We describe the discrete time BPS in the framework of piecewise deterministic MCMC established in Section 4.3.1. We assume the target sample space $\mathbb{X}$ is $\mathbb{R}^d$. We also take velocity space $\mathbb{V} = \mathbb{R}^d$. The reference measures on $\mathbb{X}$ and $\mathbb{V}$ are both taken as the Lebesgue measure on $\mathbb{R}^d$. For each x, we define $R(x) : \mathbb{V} \to \mathbb{V}$ to be a linear transform such that $R(x) \circ R(x) = \mathrm{id}$. The density of the velocity distribution $\psi(\cdot)$ is taken such that $\psi(v) = \psi\{R(x)v\}$ for all $v \in \mathbb{V}$. These conditions are exactly (4.4) and (4.6) in Section 4.3.1. In the current setting where $R(x)$ is a linear operator on $\mathbb{V}$, the condition that $R(x)$ preserves the reference measure dv follows immediately from the fact that $R(x)$ is a self-inverse operator, since it implies that $|\det R(x)| = 1$.

Although the algorithm is well defined for any choice of $\{R(x)\,; x \in \mathbb{X}\}$ and $\psi(\cdot)$ satisfying the above conditions, we note that a convenient choice for $\psi(\cdot)$ can be a multivariate Gaussian distribution

$$\psi(v) = \frac{1}{\sqrt{2\pi}^d \, |\det \Sigma|^{1/2}} \exp\{-v^T \Sigma^{-1} v\},$$

where Σ is a positive definite matrix. In this case, the conditions (4.4) and (4.6) hold if and only if

$$R(x) = \Sigma^{1/2}(I - 2P)\Sigma^{-1/2}$$

for a symmetric projection matrix P, that is $PP{=}P$ and $P^T{=}P$. A matrix P is a symmetric projection matrix in $\mathbb{R}^d$ if and only if it is a projection onto a subset of an orthonormal basis of $\mathbb{R}^d$, that is $P = \sum_{j \in A} e^j (e^j)^T$ for some $A \subseteq \{1, 2, \ldots, d\}$ and some orthonormal basis $(e^1, \ldots, e^d)$.

In the context of Algorithm 4, the time evolution map S_τ for the discrete time bouncy particle sampler is defined as

$$S_\tau(x, v) = (x - R(x)v\tau,\ -R(x)v).$$

The condition (4.7) can be easily checked for this S_τ. The map S_τ can be decomposed as $S_\tau = S_\tau^{(2)} \circ S_\tau^{(1)}$, where $S_\tau^{(1)}(x, v) = (x, -R(x)v)$ and $S_\tau^{(2)}(x, v) = (x + v\tau, v)$. Both $S_\tau^{(1)}$ and $S_\tau^{(2)}$ have determinants with unit absolute value. It follows that $|\det S_\tau| = 1$, and S_τ is preserves the Lebesgue measure on $\mathbb{X} \times \mathbb{V}$.

The multiple proposal piecewise deterministic algorithm (Algorithm 4) in this setup can be supplemented with one additional operation. This operation, which is optional, updates the velocity variable at the end of each iteration with a set of linear operators $R'(x) : \mathbb{V} \to \mathbb{V}$ defined for each $x \in \mathbb{X}$, which satisfies the conditions (4.4) and (4.6). That is, if the next state of the chain $(X^{(i+1)}, V^{(i+1)})$ is determined at the i-th iteration, we can optionally reflect the velocity once again such that $V^{(i+1)}$ is updated to $R'(X^{(i+1)})V^{(i+1)}$. For any point x in $\mathbb{X}$, the density $\psi(v)$ is left invariant by this operation, because under the change of variable $v' = R'(x)v$,

$$\psi(v')dv' = \psi\{R'(x)v\}\ |\det R'(x)|\, dv = \psi(v)dv.$$

The pseudocode for the multiple proposal discrete bouncy particle sampler is provided in appendix (Section 4.E).

In practical applications, $R(x)$ can be simply taken to be $-\mathrm{id}$ in order to minimize the number of computations $R(x)v$. In this case, given the current state of the Markov chain $(X^{(i)}, V^{(i)})$, a point along the line $X^{(i)} + V^{(i)}t$ for some $t \geq 0$ will be taken as the next state of the Markov chain. If proposals were rejected in this iteration, the next state $X^{(i+1)}$ will remain the same, but the velocity will be reflected

$V^{(i+1)} = R(X^{(i)})V^{(i)} = -V(i)$. The optional reflection with $R'(x)$ can facilitate mixing without refreshing the velocity by drawing directly from $\psi(\cdot)$.

Another possible choice of the operator $R(x)$ is the reflection with respect to the hyperplane perpendicular to the gradient of the log target density. If we denote $U(x) := -\log \pi(x)$, we define

$$R(x)v := v - 2\frac{\langle \nabla U(x), v \rangle}{\|\nabla U(x)\|^2} \nabla U(x). \tag{4.12}$$

Here, $\langle \cdot, \cdot \rangle$ and $\|\cdot\|$ denote the usual inner product and the $\mathcal{L}^2$ norm in the Euclidean space. The original BPS algorithm proposed in Peters et al. [2012] uses this reflection operator $R(x)$.

The original BPS algorithm can be viewed as a continuous time limit of a discrete BPS algorithm. In the infinitesimal time interval $[t, t+\Delta t)$, the probability of velocity reflection is given by

$$1 - \min\left(\frac{\pi\{x(t+\Delta t)\}}{\pi\{x(t)\}}, 1\right) = \max\{\langle \nabla U(x), v \rangle, 0\} \cdot \Delta t(1 + o(1)),$$

where $o(1)$ converges to zero as Δt tends to zero. The time of velocity reflection can be simulated as the first arrival time of non-homogeneous Poisson process, whose rate is given by

$$\lambda(t) := \max\left(\langle \nabla U(x + vt), v \rangle, 0\right). \tag{4.13}$$

The continuous time BPS can be particularly useful if the target sample space $\mathbb{X}$ is multidimensional and the target density can be factorized into many terms each of which is function of only a few coordinates [Bouchard-Côté et al., 2017, Vanetti et al., 2017]. Multi-particle systems such as the Potts model are well known models possessing this property [Peters et al., 2012]. However, a major drawback of this algorithm is that it can be difficult to sample the arrival time of the non-homogeneous

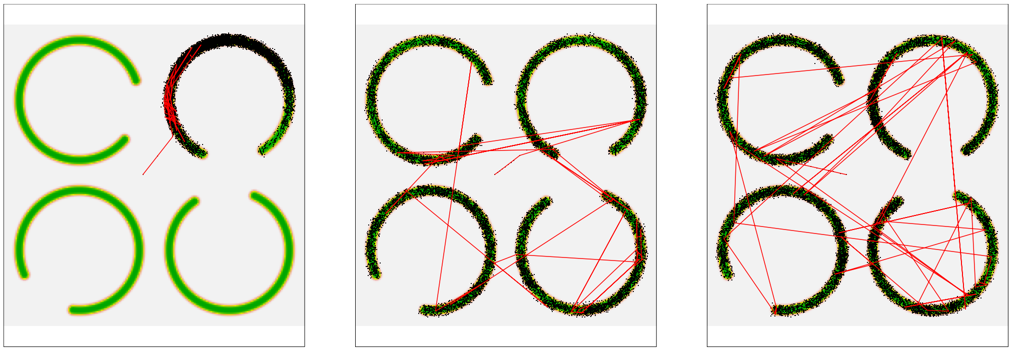

Figure 4.1: The first 30,000 sample points from the BPS for the model with four "C"s. The trajectory of every fourth point is shown in red segments up to one hundred points. Left, $N = 1$; middle, $N = 10$; right, $N = 20$.

Poisson process with rate given by (4.13). Techniques such as thinning or superposition can be used for sampling, but analytically tractable upper bounds on the integrated Poisson rate are needed for good numerical efficiency in many applications [Bouchard-Côté et al., 2017]. The discrete time BPS described earlier in the section does not face this difficulty and can be easily implemented for any target density evaluable up to a multiplicative constant.

4.5.1 Numerical example

We defined a density on a two dimensional square. The regions of high likelihood density look like four open rings, or four rotated letters of "C", as shown in Figure 4.1. We applied the multiple proposal discrete time BPS on this model with varying algorithmic parameters. In every experiment, we generated 30,000 sample points, where the maximum number of acceptable proposals L was fixed at one and the jump size varied uniformly between $\tau = 0.08$ and 0.12. We did not use an optional reflection by another operator $R'(x)$ at the end of iterations.

Figure 4.1 shows the 30,000 sampled points in black dots. Starting from the initial point, the trajectory of every fourth point is shown in red segments. We varied the maximum number of proposals N at one, ten, and twenty. The velocity refreshment

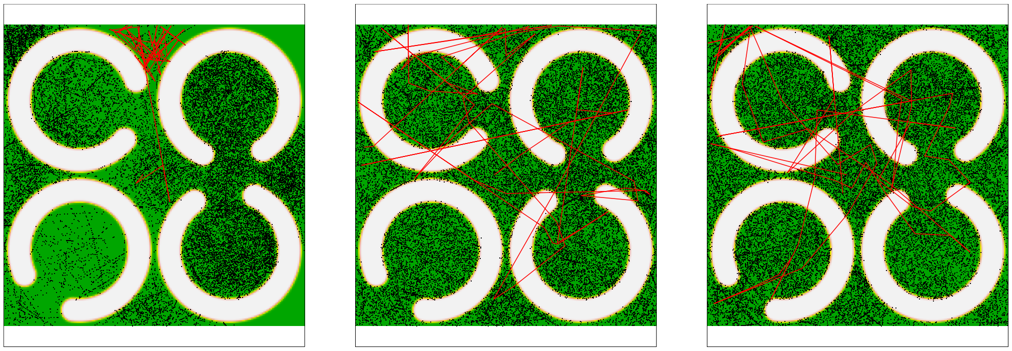

Figure 4.2: The first 30,000 sample points from the BPS for the inverted four “C” model. Left, $N = 1$; middle, $N = 10$; right, $N = 20$.

probability was fixed at $p^{\text{ref}} = 0.1$ for this experiment. We used the reflection operator $R(x) = -\text{id}$. In the case $N = 1$, in which no subsequent proposals are made after the first proposal is rejected, the Markov chain does not jump from one “C” to another. As we increase N to ten and twenty, we see that the jump between “C”s happen more frequently, and the mixing is faster. The computation time took 16.0 seconds for $N = 1$, 33.0 seconds for $N = 10$, and 39.3 seconds for $N = 20$.

Figure 4.2 shows the same experiment, when the target density is inverted from the original model (i.e., the log target density has the opposite sign). The four “C”s act as barriers that are difficult for particles to penetrate. For $N = 1$, the particles almost never pass through the barriers. For $N = 10$ or 20, however, particles can pass through the barriers more freely, and the mixing happens faster.

Figure 4.3 shows the results of sampling from the original model for various choice of the reflection operator and the velocity refreshment probability. The maximum number of proposals N was fixed at twenty. The left column shows the result when the reflection operator $R(x) = -\text{id}$ was used. In the middle column, the reflection operator defined in (4.12), which will be denoted by $R_{\nabla U}(x)$, was used. This operator $R_{\nabla U}(x)$ tends to direct the movement of the particle along the gradient of the log target density, because the velocity components in directions perpendicular

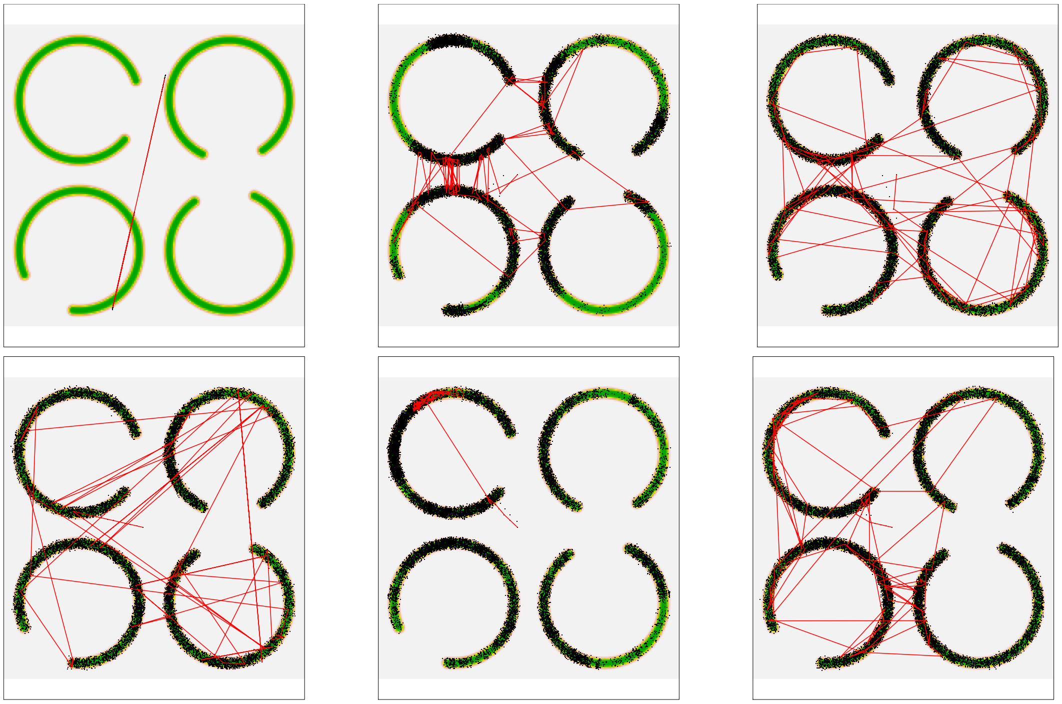

Figure 4.3: The first 30,000 sample points from the BPS for the four "C" model for varying p^{ref} and reflection operators. Top row, $p^{\text{ref}} = 0$; bottom, $p^{\text{ref}} = 0.1$.

to the gradient change signs at each jump so that two consecutive jumps in these perpendicular directions almost cancel each other. In the right column, the reflection operator was randomly chosen between $-\text{id}$ and $R_{\nabla U}(x)$ with equal probability whenever the reflection operator was called in the algorithm. When the velocity was not refreshed (top row) and the reflection operator was $-\text{id}$, the particle trajectory was linear, and the algorithm was not ergodic. When we used $R(x) = R_{\nabla U}(x)$, the algorithm was ergodic, but the mixing was poor. The mixing could be improved if the two choices of reflection operator were mixed. When the velocity was refreshed with probability 0.1 (bottom row), all the choices of reflection operator led to an ergodic chain. The mixing for $R(x) = -\text{id}$ or the mixed choice was better than the case $R(x) = R_{\nabla U}(x)$. From these results, we see that occasional refreshment of velocity is desirable. Randomly choosing the reflection operator among several possibilities can also be a good strategy.

4.6 Appendix for chapter IV

4.A Equivalence between multiple proposal Metropolis-Hastings algorithms and the delayed rejection method

The delayed rejection method is described in Mira et al. [2001] as follows. Given the current state of the Markov chain x, the first candidate value y_1 is drawn from $q_1(\cdot \mid x)$ and accepted with probability

$$\alpha_1(x, y_1) = 1 \wedge \frac{\pi(y_1) q_1(x \mid y_1)}{\pi(x) q_1(y_1 \mid x)}.$$

If y_1 is rejected, a next candidate value y_2 is drawn from $q_2(\cdot \mid y_1, x)$. The acceptance probability for y_2 equals

$$\alpha_2(x, y_1, y_2) = 1 \wedge \frac{\pi(y_2) q_1(y_1 \mid y_2) q_2(x \mid y_1, y_2)\{1 - \alpha_1(y_2, y_1)\}}{\pi(x) q_1(y_1 \mid x) q_2(y_2 \mid y_1, x)\{1 - \alpha_1(x, y_1)\}}.$$

If $y_1, \ldots, y_{n-1}$ are rejected, y_n is drawn from $q_n(\cdot \mid y_{n-1:1}, x)$ and accepted with probability

$$\alpha_n(x, y_{1:n}) = 1 \wedge \frac{\pi(y_n) \left\{\prod_{j=2}^{n} q_{n-j+1}(y_{j-1} \mid y_{j:n})\right\} q_n(x \mid y_{1:n}) \prod_{j=1}^{n-1}\{1 - \alpha_j(y_{n:n-j})\}}{\pi(x) q_1(y_1 \mid x) \left\{\prod_{j=1}^{n} q_j(y_j \mid y_{j-1:1}, x)\right\} \prod_{j=1}^{n-1}\{1 - \alpha_j(x, y_{1:j})\}}.$$

If all proposals are rejected up to a certain number N, the next state of the Markov chain is set to x.

It is relatively easy to check that the target density $\pi(\cdot)$ is invariant in the delayed rejection method. It suffices to check the detailed balance equation for each $n \in 1\!:\!N$. The probability density that starting from x, the proposals $y_1, \ldots, y_n$ are drawn and

y_n is the first accepted value equals

$$\pi(x)q_1(y_1\,|\,x)\prod_{j=2}^{n} q_j(y_j\,|\,y_{j-1:1},x)\prod_{j=1}^{n-1}\{1-\alpha_j(x,y_{1:j})\}\cdot\alpha_n(x,y_{1:n})$$
$$=\left[\pi(x)q_1(y_1\,|\,x)\prod_{j=2}^{n} q_j(y_j\,|\,y_{j-1:1},x)\prod_{j=1}^{n-1}\{1-\alpha_j(x,y_{1:j})\}\right]$$
$$\wedge\left[\pi(y_n)\prod_{j=2}^{n} q_{n-j+1}(y_{j-1}\,|\,y_{j:n})\cdot q_n(x\,|\,y_{1:n})\prod_{j=1}^{n-1}\{1-\alpha_j(y_{n:n-j})\}\right].$$

Since the above quantity is symmetric with respect to reversing the order $(x, y_{1:n})$ to $(y_{n:1}, x)$, it also equals the probability density that starting from y_n, the proposals $y_{n-1},\ldots,y_1,x$ are drawn and x is the first accepted value, which is given by

$$\pi(y_n)\prod_{j=2}^{n} q_{n-j+1}(y_{j-1}\,|\,y_{j:n})\cdot q_n(x\,|\,y_{1:n})\prod_{j=1}^{n-1}\{1-\alpha_j(y_{n:n-j})\}\cdot\alpha_n(y_{n:1},x).$$

We now show the following proposition.

Proposition IV.5. *Multiple proposal Metropolis-Hastings algorithm (Algorithm 3) for $L = 1$ and fixed N constructs a Markov chain that has the same law as those constructed by the delayed rejection method described above.*

Proof. Suppose that in Algorithm 3 and the delayed rejection algorithm, the Markov chain is at state x at a certain iteration. In both algorithms, the probability density that a sequence of proposals $y_{1:N}$ are drawn from the initial state x equals

$$q_1(y_1\,|\,x)\prod_{j=2}^{n} q_j(y_j\,|\,y_{j-1:1},x).$$

Thus it suffices to show that for every $n \in 1\!:\!N$, the probability that all of the drawn proposal values $y_{1:n}$ are rejected is the same in both algorithms. We define for $n \in 1\!:\!N$,

$$c_n := \frac{\pi(y_n)\left\{\prod_{j=2}^{n} q_{n-j+1}(y_{j-1}\,|\,y_{j:n})\right\} q_n(x\,|\,y_{1:n})}{\pi(x)q_1(y_1\,|\,x)\prod_{j=2}^{n} q_j(y_j\,|\,y_{j-1:1},x)}.$$

We also define

$$\bar{c}_j^n := \frac{\pi(y_{n-j}) \prod_{k=1}^{j} q_k(y_{n-j+k} \,|\, y_{n-j+k-1:n-j})}{\pi(y_n) \prod_{k=1}^{j} q_k(y_{n-k} \,|\, y_{n-k+1:n})}$$

for $j \in 1\!:\!n{-}1$ and $n \in 1\!:\!N$. Note that $\bar{c}_j^n$ equals the probability density of sequentially drawing $y_{n-j}, y_{n-j+1}, \ldots, y_n$ divided by the probability density of drawing $y_n, y_{n-1}, \ldots, y_{n-j}$ in sequence. From these definitions we can check that

$$c_n \bar{c}_j^n = c_{n-j}$$

for $j \in 1\!:\!n{-}1$ and $n \in 1\!:\!N$. For the delayed rejection method, we denote

$$\beta_j := \alpha_j(x, y_{1:j}), \quad \bar{\beta}_j^n := \alpha_j(y_{n:n-j}), \quad j \in 1\!:\!n{-}1.$$

In Algorithm 3, the probability that $y_{1:n}$ are all rejected equals

$$\mathbb{P}[\Lambda \geq c_j \text{ for all } j \in 1\!:\!n] = 1 - \max(c_{1:n}) \wedge 1$$

where $\Lambda \sim \text{unif}(0, 1)$. Thus our goal is to show that

$$\prod_{j=1}^{n} (1 - \beta_j) = 1 - \max(c_{1:n}) \wedge 1. \tag{4.14}$$

We prove (4.14) by induction. The case for $n = 1$ is obvious from the definition of β_1. Suppose we have

$$\prod_{j=1}^{n-1} (1 - \beta_j) = 1 - \max(c_{1:n-1}) \wedge 1.$$

In the above expression, both $\beta_{1:n-1}$ and $c_{1:n-1}$ are functions of x and $y_{1:n-1}$. By exchanging the role of the sequence of variables $x, y_1, \ldots, y_n$ in the above equation with $y_n, \ldots, y_1, x$, we obtain

$$\prod_{j=1}^{n-1} (1 - \bar{\beta}_j^n) = 1 - \max(\bar{c}_{1:n-1}^n) \wedge 1.$$

We observe that

$$\begin{aligned}\prod_{j=1}^{n}(1-\beta_j) &= \prod_{j=1}^{n-1}(1-\beta_j)\cdot\left(1 - c_n \frac{\prod_{j=1}^{n-1}(1-\bar{\beta}_j^n)}{\prod_{j=1}^{n-1}(1-\beta_j)} \wedge 1\right)\\ &= (1-\max(c_{1:n-1})\wedge 1) - \{c_n(1-\max(\bar{c}^n_{1:n-1})\wedge 1)\}\wedge(1-\max(c_{1:n-1})\wedge 1)\\ &= (1-\max(c_{1:n-1})\wedge 1) - (c_n - \max(c_{n-1:1})\wedge c_n)\wedge(1-\max(c_{1:n-1})\wedge 1).\end{aligned}$$

We now turn to the right hand side of (4.14). Using the relation

$$(u\vee v)\wedge w \equiv u\wedge w + v\wedge w - u\wedge v\wedge w,$$

we see that

$$\begin{aligned}1-\max(c_{1:n})\wedge 1 &= 1-(\max(c_{1:n-1})\vee c_n)\wedge 1\\ &= 1-\max(c_{1:n-1})\wedge 1 - c_n\wedge 1 + \max(c_{1:n-1})\wedge c_n\wedge 1.\end{aligned}$$

Thus showing (4.14) reduces to checking

$$(c_n - \max(c_{n-1:1})\wedge c_n)\wedge(1-\max(c_{1:n-1})\wedge 1) = c_n\wedge 1 - \max(c_{1:n-1})\wedge c_n\wedge 1.$$

However, this follows from the relation

$$(u - w\wedge u)\wedge(v-w\wedge v)\equiv u\wedge v - w\wedge u\wedge v.$$

The proof is now complete. □

4.B Acceptance probabilities in multiple proposal MALA

Now we argue that the acceptability criterion (4.2) is met frequently even for large n. We will demonstrate that the ratio of the products

$$\frac{\prod_{j=1}^{n} q(Y_{j-1}\,|\,Y_j)}{\prod_{j=1}^{n} q(Y_j\,|\,Y_{j-1})}$$

does not often become extremely small, even though it involves many multiplicative factors. It is easier to analyze the ratio in the continuous time limit, and we do so.

We fix $t > 0$ and let $n := \left[\frac{t}{\epsilon}\right]$. We can compute that

(4.15)

$$c_t^\epsilon := \frac{\pi(Y_n)\prod_{j=1}^n q(Y_{j-1}\,|\,Y_j)}{\pi(X)\prod_{j=1}^n q(Y_j\,|\,Y_{j-1})} = \frac{\exp\left\{-\frac{1}{4\epsilon}\sum_{j=1}^n \|Y_{j-1} - Y_j - \epsilon\nabla\log\pi(Y_j)\|^2\right\}}{\exp\left\{-\frac{1}{4\epsilon}\sum_{j=1}^n \|Y_j - Y_{j-1} - \epsilon\nabla\log\pi(Y_{j-1})\|^2\right\}}$$

$$= \exp\left[-\frac{1}{2}\sum_{j=1}^n (Y_j - Y_{j-1})\{\nabla\log\pi(Y_j) + \log\pi(Y_{j-1})\}\right.$$

$$\left. + \frac{\epsilon}{4}\left(\|\nabla\log\pi(Y_0)\|^2 - \|\nabla\log\pi(Y_n)\|^2\right)\right].$$

The sequence of random variables Y_j, $j \in 1\!:\!n$ is a discrete time approximation of the diffusion $\{\tilde{Y}_t\,;t \geq 0\}$ defined by the Langevin stochastic differential equation

$$d\tilde{Y}_t = \nabla\log\pi(\tilde{Y}_t)dt + \sqrt{2}W_t, \tag{4.16}$$

where W_t is a Brownian motion in $\mathbb{R}^d$. It is well known that the distribution with an unnormalized density π is the invariant distribution and the ergodic limit in total variation distance for the Langevin diffusion (4.16), provided that π is suitably smooth [Roberts and Tweedie, 1996]. If $\log\pi$ is C^3, that is, three times continuously differentiable, then $\{\nabla\log\pi(\tilde{Y}_t)\,;t \geq 0\}$ is a semimartingale due to Ito's formula [Rogers and Williams, 1994]. As $\epsilon \to 0$, the quantity (4.15) converges uniformly in probability in any compact interval $t \in [0, K]$ to [Rogers and Williams, 1994, Lemma IV.47.3]

$$c_t := \exp\left\{-\int_0^t \nabla\log\pi(Y_s)\cdot\partial Y_s\right\},$$

where $\int_0^t X_s\cdot\partial Y_s = \sum_{j=1}^d\int_0^t X_s^j\partial Y_s^j$ denotes the sum of Stratonovich integrals for d dimensional semimartingales X_t and Y_t. Uniform convergence in probability on compact intervals means that

$$\lim_{\epsilon\downarrow 0}\mathbb{P}\left(\sup_{0\leq t\leq K}|c_t^\epsilon - c_t| > \delta\right) \to 0$$

for any $\delta > 0$ and $K > 0$. The infinitesimal increment of the (i, j)-th entry of the quadratic variation-covariation matrix of $\tilde{Y}$, which is often denoted by $d\tilde{Y}_t^j d\tilde{Y}_t^k$, is given by

$$d\left([\tilde{Y}]_t\right)_{ij} = 2\delta_j^k dt,$$

where $\delta_j^k = 1$ if $j = k$ and 0 otherwise. For a C^2 function f on $\mathbb{R}^d$, we write the matrix of second-order partial derivatives, i.e., Hessian, of f as $\nabla^2 f$. We see by Ito's formula,

$$\begin{aligned} d\frac{\partial}{\partial x^i}\log\pi(\tilde{Y}_t) &= \sum_j \frac{\partial^2}{\partial x^i \partial x^j}\log\pi(\tilde{Y}_t)d\tilde{Y}_t^j + \frac{1}{2}\sum_{j,k}\frac{\partial^3}{\partial x^i \partial x^j \partial x^k}\log\pi(\tilde{Y}_t)d\tilde{Y}^j d\tilde{Y}^k \\ &= \left(\nabla^2 \log\pi(\tilde{Y}_t)d\tilde{Y}_t\right)_i + \sum_j \frac{\partial^3}{\partial x^i \partial x^j \partial x^j}\log\pi(\tilde{Y}_t)dt \\ &= \left(\nabla^2 \log\pi(\tilde{Y}_t)\cdot\sqrt{2}dW_t\right)_i + h(\tilde{Y}_t)dt, \end{aligned}$$

where $h(\tilde{Y}_t)$ is some function of $\tilde{Y}_t$. It follows that

$$\left[\tilde{Y}, \nabla\log\pi(\tilde{Y})\right]_t = \sqrt{2}\left[W, \int \nabla^2\log\pi(\tilde{Y}_t)\cdot\sqrt{2}dW\right]_t = 2\nabla^2\log\pi(\tilde{Y}_t)dt.$$

The Stratonovich integral $-\log c_t$ can be written as

$$\begin{aligned} \int_0^t \nabla\log\pi(\tilde{Y}_s)\cdot\partial\tilde{Y}_s &= \int_0^t \nabla\log\pi(\tilde{Y}_s)\cdot d\tilde{Y}_s + \frac{1}{2}\mathrm{Tr}\left(\left[\nabla\log\pi(\tilde{Y}), \tilde{Y}\right]_t\right) \\ &= \int_0^t \nabla\log\pi(\tilde{Y}_s)\cdot\sqrt{2}dW_s + \int_0^t \left\|\nabla\log\pi(\tilde{Y}_s)\right\|^2 ds + \int_0^t \Delta\log\pi(\tilde{Y}_s)ds, \end{aligned}$$

where $\int_0^t X_s \cdot dY_s := \sum_{j=1}^d \sum_{j=1}^d X_s^j dY_s^j$ denotes the sum of Ito integrals and $\Delta h := \sum_{j=1}^d \frac{\partial^2}{\partial(x^j)^2}h$ denotes the Laplacian of function h. By Ito's formula applied to the function $x \mapsto e^{-x}$, we have

$$\begin{aligned} dc_t = -c_t\cdot\Big(&\nabla\log\pi(\tilde{Y}_t)\cdot\sqrt{2}dW_t \\ &+ \left\|\nabla\log\pi(\tilde{Y}_t)\right\|^2 dt + \Delta\log\pi(\tilde{Y}_t)dt\Big) + \frac{1}{2}c_t\left\|\nabla\log\pi(\tilde{Y}_t)\cdot\sqrt{2}\right\|^2 dt \\ = -c_t&\nabla\log\pi(\tilde{Y}_t)\cdot\sqrt{2}dW_t - c_t\Delta\log\pi(\tilde{Y}_t)dt. \end{aligned}$$

The Ito integral of the first term

$$I_1(t) := \int_0^t -c_s \nabla \log(\tilde{Y}_t) \cdot \sqrt{2} dW_s$$

is a local martingale, and in particular if we define the stopping time

$$T_M := \inf\{t \geq 0\,;\, \int_0^t c_s^2 \left\| \nabla \log \tilde{Y}_s \right\|^2 ds > M\}$$

for some $M > 0$, then $\{I_1(T_M \wedge t)\,; t \geq 0\}$ is a martingale. Thus $\mathbb{E} I_1(T_M \wedge t) = 0$. Moreover, when the density π is log-concave, $-c_t \Delta \pi(\tilde{Y}_t)$ is always non-negative. It follows that for large M, ignoring the rare event $\{T_M < t_0\}$ where t_0 is the length of time for which we simulate the Langevin diffusion, the expected value of the c_t is increasing. Since c_0 is unity, one can expect that the continuous time limit of the acceptance probability, given by c_t, does not frequently become very small.

4.C Proof of Lemma IV.3

From (4.4), we have $T \circ T = \text{id}$. Thus from (4.7), we have $T = T \circ T \circ S_\tau \circ T \circ S_\tau = S_\tau \circ T \circ S_\tau$. Thus, we can see that $S_\tau^n \circ T$ is a self-inverse for any $n \geq 1$ from induction

$$S_\tau^n \circ T \circ S_\tau^n \circ T = S_\tau^{n-1} \circ S_\tau \circ T \circ S_\tau \circ S_\tau^{n-1} \circ T = S_\tau^{n-1} \circ T \circ S_\tau^{n-1} \circ T.$$

It also follows that $S_\tau \circ T \circ S_\tau \circ T = T \circ T = \text{id}$. Thus, since $f \circ g = \text{id}$ implies that function f is surjective and g is injective, the relation $S_\tau \circ (T \circ S_\tau \circ T) = \text{id}$ implies that S_τ is surjective and $(T \circ S_\tau \circ T) \circ S_\tau = \text{id}$ implies that S_τ is injective.

4.D Proof of invariance of target distribution for Algorithm 4

In Section 4.2.2, we proved that the detailed balance of the Markov kernel is established by the multiple proposal Metropolis-Hastings algorithm (Algorithm 3). In this section, we will show that each iteration of Algorithm 4 consists of two operations, for each of which detailed balance holds. The first operation applies the map $T : (x, v) \mapsto (x, R(x)v)$. The second operation samples from a kernel K, which is defined as follows. First, a uniform random variable $\Lambda \sim \text{unif}(0, 1)$ is drawn. Given (y_0, w_0), a sequence of proposals $(y_n, w_n) := S_\tau^n \circ T(y_0, w_0)$ are obtained, where (y_n, w_n) is deemed acceptable if $\Lambda < \frac{\pi(y_n)\psi(w_n)}{\pi(y_0)\psi(w_0)}$. If there are at least L acceptables among $(y_1, w_1), ..., (y_N, w_N)$, the L-th acceptable proposal is taken as a draw from

$$K(\cdot\,; y_0, w_0).$$

Otherwise, (y_0, w_0) is taken as the draw.

Proposals in Algorithm 4 are obtained as $(Y_n, W_n) = S_\tau^n(X^{(i)}, V^{(i)})$. Since $S_\tau^n = (S_\tau^n \circ T) \circ T$, each iteration of Algorithm 4 can be thought of as as applying the above two operations. That is, proposals are obtained by applying the maps $S_\tau^n \circ T$ to $T(X^{(i)}, V^{(i)}) = (X^{(i)}, R(X^{(i)})V^{(i)})$, and each proposal is checked for acceptability.

It is a simple matter to see that the first operation $T : (x, v) \mapsto (x, R(x)v)$ satisfies detailed balance with respect to the augmented target distribution $\bar{\pi}(x)\psi(v)$. If we write $v' = R(x)v$, since $\psi(v) = \psi(v')$ and $R(x)$ preserves the measure dv (property (4.5) and (4.6)), we have

$$\bar{\pi}(x)\psi(v)dxdv = \bar{\pi}(x)\psi(v')dxdv'.$$

The second operation also satisfies detailed balance.

Proposition IV.6. *Suppose a probability kernel K is defined as above. Then, the kernel K satisfies detailed balance with respect to the density $\bar{\pi}(y)\psi(w)$.*

Proof. The proof is essentially the same as the proof of Proposition IV.1. We will prove detailed balance for the n-step transition for arbitrary n. The probability density of drawing (y_0, w_0) and taking $(y_n, w_n) = S_\tau^n \circ T(y_0, w_0)$ as a draw from K

equals

$$\begin{aligned}(4.17)\quad &\frac{1}{Z}\pi(y_0)\psi(w_0)\left(\left[r_{L-1}\left(\left\{\frac{\pi(y_i)\psi(w_i)}{\pi(y_0)\psi(w_0)}\,;i\in 1\!:\!n-1\right\}\right)\wedge\frac{\pi(y_n)\psi(w_n)}{\pi(y_0)\psi(w_0)}\wedge 1\right]\right.\\ &\qquad\left.-\left[r_L\left(\left\{\frac{\pi(y_i)\psi(w_i)}{\pi(y_0)\psi(w_0)}\,;i\in 1\!:\!n-1\right\}\right)\wedge\frac{\pi(y_n)\psi(w_n)}{\pi(y_0)\psi(w_0)}\wedge 1\right]\right)\\ &=\frac{1}{Z}\left([r_{L-1}(\{\pi(y_i)\psi(w_i)\,;i\in 1\!:\!n-1\})\wedge\pi(y_n)\psi(w_n)\wedge\pi(y_0)\psi(w_0)]\right.\\ &\qquad\left.-[r_L(\{\pi(y_i)\psi(w_i)\,;i\in 1\!:\!n-1\})\wedge\pi(y_n)\psi(w_n)\wedge\pi(y_0)\psi(w_0)]\right).\end{aligned}$$

On the other hand, if we started from (y_n, w_n), the k-th proposal is obtained by

$$S_\tau^k\circ T(y_n,w_n)=S_\tau^k\circ T\circ S_\tau^k(y_{n-k},w_{n-k})=T(y_{n-k},w_{n-k})=(y_{n-k},R(y_{n-k})w_{n-k}),$$

because $S_\tau^k\circ T\circ S_\tau^k\circ T=\mathrm{id}$ from Lemma IV.3. Also, $\psi\{R(y_{n-k})w_{n-k}\}=\psi(w_{n-k})$. Thus, the probability density of drawing (y_n,w_n) from $\bar{\pi}(\cdot)\psi(\cdot)$ and taking $(y_0,w_0)=S_\tau^n\circ T(y_n,w_n)$ as a draw from K is the same as the right hand side of (4.17), which is symmetric with respect to the order reversal $i\mapsto n-i$. Furthermore, since both T and S_τ measure preserving, we have $dy_0dw_0=dy_ndw_n$. Therefore, the detailed balance for the n-step transition holds. The claimed detailed balance follows by combining the cases for $n=1\!:\!N$.

□

4.E Pseudocode for multiple proposal discrete bouncy particle sampler

Algorithm 5: Multiple proposal discretized bouncy particle sampler

```
Input  : The distribution of the maximum number of proposals and the number of accepted
         proposals $\nu(N, L)$
         Reflection operators $R(x), R'(x)$
         Time step length distribution $\mu(d\tau)$
         Velocity distribution density $\psi(v)$
         Velocity refreshment probability $p^{\mathrm{ref}}$
         Number of iterations, $M$
Output: Markov chain $\left(X^{(i)}\right)_{i=1,\ldots,M}$
Initialize: Set $X^{(0)}$ arbitrarily and draw $V^{(0)} \sim \psi(\cdot)$.
for $i \leftarrow 0 : M-1$ do
    Draw $N, L \sim \nu(\cdot, \cdot)$
    Draw $\epsilon \sim \mu(\cdot)$
    Draw $\Lambda \sim \mathrm{unif}(0, 1)$
    Set $(X^{(i+1)}, V^{(i+1)}) \leftarrow (X^{(i)}, R(X^{(i)})V^{(i)})$
    Set $n_a \leftarrow 0$
    for $n \leftarrow 1 : N$ do
        Set $(Y_n, W_n) = (Y_{n-1} - R(Y_{n-1})W_{n-1}\epsilon, -R(Y_{n-1})W_{n-1})$, where we understand
         $Y_0 := X^{(i)}$ and $W_0 := V^{(i)}$
        if $\Lambda < \dfrac{\pi(Y_n)}{\pi(X)}$ then $n_a \leftarrow n_a + 1$
        if $n_a = L$ then
            Set $(X^{(i+1)}, V^{(i+1)}) \leftarrow (Y_n, W_n)$
            Break
        end
    end
    (Optional) Reflect the velocity vector $V^{(i+1)} \leftarrow R'(X^{(i+1)})V^{(i+1)}$
    With probability $p^{\mathrm{ref}}$, refresh $V^{(i+1)} \sim \psi(\,\cdot\,)$
end
```

CHAPTER V

Conclusion

Modern data analyses employ complex models in order to gain detailed information about large and complicated systems. Rapidly increasing demands for inference from complex models call for new computational methods that enable these inferential tasks. In this thesis, I proposed the following two novel methods in computational inference procedures. I will briefly summarize these contributions and discuss future research directions.

Scalable inference methods for coupled partially observed Markov processes Coupled dynamic processes have complex spatial and temporal dependence structures. The large amount of information contained in spatiotemporal datasets is difficult to infer about because the nonlinearity of the processes often deny analytical approaches. Computational algorithms numerically represent the process in order to draw inference. I proposed a computational inference algorithm for coupled stochastic dynamic processes that are partially observed. This algorithm is a fully likelihood based method that enables the use of mechanistic models that might not have analytically tractable densities. The method is an extension of the particle filtering algorithm, which uses a number of Monte Carlo random draws to represent the hidden state of the process. The method I developed alters the analysis time scale of the particle filter and uses a guide function that aids in selecting particles that are consistent with future observations. Like other particle filtering algorithms, the method yields unbiased likelihood estimate of data and asymptotically consistent estimates of the posterior distribution of the process given the data. I also derived a probabilistic upper bound on the filtering error when finite Monte Carlo samples are used. This theoretical result explains how this method can scale much better with increasing space dimension than other particle filtering algorithms.

Practical accessibility of this algorithm hinges on how well the guide function can be designed for a given model. The method is a properly weighted particle filter for any choice of positive guide functions, but the numerical efficiency depends on how closely the guide functions approximate the likelihood of future data points. The flexibility with which the guide function can be chosen may be a strength of this method, but the fact that a good guide function is needed can limit its applicability. In this thesis, I proposed a few ways of designing the guide function. One method assumes that the likelihood of data points can be approximated semi-analytically by matching moments. Another method relies on the weak coupling between the components of the state process. Dependence on assumptions like these implies the fundamental difficulty in representing arbitrary high dimensional distributions with purely computational means. From an information theoretical perspective, the number of possible states that need to be realized to fully represent high dimensional distributions increase exponentially with the number of dimensions. In order to avoid infeasibly high computational costs, inference algorithms for coupled stochastic dynamic processes will have to make use of certain structure of the given process.

Further research on this topic can explore other possibilities for designing the guide functions. I expect that efforts to solve real scientific inference problems using the algorithm will enable development of efficient and practically useful guide functions.

I have found out that running parallel particle filters can often lead to better numerical results when particle weights tend to be highly unbalanced. Especially, I found out that weighting the parallel filters with improper weights—that is, the parallel filters are not weighted by their likelihood estimates but by some more balanced weights—can lead to smaller filtering error. This might be explained as the bias-variance trade-off, but theoretical analysis seems necessary to fully understand this phenomenon. I have not systematically investigated the potential benefits of this approach, but further pursuit of this idea may enable development of a useful algorithm.

The main theory for the algorithm (Theorem II.2) may be developed further. The assumptions on which the theory is based are difficult to check in practice. Sufficient conditions for the assumptions, if can be found, might be able to validate the practical use of the algorithm. Advances in the theory of guided intermediate resampling algorithms may deepen the understanding of the computational representation of high dimensional stochastic processes.

Likelihood based inference for spatiotemporal dynamics of infectious diseases I have shown that the guided intermediate resampling filter can be used for inference on spatiotemporal measles transmission dynamics in linked geographic locations. I have estimated the coupling parameter in a joint SEIR model with spatial interactions. Parameter estimation was carried out by combining the GIRF algorithm with a recently developed stochastic optimization algorithm [Ionides et al., 2015]. The strength of coupling between dynamics at different locations is difficult to reliably estimate, because the joint distribution of the dynamics processes has to be accurately represented computationally.

The model assumed that the coupling between dynamics were mediated by infectious travelers, whose numbers were determined by the gravity model. In the future, it will be interesting to compare the estimated spatial coupling strength with the values estimated from other spatiotemporal data. The comparison may tell us about the validity of the inference procedures. It may also provide important clues regarding the consistency between different spatiotemporal models and offer a guidance on translating the conclusion from one analysis into a different context.

There are several directions along which future research might be pursued. One direction that might be of scientific interest is to compare the gravity model with an alternative model such as the radiation model of Simini et al. [2012]. These authors showed that the radiation model explained data on commuting travels, migration, phone calls, and cargo volume in the U.S. and Europe better than the gravity model. Another comparison between the two models based on the analysis of infectious disease data using a likelihood based method might provide a deeper insight into spatial mixing patterns.

Application of the GIRF for spatiotemporal inference on other infectious diseases than measles might also yield interesting results. Infectious diseases with environmental reservoirs or vector-borne diseases are likely to have different spatiotemporal transmission mechanisms, which might be inferred from analyses using complex models that take into account the geography of the region or the ecology of the disease-carrying organisms.

The GIRF algorithm may be used for inference on coupled dynamic models other than spatiotemporal processes. For example, a compartment model that is structured into many sub-categories divided by age or other demographic variables can make a high dimensional POMP model. The algorithm may also be used for inference from heterogeneous data, such as the clinical case data combined with the pathogen genotype data. Recent developments in phylodynamic inference employ a joint model for transmission dynamics and pathogen evolution process [Smith et al., 2017]. Genetic data may contain large amount of information, and its measurement density can lead to weight degeneracy among particles. The tree-structured state processes can create an interesting inferential challenge, because the dimension of the state space increases with time. Since the state process accumulates its own history without forgetting its past, algorithms that make use of the mixing property might not perform well. Nevertheless, the guided intermediate resampling algorithm can be useful for phylodynamic inference, because it is a general strategy for mitigating computational difficulties provided by highly informative observations.

I have not yet implemented the guide function design in which the likelihood of future observations is estimated by a SMC-type method using weak coupling assumptions. Implementing this idea for real scientific applications is left as a future task. For weakly coupled processes, this approach can be more practically useful due to its flexibility compared to the semi-analytical moment matching approach.

Flexible, numerically efficient Markov chain Monte Carlo sampling strategy I developed a framework that generalizes various MCMC algorithms that adopt Metropolis-Hastings type acceptance or rejection strategy. The new framework proposes, after a proposal is rejected, a subsequent proposal that depends on the current, rejected proposal. The proposals continue until a certain number N of proposal is drawn or until a certain number L of acceptable proposals are drawn. When the number of acceptable proposals is less than L until N proposals are tried, the next state of the Markov chain is set to the current state of the Markov chain. This strategy can be used to generalize most frequently used MCMC algorithms, such as random walk Metropolis algorithm, Metropolis adjusted Langevin algorithm (MALA), Hamiltonian Monte Carlo (HMC), or the bouncy particle sampler (BPS). The multiple proposal framework offers increased flexibility in proposal draws, because proposals can be drawn from a composition of several kernels, each of which governs how the next proposal is drawn conditional on the current proposal. This framework is straightforward to implement—adding only a few lines of code will turn an MCMC algorithm into a multiple proposal algorithm.

The multiple proposal strategy can bring various advantages, depending on which algorithm it is applied to. In HMC, trying multiple proposals increases the overall acceptance probability of proposals, and thus increases the effective sample size per computational cost. This is possible because the numerically computed proposal paths stay close to the level set of Hamiltonian, and the acceptance probabilities can stay reasonably high even in long trajectories.

There may be various other possibilities that can increase the efficiency of HMC using the multiple proposal strategy, for which further explorations should follow. Combination of the multiple proposal strategy with existing schemes that are known to boost numerical efficiency, such as the No-U-Turn sampler by Hoffman and Gelman [2014], may also be possible. Potential synergy with MALA, which can be understood as a special case of HMC, may be further investigated.

The multiple proposal strategy can also be useful in combination with the bouncy particle sampler. The BPS has been being actively studied in recent years. Despite its many favorable numerical properties, the BPS is not good at making jumps across regions of low probability densities. This is because the piecewise linear sample paths tend to reflect when they encounter the low-density region. However, the multiple proposal BPS can pass through the low density regions, and facilitate better global mixing. With a numerical example, I demonstrated that this feature of the multiple proposal BPS can be particularly useful in sampling from complex distributions. Investigating the performance in high dimensional distributions is left as a future task.

Inference from large amount of data that enables in-depth understanding into complex systems is in high demand in science and engineering. Developing computational methodology for inference is a core task in an effort to meet this demand. In this thesis, I developed a few novel computational algorithms for complex models, focusing on likelihood based inference. With these developmens I showed a way to make progress in scientific inference procedures, which may open up possibilities for new scientific discoveries.

BIBLIOGRAPHY

Mel Ades and Peter J Van Leeuwen. The equivalent-weights particle filter in a high-dimensional system. *Quarterly Journal of the Royal Meteorological Society*, 141(687):484–503, 2015.

Christophe Andrieu, Arnaud Doucet, and Roman Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010. ISSN 1467-9868. doi: 10.1111/j.1467-9868.2009.00736.x.

Thomas Bengtsson, Peter Bickel, and Bo Li. Curse-of-dimensionality revisited: collapse of the particle filter in very large scale systems. In *Probability and Statistics: Essays in Honor of David A. Freedman*, pages 316–334. Institute of Mathematical Statistics, 2008.

Alexandros Beskos, Dan Crisan, and Ajay Jasra. On the stability of sequential Monte Carlo methods in high dimensions. *The Annals of Applied Probability*, 24(4):1396–1445, 2014a.

Alexandros Beskos, Dan O Crisan, Ajay Jasra, and Nick Whiteley. Error bounds and normalising constants for sequential Monte Carlo samplers in high dimensions. *Advances in Applied Probability*, 46(1):279–306, 2014b. doi: https://doi.org/10.1017/S0001867800007047.

Alexandros Beskos, Dan Crisan, Ajay Jasra, Kengo Kamatani, and Yan Zhou. A stable particle filter for a class of high-dimensional state-space models. *Advances in Applied Probability*, 49(1):24–48, 2017.

Michael Betancourt. A conceptual introduction to Hamiltonian Monte Carlo. *arXiv preprint arXiv:1701.02434*, 2017.

Benjamin Bolker and Bryan Grenfell. Space, persistence and dynamics of measles epidemics. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 348(1325):309–320, 1995.

Alexandre Bouchard-Côté, Sebastian J Vollmer, and Arnaud Doucet. The bouncy particle sampler: a non-reversible rejection-free Markov chain Monte Carlo method. *Journal of the American Statistical Association*, (just-accepted), 2017.

Carles Bretó and Edward L Ionides. Compound Markov counting processes and their applications to modeling infinitesimally over-dispersed systems. *Stochastic Processes and their Applications*, 121(11):2571–2591, 2011.

Carles Bretó, Daihai He, Edward L Ionides, and Aaron A King. Time series analysis via mechanistic models. *The Annals of Applied Statistics*, pages 319–348, 2009.

Olivier Cappé, Simon J Godsill, and Eric Moulines. An overview of existing methods and recent advances in sequential Monte Carlo. *Proceedings of the IEEE*, 95(5):899–924, 2007.

Rong Chen, Xiaodong Wang, and Jun S Liu. Adaptive joint detection and decoding in flat-fading channels via mixture Kalman filtering. *IEEE transactions on Information Theory*, 46(6):2079–2094, 2000.

Nicolas Chopin. Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference. *Annals of statistics*, pages 2385–2411, 2004.

T.C. Clapp and S.J. Godsill. Fixed-lag smoothing using sequential importance sampling. *Bayesian statistics 6: Proceeding of the Sixth Valencia International Meeting*, 6:743–752, 1999.

William S Cleveland, Eric Grosse, and William M Shyu. Local regression models. In J.M. Chambers and T. Hastie, editors, *Statistical Models in S*, pages 309–376. Chapman and Hall, 1992.

Michael Creutz. Global Monte Carlo algorithms for many-fermion systems. *Physical Review D*, 38(4):1228, 1988.

Pierre Del Moral. *Feynman-Kac Formulae*. Springer New York, 2004.

Pierre Del Moral and Alice Guionnet. On the stability of interacting processes with applications to filtering and genetic algorithms. In *Annales de l'Institut Henri Poincaré (B) Probability and Statistics*, volume 37, pages 155–194, 2001.

Pierre Del Moral and Jean Jacod. Interacting particle filtering with discrete observations. In *Sequential Monte Carlo methods in practice*, pages 43–75. Springer, 2001.

Pierre Del Moral and Lawrence M Murray. Sequential Monte Carlo with highly informative observations. *SIAM/ASA Journal on Uncertainty Quantification*, 3 (1):969–997, 2015.

Peter J Diggle and Richard J Gratton. Monte Carlo methods of inference for implicit statistical models. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 193–227, 1984.

Vu Dinh, Arman Bilge, Cheng Zhang, IV Matsen, and A Frederick. Probabilistic path Hamiltonian Monte Carlo. *arXiv preprint arXiv:1702.07814*, 2017.

Arnaud Doucet and Adam M Johansen. A tutorial on particle filtering and smoothing: fifteen years later. In Dan Crisan and Boris Rozovskii, editors, *Oxford Handbook of Nonlinear Filtering*. Oxford University Press, 2011.

Arnaud Doucet, Nando De Freitas, and Neil Gordon. *Sequential Monte Carlo methods in practice*. Springer, 2001.

Arnaud Doucet, Mark Briers, and Stéphane Sénécal. Efficient block sampling strategies for sequential Monte Carlo methods. *Journal of Computational and Graphical Statistics*, 15(3):693–711, 2006.

Arnaud Doucet, MK Pitt, George Deligiannidis, and Robert Kohn. Efficient implementation of Markov chain Monte Carlo when using an unbiased likelihood estimator. *Biometrika*, 102(2):295–313, 2015.

Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid Monte Carlo. *Physics letters B*, 195(2):216–222, 1987.

Geir Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research: Oceans*, 99(C5):10143–10162, 1994.

Geir Evensen. The ensemble Kalman filter: theoretical formulation and practical implementation. *Ocean Dynamics*, 53(4):343–367, 2003.

Matteo Fasiolo, Natalya Pya, and Simon N Wood. A comparison of inferential methods for highly nonlinear state space models in ecology and epidemiology. *Statistical Science*, 31(1):96–118, 2016.

Neil J Gordon, David J Salmond, and Adrian FM Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings F (Radar and Signal Processing)*, volume 140, pages 107–113. IET, 1993.

Peter J Green and Antonietta Mira. Delayed rejection in reversible jump Metropolis–Hastings. *Biometrika*, 88(4):1035–1053, 2001.

W Keith Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

Daihai He, Edward L Ionides, and Aaron A King. Plug-and-play inference for disease dynamics: measles in large and small populations as a case study. *Journal of the Royal Society Interface*, 2009.

Matthew D Hoffman and Andrew Gelman. The No-U-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(1):1593–1623, 2014.

Peter L Houtekamer and Herschel L Mitchell. A sequential ensemble Kalman filter for atmospheric data assimilation. *Monthly Weather Review*, 129(1):123–137, 2001.

Edward L Ionides, Dao Nguyen, Yves Atchadé, Stilian Stoev, and Aaron A King. Inference for dynamic and latent variable models via iterated, perturbed Bayes maps. *Proceedings of the National Academy of Sciences*, 112(3):719–724, 2015.

Edward L Ionides, C Breto, J Park, R A Smith, and Aaron A King. Monte Carlo profile confidence intervals for dynamic systems. *Journal of The Royal Society Interface*, 14(132):20170126, 2017. doi: http://dx.doi.org/10.1098/rsif.2017.0126.

Adam M Johansen and Arnaud Doucet. A note on auxiliary particle filters. *Statistics & Probability Letters*, 78(12):1498–1504, 2008.

Lyle V Jones. *The Collected Works of John W. Tukey: Philosophy and Principles of Data Analysis 1965-1986*, volume 4. CRC Press, 1987.

S. Karlin and H. M. Taylor. *A Second Course in Stochastic Processes.* Academic Press, New York, 1981.

Hiroaki Kitano. Computational systems biology. *Nature*, 420(6912):206, 2002.

Peter E Kloeden and Eckhard Platen. *Numerical Solution of Stochastic Differential Equations.* Springer-Verlag, Berlin Heidelberg, 3 edition, 1999.

Augustine Kong, Jun S Liu, and Wing Hung Wong. Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association*, 89(425):278–288, 1994.

Jing Lei, Peter Bickel, and Chris Snyder. Comparison of ensemble Kalman filters under non-Gaussianity. *Monthly Weather Review*, 138(4):1293–1306, 2010.

Benedict Leimkuhler and Sebastian Reich. *Simulating Hamiltonian dynamics*, volume 14. Cambridge university press, 2004.

Daniel Levy, Matthew D Hoffman, and Jascha Sohl-Dickstein. Generalizing Hamiltonian Monte Carlo with neural networks. *arXiv preprint arXiv:1711.09268*, 2017.

Ming Lin, Rong Chen, and Jun S Liu. Lookahead strategies for sequential Monte Carlo. *Statistical Science*, 28(1):69–94, 2013.

J Liouville. Note on the theory of the variation of arbitrary constants. *J. Math. Pure. Appl.*, 3:342–349, 1838.

Jun S Liu. *Monte Carlo strategies in scientific computing.* Springer Science & Business Media, 2008.

Robert N Miller, Everett F Carter, and Sally T Blue. Data assimilation into nonlinear stochastic models. *Tellus A: Dynamic Meteorology and Oceanography*, 51(2):167–194, 1999.

Antonietta Mira et al. On metropolis-hastings algorithms with delayed rejection. *Metron*, 59(3-4):231–241, 2001.

Radford Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2, 2011.

Radford M Neal. An improved acceptance procedure for the hybrid Monte Carlo algorithm. *Journal of Computational Physics*, 111(1):194–203, 1994.

Radford M Neal. Annealed importance sampling. *Statistics and Computing*, 11 (2):125–139, 2001.

Akihiko Nishimura, David Dunson, and Jianfeng Lu. Discontinuous Hamiltonian Monte Carlo for sampling discrete parameters. *arXiv preprint arXiv:1705.08510*, 2017.

Jamie Owen, Darren J Wilkinson, and Colin S Gillespie. Scalable inference for Markov processes with intractable likelihoods. *Statistics and Computing*, 25(1): 145–156, 2015.

Elias AJF Peters et al. Rejection-free Monte Carlo sampling for general potentials. *Physical Review E*, 85(2):026703, 2012.

Michael K Pitt and Neil Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599, 1999.

Martyn Plummer, Nicky Best, Kate Cowles, and Karen Vines. Coda: Convergence diagnosis and output analysis for mcmc. *R News*, 6(1):7–11, 2006. URL https://journal.r-project.org/archive/.

Patrick Rebeschini and Ramon Van Handel. Can local particle filters beat the curse of dimensionality? *The Annals of Applied Probability*, 25(5):2809–2866, 2015.

Gareth O Roberts and Jeffrey S Rosenthal. Optimal scaling of discrete approximations to Langevin diffusions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(1):255–268, 1998.

Gareth O Roberts and Richard L Tweedie. Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli*, pages 341–363, 1996.

Gareth O Roberts, Andrew Gelman, and Walter R Gilks. Weak convergence and optimal scaling of random walk Metropolis algorithms. *The annals of applied probability*, 7(1):110–120, 1997.

L Chris G Rogers and David Williams. *Diffusions, Markov processes and martingales: Volume 2, Itô calculus*, volume 2. Cambridge university press, 1994.

JC Sexton and DH Weingarten. Hamiltonian evolution for the hybrid Monte Carlo algorithm. *Nuclear Physics B*, 380(3):665–677, 1992.

Filippo Simini, Marta C González, Amos Maritan, and Albert-László Barabási. A universal model for mobility and migration patterns. *Nature*, 484(7392):96–100, 2012.

Richard A Smith, Edward L Ionides, and Aaron A King. Infectious disease dynamics inferred from genetic data via sequential Monte Carlo. *Molecular biology and evolution*, 34(8):2065–2084, 2017.

Chris Snyder, Thomas Bengtsson, Peter Bickel, and Jeff Anderson. Obstacles to high-dimensional particle filtering. *Monthly Weather Review*, 136(12):4629–4640, 2008.

Andreas Svensson, Thomas B Schön, and Fredrik Lindsten. Learning of state-space models with highly informative observations: a tempered sequential Monte Carlo solution. *Mechanical Systems and Signal Processing*, 104:915–928, 2018.

Luke Tierney and Antonietta Mira. Some adaptive Monte Carlo methods for Bayesian inference. *Statistics in medicine*, 18(1718):2507–2515, 1999.

Paul Vanetti, Alexandre Bouchard-Côté, George Deligiannidis, and Arnaud Doucet. Piecewise deterministic Markov chain Monte Carlo. *arXiv preprint arXiv:1707.05296*, 2017.

Christelle Vergé, Cyrille Dubarry, Pierre Del Moral, and Eric Moulines. On parallel implementation of sequential Monte Carlo methods: the island particle model. *Statistics and Computing*, 25(2):243–260, 2015.

Yingcun Xia, Ottar N Bjørnstad, and Bryan T Grenfell. Measles metapopulation dynamics: a gravity model for epidemiological coupling and dynamics. *The American Naturalist*, 164(2):267–281, 2004.

Yizhe Zhang, Changyou Chen, Ricardo Henao, and Lawrence Carin. Laplacian Hamiltonian Monte Carlo. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 98–114. Springer, 2016.